

Projects Manual for Developing Web-Enabled Decision Support Systems

Using Access, VB .NET and ASP .NET

Sandra D. Ekşioğlu

**Industrial and Systems Engineering
Mississippi State University, Mississippi State**

Abhijit A. Pol

**Computer and Information Science and Engineering
University of Florida, Gainesville**

Ravindra K. Ahuja

**Industrial and Systems Engineering
University of Florida, Gainesville &
Innovative Scheduling, Inc. Gainesville**

Contents

PROJECT 1	Book Store Database
PROJECT 2	Library Operations
PROJECT 3	Housing Department
PROJECT 4	Course Timetabling
PROJECT 5	Marble Factory
PROJECT 6	Regional Utilities
PROJECT 7	Decision Rules for Race Horse Wagering
PROJECT 8	Tennis Ranking System
PROJECT 9	Redesigning the MBA curriculum
PROJECT 10	Inventory Management at Blue Bell
PROJECT 11	Selecting an All-Star Team
PROJECT 12	Farm Management
PROJECT 13	The Annual Bolder Boulder Race
PROJECT 14	Supply Chain Design
PROJECT 15	Inventory Routing Problem
PROJECT 16	Billing System at University of North Carolina
PROJECT 17	Pharmaceutical Care System
PROJECT 18	Internet and Exchange Billing
PROJECT 19	Database for Course Timetabling
PROJECT 20	Outdoor Patient Division Management
PROJECT 21	Database for a Blood Bank

PROJECT 22	Equipment Maintenance Database
PROJECT 23	Spare Parts Management
PROJECT 24	Allocating Crude Oil Supply to Refinery Tanks
PROJECT 25	Job Scheduling Using Heuristic Approach
PROJECT 26	Personnel Career Management
PROJECT 27	Management of Health Care Services
PROJECT 28	Tender Evaluation Using a Decision Support System
PROJECT 29	Incentive Bonus Database
PROJECT 30	Performance Based Compensation in an Organization
PROJECT 31	DSS for an Electric Utilities Company: Compliance with Clean Air Act
PROJECT 32	Wood Panel Manufacturing
PROJECT 33	Knowledge Management Database
PROJECT 34	Managing the Outdoor Patient Division in a Hospital
PROJECT 35	Water Treatment Facilities in a Municipal Area
PROJECT 36	Asset and Liability Management
PROJECT 37	Personalization in Data Mining
PROJECT 38	International Student Information System
PROJECT 39	Vet-Med Decision Support System
PROJECT 40	CD Warehouse Management
PROJECT 41	Emerald Realty Group Inc
PROJECT 42	Passenger Reservation System
PROJECT 43	Decision Support System for a Financial Institution
PROJECT 44	Decision Support System for Taiwan's Federal Aviation Administration
PROJECT 45	Decision Support System for a Car Rental Company
PROJECT 46	Human Resources Management at Shand's Hospital
PROJECT 47	Information System for the "Clean-and-Go" Shop
PROJECT 48	Information System for a Movie Theater
PROJECT 49	Information System for an On-Line Electronics Shop
PROJECT 50	Information System for the "Mediterranean Food" Store

Problem Description

A small sized bookstore has been keeping track of its business mainly on paper. The owner is planning to grow her business. She would like to improve the bookkeeping, and as a result provide better service to customers, through a state-of-the-art database system. As she is not familiar with recent technology and tools, she would like to hire a college student who would develop such a system over the next summer.

The new system will help the bookstore maintain details about books, publishers, customers, receipts/payments, etc. A book may be a textbook, a novel, a comic, a children's book, or a cookbook. Publishers are the suppliers of the books. The bookstore buys books from different publishers. Typical customers of the store are libraries, institutions, or individuals such as students, faculty, and others. Customers may open an account with the store if they wish and receive a customer number. The bookstore is planning to build a website that will allow customers to perform certain activities on-line.

When customers place an order on-line, they can pay either by a major credit card or store credit. If the transaction takes place in the store, then customers can pay by cash or check in addition to credit card and store credit. If the required number of copies is available for on-line orders, the store sends the books and updates the stock level. If the required number of copies is not available, the store may send a partial consignment, provided that it is acceptable to the customer.

The bookstore owner would like to send information about new books, new editions of a book, or deals to customers based on their profiles. If a customer were a faculty member at a university, then the owner would like to offer a free copy of a new textbook or a new edition of an existing textbook. If a customer were a student who liked reading science fiction, the owner would like to send monthly notices about new releases. Thus, the system should be able to identify potential customers of new books and new editions and generate reports that contain their contact information.

The owner of the book store would also like to use this system as decision support tool. Every month she receives a list of new books from the publishers, and she has to decide which ones to order and how many. Before she decides, she checks her database and goes through the sales data for the past six months to see how many books of that particular author were sold, how many books on related topics were sold, how many books in similar areas were returned, etc. Moreover, she has to make decisions about her existing stock. The new system will prepare every week a list of books that fall under a certain level of availability. She then has to decide whether she wants to replenish her stock or not. In addition to checking her sales data she also checks the reviews received for that book, reviews for all books by that author, reviews on similar books by other authors, etc. After checking all this information she will decide how many copies to order, if any.

Database Design

The following are the main entity types of the bookstore database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Customer*: The main attributes are customer identification number, name, address, reading preferences, and profession.
2. *Customer History*: The main attributes are history identification number, ISBN of the book ordered, date of order, date of delivery, quantity delivered, and comments.
3. *Book*: The main attributes are ISBN, title, author, amount in the inventory, price, subject area, and year of publication.
4. *Publisher*: The main attributes are publisher identification number, name, address, and contact number.
5. *Transaction*: The main attributes are transaction number, number of books sold/bought, and amount of money involved in the transaction.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Every holiday season, the bookstore sends coupons to the top 10% of its customers. Customers that have spent the most during the current year make the top of the list. Generate a list with names and addresses of these customers.
2. College professors are targeted when advertising new textbooks. List all customers who are college professors.
3. Students are the main customers for textbooks. The bookstore informs the students about promotions on textbooks, sends out coupons, and advertises new textbook editions. List all customers of the bookstore who are currently students in the nearby university.
4. Developing efficient search engines facilitates the process of retrieving the necessary information from the database. Create queries that do the following:
 - a. List all the books published by an author.
 - b. List all the books in a given subject area.
 - c. List the books that sold out quickly.
 - d. List the most popular authors.
 - e. List the month's/year's bestseller.
 - f. List the number of copies sold for a book in a specific subject area on a certain topic.
5. The new database helps the bookstore to manage the inventory in a more efficient way. List for each book the quantity on-hand and the quantity on-order.
6. The bookstore is interested to learn about customers' satisfaction with its service. List the names of the customers who canceled at least three orders in the last month.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions (e.g., buy/sell transactions, books, customers, etc). These forms allow the

user to add, update, and delete information about buy/sell transactions, books, customers, etc.

3. Create a form that allows the bookstore employees to browse through the names of the authors registered in the database. Create a subform to present for each author the list of books that they have published.
4. Create a form that browses the publisher's information. Create a subform that presents for each publisher a list of books that they offer.
5. You developed a few queries that enable retrieving information from the database. Create a form that allows the user to choose to run one of the queries. In this case you can use command buttons that, when clicked on, open one of the queries already created.
6. Create a form that allows the user to choose the name of a customer, say from a combo box. Once the customer's name is chosen, in a subform (included in this form) present all the buy transactions performed by this customer. Calculate the total amount of money the customer has spent so far.

Reports:

1. Report the name, address, profession, and amount of money spent by the top 10% customers of the current year. Customers that spend the most during the current year make the top of the list.
2. Report details about new books or new editions of books.
3. Report a list of the bestsellers of the year. Include in this list the book title, author name(s), and ISBN.
4. Report the most popular authors of the year. Include in this list author name(s), list of the books written by them, the publication year for each book, and the name of the publisher.
5. Report monthly statements for customers who have an outstanding balance.
6. Create a report that will be used at the end of each month to present the monthly costs/revenues of the bookstore. This report should present the costs/revenues from the transactions, grouped by publishers. Create another report that groups the costs/revenues by book. A similar report is used at the end of the year to present the total revenues generated. Present the total monthly/yearly balance of the business.

Visual Basic.NET Application Development

The main users of this database will be the bookstore employees. When users start using the application, they are asked to enter their user name and password in order to log in to the database. In the case of new users, they should be allowed to sign up. After successful login, we would like to display a list of activities they might like to perform. It is up to you to group the activities together. The following is an example of a list similar to what you will create. The list consists of Books, Customers, and Publishers.

- If users choose the option "Books," a new form opens that provides a list of activities that they can perform that involve books, such as the following: add a new book, update the information about a current book, delete the information about a book from the database, search the database and retrieve information about a particular book, etc.

- If users choose the option “Customers,” a new form opens that provides a list of activities that they can perform that involve customers, such as the following: add a new customer, update the information about a current customer, delete the information about a customer from the database, set an order, cancel an order, search the database and retrieve information about a particular customer such as current balance, etc.
- If users choose the option “Publishers,” a new form opens that provides a list of activities that they can perform that involve publishers, such as the following: add a new publisher, update the information about a current publisher, delete the information about a publisher from the database, search the database and retrieve information about a particular publisher such as a list of books provided, view the monthly/yearly revenues generated, set an order, cancel an order, etc.

Web Extension

We mentioned in the project description that the bookstore customers may open an account if they wish. In this case, the bookstore will provide them with a customer number. With their customer number and a password that they will set, the customers will be able to log in to the database using their PC at home. They will be able to search for books, place orders, cancel orders, check their account status, update their personal information such as name and address, and submit reviews about books they have read. The customers, however, should not be allowed to add/update/delete information about other customers, books, or publishers.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above.

2 *Library Operations*

Problem Description

The main library at your university is having problems with its current system and is interested in developing a new system that will operate faster. The library carries the following items: books, journals, conference proceedings, reference textbooks, and copies of some recorded lectures on DVD. These items are loaned to members of various categories.

Assume that the university currently has 20,000 undergraduate students, 6,000 graduate students, 5,000 faculty members, 4,000 staff, 500 part-time students, and 100 visiting scholars, and all are members of the main library. Also assume that the library has a total of 250,000 items. The university issues all members an ID that can also be used at the library. Students and visiting scholars have to renew their ID every semester. Faculty and staff ID are valid as long as they are still employed.

A member who borrows a DVD cannot take it outside the library and has to use the computers in the library to watch it. DVDs can be borrowed for a maximum of three hours. Journals and conference proceedings can be borrowed for two days, and reference textbooks can be taken on loan only overnight. Faculty and graduate students can borrow books for a period of three weeks; other students and visiting scholars can borrow them for two weeks; and staff can borrow books for only 1 week. The library charges its members for returning an item late. The following table presents the amount charged for each item that is overdue:

Item	Overdue charge
Books	\$1/day
Journals	\$5/day
Conference Proceedings	\$5/day
DVDs	\$1/hour
Reference Textbooks	\$1/hour

The library has restrictions on the number of items that a member can borrow at a time. The following table presents the different member categories, the items they can borrow, and the maximum number of items they can borrow.

Member Category	Items that can be borrowed	Number of items
Faculty	Book, Journal, Conference Proceedings	20
Graduate student	Book, Journal, Reference Textbook, DVD	15
Undergraduate student	Book, Reference Textbook, DVD	10
Visiting scholar	Book, Journal, Conference Proceedings	5
Part-time student	Book, DVD	5
Staff	Book	3

The library follows an international system for classification of items. Thus the coding scheme is fixed and cannot be changed arbitrarily. The classification is done subject-wise. For example, consider the number 521.43. The number 521 indicates Industrial Engineering, and 43 a subdivision within Industrial Engineering. Every item also has a unique access code. Using this code the librarian will be able to identify the type of the

item (book, journal, DVD, etc.), the title, the author(s), and whether the item is on loan, overdue, or on shelf.

Database Design

The following are the main entity types of the library database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Member*: The main attributes are member identification number, name, address, classification (faculty, graduate student, undergraduate student, visiting scholar, part-time student, and staff), etc.
2. *Member History*: The main attributes are history identification number, date the book was borrowed, due date, return date, amount charged in the case that the book was overdue, etc.
3. *Item*: An item in this database could be a book, journal, DVD, or conference proceedings. The main attributes of the entity type *book* are ISBN, title, author, number of copies owned, number of copies in the shelves, number of copies borrowed, subject area, year of publication, keywords, the condition in which the book stands (good/poor), etc. The main attributes of the entity type *journal* are item number, title, subject number, number of times published in a year, name of the research society affiliated with, etc. The main attributes of the entity type DVD are item number, producer, title, comments, subject, etc. The main attributes of the entity type *conference proceedings* are item number, title, date the conference was held, etc.
4. *Employee*: The main attributes are employee identification number, name, address, date hired, education, etc.
5. *Publisher*: The main attributes are publisher identification number, name, address, contact number, etc.
6. *Transaction*: There are two main transaction types performed in the library. One is the buy-transaction, which involves buying books from publishers, and the other is lend-transactions, which involves lending books to members. The main attributes of the *buy-transaction* entity type are transaction number, transaction amount (e.g., number of items bought), amount of money involved in the transaction, order date, order receiving date, etc. The main attributes of the *lend-transaction* entity type are transaction number, transaction amount, borrowing date, return date, due date, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The library is concerned about not having an item on the shelves when members ask for it. The library is interested to know for each item how many copies are on the shelves on average. Also, they need to know how frequently (how many times in a month) a particular item is reserved. This, and other information you could think about

and present in here, will help the library to decide whether they need to purchase more copies of a particular item.

2. The library is concerned about having items that are overdue, as overdue items disrupt the planning process. The library sends an e-mail (or mail, based on the member's preference) notifying the member that the item is overdue and indicating the amount of fine due. List the members who have overdue items. The list should include the members' name, address, and overdue amount, and title, ISBN, and author of the items overdue.
3. As mentioned above, the overdue items disrupt the planning process of the library. From a recent study performed, it was concluded that some of the members have the habit of returning an item late. The library wants to list the names of these members. The library is planning to restrict their rights for borrowing items.
4. There are restrictions about the number of items that members can borrow from the library. Therefore, when members want to borrow a new item, the database should be checked to see how many items they have already borrowed. List all items currently borrowed by a member.
5. At the beginning of every year, the library checks the condition of each book and sends orders to the publishers to replace the books that are in poor condition. List the ISBN, title, author, and publisher name and contact number for each book that is in poor condition.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: buy-lend transactions, items, members, employees, publishers, etc. These forms allow the user to add, update, and delete information about transactions, items, members, publishers, etc.
3. Create a form that allows the user to search the database and retrieve information about an item of interest. The search can be done by title, subject, author, or keyword. The search should list all relevant items with their status (on-loan or on-shelf). If the item has multiple copies, the system should tell which copies are on-loan, which are overdue, and which are on-shelf. Once the item is located, the user should be able to reserve the item. Keep in mind the following: if there are no copies on-shelf, then the item cannot be reserved; there are restrictions on how many items members can borrow and how many days they can borrow the item for (details are given in the problem description).
4. Create a form that allows the user to browse through the name of the authors registered in the database. Create a subform to present for each author the list of books that they have published.
5. Create a form that browses the publishers' information. Create a subform that presents for each publisher a list of books that they offer.
6. Create a form that allows the user to choose the name of a member, say from a combo box. Once the member's name is chosen, in a subform (included in this form) present all the items borrowed by this member. Calculate and present the total amount of money due because of overdue items.

Reports:

1. Report the name, address, and contact number of the publishers that the library does business with. For each publisher, present the total amount of money from buy-transactions in the current year.
2. For each overdue item, prepare a report that includes the following: the title of the book, the author name, the date the book was borrowed, the date the book was due, the amount of money charged, and the name and address of the member who borrowed the book. This report will be mailed to the library member as a reminder to return the book.
3. Prepare a report with all the books that are reserved and will be picked up on a particular day.
4. Report a detailed list on new books or new editions of books. The list should present ISBN, title, author, and publisher.
5. Report a list of the books that are not in good condition. Include in this list the book title, author(s) name, ISBN, and detailed information about the publisher.
6. Report monthly statements for members who have outstanding balance.
7. Create a report for the monthly costs/revenues of the library. The costs mainly come from buying items for the library, and the revenues come from overdue charges. Group the costs by publisher. Create another report that groups the costs/revenues by book. Present the total monthly/yearly balance of the library.

Visual Basic.NET Application Development

Library employees are the main users of this database. When users start the application, they are asked to enter a user name and password in order to log in to the database. In the case of new employees, they should be allowed to sign up. After successful login, we would like to display a list of activities that employees might like to perform. It is up to you to group the activities together. The following is an example of a list similar to what you will create. The list consists of Items, Members, Employees, Publishers, and Transactions.

- An item in the database could be a book, journal, DVD, etc. Therefore, if users choose the option “Item,” a new form opens that allows them to choose books, journals, etc. Once one of these options is selected, users can perform several activities, such as add a new item, update the information about a current item, delete the information about an item from the database, search the database and retrieve information about a particular item, etc.
- If users choose the option “Employees,” a new form opens that provides a list of activities that they can perform that involve employees, such as add a new employee, update the information about a current employee, delete the information about an employee from the database, and search the database about a particular employee (for example, search for the contact number of an employee), etc.
- If users choose the option “Publishers,” a new form opens that provides a list of activities that they can perform that involve publishers, such as add a new publisher, update the information about a current publisher, delete the information about a publisher from the database, search the database and retrieve information about a particular publisher.

- If users choose the option “Transactions,” a new form opens that provides a list of activities that they can perform that involve buy and loan transactions, such as list the monthly/yearly expenses generated from buy-transactions, set an order, cancel an order, etc.

Web Extension

The library allows its members to browse through its database on-line from their PC at home or in their office or from the computers in the library. Members can use their membership number to log in to the database. They are able to search for items by title, subject, author, or a keyword. The search lists all relevant items and their status. Members can also check their account status, access information about the publishers, and update their personal information, such as name, address, etc. Members, however, should not be allowed to add/update/delete information about other members, books, or publishers.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above.

3 *Housing Department*

Problem Description

The University Housing office receives many applications from graduate and married students requesting an apartment on campus. There are five housing villages in campus, and each village has about 500 apartments. Each apartment falls into one of the categories given in the table below. Village 1 has apartments in categories 1 and 3 only. The apartments in villages 2 and 3 are typically in categories 2 and 4. The other two villages have apartments in categories 5-12.

Category	Central AC / Window Unit / No AC (C / W / N)	Single / Double (S / D)	Furnished (Y / N)	Dish Washer (Y / N)
1	C	D	N	Y
2	C	D	N	N
3	C	S	N	Y
4	C	S	N	N
5	W	D	Y	N
6	W	D	N	N
7	W	S	Y	N
8	W	S	N	N
9	N	D	Y	N
10	N	D	N	N
11	N	S	Y	N
12	N	S	N	N

To be eligible to apply, students must be graduate students or married undergraduate/graduate students. Priorities in assigning an apartment are given based on marital status, degree pursued, and application date. Married students have the highest priority whether they are graduate or undergraduate students. Doctoral (PhD) students have higher priority than masters (MS) students.

Applicants can mark three preferences for the village and apartment type desired. Based on availability, they will be offered an apartment. Applicants can reject the offer. After three rejections, they will have to apply again. A non-refundable fee is required for each application. Only one application is permitted for each student. Married couples, however, can have two separate applications if they are both students. Single students are not allowed to share apartments. A refundable fee is required upon moving in. Notification for vacating premises is required one month prior to the move-out date. After graduation, students are allowed to stay in the apartment at most one more semester.

Database Design

The following are the main entity types of the Housing department database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Resident*. The main attributes are identification number, name, gender, marital status, college, and department of the family head; address, telephone number, etc.
2. *Applicant* (a student who has applied for on-campus housing but has not yet been assigned an apartment): The main attributes are identification number, name, gender, address, telephone number, marital status, college, department, preference about the village (the name of the village they want to stay in), preference about the apartment (single bedroom, double bedroom, furnished, unfurnished, etc.), etc.
3. *Maintenance Request*. The main attributes are address (this includes building and apartment numbers), description of the maintenance problem, name of the resident, submission date, date the maintenance problem was fixed, name of the employee responsible, etc.
4. *Apartment*. The main attributes are address (it consists of village name, building number, and apartment number), number of bedrooms, air-conditioning status (central AC, window unit AC, or no AC), furniture status (furnished or unfurnished), dish washer status (whether it has a dish washer), etc.

Access Application Development

The following are some of the queries, forms, and reports one can create to increase the functionality of the database.

Queries:

1. At the end of each semester, the Housing department prepares a list of all the apartments that will be available for the next semester. The list provides details about the address, air conditioning status, furniture status, etc. of all the apartments available.
2. In assigning an apartment, the Housing department prioritizes married couples (versus unmarried), PhD students (versus MS students), and early applications. List the identification number, name, address, and telephone number of the applicants who need an apartment next semester. Sort the information about the applicants based on their priority level.
3. Most of the apartments are vacated at the end of the semester. On the checkout date, an employee should visit the apartment and check its status and inventory. The Housing department has a limited number of employees, and, therefore, for scheduling purposes it is necessary to know in advance which apartments will be vacated and when the checkout date is. List the addresses of the apartments that will be vacated at the end of the semester together with the checkout dates.
4. In order to help the department with scheduling for apartment maintenance service, prepare a list of the apartments that have submitted a maintenance order for a particular date.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: residents, applicants, maintenance orders, etc. These forms allow the user to add, update, and delete information about residents, applicants, maintenance orders, etc.

3. Create a form that allows the user to check the availability of apartments in a particular category. One way that can be followed to build such a form is by using a combo box to present all the apartment categories. Once a category is chosen from the combo box, a subform (included in this form) presents the address of all the available apartments in this category.
4. Create a form that allows the user to browse through the table that contains information about the residents. Create a subform that presents for each resident the corresponding billing information. Present for each resident the total amount of money paid to the Housing department so far.
5. Create a form that allows the user to browse through the table that contains information about the applicants. Create a subform that presents for each applicant his/her marital status, degree pursued, application date, and apartment preferences.
6. The Housing department is performing a demographical study of their residents. They are interested to know about the changes in the following:
 - a. The number of married residents (per year) during the last 10 years
 - b. The number of undergraduate married residents (per year) during the last 10 years
 - c. The number of unmarried residents (per year) during the last 10 years
 - d. The number of residents pursuing a PhD degree (per year) during the last 10 years
 - e. The number of residents pursuing an MS degree (per year) during the last 10 years
 - f. The number of female residents (per year) during the last 10 years
7. Create a form that allows the user to browse through the apartments' table. Create a subform that presents for each apartment the maintenance orders that are still due.
8. Create a form that allows the user to check the status of maintenance requests for a particular day.

Reports:

1. Every day, a report with the maintenance requests due and the requests that were not handled (overdue) is presented to the maintenance department. The report contains the name of the resident, apartment address, submission date, due date, and a short description of the problem to be handled.
2. The Housing department is considering remodeling some of the apartments. Priority will be given to the apartments and villages that have had the largest number of maintenance problems during the last year. Prepare charts that present the number of maintenance requests placed by each village and by each apartment type per month during the last year.
3. Prepare a report consisting of the address and characteristics (such as, number of bedrooms, AC availability, etc.) of all the apartments that will be available next semester.
4. Report the name and current address of the applicants that need an apartment next semester. Rank the applicants based on marital status and degree pursued.

5. Every month, the Housing department mails to its residents a statement of their financial obligations (rent payments, electricity payments, etc.).
 - a. Create a report that presents the monthly financial obligations of each resident.
 - b. Use the label wizard to create a report that contains labels with the addresses of the residents.

Visual Basic.NET Application Development

This database will mainly be used by the Housing department employees. Users are asked to enter a user name and password to log in to the database. The new users are allowed to sign up. After successful login, users choose from a list the activity that they would like to perform. It is up to you to group the activities together. The following is an example of a list similar to what you will create. The list of activities consists of Resident, Applicant, Maintenance, Apartment, and Demographical Studies.

- If the option “Resident” is chosen, a new form opens that provides a list of activities that involve residents, such as add a new resident, update the information about a current resident, delete the information about a resident from the database, check a resident’s financial status, maintenance requests submitted, etc.
- If the option “Applicant” is chosen, a new form opens that provides a list of activities that involve applicants, such as add a new applicant, update the information about a current applicant, delete the information about an applicant from the database, check the application status, etc.
- If the option “Maintenance” is chosen, a new form opens that provides a list of activities that involve maintenance services, such as add a new maintenance request, update the status of a request, list the requests submitted or due on a particular date, etc.
- If the option “Demographical Studies” is chosen, a new form opens that allows the user to choose the chart to be displayed (list the charts already built in part 6 of the section about the reports).

Web Extension

The Housing department allows the residents and applicants to browse through their database on-line from their PC at home or in the office. The residents/applicants can use the member identification number to log in to the database. The residents should be able to check their financial status, update their account information, and submit maintenance requests on-line. The applicants should be able to apply and check their application status on-line.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above.

Problem Description

Preparing course timetables is an important and difficult task for academic institutions. Given (a) a list of courses offered that semester, (b) a list of faculty members, (c) a list of students registered for the courses, (d) a list of classrooms available, and (e) a list of periods in a week, the task is to prepare a timetable for the current semester. To make this task easier, the timetable from the previous year/semester will be taken as an input. If there are no changes, the same timetable can be used. Possible changes should be identified and it should be ensured that they would not cause any conflicts. For example, if the number of students taking a course changes, this may force a change in the classroom assigned due to room capacity; if the instructor of a course changes, this may also cause a conflict in that instructor's weekly schedule.

The courses are grouped by college and department, and within a department they are grouped into four different levels (i.e., courses offered to freshmen, ..., seniors). In a semester, a group of students follows a set of courses and hence these courses must be assigned to different time periods. The business rules to be followed in building a course timetable are identified as follows:

1. All courses must be assigned to the required number of periods.
2. An instructor cannot be assigned to more than one course in a period.
3. Courses belonging to the same group cannot be assigned to the same time period.
4. A course can only be assigned to an available room in a period if the number of students taking that course is at least 75% and at most 105% of room capacity.
5. The number of courses assigned in a particular period cannot be greater than the number of rooms available.

Database Design

The following are the main entity types of the academic institution database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Instructor*: The main attributes are identification number, name, address, name of the department and college s/he is working for, area of specialty, etc.
2. *Course*: The main attributes are course number, name, prerequisites, number of credit hours, group of courses it belongs to, etc.
3. *Student*: The main attributes are identification number, name, address, birthday, gender, department and college s/he is in, etc.
4. *Classroom*: The main attributes are room number, name of the building, capacity, and description of any special feature (for example, media equipment, computers, video, etc.).
5. *Period*: The main attributes are period number, start time, and end time.

6. *Assignment*: This is an associative entity. Its attributes are number of students taking a particular course taught by a particular professor during a particular time period.

Note that the relation between students and courses is a many-to-many relationship. Students get a grade for each course they take.

Access Application Development

The following are some of the queries, forms, and reports that students can create in order to increase the functionality of the database:

Queries:

We already mentioned that the timetable from the previous year/semester is taken as an input in building the current semester's timetable. The schedule would change in the following cases: (a) A professor prefers to teach in a different classroom in a different time period; (b) A professor is offering a new class; (c) The number of students registered for the class is bigger than classroom capacity; etc. The following queries facilitate the process of updating the course timetable.

1. Professors are interested to know their weekly schedule. Create a query that prompts for the professors' name and returns their weekly schedule.
2. Create a query that prompts for students' identification numbers and returns their timetables for the current semester.
3. Create a query that prompts for the name of a course and returns its current schedule.
4. For each course offered in the current semester, present the total number of students registered, the classroom capacity, and the classroom description. This query will help to identify possible changes to the timetable due to the class capacity or equipment (computers, video, etc.) availability.
5. Every semester there are a number of new courses offered that have to be assigned to a period/room.
 - a. List all the courses offered for the first time.
 - b. List the eligible periods/classrooms for an unassigned course. This list should adhere to all the business rules given above.
6. For each time period, list the addresses of the classrooms that are available. This list will be given to the professors. In the case that professors do not like their current schedules, they can re-allocate the class in one of the available period/classroom combinations.
7. Create a query that prompts for the name of a department and course level (freshmen, ..., senior) and returns the timetable of the courses for this particular group. The results from this query are used to check whether the courses belonging to this particular group are assigned to the same class period.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: courses, students, instructors, etc. These forms allow the user to add, update, and delete information about courses, students, instructors, etc.
3. Create a form that allows the user to browse through the courses offered in the current semester. Create a subform that presents for each course the corresponding timetable.
4. Create a form that allows the user to choose the name of a professor, say from a list box. Once the professor's name is chosen, use a subform to present that professor's timetable for the semester.
5. Create a form that allows the user to choose the name of a student from a combo box. Once the student's name is chosen, use a subform to present that student's timetable for the semester.
6. Create a form that presents all the available period/room combinations. For each room, present its capacity together with a list of equipment available.
7. Create a form that allows the user to choose a department/course level combination. For each combination, the timetable of the courses scheduled for the semester is presented.

Note: Create a logo for this academic institution and insert it in all the above forms. The background color of the forms created should be the same as the institution's colors (choose the colors yourself).

Reports:

1. For each of the courses offered in the current semester, report the corresponding timetable.
2. Report the name and address of the professors scheduled to teach in the current semester. Present the current schedule for each professor.
3. For each department/course level combination, report the current timetable.
4. For each student, present the corresponding timetables. The report should present the following: name of the courses the student is taking, period number, name of the professor that is teaching the course, and classroom number.
5. Report the currently available room/period combinations. For each room, present its capacity and equipment availability.
6. Report all courses involved in a schedule change during the current semester.

Visual Basic.NET Application Development

This database will mainly be used by the institution's employees. Users are asked to enter a user name and password to log in to the database. New users are allowed to sign up. After successful login, users choose from a list the activity that they would like to perform. It is up to you to group the activities together. The following is an example of a list similar to what you will create. The list of activities consists of Instructor, Student, Course, Room, and Timetable.

- If the option “Instructor” is chosen, a new form opens that provides a list of activities that involve instructors, such as add a new instructor, update the information about a current instructor, delete the information about an instructor from the database, check an instructor’s current timetable, etc.
- If the option “Student” is chosen, a new form opens that provides a list of activities that involve students, such as add a new student, update the information about a current student, delete the information about a student from the database, check the student’s current timetable, etc.
- If the option “Course” is chosen, a new form opens that provides a list of activities that involve courses, such as add a new course, update the information about a course, present its timetable, etc.
- If the option “Room” is chosen, a new form opens that provides a list of activities that involve rooms, such as check room capacity and equipment availability, present the periods when the room is busy and when it is available, list the classes that will be taught in this classroom, etc.
- If the option “Timetable” is chosen, a new form opens that provides a list of activities that involve timetables, such as update the current timetable; present the timetable of a professor, student, course and department/course level combination; etc.

Web Extension

The students and professors browse through the database on-line from their PC at home or in the office. They use their identification number to log in to the database. The students and professors should be able to check their weekly schedule. Neither one is allowed to update the current schedule. The professors, however, can request changes in the schedule. The database administrator is the only person who can actually update the current schedule.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above.

Problem Description

Your team is assigned to build a database system for a small factory in Venezuela, called “Marbles of Venezuela”. The factory produces different type of marbles that are sold in the Venezuelan or USA market. This is a small sized company that uses a simple database management (DBM) system. The factory keeps the information about final products, raw materials, and prices in simple Excel spreadsheets. In the process of extending the business, the management realized the need for re-organizing the database.

The factory produces different kinds of products that distinguish from each-other by their color, size, and shape. The two main products are Gems and Marbles. There are two different sizes of Gems and six different sizes of Marbles. Both products are offered in 25 different colors. Each colored Gem is classified as matte or luster Gem. Finished products are delivered in boxes. Each box contains products packed either in nets of different sizes, big plastic bags, or hard plastic holders.

The factory does not have a proper inventory management system. The inventories are handled in the following way: the manager knows roughly the amount of inventory for each final product. Based on previous experiences, the manager decides how much and when to produce. As the business is growing, it is getting difficult to keep track of all the inventories and decide wisely about the production schedule. The company is facing the problem of not being able to deliver the products on the due date. Their first step toward re-organizing themselves is building a proper database management system. The new system will enable them to keep track of the inventory level of final products and raw materials and help with production scheduling.

The new database management system keeps track of the inventory level, the backorder level (amount of final product orders not yet satisfied), and the number of outstanding orders (raw materials orders that are not received yet). Whenever an order for final products arrives, the inventory of final products should be reduced by the amount in the order. If the final product inventory is not enough to satisfy the order, then production should start. In this case, the raw material inventory should be checked. If there are not enough raw materials in stock, then an order for these raw materials should be placed.

Database Design

The following are the main entity types of the database of this company. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Employee*: The main attributes are SSN, name, address, profession, etc.
2. *Customer*: The main attributes are identification number, name, address, etc.
3. *Supplier*: The main attributes are identification number, name, address, etc.
4. *Raw material*: The main attributes are identification number, name, description, quantity on hand, unit price, etc.

5. *Final product:* The main attributes are identification number, name, product characteristics (color, size, etc.), quantity on hand, unit price, etc.

Note the following:

1. For each customer order, the company keeps information about quantity ordered, order date, due date, delivery date, and total amount of money charged.
2. For each (raw materials) order set, the company keeps information about quantity ordered, order date, delivery date, and amount to be paid.

Queries:

1. To help with the process of scheduling production and managing inventories, it is very important to know the level of on-hand inventories. Prepare a list consisting of product name (final product and raw materials), and corresponding inventory level.
2. The decision about how much raw material should be ordered depends not only on the amount of on-hand inventory, but on the level of outstanding orders as well. Prepare a list consisting of raw materials name, quantity-on-hand, and outstanding orders.
3. Different raw materials of different quantities are used to produce a particular product. Every time that a customer order arrives, the managers should do the following:
 - a. Identify what and how much raw material is needed
 - b. Check the level of on-hand inventory and outstanding orders for the raw materials needed
 - c. Decide how much to order to be able to satisfy the demand.

Create a query that prompts for the identification number of a particular product and returns the following: the name of all raw materials that are needed to produce the final product; for each raw material, the quantity needed, the level of on-hand inventory, and outstanding orders.

4. The company is looking into the possibility of establishing long-term relationships with its suppliers. To help with analyzing the data and drawing conclusions, provide the following:
 - a. For each supplier, list its name and total amount (in \$) of raw materials delivered during the last year.
 - b. For each order that is (or has been) overdue, list the name of the supplier and number of days overdue.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: products, raw materials, suppliers, customers, orders, etc. These forms allow the user to add, update, and delete information about products, raw materials, suppliers, customers, orders, etc.
3. Create a form that allows the user to browse through the information about the final products. Create a subform that presents (for the selected product) the name of the

raw materials needed to produce the product. Create another subform that presents (for the selected product), the level of on-hand inventory and backorders.

4. Create a form that allows the user to choose from a list the name of the desired raw material. Create a subform that presents the level of inventory on-hand and outstanding orders for the chosen raw material.
5. Create a form that allows the user to choose from a list (you can use a combo box) the name of a supplier. Upon selection, in an included subform, present the orders made to this supplier. Sort the orders based on the due date.
6. Create a form that allows the user to choose from a list (you can use a list box) the name of a customer. Upon selection, in an included subform, present the customer's outstanding orders.

Reports:

1. On a daily basis, the management requires a report with detailed information about the following:
 - a. *Backorders*. The report includes customer's name, quantity ordered, due date, total \$ amount due, etc.
 - b. *Orders to be delivered that day*. The report includes customer's name, address, quantity ordered, due date, total \$ amount due, etc.
 - c. *Outstanding orders*. The report includes supplier's name, phone number of the contact person, quantity ordered, delivery date, etc. Sort the information by due date.
2. Prepare the business' balance at the end of the month. The report should present the amount of money generated as revenues from sales, the amount of money spent in buying raw materials and on salaries, and the total amount of money earned. Itemize the revenues by customer and the expenses by supplier.
3. The company is interested to know about monthly/yearly sales for each final product. This information helps the management identify the following:
 - a. Potential problems in production lines
 - b. Growth trends.

Use the chart wizard to present monthly sales for each product during the last year.

4. The management is interested to learn more about customers and their preferences. Use the chart wizard in order to:
 - a. Present for each customer the monthly purchases made during the last year.
 - b. Identify five products that sold the most during the last year, and chart their monthly sales.
 - c. Identify the products that sold less during the last year, and chart their monthly sales.
5. Use the label wizard to prepare labels (that contain the address) for customers, employees, and suppliers.

Visual Basic.NET Application Development

This database will mainly be used by the company's managers. Users are asked to enter a user name and password to log in to the database. New users are allowed to sign up. After successful login, users choose from a list the activity that they would like to perform. It is up to you to group the activities together. The following is an example of a list similar to what you will create. The list of activities consists of Customer, Supplier, Raw Material, Product, Order, Data Analysis, etc.

- If the option "Customer" is chosen, a new form opens that provides a list of activities that involve customers, such as add a new customer, update the information about a current customer, delete the information about a customer from the database, check the status of the customer's orders, etc.
- If the option "Supplier" is chosen, a new form opens that provides a list of activities that involve suppliers, such as add a new supplier, update the information about a current supplier, delete the information about a supplier from the database, check the status of the orders sent to a supplier, etc.
- If the option "Raw Material" is chosen, a new form opens that provides a list of activities that involve raw materials, such as add a new raw material, delete the information about a raw material from the database, check the level of the on-hand inventory, identify the suppliers that provide this raw material, etc.
- If the option "Product" is chosen, a new form opens that provides a list of activities that involve products, such as add a new product, delete the information about a product from the database, check the level of the on-hand inventory, identify the customers of a product, chart the monthly sales during the last year, etc.
- If the option "Order" is chosen, a new form opens that provides a list of activities that involve orders, such as add a new order, update the information about an order from the database, check the due date of an order, etc.
- If the option "Data Analysis" is chosen, a new form opens that provides a list of analyses that can be performed, such as chart the monthly sales, chart the monthly expenses, chart the monthly revenues, etc.

Web Extension

The customers and suppliers browse through this database on-line from their PC at home or in the office. They use a login name and a password to enter the database. The customers should be able to (a) check the status of their order; (b) update their contact information or address; (c) check the product catalog; etc. The suppliers access the database in order to (a) update their contact information or address; (b) check the catalog of the products offered; etc. The suppliers are interested to know about the final products. This knowledge enables them to customize their orders. Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above.

6 Regional Utilities

Problem Description

GERU is a regional multi-service utility company that provides electricity (E), natural gas (NG), water (W), and telecommunications (T) services to its customers. GERU is interested in developing a system that will help manage its operations and enable its customers to track their energy consumption, check their payment history, and report power failures on-line.

The company has classified its customers into three major groups: domestic (D), commercial (C), and agricultural (A) customers. Currently, the company has about 4,500 domestic, 1,200 commercial, and 100 agricultural connections. Each connection has associated characteristics and rates. The following table provides the unit rates that the company charges. Note that the rates vary by connection and service type.

Utility	Rate (\$/unit)			Minimum monthly charge (\$)		
	D	C	A	D	C	A
E	0.015	0.0350	0.0250	10	100	50
NG	0.020	0.0400	0.0400	5	30	25
W	0.025	0.0550	0.0650	10	80	100
T	0.005	0.0065	0.0055	15	120	120

The customer's bill is calculated using the rates presented above. The minimum monthly charge is assessed each month in which the service is active, regardless of whether the service is used. This charge includes the cost of meter reading, managing customer accounts, maintaining facilities, system readiness, etc. The city utility tax is assessed on the following bills: (a) Electric (10%); (b) Natural Gas (10%); (c) Water (25%).

Customers are permitted to pay without the bill in the case of loss of bill, late bill delivery, etc. The amount they pay is based on the previous month's consumption. The balance is to be adjusted the next month, and interest needs to be added if the last month's bill was not paid in full. Customers may pay either by cash, check, credit card, or direct transfer from a bank account.

Database Design

The following are the main entity types of the database of this company and their corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Customer*: The main attributes are identification number, name, address, classification, type of services offered, sign-in date, etc.
2. *Service*: The main attributes are identification number, type, description, rate charged, etc.

Note that a customer has at least one type of connection. At the end of the month, each customer gets a statement that presents the monthly consumption. The statement itemizes the customer's expenses and presents the total \$ amount due and due date.

Queries:

1. At the end of each month, a statement is sent to all the customers. The statement presents the total consumption by service and total \$ amount due. Create a list that has the following information: customer name, customer's total consumption (by service), and total \$ amount due.
2. To analyze how the business is doing, the management needs the following information:
 - a. List the total number of new connections set per month during the last year.
 - b. List the monthly income that was collected from domestic, commercial, and industrial customers during the last year.
 - c. List the monthly income that was collected by electric, natural gas, water, and telecommunication services during the last year.
3. The management has realized that some of the customers have made it a habit to submit late payments. List the customers that always pay late.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: customers, services, etc. These forms allow the user to add, update, and delete information about customers, services, etc.
3. Create a form that allows browsing through the customers' information. Create a subform that presents for each customer the current month's electricity, water, and gas consumption.
4. Create a form that allows the user to choose from a combo box a service type (water, electricity, etc). Create a subform that lists the name of the customers that used this service and the corresponding monthly consumption.
5. On the last day of the month, the company calculates the total consumption by service as well as the total consumption by customer type. Create a form that allows the user to choose from a list box a date (month and year). Create a subform that presents the total consumption by service on that date. Create another subform that presents the total consumption by customer type on that date.

Reports:

1. Create the customer's monthly statement. The monthly statement includes the following information: customer name, address, itemized monthly consumption, unit rate charged, itemized \$ amount charged, total \$ amount due, and due date. Create a logo for this company. Add the logo to the monthly statement.
2. Report the business' balance for the current month. The report should include a date, total income, total expenses, and total revenues generated in the current month. Itemize expenses and revenues by service and customer type.

3. Prepare a list of the new connections made during the last month. The list should contain name and address of the customer and type of service(s) provided.
4. Prepare the following charts that help to analyze the performance of this business:
 - a. Chart the total income generated per month during the last year.
 - b. Chart the number of new connections made per month during the last year.
 - c. For each service type, chart the monthly consumption during the last year.
 - d. For each customer type, chart the monthly consumption during the last year.
 - e. Chart the revenues generated per year during the last ten years.
5. Use the label wizard to create labels with the addresses of customers.

Visual Basic.NET Application Development

This database will mainly be used by the employees. Users are asked to enter a user name and password to log in to the database. New users are allowed to sign up. After successful login, users choose from a list the activity that they would like to perform. It is up to you to group the activities together. The following is an example of a list similar to what you will create. The list of activities consists of Customer, Service, Data Analysis.

- If the option “Customer” is chosen, a new form opens that provides a list of activities that involve customers, such as add a new customer, update the information about a current customer, delete the information about a customer from the database, check the customer’s current balance, check the customer’s monthly consumption by service, etc.
- If the option “Service” is chosen, a new form opens that provides a list of activities that involve services, such as add a new service type, update the information about a service type (update unit rate charged) from the database, check the monthly consumption by service, check the monthly revenues by service, etc.
- If the option “Data Analysis” is chosen, a new form opens that provides a list of analyses that can be performed, such as chart the monthly income by service, chart the monthly expenses, chart the monthly revenues, etc.

Web Extension

The customers browse this database on-line from their PC at home or in the office. They use a login name and a password to enter the database. The customers should be able to do the following: (a) review their energy consumption and unit rate charged; (b) update their contact information or address; (c) check their monthly statement; etc. Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above.

Decision Rules for Race Horse Wagering

Problem Description

People have bet on the outcome of horse races for a very long time. They try to earn large profits from their bets by finding ways of outwitting other betters. A large number of books have been published on horse races, and each author claims that following his/her methods will lead to profits. So far you have not been able to make big profits using any of the existing methods. Therefore, you decide to build a system that will help you and others place sophisticated bets.

Horse racing is one of the most documented sports. There are thousands of pieces of data published on each upcoming race. This data is available to you, and you need to build a database to keep all this information. Once you build your betting support system, you need to test it on some sample data and report your observations.

Database Design

We present in here the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Horse*: The main attributes are identification number, name, birthday, height, weight, breed (Arabian, English, etc.), races attended, races won, injuries, etc.
2. *Owner* (of the horse): The main attributes are identification number, name, address, etc.
3. *Jockey*: The main attributes are social security number, name, address, weight, height, birthday, years of experience, race history, major achievements, etc.
4. *Racetrack*: The main attributes are name, location (address), type (dirt, turf, etc.), distance, etc.
5. *Race*: The main attributes are name, date, distance, number of times it is organized in a year, location, number of horses participating, number of jockeys participating, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of this database:

Queries:

1. It has been shown that there is a high probability that the horses that have won the race at least once will win the race again. Therefore, there is a tendency in the public to bet on the horses that won at least once. To help with the decision about which horse to bet on, run a query that presents for each horse the following information:
 - a. Name, breed, age, height and weight.
 - b. Percentage of time the horse won a race.

- c. Average number of times the horse won.
- d. Percentage of time a horse won the races that were held on types of tracks such as today's track.
- e. Percentage of time a horse won a big race (races with attendance more than 30 horses).

Rank this information by percentage and average number of wins.

2. In fact, percentage and average number of wins are not very good indicators of how a horse will perform in the current race. However, this information combined with the information about the time of the last race, changes in weight carried, current speed rating, and previous class rating are a better indicator. Run a query that puts together this information.
3. People believe that the following are indicators of a good (forthcoming) performance:
 - a. The horse has not raced within one month of today's race.
 - b. The horse won at least two of its last five races.
 - c. The horse finished no worse than second in at least five of the last ten races.
 - d. The horse and its jockey have raced together in the last two years.
 - e. The horse has either raced and finished second or better in the last ten days or shown an exceptional workout in the last four days.

Run a query that lists the horses in each of the categories listed above.

4. The following are other (secondary) factors to be considered in choosing a horse to bet for:
 - a. Find the horse that was first at the last stretch and won the race by at least one and a half lengths.
 - b. Find the horses that in their last race did not finish in the money.
 - c. Find the horses that in their last race lost more than $\frac{3}{4}$ lengths in the stretch.
 - d. Find the horses that have a speed rating at today's distance within five points of the highest speed rating of all the competing horses in the past four races.
 - e. Find the horses that have participated in at least ten races during the last year.
5. Some people demonstrate interest about, or tend to be loyal to, a particular horse. Create a query that prompts for the identification number of a horse and returns the following:
 - a. The date and time of the last race the horse ran.
 - b. The time the horse made on the last race run that was at a distance close to today's distance.
6. People believe that there is a high probability a horse will not win a race given the following conditions:
 - a. The horse never won a race.
 - b. The horse did not win during the last (say, 10) races.
 - c. The horse never had a race at today's track.
 - d. The horse never ran today's distance at today's track.

Run a query that lists the horses in each of the categories listed above.

You can define other criteria as you feel necessary and run the above queries to get a list of horses for each race.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: horses, owners, jockeys, races, etc. These forms allow the user to add, update, and delete information about transactions, books, customers, etc.
3. Create a form that allows the user to browse through the name of the horses recorded in the database. Create a subform that presents for each horse the races run, the time made, and its placement.
4. Create a form that allows the user to choose (say, from a combo box) the name of an owner. Create a subform that presents the name of the horses owned by this owner.
5. Create a form that allows the user to choose (say, from a list box) the name of a jockey. Create a subform that presents the races the particular jockey ran, the name of the horse s/he run with, the time made, and her/his placement on each race.
6. You have already created a few queries that help to retrieve information from this database. Consider the third set of queries you built. These queries help identify horses that have a potential to win. One can use a command button to run each of these queries. Create a form titled, "Hints for Choosing a Horse." In this form put command buttons. When one of the buttons is clicked on, one of the queries is run, and the results from the run are presented to the user.
7. Create a form that allows the user to browse through the information recorded in this database about the horses, jockeys, and owners. Use the tab-control to do so.

Design a logo for this database. The logo can contain the picture of a horse, or the picture of a jockey on a horse. Put this logo in the forms created above. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

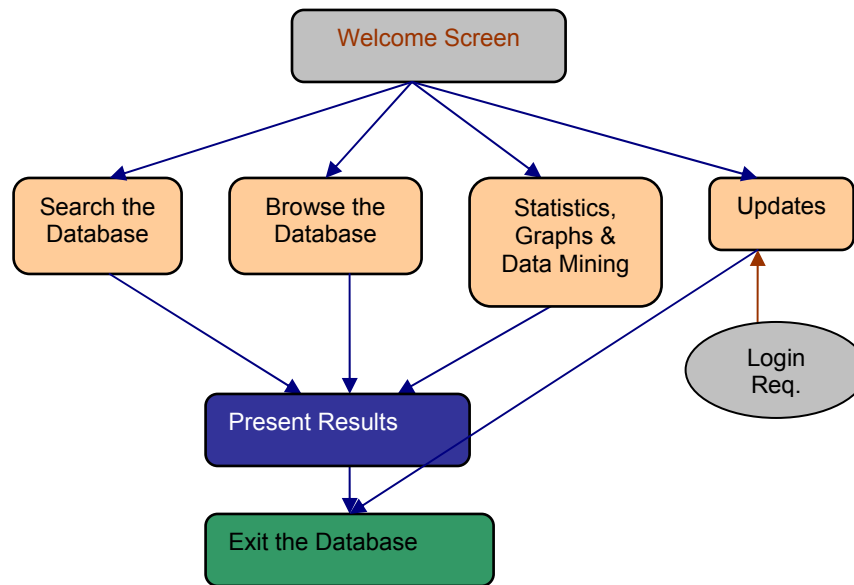
Reports:

1. Report on the ranking of the horses based on winning percentage, average winnings, speed rating, previous class rating, time of last race and last workout, and change in their body weight.
2. Report the date and time of the last race a particular horse raced that was at a distance close to today's distance and in which the horse was first at the last stretch and won the race by at least one and a half lengths.
3. Report the name and owner of the horses that meet these descriptions:
 - a. Never won a race.
 - b. Had no wins in the last ten races.
 - c. Never had a race at today's track.
 - d. Never ran today's distance at today's track.

4. Report the name and owner of the horses that meet these descriptions:
 - a. Did not race within one month of today's race.
 - b. Won at least two of its last five races.
 - c. Finished no worse than second in at least five of the last ten races.
 - d. Either raced and finished second or better in the last ten days or has shown an exceptional workout in the last four days.

Visual Basic.NET Application Development

This database application can be used by individuals who are interested in betting or by the database administrator. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important.

Search: The user can search the database to retrieve information about horses, jockeys, forthcoming races, etc. We suggest that the search queries and forms already created in the Access Application Development section be included in here.

Browse: The user can browse the database to learn more about horses, jockeys, races, etc.

Statistics, graphs, and data mining: This option provides various statistics, plots trends, and performs data analysis. The following are some examples:

1. Plot the average number of the races run by a horse per month during the last year.
2. The total number of races won per year during the last ten years by a horse of a particular breed (e.g., English horses).
3. Plot the time it took to finish the same race by the winner per year during the last ten years.

4. Plot the time it took to finish the same race by the worst performer per year during the last ten years.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about horses, jockeys, races, etc.

Web Extension

Users may access this database from personal computers at home or in the office. To be able to use the database, users pay a membership fee. Members have a login name and password. Users should be able to search for horses, jockeys, and forthcoming events, update their personal information, etc. Members, however, should not be allowed to update/delete information about other members, horses, jockeys, races, etc. Only the administrator of the database can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

R. C. Vergin, "An investigation of decision rules for thoroughbred race horse wagering," *Interfaces* 8(1), 1977, pp.34-45.

Problem Description

This project is about developing a ranking system of tennis players that is objective, consistent, and broad-based. These qualities are important in a ranking system since rankings directly affect the acceptance of tennis players' entry and their placement in the draw. The United States Tennis Association (USTA) is not satisfied with the existing ranking methods, and the officials at the USTA have approached you to help them with this project.

Your team has examined most of the existing ranking methods and identified the strengths and the weaknesses of each of these methods. Finally, you have come up with the following formula, which calculates the ranking points for a player.

$$P_i = \frac{1}{(NT)_i} \sum_t T(i,t) + \frac{\alpha}{(NM)_i} \sum_j W(i,j)P_j$$

Where

P_i	is the total ranking points for player i
$(NT)_i$	the number of tournaments played by player i
$T(i,t)$	the tournament points earned by player i
α	the weighing factor of 1.35
$(NM)_i$	the number of matches played by player i
$W(i,j)$	the number of wins player i has over player j .

The strength of the tournament is a function of the quality of the players and the size of the tournament and it is calculated by the following formula:

$$\text{Tournament Strength} = \frac{\text{average rank of contestants}}{\text{ideal average rank}} \times \frac{\text{number of players}}{\text{number of ranked players}}$$

The ideal average rank is simply the following: If NP is the number of players in the tournament, then the ideal average rank is $(NP+1)/2$. The following table presents the points needed to advance to certain rounds in a tournament.

Tournament Strength	Winner	Runner-up	Losing Semis	Losing Quarters	Losing R 16	Losing R 32	Δ factor
3-4.9	10	8	6	4	2	0	2
5-6.9	8	6.4	4.8	3.2	1.6	0	1.6
7-9.9	6	4.3	3.6	2.4	1.2	0	1.2
10-14.9	4	3.2	2.4	1.6	0.8	0	0.8
15-19.9	3	2.4	1.8	1.2	0.6	0	0.6
20-29.9	2	1.6	1.2	0.8	0.4	0	0.4
30-49.9	1.3	1.04	0.78	0.52	0.26	0	0.26
50-99.9	0.7	0.56	0.42	0.28	0.14	0	0.14
100-199.9	0.35	0.28	0.21	0.14	0.07	0	0.07
≥200	Is counted as "matches only" for ranked players						

The "Δ factor" in the table indicates values that are added for each incremental change in a player's advancement.

Database Design

We present here the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Player*: The main attributes are social security number, first name, last name, history of ranking points (consists of date and corresponding ranking points), current ranking, age, weight, height, history of awards won, etc.
2. *Court*: The main attributes are name, type (grass, clay, hard surface, etc.), location, etc.
3. *Tournament*: The main attributes are name, location, date, tournament strength, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. To help USTA members with the process of selecting the tennis players for the upcoming tournament, create the queries described below. The information from these queries will be used to rank the players of the tournament based on the formula presented in the problem description. For each player:
 - a. Find the number of tournaments played.
 - b. Find the tournament points earned.
 - c. Find the number of matches played.
 - d. Find the number of wins.

2. The price of the ticket for attending a game in this tournament depends, among other factors, on the quality of the players. The following queries help to identify the quality of a player:
 - a. Sort the players based on the number of wins. Consider only the wins in a major tournament (a major tournament is a tournament with a certain minimum strength).
 - b. List the players with a winning percentage higher than 50% on clay courts.
3. Find the tournaments that on average have an attendance of at least 60 players.
4. List all major tournaments played on grass, clay, and hard surface.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: players, courses, tournaments, etc. These forms allow the user to add, update, and delete information about players, courses, tournaments, etc.
3. Create a form that allows the user to browse through the names of the players attending the current tournament. Create a subform that presents for each player the corresponding ranking, together with the name, location, and date of the races won. Include a command button that, when clicked on, calculates and presents the number of points gained by this player.
4. Create a form that allows the user to choose (say, from a combo box) a tournament. Create a subform that presents the name and the number of points of the winner of the tournament.
5. Create a form that includes the following:
 - a. A list box that is used to choose a court type.
 - b. A subform that presents detailed information about the races held on the chosen (from the list box) court type.
 - c. An “Exit” button that closes this form.

Design a logo for this database. The logo can contain the picture of a tennis player, the picture of a tennis ball, etc. Put this logo in the forms created above. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

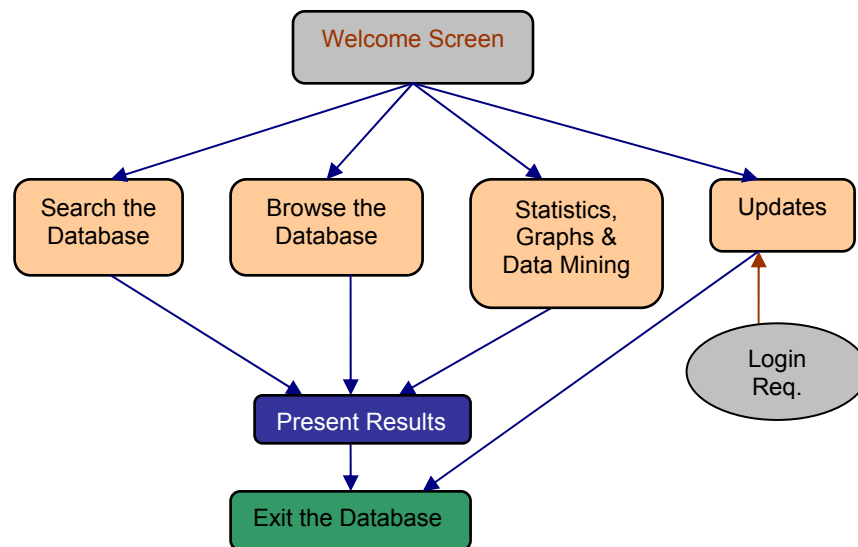
Reports:

1. Report the ranking points for each player. Include in this report the social security number, name, and age of a player. Sort this information by number of points.
2. Use the chart wizard to present the following:
 - a. The number of participants in a particular tournament per year for the last ten years.
 - b. The number of matches played in a particular tournament during the last ten years.
 - c. The average number of matches played per day in a particular tournament during the last ten years.

- d. This information is needed in the process of planning for accommodating the participants of a tournament.
3. Create the following summary reports for the current tournament:
 - a. List the top 10% of tennis players.
 - b. Provide detailed information about the winner of this tournament. The information should include the following: social security number, name, age, a list of previous wins, etc.
4. List the players that have won a major tournament.
5. Report all the major tournaments played on clay.

Visual Basic.NET Application Development

This database application can be used by individuals who are interested about attending the tournament, the players, or the database administrator. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important.

Search: The user can search the database to retrieve information about players, tournaments, etc. We suggest that the search queries and forms already created in the Access Application Development section be included in here.

Browse: The user can browse the database to learn more about players, tournaments, courts, etc.

Statistics, graphs, and data mining: This option provides various statistics, plots trends, and performs data analysis.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about players, tournaments, etc.

Web Extension

The users may access this database from personal computers at home or in the office. To be able to use the database, a user should be a member. Each member has a login name and password. A user should be able to search for players, tournaments, etc. A member, however, should not be allowed to update/delete information about other members, players, tournaments, etc. Only the administrator of the database can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

T. M. Musante and B. A. Yellin, "The USTA/Equitable Computerized Tennis Ranking System," *Interfaces* 9(4), 1979, pp.33-37.

Problem Description

The traditional MBA program has been receiving criticism because it is mainly focused on analytical training, whereas employers are looking for executives with broader education. Some of the concerns regarding the existing system are as follows:

- The curriculum may not be relevant to many political, legal, or social business problems.
- The MBAs need better communication and interpersonal skills.
- Currently the MBAs are oriented to analysis rather than decision-making.
- The MBAs are focused on short-term rather than long-term performance.
- The MBAs are not oriented to competition in the international market.

The American Assembly of Collegiate Schools of Business (AACSB) requires the following components in an MBA curriculum:

- Common body of knowledge in management.
- A field of specialization.
- General competence for overall management.

The following is a list of courses offered in the MBA program.

Foundational areas

Accounting
Communications
Information Systems
Managerial Economics
Organizational Behavior
Quantitative Methods

Functional Applications

Finance
Human Resources
Marketing
Production and Operations

General Management

Economic System
International Marketplace
Legal and Governmental Processes
Society and Ethics
Strategy, Planning, Policy

Sectoral and Institutional Areas

Arts Management
Educational Administration
Health Care Management
Public Management
Transportation

Many Business Schools are trying to improve their MBA program, and this is not an easy task. You were hired to build a support system that will be used to redesign the existing MBA curriculum. You can use the following guidelines to build your system, but you should be creative and add additional features to your system.

Database Design

We present here the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational

database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Curriculum*: The main attributes are curriculum code, a list of courses offered, etc.
2. *School*: The main attributes are name, location (suburban, major city, etc.), code of their current curriculum, number of MBAs they graduate every year, size of their library, budget allocated to the MBA program, AACSB accreditation status, etc. (AACSB accredit those schools that meet certain requirements. AACSB accreditation is important since it indicates a high quality program.)
3. *Alumni*: The main attributes are social security number, name, name and address of the current employer, current position, current salary, GPA at graduation, etc.
4. *Course*: The main attributes are course code, name, type (foundational, functional, general, or institutional), topics covered, etc.
5. *Student*: The main attributes are social security number, name, address, current GPA, a list of courses completed, enrollment date, expected graduation date, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. There are a few indicators of the quality of a particular MBA program. Create a query that prompts for the name of a school and returns the following information:
 - a. The total number of MBA graduates per year during the last ten years.
 - b. A list of graduates who are currently CEOs. Include in this list the name of the company they work for and estimated yearly salary.
 - c. A list of graduates who make more than \$60,000/year. For each graduate report the corresponding GPA.
 - d. A list of the graduates who had a GPA higher than 3.3. For each graduate present the name of the current employer and position.
2. The experience from the best Business Schools in the nation should be considered when designing the curriculum of an MBA program. Create the following queries to help identify such schools:
 - a. List the schools that have graduated more than 1,000 MBAs in the last 10 years.
 - b. For all schools in the database, find the total number and the percentage of MBAs (out of all MBA graduates) who currently have a top management position with a company. Sort the information by total number of MBA graduates on top management position.
 - c. Show for each MBA program how long it has been AACSB accredited. Sort the information by number of years.
 - d. For each school, present the total number of graduates who had a GPA higher than 3.3 and are currently working as managers. Consider the data for the last ten years. Sort the information by total number of graduates.

3. In order to judge the validity of a particular curriculum, it is important to identify whether it has been implemented and how many schools are currently using it. List the codes of the curriculums that are currently implemented, and for each one present the total number of schools that are currently using it.
4. In the accredited MBA programs, students are required to complete their foundational courses. For each accredited MBA program, list the number of current students who have completed the foundational courses. For the students who have not completed their foundational courses, identify how far they are in the program. Count the number of students who are planning to graduate in the current semester and have not completed the foundational courses. These students will have to postpone their graduation.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: curriculums, schools, students, courses, etc. These forms allow the user to add, update, and delete information about curriculums, schools, students, courses, etc.
3. Create a form that allows the user to select from a combo box the name of a Business School. Create a subform that presents the following information about the selected school: social security number and name of the students who are currently enrolled in the program.
4. Create a form that allows the user to browse through the information about the students who are currently enrolled in an MBA program. Create a subform that presents the following information about a student: social security number, name, address, enrollment date, and a list of courses taken so far. Group the courses by type (foundational, functional, general, or institutional).
5. Create a form that allows the user to choose a particular curriculum from a list box. Create a subform that presents for the chosen curriculum the name and code of the courses offered.
6. We designed a number of queries that help identify the best MBA programs in the nation. Create a form that enables the user to run these queries. Command buttons can be used for this purpose. Name this form "Identifying the Best MBA Programs." When the user clicks on a particular command button, one of the queries is run, and the outcome is presented.

Design a logo for this database. Put this logo in the forms created above as well as on the reports that you will be creating next. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

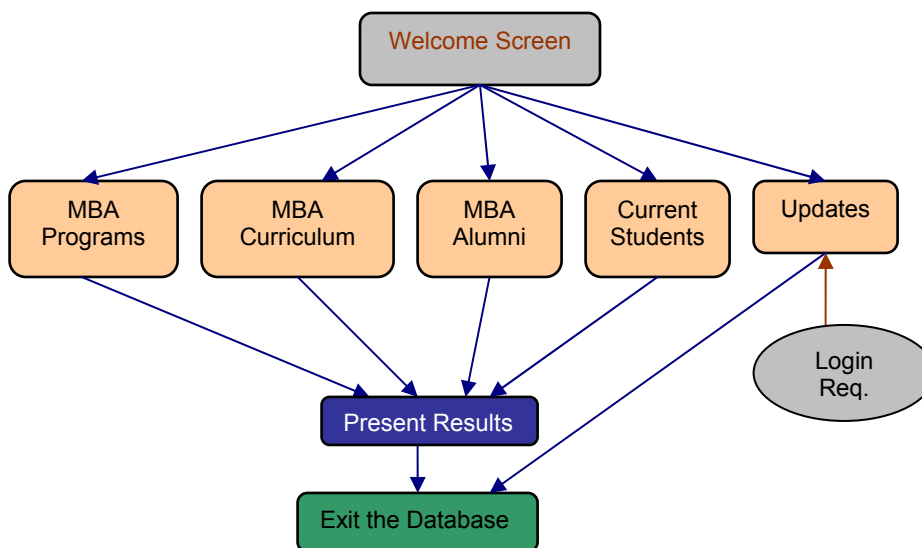
Reports

1. Prepare a report on existing curriculums. This report should include information such as the number of schools currently using that curriculum, the courses offered, and the date the curriculum was last updated.
2. Every month, MBA students get a newsletter. Use the label wizard to create labels with the address of the students who are currently enrolled in an MBA program.
3. Use the chart wizard to plot the following:

- a. The total number of students graduated from an MBA program per year during the last ten years.
 - b. The total number of students enrolled in an MBA program per year during the last ten years.
 - c. The total number of MBA graduates who have a top management position. To build this plot use the data about MBA graduates of the last ten years.
 - d. The number of AACSB accredited MBA programs per year for the last ten years.
 - e. The average student/faculty ratio of each MBA program.
 - f. Average GPA of MBA graduates per year during the last ten years.
4. Report for each of the current students the classes they have taken. Group the classes by type.
 5. List the schools that are AACSB accredited.

Visual Basic.NET Application Development

This database application can be used by prospective MBA students, companies that plan to hire MBAs, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important.

MBA Programs: The user can search the database to retrieve information about the business schools that offer MBA programs. We suggest that the search queries and forms already created in the Access Application Development section be included in here. These forms will enable the user to identify the best MBA programs in the nation, the placement of an MBA program of preference, etc.

MBA Curriculum: The user can browse the database to learn more about curriculums offered in business schools. The user can identify the most successful curriculums, the courses included in a particular curriculum, etc.

MBA Alumni: This option provides information about MBA graduates. We encourage the use of statistical analysis tools, usage of plots (similar to the ones created in the Access Application Development section), etc. that enable the user to draw conclusions about the placement of an MBA program.

Current Students: This option provides information about current MBA students.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about MBA programs, students, etc.

Web Extension

Users of this database are alumni, current students, professors, companies, etc. A user can access the database from a personal computer at home or in the office. To be able to use the database, a user should be a member and have a login name and password. To become a member, the user should submit the following information: name, name of the employer (if applicable), address, profession, etc. The user should be able to search the database to find out about the current curriculums, AACSB accredited schools, etc. A member, however, should not be allowed to update/delete information about other members, curriculums, schools, etc. Only the administrator of the database can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

D. Windsor and F. D. Tuggle, "Redesigning the MBA curriculum," *Interfaces* **12**(4): 72-77, 1982.

Problem Description

Blue Bell is a large apparel manufacturer. They run three major businesses. The largest is the Wrangler group. The Wrangler group manufactures jeans and several other lines of sports and casual apparel. Wrangler markets to men, women, and children in the USA and has a substantial international business. Red Kap, the second largest business, makes a variety of durable garments used for on-the-job wear by production, service, and white-collar personnel. Jantzen, the third major business, manufactures a variety of sports and casual apparel product lines, including popular lines of swimwear and sweaters.

Each of the above production lines breaks down into styles, lots, sizes, and stock keeping units (SKU). The management has realized that the number of product lines, styles, and SKUs had grown tremendously. The production process had incorporated several new automated manufacturing operations in response to changes in customer taste. The management was concerned about the high investment in working capital. A large part of the working capital is inventories. In this process of extending the business, it is crucial for Blue Bell to have an effective coordination of their activities. Since the investment on the inventories is quite high, we will propose a database management system that will help Blue Bell manage their inventories.

Database Design

We present here the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Production line*: The main attributes are identification number, name (e.g., jeans, t-shirts, swimwear production line, etc.), and classification (e.g., menswear, women's wear, boys' wear and kids).
2. *Style*: The main attributes are identification number, name, etc. Note that there are different styles for a production line. For example, the jeans production line produces the following styles: straight-leg jeans, boot-leg cut jeans, etc.
3. *Lot*: The main attributes are identification number, name (e.g., blue, green, etc.), etc. Note that there are different lots produced for a particular style. For example, the straight-leg jeans come in blue, green, black, etc.
4. *Size*: The main attributes are identification number, description, etc. Note that depending on the product, the measurements reported vary. For example, for pants we keep information on waist number and length number.
5. *Customer*: The main attributes are identification number, name, address, telephone number, etc.
6. *Transaction*: The main attributes are identification number, date, quantity, total dollar amount paid/received, etc.

7. *Product.* The main attributes are identification number, name, on-hand inventory, outstanding orders, historical information on production (includes date and quantity produced), etc. Note that a final product is produced in a particular production line and has a particular style, lot, and size.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The managers want to identify customers with whom the company is interested in keeping long-time relationships. The following queries help the managers in making their selections:
 - a. Find the top ten customers with whom the company does most of its business. Include in this query the identification number and name of the customers as well as the value of the products sold to them so far.
 - b. List the top ten customers who have been doing business with the company for more than five years.
 - c. List the top five customers that have the highest yearly revenues.
2. The management is considering a reallocation of resources. They plan to allocate more resources to products that sell most, have a high profit margin, etc. The following queries will help the management to make a better decision:
 - a. List the five final products that sold most during the last year. Include in this query the name and identification number of the product, as well as the quantity sold in the last year and the dollar amount generated.
 - b. List the five styles that sold the most last year. Include in the query the quantity sold per style and the dollar amount generated.
 - c. Find the size that sold the least last year. Include in this query the quantity sold per size and the dollar amount generated.
 - d. List the ten final products that have the highest level of inventory. Include the following in this query: identification number, name, amount in the inventory, and the value of the inventory.
 - e. List the five final products that have the highest profit margin. For each product, present the quantity sold during the last year and the profit made.
3. The following information helps the management to update the production schedule in such a way that inventory level is minimal and demand is fully satisfied:
 - a. Find the period of time (the month) that has the highest sales. Include in this query the total quantity sold and the dollar amount generated.
 - b. Identify the period of time that has the least sales.
 - c. Find the product that sells the most. Include in this query the identification number of the product, amount sold, and the corresponding value.
 - d. Identify the quantity on-hand per product in each month during the last year.
 - e. Identify the quantity sold per product in each month during the last year.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: products, customers, production lines, transactions, etc. These forms allow the user to add, update, and delete information about products, customers, production lines, transactions, etc.
3. Create a form that allows the user to browse through the names of the final products. Create a subform that presents the following information about each product: the corresponding production line, style, lot, and size; the level of on-hand inventory; the number of backorders; the unit cost, and the unit price.
4. Create a form that allows the user to choose a customer from a combo box. Create a subform that presents for the selected customer information about orders that have not yet been delivered.
5. Create a form that allows the user to choose from a list box a production line. Create a subform that presents for the selected production line the identification number and name of the final products produced, monthly production, unit cost, and unit price.
6. We presented a number of queries that can be created to help the managers decide about resource allocation. Create a form that enables the user to run these queries. Command buttons can be used for this purpose. When the user clicks on a particular command button, one of the queries is run, and the outcome is presented.

Design a logo for this database. Put this logo in the forms created above as well as on the reports that you will be creating next. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

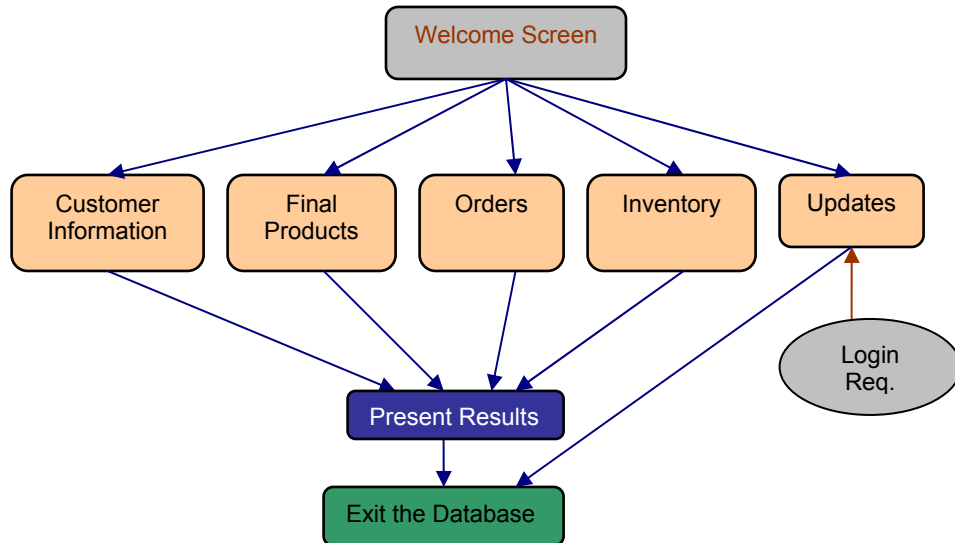
Reports

1. Prepare a report for each sales transaction. This report should include a date, name and identification number of the customer, name and identification number of the product sold, the quantity sold, and the value of sale.
2. Prepare a report that presents the inventory level in the company at the end of the month. This report should include for each product the identification number and name. Give details on production line, style, lot, size, amount in the inventory, and value of the corresponding final product.
3. At the end of the year, the managers prepare a balance of payments. Among other items, include in this report the total amount of money received during the last year by each customer, and include the total amount of money in the inventory.
4. Use the label wizard to create labels with the address of each customer.
5. Use the chart wizard to plot the following information:
 - a. Total sales (in dollar amount) per year during the last ten years.
 - b. The value of the inventory per year during the last ten years.
 - c. For each product, the unit cost and the unit price per year during the last ten years.

- d. For the product that sold most in the last year, plot the yearly sales per year during the last ten years.

Visual Basic.NET Application Development

This database application can be used by customers, managers, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important.

Customer Information: The user can search the database to retrieve information about a customer. We suggest that related search queries and forms already created in the Access Application Development section be included in here. These forms will enable the user to: find the name, address, and telephone number of a customer; identify the most valuable customers; find whether a customer has set an order; etc.

Final Products: The user can browse the database to learn more about a final product. The user can identify the following: the production line where the product is produced; the level of inventory, unit cost, unit price, and total sales for the current year; etc. We encourage in here the use of statistical analysis tools and usage of plots similar to the ones created in the Access Application Development section.

Orders: This option provides information about orders, such as outstanding orders, due date, quantity ordered, value of the order, etc.

Inventory: This option provides information about the current inventory.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about products, customers, orders, etc.

Web Extension

Users of this database are managers, customers, database administrators, etc. A user can access the database from personal computers at home or in the office. To be able to use the database, a user should be a member and have a login name and password. A customer should be able to browse through the information about final products, set orders, check the status of an order, etc. The managers use the database in order to learn about the sales transactions performed, check the inventory level of a product, etc. Only the database administrator has access to the data entry forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

J. R. Edwards, H. M. Wagner, and W. P. Wood, "Blue Bell trims its inventory," *Interfaces* **15**(1): 34-52, 1985

Selecting an All-Star Team

Problem Description

In Canada, the national intercollegiate football championship game, the Vanier Cup, is played toward the end of November at the SkyDome in Toronto. A committee of Canadian Intercollegiate Athletic Association (CIAU) head coaches selects the team, deeming the players selected to be the best university football players in the country. A lot of work goes into the selection of the team. Here is roughly the way it works. Each of the four conferences in the CIAU (the Atlantic, the Ontario-Quebec, the Ontario, and the West) selects an all-stars team. The all-Canadian team is then selected from players of these four all-star teams. Below we describe the selection of all-stars team in the Ontario-Quebec Conference.

Up to 1995, all-star selection in the Ontario-Quebec Intercollegiate Football Conference was accomplished through a head coaches' meeting on the Sunday before the conference semifinal play-off game. Each team nominates any player believed to be deserving of membership on the all-star team. Each coach and assistant coach, members of CIAU, evaluates each nominee on a scale from 1 to 10. Only the players that get the highest score would make the all-star team. This system had two drawbacks. First, coaches involved in the play-off games of the next Sunday lose a day of game planning. Second, there were assistant coaches who felt this "horse-trading" season was not very objective. The objective of this project is to build a database decision support system that will facilitate the process.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Championship teams*: The main attributes are team identification number, team name, coach name, name of the university it represents, rank of the team in the current season, number of games they won and the number of games they lost in the current season, etc.
2. *Players*: The main attributes are social security number, name, address, birth date, current position in the team, name of the university he represents, number of years with the team, number of years with the university (freshman, sophomore, junior, or senior), number of times he made the all-star team, etc.
3. *Coach*: The main attributes are social security number, name, address, birth date, name of the university he represents, number of years of coaching in the current university, total number of years of coaching, number of times the team he leads won a championship, number of times he took a team to the semifinals, etc.
4. *Assistant coach*: The main attributes are social security number, name, address, birth date, name of the university he represents, number of years working as assistant

coach in the current university, total number of years working as assistant coach, current specialization (e.g., defensive coordinator, offensive coordinator, etc.), etc.

5. *All-star game*: The main attributes are date and place of the event, winning team, name of the coach of the winning team, name of the players of the winning team, final result (the score) of the game, etc.
6. *All-star team nominees*: The main attributes are social security number, name, address, birth date, current position in the team, name of the university he represents, number of years with the team, number of years with the university (freshman, sophomore, junior, or senior), number of times he made the all-star team, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The first step in this selection process is to identify the nominees from each team. A football team has 7 positions: quarterback, running backs, inside receivers, wide receivers, center players, guards, and tackles. Create a query that for each position lists the names of the candidates and the total score from the evaluation process. Group this information by position. Within each group sort the candidates based on their total score.
2. To facilitate the process of selecting the all-star game players from the nominees of each position, create a query that finds the name and SSN of the top five nominees for each position. In the same query present the name of the university the player represents, the number of years playing, and the total score from the evaluation process. Sort the candidates based on their total score.
3. A different approach that can be used to rank the players is the following: for each player find the median (the value of the middle observation) of the evaluation scores, and then rank the players based on the median. This approach has shown to reduce biased rankings. This query, for each team position, ranks the candidates based on the median of the evaluation score.
4. Create the following search queries:
 - a. This query prompts the user for the name of a team and presents the name of its players, the name of its coach, and the rank of the team.
 - b. Create a query that prompts for the name of a team and provides a list of the games this team played and the corresponding results.
 - c. Create a query that prompts for the name of a coach and provides a list of the teams he coached.
 - d. Create a query that prompts for a date and presents the name of teams that played on that date, the name of team they played against, and the final result of the game.
 - e. Create a query that prompts for a year and presents the name of the players that made the all-star team that year and the result of the game.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: players, teams, games, coaches, etc. These forms allow the user to add, update, and delete information about players, teams, games, coaches, etc.
3. Create a form that allows the user to browse through the information about the coaches. Create a subform that presents for the selected coach all the all-star games s/he participated in, the name of the teams s/he represented on these games, and the final score of the games he participated in. Include in the same form a command button that, when clicked-on, calculates and presents the number of the players trained by this coach.
4. Create a form that allows the user to browse through the information about the assistant coaches. Create a subform that presents for the selected assistant coach the all-star games s/he participated in, the name of the teams s/he represented on these games, and the final score of the games s/he participated in. Include a command button that, when clicked-on, presents a list of the universities this assistant coach worked for.
5. Create a form that allows the user to choose from a combo box the name of a player. Create a subform that presents for the selected player the following information: social security number, name, address, name of the team he plays for, and name of his coach. Include in this form a command button that, when clicked-on, calculates and presents the player's total score. Include another command button that when clicked-on lists all the all-star games this player participated in.
6. Create a form that allows the user to choose from a list box an all-star game. For the selected game, present in a subform the following information: the name of the players who made the team, and, for each player, the position he played and the name of the university he represented. Include a command button that, when clicked-on, presents the result of the game.
7. Create a form that allows the user to browse through the name of the teams participating in this championship. For each team, present the name of the university it represents, the name of its coach, and the rank of the team in the current season. Include a command button that, when clicked on, lists the names of the players on this team. Include another command button that, when clicked on, lists the games the team played in the current season, the name of the team played against, and the result of the game.

Design a logo for this database. The logo can contain the picture of a football player, the picture of a football, etc. Insert this logo in the forms created above. Make the background color of the forms green and the border color for the title yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create a report that presents the following information about the all-star team nominees: social security number, name, position played, name of the team he represents, name of the university he represents, name of his coach, and his total score collected during the evaluation process. Include in this report for each player

the average and the median of his score. Have this information grouped by position, and within each group sort the information by total score.

2. In the process of choosing the players, it is helpful to see the distribution of the scores each player received. Use the chart wizard to present the distribution of the scores each player received during the evaluation process.
3. Create a report that includes the following information about the each player of the all-star team: social security number, name, birthday, name of the university he represents, and his position in the all-star game.
4. Create a report that includes the following information about the result of the all-star game: date and place of the event, the final result of the game, the name of the team that won, the name of the coach, assistant coaches, and players of the team.
5. Create a report that presents historical results of the all-star games (winning team, score, and name of the coach) in the last 10 years.
6. Thank-you notes will be sent out to all the teams participating in this championship and their coaches. Use the label wizard to create labels with the address of each coach.

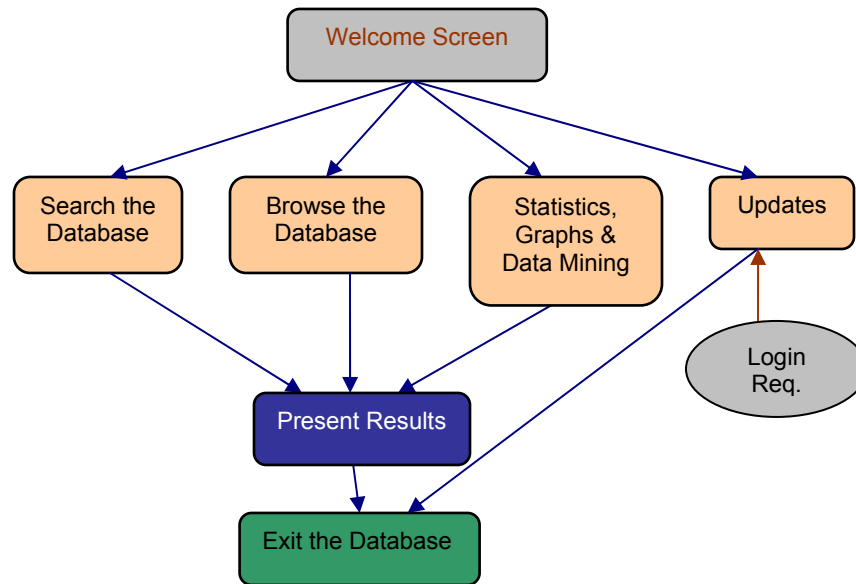
Visual Basic.NET Application Development

This database application can be used by coaches, assistant coaches, players, and fans. In the following figure we present a tentative layout of the system.

In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important.

Search: The user can search the database to retrieve information about players, teams, coaches, etc. We suggest that the search queries and forms already created in the Access Application Development section be included in here.

Browse: The user can browse the database to learn more about players, teams, coaches etc.



Statistics, graphs, and data mining: This option provides various statistics, plots trends, and performs data analysis. The following are some examples:

1. Plot the score that each nominee received in the evaluation process.
2. For each team, plot the total number of wins in a championship during the last ten years.
3. Plot the number of participating teams/players in the football championship during the last ten years.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about players, teams, coaches, etc.

Web Extension

A user may access this database from personal computers at home or in the office. To be able to use the database, the user needs a login name and a password. A user should be able to search for players, teams, coaches, etc. A member, however, should not be allowed to update/delete information about players, teams, coaches, etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

W.J. Hurley, "An efficient, objective technique for selecting an all-star team," *Interfaces* **28**(2): 51-57, 1998.

12 Farm Management

Problem Description

John's family owned and operated a 640-acre farm for several generations. John, a recent graduate of the Industrial and Systems Engineering Department, is now in charge of the family business. John has to decide about what type of livestock and how many of each type they should keep in the farm as well as what type of crops and how much they should plant the next year. The amount produced for each crop depends on the weather conditions. Help John to build a database that will assist him in managing the business.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Livestock*: The main attributes are identification number, name, type, breed, age, description, etc.
2. *Crop*: The main attributes are identification number, name, type, breed, description, average yield, etc.
3. *Livestock transactions*: The main attributes are transaction number, transaction date, selling price per head, number of livestock sold, total amount of money earned from sales, etc.
4. *Crop transactions*: The main attributes are transaction number, transaction date, crop name, amount sold, price per unit, amount of money earned, etc.
5. *Labor*: The main attributes are date, amount of person-hours used, amount of money paid per hour, total amount paid, job description, etc.
6. *Weather*: The main attributes are date, temperature, amount of rain, description, etc.
7. *Customer*: The main attributes are identification number, name, address, telephone number, fax number, etc.
8. *Supplier*: The main attributes are identification number, name, address, telephone number, fax number, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create the following search queries:
 - a. This query prompts for the name of a customer and returns a list of the purchases made by this customer. For each purchase include the following information: purchase date, quantity purchased, shipping date, amount paid, payment date, etc.
 - b. This query prompts for the name of a supplier and returns a list of its outstanding orders (orders not yet received). For each order include the following information: order release date, order delivery date, description of the products included in the order, quantity ordered, amount to be paid, payment date, etc.
 - c. This query prompts for the name of a crop and returns a list of customers who have purchased this particular crop. This list is used by the management to inform customers about price changes, special deals, etc. For each customer, include the address and the telephone number.
 - d. This query prompts for the type of a livestock and returns the following information: selling price per head per year during the last ten years, number of livestock at the end of the year during the last ten years, number of livestock sold in a year during the last ten years, total amount of money earned from sales per year during the last ten years, etc.
 - e. This query prompts for the name of an item (livestock or crop) and return a list of the suppliers that provide this item. For each supplier display the address and telephone number.
2. The following queries help the management to understand which crops/livestock are selling more/less. This information will be used to reallocate the investment funds for the coming year.
 - a. List the five crops that have generated the highest income in the last year.
 - b. List the five livestock that have generated the highest income in the last year.
 - c. Find the crop that has generated the lowest income in the last year.
 - d. Find the livestock that has generated the lowest income in the last year.
3. John wants to establish long-term relationships with some of his most qualified customers and suppliers. The following queries will help him to identify the potential customers/suppliers.
 - a. Find the five customers that have purchased the most (had the highest yearly spending) during the last year.
 - b. Find the five customers that have purchased the most (had the highest average yearly expenses) during the last five years.
 - c. Find the five suppliers from whom the company has purchased the most (had the highest yearly spending) during the last year.
 - d. Find the five suppliers from whom the company has purchased the most (had the highest average yearly spending) during the last five years.
 - e. List the suppliers that never sent a late shipment during the last year.
4. Create a query that presents the total amount of money generated from livestock and crops per year during the last ten years. Sort the information by the total income

generated from both livestock and crops. In the same query include a description about the weather conditions in each year. This query will help John to (a) understand the impact of the weather conditions on crop yield and (b) decide whether crops or livestock should be the main stream of the business.

5. Create a query that presents the total amount of money spent and the total amount of money earned per year during the last five years.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: crops, livestock, crop transactions, weather, etc. These forms allow the user to add, update, and delete information about crops, livestock, crop transactions, weather, etc.
3. Create a "Search" form. This form consists of five command buttons. When a particular command button is clicked on, one of the five search queries that we have already built will run and the corresponding results will be presented.
4. Create a form that browses through the information about crops. The form presents the following information for each crop: identification number, name, type, breed, average yearly yield, etc. Create a subform that for the selected crop presents the yield, price, and total earnings per year during the last ten years. Create a command button that, when clicked on, prompts the user for a date and returns the weather conditions on that date. Create another command button that, when clicked on, prompts the user for a year and returns the name and identification number of the crop that sold the most in that particular year.
5. Create a query that browses through the information about livestock. The form presents the following for each livestock: identification number, name, type, breed, etc. Create a subform that for the selected livestock presents the average selling price per head, number of livestock at the end of the year, number of livestock sold, and total amount of money earned from sales per year during the last ten years. Create a command button that, when clicked on, prompts the user for a year and returns the name and identification number of the livestock that sold the most in that particular year.
6. Create a form that allows the user to choose a customer from, say, a combo box. Create a subform that for the selected customer presents a list of backorders (orders not yet delivered). For each backorder present the following: order date, quantity ordered, description of the products ordered, etc. Create a command button that, when clicked on, prompts an order number and returns the number of days that a particular order is late or the number of days left until delivery date.
7. Create a form that allows the user to choose a supplier from, say, a list box. Create a subform that for the selected supplier presents a list of outstanding orders (orders not yet received). For each outstanding order present the following: order date, quantity ordered, description of the products ordered, etc. Create a command button that, when clicked on, prompts an order number and returns the number of days that a particular order is late or the number of days left until delivery date.

Design a logo for this database. The logo may contain the picture of a crop, livestock, a farm, etc. Insert this logo in the forms created above as well as on the reports that you will be creating next. Include the following in the forms created: record navigation command

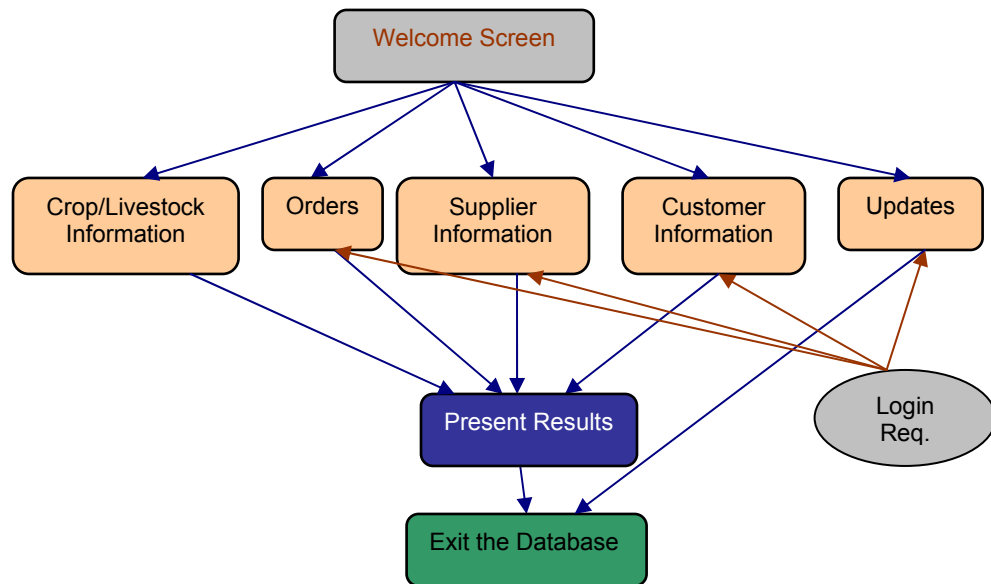
buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create a report that presents all the buy/sell transactions performed in the current month. Include in this report the transaction identification number, transaction date, description, name and address of the supplier/customer, amount of money paid/received, and payment date.
2. Create a report that is used at the end of the year to present the total amount earned from sales, the total amount of money spent, the amount of credit (e.g., sales for which a payment is not received yet) and debt (e.g., purchases for which a payment is not made yet), the amount of crops in the inventory expected to be sold, and the value of livestock owned.
3. Create a report that lists all the customers with whom the farm did business in the past year. For each customer, present the name and a list of orders submitted. For each order, present the quantity ordered, order description, amount paid (or to be paid), and payment date. Group this information by customer. Sort the customers by last name and by order date.
4. Create a report that lists all the suppliers with whom the farm did business in the past year. For each supplier, present the name and a list of orders submitted. For each order, present the quantity ordered, order description, amount paid (or to be paid), and payment date. Group this information by supplier. Sort the suppliers by last name and by order date.
5. Create a report that presents for each crop planted in the farm the following information: name, type, breed, yield in the current year, and unit price.
6. Create a report that presents for each livestock the following information: name, type, breed, selling price per head, total number of livestock, number of livestock sold during the current year, and total amount of money earned from sales.
7. Use the chart wizard to present the following information:
 - a. Annual yield per crop type during the last ten years.
 - b. Annual yield per livestock type during the last ten years.
 - c. Total earnings from crop sales per year during the last ten years.
 - d. Total earnings from livestock per year during the last ten years.
 - e. Total amount of working hours on the farm per year during the last ten years.
8. Use the label wizard to create labels with the address of each customer and each supplier.

Visual Basic.NET Application Development

This database application can be used by managers, suppliers, customers, and the database manager. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important.

Crop/Livestock Information: The user can browse the database to retrieve information about crops and livestock. We suggest that the queries and forms about crops and livestock that we have already described in the Access Application Development section be included in here.

Orders: The user can place an order for a product or check the status of an order. In order to access this information, a login name and password are required.

Suppliers Information: The user can browse the database to learn more about suppliers. Only the farm manager and database administrator have access to this information.

Customer Information: The user can browse the database to learn more about customers. Only the farm manager and database administrator have access to this information.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about crops, livestock, customers, suppliers, etc.

Web Extension

The users of this database are the customers, the suppliers, the farm managers, and the database administrator. A user may access this database from personal computers at home or in the office. A user should be able to search for crops, livestock, yield information, etc. Customers and suppliers, however, should not be allowed to update/delete information about customers, suppliers, crops, etc. Only the database administrator can have access to the update forms. Only the database administrator and farm managers can access the information about the customers and suppliers.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

W.L. Winston, "Operations Research: Applications and Algorithms," *Duxbury Press*, 3rd Edition, 1993.

13 *The Annual Bolder Boulder Race*

Problem Description

The annual Bolder Boulder is one of America's top 10-km races. The race is held each Memorial Day in Boulder, Colorado. This race attracts world-class runners as well as casual joggers. The race has grown to approximately 20,000 participants. The race is a point-to-point race beginning at the Bank of Boulder at the northeast corner of the city, winding throughout the city streets, and ending near to the town center in the University of Colorado's football stadium.

As the race grew in size, problems arose. The quality of the race began to suffer because there was little running room in the crowded streets. A bigger concern was the long queues in the finish lines. Because of the limitations on space, number of chutes and equipment, large queues of runners built up in front of the finish line at certain times.

To address these organizational issues we will build a decision support system.

Database Design

We present here the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Participant*: The main attributes are social security number, name, address, birth date, gender, certified personal record (PR) running times for a 10 km race, etc.
2. *Race*: The main attributes are date; total number of runners registered for the race; number of runners that finished the race; number of female runners; number of male runners; name, age, and running time of the male winner; name, age, and running time of the female winner; name, age, and running time of the male master (male older than 40 years old) winner; name, age, and running time of the female master (female older than 40 years old) winner.

Note that the database keeps information about the running time of each runner for all the races organized so far.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. In such big races the "citizen runners" are separated from the "elite runners" (runners with certified PR of 38 min. for men and 41 min. for women). This separation enables the audience at the football stadium to watch the finish of the world-class runners. The runners are notified if they qualify as "elite runners." Create a query that provides the following information about each of the "elite runners": social security number, name, address, and PR. Have this information sorted by PR running times.

2. In order to avoid crowded finish lines, the runners (other than “elite runners”) are distributed into 19 groups based on stated or proven ability (PR). Each group consists of up to 1,000 runners, and the groups are released (start the race) at fairly constant intervals of one minute. The runners are notified about their group number one week in advance. Create a query that groups the runners based on their performance.
3. Create a query that presents the final results of the race. This query classifies the participants by age group and gender (e.g., an age group consists of females from 5-10 years old, or 10-15 years old, or 15-20 years old, etc.). Within each group, sort the runners based on their running times.
4. Use the information from the current and previous races to find the average running times for each combination of age and gender groups.
5. Use the historical data and the results from the current race to find the average age of the best 1% runners of this race. This query should include the name and social security number of these runners.
6. Using historical data and the results from the current race, find the names and ages of the ten youngest runners of the race.
7. Create the following search queries:
 - a. Create a query that prompts for the name of a runner and presents the running time and placement in the runner’s age group during the last five races.
 - b. Create a query that prompts for a date and returns the results of the race run on that date.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: participants, and races. These forms allow the user to add, update, and delete information about participants and races.
3. Create a form that allows the user to browse through the results of the current and previous races. For each race the following information is presented: total number of runners registered for the race; number of runners who finished the race; number of female runners; and number of male runners. Create a subform that presents for the selected race the names and running times of the three best runners in each age group. Create a command button that, when clicked on, presents the name and time of the winner of this race.
4. Create a form that allows the user to choose from a list box the name of a participant. Create a subform that presents for the selected participant a list of races run and the corresponding running time. Create a command button that, when clicked-on, prompts for the name of a city and returns a list with the name and race results of all the participants from that city.

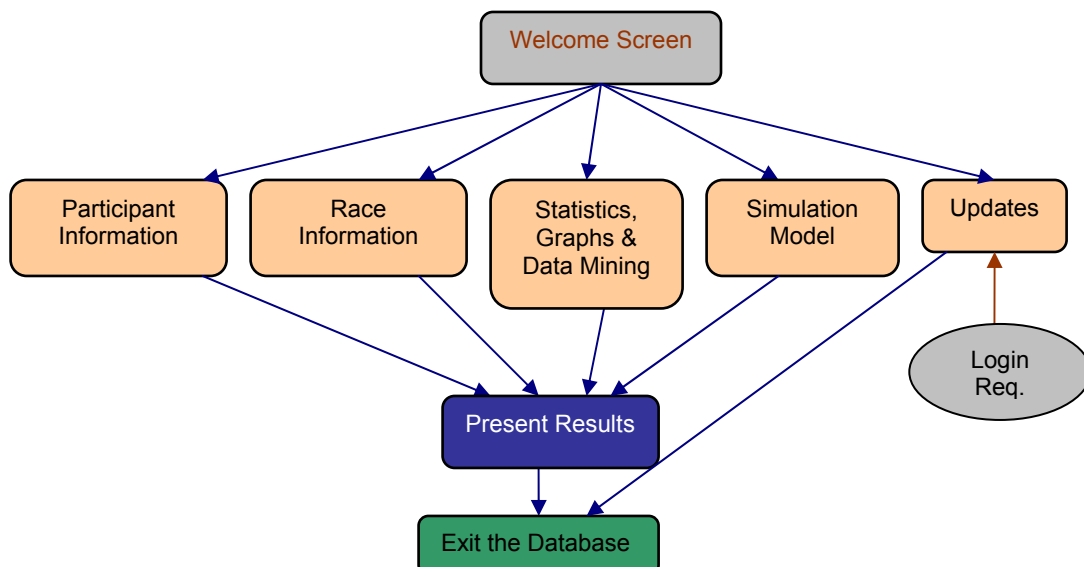
Design a logo for this database. The logo may contain the picture of a runner. Insert this logo in the forms created above as well as on the reports that you will be creating next. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create a report that contains information about the times and names for male and female open (best overall male winner and best overall female winner) and master male and female open for the last five races. The race manager will present this report to the race newsletter.
2. Create a report that contains information about the number of the participants in the race during the ten last years. This report will be presented to the organizers of “Road Runners Club of America” in order to classify this race as a world-class race.
3. Create a report that presents the following information about the “elite runners”: social security number, name, address, birth date, gender, running time in the last race, and PR.
4. Create a report that presents the following information about the runners of the current race: name, age, gender, and running time. Group this information by age and gender. Within each group sort the information by running time.
5. Use the chart wizard to perform the following tasks:
 - a. Graph the running times of the winners of this race during the last ten years.
 - b. Graph the running times of the female winners of this race during the last ten years.
 - c. Graph the running times of the female master winners of this race during the last ten years.
 - d. Graph the running times of the male winners of this race during the last ten years.
 - e. Graph the running times of the male master winners of this race during the last ten years.
 - f. Graph the age of the youngest runners of this race during the last ten years.
6. Use the label wizard to create labels with the address of each participant.

Visual Basic.NET Application Development

This database application can be used by runners, organizers of the race, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important.

Participant Information: The user can search the database to retrieve information about a participant. We suggest that related search queries and forms already created in the Access Application Development section be included in here. These forms will enable the user to do the following: find the name, PR, and running times of a participant, etc.

Race Information: The user can browse the database to learn more about a particular race. The user can identify the following for a particular race: the number of participants, the name and running time of female/male winner, the name and running time of male/female master winner, etc.

Statistics, Graphs, and Data Mining: The statistical analysis tools, such as plots (similar to the ones created in the Access Application Development section), are included in here.

Simulation Model: In this system we plan to include a simple simulation model. This simulation model will help to manage the race.

Complaints have been received about long waiting queues at the end of the race because of the limited number of chutes placed. The race management is thinking of increasing the number of chutes at the finish line. It is understandable that the more chutes that are placed, the easier it would be to handle the runners. However, there is a limit in terms of volunteers who help at the chutes and a limit on the number of chutes that can be placed. The interior dimensions of the stadium allow for at most 15 chutes to be placed.

This is a queuing problem. Since this race has been going on for many years, there is at least ten years' history of data. Use the historical data on running times to find the distribution of the running times and estimate the corresponding parameters. Consider the following: there are two people in each chute who keep track of the time and the identification number of the runners as they pass the finish line. The service time in the chute is uniformly distributed with the mean 2 min. and the standard deviation 1 min. The management wants to simulate the model and identify the optimal number of chutes needed.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about runners, races, etc.

Web Extension

Users of this database are runners, race managers, database administrators, etc. A user can access the database from personal computers at home or in the office. A runner should be able to browse through the information about the races, other runners, etc. but should not be able to update this information. Only the database administrator has access to the data entry forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

R. Farina, G.A. Kochenberger, and T. Obremski "The computer runs the Bolder Boulder: A simulation of a major running race," *Interfaces* **19**: 48-55, 1989.

14 Supply Chain Design

Problem Description

Supply chain design problems concentrate on determining a logistics infrastructure over a planning horizon. Decisions that facilitate the supply chain design include the following: identify the number, type, and location of plants and warehouses; and identify the sourcing assignments between them and the customers. The objective is to provide the most effective strategic solution in terms of cost and/or service.

In this case study, we consider the distribution of automotive parts and supplies to the Ford authorized dealers throughout North America. Ford is faced with pressure to provide excellent customer service, which means timely distribution of parts to the dealers, with minimal logistics investment, both in capital and operations. Planners want to determine the most effective warehouse locations and dealer service regions to meet these objectives.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Dealer*. The main attributes are identification number, name, address (consisting of street address, city, state, and zip), name and telephone number of the contact person, location (consisting of X longitude coordinate and Y latitude coordinate), historical information on demand, facility maintenance cost, etc.
2. *Distribution Center*. The main attributes are identification number, name, address (consisting of street address, city, state, and zip), name and telephone number of the contact person, location (consisting of X longitude coordinate and Y latitude coordinate), historical information on demand, facility maintenance cost, etc.
3. *Product*. The main attributes are identification number, weight (per hundred pieces), cost (per hundred pieces), price (per hundred pieces), inventory level, inventory carrying cost (per hundred pieces), etc. Carrying cost is broken down into capital, taxes, shrinkage, and obsolescence.

Note that when a shipment is sent from the production facility to a distribution center or from a distribution center to a dealer, the following information is recorded: quantity shipped, shipping date, unit price charged, transportation costs, material handling costs, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following are important questions posed to supply chain managers: Does the current supply chain have the right number of distribution centers? Are the distribution centers placed in the right locations? To help the managers answer these questions build the following queries:
 - a. Present the total demand and total inventory for each product. Include in this query the identification number and name of each product.
 - b. Find for each distribution center the distance to each dealer. Include in this query the distributors' and dealers' names.
 - c. Create a query that reports the transportation unit costs and transportation fixed cost from each distribution center to each dealer.
 - d. Assume that each dealer is served by the nearest distribution center. Calculate the total shipping costs for each dealer.
2. Find for each distribution channel (distribution center to dealer channel) the distance traveled in mileage. Include in the query the distribution center name, dealer name, product name, and average amount shipped. Add to this query for each distribution channel the total shipping cost and material handling costs.
3. Create the following query to help the managers improve the current distribution channel. For each dealer find the three nearest distribution centers. Calculate the total transportation and material handling costs for each dealer-distribution center pair.
4. From the information presented in the database, one can see that shipments to dealers are made once a week. Ford is considering increasing the frequency of the deliveries. From now on Ford will send two shipments a week. How is this going to affect the costs? Build a query that finds the total cost (transportation costs, inventory holding costs, etc.) if two shipments are made in a week instead of one.
5. Create the following search queries:
 - a. Create a query that prompts for the identification number of a distribution center and returns a list of dealers that get shipments from this distribution center.
 - b. Create a query that prompts for the identification number of a product and returns a list of dealers that have this product in their inventory and the corresponding inventory level.
 - c. Create a query that prompts for the name of a state and returns the name and address of the distribution centers and dealers located in that state.
 - d. Create a query that prompts for a shipment number and returns the following information: shipment amount, shipment date, and the value of the shipment.
 - e. Create a query that lists the top three distribution centers (distribution centers that have the highest sales in the current year).
 - f. Create a query that lists the top three dealers (dealers that have the highest sales in the current year).
 - g. Create a query that presents the three most preferred products (products that sold most).

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: products, distribution centers, dealers, shipments, etc. These forms allow the user to add, update, and delete information about distribution centers, dealers, shipments, etc.
3. Create a form that allows the user to browse through the information about the products. Create a subform that, for the selected product, presents the quantity sold in the current year and the inventory level at each dealer. Include a command button that, when clicked on, lists the three most preferred products.
4. Create a query that allows the user to select a distribution center from a combo box. Create a subform that, for the selected distribution center, lists the names of the dealers served, distance to each dealer, and sales of each dealer in the current year. Sort this information by sales.
5. Create a “Search” form. This form includes seven command buttons. When a particular command button is clicked on, one of the search queries (presented previously) is run and the results are presented.

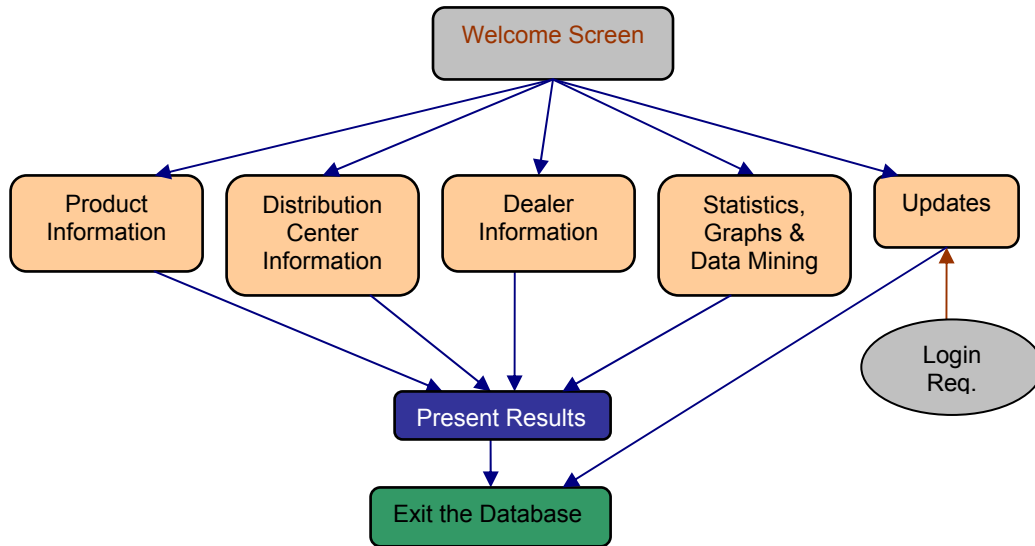
Design a logo for this database. The logo may include the picture of a Ford car, etc. Insert this logo in the forms created above as well as on the reports that you will be creating next. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create a report that presents detailed information about shipments made from a distribution center to a dealer on a particular day. This report should include the name and address of the dealer, the name and address of the distribution center, the names of the products shipped, the quantity shipped per product, the amount of money charged, and the day of the delivery.
2. Create a report that presents the transactions performed in a distribution center during a particular week. The reports should include transaction date and description, identification number, and name and address of the dealer involved in this transaction.
3. List all the products offered by Ford. For each product, present the unit price, inventory level (overall distribution centers and dealers), and total sales in the current year.
4. Use the chart wizard to perform the following tasks:
 - a. Graph the monthly sales for a particular product (overall dealers) during the last twelve months.
 - b. Graph the monthly transportation costs per dealer for the last twelve months.
 - c. Graph the end-of-month inventory for a particular product (overall dealers) for the last twelve months.
 - d. Graph the monthly sales for each distribution center during the last twelve months.
5. Use the label wizard to create labels with the address of each dealer.

Visual Basic.NET Application Development

This database application can be used by Ford managers, managers of the distribution centers and dealers, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important.

Product Information: The user can search the database to retrieve information about a particular product. We suggest that the search queries and forms already created in the Access Application Development section be included in here. These forms will enable the user to identify the following: the inventory level, total sales, and cost and price information about a particular product; the best selling products; etc.

Distribution Center: The user can browse the database to learn more about distribution centers. The user can identify the most successful distribution center, list the products offered by a distribution center, etc.

Dealer Information: This option provides information about the dealers.

Statistics, Graphics, and Data Mining: We encourage the use of statistical analysis tools, usage of plots (similar to the ones created in the Access Application Development section), etc. that enable the user to draw conclusions about the performance of a distribution center, dealer, or product.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about dealers, products, etc.

Web Extension

Users of this database are dealers, distribution centers, the database administrator, etc. A user can access the database from personal computers at home or in the office. To be

able to use the database, a user should be a member and have a login name and password. The users should be able to search the database to find out about products, distribution centers, shipments, etc. A member, however, should not be allowed to update/delete information about other members, shipments, products, etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Adapted from: "A supply Chain Design Case Study," The Logistics Institute, Georgia Institute of Technology.

15 *Inventory Routing Problem*

Problem Description

Coca-Cola Co. in Georgia keeps a fleet of vehicles to facilitate the delivery of its products to all customers in the state. Coca-Cola Co. has five production facilities that produce different products. The facilities produce at given rates and have ample storage capabilities for the products. Customers consume products at a given rate and have limited storage capabilities. Products are delivered to customers on weekly bases. A major challenge is to decide on the delivery routes for each vehicle. The assignment of the vehicles on different routes should be such that the overall costs are minimized.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Customer*: The main attributes are customer identification number, address, sign-in date, X (longitude) and Y (latitude) coordinates of their location, amount of time (fraction of an hour) required to make a stop at a customer, list of product(s) used by a customer, mean rate and standard deviation of the weekly consumption at a customer, an upper bound on the inventory level that can be held at a customer, name and telephone number of the contact person, etc.
2. *Employee*: The main attributes are employee identification number, name, address, annual salary, etc.
3. *Plant*: The main attributes are plant identification number, address, number of vehicles in the inventory, X (longitude) and Y (latitude) coordinates of their location, amount of time (fraction of an hour) required for making a stop at a plant while the driver is on a tour, list of products that a plant produces, etc.
4. *Product*: The main attributes are product identification number, name, description, average daily production (overall plants), current inventory level, etc.

Note: (a) When a product is ordered by a customer, the following information is recorded: transaction identification number, order date, delivery date, payment due date, quantity ordered, and order descriptions; (b) When the production of a product is completed, the following information is recorded: production date and quantity produced.

Access Application Development

The managers prepare, on weekly bases, a delivery schedule of the final products from the plants to the customers. A big concern in preparing the weekly schedule is deciding about the delivery routes. The assignment should be such that the resources are utilized well and costs are minimized. One could build an optimization model to solve this problem optimally. Obviously, this is not an easy task. Create the following queries to facilitate the management's decision about the weekly delivery routes.

Queries:

1. The following query will help the management decide about the customers that will be receiving a delivery on a particular day of the week:
 - a. List the customers based on the order date. Usually higher priority is given to orders submitted earlier. Present the order quantity for each customer.
 - b. List the customers based on the order quantity. Priority will be given to large orders.
 - c. List the customers that are located on the east/west/north/south side of the state. In each of the four queries created, list customer names and X and Y coordinates of their locations.
 - d. The assignment of the customers that will be receiving a shipment is done in such a way that the customers' inventory is not depleted. A customer might require an earlier delivery in the case that the inventory level is very low or there is an important upcoming event. These inquiries are recorded in the table orders under the transaction description. List the customers that have made an inquiry for change in the delivery date.
2. A truckload is enough to satisfy the weekly demand of a number of customers. Given the information about the customers to be visited on a particular day (the results from the previous set of queries), decide about grouping together the customers that will be served by the same truck. The assignment should be such that the total transportation costs are minimized. To help with this decision, create a query that presents for each customer the distance to the rest of the customers that will be receiving a delivery.
3. Create the following search queries:
 - a. Create a query that prompts for the name of a final product and returns its inventory level (overall plants), unit cost, and unit price.
 - b. Create a query that prompts for the name of a customer and returns a list of the products consumed by this customer and the weekly consumption rate.
 - c. Create a query that prompts for the name of a customer/plant and returns the name and address of other customers that are located nearby (within 30 miles). One can use the X and Y coordinates to calculate the Euclidean distance between the two locations.
 - d. Create a query that prompts for the name of a plant and returns a list of the products produced on that plant, total monthly production, and inventory level of each plant.
4. The following queries measure a company's performance:
 - a. Present monthly sales during the last twelve months.
 - b. Present the end-of-month inventory for the last twelve months.
 - c. List the customers that have signed-in with the company in the last twelve months.
 - d. List the three most popular products.

- e. List five of the best customers. Choose the customers based on total order quantity during the last twelve months.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: products, plants, customers, etc. These forms allow the user to add, update, and delete information about products, plants, customers, etc.
3. Create a “Search” form. This form will allow the user to run each of the search queries and view the results. One can use command buttons to run the search queries.
4. Create a form that allows the user to browse through the information about products. For each product, present the identification number, name, and average daily production. Create a subform that, for the selected product, presents a list of the plants that produce this product. Create a command button that, when clicked-on, presents a list of the customers that have ordered this product.
5. Create a form that allows the user to choose a plant (identification number) from a combo box. For the chosen plant, present the plant name, address, and name and telephone number of the contact person. Create a subform that presents for each plant a list of the products produced and corresponding production quantity. Create a command button that, when clicked on, prompts for the location of a plant and returns a list of the customers located in the vicinity (within 30 miles).
6. Create a form that allows the user to choose a customer (customer identification number) from a list box. For the chosen customer, present the following information: name, address, mean rate and standard deviation of the weekly consumption, an upper bound on the inventory level, and name and telephone number of the contact person. Create a command button that, when clicked on, provides a list of products ordered by this customer. Create another command button that, when clicked on, presents the name of the nearest plant to this customer.
7. Create a form that allows the user to browse through the information about the employees. For each employee, present the following information: employee identification number, name, and address; name of the plant the employee works at and name of the employee’s supervisor.
8. Use a tab control to create a form that presents the information about plants, products, and customers.
9. Create an order form. This form allows the user to place an order for a product. The following are some features one may include in this form:
 - a. Set the default value for the Order Date textbox at today’s date.
 - b. Set the default value for the Delivery Date textbox at one week from today’s date.
 - c. Use Option Group to allow the customer to choose one of the five plants to get the delivery from.
 - d. Use a combo box to allow the customer to choose a product.

Design a logo for this database. The logo may contain the picture of a Coca-Cola bottle, etc. One is encouraged to visit the Coca-Cola website to get ideas when designing this logo (www.cocacola.com). Insert the logo in the forms created above. Have the background color of the forms light red and the border color for the titles yellow. Include

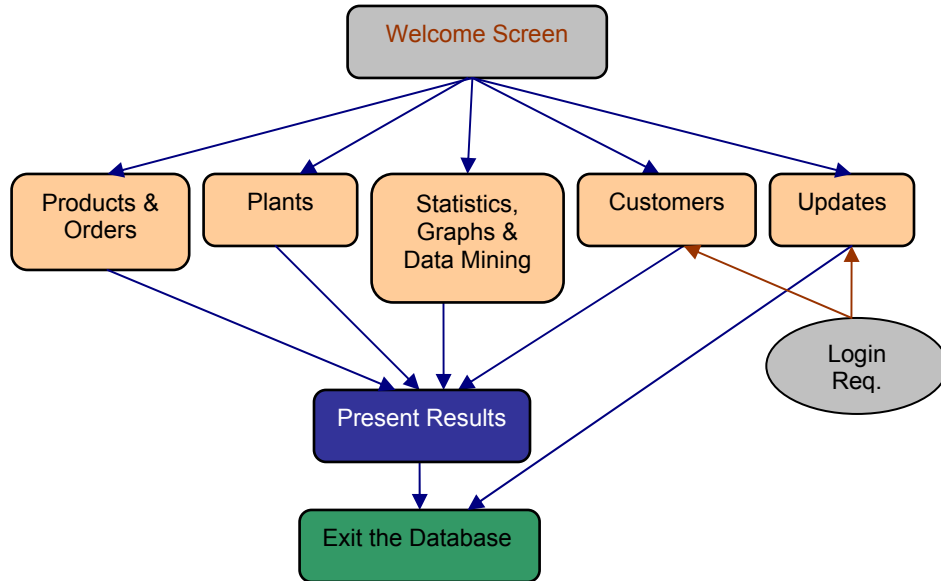
the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create next week's daily route report for each driver. The reports should indicate the plants from where each shipment will be initiated, the route to be followed, the customers that will be visited, and the name and amount of the products to be delivered.
2. Create the monthly plant activity report. This report presents for each plant the type of products produced during the last month and the corresponding quantities.
3. Create a customer order report. The report should include the following information about an order: order number, customer name and address, quantity ordered, order date, and delivery date.
4. Create a delivery costs report. This report presents the total delivery cost per week for the last 10 weeks. Use the chart wizard to plot this information.
5. Use the label wizard to create labels with the address of each customer.
6. Create the following "end of the year" performance reports:
 - a. List the five most popular products.
 - b. List the five most preferred customers. Choose the customers based on quantity ordered.
 - c. Present total revenues generated from sales.
 - d. Present the best plant of the year. (The plant that generated the highest revenues.)
7. Use the chart wizard to plot the following:
 - a. Total sales (quantity ordered) per month during the last twelve months.
 - b. Total production (overall plants) per month during the last twelve months.
 - c. Total distance traveled per month during the last twelve months.
 - d. Total transportation costs per month during the last twelve months.

Visual Basic.NET Application Development

This database application can be used by customers, plant managers, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Products & Orders: The user can browse the database about products and place orders.

Plants: The user can browse the database to learn more about each plant (e.g., product produced at a plant, quantity produced, plant location, etc.).

Statistics, graphs, and data mining: this option provides various statistics, plots trends, and performs data analysis.

Customers: This form allows the user to browse through the information about each customer. Only the plant managers are allowed to have access in this part of the database.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about products, customers, plants, orders, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a customer, a plant manager, the database administrator, etc. The user should be able to search for products, plants, orders, etc. To be able to place an order, the customer needs a login name and a password. Only the managers should have access to customer information and order forms. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Adapted from: “*A Supply Chain Design Case Study.*” The Logistics Institute at Georgia Institute of Technology.

Problem Description

For the last ten years, Jane has been the scholarship billing coordinator for the student accounts and university receivables (University Cashier) at the University of North Carolina, Chapel Hill. Her responsibilities include scholarship billing during the academic terms: fall, spring, summer I, and summer II. As the number of students receiving scholarships increased, Jane was facing difficulties in managing the billing process. Billing was done manually, and that was a real challenge.

In 1993, the university adopted a Student Information System (SIS). This system was supported by a large database used by various departments for administrative purposes. However, Jane was still recording the scholarship billing transactions in Excel spreadsheets. The Excel spreadsheets and Word documents were used to generate invoices and reports. This process was time-consuming and fraught with errors. Upon implementation of the SIS, Jane was assigned the task of designing a module dedicated to special scholarship billing. Jane has a good understanding of how the system works; however, she never took a database management class. Jane is asking you to build the database system for her. Below we present her description of the current billing system.

The university has roughly 450 scholarship athletes during the fall and spring terms. In summer session I of 1999, there were approximately 160 students; 90-100 participated in summer session II of 1999. A full scholarship includes the following: tuition payments, fees, and payments for room, food, and books. A small number of athletes would qualify for a full scholarship. Usually, athletes qualify for tuition payments only, room payments only, etc. Sometimes, athletes may receive any combination of tuition, fees, room, food, or books with any imaginable combination of caps (e.g., full fee payments and \$1,000 in tuition; or full payment for books and \$500 per month for boarding).

The coach recruits the athletes and signs an Athletic Scholarship Agreement. The University Department of Athletics sends a copy of the agreement to the Office of Scholarships, Student Aid and the University Cashier. Student Aid mails an award letter to the students and mails a copy of the award to the University Cashier, bookstore, on-campus housing office, and on-campus food court notifying them about the scholarships awarded.

Once a month, the University Cashier is billed by the bookstore, on-campus housing office, off-campus apartments (if any athlete is living off campus), etc. The University Cashier verifies the bills and makes payments.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Athlete*: The main attributes are social security number, name, gender, birthday, address, citizenship, college, major, scholarship type, special program, sport, tax status, classification (e.g., first year professional, ..., fourth year professional, graduate doctorate, graduate masters, exchange student), etc.
2. *Scholarship*: The main attributes are identification number, name, description (e.g., state scholarship, federal, military, international, mixed, etc.), name of the person responsible, amount of money provided, etc.
3. *Vendor*: The main attributes are account number, name, address, telephone number, fax number, name of the contact person, etc. Vendors usually are the on-campus bookstore, the on-campus food courts, on-campus and off-campus housing providers, etc.
4. *Club*: The main attributes are name (e.g., football, soccer, baseball, basketball, volleyball, tennis, track and field club, etc.), address, name and telephone number of the contact person, etc.

Note that once a month, each vendor mails to the University Cashier detailed information about the transactions performed by the athletes. The University Cashier makes a payment (usually) equal to the amount charged. For each transaction, the database should keep record of the following: transaction identification number, amount charged, amount paid, payment date, and descriptions.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create a query that lists the payments made by the University Cashier during the current month. Include the following in this query: payment date, description, amount paid, name of the vendor, and name of the student involved in this transaction.
2. Create a query that lists the social security number, name, and address of the students that have received an award during the current academic year. Include in this query the total amount of money awarded for each student.
3. Create the following search queries:
 - a. List the athletes that are receiving full scholarships.
 - b. Present the club that has the highest number of athletes receiving scholarships.
 - c. List the three most popular clubs.
 - d. List all the athletes that are currently enrolled in a graduate program. Include in this query the name of the corresponding graduate program.
 - e. List the name and address of the off-campus vendors.
4. Create a query that prompts for the name of an athlete and returns the total payment made by the university during the current academic year.
5. Create a query that prompts for the name of a vendor and returns the total amount of money paid to the vendor during the current academic year.

Forms:

1. Create a user sign-in form together with a registration form for new users.

2. Create the following data entry forms that are used for database administrative functions: athletes, scholarships, vendors, etc. These forms allow the user to add, update, and delete information about athletes, scholarships, vendors, etc.
3. Create a search form. This form allows the user to search the database and learn more about athletes, scholarships, etc. Using this form, one can run the search queries presented above. Command buttons can be used to run the search queries and present the corresponding results.
4. Create a form that allows the user to browse through the different types of scholarships provided by the university. Create a subform that presents a list of the students who are receiving this particular scholarship. Create a command button that, when clicked on, calculates and presents the total amount of money the university is currently paying for this type of scholarship.
5. Create a form that allows the user to choose from a list box the name of an athlete. Create a subform that presents the following information about the chosen athlete: name of the sport club, degree pursued, name of the department and college the athlete is enrolled in, type of scholarship received, etc.
6. Create a form that allows the user to choose from a combo box the name of a sport club. Create a subform that for the selected sport club presents a list with the social security number and name of the athletes who are members of the club. For each athlete, report the amount of money awarded. Include a command button that, when clicked-on, returns the three most popular sport clubs. Include another command button that, when clicked-on, returns the number of athletes currently participating in each sport club.

Design a logo for this database. The logo may contain the picture of a football, basketball, athlete, etc. Insert this logo in the forms created above. Have the background color of the forms orange and the border color for the titles blue. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

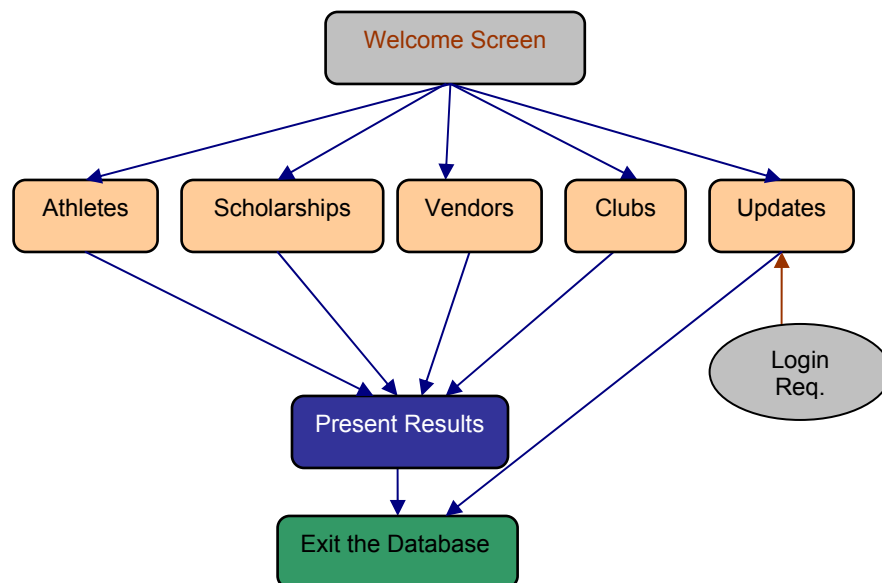
Reports

1. Create a report that presents a detailed summary of the transactions performed in the current month. The report contains the following information: account name, vendor name, name of the student involved in this transaction, payment date, amount charged, and descriptions. Group the information by account number and vendor name. Within each group, sort the information by amount paid. Calculate and present the total amount paid to each vendor in the current month.
2. Create an “end of the year” transaction report. This report presents a summary of the payments made to each vendor during the past academic year. The report contains the following information: account number, vendor name, and total amount of money paid. Sort the information by vendor name. Calculate and present the total amount of money paid to vendors.
3. Use the label wizard to create labels with the address of vendors and athletes.
4. Create the following “end of the year” reports:
 - a. List the three most popular sport clubs.
 - b. List the students who receive full scholarships.
 - c. Report the account numbers and names of the off-campus vendors.

- d. Report the names and social security numbers of the athletes pursuing a graduate degree.
5. Use the chart wizard to plot the following:
 - a. Total payments to vendors per month during the last twelve months.
 - b. Total number of scholarships awarded per year during the last ten years.
 - c. Total amount paid for each sport club per year during the last ten years.

Visual Basic.NET Application Development

This database application can be used by athletes, sport club managers, vendors, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Athletes: The user can browse the database to learn more about the athletes.

Scholarships: The user can browse the database to learn more about each scholarship (e.g., sport activities that are awarded scholarships, scholarship types, etc.).

Vendors: This option provides information about the vendors that do business with the university.

Clubs: This form allows the user to browse through the information about each sport club.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about athletes, sport clubs, vendors, scholarships, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a vendor, a sport club manager, an athlete, the database administrator, etc. The user should be able to search the database and learn more about vendors, scholarships, athletes, etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Adapted from a case study found at <http://www.ibiblio.org/faint/finosaur/db>.

Problem Description

The concept of pharmaceutical care practice is quite novel in the USA. It provides patients with the attention and counseling that they may not be getting from their doctor. Studies have shown that billions of dollars would have been saved in health care if patients had been getting personalized counseling. We want to build a pharmaceutical care system that will help the pharmacists to do the following: (a) decide about the remedies that should be used by patients to cure a particular health condition and (b) provide guidelines when counseling a patient. Further, the system provides the following information about the drugs kept in the inventory: current inventory level, average monthly consumption, expiration date, etc.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Patient*: The main attributes are social security number, name (consisting of first, middle, and last names), birthday, gender, address (consisting of street address, city, state, and zip), height, weight, insurance information (consisting of carrier name, address, and telephone number; plan number, start date, and end date), patient's health history (consisting of date, disease, description, etc.), history of the patient's lifestyle (consisting of date, description of the exercises performed, frequency, length, etc.), etc.
2. *Drug*: The main attributes are identification number, name, price, description of side effects, etc.
3. *Physician*: The main attributes are identification number, name, address, telephone number, fax number, specialization, employer's name and address, etc.
4. *Pharmacist*: The main attributes are identification number, name, address, telephone number, fax number, specialization, etc.
5. *Disease*: The main attributes are name, description of the symptoms, description of the precautionary actions to be taken, etc.

Note that (a) When a patient buys a drug, the following information is recorded: purchase date, quantity purchased, and amount paid; (b) The same drug can be used to cure different diseases; however, the amount of drug used differs, so details about the amount of a particular drug needed to treat a particular disease is recorded.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create a query that prompts for the name of a disease and returns descriptions of the symptoms and precautionary actions to be taken.
2. Create a query that prompts for the name of a disease and returns a list with the name and amount of the drugs needed to cure this disease.
3. Create a query that prompts for the name of a disease and returns a list with the name, address, and telephone number of the physicians who are specialized in curing this particular disease.
4. Create a query that prompts for the name of a disease and returns a list with the names, addresses, and telephone numbers of the patients who suffer from this disease.
5. Create a query that prompts for the name of a patient and returns a list of the drugs that patient purchased. Include in this query the purchase date, drug name, and quantity purchased.
6. Patients who are taking a particular drug are often notified about better substitutes, recent results from scientific research about side effects of this drug, etc. Create a query that prompts for the identification number of the drug and returns a list with the names, addresses, and telephone numbers of the patients who are currently taking the drug.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: patients, diseases, pharmacists, physicians, etc. These forms allow the user to add, update, and delete information about patients, diseases, pharmacists, physicians, etc.
3. Create a form that allows the user to browse through the information about the patients. For each patient, present the following: social security number, name, address, gender, and birthday. Create a subform that presents the patient's health history. Include a command button that, when clicked on, returns a list with the name of the drugs used by this patient. For each drug, report the date the drug was used and quantity used.
4. Create a query that allows the user to choose the identification number of a drug from a combo box. The following information about the chosen drug is presented: name, expiration date, and description of side effects. Create a subform that lists the disease that can be cured using this particular drug. Include a command button that, when clicked-on, returns a list with the name, the address, and the telephone number of the patients who are currently taking this drug.
5. Create a query that allows the user to choose from a list box the name of a disease. The following information is presented for the chosen disease: description of the symptoms and description of the precautionary actions to be taken. Include a command button that, when clicked on, returns a list with the name and address of the physicians who are specialized in curing this particular disease. Create another command button that, when clicked on, returns a list with the name and address of the patients suffering from this disease.

Design a logo for this database. The logo may contain the picture of a medicine, etc. Insert this logo in the forms created above. Have the background color of the forms open blue and the border color for the title dark blue. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

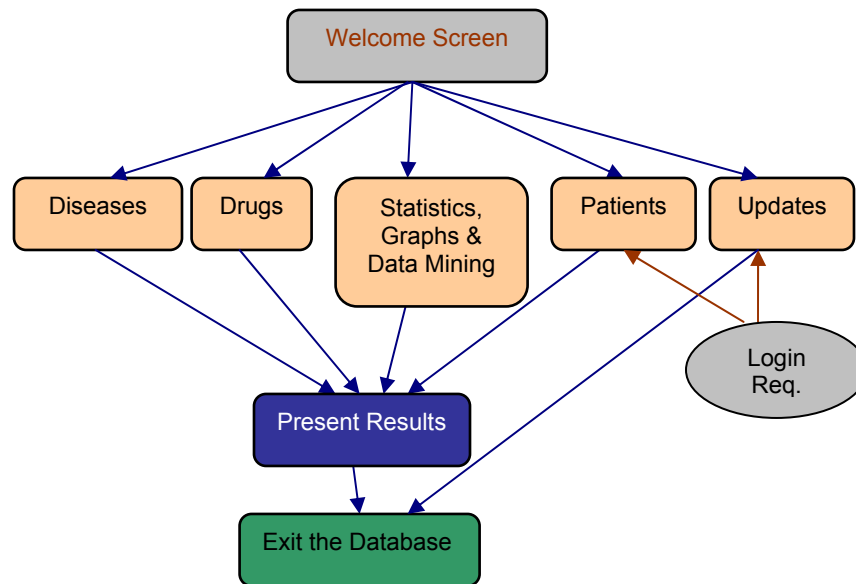
Reports

1. The following report is created each time a patient visits the pharmacy. This report will be handed to the patient together with the drugs. The report should include the following: visit date; patient name, address, and telephone number; name of the disease; precaution actions to be taken by the patient; name and quantity of the drug(s) to be used; timetable for taking the drugs; and refill dates.
2. The following report is handed to the insurance company after each patient's visit. The report includes the following: visit date; patient's plan number, name, and address; name of the drug purchased, quantity purchased, amount to be paid, and payment date; name, address, and telephone number of the insurance company.
3. The following report about the drugs inventory is prepared at the end of each month. The report includes the name of the drugs in the inventory, the corresponding inventory level, expiration date, average monthly consumption, and price.
4. Use the label wizard to create labels with the addresses of the insurance companies, patients, and physicians.
5. Use the chart wizard to plot the following:
 - a. The total number of patients signed-in per year during the last five years.
 - b. The total number of patients' visits per month during the last twelve months.
 - c. The value of drugs consumed per month during the last twelve months.
 - d. The number of patients diagnosed with heart diseases per year in the last five years.

Visual Basic.NET Application Development

This database application can be used by patients, physicians, pharmacists, etc. In the following figure we present a tentative layout of the system.

In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries and forms already created in the Access Application Development section be included in here.



Diseases: The user can search the database to retrieve information about different diseases.

Drugs: The user can browse the database to learn more about drugs or find drugs that cure a particular disease, etc.

Statistics, graphs, and data mining: This option provides various statistics, plots trends, and performs data analysis.

Patients: The user can browse through or search the database to find more about a particular patient. Only the pharmacists and physicians have access to this part of the database.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about patients, diseases, drugs, etc.

Web Extension

The next step of this project is to develop an ASP.NET web application that will enable the users to access this database from personal computers at home or in the office. Having web access to this database will allow patients to place orders on-line, check the status of their order, or browse the database to learn about the remedies that should be used to cure a particular health condition. To be able to place an order, the user needs a login name and a password. The pharmacists and physicians access the database to learn more about patients, to find the inventory level of a particular drug, etc. Only the database administrator can have access to the update forms. In developing the ASP.NET web application, create forms similar to the ones described in the VB.NET Application section.

18 *Internet and Exchange Billing*

Problem Description

A small telecommunication company located in Chicago provides basic local calling service, long-distance calling service, and internet service to customers located in the south part of Illinois. Currently, the information about customers, billing transactions, rates offered, etc. is recorded in Excel spreadsheets. The company is interested in building a decision support system that will (a) help manage the information about customers and transactions; (b) facilitate the process of billing the customers; (c) help the management make decisions about extending their services; etc.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Customer*: The main attributes are social security number, name, address, telephone number, sign-in date, etc.
2. *Service*: The main attributes are name and description. Note that (a) for the internet usage and basic local call package, a fixed monthly rate is charged; (b) for the long-distance calls, the customer could buy the “special” long-distance package that costs \$5/month and then for each call pay \$0.04/minute, or otherwise pay \$0.1/minute.

Note that the company records the following information about the customers’ monthly consumption: date, consumption amount, amount due, and due date.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create the following search queries:
 - a. Create a query that prompts for the name of a customer and returns the type of services offered to this customer.
 - b. Create a query that prompts for the name of a service (e.g., internet, long-distance calls, etc.) and returns the monthly revenues of the company from this particular service.
 - c. Create a query that prompts for the name of a customer and returns the monthly consumption of this customer in the last twelve months.
2. Create a query that reports at the end of a month the total amount due from each customer. Include in this query the name of the customer, and itemize the customer’s monthly consumption by service type.

3. The following information demonstrates the progress of this business in the last year.
 - a. List the total number of new internet connections per month in the last year.
 - b. List the monthly income collected from internet, local calls, and long-distance calls during the last year.
4. The management has realized that some of the customers have made it a habit to submit late payments. Service restrictions will be imposed on these customers. List the customers who have made at least five late payments.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: customers, services, etc. These forms allow the user to add, update, and delete information about customers, services, etc.
3. Create a form that allows the user to browse through the name of the customers. For the selected customer, present the following: social security number, name, address, type of service provided, and telephone number. Create a subform that presents the following information about the selected customer: a list of monthly payments made during the last twelve months and the corresponding monthly consumption by service type (e.g., internet service, local calls, or long-distance calls). Insert a command button that, when clicked-on, presents the amount due and due date for the current month's bill of this customer.
4. Create a form that allows the user to choose from a combo box a service type. Create a subform that lists the name, address, and average monthly payments of the customers who use this particular service type. Insert a command button that, when clicked on, calculates the total revenues from this service in the last twelve months. Insert another command button that, when clicked-on, returns a list of the new connections set during the last month.
5. Create a "Search" form. This form includes at least three command buttons. When a particular command button is clicked-on, one of the search queries (presented previously) is run and the results are presented.
6. Use a tab control to create a form that presents the information about different services provided by the company.

Design a logo for this database. The logo may include the picture of a computer, telephone, etc. Insert this logo in the forms created above as well as on the reports that you will be creating next. Have the background color of the forms light yellow and the border color for the titles dark green. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

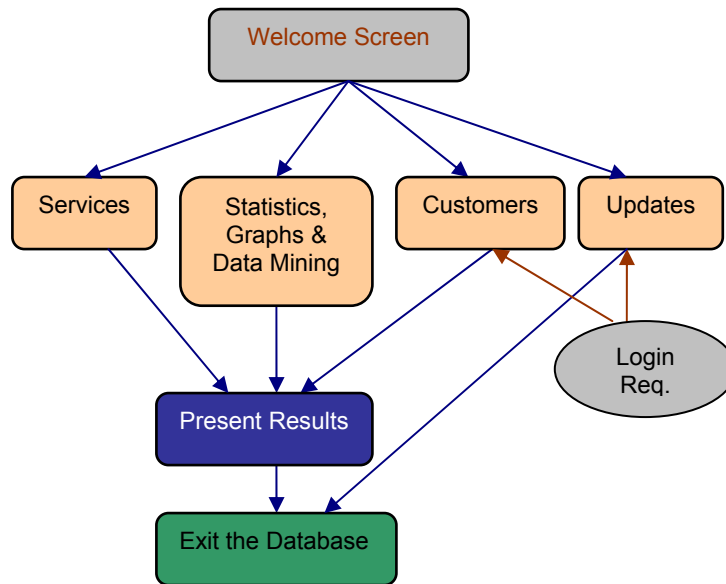
Reports

1. The following reports are created at the end of the month and include the following:
 - a. List of the customers who use the internet service. For each customer, report the following: name, address, amount due, and due date.
 - b. List of the customers who use the local phone service. For each customer, report the following: name, address, amount due, and due date.

- c. List of the customers who use the long-distance calling service. For each customer, report the following: name, address, amount due, and due date.
2. Create the end-of-month billing report for a customer. This report should include the following: customer's social security number, name, and address; monthly consumption by service type; amount due by service type; total amount due; and due date.
3. Create the end-of-month transaction report. This report presents a list with the name of the customers who made a payment in the last month and the amount paid. Calculate the total amount received (overall customers).
4. Use the label wizard to create labels with the addresses of customers.
5. Use the chart wizard to plot the following:
 - a. Total number of new internet connections per month during the last twelve months.
 - b. Total monthly revenues by service type in the last twelve months.

Visual Basic.NET Application Development

This database application can be used by clients, managers, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries and forms already created in the Access Application Development section be included in here.

Services: The user can search the database to retrieve information about different services offered, such as service type, rates offered, etc.

Statistics, graphs, and data mining: This option provides various statistics, plots trends, and performs data analysis.

Customers: The user can browse through or search the database to find more about a particular customer. Only the managers have access to this part of the database.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about customers, services, etc.

Web Extension

The next step of this project is to develop an ASP.NET web application that will enable the users to access this database from personal computers at home or in the office. Having web access to this database will allow customers to set service connections on-line, check their accounts, or browse the database to learn about promotions. To be able to see a billing statement, the user needs a login name and a password. Only the database administrator can have access to the update forms.

In developing the ASP.NET web application, create forms similar to the ones described in the VB.NET Application section.

Problem Description

An academic institution keeps the information about courses, teachers, and classrooms in a database. This database is then used each semester to prepare the timetable for the courses offered. The course timetable should be such that: (a) each course is assigned to the required number of periods; (b) a teacher is not assigned to teach more than one course in a class period; (c) a course is assigned to an available room in a period; (d) the number of students taking a course fills at least 75% of the classroom capacity but no more than 105%; (e) in any class period, the number of courses offered does not exceed the number of classrooms available. Courses are grouped together based on the curriculum requirements. For example, senior students in the mechanical engineering department should take certain courses. Courses belonging in the same group are not allowed to be assigned to the same class period.

We propose an algorithm to help in preparing the course timetable. This is a semi-automatic approach. It takes the previous year's/semester's timetable as input and then updates the timetable based on the changes in the requirements.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Course*: The main attributes are identification number, name, number of periods per week assigned to this course, expected number of students, name of the group of courses to which it belongs, etc.
2. *Teacher*: The main attributes are identification number, name, on-campus address, department, area of specialty, etc.
3. *Period*: The main attributes are period number, day of the week, start time, and end time.
4. *Room*: The main attributes are room number, type, capacity, etc.

Note that the timetable of courses presents the assignment of a teacher to teach a particular course in a particular class period. Attributes particular to this assignment are student evaluation, number of students taking the course, and date (e.g., during the fall semester of 2004, 20 students took the *Operations Research* course taught by *Professor J. Smith* in classroom *Weil Hall 230*, offered *Monday 1-3 p.m.*).

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. For each course, count the number of periods this course is offered in a week.
2. Create a query that prompts for the name of a professor and returns a list of courses offered by this professor in the current semester. For each course, present the classroom number and class period.
3. Create a query that prompts for the name of a course and lists the names of the professors that have taught this course at least once.
4. Create a query that prompts for a class period and lists the names of the courses that are offered in this class period and the corresponding classroom number and address.
5. Create a query that prompts for a class period and returns the total number of courses scheduled in this period, the total number of classrooms occupied, and the total number of available classrooms.
6. Create a query that lists the courses offered in a particular semester. For each course, present the number of students registered to take the course, the classroom number, and the classroom capacity, and calculate the percentage of the capacity filled by the students registered.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: courses, teachers, classrooms, etc. These forms allow the user to add, update, and delete information about courses, teachers, classrooms, etc.
3. Create a form that allows the user to browse through the names of the teachers registered in this database. Create a subform that lists for each teacher the name of the courses that teacher is currently teaching. For each course, present the following: the name, the corresponding classroom number and capacity, the class period, the number of students taking the course, and the percent of the capacity filled by the students. Insert a command button that, when clicked on, prompts for the name of a teacher and returns a list of courses taught so far by this teacher.
4. Create a form that allows the user to choose a class period from a combo box. Create a subform that presents for the selected class period a list of the courses offered and the corresponding classroom number. Insert a command button that, when clicked-on, prompts for a class period and returns a list of the classrooms that are available in that class period.
5. Create a form that allows the user to choose the name of a course from a list box. Create a subform that presents the following information about the selected course: the name of the teacher teaching the course, the class period, the classroom number and capacity, and the number of students registered to take this course. Insert a command button that, when clicked-on, returns a list of all courses offered in the current semester. Insert another command button that, when clicked-on, returns the name of the group this course belongs to.
6. Create a form that allows the user to select a classroom from a combo box. Create a subform that presents for the selected classroom a list of the courses taught in this classroom and the corresponding class period. Insert a command button that, when clicked-on, returns the class periods this classroom is available.

Design a logo for this database. The logo may contain the picture of an academic institution, a textbook, etc. Insert this logo in the forms created above. Have the background color of the forms yellow and the border color for the titles blue. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

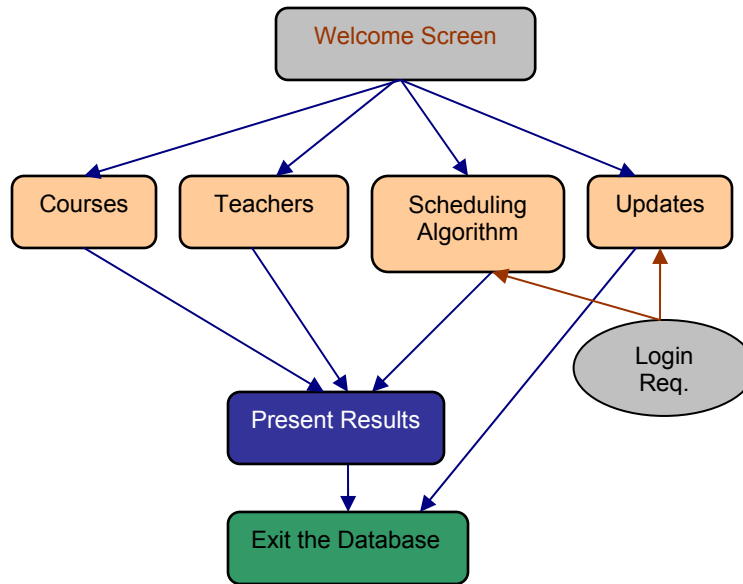
Reports:

1. Create a report that lists the identification number and name of the courses offered in the current semester. For each course, present the following: the name of the teacher, the name of the group(s) this course belongs to, the classroom number and address, and the corresponding class period.
2. Create a report that lists the names of the teachers who are teaching a class in the current semester. For each teacher, list the names of the courses that teacher is teaching and the corresponding class periods.
3. Create a report that presents for each classroom the class periods that this classroom is available.
4. Use the label wizard to create labels with the on-campus addresses of the teachers.
5. Use the chart wizard to plot the following:
 - a. Total number of courses offered per semester during the last four years.
 - b. Total number of teachers offering classes in this academic institution per semester during the last four years.
 - c. Total number of students registered to take the Operations Research course per semester during the last four years.
 - d. Most of the teachers would prefer to teach in the morning from 10 to 11 a.m. Present the total number of classrooms available during this class period. Group this information by classroom type.

Visual Basic.NET Application Development

This database application can be used by teachers, students, etc. In the following figure we present a tentative layout of the system.

In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.



Courses: The user can browse the database to learn more about the courses offered: the courses offered in a particular semester and the name of the teachers teaching the course, students' evaluation of a course, the timetable for a course, etc.

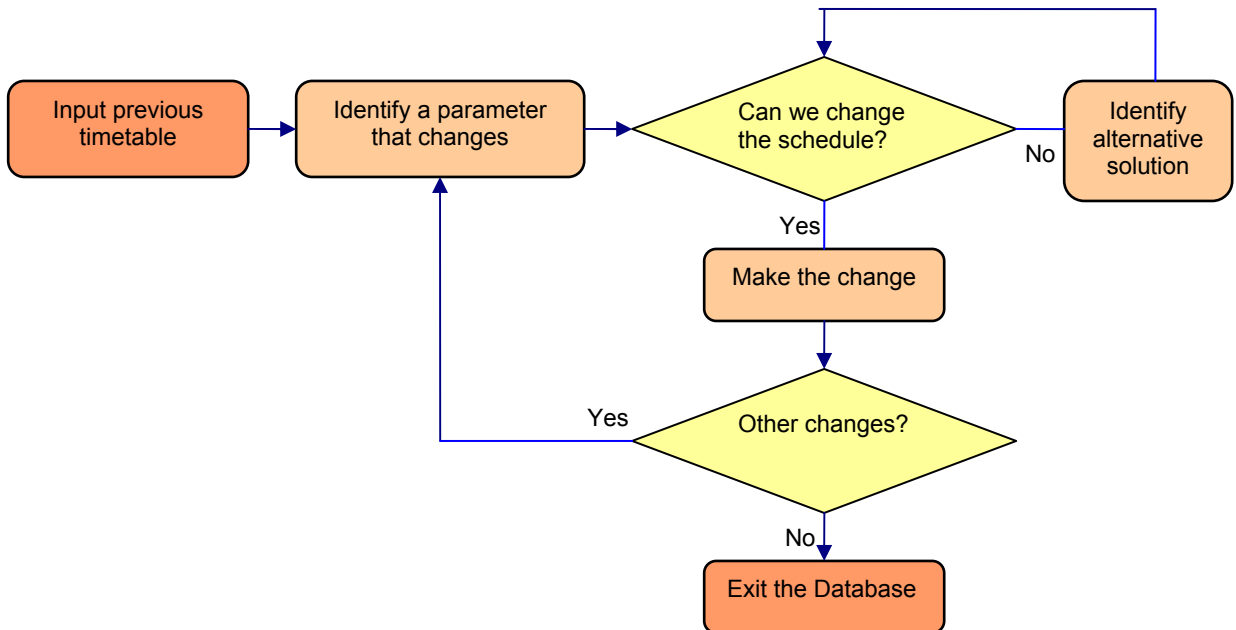
Teachers: The user can browse the database to learn more about the teachers: a list of courses offered by a particular teacher, the timetable of a teacher in the current semester, etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about courses, teachers, classrooms, etc.

Scheduling Algorithm: We provide an algorithm that can be used to create the course timetable for this academic institution. This is a semi-automatic approach that takes the previous year's/semester's timetable as input. A list of updates to be made is inputted as well. The timetable is then updated to reflect the changes in the courses offered.

Input previous timetable: The current semester's schedule is used as an input to create the next semester's schedule.

Identify a parameter that changes: The following are some of the parameters that can change: (a) a new course is offered; (b) the number of students registering for the course is larger than classroom capacity; (c) a teacher prefers to teach in a different time period; etc.



Identify alternative solution: There are alternative solutions to deal with changes in the course timetable. For example: Professor Smith is offering a new class next semester. He prefers to teach this class Mondays and Thursdays during the 4th class period. There are 15 students registered for the class so far, and it is expected that five more students will be registered. We have to identify whether there are available classrooms during this class period. We run a query that lists all the available classrooms during the specified class period. From this list we choose the classroom that satisfies the capacity requirements and update the timetable. If there is no available classroom, we have to select a different class period for this course, and this class period should not conflict with the schedule of Professor Smith.

Make the change: In the case that the solution found is feasible, we update the timetable and go back to identifying other parameters that have changed.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a student, a teacher, the database administrator, etc. Users should be able to search the database to identify their schedule. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

Hospital managers are concerned not only with managing the flow of the inpatients and outpatients at a hospital, but also with managing the personnel, equipments, operating hours, etc. Very often managers have to decide about hiring new personnel, setting operating hours that do not conflict with the schedule of the surgeon, buying new equipments, replacing or buying beds, etc. A survey is conducted every year in order to get feedback from the patients coming to the Outdoor Patient Division (OPD). The survey captures the following: the time a patient enters the hospital, the time the patient leaves the hospital, the departments that were visited, and the services that were provided to the patient. The survey is conducted on pre-defined days of the week and pre-defined seasons of the year.

The data collected from this survey is stored in a database. The managers need help in building a decision support system that would analyze the data and provide recommendations about improving the service offered. The management is mainly concerned with the following: (a) deciding how many doctors and nurses to hire, (b) setting operating hours that do not conflict with surgeons' and patients' schedules, (c) deciding how much and what equipment to buy; (d) deciding about buying extra beds or replacing existing ones, etc.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Doctor*: The main attributes are social security number, name, address, area of specialty, annual salary, etc.
2. *Nurse*: The main attributes are social security number, name, address, area of specialty, annual salary, etc.
3. *Patients*: The main attributes are social security number, name, address, gender, birthday, etc.
4. *Department*: The main attributes are name, address, descriptions, etc. A department may have sub-divisions within. Each sub-section has its own name and address.
5. *Equipment*: The main attributes are identification number, name, purchase price, purchase date, description of its functionalities, etc.
6. *Room*: The main attributes are room number, address, capacity, description of room functionalities, etc. The rooms of the hospital are classified into: doctor offices, waiting areas, surgery rooms, equipment rooms, and wards. There are different ward types. Wards of the same type have the same number of beds. The equipment rooms are equipped with different types of equipment, etc.

7. *Survey*: The main attributes are identification number, in-time, out-time, etc. This entity type presents the information about each OPD survey collected by the hospital.

Note that when a patient visits a doctor, the following information is recorded: visit date, patient's symptoms, and description of doctor recommendations. Doctors and nurses consult with patients during their working hours.

Access Application Development

Note that the data from the surveys is collected and entered manually into this database. Therefore, this data is prone to errors. Use validation rules to prohibit data entry errors, or set provisions that check the validity of data. The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following are some indicators of the quality of the service provided to patients: average waiting time it takes a patient to see a doctor in a particular department, average time a patient spends talking/consulting with a doctor in a particular department, total time a patient spends in the hospital during a visit, etc. Create queries that would provide this information. To create the queries, use the data collected from the surveys.
2. It is important to identify the bottleneck department(s) of the hospital. These departments usually have high patient-waiting times or a large number of patient visits. To solve this problem, more nurses and doctors could be hired. However, a re-scheduling of the working hours of the current employees may help if the high waiting times are usually during specific time periods. Experience has shown the patient waiting time can be reduced by promoting the arrival of patients during the off-peak hours (say, in the afternoon).
 - a. Create a crosstab query that presents for each department the average number of patient visits during the following time periods: 8-11 a.m.; 11 a.m.-2 p.m.; 2-5 p.m.; 5-10 p.m.; 10 p.m.-3 a.m. and 3-8 a.m.
 - b. Create a crosstab query that presents the average patient waiting time in a department during the time periods defined in 2.a.
 - c. Create a crosstab query that presents the maximum patient waiting time in a department during the time periods defined in 2.a.
 - d. Create a crosstab query that presents the minimum patient waiting time in a department during the time periods defined in 2.a.
 - e. Create another crosstab query that presents the total number of doctors and nurses available (in each department) during the time periods defined in 2.a.
3. Create a query that presents the average daily utilization of each piece of equipment. Present the corresponding age and purchase price.
4. Create a query that presents the average daily utilization per ward. For each ward, present the total number of beds and the total number of beds available.
5. Create a query that presents the following information about each department in the hospital: total number of doctors, total number of nurses, amount of money that goes for doctors' salary, amount of money that goes for nurses' salary, average number of patients a doctor (or nurse) consults per day, and average number of patient visits per day.

6. Create the following search queries:
 - a. Create a query that prompts for the name of a disease and returns a list of the doctors who can cure this particular disease.
 - b. Create a query that prompts for the name of an equipment and returns a list of its functionalities.
 - c. Create a query that prompts for the name of a nurse and returns that nurse's weekly schedule.
 - d. Create a query that prompts for the name of a doctor and returns that doctor's weekly schedule.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: doctors, nurses, equipment, etc. These forms allow the user to add, update, and delete information about doctors, nurses, equipment, etc.
3. Create a form that allows the user to browse through the information about the doctors. Create a subform that presents for each doctor the social security number and name of the patients that doctor has visited. Create a command button that, when clicked-on, calculates and presents the average, minimum, and maximum amount of time a patient spends when consulting this doctor. Create another command button that, when clicked- on, prompts for the name of a department and returns a list of all the doctors and nurses working there. Create a command button that, when clicked-on, presents the schedule of the doctor.
4. Create a form that allows the user to choose from a combo box the identification number of a survey. Create a subform that presents the following information about this survey: name of the patient who filled out the survey, reason for this visit, name of the doctors the patient saw, description of the patient's symptoms and doctor's recommendations, total waiting time, total time spent with the doctor, and total visit time. Create a command button that, when clicked-on, lists the visits the patient has had so far with the hospital.
5. Create a form that allows the user to choose the identification number of one of the hospital's equipments from a list box. Create a subform that presents the following information about this equipment: name, purchase price, purchase date, and descriptions of its functionalities. Create a command button that, when clicked-on, calculates the average daily usage of this equipment. Create another command button that, when clicked-on, calculates the total number of similar equipment owned by the hospital and their location.
6. Create a query that allows the user to choose the name of a patient from a combo box. Create a subform that presents a list of the patient's visits during the last year. For each visit, present the following: visit date, name of the doctor consulted, reason for this visit, doctor's recommendations, waiting time, consulting time, and total time of the visit.

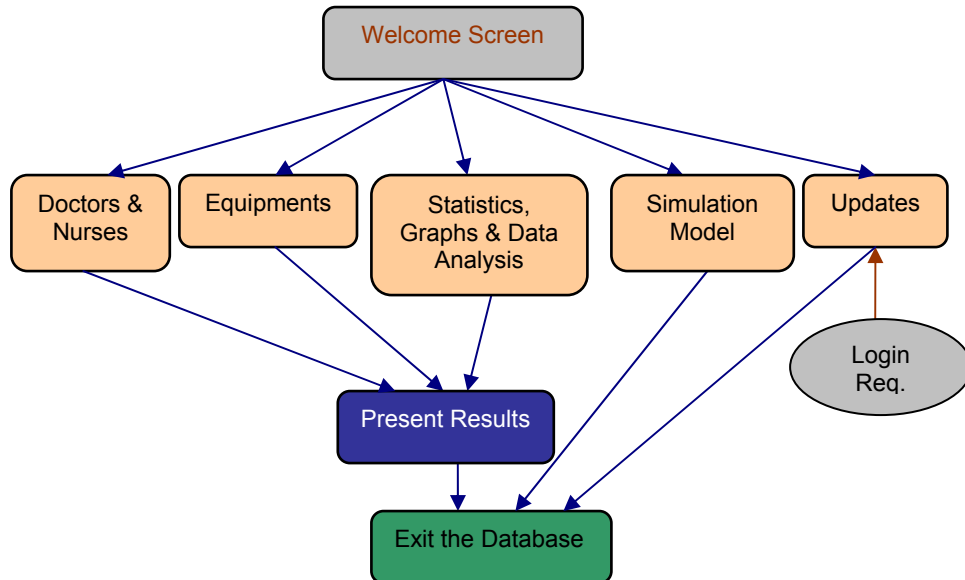
Design a logo for this database. The logo may contain the picture of a hospital, a doctor, etc. Insert this logo in the forms created above. Have the background color of the forms light blue and the border color for the titles dark blue. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Use the chart wizard to plot the following:
 - a. The average daily number of visits per time period. The following are the time periods to be considered: 8-11 a.m.; 11 a.m.-2 p.m.; 2-5 p.m.; 5-10 p.m.; 10 p.m.-3 a.m. and 3-8 a.m.
 - b. The maximum daily number of visits per time period.
 - c. The minimum daily number of visits per time period.
 - d. The total number of doctors available (overall departments) per time period.
 - e. The total number of nurses available (overall departments) per time period.
 - f. The overall equipment utilization rate per time period.
 - g. The overall ward utilization by ward type.
 - h. The average patient waiting time by department.
 - i. The maximum patient waiting time by department.
 - j. The minimum patient waiting time by department.
 - k. The average patient consulting time by department.
 - l. The maximum patient consulting time by department.
 - m. The minimum patient consulting time by department.
 - n. The average patient waiting time by time period. Patients are grouped into different time periods based on the time they came to the hospital.
 - o. The maximum patient waiting time by time period.
 - p. The minimum patient waiting time by time period.
 - q. The average patient consulting time by time period.
 - r. The maximum patient consulting time by time period.
 - s. The minimum patient consulting time by time period.
2. Use the label wizard to create labels with patient addresses.
3. Create a report that presents for each department the department name and address and a list of the doctors and nurses working in this department. For each doctor, present the area of specialty. Group the information by department. Within each group, sort the information by area of specialty.
4. Create a report that lists all the patients who visited the hospital during the current year. Group this information by patient name. For each patient, present the name and address. For each visit, present the following: visit date, patient's symptoms, doctor's recommendations, total doctor consulting time, and total waiting time.
5. Create a report that presents the current schedule of the doctors and nurses of the hospital.

Visual Basic.NET Application Development

This database application can be used by doctors, hospital managers, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Doctors & Nurses: The user can browse the database to learn more about the doctors and nurses of this hospital. The user can search to learn about the daily schedule of a particular doctor (or nurse); the available doctors that are specialized in a particular area; etc.

Equipment: The user can browse the database to learn about the equipment that is available in the hospital, such as equipment utilization rate, equipment purchase date, total number of a particular equipment type, etc.

Statistics, Graphs and Data Analysis: Managers use statistics, graphs, and data analysis tools to support their decisions. The plots built in the Access Application Development section can be included in here.

Simulation model: We suggest that you build a simple model to simulate this system. The objective of this simulation model is to identify the optimal number of doctors and nurses to be hired; identify whether new equipment is needed and, if needed, how much; identify whether more beds are needed and, if needed, how many; etc. This information will be used by the management to improve the service level of the hospital.

This is a queuing problem. The data collected from the surveys could be used to identify the arrival rate of the patients in the hospital, patient waiting time, patient consulting time, equipment utilization, etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about doctors, nurses, equipment, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a doctor, a nurse, a manager, the database administrator, etc. The doctors and nurses should be able to search the database to identify their schedules. The managers need to access the database to retrieve information that will help in improving the service level. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

“Blood banks” carry an inventory of different blood types that is used by health care providers, such as hospitals and clinics. Carrying enough inventories to provide the required blood type to patients in critical, life-threatening situations is an important issue that blood banks have to deal with. A big challenge in managing the inventory is that banks cannot “order” or “get shipments” of a particular blood type at any moment in time. Blood is donated by donors, and this process is performed voluntarily, thus it requires planning. The blood bank is considering implementing a warning system that informs the staff when the inventory falls under a pre-specified level. These warnings will initiate special drives to collect the particular type of blood the bank is short of.

The blood bank has data about the blood collected by donors as well as the patients that used blood from this bank. We want to build a decision support system that would assist with managing the inventory in a blood bank.

The methodology one can use to determine the demand for a particular blood type and decide about the corresponding inventory level could be similar to the ones used by manufacturing companies. The following paragraph explains how this can be done.

The blood banks can retrieve the following information from the data kept in their database: number of patients that suffer a particular disease per age group; expected number of accidents in a holiday season per age and blood type; and expected amount of blood donated by age and blood type. This information can then be used to estimate the amount of a particular blood type that will be needed and the amount of blood that will be available during the same time period in the current year. If the bank foresees an inventory shortage, it can take different initiatives to increase this level, such as borrowing from other banks or initiating special drives to collect blood.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Blood pouch*: The main attributes are identification number, blood type, location, cholesterol level, etc.
2. *Blood bank*: The main attributes are identification number, name, address, name and telephone number of the contact person, etc.
3. *Donor*: The main attributes are social security number, name, address, gender, birthday, telephone number, e-mail, blood type, HIV positive?, etc.
4. *Patient*: The main attributes are social security number, name, address, gender, birthday, telephone number, e-mail, blood type, HIV positive?, etc.
5. *Doctor*: The main attributes are social security number, name, address, telephone number, area of specialization, etc.

6. *Operation*: The main attributes are name, description, expected amount of blood needed, etc.
7. *Health care provider*: The main attributes are name, address, name and telephone number of the contact person, etc.

Note that when a donor gives blood (a) the blood is carried in a blood pouch. These blood pouches are instances of the *Blood inventory* entity type; (b) the following information is recorded about a donation: donation date, expiration date, and quantity donated; and (c) the inventory level of the particular blood type in the blood bank is updated. Note that when blood is used (a) the following information is recorded: date and quantity used; and (b) the inventory level of the particular blood type is updated.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Use a make table query to create a table that presents information about the current inventory level of each blood type. Name this table tblInventory.
2. Create an update query that updates the inventory level of a particular blood type in the table tblInventory whenever a donation is made or whenever a patient uses blood.
3. The information about new donors (donors who donated for the first time in the current year) is kept in a separate table. At the end of the year, the information about the new donors is appended to the table that keeps the information about the rest of the donors. Do the following:
 - a. Use a make table query to create a new table (called tblNewDonors) to present the data about the new donors.
 - b. Use an append query to append, at the end of the year, the information about the new donors to the table that has the data about the existing donors.
 - c. Use a delete query to delete the information about the donors in the table tblNewDonors.
4. Create a select query that lists the identification number of all the blood pouches that are expired. Create a delete query that deletes the information about the expired blood pouches from this database. Create an update query that updates the inventory level of each blood type in the table tblInventory after the expired blood pouches are disposed.
5. When a patient needs blood of a particular type, it is suggested that the pouch with the earliest expiration date be used. Create a query that prompts the user for the blood type and returns the identification number and location of the pouch that carries the particular blood type with the earliest expiration date.
6. Historical data about blood donations and usage will help in building the following queries. These queries will help to determine the need and supply for a particular blood type.
 - a. Create a query that calculates the average amount of blood used per month in a year.

- b. Create a query that calculates the average amount of blood donated per month in a year.
 - c. Create a query that calculates the amount of excess/shortage of each blood type in the current month. To calculate these amounts, use the information from the above queries.
 - d. Create a query that calculates the average amount of blood used during the following holiday seasons: New Year, Easter, Independence Day, Labor Day, Thanksgiving, and Christmas.
7. Donors are allowed to donate up to a certain amount of blood per month. Create a query that calculates the amount of blood donated within the month by each donor.
 8. Create a query that calculates the amount of blood (of each blood type) used by health care providers.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: blood pouches, donors, patients, etc. These forms allow the user to add, update, and delete information about blood pouches, donors, patients, etc.
3. Create a form that allows the user to choose from a list box a blood type. Create a subform that presents a list of the blood pouches that carry the particular blood type. For each blood pouch, present the following: identification number, location, donation date, and expiration date. Sort the information by expiration date. Insert a command button that, when clicked-on, calculates the total amount of blood of a particular blood type in the inventory. Insert a command button that, when clicked-on, presents a list of the donors of a particular blood type.
4. Create a form that allows the user to browse through the information about each health care provider. Create a subform that presents for each provider the total amount of blood (of each blood type) consumed per month during the current year.
5. Create a form that allows the user to browse through the information about each blood bank. Create a subform that presents for each blood bank the total amount of blood (of each blood type) in the inventory.
6. Create a form that allows the user to choose the name of a donor from a combo box. Create a subform that presents a list of visits for each donor. For each visit, the following is reported: visit date, quantity donated, blood pressure level, sugar level, cholesterol level, descriptions of donor's health conditions, etc.

Design a logo for this database. The logo may contain the picture of a hospital, a blood pouch, etc. Insert this logo in the forms created above. Have the background color of the forms light yellow and the border color for the titles dark red. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

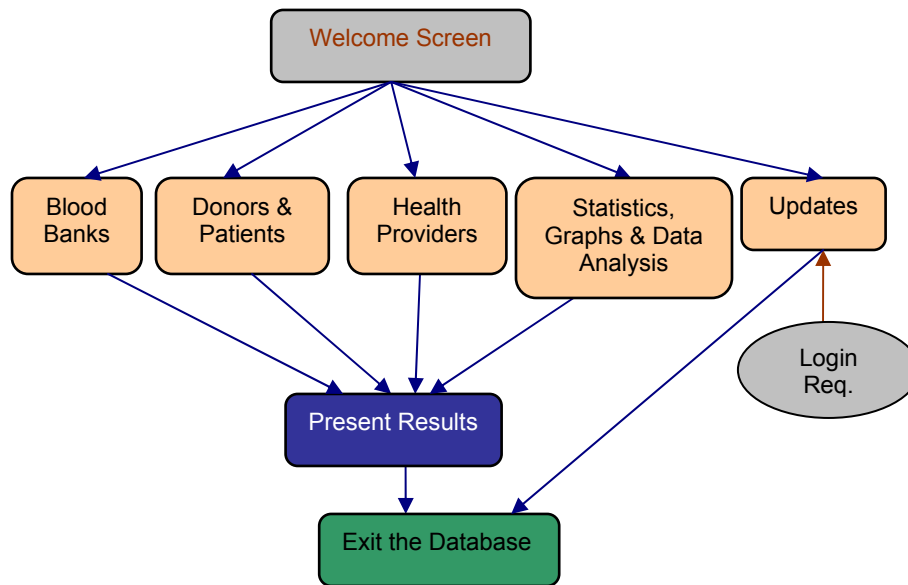
Reports

1. Blood banks provide the following services to the donors: test for HIV and test for sugar content and cholesterol. The bank mails summary reports to donors after each donation. Create a report that presents these test results for a donor.
2. Use the label wizard to create labels with the address of each donor.

3. Usually, doctors order more blood for an operation than the amount required. They follow this practice to be prepared in case of an emergency. However, the blood banks do not like this procedure, as they feel the remaining blood from the operations is misused. For each operation, the database carries information about the expected amount of blood to be used. Historical data can also be used to calculate the amount of blood used for a kind of operation for a patient of a particular age. Create a report that presents the following for a particular kind of operation: the expected amount of blood to be used, the average amount actually used, and the maximum amount and minimum amount ever used.
4. Use the chart wizard to plot the following:
 - a. Average amount of blood used by health providers per month during the current year.
 - b. Maximum amount of blood used by a health provider per month during the current year.
 - c. Minimum amount of blood used by a health provider per month during the current year.
 - d. Total amount of blood donated per month during the current year.
 - e. Total amount of blood used by health providers per month during the current year.
 - f. Total number of expired blood pouches per month in the current year.
 - g. Shortage/excess of donations versus usage of blood per month in the current year.
 - h. Total number of donors per year in the last five years.
 - i. Total number of new donors per year in the last five years.
5. List the identification number, location, and expiration date of the expired blood pouches.
6. At the end of the year, the blood bank mails to health providers a report containing detailed information about the amount of blood procured by blood type, procurement date, etc. Create such a report. Use the label wizard to prepare labels with the address of each provider.

Visual Basic.NET Application Development

This database application can be used by blood banks, health providers, doctors, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Blood Banks: This part of the database will mainly be used by health care providers to place orders and by other blood banks in case they face a shortage or excess of blood. Each blood bank presents the amount of blood they have in their inventory by blood type.

Donors & Patients: This part of the database is mainly used by the blood banks, and sometimes by patients and donors. The blood banks browse this part of the database to learn about the blood type of a particular donor, the history of a donor's donations, etc. The donors can use the database to check their health condition (test results) after the last donation, etc.

Health Providers: This part of the database is mainly used by health providers and blood banks. Blood banks check the database to identify potential providers (customers). Health providers check the database to identify sources of a particular blood type.

Statistics, Graphs, and Data Analysis: This part of the database presents graphs and different statistics that are used to identify trends in the donation and consumption of blood. The plots already created in the Access Application Development part can be used in here.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about patients, donors, blood banks, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a blood bank, a doctor, a patient, the database administrator, etc. Health providers search the database to identify sources of blood. The blood banks use the database to identify potential customers (health providers), keep in contact with patients

and donors, recruit donors, identify ways to improve their service level, etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

In manufacturing plants, managers are responsible for setting the equipment maintenance schedule. This process is challenging, especially when the managers have limited resources on hand. In building the schedule, the managers should consider the following issues: specific maintenance requirements for the equipments, availability of the maintenance employees, etc. It is important to understand that in a plant there are a number of equipments, and each one has its own required shutdown or breakdown maintenances. Shutdown maintenances are planned and are therefore easy to handle. However, breakdown maintenances are unplanned and hence disturb the maintenance schedule. On top of that, it is very difficult to analyze the behavior of equipments and predict when a breakdown will happen.

The objective of this project is to build a decision support system that will facilitate the process of building maintenance schedules in manufacturing plants. The system will enable managers to make the right decisions by doing the following: (a) keeping record of maintenance performed on different machines; (b) creating reports about the equipments that require maintenance as well as the type of maintenance required; (c) analyzing the breakdown maintenance for equipment, determining root causes, and deciding the replacement required; (d) preparing the maintenance schedule based on certain business rules; etc.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Equipment*: The main attributes are identification number, name, description, location, purchase cost, purchase date, estimated cost of a breakdown, average man-hours required for a shutdown maintenance, expected lifetime, etc.
2. *Component*: The main attributes are component number, name, purchase date, expected lifetime, inventory level, unit cost, name of the company that produces this component as well as the name and telephone of the contact person, etc.
3. *Employee*: The main attributes are social security number, name, address, department, specialty, employment date, etc.
4. *Maintenance*: The main attributes are identification number, name, type (shutdown or breakdown), description, total man-hours required, etc.
5. *Maintenance Company*: The main attributes are identification number, name, address, name and telephone number of the contact person, description of the maintenance services it provides, etc.

Note the following: (a) A maintenance company is hired in the case of a severe equipment failure that cannot be handled by the maintenance employees; (b) When maintenance is performed on equipments, the following information is recorded in the database: maintenance date and time, amount of time to repair the equipment, amount of time the equipment is down, name of the major component replaced, cost of maintenance, description of the process, etc; (c) Components are part of equipments. The company uses components to replace damaged parts of equipments. When a component is used, the following information is recorded: replacement date, man-hours used, quantity used, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. List all the maintenance work performed on particular equipments. For each maintenance work, display the following: number of hours the equipment was down, name of the employee or maintenance company involved, and major component(s) replaced. The query should prompt the user for the identification number of the equipment.
2. The table *Equipment* does not keep information about the average number of breakdowns per month. Create a new field in the table *Equipment* called "Average breakdowns." Build an update query that calculates the average number of breakdowns per month per equipment and inputs this information in the field "Average breakdowns" of the table *Equipment*.
3. List all the equipments that require high maintenance. Equipments are classified as "high maintenance equipment" if the cost of a breakdown is more than \$2,000. Also, equipments that break down more than three times a month are considered "high maintenance equipment."
4. Create a query that prompts for the social security number of a particular employee and returns a list of shutdown/breakdown maintenance services that the employee participated in. For each maintenance service, present the following: date, description of the service provided, employee skills required, etc.
5. Prepare a monthly maintenance schedule. The following queries help with this process:
 - a. For each equipment list the following: average number of breakdowns per month, the most common type of breakdown, average man-hours required repairing the "most common type of breakdown," and average man-hours required for maintenance. Sort this information in such a way that at the top of the list would be the "high maintenance equipments." These equipments have higher priority.
 - b. Create a crosstab query that presents for each employee the type of maintenance/repair that this employee is qualified for.
 - c. Create a crosstab query that presents for each maintenance company the type of maintenance/repair the company is qualified to perform.

The information from these queries will enable the managers to create the maintenance schedule. The process goes as follows: the first equipment in the list created in (5.a) is "high maintenance equipment" that needs, say, four maintenances in a month. The

managers decide to schedule maintenance for this equipment every Monday. Query (5.b) is used to identify the employee(s) who would perform the maintenance service. Once the maintenance schedule for the first equipment in the list is prepared, we proceed with the next equipment. Note that the employees who are scheduled to repair the first equipment will not be available on Mondays for as long as it takes to perform the service. The time needed can be identified using the information about “average man-hours required for maintenance” from (5.a).

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: equipment, maintenance, employee, etc. These forms allow the user to add, update, and delete information about equipment, maintenance, employees, etc.
3. Create a form that allows the user to choose the name of equipment from a combo box. Create a subform that presents a list of the shutdown maintenances performed on this equipment in the last six months. For each shutdown maintenance service, present the following: maintenance date, description of the service provided, amount of time the equipment was down, and cost of maintenance. Create another subform that presents a list of the breakdown maintenance performed on this equipment in the last six months. For each breakdown maintenance service, present the following: maintenance date, description of the service provided, amount of time the equipment was down, and cost of maintenance. Use a textbox to present the following: the total cost of shutdown maintenance services, the total cost of breakdown maintenance services, and the overall cost of maintenance in the last six months. Insert a command button that, when clicked-on, lists the maintenance company that performed most of the breakdown repairs for this equipment.
4. Create a form that allows the user to browse through the information recorded in the table Equipment. Create a subform that for the selected equipment presents a list of the components it consists of. For each component, present the following: component number, name, replacement date (if it has ever been replaced), cost, etc.
5. Create a form that allows the user to choose from a list box the name of a shutdown maintenance service. Create a subform that lists the names of the employees who are qualified to perform this service. For each employee, present the following: social security number, name, and department. In the same form include another list box that allows the user to choose the name of a breakdown maintenance service. Create a subform that lists the following information about the maintenance company that can perform the service: name of the company and name and telephone number of the contact person. Insert a command button that, when clicked-on, prompts for the social security number of an employee and returns a list of the maintenance services that this employee participated in. For each service, present the following: service date, description, skills required, etc.

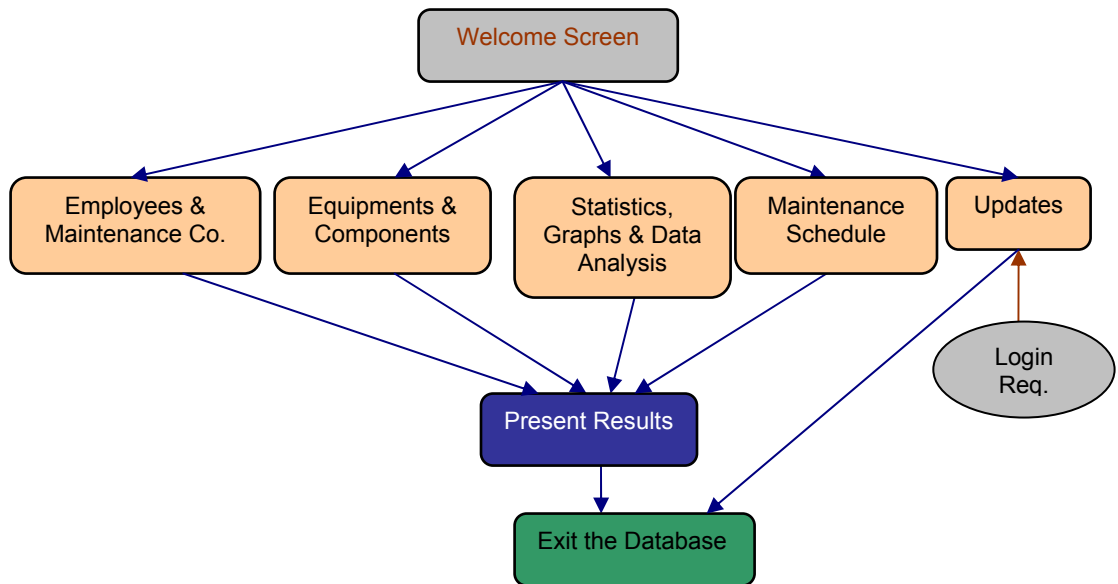
Design a logo for this database. The logo may contain the picture of an equipment, etc. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Prepare a report that presents the major components that have been used beyond their expected lifetime. For each component, present the following: component number, name, name of its supplier, and supplier's telephone number. This information is important in deciding whether the same component will be used again or if it should be replaced.
2. Create a report that prompts the user for the identification number of an equipment and returns a list of the breakdown and shutdown maintenance services performed in the last six months. For each maintenance service, provide the following information: date, service description, name of the employee (or the maintenance company) who performed the service, etc.
3. Use the chart wizard to plot the following:
 - a. The total number of shutdown maintenance services performed per month during the last twelve months.
 - b. The total number of breakdown maintenance services performed per month during the last twelve months.
 - c. The total cost for shutdown maintenance services performed per month during the last twelve months.
 - d. The total cost for breakdown maintenance services performed per month during the last twelve months.
 - e. The total number of breakdown maintenance services of "high maintenance equipments" per month during the last twelve months.
 - f. The total cost of breakdown maintenance services of "high maintenance equipments" per month during the last twelve months.
 - g. Identify the component that has the highest number of breakdowns. For this component, plot the total number of replacements per month during the last twelve months.

Visual Basic.NET Application Development

This database application can be used by managers, employees, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Employees & Maintenance Co.: Managers browse this part of the database in order to: (a) learn about the skills of a particular employee; (b) identify the employees that can perform a particular shutdown/breakdown maintenance service; (c) identify the companies that can perform a particular breakdown maintenance service; etc.

Equipments & Components: Managers and employees browse this part of the database to learn about: (a) the inventory level of a particular component; (b) the maintenance service schedule of a particular equipment; etc.

Statistics, Graphs & Data Analysis: Managers often visit this part of the database to identify trends in: (a) the number of breakdown of particular equipments; (b) the number of breakdown of a particular component; etc.

Maintenance Schedule: Managers and employees browse this part of the database to learn about the maintenance schedule. The information provided in here helps the managers to schedule the maintenance of particular equipments. Employees visit this part of the database to learn about their weekly work schedules.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about equipments, components, employees, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be an employee, a manager, a maintenance company, the database administrator, etc. The employees should be able to search the database to identify their weekly schedules. The information provided in this database should help the managers

decide on the shutdown maintenance service schedule. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

23 Spare Parts Management

Problem Description

In manufacturing plants that own a large number of equipment, managing the spare parts properly and in a timely manner is a challenging task. Usually, spare parts are categorized into two main groups: fast moving and slow moving parts. Fast moving spares are those that are very often required, and slow moving spares are those that are rarely required. The managers find it difficult to keep track of the spare parts used and determine future demand manually.

The objective of this project is to build a database management system that will help the managers with the process of managing spare parts. In particular, the system should do the following: (a) keep record of the spare parts required for a particular type of maintenance; (b) keep record of the spare parts received and used in the past; (c) schedule forthcoming major maintenance services; (d) keep record of spare parts vendors; (e) forecast future demand for fast moving spare parts based on past consumptions; etc. In this model we assume that lead-time for all spare parts is one week.

The following are other functionalities of this database: (a) generating weekly order reports for each spare part. These reports will be mailed to vendors; (b) providing weekly feedback to managers to enable decisions about expenditures on equipment maintenance; (c) providing information about economical viability of an equipment. For example, if the annual consumption of a particular spare part is more than 30 % of the book value of that equipment, it is better to purchase new equipment instead; (d) often, the slow moving items get piled up in the inventory and managers loose track of them. The system should generate reports of on-hand inventory of all spare parts and on-hand inventory of spare parts that have been in the inventory for more than three months. These reports will enable the managers to decide the proper disposal/utilization plan of the spares.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Equipment*: The main attributes are identification number, name, description, location, purchase price, purchase date, estimated cost of a breakdown, average man-hours required for a shutdown maintenance, expected lifetime, etc.
2. *Maintenance*: The main attributes are identification number, name, type (shutdown or breakdown), description, total man-hours required, etc.
3. *Vendor*: The main attributes are identification number, name, address, name and telephone number of the contact person, etc.
4. *Spare part*: The main attributes are identification number, name, type (slow or fast moving part), purchase date, expected lifetime, inventory level, unit cost, name of the

company that produces this component as well as the name and telephone of the contact person, etc.

Note the following: (a) When maintenance is performed on equipments, the following information is recorded: maintenance date and time, amount of time to repair the equipment, amount of time the equipment is down, name of the major spare part used, cost of maintenance, description of the process, etc; (b) Spare parts are part of equipments. The company uses spares to replace damaged parts of equipments. When a spare part is used, the following information is recorded: replacement date, man-hours used, number of spare parts used, etc; (c) When parts are bought from a vendor, the following information is recorded: transaction number, order releasing date, order receiving date, quantity purchased, unit price, total amount paid, payment due date, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create a query that prompts for the identification number of a spare part and returns a list of the vendors that provide the spare part. The list should contain the following information: vendor's name, address, and name and telephone number of the contact person.
2. The following queries will help the managers in deciding about the weekly purchase order for the spare parts:
 - a. The level of on-hand inventory for each slow moving spare part.
 - b. The level of on-hand inventory for each fast moving spare part.
 - c. The average weekly usage of slow moving spare parts during the last three months.
 - d. The average weekly usage of fast moving spare parts during the last three months.
3. Create a query that presents a list of spare parts used in a particular type of maintenance. In order to identify the spare parts used, one should look at the historical data collected about previous maintenances performed. This query should prompt the user for the maintenance identification number.
4. Create a query that presents the average lead-time for each vendor. In order to calculate the lead-times, one can use historical data about order release date and order receiving date for the spare parts.
5. Managers are interested in calculating the book value of the equipments. In order to calculate the book value use straight-line depreciation. The purchase date and purchase price of equipment should be used in these calculations.
6. Create a query that presents the total amount spent last year in maintenance (for example, cost of spare parts used) for the equipments owned by the company. This information, together with the book values generated previously, will help the managers to decide about replacing some of the existing equipments.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: spare parts, equipments, vendors, etc. These forms allow the user to add, update, and delete information about spare parts, equipments, vendors, etc.
3. Create a form that allows the user to browse through the information about the equipments owned by the company. Create a subform that lists the maintenance services performed on the selected equipment during the last year. Insert a textbox that presents the total cost of maintenance. Insert a command button that, when clicked-on, calculates and presents the book value of the equipment.
4. Create a form that allows the user to choose a slow moving spare part from a combo box. Create a subform that presents the following information about the selected spare: vendor's name, name and phone number of vendor's contact person, unit cost of the spare part, and the level of on-hand inventory. Insert a command button that, when clicked-on, presents last year's maintenance services that have used this particular spare part.
5. Create a form similar to the one described in part (4) for the fast moving spare parts.

Design a logo for this database. The logo may contain the picture of an equipment, a spare part, etc. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

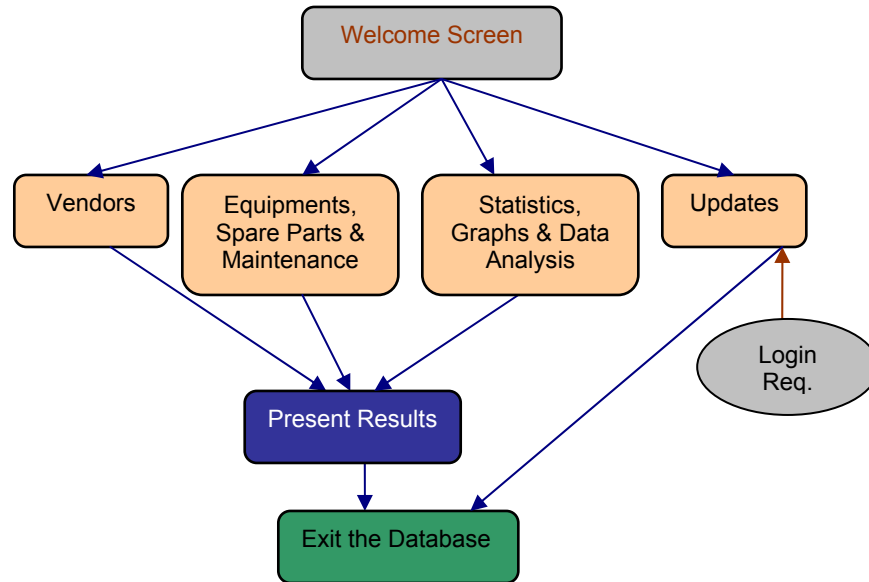
Reports:

1. Create a report with the information about the vendors of a particular spare part. The report should provide the following information: vendor's name, name and telephone number of vendor's contact person, vendor's average lead-time, and unit cost for the part.
2. Create a purchase order report. This report should include the following: transaction number, order date, list of parts ordered, and quantity ordered.
3. Use the chart wizard to plot the following:
 - a. The total on-hand inventory of the slow moving spare part per month during the last twelve months.
 - b. The total on-hand inventory of the fast moving spare part per month during the last twelve months.
 - c. The weekly usage of the slow moving spare part during the last six months.
 - d. The weekly usage of the fast moving spare part during the last six months.
4. Use the label wizard to create labels with the address of each vendor.
5. Create a report that contains the following information about the equipments owned by the company: equipment identification number, name, purchase date, purchase price, and book value.
6. Create a report that presents the following information about each spare part used by the company: identification number, name, vendor's name, purchase price, and list of maintenance services that used this part.

7. Create a report that presents a list of the spare parts needed for the forthcoming maintenance services scheduled. For each spare part, report the quantity needed, on-hand inventory, and (if needed) the number of parts to be ordered from vendors.

Visual Basic.NET Application Development

This database application can be used by managers, employees, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Vendors: Managers browse this part of the database to do the following: (a) learn about the vendors of a particular spare part; (b) compare vendors by price and lead-time; etc.

Equipments, Spare Parts & Maintenance: Managers and employees browse this part of the database to learn about: (a) the inventory level of a particular spare part; (b) the maintenance service schedule of particular equipment; (c) the spare parts needed for a particular maintenance service; etc.

Statistics, Graphs & Data Analysis: Managers often visit this part of the database to identify trends on: (a) the number of spare parts used per month during the last twelve months; (b) the average monthly on-hand inventory level of a particular spare part during the last twelve months; etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about equipments, spare parts, vendors, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be an employee, a manager, a vendor, the database administrator, etc. The information provided in this database should help the managers decide about the following: (a) the order quantity of spare parts; (b) the equipment replacement schedule; (c) the vendors to order from; etc. Vendors search the database to learn about the inventory level of particular spare parts, orders set by the company, etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

The main objective of this project is to develop a decision support system that will facilitate the process of allocating the crude oil supply from tanker ships to port and then to refinery tanks. Figure 1 presents production and distribution channels of an oil company. The arrows depict the material flow associated with the refinery process. Crude oil arrives at the port on a ship tanker. Port jetties are usually available to land the tanker. Pipelines connect the jetties to crude oil storage tanks. Oil is then pumped to the refinery crude distillation units to be distilled. At the refinery, other tanks store distilled oil. The distilled oil is finally distributed to retailers. The system that we build supports only the decisions made at the yellow boxes of Figure 1.

Production of distilled oil is highly governed by market demand. The refinery management use demand forecasts, seasonal use of oil, and market trends to prepare production plans for the next month. The following are some of the issues faced by plant managers during this process: (a) Crude oil loads of different quality are generally segregated and therefore assigned to different tanks. The reason for that is to avoid contamination. (b) Oil cannot be pumped in and out of a tank at the same time, nor is it possible to pump oil from a tanker to more than one tank at a time. These limitations influence oil processing and transferring times and, as a result, impact operation costs. (c) It is difficult to predict the exact arrival time of ships in the port, since this depends on weather conditions. Therefore, defining a detailed plan for oil allocation over a long period of time is difficult. (d) The refinery processes and operations are usually slow. For example, the process of transferring or producing a specific oil product lasts for hours if not for days. Therefore, the decision about producing a certain kind of oil can affect the system for a very long time.

Keeping these issues in mind, the main objectives of this system are as follows: (a) minimize tanker unload time; (b) avoid idle time waiting for tank availability; (c) allocate the crude oil supply to appropriate tanks.

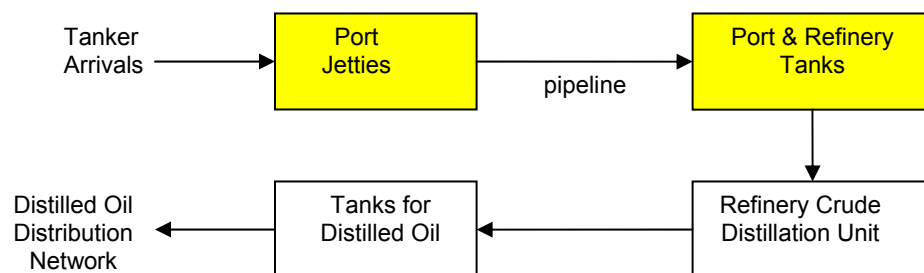


Figure 1. The structure of a generic oil-refining company.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the

primary key(s) and the foreign key(s) for each relation. Draw the relational integrity constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Oil*: The main attributes are identification number, name, description of properties, etc.
2. *Pipeline*: The main attributes are identification number, location, capacity, type, etc.
3. *Port*: The main attributes are name, location, number of jetties available, etc.
4. *Refinery*: The main attributes are identification number, name, address, name and telephone number of contact person, distance from the port, expected monthly demand, etc.
5. *Refinery Tank*: The main attributes are identification number, type, capacity, expected loading/unloading costs, expected loading/unloading time, etc.
6. *Supplier*: The main attributes are name, location, expected yearly supply, unit price, etc.
7. *Tanker*: The main attributes are identification number, capacity, etc.

Note the following: (a) Shipments sent by suppliers consist of a number of crude oil tankers. For each shipment, the shipment initiation date, shipment quantity, loading costs, loading time, and expected lead-time are recorded. (b) When a shipment arrives at a particular port, the following information is recorded: shipment arrival date, unloading costs and time, waiting time for a free jetty, waiting time for a free pipeline, demurrage costs, etc. Demurrage costs occur when a ship is kept loaded in the port, waiting for a free jetty, a free pipeline, or empty refinery tanks. (c) Crude oil is shipped from the port to a particular refinery. The following information is recorded: shipment date and quantity, amount shipped, cost of shipment, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. List the shipments that are expected to arrive at a particular port on a particular date. Prompt the user for the name of a port and a shipment arrival date. The query returns the following: tanker identification number, supplier name, quantity shipped, description of the product shipped, unit cost, shipment initiation date (the date that the shipment left the supplier), etc.
2. Create a query that prompts for the classification of oil and returns the available pipelines that can be used to pump the oil from the tanker ship to refinery tanks.
3. The following are two heuristic approaches that can be employed to choose the refinery tanks to be used for unloading a tanker:
 - a. Create a query that lists the tanks that have available capacity and are capable of carrying the particular oil type. Sort the tanks by descending order of the available capacity. The tanks that have the maximum available capacity are the ones that will be used first.
 - b. Create a query that lists the tanks that have available capacity and are capable of carrying the particular oil type. Sort the tanks by ascending order of the available capacity. The tanks that have the minimum available capacity are the ones that will be used first. This heuristic favors full utilization of tank capacity.

4. Usually, if there is not enough space to carry inventory at the refinery, the extra shipment is sold to other parties. Create a query that calculates the total space available in each refinery.
5. The following queries help predict oil demand:
 - a. Create a query that calculates the average monthly purchases of crude oil by the oil refinery during the last twelve months.
 - b. Create a query that finds the month that had the highest total purchase of crude oil during the last year.
 - c. Create a query that finds the month that had the smallest total purchase of crude oil during the last year.
6. The following queries help to predict shipment lead times:
 - a. Create a query that presents the actual travel time for each shipment received in the last six months.
 - b. Create a query that presents the estimated travel time for each shipment received in the last six months.
 - c. Create a query that presents the actual loading/unloading time for each shipment received in the last six months.
 - d. Create a query that presents the monthly average lead-time (travel time plus loading/unloading time) for shipments received in the last year.
 - e. Create a query that presents the monthly average expected lead-time for shipments received in the last year. Queries build in (6.c) and (6.d) will help to identify seasons where the deviation of the expected lead-time was bigger than the actual lead-time.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: shipments, tankers, refinery tanks, etc. These forms allow the user to add, update, and delete information about shipments, tankers, refinery tanks, etc.
3. Create a form that allows the user to choose a shipment identification number from a combo box. Create a subform that presents the following information about this shipment: shipment initiation date, expected shipment arrival date, type of products shipped, and shipment quantity. Insert a command button that, when clicked on, lists the expected lead-time and actual lead-time for all the shipments that came from the same supplier during the last year. Insert another command button that, when clicked on, reports an estimate for the unloading time. This estimated unloading time is equal to the average unloading time of the shipments of the same size.
4. Create a form that allows the user to browse through the information about the ports that the oil refinery gets shipments from. Create another subform that presents a list of shipments received during the last month from a particular port. Insert a command button that, when clicked on, reports the name and location of the port that has the best loading/unloading system. Insert a command button that, when clicked on, presents the name and location of the port with the highest number of pipelines installed.

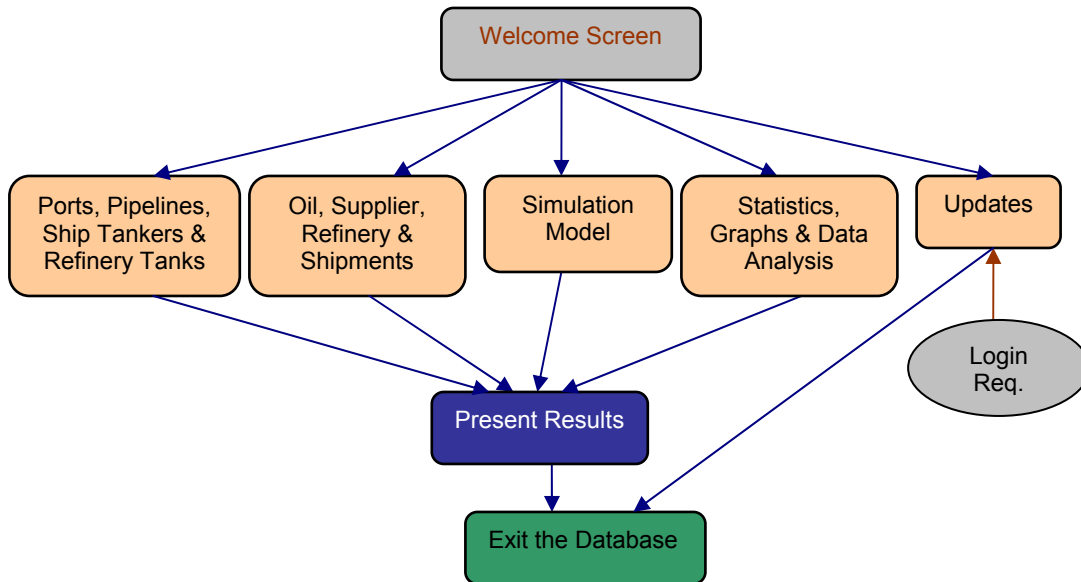
Design a logo for this database. The logo may contain the picture of an oil refinery plant, a tanker, etc. Insert this logo in the forms created above. Have the background color of the forms light yellow and the border color for the titles dark red. Include in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports:

1. Create a report of shipments that will be received one month from today. For each shipment, present the following: identification number, shipment initiation date, expected shipment arrival date, amount shipped, unit price, and total amount of payment.
2. Create a report that lists the refinery tanks that are available and ready to be used. For each tank, provide the following information: identification number, capacity, and type of oil that it carries.
3. Use the chart wizard to plot the following:
 - a. The total amount of oil that arrived in each port every month during the last twelve months.
 - b. The gap between the actual and expected lead-times for shipments that have arrived during the last month.
 - c. Average monthly unloading costs during the last year.
 - d. The total amount of oil that was sold to third parties each month during the last year.
 - e. Demurrage costs paid per month during the last year.
 - f. The total number of refinery tanks owned by the company each year during the last ten years.
 - g. The total number of free refinery tank-hours per month during the last year.
 - h. The average tanker ship waiting time for free refinery tanks per month during the last year.
 - i. The average tanker ship waiting time for free pipelines per month during the last year.
 - j. The average tanker ship waiting time for free jetties per month during the last year.

Visual Basic.NET Application Development

This database application can be used by managers, employees, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Ports, Pipelines, Ship Tankers & Refinery Tanks: This part of the database can be used to learn about: (a) ports; for example, identify ports located nearby a particular refinery, ports that have the most number of available jetties, ports that have the best loading/ unloading system, ports located near a particular supplier, etc; (b) pipelines; for example, present available pipelines on a particular port, available pipelines that can carry a particular type of oil, etc; (c) ship tankers; for example, present a list of ship tankers that can be used to transport a particular type of oil, a list of ship tankers located in a particular port, etc; (d) refinery tanks; for example, present the number of available tanks of a particular capacity, list the available tanks that can carry a particular oil type, etc.

Oil, Supplier, Refinery & Shipments: This part of the database can be used to learn more about: (a) oil; for example, special requirements about transporting a particular oil type, list of pipelines and refinery tanks that can carry a particular oil type, etc; (b) supplier; for example, list of suppliers of a particular oil type, list of ports located near a particular supplier, average order processing time for a supplier, etc; (c) refinery; for example, total number of free tanks at a particular refinery, average amount of oil processed in a month at a particular refinery, etc; (d) shipments; for example, expected arrival date of a particular shipment, list of shipments expected to come on a particular date, actual unloading time of a shipment, average waiting time for a free jetty, etc.

Simulation Model: We describe a simulation model that can be used by refinery managers to handle the process of allocating the crude oil supply from tanker ships to port and refinery tanks. This model can be used to perform a “what if” type of analysis.

Below we present the main factors that impact this allocation process:

1. Amount of crude oil arriving (at the port) on a particular day. The distribution of the daily amount of oil shipped is identified using the information about previous shipments. The amount of oil arriving on a particular day (in a simulation run) will be randomly generated from this distribution.
2. Number of refinery tanks to transport a particular type of oil. The data collected in the database can be used to identify the distribution of empty tanks of a particular type and available capacity on a particular date.
3. Number of jetties, number of pipelines, loading time and cost, unloading time and cost, etc. These data can also be derived from the information recorded in the database.

Note that this is a queuing problem. The factors described above will help you to build the model.

Statistics, Graphs & Data Analysis: This part of the database is to identify trends in the following: (a) oil supply; (b) unit price of a particular oil type; (c) loading/unloading time; (d) waiting time to get a free jetty; etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about shipments, refinery tanks, ship tankers, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a refinery manager, a supplier, the database administrator, etc. The information provided in this database should help the refinery managers to do the following: (a) minimize tanker download time; (b) avoid idle time waiting for a tank to be available; (c) allocate the crude oil supply to appropriate tanks to minimize the amount of unaccepted crude oil. Suppliers search the database to learn about shipment status. As it is explained in this project, the cost of keeping the ship idle at the port is high, and on top of that, unpredictable weather conditions impact shipping time. These factors concern suppliers, and as a result, the suppliers need to stay well informed about shipment status. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Paolucci, M., Sacile, R., Boccalatte, A., "Allocating crude oil supply to port and refinery tanks: a simulation-based decision support system.", *Decision Support Systems* **33**: 39-54, 2002.

Problem Description

Job scheduling is a task that managers have to deal with every day. We want to build a decision support system that will enable the managers to make the proper assignment of jobs to employees in a timely manner. Every morning, managers get a list of jobs to be completed and a list of employees available. Ideally, they would want to assign employees to the best job each can perform. However, there are a few problems with that: (a) There might be more than one employee qualified to perform a particular job, and there might be no one qualified to perform another job in the list; (b) The assignment should be such that it complies with union-management agreements, disability act, and ergonomic restrictions; etc.

The decision support system is based on a database that keeps information about jobs, employees, tasks that compose a job, skills required to perform a job, etc. We present a heuristic that can be employed during the process of assigning employees to jobs.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Employee*: The main attributes are social security number, name, address, department, skills, degree, availability, etc.
2. *Job*: The main attributes are identification number, description, expected length (in man-hours), tasks (it is composed of), etc.

Note the following: (a) The attribute *skills* of employee entity type is a multi-value attribute; (b) The attribute *tasks* of employee entity type is a multi-value attribute; (c) Each task requires specific skills; (d) When an employee is assigned to perform a particular job, the following information is recorded: assignment identification number, starting time and date, expected finish time and date, actual finish time and date, descriptions, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create a query that prompts the user for the identification number of a particular job and lists the tasks that compose this job.
2. Create a query that prompts the user for the name of a task and lists the skills required to perform this task.
3. Create a query that prompts the user for the name of a skill and lists the employees who have this skill.

4. Employees have more than one skill. However, they have a stronger background in some disciplines than in others. Employees rank their skills on a scale of 1 to 10 (with 1 being the best). We want to add a new field in the table Employee-Skills, called "Rank." This field stores the ranking of employees' skills. Create an update query to populate this field.
5. Create a query that prompts for a date and the name of a job and lists the available employees who can perform the job. Sort this information in such a way that at the top of the list will be the employees who have the strongest background to perform the job.
6. Create a query that presents the following: starting time and date, expected finish time and date, actual finish time and date, expected duration, and actual duration of the tasks completed during the last month.
7. Create a query that prompts for the name of an employee and lists the jobs that employee worked on during the last month.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: jobs, employees, assignments of employees to jobs, etc. These forms allow the user to add, update, and delete information about jobs, employees, assignments of employees to jobs, etc.
3. Create a form that allows the user to browse through the information stored in the table *Job*. Create a subform that presents a list of the tasks that compose the selected job. Create another subform that lists the skills required to perform the selected job. Insert a command button that, when clicked-on, lists the names and departments of the available employees qualified to perform the selected job.
4. Create a form that allows the user to select the name of an employee from a combo box. Create a subform that lists the jobs accomplished by this employee during the current month. For each job, present the following: starting time and date, expected finish time and date, and actual finish time and date. Insert a command button that, when clicked-on, presents the total number of working hours that the selected employee accumulated during the last month.
5. Create a form that allows the user to choose an assignment (of a job to an employee) from a list box. Create a subform that lists the skills of the employee involved in this assignment. Insert a command button that, when clicked-on, lists the names of all the employees who have been working on the same job. Create a text box that presents today's date.

Design a logo for this database. The logo may contain the picture of an equipment, etc. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports:

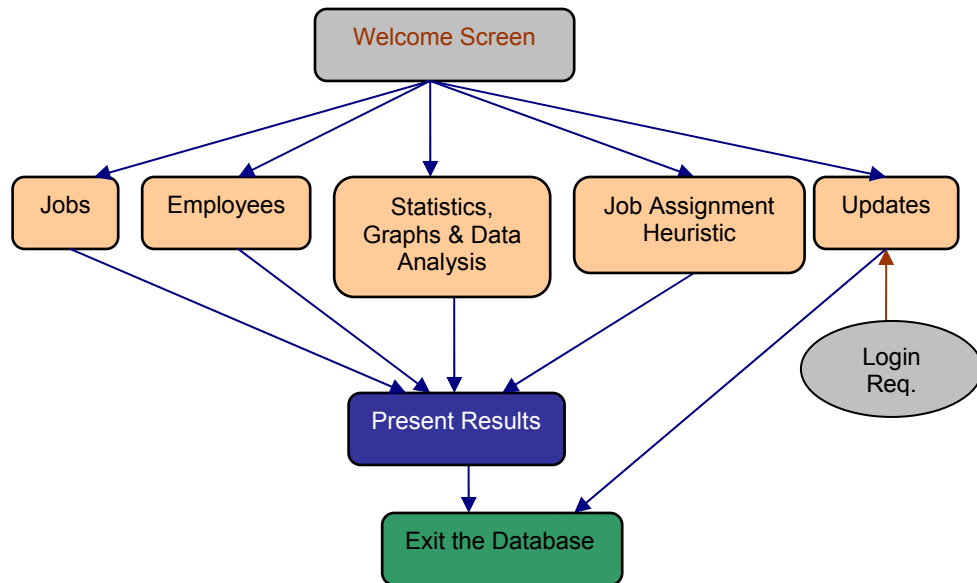
1. Create a report that lists the name of a job and the names of the employees assigned to perform the job. The report should include only the jobs that are currently proceeding.
2. Create a report that presents the name of the jobs accomplished during the last month. For each job, present the following: the starting time and date, expected finish time and

date, actual finish time and date, expected duration, and actual duration. Present the total amount of man-hours used during the last month.

3. Create a report that prompts for a particular skill and lists the name and department of the employees who have this skill. For all employees, indicate their availability and proficiency (rank) level.
4. The same jobs are performed more than once in a year. The management is interested to identify any trends in the actual amount of man-hours used to perform a particular job. Use the chart wizard to present the actual man-hours used to perform a job during the current year.
5. For all the jobs that are not scheduled yet, report (if applicable) related union-management agreements, disability act agreements, and ergonomic restrictions.
6. Report for each employee the name of the jobs that the employee is qualified for.
7. Report for each job the name of the employees who are qualified to perform the job. Present also for each job the number of employees qualified to perform the particular job.

Visual Basic.NET Application Development

This database application can be used by managers, employees, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Jobs: Managers browse this part of the database to identify: (a) the tasks that compose a particular job; (b) the skills needed to perform a particular job; (c) the employees who have the qualifications to perform a particular job; etc.

Employees: Managers and employees browse this part of the database to identify: (a) the particular skills of an employee; (b) the jobs the employee is participating in; etc. Employees visit this part of the database to learn about their assignments.

Statistics, Graphs & Data Analysis: Managers often visit this part of the database to identify trends in the following: (a) the number of man-hours needed to perform a particular job; (b) the number jobs completed in a particular time period; etc.

Job Assignment Heuristic: We propose a heuristic approach that helps with the assignment of employees to jobs. This procedure consists of two stages. In the first stage, employees are screened to identify the ones who are qualified to perform a particular job. In the second stage, certain rules are employed in assigning jobs to the qualified employees.

In first stage, the following are the screening criteria: skills, ergonomic constraints, attendance, and availability. Based on the complexity of the project, one can include additional criteria such as: (a) the number of days passed before we can re-assign an employee to the same job; (b) the maximum number of days an employee is allowed to work on a particular job; etc. The screening of employees can easily be completed by running some of the queries described above. For all employees, count the number of jobs each is qualified for. Then, rank the employees in ascending order of the number of jobs they can perform. As a result, the least flexible employees will be at the top of the list.

In the second stage, (a) sort the employees based on the number of jobs they are qualified for; (b) sort the jobs by ascending order of the ratio between the number of qualified employees who are available and the total number of employees who are required to complete the job.

The heuristic proceeds with assigning the highest-ranked jobs to the highest-ranked employees. If there are not enough employees to perform a particular job, the job should be marked for manual scheduling. If not all the jobs are scheduled, the system should repeat the earlier screening activity and revise the rankings to reflect both employees' standard time assignments and the removal of the previous job. This process continues until all the jobs have been scheduled.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about employees, jobs, job-employee assignment, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be an employee, a manager, the database administrator, etc. The employees search the database to identify their weekly schedules. The information provided in this database should help the managers decide on the assignment of employees to jobs. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Schniederjans, M. J., Carpenter, D. A., "Heuristic job scheduling decision support system - A case study.", *Decision Support Systems* **18**: 159-166, 1996.

Problem Description

The objective of this project is to build a decision support system that will assist the management of a manufacturing or service company in the process of identifying (a) training needs for employees; (b) job positions; and (c) employees suitable for a job position. This system is relevant to companies where skill requirements for a job change frequently and, as a result, employee training is needed. We provide a heuristic approach to the issues discussed above.

Suppose that there is a list of different skills needed to do the jobs performed in the company. The skills are grouped into four main categories, namely: (a) functional skills that consider knowledge of the processes involved; (b) managerial skills required to handle and organize employees; (c) methodological skills required in organizing employees' work according to some specific methodologies; and (d) technical skills that provide the technical expertise in performing the jobs. In the process of identifying the skills required to perform a job, skills are ranked from 0 to 6 according to their significance in performing the job. The skills not required by the job get 0. The degree of proficiency of employees in performing a job is marked on the same scale. The expected level of skills gained from training is assumed to be fixed.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Employee*: The main attributes are social security number, name, address, skills, department, etc.
2. *Job*: The main attributes are identification number, name, description, tasks, openings, opening date, etc.
3. *Training*: The main attributes are identification number, name, descriptions, skills gained, etc.

Note the following: (a) *Skills* is a multi-value attribute of the employee entity type; (b) When an employee is assigned to a job, the following information is recorded: assignment identification number, start date, expected finish date, actual finish date, etc; (c) When an employee gets into a training program, the following is recorded: identification number, start date, finish date, skills gained, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create a query that lists all the jobs that have an opening. For each job, present the name and the date the opening is posted.
2. Create a query that prompts the user for the name of a job and returns the skills required to perform the job. Sort the skills based on the impact they have in performing the job.
3. Create a query that prompts for the name of a training program and returns a list of the skills that are expected to be gained.
4. List the jobs that are currently in process.
5. List the names of the employees who are currently involved in a training program.
6. List the employees that have the following skills needed to perform a particular job:
 - a. Functional skills
 - b. Managerial skills
 - c. Methodological skills
 - d. Technical skills

Sort the information based on the employees' proficiency in a particular skill.
7. Create a query that prompts the user for the name of a skill and returns a list of the training programs that can help the employees to develop the particular skill.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: employee, job, training, etc. These forms allow the user to add, update, and delete information about jobs, employees, training programs, job postings, etc.
3. Create a form that allows the user to browse through the table that has the information about the jobs. Create a subform that presents a list of the skills require to perform the selected job. Insert a command button that, when clicked-on, presents the openings (if any) for the selected job. Insert a command button that, when clicked-on, prompts the user for the name of a skill and returns a list of the training programs that enable the employees to develop the particular skill.
4. Create a form that allows the user to choose the name of an employee from a combo box. Create a subform that presents a list of jobs that the employee has performed so far. For each job, present the starting date, expected finish date, and actual finish date. Insert a command button that, when clicked-on, presents a list of jobs that the employee is currently working on.
5. Create a form that allows the user to choose the name of a training program from a list box. Create a subform that presents a list of the employees who have been in this training program. For each employee, present the starting and ending dates of the training program. Include a command button that, when clicked-on, counts the total number of training programs offered in the current year.

Design a logo for this database. The logo may contain the picture of equipment, a manufacturing plant, etc. Insert this logo in the forms created above. Have the background color of the forms light yellow and the border color for the titles dark blue. Include in the

forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports:

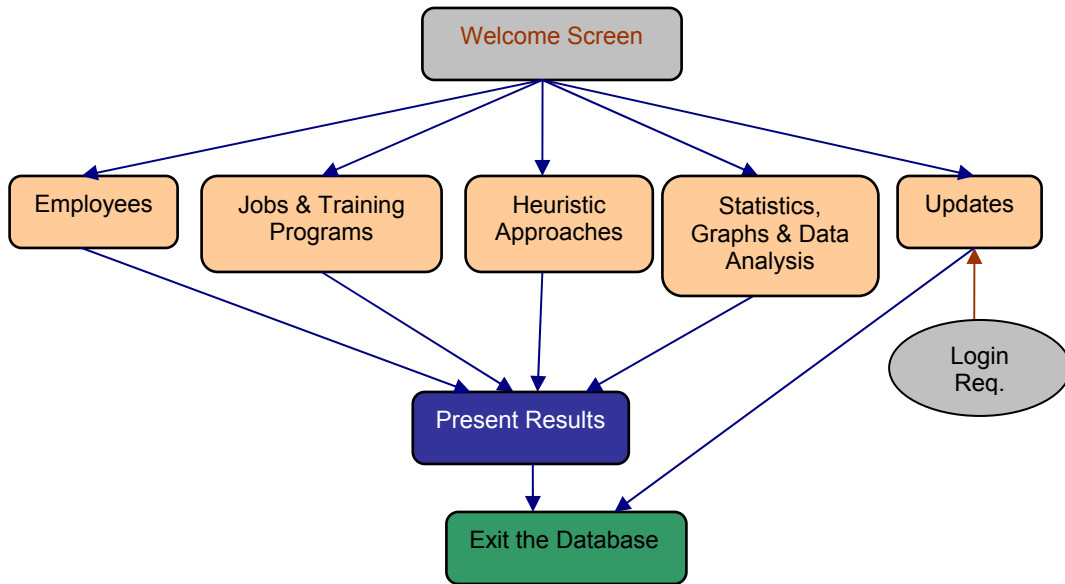
1. Report the names of the current job openings. For each opening, present the corresponding posting date and a list of skills required.
2. Report the following for each job that is currently proceeding: starting date, expected finishing date, and a list of employees involved.
3. Report the following for each job that is in the process of being scheduled: expected starting date, expected finishing date, and a list of skills required.
4. Report the name of the training programs developed in the current year. For each program, present the following: starting date, finish date, and a list of skills acquired.
5. Report for each of the training programs developed in the current year the name of the employees who participated.
6. Use the chart wizard to plot the following:
 - a. The total number of employees hired every three months in the last five years.
 - b. The total number of training programs offered every three months in the last five years.
 - c. The same job is performed more than one time. The managers are interested to identify any trends in the amount of time used to accomplish a particular job. Usually, because of the learning curve, the time it takes to perform the same job gets shorter. Plot the number of days used to perform the same job during the last two years.

Visual Basic.NET Application Development

This database application can be used by managers, employees, etc. In the following figure we present a tentative layout of the system. In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Employees: This part of the database can be used by the managers to identify: (a) the employees who have a particular skill; (b) the employees who are assigned to a particular job; etc. The employees use the database to check their job assignments.

Jobs & Training Programs: This part of the database is used by the managers to identify: (a) the current job openings; (b) the training programs that are currently going on;



(c) the skills required to perform a particular job; (d) the skills required for an open position; (d) the skills that will be gained from a training program; etc. The employees check this part of the database to identify: (a) job openings; (b) adequate training programs; etc.

Heuristic Approaches: Below we describe two major problems faced by the managers and provide heuristics to approach these problems.

Problem 1: It is important to identify the training needs of an employee. The training program should be designed to enhance the skills needed to carry out a job in the most efficient way. This problem can be defined as follows:

- ❖ Given an employee hr_i ($i = 1, \dots, I$) that has been assigned to job jp_k ($k = 1, \dots, K$)
- ❖ Determine the training programs (TPs) that hr_i should attend in order to reach the skill level required for jp_k on the basis of following objectives:
 - hr_i should attend the minimum number of TPs
 - skill levels for hr_i , after having attended the selected TPs, should be as close as possible to the minimum skill levels required for jp_k
 - TPs should be chosen from an a priori limited set.

Heuristic Approach:

- ❖ For any skill s such that jp_{ks} (required level of skill s for job jp_k) is positive, a generic training program tp_m is initially included in T_s (training programs to develop skill s) if $tp_{ms}^- < jp_{ks}$ and $jp_{ks} \leq tp_{ms} \leq jp_{ks} + \Delta t_s$, where Δt_s is threshold fixed by the user, tp_{ms}^- is the minimum level of skill s required to attend training, and tp_{ms} is the expected level of skill s after training.
- ❖ Keep the first n (a parameter set by the user) training programs that minimize the gap between the job requirement and skills gained from training:
- ❖ $d_i(jp_k, tp_m) = \sum_{s \in S'} (jp_{ks} - tp_{ms})^2$,
- ❖ where, $tp_m \in T_s$, S' is the set of skill requiring a level higher than zero for the position jp_k .

- ❖ Append to T_s the training programs if the current skill level of hr_i is compatible to the minimum requirements for that training program.
- ❖ Eliminate all the training programs from T_s if the current skill level of hr_i is incompatible with the minimum requirements for that training program.
- ❖ Implicitly enumerate n training programs in T_s that cover all the skill levels required by job jp_k that employee hr_i should take.

Problem 2: The assignment of employees to jobs is not an easy task. Flowing is a very helpful heuristic approach. The problem of assigning employees to jobs can be defined as follows:

- ❖ Given an employee (say hr_i) and a set R of jobs, determine the jobs in R that best suit the hr_i skills on the basis of the following objectives:
 - The skill level required for the selected job should be very close to the skills of hr_i and vice-versa.
 - The selected jobs should require hr_i to increase the minimum number of skills.

Heuristic Approach:

- ❖ Select first n (a parameter set by user) jobs that minimize the gap between the skills required by job jp_r and skills owned by employee hr_i :
- ❖ $d_2(hr_i, jp_r) = \sum_{s \in S_r} (jp_{rs} - hr_{is})^2$,
- ❖ where S_r is the set of all skills needed to complete job r .
- ❖ Sort the jobs presented above in descending order of d_2 .
- ❖ Display the sorted list to the user with additional information of employee hr_i .

Statistics, Graphs & Data Analysis: This part of the database is to identify trends in: (a) the number of job openings; (b) the number of training programs offered; (c) the number of hours needed to perform a job; etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about employees, jobs, job openings, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a refinery manager, a supplier, the database administrator, etc. The information provided in this database should help the managers to identify (a) needs for new hiring; (b) needs for special training programs; etc. Employees search the database to learn about training opportunities, weekly schedule, etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Bellone, M., Merlino, M., Pesenti, R., "ISPM: A DSS for personnel career management." *Decision Support Systems* **15**: 219-227, 1995.

Problem Description

Decision support systems are all about helping managers making a decision that is heavily supported by the data collected, past experiences, etc. Often, decision support systems are based on more than one database. This allows for integration of more data, and it results in a better decision-making process. Integrating different databases, however, is not an easy task. For example, although very critical, there are cases where the databases at clinics and insurance companies are not integrated. This lack of integration has a twofold effect: (a) insurance companies find it difficult to design or negotiate health care programs; and (b) practitioners are unable to take advantage of or participate in designing insurance programs.

The aim of this project is to design a decision support system that will integrate the data stored in different insurance companies and health care providers. The objective is to provide supporting information to the participants to make the right decision about insurance programs.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Doctor*: The main attributes are identification number, name, address, area of specialty, major degree, etc.
2. *Disease*: The main attributes are identification number, name, expected cost and time needed to recover a patient from this disease, treatments that can be used, medicines that can be used to cure a patient, etc.
3. *Healthcare Provider*: The main attributes are identification number, name, address, name and telephone number of the contact person, etc.
4. *Healthcare Plan*: The main attributes are identification number, name, description, etc.
5. *Insurance Company*: The main attributes are identification number, name, address, name and telephone number of the contact person, etc.
6. *Patient*: The main attributes are social security number, name, address, gender, birthday, employer's name and telephone number, description of patient's health conditions, etc.

Note the following: (a) Patients buy health plans offered by the insurance companies. The following information is recorded when a patient purchases a health plan: transaction identification number, date, amount charged, payment date, amount paid, etc. This integrated database allows patients to look over a number of health plans before choosing the one that best fits their needs; (b) Physicians coordinate the patients' health care programs with investigation agencies and specialists. Physicians process the claims about the service provided to patients with the insurance companies; (c) Healthcare providers work with insurance companies to decide about the level of service provided, amount of money

charged to patients, and the distribution of this amount between the health care provider and insurance company; (d) When patients visit the health care provider, the following information is recorded in the database: visit identification number, date, and description of patient's symptoms (e) Health care providers are hospitals, general practitioners and specialists, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The insurance companies and health care providers are interested in getting feedback from the patients about the service level. The table *Visit* records information about patients' visits. Add a new field in this table called "Patients Comments." After each visit, patients are asked to grade on a scale from 1 to 10 (with 10 being very satisfied) the quality of the service received.
 - a. Create a query that prompts for the name of a provider and lists the average number of points accumulated by each of its doctors and the corresponding standard deviation.
 - b. List the doctors who got excellent evaluations from the patients most of the time.
 - c. Create a query that prompts for an area of specialty and returns the best doctors who are currently working on that particular area.
 - d. Create a query that presents the average number of points accumulated by each provider.
2. Insurance companies deal with employers instead of individuals, since individuals usually buy the health care plan through the company they work for. Employers are interested to know about the performance of the health care provider, patient satisfaction, physicians' competence and experience, etc. The following queries provide the means to measure quality:
 - a. For each health care provider, present the number of physicians they have.
 - b. For each health care provider and each doctor, present the total number of patients they serve.
 - c. For each health care plan, present the number of patients enrolled.
3. The decision to participate in a healthcare plan is as important to a provider as it is to an insurance company. This decision requires a deep financial analysis of the plan by both parties. The system should provide the necessary cost information to help the participants in the decision-making process. Create a query that prompts for the name of a plan and presents the costs incurred and the fees paid by the insurance company to all the providers included in this database.
4. One of the main functionalities of this system is to help the insurance companies to decide about the terms of a healthcare plan. For example, surveys have shown that there is little benefit in keeping a patient in an alcoholic rehabilitation program for more than one month. This information is valuable to insurance companies that can now justify providing healthcare for only one month to alcoholics. In similar terms, if the recovery rate from a disease is very slow, insurance companies can choose not to cover this disease in the plans they offer or charge higher rates to patients. Create a

query that prompts for the identification number of a disease and returns the expected time and cost of recovery.

5. Keeping a central database of the health care providers helps in identifying doctors who have the background to treat a particular disease. Usually, patients visit their health care provider in case of a health problem. A doctor visits the patient, and based on the patient's symptoms, the doctor recommends a specialist for further investigation. Sharing the information about doctors will help in identifying the doctor who has the required skills and is available to meet the patient. Create a query that prompts for an area of specialty and a desired appointment time and returns the name, address, and telephone number of the doctors who are available at that time.
6. It is observed that practitioners often use different kinds of treatments and medications to cure patients. This system can help to standardize the use of treatments and medications for a particular disease. Create a query that prompts for the name of a disease and lists the medications and treatments that can be used to cure this disease.
7. In the process of clearing claims from the practitioner and health care providers, the medical directors of insurance companies need to have a better understanding of the required and unrequired treatments and medications provided to the policy holder. Create a query that prompts for the name of a disease and lists the medications and treatments that should be used to cure this disease. This query is similar to the one in part (6).
8. The system should help insurance companies in identifying whether the intended mix of patients are taken care at a particular healthcare provider. Create the following queries:
 - a. This query prompts for the name of a healthcare provider and returns the number of patients who are treated for a particular disease.
 - b. This query prompts for the name of a healthcare provider and returns the number of female and male patients treated.
 - c. This query prompts for the name of a healthcare provider and returns the number of patients of a particular age group who are treated.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: insurance company, healthcare provider, healthcare plan, etc. These forms allow the user to add, update, and delete information about insurance companies, healthcare providers, healthcare plans, etc.
3. Create a form that allows the user to browse through the information about healthcare providers. Create a subform that for each provider presents a list of insurance companies they are affiliated with. For each insurance company, present the following information: name, address, and name and telephone number of the contact person. Insert a command button that, when clicked-on, returns the name and area of specialty for the doctors working for the selected healthcare provider.
4. Create a search form. This form includes a number of command buttons that, when clicked-on, displays the following information:
 - a. A list of doctors who are specialized in curing a particular disease.
 - b. A list of patients who suffer from a particular disease.
 - c. A list of treatments and medications that cure a particular disease, etc.

5. Create a form that allows the user to choose the name of a doctor from a combo box. Create a subform that presents a list of the patients the selected doctor is treating. For each patient, present the following: patient identification number, name, name of the disease the patient is treated for, name of the medications the patient is taking. Insert a textbox that presents the name of the healthcare provider the doctor is working for.
6. Create a form that allows the user to choose a patient identification number from a list box. Create a subform that presents a history of patients' visits to doctors and details about treatments and medications provided. Include a textbox that presents the total amount of money claimed to the insurance companies by the healthcare providers for the selected patient. Include another textbox that presents the amount of money cleared by the insurance companies for the selected patient.

Design a logo for this database. The logo may contain the picture of a medication, hospital, etc. Insert this logo in the forms created above. Have the background color of the forms light yellow and the border color for the titles green. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

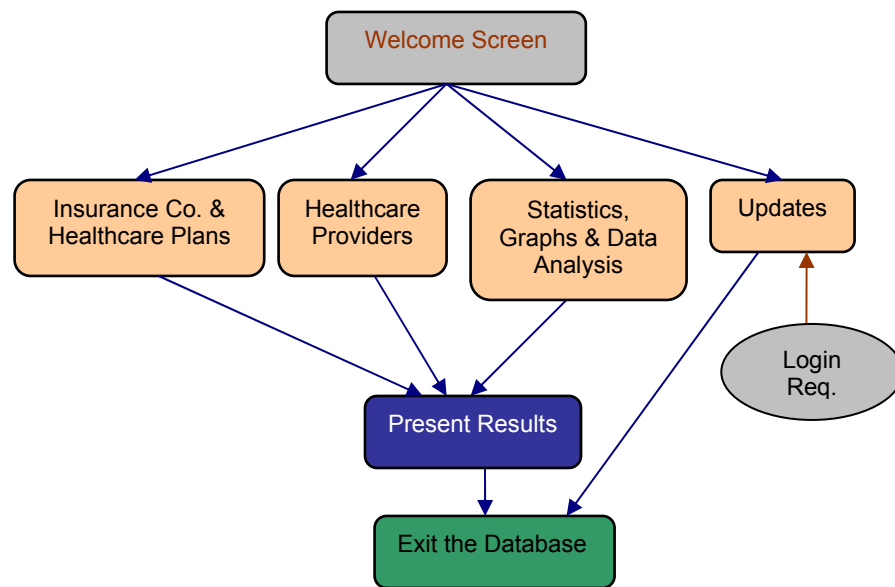
Reports:

1. Create a report that presents the following information about each insurance company: company's name, total number of members (patients), total amount of money from members in the current year, total amount of money paid to healthcare providers in the current year, and profit made. Sort the information by profit.
2. Create a detailed report about the patients with high care expenses (expenses higher than \$10,000/year). The report should include: patient identification number, name, name of the disease the patient is being treated for, name of the corresponding provider, and total amount of money paid by the insurance company in the current year. This report should also present the amount of money the insurance company has paid in the current year for all the "high care cost" patients.
3. Create a detailed report about the healthcare providers in this database. The report should include the following: provider's name, address, total number of patients, total number of claims filed in the current year, and total amount of money cleared from insurance companies in the current year.
4. Create a detailed report about the doctors. The report should include the following: doctor's name, address, telephone number, area of specialty, name of the provider the doctor is working for, total number of patients the doctor is treating, and average performance score from patients' evaluations.
5. Create a detailed report about the healthcare plans provided. The report should include the following: plan's identification number, name, description, total number of referrals, etc.
6. Use the chart wizard to plot the following:
 - a. For each insurance company plot the total number of referrals per year during the last ten years.
 - b. For each insurance company plot the total amount of money from referrals per year during the last ten years.
 - c. For each healthcare provider plot the total number of patients per year during the last ten years.

- d. For each healthcare provider plot the total amount of money from patients per year during the last ten years.
 - e. For each healthcare provider plot the total number of claims per year during the last ten years.
 - f. For each healthcare provider plot the total number of female/male patients per year during the last ten years.
7. Use the label wizard to create labels with the addresses of the insurance companies, healthcare providers, doctors, and patients.

Visual Basic.NET Application Development

This database application can be used by healthcare providers, insurance companies, doctors, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Insurance Co. & Healthcare Plans: Insurance companies, healthcare providers, doctors, and patients browse this part of the database to learn about the insurance companies and the plans they offer. Patients want to learn about plans that best fit their needs. Insurance companies want to learn about plans offered by other companies. Doctors analyze these plans to identify the ones that best fit their patients' needs.

Healthcare Providers: Insurance companies, patients, and doctors browse this part of the database to learn about the following: doctors who are working with a provider, type of service provided, patients' satisfaction from the service, etc.

Statistics, Graphs & Data Analysis: This part of the database is investigated in order to identify trends in: (a) the number of patients using a particular healthcare plan; (b) the

number of members (patients) of a particular insurance company; (c) the number of patients visiting a particular healthcare provider; etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about healthcare providers, insurance companies, doctors, patients, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be an insurance company, a healthcare provider, a doctor, a patient, the database administrator, etc. Insurance companies use the database to process claims from healthcare providers, identify successful healthcare plans, identify healthcare providers, etc. The information provided in this database should help healthcare providers and doctors to identify reliable insurance companies, process claims with insurance companies, identify treatments and medicines to treat a particular patient, identify doctors who have a specific area of specialty, etc. Patients visit the database to learn about insurance policies, reliable doctors, etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Dutta, A., Heda, S., "Information systems architecture to support managed care business processes." *Decision Support Systems* **30**: 217-225, 2000.

Problem Description

Local governments usually organize tenders where local companies bid for large scale projects supported and financed by the government. Tenders involve large amounts of money. Since the government supports the projects, on one side the companies find it very prestigious to be part of it, and on the other side, the public is very sensitive about how well the money is used. A multi-disciplinary committee is constituted in order to evaluate the participants. The evaluation process consists of two phases: first is the pre-qualification phase where tenders are scrutinized based on their legal and technical system, and second is the final phase where tenders are evaluated based on a costs/performance analysis.

In the first phase, participants submit general information about the company, their legal and technical system, number of employees, etc. The companies are evaluated based on the following criteria: *ISO 9000* certification, 6-sigma process control, etc. In the second phase, participants submit information on prices and product quality. The companies are then evaluated based on the following criteria: price, product quality, and technical competence.

The aim of this project is to develop a multi-level, multi-objective decision support system that will assist members of the committee in evaluating the participants and making a final recommendation about the company that will carry on the project.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Participant*: The main attributes are identification number, name, address, name and telephone number of the contact person, number of employees, compliance with *ISO 9000*, compliance with 6-sigma process control, etc.
2. *Project*: The main attributes are project identification number, name, description, requirements, etc.
3. *Project Evaluation Committee*: The main attributes are social security number, name, address, telephone number, degree, area of specialty, etc.
4. *Evaluation Criteria*: The main attributes are identification number, name, and weight.

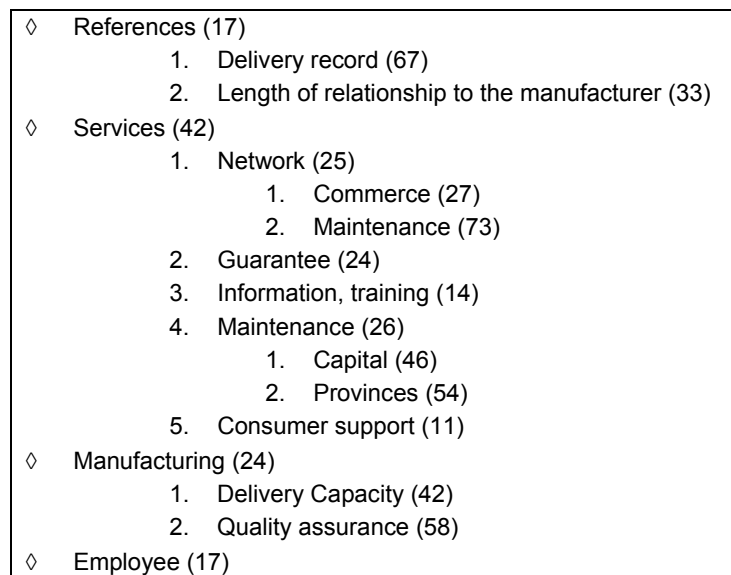
Note that the evaluation criteria are used to identify the competency of a participant to undertake a project. The data collected from the two phases of the evaluation process is recorded in this database. When a participant wins the tender, the following information is recorded: estimated starting date, estimated finish date, actual starting date, actual finish date, estimated costs, actual cost, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. List the name, address, and telephone number of the companies that comply with ISO 9000 and 6-sigma process control.
2. A number of criteria are used to identify the companies that have adequate technical capabilities to carry on the project. Each criterion has its own weight. The figure below presents the decision tree considered during this phase. The numbers in brackets are the weights given to each criterion. The members of the project evaluation committee evaluate each company on a scale of 1-100 (consider only leaf criteria). Create a query that calculates the total score for each company. Sort this information by total score.



3. In the second phase, companies are evaluated based on a cost/performance analysis. Suppose that the members of the project evaluation committee give the following weights to the evaluation criteria of this phase: weight of price is 50, weight of product quality is 30, and weight of technical competence is 20. Then, the companies are evaluated on a scale of 1-100 points. As price is a quantitative measure but quality and technical competence are subjective measures, a price equivalent is calculated using the following procedure:
 - a. Calculate the fractional weighted sum of scores for product quality and technical competence. For example, for each tender calculate

$$p = \text{score} * \text{percentage weight}/100.$$
 - b. Calculate the harmonic mean and quadratic mean of the scores accumulated. Group together tenders with a score less than the harmonic mean; and group together tenders with a score more than the quadratic mean.
 - c. Calculate the geometric mean of the score (p_i) and price offer (C_i) for the tenders in the first group and similarly p_2 and C_2 for the tenders in the second group.

- d. Calculate benchmark price:

$$N = (C_2 - C_1) / (p_2 - p_1).$$

- e. Finally calculate the adjusted price for each tender as follows:

$$C^* = C + (1-p)N,$$

if weight of criteria price is 50, or

$$C^* = C + 2/3(1-p)N,$$

if the weight is 60.

Create queries that would enable the user to complete the second evaluation phase.

4. Create a crosstab query that presents for each company participating in the tender the points earned from each member of the project evaluation committee.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: companies, projects, project evaluation committee, etc. These forms allow the user to add, update, and delete information about companies, projects, the project evaluation committee, etc. Note that when creating data entry forms to be used by companies to submit tenders, you must assure that the proper validation rules are implied in order to minimize data entry errors. Create two such forms, one for the pre-qualification phase and another for the final (second) phase.
3. Create a form that allows the user to choose the name of a company from a combo box. Create a subform that presents the following information about the selected company: the points received from each member of the project evaluation committee on all the evaluation criteria during the first phase. Create another subform that presents the points received from each member of the project evaluation committee on all the evaluation criteria during the second evaluation phase. Insert three textboxes to display: (a) the total score from the first evaluation phase, (b) the total score from the second evaluation phase, and (c) the name of the company that wins the tender.
4. Create a form that allows the user to choose from a list box the name of a project. Create a subform that provides detailed information about the participants in the tender. Insert a textbox that displays the name of the winner.

Design a logo for this database. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

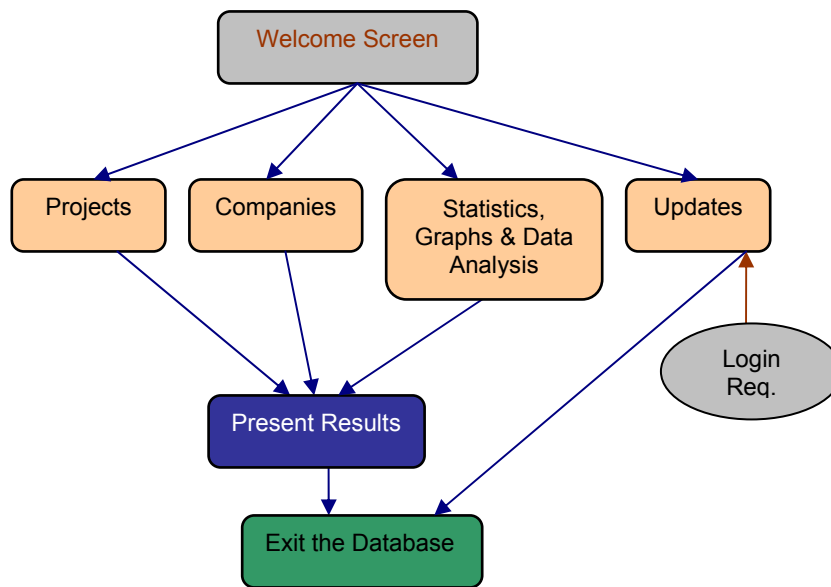
Reports

1. Create a report that presents a table that has the name of the criteria that will be used for evaluation as column headings and the names of the companies participating in the tender as row headings. These reports will be submitted to the members of the project evaluation committee to enter the scores during the evaluation process.
2. Create reports (letters) that will be sent to companies whose offer was rejected. The report should also present the reason(s) for rejection.

3. Prepare a summary report that contains the following information: number of tenders considered in the first and second phases of the evaluation process, a list of criteria that was considered, recommendations, etc.
4. Create a summary report that consists of the following information about the participants in the tender: (a) name, (b) total, average, and standard deviation of the scores received in each evaluation criterion, and (c) total, average, and standard deviation of the overall score (overall criteria).
5. Use the label wizard to create labels with the address of each participating company.

Visual Basic.NET Application Development

This database application can be used by companies, project evaluation committee members, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Projects: Companies mainly visit this part of the database in order to (a) identify potential projects, (b) submit tenders, etc.

Companies: The project evaluation committee members mainly visit this part of the database in order to learn about the companies that are participating in a tender.

Statistics, Graphs & Data Analysis: Project evaluation committee members mainly visit this part of the database. Statistics and data analysis tools offered in here facilitate the decision-making process.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about companies, evaluation criteria, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a company, a member of the project evaluation committee, the database administrator, etc. The companies can access the database to check available projects and apply for tenders. Members of the project evaluation committee check the database in order to learn about the participants or to use the statistical analysis tools offered to process data, draw conclusions, and make recommendations. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Rapcsák, T., Sági, Z., Tóth, T., Kétszeri, "Evaluation of tenders in information technology." *Decision Support Systems* **30**: 1-10, 2000.

Problem Description

A common practice to motivate employees to improve their performance is using incentive bonus systems. Such a system sets a number of individual and department performance measures. A weight is assigned to each measure. Employees are then evaluated based on their individual and group performance. Employees who collect the highest weighted number of points are promoted through salary increase. The objective of this project is to build a database that will carry the information about employees, performance evaluation measures, etc. and create a decision support system that will help the management decide about promotions.

Below we present the main steps followed during the employee performance evaluation process. Although the steps to be followed are straightforward, there are a few challenges managing the system manually.

1. Based on their individual and group performance, employees get a number of points on each performance evaluation measure. The points are then weighted using the weights assigned to each measure. Employees are promoted based on the sum of the weighted points.
2. Employees are grouped into two groups: direct employees (the ones directly involved in production, such as operators) and indirect employees (the ones indirectly involved in production such as office staff and service employees). The main reason for grouping employees is that direct employees get promotion points based on their individual and departmental performance whereas indirect employees get promotion points based on individual performance and overall performance of the plant.
3. Promotion bonus points of direct employees are calculated as follows:
 - ❖ Employees get 60 points if the unit production target is met. Otherwise, depending on the deviation from the target, employees get more or fewer points.
 - ❖ Employees are evaluated based on some efficiency and quality factors (such as energy used, quality of product, etc.). The maximum amount of points that an employee gets when the efficiency and quality targets are met is 20.
 - ❖ Employees get points based on labor utilization factors. The maximum number of points one can get is 10. This measure gives incentive to managers to assign the minimum number of employees needed to a job.
 - ❖ Employees get points based on the overall performance of the plant. The maximum number of points one can get is 10.
4. Incentive bonus points for indirect employees depend on the overall department/plant performance. For example, Ms. Jane works as the administrative secretary of the production department. Let X be the total weighted incentive points collected by the direct employees of this department and Y be the total weighted incentive points from direct employees of the plant. Ms. Jane gets a $Z_1\%$ of the X points and $Z_2\%$ of the Y points.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrity constraints; (c) Indicate the normal form of each relation created. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Employee*: The main attributes are social security number, name, address, department, skills, degree, type (direct or indirect employee), etc.
2. *Performance evaluation measure*: The main attributes are identification number, name, description, weight, etc.
3. *Plant*: The main attributes are identification number, name, manager's name, total number of employees, total number of departments, targeted monthly production, history of monthly production, targeted efficiency level, etc.
4. *Department*: The main attributes are identification number, name, manager's name, total number of employees, targeted monthly production, history of monthly production, targeted efficiency level, etc.

Note the following: (a) Every time the employees are evaluated, the following information is recorded: transaction identification number, date, number of points, etc; (b) Departments belong to a plant.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following queries enable the user to calculate the total weighted points per employee:
 - a. Create a query that prompts for the name of a direct employee and presents the following: the points received in all performance measures, the weight of each measure, and the weighted points for each measure.
 - b. Create a query that presents the total of the weighted points from direct employees for each department and the plant.
 - c. Create a query that presents the following information about the indirect employees: the number of points received in all the performance measures, the weight of each measure, and the corresponding weighted points.
 - d. Create a query that presents the total of the weighted points from indirect employees for each department and the plant.
2. Create a query that presents the total, average, minimum, and maximum number of points collected by the direct employees of all the departments in this database.
3. Create a query that presents the total, average, minimum, and maximum number of points collected by the indirect employees of all the departments in this database.
4. Create a query that presents the name of the employee who received the highest number of weighted points.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: employees, performance evaluation measures, departments, etc. These forms allow the user to add, update, and delete information about employees, performance evaluation measures, departments, etc.
3. Create a form that allows the user to browse through the information about each employee. Create a subform that presents for each employee the number of points received in each performance evaluation measure, the weight of each measure, and the weighted points for the employee. Create a textbox that calculates the total weighted points received by the employee. Insert a command button that, when clicked-on, presents the name of the employee who received the maximum number of weighted points.
4. Create a form that allows the user to choose a department name from a combo box. Create a subform that presents the following information about the selected department: employee names and total number of weighted points each one received. Insert a textbox that presents the total number of weighted points received by the selected department.
5. Create a form that allows the user to select a performance measure from a list box. Create a subform that presents the following information: the name of the departments listed in the database and for each department the total points collected on the selected performance measure. Sort the information presented in the subform by total number of points.

Design a logo for this database. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

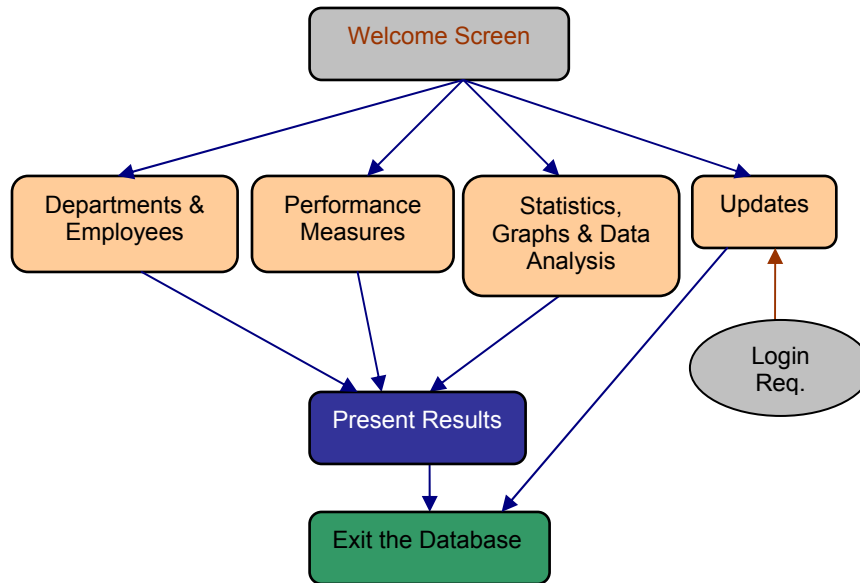
Reports:

1. Report the total number of points collected by each direct employee. Sort the information in a descending order of total number of points.
2. Report the total number of points collected by each indirect employee. Sort the information in a descending order of total number of points.
3. Report for each employee the points received in each performance measure. Calculate and present the total number of points received by each employee.
4. Use the chart wizard to present:
 - a. The average number of weighted points received in the plant during the last ten bonus incentive programs. This chart will enable the managers to identify trends and analyze the impact of the program itself on the plant's performance.
 - b. The average number of weighted points received by each department during the last ten bonus incentive programs. This program may affect departments differently. Apparently, the same set of measures cannot be used to evaluate the performance of all departments. This chart will enable managers to identify the right set of performance measures to measure performance and motivate employees in different departments.

- c. The average number of weighted points received by each employee during the last ten bonus incentive programs.
5. Use the label wizard to create labels with the address of each employee.

Visual Basic.NET Application Development

This database application can be used by managers, employees, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Departments and Employees: This part of the database keeps information about employees and departments. Managers browse the database to learn about the points collected by a particular employee, a particular department, etc.

Performance Measures: Managers and employees browse this part of the database to learn more about different performance measures.

Statistics, Graphs & Data Analysis: Managers often visit this part of the database to identify (a) trends in total number of points collected by employees, departments, and plants; and (b) the best set of performance measures to be used to evaluate the performance of a particular employee, department, plant, etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about employees, performance measures, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be an employee, a manager, the database administrator, etc. The employees search

the database to identify the points collected in the evaluation process. The information provided in this database should help the managers decide on the promotion of the employees. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Performance Based Compensation in an Organization

Problem Description

In many manufacturing plants, compensation plans for officers are completely different from the plans used to compensate other employees. Usually, workers and supervisors get a fixed annual increment in their salary. However, officers receive salary increments and monetary benefits whose values are determined on annual bases. The size of salary increase for officers is determined by top management and depends on individual performance as well as on the overall performance of the company. The main objective of this project is to create a decision support system that will facilitate the process of compensating officers.

Officers are categorized into two main groups: E-level officers and O-level officers. E-level officers supervise O-level officers, and the O-level officers supervise other employees. The officers get a certain yearly percentage increase on their salary and a one-time monetary benefit that ranges from \$5,000 to \$50,000. The decision about the officers' salary increases as well as promotions is made on yearly bases. Officers' performances are measured by superiors on a fixed scale (1-10). Major milestones of the work done by the officer are also considered in this evaluation process. The evaluation process captures training imparted to the officer and helps in determining future training needs.

The process of evaluating the performance of the officers follows these steps:

1. Officers:
 - a. whose rating has exceeded the company's average rating by at least 10% in the last two years and
 - b. who have been recommended by the division head for promotion

will be promoted by one level and receive an annual increment of 10 % on their salary.
2. Officers:
 - a. whose rating has exceeded the company's average rating by at least 5% in the last three years and
 - b. who have not been promoted in last three years

will be promoted by one level and receive an annual increment of 10 % on their salary.
3. Officers, whose rating exceeded the company's average rating by 5%, will receive a 10 % annual increase in the salary.
4. Officers whose rating exceeded the company's average rating by 0-5% will receive a 5 % annual increase in salary.
5. The rest of the officers will not receive a salary increase or promotion, and a letter will be sent with expectations.

The one-time monetary benefit the officers receive is determined based on the milestones achieved by the officers, such as successful accomplishment of cost reduction initiatives, quality improvement initiatives, etc. The monetary benefits of O-level officers are calculated based on the following table.

Table 1: Performance-based benefits.

Performance	Benefit
Cost reductions by at least \$N Millions (N is the number of participants in the team) and an improvement in all the quality measures by at least 10%.	\$50,000
Cost reductions by at least \$N/2 Millions and an improvement in all the quality measures by at least 10 %.	\$40,000
Cost reductions by at least \$N/3 Millions and an improvement in all the quality measures by at least 10 %.	\$30,000
Cost reductions by at least \$N/4 Millions and an improvement in all the quality measures by at least 10 %.	\$20,000
Cost reductions by at least \$N/4 Millions or an improvement in all the quality measures by at least 10 %.	\$10,000
Cost reductions by at least \$N/6 Millions	\$5,000

The quality measures shown in Table 1 give incentives to officers to work toward increasing the productivity of the unit. The monetary benefits the E-level officers receive are twice as big as the benefits of the O-level officers. The expectations for quality improvements and cost reductions are kept the same as for the O-level officers.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) Indicate the normal form of each relation created. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Officer*: The main attributes are identification number, name, address, department, skills, type (O-level or E-level officer), history of annual ratings, history of annual salary increases, history of annual benefits, history of achievements, etc.
2. *Quality Performance Measure*: The main attributes are identification number, name, description, etc.
3. *Team*: The main attributes are name, department, name of the employees on the team, etc.
4. *Training Program*: The main attributes are identification number, name, descriptions, list of skills the program enhances, etc.

Note the following: (a) When an officer takes a training program, the following information is recorded in the database: training date, description of special activities performed, grade, etc; (b) Officers' promotion depends on their achievements. When officers are evaluated, their achievements on each quality performance measure are recorded. For this purpose, cost reductions or unit productivity are considered quality performance measures as well; (c) Officers are responsible for a particular team. When an officer is assigned to a team, the following information is recorded: assignment date, list of goals set (such as production quantity, cost reduction, etc.), list of achievements (such as actual production quantity, actual cost reductions, actual improvements in different quality measures, etc.), etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following queries allow the user to search the database to learn about training programs:
 - a. Create a query that prompts for the name of an officer and returns the name and date of the training programs that officer has participated in.
 - b. Create a query that prompts for the name of a training program and lists the skills an employee would gain by following the program.
 - c. Create a query that prompts for the name of a training program and a date and lists the name of the officers who participated in that program.
 - d. Create a query that prompts for the name of a skill and returns the identification number and name of the training programs that help enhance this skill.
2. The following queries allow the user to search the database to learn about officers' performance:
 - a. Create a query that prompts for the name of an officer and returns that officer's ratings, promotions, salary increases, and benefits in the last ten years.
 - b. Create a query that prompts for the name of a team and returns the name of the officer managing the team and a list of goals set and achievements of the team.
 - c. Create a query that presents the name of the team that has the maximum gap between the goal and actual production quantity.
 - d. Create a query that presents the name of the team that has the maximum gap between the goal and actual cost reductions.
3. The following queries will help to calculate the salary increases and benefits to be given to officers:
 - a. Create a query that lists the O-level officers.
 - b. Create a query that lists the E-level officers.
 - c. Create a query that prompts for the name of an officer and returns the following: cost reductions recorded in the current term by that officer's team; cost reduction goal; actual production quantity; production quantity goal; the achievement on other performance quality measures and the goals set for each measure; team's % increase (or decrease) in the production quantity, costs, and performance quality measures as compared to the goals set and overall performance of the company.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: officers, teams, quality performance measures, etc. These forms allow the user to add, update, and delete information about officers, teams, quality performance measures, etc.
3. Create a form that allows the user to browse through the information about the officers. Create a subform that for the selected officer presents the following: the name, date,

and grade received in the training programs attended. Create another subform that presents a list of salary increases, benefits, and promotions of the selected officer. Insert a subform that presents the following: actual production quantity and cost reductions of the team the selected officer supervises; production quantity and cost reduction goals; the company's actual production quantity and cost reductions; and finally the percent increase (or decrease) in the production quantity, cost reductions, and other quality measures for the team supervised by the selected officer as compared to the company's performance. Insert a command button that, when clicked on, presents the name of the team and the name of the supervising officer who had the best performance.

4. Create a form that allows the user to choose from a combo box the name of a training program. Create a list box that presents the dates this program was offered. Create another list box that presents the name of the skills the selected program enhances. Create a subform that lists the name of the officers who have taken this program together with the program starting date and the grade assigned to each officer.

Design a logo for this database. Insert this logo in the forms created above. Have the background color of the forms light yellow and the border color for the titles green. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

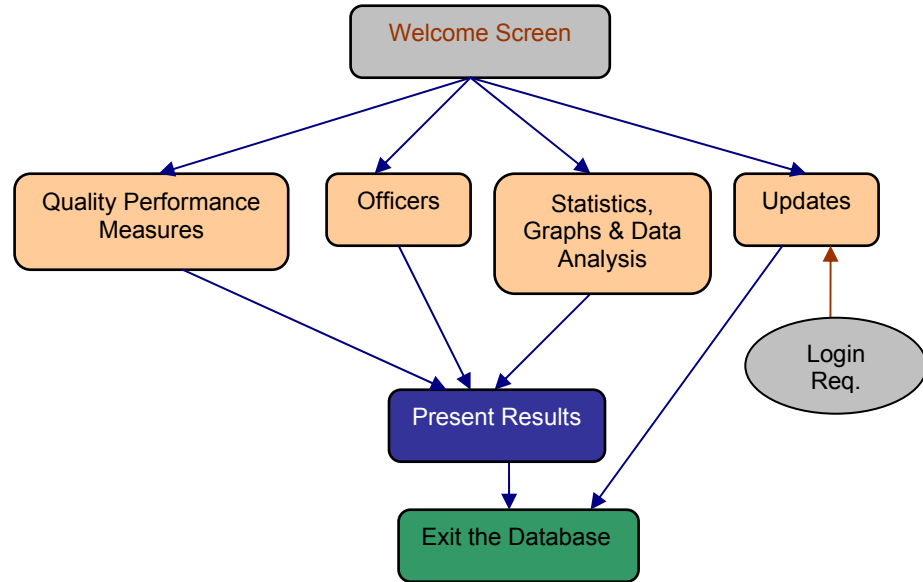
Reports

1. Create a summary report that contains the names of the officers who received a salary increase, a bonus, or a promotion. For each officer, indicate the type of compensation received.
2. Use the chart wizard to plot the following:
 - a. The overall production quantity for each team during the last ten quarters.
 - b. The overall cost reductions for each team during the last ten quarters.
 - c. The percentage improvement in the production quantity for each officer during the last ten quarters.
 - d. The percentage improvement in the costs for each officer during the last ten quarters.
 - e. The percentage improvement in the quality performance measures (as compared to the company's performance) for each officer during the last ten quarters.
3. Create a congratulation letter to be sent to the officers that received the following:
 - a. A promotion,
 - b. A salary increase,
 - c. A bonus,
 or any combination of the above.
4. Create a letter that will be sent to the officers who did not receive compensation. The report should include these facts: (a) the officer's team performance in quality, actual production quantity, and actual cost reductions; (b) the company's quality performance, production quantity, and cost reductions; and (c) comparisons of the team's performance with the company's performance.

- Use a label wizard to create labels with the address of each officer.

Visual Basic.NET Application Development

This database application can be used by supervisors, officers, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Quality Performance Measures: This part of the database is browsed by managers, officers, etc. in order to learn about different quality measures used to evaluate the performance of an officer.

Officers: This part of the database is browsed by officers and managers to learn more about the performance of officers; identify trends in performance; learn about promotions and salary increases; etc.

Statistics, Graphs & Data Analysis: This part of the database is investigated in order to identify trends in: (a) the performance of officers, (b) the performance of teams, (c) the performance of the company, etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about officers, training programs, quality performance measures, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be an officer, a manager, the database administrator, etc. Officers browse the database in order to: (a) understand how they are doing as compared to other teams in the company; (b) identify adequate training programs; (c) identify quality performance measures that need more attention; etc. The information in this database will help the management to

Case Study 30 ■ Performance Based Compensation in an Organization

decide about officers' salary increase, bonus, and promotion. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

DSS for an Electric Utilities Company: Compliance with Clean Air Act

Problem Description

The objective of this project is to build a decision support system that will facilitate the process of making strategic decisions in a power producing (utility) company. The company produces power by using coal-fired boilers. As the government approved the Clean Air Act (CAA) Amendment proposed by the Environmental Protection Agency (EPA), the company faced new challenges in managing the business. The amendment limits the amount of sulfur dioxide (SO₂) emission from electric utility companies. Meeting the new limits requires considerable reduction in sulfur dioxide emission.

After long discussions, the management came to the conclusion that in order to meet the sulfur dioxide emission limits, the company should take one of the following actions:

- ❖ Retrofitting existing coal-fired units with fuel-gas desulfurization units. This is one of the many pollution control technologies the company could follow.
- ❖ Using fuel that has lower sulfur content instead. This, however, would increase costs.
- ❖ Purchasing (selling) allowances. In this case, if the emission of SO₂ is higher (smaller) than the limits, the company should purchase (sell) allowances.
- ❖ Modifying the existing unit utilization scheme of the plant. This means using units that have lower emission rates more intensively than units with higher emission rates. This strategy will allow the company to meet demand for electricity and keep the total emission within the limits.

Note that, the electric utility company has several plants, and each plant has several coal-fired units. The objective of this project is to build a decision support system that will help the managers decide about the action that should be taken. For this purpose, a database will be created that contains data about the company as well as external data. The database should be organized properly for easy extraction and utilization of the data stored in it. The information from the database is then inputted to the solver that analyzes the data and provides information that will enable the managers to make the right decision.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) Indicate the normal form of each relation created. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Plant*: The main attributes are plant identification number, name, address, type, manager's name and contact number, total number of units in the plant, etc.
2. *Unit*: The main attributes are identification number, name, address, type, manager's name and contact number, total number of boilers, etc.
3. *Boiler*: The main attributes are identification number, name, capacity, purchase date and price, description of the functionalities, location, etc.

4. *Coal*: The main attributes are name, sulfur content (SO₂/lb), unit cost, price of purchasing a unit of allowance, scrubber rate, etc.

Note that when the boiler is fired to burn coal, the following data is recorded: boiler status, coal burning rate (in loads burned per hour), amount of coal to be burned, heat rate, amount of sulfur emitted, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following queries help to identify whether the company is currently complying with the CAA Amendment.
 - a. Create a query that calculates the total amount of coal burned in all the plants during the current year.
 - b. Create a query that calculates the total amount of energy generated from all the plants during the current year.
 - c. Different types of coal have different sulfur content. Consider this fact when creating a query that calculates the total amount of sulfur emitted from this company in the current year.
2. The following queries help the management identify strategies to be followed in order to comply with the CAA Amendment:
 - a. Create a query that prompts for the name of a plant and presents the name of the units located in this plant. For each unit, present the following: the total amount of coal (per coal type) burned during the current year and the total amount of sulfur emitted in the current year.
 - b. Create a query that prompts for the name of a unit and presents all the boilers located in this particular unit. For each boiler, present the following: the total amount of coal (per coal type) burned during the current year and the total amount of sulfur emitted in the current year.
 - c. The management believes that there is a positive relation between the age of a boiler and emission rate. If this is the case, the management will consider replacing some of the existing boilers. Create a query that prompts for the name of a coal type and presents the following for each boiler in the database: (a) the actual amount of sulfur emitted from burning the particular coal type, and (b) the corresponding age. Sort the information by boiler age.
 - d. The management is considering replacing existing coal-fired units with fuel-gas desulfurization units. Use a make-table query to create a new table tblFuelBoiler. The main fields of this table are as follows: identification number, name, capacity, purchase date and price, description of the functionalities, etc. Create a query that counts the coal-burning boilers that are at least five years old. Create another query that calculates the total cost of replacing the old (older than five years) coal-burning boilers with fuel-burning boilers.
 - e. Create a query that lists all the different types of coal that can be used. For each coal type, present the following: name, amount of sulfur per unit of coal, and unit price.

3. Different coal types have different burning rates and sulfur emission rates. Having knowledge of the total energy generated in the current year, create a query that prompts for the name of a coal type and presents the total amount of this type of coal needed to satisfy the demand for energy. The query should also calculate the total sulfur dioxide that would be generated from burning this type of coal.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: plants, boilers, coals, etc. These forms allow the user to add, update, and delete information about plants, boilers, coals, etc.
3. Create a form that allows the user to select the name of a plant from a combo box. Create a subform that presents the following information about the selected plant: name, address, total number of boilers, total amount of coal burned in the current year, and total amount of sulfur dioxide emitted in the current year. Create another subform that lists the names and addresses of all unit of this plant. Insert a command button that, when clicked on, returns the name of the plant that has the minimum sulfur dioxide emission rate.
4. Create a form that allows the user to select the name of a unit from a list box. Create a subform that presents the following information about the selected unit: the name, address, total number of boilers, total amount of coal burned in the current year, and total amount of sulfur dioxide emitted in the current year. Create another subform that lists the name and location of all boilers of this unit. Insert a command button that, when clicked-on, returns the name of the unit that has the minimum sulfur dioxide emission rate.
5. Create a form that allows the user to select the name of a boiler from a combo box. Create a subform that presents the following information about the selected boiler: identification number, name of the unit it belongs to, name of the plants it belongs to, total amount of coal burned in the current year, and total amount of sulfur dioxide emitted in the current year. Insert a command button that, when clicked-on, returns the name of the boiler that has the minimum sulfur dioxide emission rate. Insert another command button that, when clicked-on, returns the name of the ten oldest boilers.
6. Create a form that allows the user to select the name of a coal type from a list box. Create a subform that presents the following information about the selected coal type: name, sulfur content (SO₂/lb), and unit cost. Insert a command button that, when clicked-on, presents the total amount of coal (of the selected type) burned during the current year and the corresponding total purchase cost. Insert another command button that, when clicked-on, presents the name and unit cost of the coal that has the minimum sulfur content.
7. Create a form that would allow the user to define different scenarios about the type of coal to be used and corresponding quantity and calculate the total energy generated and sulfur dioxide emitted.

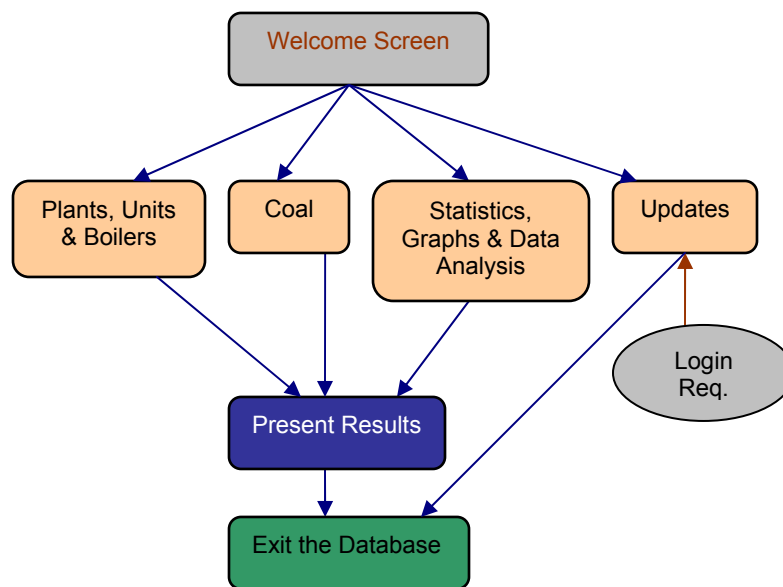
Design a logo for this database. Insert this logo in the forms created above. Have the background color of the forms light yellow and the border color for the titles green. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create a summary report for the company. The report should contain the following: the company's name, total number of plants, total number of units, total number of boilers, total energy produced, total sulfur dioxide emitted, total production cost, etc.
2. Create a summary report for each plant. The report should contain the following: the plant's name, total number of units, total number of boilers, total energy produced, total sulfur dioxide emitted, total production cost, etc.
3. Create a summary report for each unit. The report should contain the following: the unit's name, total number of boilers, total energy produced, total sulfur dioxide emitted, total production cost, etc.
4. Create a summary report about the boilers used. The report should present the following: boilers' identification number, name of the unit and plant it belongs to, purchase date, and cost, age, capacity, etc.
5. Use the chart wizard to plot the following:
 - a. The total energy produced per year in the last ten years.
 - b. The total sulfur dioxide emitted per year in the last ten years.
 - c. The total number of boilers used per year in the last ten years.
 - d. For the company's ten oldest boilers, present the total energy produced and total sulfur dioxide emitted (per unit of coal used) per year in the last ten years.

Visual Basic.NET Application Development

This database application can be used by managers, database administrators, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Plants, Units & Boilers: The management browses this part of the database to identify: (a) the amount of energy generated in different plants, units, and boilers; (b) the amount of sulfur dioxide emitted in different plants, units, and boilers; (c) the boilers that need to be replaced; etc.

Coal: This part of the database keeps useful information that helps the managers identify the type of coal and quantity to be used in order to satisfy demand for energy at minimum cost without violating the sulfur dioxide emission limits set by the CAA Amendment.

Statistics, Graphs & Data Analysis: This part of the database is investigated in order to identify trends in: (a) the total energy demanded; (b) the total sulfur dioxide emission; etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about plants, units, boilers, coal, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a manager, the database administrator, etc. The managers use the database to identify opportunities that would enable the company to satisfy demand for energy at the minimum cost without violating the limits set by the CAA Amendment on the total sulfur dioxide emission. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Ghandforoush, P., Sen, TK., Wander, M., "A decision support system for electric utilities: compliance with Clean Air Act." *Decision Support Systems* **26**: 261-273, 1999.

Problem Description

The objective of this project is to build a decision support system for a wood panel manufacturer. This system will allow the company to identify ways to improve their processes and minimize costs of manufacturing. Wood panels are manufactured using small pieces of wood (for example, chips, flakes, splinters, strands, shreds, etc.) that are bound together using an organic binder. There are a number of raw materials that can be used to produce these wood particles. Manufacturers have flexibility in selecting the raw materials.

There are a variety of wooden panels, and each one has distinguished properties. Depending on their usage, wood panels have different quality requirements. For example, quality requirements for wood panels used in middle layers are different from quality requirements for wood panels used in outer layers. The total production of a wood panel manufacturer depends on the following: (a) the type of raw materials used and (b) the quality requirements for the final product.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) Indicate the normal form of each relation created. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Raw Material*: The main attributes are identification number, name, description, unit cost, expected monthly consumption, inventory level, etc.
2. *Final Product*: The main attributes are identification number, name, description, unit cost, unit price, expected monthly production, current inventory level, list of raw materials and corresponding quantities needed to produce one unit of the final product, density requirements, etc.
3. *Equipment*: The main attributes are identification number, name, location, capacity, etc.

Note the following: (a) When a raw material is used to produce a particular product, the following information is recorded: transaction identification number, date, amount of final product produced, quality level of the final product, amount of raw material used, cost of raw material, etc; (b) A final product can be produced using different combinations of raw material. Depending on the type of the raw material used and quantity used, the resistance and quality of the final products varies; (c) Particular equipments cannot be used to process all the raw materials and produce all the final products. Also, the yield of a final product depends not only on the type of raw material, but on the equipment used as well. Therefore, the database keeps track of the following: (i) the raw materials that equipment can process; (ii) the final products that equipment can produce; and (iii) the yield of the final products produced on that equipment.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create a query that prompts for the name of a raw material and lists the name of the final products that can be produced using this raw material. The list should also contain the quantity of raw material needed to produce one unit of final product.
2. Create a query that prompts for the name of a final product and presents a list of the raw materials that can be used. Present the quantity needed of each raw material to produce one unit of the final product.
3. Create a query that prompts for the name of an equipment and presents a list of the final products that can be produced using this particular equipment.
4. Create a query that presents for each final product the total production in the current year.
5. Create a query that presents for each raw material the total consumption in the current year.
6. Create a query that prompts for the name of a final product, a quality level, and a resistance level and returns the combination (type and quantity) of the raw materials that would produce the required final product.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: final products, raw materials, equipments, etc. These forms allow the user to add, update, and delete information about final products, raw materials, equipments, etc.
3. Create a form that allows the user to choose the name of a final product from a combo box. Create a subform that presents the name and quantity of the raw materials that can be used to produce one unit of the selected final product. Insert a textbox that presents the cost of producing one unit of the final product. Insert another textbox that presents the price per unit of this final product.
4. Create a form that allows the user to choose the name of a raw material from a listbox. Create a subform that presents a list of products that contain this raw material. Present the number of units of the selected raw material needed to produce one unit of each final product.
5. Create a form that allows the user to browse through the information about the equipments. Create a subform that lists the name of the final products that can be produced on the selected equipment. For each final product present the corresponding production yield. Create a subform that lists the names of the raw materials that can be processed on the selected equipment.

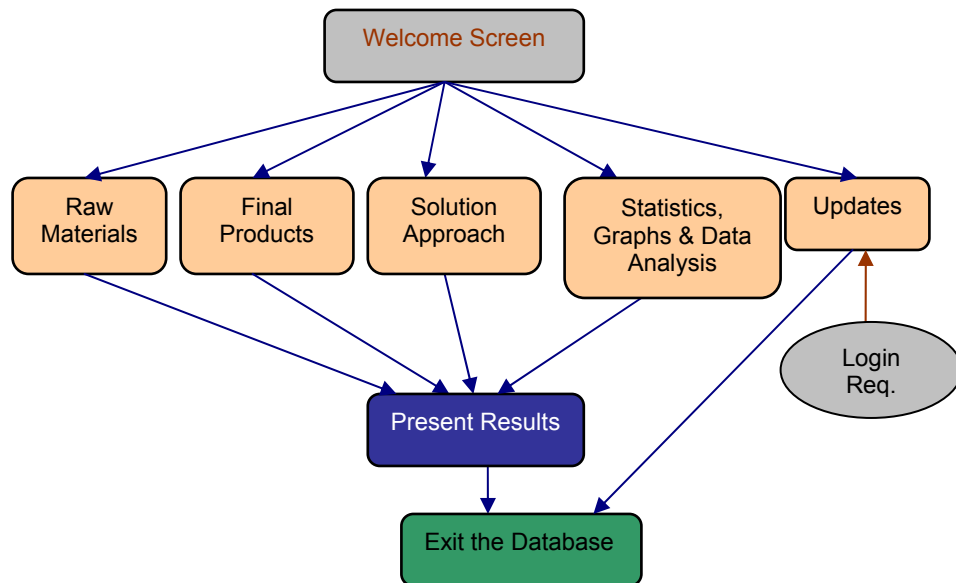
Design a logo for this database. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create a summary report that presents the production amount for each final product. The report should include the following: unit production cost, total cost, unit price, sales, and earnings from sale.
2. Create a summary report about the raw materials used. The report should contain the following information: unit cost and total consumption. Calculate the overall cost of raw materials.
3. Create a report that presents for each final product the raw material (name and quantity) combination that gives the highest resistance and quality. Present the cost of producing one unit of the final product and corresponding sale price.
4. Create a summary report for the equipments. The report should present the name of the equipments owned by the company. For each equipment, present the amount and name of final products produced in the current year.
5. Use the chart wizard to plot the following:
 - a. Production quantity for each final product per year during the last ten years.
 - b. Earnings from sales of each final product per year during the last ten years.
 - c. For each raw material, the amount used per year during the last ten years.
 - d. For each raw material, the unit cost paid per year during the last ten years.

Visual Basic.NET Application Development

This database application can be used by managers, employees, database administrators, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Raw Materials: Managers visit this part of the database in order to: (a) check the inventory level and unit cost of a particular raw material; (b) identify final products that contain a particular raw material; etc.

Final Products: Managers visit this part of the database in order to: (a) identify the combination of raw materials needed to produce a final product with a particular resistance and quality level; (b) identify the total production of a particular product during the current year/month; (c) identify the cost and price of producing one unit of final product; etc.

Solution Approach: We propose a mathematical model to find the combination of raw materials that can be used to produce one unit of the final product at the minimum cost.

The decision variables for this model are as follows:

x_{ijk} presents the amount of raw material i used to produce layer k of panel j .

The objective function minimizes the total cost incurred to produce the required amount of particleboards of a certain quality level.

$$\sum_{i=1}^R \sum_{j=1}^P \sum_{k=1}^2 (RC_i + PC_{ijk}) x_{ijk},$$

where, R presents total number of raw materials, P presents the total number of finished products, RC_i presents the unit cost of raw material i , PC_{ijk} presents the unit production cost for raw material i , used in panel j and layer k .

The constraints of this model are the following: (a) equipment capacity constraints; (b) panel quality requirements; (c) raw materials availability; and (d) non-negativity constraints.

Statistics, Graphs & Data Analysis: This part of the database is used to identify trends in: (a) the raw material consumption; (b) the raw material cost; (c) the final product cost and price; (d) the final product production; etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about raw materials, final products, equipments, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a manager, the database administrator, etc. The manager browses the database to learn more about the following: (a) the different combinations of raw materials that can be used to produce a unit of final product with the desired resistance and quality; (b) the inventory level and unit cost of the raw materials; (c) unit production cost, unit price, and total production quantity of the final product; etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Buehlmann, U., Ragsdale, C.T., Gfeller, B., "A spreadsheet-based decision support system for wood panel manufacturing." *Decision Support Systems* **29**: 207-227, 2000.

Problem Description

In large organizations, there is high duplication of basic technology and knowledge. For example, similar pollution-controlling devices may be installed in different units within the organization. As a result, although not necessary, the expertise gained from using these equipments is maintaining in more than one unit. Companies are also concerned about losing the knowledge and expertise gained by employees when they leave the company.

Knowledge Management is a novel concept that emphasizes capturing expertise or standards and providing tools to share them throughout different units within the organization. A number of organizations have invested a lot of time and capital in developing knowledge management systems. Lotus Notes from IBM is a widely used framework for these systems.

This project intends to design and develop a database that can be used as a knowledge management system for small organizations. The system will capture the knowledge in a particular format, organize the knowledge, present it to the expert for approval, and finally present the results to the end-user whenever required.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Department*: The main attributes are name, address, name and telephone number of the contact person, etc.
2. *Employee*: The main attributes are social security number, name, address, area of specialty, degree, hire date, etc.
3. *Equipment*: The main attributes are identification number, name, description, purchase date, price, location, activities performed using the equipment, products produced using the equipment, etc.
4. *Knowledge/skill*: The main attributes are name, description, comments/feedback, degree requirements, domain, key words, etc.
5. *Process*: The main attributes are identification number, name, description, activities associated with a process, etc.

Note the following: (a) A particular type of (instance of entity) knowledge cannot be added in this database unless it is approved by the management; (b) Managers supervise a number of employees; (c) When management approves a particular type of knowledge, the approval date is recorded; (d) In order to operate a particular equipment, the employees need to have certain skills (knowledge); (e) A process requires the involvement of a number of equipment and employees; (f) A department owns a number of equipments.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create a query that prompts for the name of a particular knowledge and returns the name of the employees who have that knowledge. For each employee, present years of experience. This query will help identify employees who are likely to get a promotion.
2. Create a query that prompts for the name of a particular knowledge and returns the total number of employees in each department who have that knowledge. This query will help capture the need to share the expertise among the departments.
3. The following queries help the management with their day-to-day scheduling problem:
 - a. Create a query that prompts for the name of a process and lists the knowledge required to complete that process.
 - b. Create a query that prompts for the name of a process and lists the name of the equipment used in this process.
 - c. Create a query that prompts for the name of an equipment and lists the knowledge required to operate the equipment.
 - d. Create a query that prompts for the name of an employee and lists the skills that the employee has.
4. Create a query that prompts for the name of a department and returns a list of the skills (knowledge) owned by the department. Present also the number of employees who have the particular skill (knowledge) in this department.
5. Create the following queries:
 - a. Find the knowledge that the company lacks the most.
 - b. Find the most abundant knowledge in the company.
 - c. Find the knowledge the company needs the most.
 - d. Find the knowledge the company needs the least.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: equipment, knowledge, employees, processes, etc. These forms allow the user to add, update, and delete information about equipment, knowledge, employees, processes, etc.
3. Create a form that allows the user to select a particular knowledge from a combo box. Insert a textbox that gives a description of the knowledge that is chosen. Insert a subform that presents a list of the employees who have the particular knowledge. For each employee, present the following: social security number, name, the name of the department, name of the employee's manager, and years of experience. Insert a textbox that presents a count of the total number of employees who have this particular knowledge. Insert another subform that presents a list of the departments and the corresponding number of the employees who have the selected knowledge.

4. Create a form that allows the user to browse through the information about the processes performed in the department. Create a subform that presents a list of the equipments required in a particular process. For the equipments, present: identification number, name, location, name of the department that owns the equipment, and the number of years the equipment has been in operation. Create another subform that lists the knowledge needed to perform the selected process.
5. Create the following "Search Form." The form consist of the following:
 - a. A command button that when clicked-on lists the most needed knowledge.
 - b. A command button that when clicked-on lists the least needed knowledge.
 - c. A command button that when clicked-on lists the most lacked knowledge.
 - d. A command button that when clicked-on lists the most abundant knowledge.

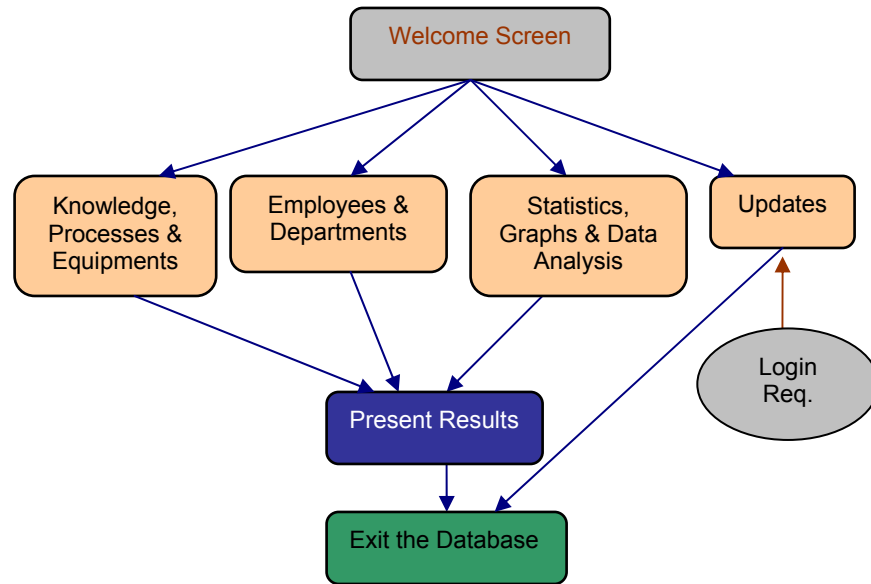
Design a logo for this database. The logo may contain the picture of an equipment, etc. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create a report that presents a count of the employees who need particular knowledge and, for the same knowledge, a count of the number of employees who already have it. One can find out the needs for a particular knowledge by identifying the processes that have to be performed and the need of each process for a particular knowledge type. This report will help the management identify needs for hiring.
2. Create a report that prompts for the name of a particular knowledge and returns the number of employees who have this knowledge in each department. Present also the needs of each department for the selected knowledge.
3. Create a report that presents for each process performed a list of the knowledge needed to complete the process.
4. Create a report that presents for each process a list of the equipments that have to be used.
5. Create a report that presents a list of skills needed to operate the equipments owned by the company.
6. Use the chart wizard to plot the following:
 - a. The total number of employees (overall departments) that have a particular knowledge type per year during the last five years.
 - b. The total number of employees in each department that have a particular knowledge type per year in the last five years.

Visual Basic.NET Application Development

This database application can be used by managers, employees, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Knowledge/Skill, Processes & Equipments: Managers browse this part of the database to learn about the following: (a) the need for a particular knowledge/skill; (b) the knowledge/skills needed to perform a particular process; (c) the knowledge/skills needed to operate an equipment; etc.

Employees & Departments: Managers and employees browse this part of the database to identify the following: (a) employees who have a particular knowledge/skill; (b) needs of a department for a particular skill; etc.

Statistics, Graphs & Data Analysis: Managers often visit this part of the database to identify trends in the following: (a) the number of employees who have a particular knowledge/skill; (b) the number of equipments needed; etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about equipments, knowledge, employees, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a manager, the database administrator, etc. The managers should have a login name and password to be able to access the database. The information provided in this database helps managers decide about the following: (a) hiring new employees who have a particular knowledge/skill; (b) distributing the knowledge (employees) among departments based on departmental needs; etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

Hospital managers are interested in not only keeping a database with detailed information about the inpatients, but also keeping information about the outpatients of the hospital. This information is intended to help management in deciding about the number of personnel, equipments, operating hours of the ward, etc. Every year managers survey the patients coming to the Outdoor Patient Division (OPD) by capturing the following: (a) the time a particular patient entered the hospital; (b) the time it took to visit the doctor, perform the necessary analysis, use equipments, etc; (c) and the time the patient left the hospital. The survey is performed on certain predetermined days. The decision about when to perform the survey is made taking into consideration the flow of patients on particular days of the week and in particular seasons.

The data collected from the surveys will be used to identify trends in the service level provided by the hospital. This information will enable the managers to identify service areas that need improvements. Managers are mainly concerned about the following: (a) identifying the number of doctors needed in each department; (b) identifying the operating hours for each department; (c) identifying the number of equipments required; (d) identifying the schedule of doctors and nurses; and (e) identifying promotional plans to help with the distribution of patient arrivals.

Note that the data collected using surveys is entered manually in the database. This often causes data entry errors. Therefore, it is essential that we include a number of validation rules to prohibit incorrect data entries.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Equipment*: The main attributes are identification number, name, description of its functionalities, type, location, purchase cost, purchase date, etc.
2. *Department*: The main attributes are identification number, name, address, total number of doctors, total number of nurses, etc.
3. *Doctor*: The main attributes are identification number, name, address, area of specialty, current working schedule, etc.
4. *Nurse*: The main attributes are identification number, name, address, skills, current working schedule, etc.
5. *Patient*: The main attributes are identification number, name, address, type (inpatient or outpatient), disease, etc.
6. *Room*: The main attributes are room number, type (operating room, ward, etc.), capacity, description, etc.

Note the following: (a) When a patient visits a particular doctor, the following information is recorded: visit date and time, length of the visit, diagnosis, doctor's comments, etc; (b) When a patient uses equipment, the following information is recorded: date and time of use, length of use, description of the process, amount charged, etc; (c) When a patient is assigned to a room, the following information is recorded: check-in date, check-out date, amount charged, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create queries that will provide the following statistics about the data collected from the surveys.
 - a. The maximum number of patient arrivals in a day.
 - b. The average daily arrivals of patients in the hospital.
 - c. The maximum patient total waiting time. For that instance present the activities performed by the patient, such as the time spent with the doctor, the time spent performing analysis, etc.
 - d. The average/minimum of the total patient waiting time.
 - e. The maximum/average/minimum of patient waiting time to see a doctor.
 - f. The maximum/average/minimum of patient waiting time to use equipment.
 - g. The maximum/average/minimum of patient total waiting. Group this information by department.
 - h. The time of the day that has the maximum/minimum patient arrivals.
 - i. The season that has the maximum/minimum patient arrivals.
 - j. The doctor who has the maximum/minimum number of patient visits.
 - k. The disease that most (least) of the patients suffer from.

The information collected using these queries will help managers identify: the bottleneck department(s), bottleneck equipment(s), bottleneck doctor(s), bottleneck area(s) of specialty, etc. This information will also be the input in managing doctors' and nurses' schedules, deciding about hiring doctors and nurses, and buying new equipments.

2. Using the queries in part (1) the managers can identify the peak hours of a day. Create a query that prompts for the time and day of the week and returns the name of the doctors and nurses who are scheduled to work. For each doctor and nurse, present the area of specialty and skills. Sort the information by area of specialty. Within each area of specialty, sort the information by doctors' last names.
3. A department is identified as the bottleneck of the operations if patients' waiting time to see a doctor is very high. Identify the five departments that have the highest patient waiting time. List also the five departments where patients' waiting time is minimal. For each department, present the total number of doctors and total number of nurses. This information could be used by the managers to: (a) decide about reallocating the doctors and nurses with the purpose of balancing patient waiting time, and (b) decide about hiring new doctors and nurses.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: doctors, nurses, patients, equipment, etc. These forms allow the user to add, update, and delete information about doctors, nurses, patients, equipment, etc.
3. Create a form that allows the user to browse through the information about the departments. Create a subform that lists the doctors working with the selected department. For each doctor, present the following: identification number, name, and area of specialty. Insert text boxes that present the following: average/minimum/maximum of the total patient waiting time and average/minimum/maximum of patient waiting time to see a doctor in the selected department. Insert a textbox that presents the total number of doctors and another textbox that presents the total number of nurses in the department. Insert a subform that lists the equipments used by the department. Present the following information about the equipments: identification number, name, and a description of the activities it performs.
4. Create a form that allows the user to select an equipment from a combo box. Insert a subform that presents a list of the departments that own the same type of equipment. For each department, present the total number of equipments (of this particular type) owned, the average/minimum/maximum weekly usage of the equipments, the average/minimum/maximum idle time, etc.
5. Create a form that allows the user to choose from a list box the name of a doctor. Use textboxes to present the following information about the selected doctor: identification number, address, area of specialty, years of experience, and average/minimum/maximum patient waiting time. Create a subform that lists the names of the patients who visit this doctor. For each patient, present the following: identification number, name, name of the disease the patient suffers from, total number of visits, and average/minimum/maximum waiting time to visit this doctor. Insert a subform that presents the current weekly schedule of the selected doctor.
6. Create a form that allows the user to choose the name of a disease from a combo box. Insert text boxes to present the following information about the selected disease: the total number of patients suffering from this disease; the total number of doctors who are specialized in curing this particular disease; and the total number of nurses who have the skills to work with patients suffering from this disease. Insert a subform that lists the patients suffering from this disease. For each patient, present the following: identification number, name, name of the doctor the patient is assigned to, etc.

Design a logo for this database. The logo may contain the picture of a hospital, doctor, equipment, etc. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

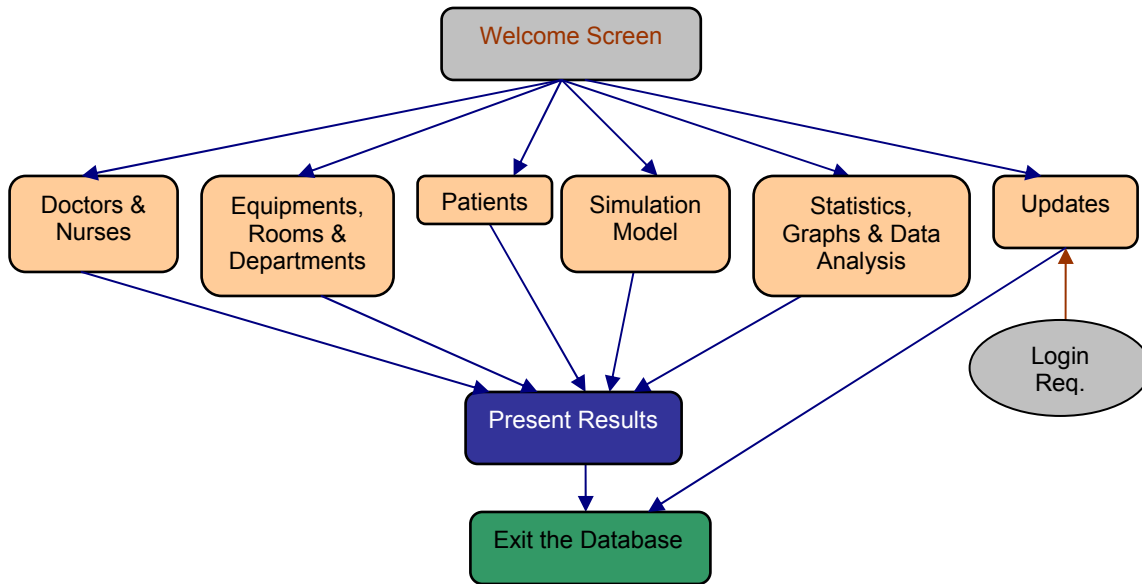
Reports

1. Create a summary report that presents the total amount of money generated in the hospital so far through the following services provided to patients: doctor visits, equipment usage, analysis performed, etc.

2. Use the information collected from the surveys to create the following summary reports:
 - a. Report the following about the five busiest doctors in the hospital: identification number, name, department, area of specialty, years of experience, total number of patients, average/maximum/minimum patient waiting time to visit the doctor, and the weekly schedule.
 - b. Report the following about the five busiest equipments of the hospital: identification number, name, location, years of usage, average/maximum/ minimum patient waiting time to use the equipment, and average/ maximum/minimum time it takes to examine a patient using this equipment.
 - c. Report the following about the five busiest departments of the hospital: name, address, total number of patients, average/maximum/minimum total patient waiting time, total number of doctors, total number of nurses, and total number of equipments.
3. Create for each department of the hospital a report that presents the total number of doctors and nurses who are working on a particular time and day of the week.
4. Use the chart wizard to plot the following:
 - a. Average/minimum/maximum patient waiting time at different times in the day. For example, average minimum/maximum waiting time at 8 a.m., 9 a.m., 10 a.m., ..., 7 a.m.
 - b. Average/minimum/maximum patient waiting time on different days of the week. For example, the average/minimum/maximum waiting time on Mondays, Tuesdays, ..., Sundays.
 - c. Average/minimum/maximum patient waiting time in different departments.
 - d. Average/minimum/maximum patient waiting time for different equipments.
 - e. Average/minimum/maximum patient waiting time for different doctors.
 - f. Average/minimum/maximum number of patients served in a day in different departments.

Visual Basic.NET Application Development

This database application can be used by managers, employees, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the six options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Doctors & Nurses: Managers browse this part of the database in order to: (a) identify the skills of a doctor/nurse; (b) identify the doctor/nurse who is specialized in curing a particular disease; (c) identify the bottleneck doctors/nurses; (d) identify and modify the working schedule of a particular doctor/nurse, etc.

Equipments, Rooms & Departments: Managers, doctors, and nurses browse this part of the database to identify the following: (a) the bottleneck equipments/departments; (b) the average/minimum/maximum waiting time to use equipment; (c) the average/minimum/maximum time a patient uses a room (operating room or ward) of the hospital; (d) the needs for new equipments; etc.

Patients: Managers and doctors browse this part of the database to identify: (a) the patients who suffer from a particular disease; (b) the patients of a particular doctor; (c) the average/minimum/maximum patient waiting time; etc.

Simulation model: We suggest that you build a simple model to simulate this system. The objective of this simulation model is to identify the optimal number of doctors and nurses to be hired; identify whether new equipments are needed and, if so, how many; identify whether more beds are needed and, if so, how many, etc. This information will be used by the management to improve the service level of the hospital.

This is a queuing problem. The data collected from the surveys could be used to identify the arrival rate of the patients in the hospital, patient waiting time, patient consulting time, equipment utilization, etc.

Statistics, Graphs & Data Analysis: Managers often visit this part of the database to identify trends in: (a) the average/minimum/maximum patient waiting time; (b) the average/maximum/minimum equipment idle time; (c) the number of equipments needed; etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about doctors, nurses, equipments, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a doctor, a nurse, a manager, the database administrator, etc. The users should use their login name and password to access the database. Doctors and nurses use the database to identify their weekly schedules, learn about appointments, learn about patients, etc. The information provided in this database helps the managers decide about replacing equipments, buying new equipments, hiring nurses and doctors, etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.+

Water Treatment Facilities in a Municipal Area

Problem Description

Every year, the local government of a municipal area develops environmental protection programs that set the standards about water and air quality. Often, the government has to decide about investing in building water treatment facilities to comply with these protection programs, etc. The process of developing an environmental protection program and making decisions takes a lot of time, as the local government has to identify the following: possible sources of pollution; water quality conditions; the need for new water and wastewater treatment facilities and estimate corresponding investment, operations, and maintenance costs; lowest cost strategies for meeting requirements for water supply and wastewater treatment specified by water quality standards; technical options for improvements; etc.

We want to build a decision support system to help the local government in the process of making a decision. The system is built based on a database that keeps detailed information about the following: water sources available and corresponding quality, investment options available and corresponding costs, operation and maintenance costs at a facility, growth rate of the industrial sector, predefined standards about the maximum level of effluents in water, etc.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Water Treatment Facility*: The main attributes are name, address, name and telephone number of the contact person, quantity of water it treats per month, list of the type of treatments performed, etc.
2. *Water Source*: The main attributes are identification number, location, current water quality and quantity, etc.
3. *Investment*: The main attributes are identification number, capital costs, operating costs, maintenance costs, estimated percent improvement on water quality because of this investment, etc.
4. *Business*: The main attributes are identification number, name, address, name of the contact person, starting date, average water consumption per month, average waste water generated per month, etc. There are two main types of businesses we identify: industrial and agricultural businesses.
5. *Treatment*: The main attributes are treatment number, description, etc.
6. *Regulation*: The main attributes are identification number, name, approval date, description, etc.

Note the following: (a) Water sources need frequent treatment. Every time the water is treated, the following information is recorded: date, quantity, description, etc. (b) A regulation proposes a number of water treatments to be performed.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create a query to identify the number of water treatment facilities located in a particular water source. For each water source, present the total amount of water treated per month.
2. Create a query that presents the following information about each water source: a description of the types of water treatments it gets and a description of the types of water treatments it is supposed to get based on the regulations.
3. The following queries help the management identify the amount of water consumed and amount of water wasted per month:
 - a. Create a query that calculates the total amount of water consumed and the total amount of water wasted per month by agricultural and industrial businesses.
 - b. Create a query that calculates the average amount of water consumed and the average amount of water wasted per month by agricultural and industrial businesses.
 - c. Create a query that presents the total number of industrial and agricultural businesses opened every quarter in the last five years.
 - d. Create a query that presents the average amount of water consumed, the average amount of water wasted per month by agricultural and industrial businesses, and the amount of water that will need treatment.
4. The management is interested to know if the current facilities will be able to handle the increasing need for water treatments. The queries we built in the previous step give insights about the increase in the number of the agricultural and industrial businesses. Create a query that prompts for the increase in the number of industrial and agricultural businesses and returns an estimate on the amount of water that will be consumed by the businesses.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: water treatment facilities, businesses, regulations, etc. These forms allow the user to add, update, and delete information about treatment facilities, businesses, regulations, etc.
3. Create a form that allows the user to browse through the different water treatments recorded in this database. Create a subform that presents a list of regulations that include the selected water treatment. Present the approval date and a detailed description for each regulation.
4. Create a form that allows the user to choose a water treatment facility from a combo box. Create a subform that presents the following information about the selected facility:

name, address, amount of water it treats per month, and years of operation. Create a subform that presents a list of the types of treatments performed in the selected facility.

5. Create a form that allows the user to choose a particular regulation from a list box. Create a subform that presents the following information about the selected regulation: identification number, approval date, and description. Create another subform that presents a list of the water treatments that should be performed on the water sources.
6. Create a subform that allows the user to choose from a combo box a water source. Create a subform that lists the businesses that use the water from the selected water source. For each business, present the following: name, type (industrial or agricultural), the amount of water consumed per month, the amount of water wasted per month, etc. Create another subform that presents a list of the facilities that treat the water coming from this source.

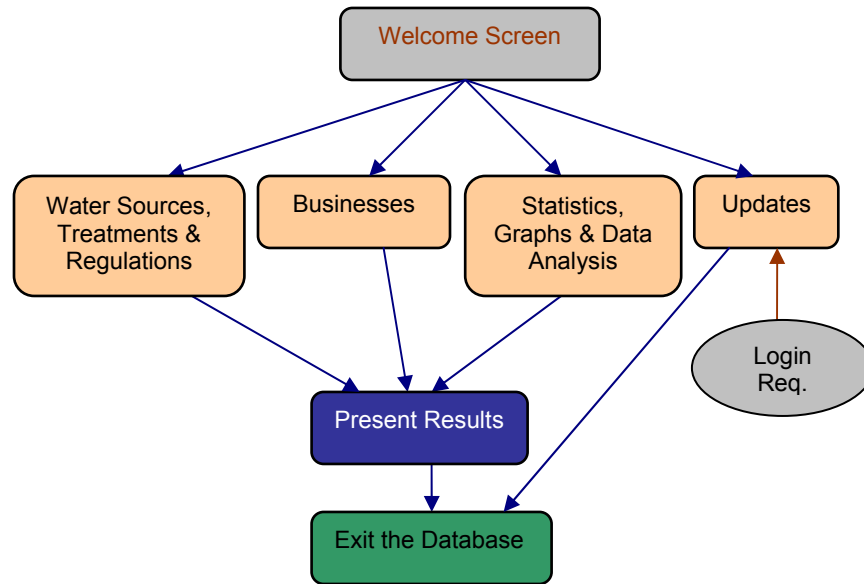
Design a logo for this database. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create a report that summarizes the activities at each water treatment facility. For each facility, the following is presented: name, years of operation, and amount of water it treats per month.
2. Create a summary report about the activities of industrial/agricultural businesses. For each business, present the following: name, the average amount of water consumed, and the average amount of water wasted per month. The report should also present the overall average amount of water consumed and water wasted per month from industrial and agricultural businesses.
3. Create a report that summarizes the regulations that have been approved during the last year.
4. Use the chart wizard to plot the following:
 - a. Average monthly consumption of water by agricultural businesses during the last year.
 - b. Average monthly consumption of water by industrial businesses during the last year.
 - c. Average amount of water wasted per month by agricultural businesses during the last year.
 - d. Average amount of water wasted per month by industrial businesses during the last year.
5. Use the label wizard to create labels with the addresses of the industrial and agricultural businesses.

Visual Basic.NET Application Development

This database application can be used by government employees, businesses, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Water Sources, Treatments & Regulations: This part of the database keeps information about water sources, different treatments used in the water treatment facilities, and regulations approved. Government employees browse this part of the database to learn about: (a) the current activities at the water treatment facilities, (b) the new regulations about water treatment, (c) the compliance with the regulations, etc.

Businesses: This part of the database keeps information about: (a) the amount of water consumed by industrial and agricultural businesses, (b) the amount of water wasted, (c) the compliance of businesses with government regulations, etc.

Statistics, Graphs & Data Analysis: Government employees often visit this part of the database to identify trends in total amount of water consumed or wasted by industrial and agricultural businesses.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about water treatment facilities, regulations, businesses, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a government employee, a business manager, the database administrator, etc. The government employees use the database to learn about water consumption, water treatments, new regulations, and compliance of different businesses with government regulations. Business managers browse this database to learn about new regulations, water treatment requirements, etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Krejčík, J. and Vaneček, S., "Application of DSS for Development of Accession Strategies in the Water Sector in Czech Republic." *Decision Support Systems*, International Workshop 6 April 2000.

Problem Description

Asset and liability management (ALM) is defined as “managing both assets and liabilities simultaneously for the purpose of mitigating interest rate risk, providing liquidity and enhancing the value of the bank.” ALM is concerned with planning, organizing, and controlling asset and liability mixes, volumes, yields, and rates in order to achieve a target interest margin. Managing the assets and liabilities of a company has become a very complex process as the number of transactions between organizations has increased. As a result, the traditional methods of managing assets and liabilities cannot be used.

Asset and liability management depends heavily on the changes of interest rates in the market. The following methods are used to measure interest rate risk: (a) *gap analysis*: compares “the speed at which a financial institution’s assets and liabilities mature, when external interest rate changes” and (b) *duration analysis*: provides a “measure of time weighted average maturity resulting from the cash flows of a financial instrument.” Both tools have advantages and limitations. A number of simulation techniques have also been used to generate interest rate scenarios and determine the interest rate risk. The objective of this project is to build a decision support system that would enable a financial institution to manage the assets and liabilities using the approaches mentioned above.

Assets and liabilities are categorized as either rate-sensitive or rate-insensitive. The value of rate-sensitive assets and liabilities is heavily affected by interest rates. This is why the decision support system provides analysis tools only for rate-sensitive assets and liabilities. The analyses are based on data from internal activities of the company as well as on external data from related financial institutions.

Database Design

We present below the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Asset*: The main attributes are identification number, name, type (rate-sensitive or not), value, rate of return, maturity date, etc.
2. *Liability*: The main attributes are identification number, name, type (rate-sensitive or not), value, rate of return, maturity date, etc.
3. *Revenues*: The main attributes are date, revenues, description, etc.
4. *Expenditures*: The main attributes are date, expenditures, descriptions, etc.
5. *Interest Rate*: The main attributes are type (short-term Treasury bill rate, long-term Treasury bill rate, LIBOR and commercial paper rates), value, date, etc.

Note that an institution has a number of assets and liabilities.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create a query that lists the daily market interest rates during the last three years. This query will enable us to catch trends on the interest rates.
2. Create a query that lists for each asset and liability the time until maturity.
3. Create a query that prompts for an interest rate and returns the present market value of the assets and liabilities of this database.
4. Create a query that calculates the total market value of the assets and liabilities of this database.
5. Create a query that calculates the interest earned on each rate-sensitive asset during the last trimester. The interest earned (IE) is calculated as follows: $IE = (AI)/100$, where A is the amount of the rate-sensitive assets at the end of the trimester and I is the interest rate on the rate-sensitive assets in the end of the trimester.
6. Calculate the rate-sensitive asset gap and liability gap for this institution. The sum of all the IE provides the rate-sensitive asset gap for this institution during the selected trimester. In a similar way, calculate the rate-sensitive liability gap during the selected trimester.
7. Create a query that would present the gap position of the financial institution. The gap position for the selected trimester is calculated as the difference between the rate-sensitive asset gap and rate-sensitive liability gap.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: assets, liabilities, interest rates, etc. These forms allow the user to add, update, and delete information about assets, liabilities, interest rates, etc.
3. Create a form that allows the user to browse through the information about the assets owned by this financial institution. For each asset, present the following: type, value, rate of return, maturity date, etc. Insert a textbox that presents the interest earned (IE) on this asset. Insert another textbox that presents the rate-sensitive asset gap for this institution.
4. Create a form that allows the user to choose a liability from a combo box. Create a subform that presents the following information about the selected liability: type (rate-sensitive or not), value, rate of return, maturity date, etc. Insert a textbox that presents the interest earned on this liability. Insert a textbox that presents the rate-sensitive liability gap and another textbox that presents the gap position of this financial institution.

Design a logo for this database. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

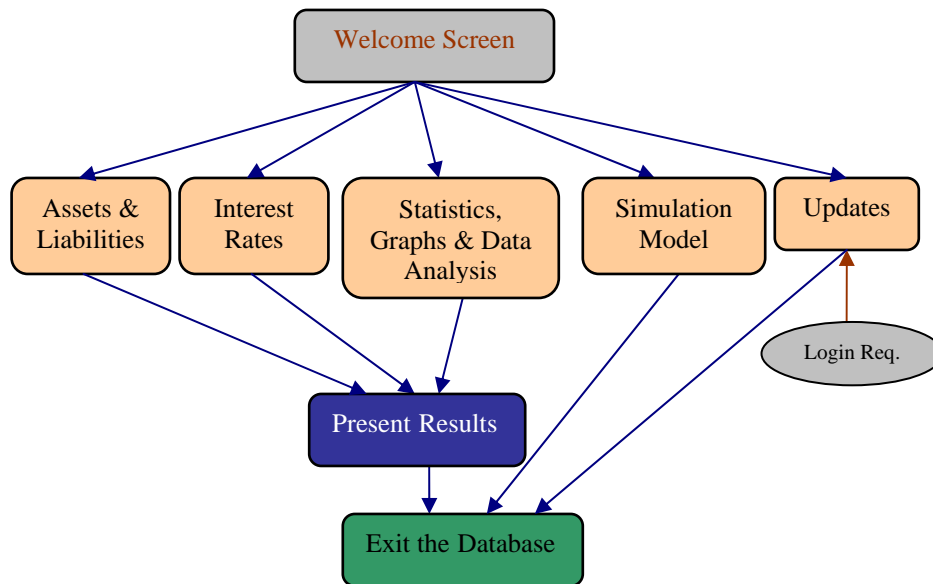
1. Create a summary report about the assets of this financial institution. For each asset, present the following: name, rate of return, maturity date, value, etc.
2. Create a summary report about the liabilities of this financial institution. For each liability, present the following: name, rate of return, maturity date, value, etc.
3. Create a summary report for the company. The report should include all the assets and liabilities of this institution. For each asset and liability, present the interest earned in the last trimester. The report should also present the following: rate-sensitive asset and liability gap as well as the gap position of this financial institution in the last trimester.
4. Use the chart wizard to plot the following:
 - a. Daily market interest rate during the last year.
 - b. Revenues generated per month during the last year.
 - c. Monthly expenses during the last year.
 - d. Interest earned in the last trimester in each asset owned by this institution.
 - e. Interest earned in the last trimester in each liability owned by this institution.
 - f. The value of the assets owned per month during the last year.
 - g. The value of the liabilities per month during the last year.

Visual Basic.NET Application Development

This database application can be used by the employees and managers of the financial institution, etc. In the following figure we present a tentative layout of the system.

In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find important. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Assets & Liabilities: The user can browse the database to learn more about the assets and liabilities of this financial institution. Information on interest earnings of a particular rate-sensitive asset and liability and gap position of the institution can be found in here.



Interest Rates: This part of the database keeps information about market interest rates. A user can check this part of the database to learn more about trends in market interest rates.

Statistics, Graphs, and Data Analysis: Managers use statistics, graphs, and data analysis tools to support their decisions. The plots built in the Access Application Development part can be included in here.

Simulation model: The objective of the “what-if” type of analysis is to provide the managers with strategies that would reduce the risk from changes in interest rate. The returns from different assets and liabilities are functions of interest rates. Therefore, as the interest rate changes, the return from these investments changes as well. The simulation system calculates the interest rate risk under different scenarios and provides a summary report to managers to assist in selecting a favorable alternative.

The following are the main steps of the simulation model:

- a. The first step in this model is identifying a market interest rate for a cycle (say, a month). We analyze the data on interest rates to identify its distribution. We then randomly generate the interest rates from this distribution.
- b. Assign the interest rate of unsecured credit loans equal to the average interest rate (generated in step (a)) for the cycle plus 7%.
- c. Assign the interest rate of real estate loans to the average interest rate (generated in step (a)) for the cycle plus 3.4%.
- d. The interest rate for the rest of the rate-sensitive assets and liabilities is estimated using the TREND function. This is a function that is provided by Microsoft Excel.

The syntax of this function is as follows:

$$\text{New_value_of_y} = \text{TREND}(\text{known_value_of_y}, \text{known_value_of_x}, \text{new_value_of_x}, \text{const})$$

For our purpose, values of x variables correspond to assets and liabilities of local financial institutions, and values of y correspond to those of the company in consideration.

- e. The historical data from the financial institution about the value of the assets and liabilities is used to estimate the value of the assets and liabilities in the current period of time using the TREND function in Microsoft Excel.
- f. Calculate the interest earned (IE) in the rate-sensitive assets and liabilities.
- g. Calculate the rate-sensitive asset gap by summing up the IE of the rate-sensitive assets.
- h. Calculate the rate-sensitive liability gap by summing up the IE of rate-sensitive liabilities.
- i. Calculate the gap position of the institution.
- j. The risk factor determines the accuracy of the decision support system. It is calculated as follows: $Error = (A - D)/A$, where A are the actual results from the database and D are the corresponding expectations. The product of these individual errors provides the system with the risk factor used to calculate the interest rate risk (IRR).
- k. Calculate $IRR = ((D - A)/A) - RF$, where RF is the risk factor. Managers are interested in maintaining a positive risk factor, which indicates that the institution will be able to keep the desired gap position.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about assets, liabilities, interest rates, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be an employee, a manager, the database administrator, etc. The managers need to access the database to retrieve information about certain assets and liabilities, run the simulation study, etc. Only the database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Reference

Moynihan, G.P. et al., "DSSALM: A decision support system for asset and liability management." *Decision Support Systems* 33: 23-38, 2002.

Problem Description

Personalized web pages are sites that present the user with only the news and advertisements of interest. Creating such websites is challenging because it is very difficult, in data mining, to scrutinize the news and advertisements by keyword and present personalized information to the user. Users' interests are generally obtained by asking questions about their preferences and by analyzing past browsing behaviors. The aim of this project is to build a database decision support system that allows classifying the news/advertisements on certain categories and then, depending on users' requests, presents the corresponding information. In fact, this system solves only part of the problem described above.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrity constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Advertisement*: The main attributes are name, description, category, targeted age group, targeted gender, etc.
2. *Category*: The main attributes are name, description, etc.
3. *News*: The main attributes are name, category, date, description, keywords, etc.
4. *User*: The main attributes are identification number, login name, password, name, address, gender, age, profession, hobbies, personal preferences, etc.
5. *Website*: The main attributes are address, category, description, keywords, etc.

Note that the database keeps track of user behavior: the number of times a user read the news of a particular category, the number of times a user visited a particular website, the number of times a user accessed advertisements of a particular category, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following queries help to identify users who have particular preferences and/or particular characteristics:
 - a. Create a query that prompts for the users' gender and lists the names of the users of the particular gender.
 - b. Create a query that prompts for the users' age and returns the names of the users who are up to five years older or five years younger than the selected age.

- c. Create a query that prompts for the users' age and gender and returns the name of the users who are of the selected gender and age.
 - d. Create a query that prompts for the name of a hobby and returns the name of the users who have that hobby.
 - e. Create a query that prompts for the name of a profession and returns the name of the users who are specialized in the selected profession.
2. Create a query that prompts for the name of a news category and returns a list of the news in the selected category.
 3. Create a query that prompts for the name of an advertisement category and returns a list of the advertisements in the selected category.
 4. Create a query that prompts for a keyword and lists the name of the news that contains the selected keyword.
 5. Create a query that prompts for a date and returns a list of news published on the selected date.
 6. Create a query that presents for each news category a count of the users who are interested in that category.
 7. Create a query that presents for each news category a count of the news stored in the database in the particular category.
 8. Create a query that presents for each advertisement category a count of the users who are interested in that category.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: news, users, advertisements, news/advertisement categories, etc. These forms allow the user to add, update, and delete information about news, users, advertisements, news/advertisement categories, etc.
3. Create a form that allows the user to browse through the information kept in the database about each user. Create a list box that presents a list of the users' preferences. Create a subform that presents a list of the news that satisfies the preferences of the selected user. Create another subform that presents the advertisements that target the age group and gender of the selected user.
4. Create a search form. The form has a number of command buttons. When the user clicks-on a particular command button, a query runs. These queries search the database for the following: news of a particular category, advertisements of a particular category, users of a particular age group, etc.
5. Create a form that allows the user to choose a particular news category. Create a subform that lists the names of the users who are interested in this particular news category. Create a subform that lists the headlines of the most recent news (the news published in the last month) of this particular category.
6. Create a form that allows the user to choose a particular advertisement category. Create a subform that lists the names of the users who are interested in this particular news category. Create another subform that presents the most recent advertisements of this category.

Design a logo for this database. Insert this logo in the forms created above. Have the background color of the forms light green and the border color for the titles yellow. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

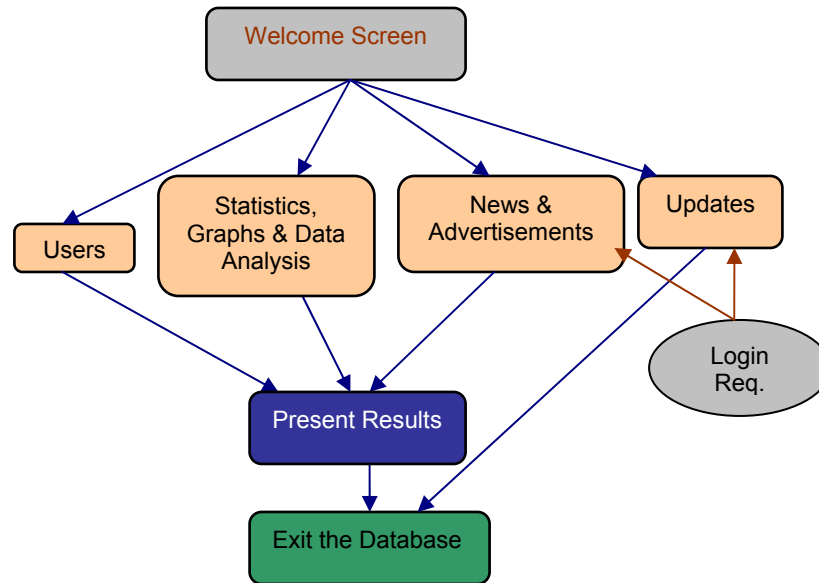
1. Create a summary report with information about the users of this database. The report should include users' name, login name, address, gender, etc.
2. This summary report should include the following statistics that are of interest for the company: total number of users, total number of female users, total number of male users, total number of teenage users, total number of adult users, total number of users who browse each news category, total number of users who browse each advertisement category, etc.
3. Report the latest news that has been the most popular among the users.
4. Report the latest advertisements that have been the most popular among the users of a particular age group and gender. The following is a way to create age groups for this database: users up to 10 years old, 10-14 years old, 14-19 years old, etc. The report should present the most popular advertisements for each age group and gender.
5. List the name of the most popular news category for each month during the last year. Also, list the name of the most popular advertisement for each month during the last year.

Visual Basic.NET Application Development

This database application can be used by the users, managers, the database administrator, etc. In the following figure we present a tentative layout of the system.

In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Users: This part of the database is used by (a) managers who are interested to learn about the users of this system and (b) users who want to update their personal information. Users need to have a login name and password to perform the update.



Statistics, Graphs & Data Analysis: Managers often visit this part of the database to identify trends in: (a) the total number of users, (b) the total number of users of a particular age group and gender; etc.

News & Advertisements: Users and managers visit this part of the database. Users are interested to read the latest news and advertisements. Users should use their login name and password. This would enable creating a personalized site for each user.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about users, news, advertisements, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a database user, a manager, the database administrator, etc. The user should have a login name and a password to be able to access the system. Another reason for using a login name and password is to enable creating personalized websites. The managers use the system to identify the following: (a) news and advertisement categories that are popular among users of a particular age group and gender; (b) trends on the total number of users; (c) trends on the total number of male/female users; etc. The database administrator can have access to the update forms. Users are allowed to update their personal information.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

The International Student office in a university is interested in developing a database to maintain information about the international students. The database will keep information about the courses the students have taken, their visa status, their on-campus jobs, etc.

The International Student office is interested in creating a new database to carry this information because there are certain rules and regulations that apply to international students only, and therefore using the university's database would be inconvenient. The following are some of the rules and regulations that are specific to international students: (a) international undergraduate students should take at least 12 credit hours per semester; (b) international graduate students should take at least 9 credit hours per semester; (c) the I-20 should not be expired; (d) international students can work up to 20 hours per week; (e) international students should be working on-campus. They would need special permission from USCIS (US Citizenship and Immigration Services) to work off campus.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrity constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *College*: The main attributes are college identification number, name, address, name and address of the dean, etc.
2. *Country*: The main attributes are name, language, capital city, ethnicity, etc.
3. *Course*: The main attributes are course identification number, name, number of credits, etc.
4. *Department*: The main attributes are department identification number, name, address, name and address of the department head, etc.
5. *Degree*: The main attributes are name, description, etc.
6. *Job*: The main attributes are job name, job type, hours/week, employer's name, employer's address, employer's telephone number, etc.
7. *Regulations and Laws*: The main attributes are identification number, name, description, etc.
8. *Student*: The main attributes are social security number, name, birthday, gender, nationality, address, enrollment date, type (graduate or undergraduate student), visa type, major department, college, degree sought, etc.

Note the following: (a) A student has an educational history. The history consists of the name of the schools attended, degree earned, GPA, etc; (b) The visa status for current students could be: F-1, J-1, etc. The visa status for students who have already graduated could be H-1, OPT (optional practical training), etc; (c) The relationship between student, degree, and college is a many-to-many relationship since a student may get a dual degree

or get minors from different departments; (d) The relationship between student and country is a many-to-many relationship since a student may have dual citizenship.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following queries help to retrieve information about undergraduate students:
 - a. List the social security number, name, address, major department, and college for all the undergraduate students.
 - b. Create a query that prompts for the name of a country and returns a list consisting of the social security numbers, names and addresses of the undergraduate students from this country.
 - c. Create a query that prompts for the name of a college and returns a list with the social security numbers and names of the undergraduate students enrolled in this college.
 - d. List the social security numbers, names, and nationalities of the undergraduate students holding a J-1 visa.
 - e. List the social security numbers, names, and addresses of undergraduate students who are currently working. For each student, report the number of working hours per week.
 - f. Create a query that prompts for the social security number of a student and returns that student's educational history (the name of the schools attended, degree earned, GPA, etc.).
2. Create the same queries as the ones described in part (1) to retrieve information about the international graduate students.
3. Present the name of the country that has the majority of the international students.
4. List the countries that have at least one representative (graduate or undergraduate student) in this university.
5. List the number of graduate and undergraduate students enrolled in this university each year for the last ten years.
6. Present the names of the college and department with the highest enrollment of international students.
7. Present the overall average GPA earned so far by undergraduate and graduate international students.
8. Present the average GPA earned so far by undergraduate and graduate international students. Group this information by country of citizenship.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: students, courses, departments, jobs, degrees, etc. These forms allow the user to add, update, and delete information about students, courses, departments, jobs, degrees, etc.

3. Create a form that presents academic information and contact information of the international students. The form should present the following academic information for each student: social security number, name, date of birth, gender, educational history, current education, and courses taken. The contact information consists of the following: student's address, e-mail, current phone number, permanent address and permanent phone number.
4. Create a form that presents detailed information about the courses that an international student is taking currently. Insert in this form a combo box to allow the user to choose the social security number of a student. Insert a textbox that presents the name of the selected student. Insert a subform that lists the identification number, name, and corresponding credit hours of the courses that the selected student is currently taking. Insert a textbox that presents the total number of credit hours the student is currently taking. In case that this number is fewer than 12 credit hours for an undergraduate student or fewer than 9 credit hours for a graduate student, display a message notifying the user that the student is enrolled for fewer than the required number of credits.
5. Create a form that presents information about the employment of an international student. Insert a combo box that allows the user to select the social security number of a student. Insert a textbox that presents the name of the selected student. Insert a subform that presents information about the current employment of the student. The subform should present the following: job type, employer's name, employer's telephone number, and employer's address. Insert a textbox that presents the total number of working hours per week.
6. Create a form that allows the user to browse through the regulations and laws related to international students. Create a subform that presents detailed descriptions of the selected regulation/law.

Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

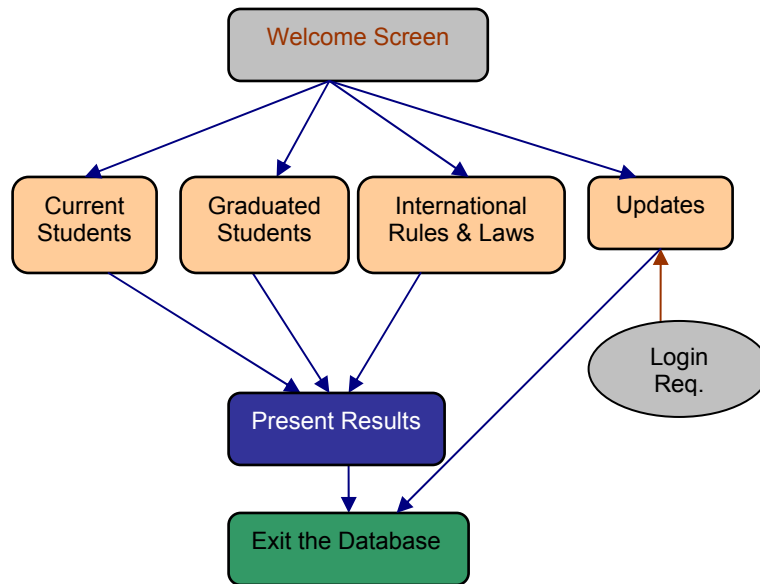
1. Report the following information for all the undergraduate international students: social security number, name, address, nationality, major department, college, GPA, and total number of credits earned so far. Group this information by nationality. Within each group, sort the information in descending order of credit hours earned.
2. Report the following information for all the graduate international students: social security number, name, address, nationality, major department, college, GPA, and total number of credits earned so far. Group this information by nationality. Within each group, sort the information in descending order of credit hours earned.
3. Report the following information about the international students who have already graduated: social security number, name, current visa type, job title, and the name and phone number of their current employer.
4. Report the following information about the students who will be graduating this coming spring: social security number, name, degree earned, name of the major department, college, enrollment date, graduation date, and GPA.
5. Report the following information about the students enrolled in the current semester: social security number, name, and a list of the courses the student is currently taking.

For each course, present the following: course identification number, name, name of the department that is offering the course, and the number of credits. Calculate the total number of credits each student is taking in the current semester.

6. Report the following information about the current employment of the international students: job title, employer's name, employer's address, and the total number of working hours per week. Group the information by students' social security number and name.
7. Report the following information about the visa status of each international student: visa type and starting date and expiration date of the visa. Group this information by student social security number and name.

Visual Basic.NET Application Development

This database application can be used by the employees of the International Student office, the database administrator, etc. In the following figure we present a tentative layout of the system.



Graduated Students: The user browses this part of the database to learn about the following: (a) academic information, (b) contact information, (c) visa status, and (d) current employment of the international students who have already graduated.

International Rules & Laws: The user browses this part of the database to learn about rules and laws concerning international students. For example, there are restrictions about the following: the total number of required credit hours per semester, the total number of working hours per semester, etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about students, departments, regulations, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be an International Student office employee, the database administrator, etc. The user

should have a login name and a password to be able to access the system. The International Student office employees use the system to identify: (a) the violations in the regulations; (b) trends in the total number of international students enrolled every year in the university; (c) trends in the performance of the students; etc. The database administrator can have access to the update forms. Users are allowed to update their personal information.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

The objective of this project is to create a database decision support system that could be used by pet clinics. This database keeps detailed information about clients, pets, employees, inventories, etc. The managers of a clinic can use this database to identify the following: products that have low inventory levels, trends in the number of clients who visit the clinic, trends in the types of treatments (services) provided, etc.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrity constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Client*: The main attributes are identification number, name, address, telephone number, e-mail, etc.
2. *Employee*: The main attributes are identification number, name, address, birthday, type (doctor or staff), hiring date, salary, specialty, degree, etc.
3. *Inventory*: The main attributes are identification number, type (medicine, equipment, etc.), description, cost, price, safety stock, quantity on-hand, etc.
4. *Pet*: The main attributes are identification number, name, type, birthday, etc.
5. *Service*: The main attributes are identification number, description, price, etc.

Note the following: (a) A pet has its own history. The following information is included in pets' history: visit date, comments, payment, etc; (b) During a visit a pet may see one or more doctors and use one or more of the services offered by the clinic; (c) When a pet is scheduled to visit a doctor, the following information is recorded: appointment date, description, etc; (d) *Pet* is a weak entity. It is a dependent of the *Client* entity type.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create the following queries that enable the user to retrieve useful information about pets registered in this database.
 - a. Create a query that prompts the user for the identification number of the client and returns a list of the pets owned by this client. For each pet, present the corresponding identification number, name, and birthday.
 - b. Create a query that prompts the user for the identification number of a client and the identification number of his/her pet and returns a detailed history of the visits with the clinic.

- c. Create a query that prompts for the name of a disease and returns a list with the name of the pets that have been diagnosed with this particular disease. Include in this list the name and identification number of the pet owner.
 - d. Create a query that prompts for a date and returns a list with the name of the pet(s) that has been scheduled to visit a doctor on that particular date. Include in the list the name and identification number of the pet's owner.
2. Create the following queries that enable the user to retrieve useful information about the doctors of the clinic:
 - a. Create a query that prompts for the name of a disease and returns a list with the name of the doctors who are qualified to treat this disease.
 - b. Create a query that prompts for the name of a doctor and returns a list of the pets that have visited the selected doctor. For each pet, present the following: name, name and identification number of its owner, and a description of the diagnosis.
 - c. Find the most popular doctor in the clinic.
 3. Create the following queries that enable the user to retrieve useful information about the inventories kept in the clinic:
 - a. Create a query that presents the quantity and value of on-hand inventory for each medicine.
 - b. Create a query that presents the name of the medicine whose on-hand inventory level has dropped below the safety stock level. For each medicine, also present the quantity on-hand and safety stock.
 - c. Create a query that presents the name and unit price of the five medicines that have the highest inventory turnover rate.
 - d. Create a query that presents the name and unit price of the five medicines that have the lowest inventory turnover rate.
 4. Create a query that presents the total earnings of the clinic during the current year. Hint: in calculating the earnings one should consider revenues from doctor visits, revenues from selling the medicines, costs of buying the medicines, revenues from using the equipments, costs of using the equipments, etc.
 5. List the five service (treatment) types that are the most popular.
 6. List the names and addresses of the clients who have a payment due in the current month.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: pets, clients, inventories, services, employees, etc. These forms allow the user to add, update, and delete information about pets, clients, inventories, services, employees, etc.
3. Create a form that allows the user to browse through the information about the doctors in this clinic. For each doctor, the form should present the following: identification number, name, contact information, degree, and specialty. Create a subform that presents a list of the pets that the doctor has visited. For each pet, present the following:

name, owner's identification number and name, diagnosis, etc. Create a command button that, when clicked-on, presents the name of the most popular doctor.

4. Create a form that allows the user to browse through the information about the clients. For each client, present the following: identification number, name, contact information, etc. Create a subform that lists the names of the pets owned by the selected client. For each pet, present the following: identification number, name, and type. Insert a command button that, when clicked-on, lists the appointments scheduled for this client. For each appointment, present the following: appointment date, name of the pet that will be visited, and name of the doctor.
5. Create a form that allows the user to learn about the medicines kept in the clinic. Insert a combo box that will allow the user to choose the identification number of a medicine. Insert text boxes that will present the following information about the selected medicine: name, unit cost, unit price, quantity on-hand, value of on-hand inventory, and safety stock level. Insert a subform that presents the following: identification number, name, and safety stock level of the medicines that are out of stock. Insert a command button that, when clicked-on, returns the identification number and name of the five medicines that have the highest inventory turnover rate.

Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create the following financial reports:
 - a. Create a report that prompts for a particular date and returns a list of the charges made to clients on that particular date as well as the payments made by clients. Include the following in this report: client identification number, client name, pet name (if applicable), diagnosis, and treatment provided.
 - b. Create a report that prompts for a particular month and returns all the financial transactions made during that month. The transactions include the following: amount of money received from doctor visits, medicines, and services provided to clients; amount of money paid for buying medicines, salaries paid, etc.
 - c. Create an annual report of the financial transactions of the clinic.
2. Use the chart wizard to plot the following:
 - a. Monthly consumption of the five most used medicines during the last year.
 - b. Monthly average on-hand inventory level for the five most used medicines during the last year.
 - c. Monthly revenues from doctor visits during the last year.
 - d. Total monthly earnings from the services offered to clients.
 - e. Annual earnings of the clinic during the last ten years.
3. Create the following reports that provide information about clients and pets:
 - a. Create a master report for the clients and pets of the clinic. The report should contain the following for each client: identification number, name, address, and identification number, name, and type of the client's pet(s).

- b. Report the name and contact information of the clients who have a payment due in the current month.
 - c. Report the names and contact information of the clients who have late payments.
 - d. Create a report that prompts for the identification number of a client and the name of that client's pet and returns the history of the pet's visits with the clinic.
4. Use the label wizard to create labels with the address of each client.
 5. Create the following reports that provide information about the doctors:
 - a. Create a master report that presents detailed information about the doctors in the clinic. The report should include the following for each doctor: identification number, name, contact information, degree, specialty, and years of experience. Group this information by specialty. Within each group, sort the information by years of experience.
 - b. Create a report that lists the appointments scheduled for the current day. For each appointment, present the following: client identification number and name, pet name and type, doctor name and specialty, diagnosis, etc.
 - c. Create a report that prompts for the name of a doctor and reports the name and diagnosis of the pets that doctor has visited.

Visual Basic.NET Application Development

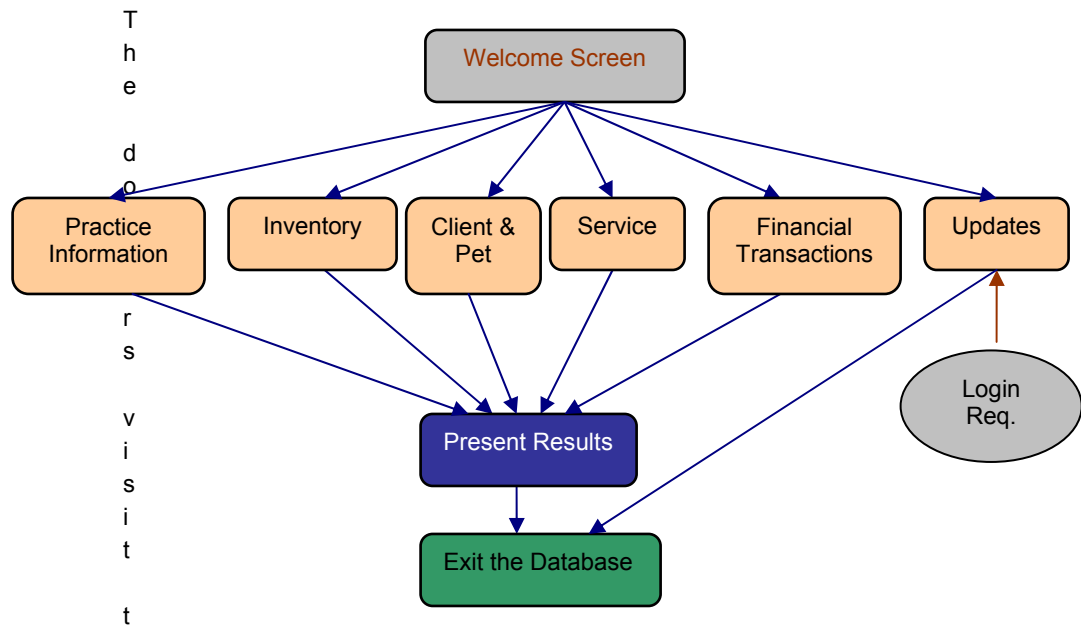
This database application can be used by the employees of the clinic (doctors and staff), the database administrator, etc. In the following figure we present a tentative layout of the system.

In the welcome screen, the user can choose one of the six options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Practice Information: This part of the database carries information about the doctors and the staff of the clinic.

Inventory: This part of the database carries information about the inventory of the medicines and equipments used in the clinic. The managers often visit this part of the database to identify medicines that are running out of stock, medicines that have the highest or lowest inventory turnover rates, etc.

Client & Pet: This part of the database carries information about the clients and pets of the clinic.



this part of the database to learn about the health history of the pets, the appointments scheduled for the current date, etc.

Service: This part of the database carries information about the services offered by the clinic. The user can find information about the type of services offered, the price charged, the employee responsible for performing a particular type of service, etc.

Financial Transactions: This part of the database keeps information about the financial transactions of the clinic. The managers visit this part of the database to learn about payments due, revenues generated, the clinic's expenses, annual earnings, etc.

Update: The update form requires an administrator login name and password. This form allows the user to add/delete/update the information kept in this database about pets, clients, employees, inventories, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a doctor or a staff member of the clinic, the database administrator, etc. The user should have a login name and a password to be able to access the system. The doctors use the system to identify (a) future appointments, (b) medical history of a pet, (c) availability of a particular medicine, etc. The clinic managers visit the database to identify (a) the on-hand inventory level of a particular medicine, (b) medicines that are out of stock, etc. The database administrator can have access to the update forms. Users are allowed to update their personal information.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

The purpose of this project is to develop an interactive information system for a CD warehouse. This system will enable the company to create an online ordering system interface (between the company and its customers) for everyday transactions.

The database that we will create has two kinds of end-users: customers and company employees. Customers have access to the company's web site and should be able to open accounts, do searches for the albums that they want to purchase, place orders, and check the current status of their account balances. While searching for an album, customers may know as little as the title of a hit song from the album to the name of the bassist of the group. Therefore, customers should be able to make searches by using one of the following: group name, album name, and song title. After finding the album that the customers want to buy, they can put it in their shopping cart, and at the end of the session they can place an order. At any time, the customers can check their order's status and their account balance.

The company's employees periodically check the database to identify the customers with positive balance and send a bill. They are also interested to know about customers' preferences. Then customers are informed about new releases according to their preferences. When an order is received, an employee checks the availability of the albums. If the albums are available, they are mailed to the customer right away, and the inventory level is updated. Customers are informed about the shipment date, and their balance is updated. If the albums are not available, the suppliers are contacted. The supplier with the lowest cost and closest distance to the warehouse is chosen. At the beginning of each month suppliers get a payment from the company. Products are shipped through UPS.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrity constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Album*: The main attributes are identification number, name, songs, members, release date, warehouse location, quantity available, category, etc.
2. *Customer*: The main attributes are identification number, name, address, telephone number, e-mail, credit card information, music preferences, password, balance, etc.
3. *Order*: The main attributes are identification number, order date, description, expected delivery date, actual delivery date, payment type, payment amount, payment date, etc.
4. *Supplier*: The main attributes are identification number, name, address, telephone number, current balance, etc.

Note the following: (a) *Music preferences* is a multi-valued attribute of the entity type *Customer*; (b) Whenever an album is purchased from a supplier, the following is recorded: purchase date, purchase price, quantity purchased. (c) When an album is ordered by a customer, the price is recorded. The price of an album changes with time. (d) The attribute

Songs of the entity type *Album* is a multi-valued attribute. (e) *Members* is a multi-valued attribute of the entity type *Album*.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following set of queries helps the management with cost analysis:
 - a. Create a query that calculates the average inventory level for each album. This query should also present the average inventory holding cost.
 - b. Create a query that calculates the quantity purchased per month per album and corresponding costs.
 - c. Create a query that calculates the total purchases made per month during the last year.
 - d. Create a query that calculates the overall monthly expenses (inventory holding costs and purchasing costs).
 - e. Create a query that calculates the revenues generated during the current month.
 - f. Create a query that calculates the revenues generated so far in the current year.
 - g. Create a query that prompts for the identification number of an album and returns its sales during the current month.
 - h. Create a query that presents the current month's profits.
2. The following queries help the managers with inventory management decisions:
 - a. Create a query that classifies the products into A, B, and C categories based on the principles of ABC analysis (20% of the products in the inventory, category A products, count for 80% of the value of the inventory).
 - b. Create a query that calculates for each album the Economic Order Quantity (EOQ).
 - c. Create a query that forecasts the next month's sales based on information about past months' sales.
3. Create a query that prompts for the identification number of a product and returns the current location of the product in the warehouse.
4. Create a query that identifies the top ten bestsellers of this CD warehouse.
5. Create a query that presents a list of albums released during the current month.
6. Create a query that prompts for the name of a song and returns the name and identification number of the album that has the selected song.
7. Create a query that prompts for the name of a singer and returns the songs sung by the selected singer and the name of the corresponding album.
8. Create a query that lists the names and addresses of all the customers who have a positive balance.
9. Create a query that lists the names and addresses of all the customers who have made late payments in the past.

10. Create a query that lists the names and addresses of the suppliers that have delayed their shipment at least once.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: albums, customers, orders, suppliers, etc. These forms allow the user to add, update, and delete information about albums, customers, orders, suppliers, etc.
3. Create a form that allows the user to browse through the information about the albums. The form presents the following: album identification number, title, release date, category, vocalist, etc. Insert a subform that presents the names of the songs of the selected album. Create another subform that presents the names of the members of the group that composed the selected album.
4. Create a form that allows the user to search for a particular album based on the following criteria: album name, name of the vocalist, name of a song, group name, etc. Insert a command button that, when clicked-on, returns the top ten bestsellers. Insert a command button that, when clicked-on, returns a list of the new releases.
5. Create a form that allows the user to browse through the information about the customers of the warehouse. The form should present the following: identification number, name, address, telephone number, current balance, etc. Insert a subform that presents all the orders placed so far by the selected customer. The orders should be sorted in a descending order of the delivery date. For each order, the subform should present the following: order identification number, order date, description, expected delivery date, actual delivery date, payment type, payment amount, and payment date.
6. Create a form that allows the user to browse through the information about the suppliers. Use textboxes to present the following: supplier identification number, name, address, telephone number, and current balance. Create a subform that presents a list of the orders sent so far to the selected supplier. Sort the orders in a descending order of the delivery date. For each order, present the following: order identification number, order date, description, expected delivery date, actual delivery date, payment type, payment amount, and payment date.
7. Create the following cost/revenue information form. One can use a tab control to manage the information provided in this form. The information about the current month's costs can be grouped together; the information about the current month's revenues can also be grouped together; etc. Use the queries about cost/revenue analysis we presented as a source of information needed to complete this form.

Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

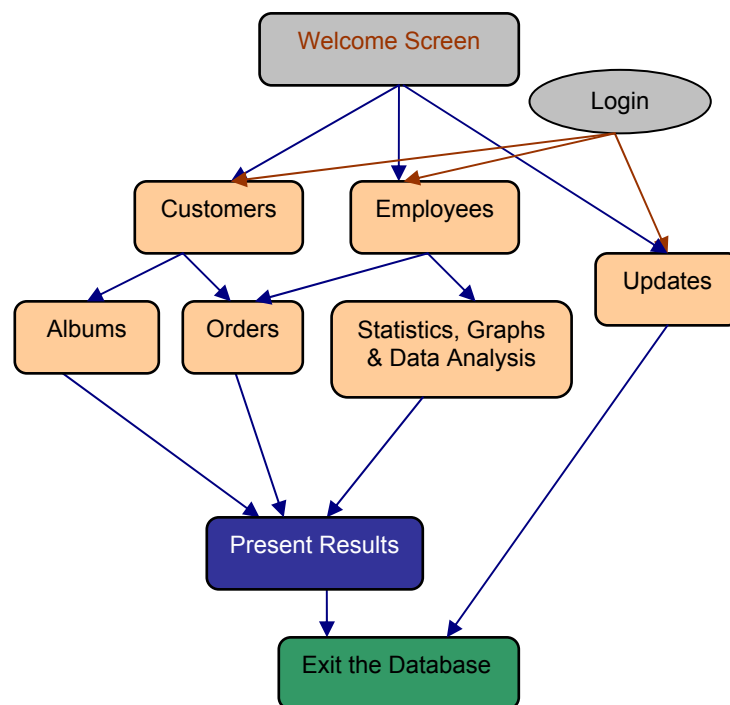
1. Use the chart wizard to plot the following:
 - a. Total sales per month during the last twelve months.
 - b. Total revenues per month during the last twelve months.
 - c. Total earnings per month during the last twelve months.

- d. Total sales, total revenues, and total earnings per month per album category (e.g., rap music, classical music, etc.) during the last twelve months.
 - e. Annual earnings during the last ten years.
 - f. Average monthly inventory level during the last twelve months.
 - g. Inventory holding costs per month in the last twelve months.
 - h. Inventory holding costs per month per album in the last twelve months.
2. Report detailed information about the customer orders placed during the current month.
 3. Report detailed information about the orders sent to suppliers during the current month.
 4. Report detailed information about delayed orders.
 5. Report detailed information about new album releases.
 6. Report detailed information about the top ten bestsellers.
 7. Use the label wizard to create labels containing the address of each customer and supplier.

Visual Basic.NET Application Development

This database application can be used by the employees and managers of the CD warehouse, the customers, the suppliers, the database administrator, etc. In the following figure we present a tentative layout of the system.

In the welcome screen, the user can choose one of the three options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.



Customers: The customers browse this part of the database to learn about new releases or bestsellers or to place an order.

Employees: The user browses this part of the database to identify trends in the sales, the costs, the revenues; identify new orders; identify late payments; identify late deliveries; forecast next month's sales based on data from past sales; identify the economical order quantity per album; etc.

Update: This form allows the user to add/delete/update the information kept in this database about albums, customers, suppliers, orders, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a customer, an employee, the database administrator, etc. The user should have a login name and a password to be able to access the system. The customers use the system to identify new releases, bestsellers, etc. The customers should be able to place and check the status of their orders on-line. The employees use the system in order to learn about new orders, check the current inventory level, place orders with the suppliers, etc. The database administrator can have access to the update forms. Users are allowed to update their personal information.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

Emerald Realty Group, Inc. is a local real estate company that is specialized in leasing office space to businesses. The company does not own all the properties it leases. The company has its own suppliers and customers. We refer to the person or the company that owns the office space as a supplier. A customer is the person or the company that rents the office space. Emerald Realty Group, Inc. facilitates the process of leasing properties to customers. For the services provided, the company gets royalties from the suppliers. The amount depends on the value of the property leased.

Currently, the company uses file folders to keep their business information. As the size of the business is growing, it is getting difficult to manage this information. Emerald Realty Group, Inc. is interested in building a database that will enable them to better manage the business.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Customers*: The main attributes are identification number, name, address, name and telephone number of the contact person, etc.
2. *Suppliers*: The main attributes are identification number, name, address, name and telephone number of the contact person, etc.
3. *Property*: The main attributes are identification number, name, address, price, size, descriptions, etc.

Note the following: (a) A property owned by a supplier is leased to a customer on a particular date, for a particular monthly rent, for a particular time period. In addition to this, details about the general liability insurance are recorded; (b) Customers have certain requirements about the property they will rent. The requirements are related to the size of the property, location, monthly rent, etc; (c) When a supplier asks for service, the following information is recorded: date, amount to be paid in royalties, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following set of queries enable the user to find properties that meet the requirements of a customer:
 - a. Create a query that prompts for the name of a city and returns a list of properties that are available.

- b. Create a query that returns a list of properties that are of a particular size.
 - c. Create a query that returns a list of properties that have a particular monthly rent.
 - d. Create a query that presents all the properties that are not leased yet.
 - e. Create a query that presents the five most expensive properties.
 - f. Create a query that presents the five largest properties.
2. The following set of queries presents important information about suppliers:
 - a. Create a query that prompts for the identification number of a supplier and returns a list of the properties owned by this supplier.
 - b. Create a query that presents the five most preferable suppliers. Preferred suppliers are the ones who have paid the largest amount of royalties to the company in the last two years.
 3. Create a query that lists all the customers who are looking for properties of certain value, size, or location.
 4. Create a query that prompts for the identification number of a property and returns the name and address of its owner and the name and address of the customer who is renting it.
 5. The following set of queries are required for financial purposes:
 - a. Create a query that presents the royalties collected during the current month from properties leased.
 - b. Create a query that presents the royalties collected during the current year.
 - c. Create a query that presents the royalties collected during the current year. Group this information by supplier.
 - d. Create a query that presents the royalties collected during the current year. Group this information by location.
 - e. Create a query that presents the royalties collected during the current year. Group this information by property size.
 - f. Create a query that presents the royalties collected per month during the current year.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: suppliers, customers, properties, leases, etc. These forms allow the user to add, update, and delete information about suppliers, customers, properties, leases, etc.
3. Create a form that allows the user to browse through the information about the customers of the company. The form should only present the customers who have not been served yet (who still need a property to lease). For each customer, present the following information: name and address, contact information, and specifications (size, location, value, etc.) about the property needed. Insert a subform that presents a list of properties that satisfy the conditions specified by the customer.
4. Create a subform that allows the user to browse through the information kept in the database about each supplier. For each supplier, present the following: name, address, and contact information. Create a subform that presents a list of properties owned by

the selected supplier that are not yet leased. Insert a textbox that presents the total amount of money collected in royalties during the current year by the selected supplier.

5. Create a form that provides detailed information about the properties leased during a certain time period. The user should be able to select the starting time and length of the period. For each property, present the following: name, address, size, monthly rent, and details about insurance; name, address, and contact information of the supplier; name, address, and contact information of the customer.

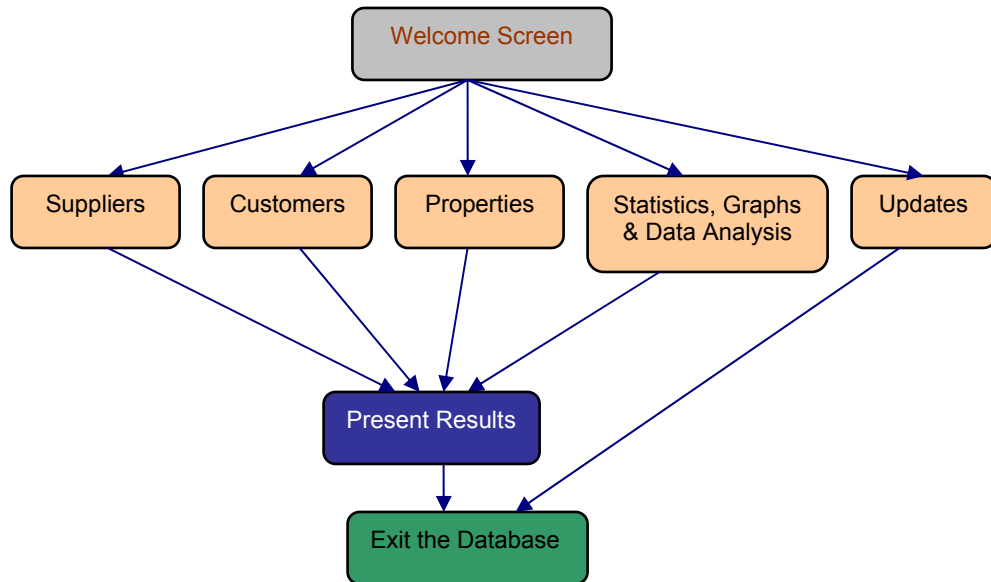
Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create a report that summarizes the financial activities of the company during the current month. The report should contain details about the royalties received.
2. Create a summary report of the financial activities of the current year.
3. Use the label wizard to create labels with the addresses of the customers and suppliers.
4. Use the chart wizard to plot the following:
 - a. Total royalties received per month during the last twelve months.
 - b. Total royalties received per year during the last five years.
 - c. Total number of properties leased per month during the last twelve months.
 - d. Total number of properties leased per year during the last five years.
 - e. Total number of new customers who have registered with the company per month in the last twelve months.
 - f. Annual increase in the number of customers during the last five years.
 - g. Annual increase in the number of suppliers during the last five years.
5. Report the names, identification numbers, and addresses of the five most preferable customers/suppliers.
6. Report details about the following properties that are waiting to be leased:
 - a. The five most expensive properties.
 - b. The five largest properties.
 - c. The five properties that have the best location.
7. Create a report that contains detailed information about the properties leased in the current year and parties involved.

Visual Basic.NET Application Development

This database application can be used by the employees and managers of the company, the customers, the suppliers, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Suppliers: This part of the database keeps information about the suppliers.

Customers: This part of the database keeps information about the customers.

Properties: Users should be able to identify properties that have certain characteristics, view properties that have been leased or are waiting to be leased, etc.

Statistics, Graphs & Data Analysis: The managers browse this part of the database to identify trends in the following: the revenues generated, number of properties leased, number of customers, etc.

Update: This form allows the user to add/delete/update the information kept in this database about properties, customers, suppliers, leases, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a customer, a supplier, a manager, the database administrator, etc. The user should have a login name and a password to be able to access the system. The customers use the system to identify properties that match their requests. The suppliers use the system to learn about the status of their properties. The employees use the system to do the following: identify trends on the royalties collected, total number of customers, etc; learn about the properties that are listed to be leased; etc. The database administrator can have access to the update forms. Users are allowed to update their personal information.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

Rutgers Transit is a passenger transportation company. They are located in Cleveland and provide several bus lines that go to a number of destinations. The company currently has about 1,000 buses spread over 60 routes. Of these, 700 are regular, 200 are semi-luxury, and the remaining are super deluxe buses. The seating capacities are 48, 42, and 36, respectively. Receiving reservations in advance facilitates the planning process. For this reason, the company is encouraging reservations of at least one week in advance by offering better prices.

As the business grew, the company faced difficulties in managing passenger reservations. The managers decided to computerize the reservation system. For this purpose they built a database that keeps detailed records of the buses owned, routes offered, passenger reservations, etc. They are now in the process of building a database decision support system that will facilitate the process of managing passenger reservations.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Bus*: The main attributes are identification number, brand name, capacity, category, descriptions, unit cost, etc.
2. *Passenger*: The main attributes are identification number, name, address, e-mail, birthday, current balance, etc.
3. *Route*: The main attributes are identification number, origin, destination, starting time and day (of the week), arrival time and day, duration, price, etc.
4. *Trip*: The main attributes are identification number, origin, destination, starting time and day, arrival time and day, duration, price, etc.

Note the following: (a) When a passenger makes a reservation, the following information is recorded: number of seats reserved, special concession required, etc. If seats are available, seats of the passenger's choice are allotted. However, if seats are not available, the system provides alternatives to help the passenger make a decision. (b) A route consists of a number of trips. For example, the route Cleveland-Miami consists of the following trips: Cleveland-Cincinnati, Cincinnati-Atlanta, Atlanta-Jacksonville, Jacksonville-Orlando, and Orlando-Miami; (c) Passengers get a discount for reservations made at least one week in advance. (d) Students and seniors get a 50% discount. (e) Employees and their eligible family members may travel for free once a year for a maximum of 3,000 miles. (f) A passenger reserves as many seats as the number of family members that are traveling. The system should keep detailed information about the passenger's dependents.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following set of queries facilitates the passenger reservation process:
 - a. List the route numbers that make a stop at the destination requested by the passenger. This destination could be the final or an intermediate point of the route.
 - b. List the total number of available seats for a particular route on a particular date.
 - c. Present the earliest date on which a certain number of seats are available for a particular destination on a particular route.
 - d. Present the cost of travel by various bus categories (regular, semi-luxury, etc.) to a certain destination.
 - e. Present the cost of travel to a destination using different routes.
2. The following set of queries helps management identify which routes are the most profitable, which routes do not generate enough income to cover their expenses, etc.
 - a. Present for each route the average monthly revenues generated. Sort the information in descending order of average revenues.
 - b. Present for each bus the average monthly revenues generated. Sort the information in descending order of average revenues.
 - c. Present the average number of passengers who travel in a particular route per week.
 - d. Present the total revenues generated per month during the last twelve months.
3. The management is trying to decide whether there is a need to buy additional buses or re-distribute the current fleet of buses on the existing routes.
 - a. Present for each bus the percentage of capacity used on average per month.
 - b. Present for each route the percentage of bus capacity used on average per month.
 - c. Present for each month (or week) the total number of passengers traveling. This query helps in identifying the busiest time periods.
 - d. Present for each route (during the busiest month) the total number of passengers traveling.
4. The following queries provide information about the demographics of the passengers who have used the service:
 - a. Present the total number of students who have used this service.
 - b. Present the total number of seniors who have used this service.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: passengers, buses, routes, trips, reservations, etc. These forms allow the user to add, update, and delete information about passengers, buses, routes, trips, reservations, etc.

3. Create a form that allows the user to select the origin and the destination for a trip. Insert a subform that presents the routes that pass by the selected cities. Insert command buttons that, when clicked-on, run the queries 1.a, 1.b, ..., 1.e. The information from these queries facilitates the reservation process.
4. Create a form that allows the user to browse through the information about the buses owned by the company. For each bus, present the following: identification number, brand name, capacity, and category. Insert a subform that presents a list of the routes run by the selected bus. For each route, present the following: identification number, city of origin and destination, start and arrival time and day, and the average percentage of the bus capacity used.
5. Create a form that allows the user to browse through the routes run by the company. The form presents detailed information about each route. Insert a subform that presents a list of the trips included on the selected route. For each trip, present the following: city of origin and destination, start time and day, arrival time and day, price, and total number of seats available.

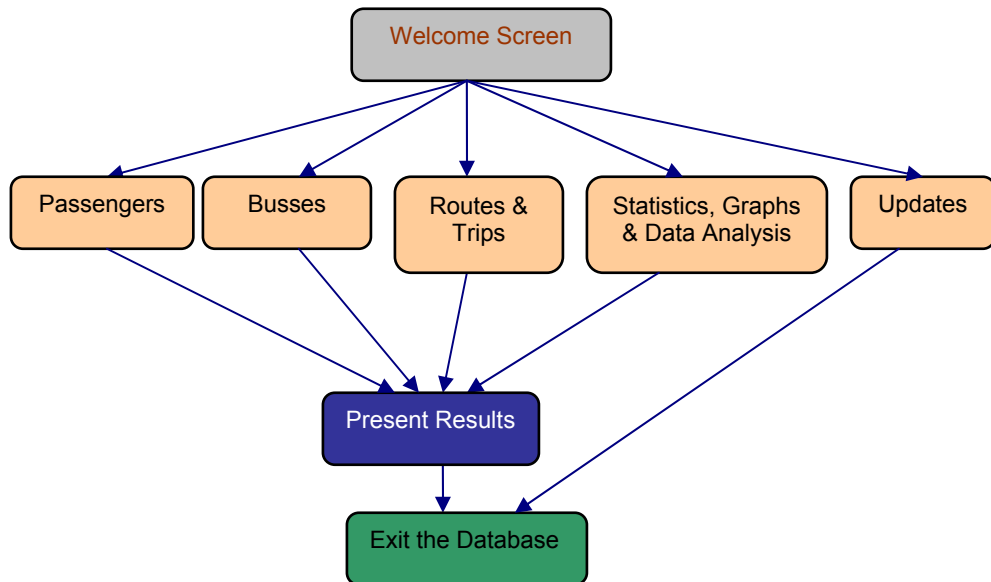
Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. The following set of reports helps with the financial analysis of the company:
 - a. Create an annual summary report of the financial activities of the company. The report should provide detailed information about the revenues generated from passenger reservations.
 - b. Create a monthly summary report of financial activities.
 - c. Present average revenues generated per month/year for each bus category.
 - d. Present average revenues generated per month/year for each route.
2. Use the chart wizard to plot the following:
 - a. Total number of passengers per month during the last twelve months.
 - b. Total number of passengers per month per bus category during the last twelve months.
 - c. Total number of seniors who have traveled per month during the last twelve months.
 - d. Total monthly revenues during the last twelve months.
 - e. Annual revenues during the last five years.
3. List the five most profitable routes.
4. List the routes that have had the least number of passengers.
5. Provide detailed information about the ten most preferable customers.

Visual Basic.NET Application Development

This database application can be used by the employees and managers of the company, the passengers, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Passengers: This part of the database keeps information about the passengers who have used the service of this company.

Buses: This part of the database keeps information about the buses used.

Routes & Trips: Users should be able to browse this part of the database to identify the route that best satisfies the requirements of the passengers.

Statistics, Graphs & Data Analysis: The managers browse this part of the database to identify trends in the following: the revenues generated, total number of passengers, total number of buses per category, etc.

Update: This form allows the user to add/delete/update the information kept in this database about buses, passengers, routes, reservations, trips, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a passenger, an employee or manager, the database administrator, etc. The user should have a login name and a password to be able to access the system. The passengers use the system to identify routes that fit their requests. The employees use the system to do the following: identify trends in the revenues generated, total number of passengers who have used the system, etc; identify needs for new buses; decide about bus re-scheduling; etc. The database administrator can have access to the update forms. Users are allowed to update their personal information.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

The aim of this project is to create a decision support system for a financial institution. The institution raises funds by issuing shares and debentures; accepting deposits from the general public/institutions; etc. These funds are then used to subsidize investments. The company is interested in computerizing these operations. Below we present details about the financial instruments used by the institution.

Fixed deposits are investments that have a fixed duration. The terms of the investment may be for one, two, or three years. The investor can choose to either collect the interest from the investment periodically and the principal at maturity (Scheme 1) or collect the interest and principal at maturity (Scheme 2). In the case that investors select Scheme 1, they may choose to get interest payments four times, twice, or once a year. Fixed deposits can be renewed at maturity, and only deposits in multiples of \$1,000 are accepted. Table 1 presents the interest rates paid for fixed deposits and the amount of money paid at maturity under Scheme 2.

Table 1: Fixed Deposits.

Category	Period of Deposit (years)					
	Interest Rate			Amount Paid at Maturity (per \$1,000)		
	1	2	3	1	2	3
General Public	5.00	5.50	6.00	1,060	1,130	1,180
Employees	5.50	6.00	6.50	1,080	1,180	1,240
Institutions	5.25	5.75	6.00	1,070	1,160	1,200

Other financial instruments used by the institution are debentures. Debentures are issued for periods of one, two, or three years. Debentures could be either convertible or non-convertible. Convertible debentures can be converted to equity shares on maturity. For the purpose of this project we assume that debentures pay an interest rate of 13%.

The institution also calls for shares periodically. Shares are classified as equity shares or preferred shares. The dividend to be paid to investors is declared at the end of the year. The institution lends the money to individuals or other institutions. The amount of interest charged depends on the amount of money borrowed and duration of the loan.

The decision support system should keep detailed information about the investors and investments. The system should help management identify trends in interest rates, dividend rates, different types of deposits, etc.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Borrower*. The main attributes are identification number, name, address, telephone number, e-mail address, type, current balance, etc.
2. *Investor*. The main attributes are identification number, name, address, telephone number, e-mail address, type, current balance, etc.
3. *Investment*. The main attributes are identification number, type, etc.

Note the following: (a) There are three main types of investments: fixed deposits, debentures, and shares; (b) There are three types of investors: general public, employees, and institutions; (c) There are two types of borrowers: individuals and institutions; (d) When an investor makes an investment, the following information is recorded: transaction identification number, date, amount invested, and descriptions.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following set of queries provides the necessary information for the annual financial analysis of the institution:
 - a. Present the total amount of money paid in dividends during the current year.
 - b. Present the total amount paid in interest from fixed accounts and debentures.
 - c. Present the total amount of money collected from investments in debentures, shares, and fixed deposits during the current year.
 - d. Present the total amount collected in interest payments from the money lent to individuals and companies during the current year.
 - e. Present the total amount of money lent to individuals and companies during the current year.
 - f. Present the total earnings of the institution during the current year.
2. Present the following information about the investors who have invested in preferred shares during the current year: identification number, name, and number of shares purchased.
3. Present for each investor the amount of money earned from dividends in the current year.
4. List the names, addresses, and contact information of the investors whose investments are close to maturity.
5. List the names, addresses, and contact information of the investors who have a balance of at least \$50,000.
6. Create a cross-tab query that presents for each investor the amount of money invested in different financial instruments (fixed deposits, debentures, and shares).
7. Create a query that presents the total amount of money invested in each financial instrument.
8. Create a query that presents the total number of investors who have invested in each financial instrument.

9. Create a query that calculates the total amount of money lent by the institution during the current year.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: investors, borrowers, transactions, investments etc. These forms allow the user to add, update, and delete information about investors, borrowers, transactions, investments, etc.
3. Create a form that allows the user to browse through the information recorded in this database about different financial instruments. For each instrument, present the following: identification number, name, description, and total amount of money invested in it. Insert a subform that presents a list with detailed information (identification number, name, address, and total amount of money invested) about the preferred investors of each instrument.
4. Create a form that allows the user to browse through the information recorded in this database about the investors. The information should include the following: investor's identification number, name, and address. Insert a subform that presents the following information about the investments made by the selected investor: investment identification number, amount invested, type of investment, and date of investment. Insert a textbox that presents the total amount of money invested so far, the interest earned, and the current balance in the investor's account.
5. Create a form that facilitates the process of managing the account of a particular investor. The following is a way to present this form. Include a number of command buttons. When one clicks-on a command button: (1) the user can terminate one of the investments; (2) the user can select an investment (that is at maturity) to re-invest; (3) the user can convert debentures to equity; etc.
6. Create a form that provides the user summary information of the accounts. This form consists of a number of command buttons. When one clicks-on a command button: (1) the user is prompted to enter an upper and lower limit. In return the user gets a list of investors who have a current balance within these limits; (2) the user can view all the deposits that are close to maturity; (3) the user gets a list of investors that should get an interest payment today; (4) the user gets a list of individuals and companies that have to make a payment today; etc.

Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

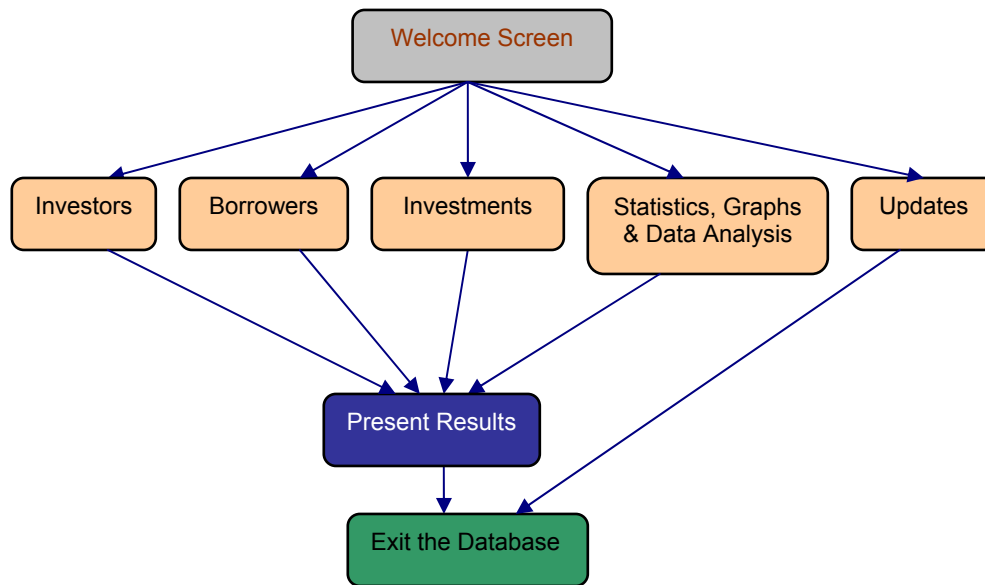
Reports

1. Create the following financial reports:
 - a. Report the following details about the investors who will receive a dividend payment at the end of the current year: identification number, name, address, current balance, amount invested in shares, and the expected amount of dividend to be paid during the current year.
 - b. Report detailed information about the amount invested in fixed deposits/debentures/shares during the current year.

- c. Report detailed information about the amount paid in interest on fixed deposits/debentures/shares during the current year.
 - d. Report detailed information about the amount lent to individuals and institutions during the current year.
 - e. Report detailed information about the amount collected during the current year on interest payments from individuals and institutions that have borrowed money from the company.
2. Use the label wizard to create labels with the address of each investor.
 3. Use the chart wizard to plot the following:
 - a. Total earnings per month during the last twelve months.
 - b. Annual earnings per year during the last five years.
 - c. Total number of accounts at the end of the year during the last five years.
 - d. Total number of new accounts opened each year in the last five years.
 - e. Total amount collected in interest (from borrowers) per year in the last five years.
 - f. Total amount of money paid in interest per month in the last twelve months.
 - g. Total amount of money paid in dividends per year in the last five years.
 4. Present identification number, name, address, and current balance of the following:
 - a. Investors who have an account balance higher than \$50,000.
 - b. Borrowers who have received at least \$50,000.
 - c. Investors who have fixed deposits close to maturity.
 - d. Investors who are about to receive an interest payment.
 - e. Borrowers who are about to make a payment.

Visual Basic.NET Application Development

This database application can be used by the employees and managers of the financial institution, the investors, the borrowers, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Investors: This part of the database keeps information about the investors. Investors can visit this part of the database in order to: (a) view and update their personal information; (b) view the current allocation of their funds; (c) re-allocate or re-invest their funds; etc.

Borrowers: This part of the database keeps information about the borrowers. Borrowers can visit this part of the database in order to: (a) view and update their personal information; (b) view the current balance on the account; etc.

Investments: This part of the database provides detailed information about investments. Managers browse this part of the database in order to identify: (a) trends on the profits earned, interest payments, interest collected, etc; (b) fixed deposits that are close to maturity; (c) investors who will receive an interest payment soon; etc. Investors visit this part of the database to identify new investment opportunities.

Statistics, Graphs & Data Analysis: The managers browse this part of the database to identify trends in profits earned, total number of accounts, etc.

Update: This form allows the user to add/delete/update the information kept in this database about investors, borrowers, investments, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be an investor, an employee or manager, the database administrator, etc. The user should have a login name and a password to be able to access the system. The investors use the system to update their personal information, view their account status, and identify investment opportunities. The employees use the system to identify trends in the profits earned, total number of accounts, etc. The database administrator can have access to the update forms.

Case Study 43 ■ Decision Support System for a Financial Institution

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Decision Support System for Taiwan's Federal Aviation Administration

Problem Description

The aim of this project is to create a database-based decision support system for Taiwan's Federal Aviation Administration (TFAA). TFAA is interested in keeping a database with detailed information about all the airline companies that offer passenger services in Taiwan, such as UniAir, TransAsia Airways, Far Eastern Transport, Great China Airlines, etc. This information is made accessible to all airlines in Taiwan with the intention of helping the companies assess their competitive position in the domestic market.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Airline*: The main attributes are identification number, name, address, name and telephone number of the contact person, total number of employees, total number of aircrafts, etc.
2. *Aircraft*: The main attributes are identification number, brand name, model, capacity, purchase price, purchase date, etc.
3. *Employee*: The main attributes are identification number, name, address, gender, birthday, employment history, qualifications, position with the company, salary, etc.
4. *Passenger*: The main attributes are identification number, name, address, birthday, gender, etc.
5. *Route*: The main attributes are identification number, origin, destination, airports visited, classification, distance, etc.

Note the following: (a) Routes are classified into domestic and international routes. (b) Different airlines fly the same routes; however, the frequency of flying a particular route, flying times, the cost per flight (including wages, maintenance, fees paid at the airports), time length of the flight, flight capacity, and size of the fleet vary by airline. (c) Airlines keep information about their ticket sale transactions. For each transaction, the following is recorded: transaction identification number, date, description, and amount of money received.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following set of queries helps identify the position of each company in the market:
 - a. Present the five most popular routes.
 - b. Present the five least popular routes.
 - c. List the flights that run on the most popular routes. For each flight, present the following: identification number, name of the airline company, origin, destination, start time, arrival time, duration, and price.
 - d. Present for each airline the total number of flights offered and total number of employees.
 - e. Present the average number of passengers that fly per month. Group this information by airlines.
 - f. Present the average revenues generated from passenger tickets per month. Group this information by airline.
 - g. Present for each airline the total number of aircrafts owned.
2. The following set of queries helps with financial analysis:
 - a. Create a query that presents the salary expenses per airline during the current year.
 - b. Create a query that presents the amount of money spent for buying aircrafts during the current year. Group this information by airlines.
 - c. Create a query that presents the annual revenues generated per airline during the current year.
 - d. Present the annual earnings per airline company.
3. Create a query that prompts for an origin and a destination and returns a list of corresponding flights. For each flight, present the following: identification number, airline, aircraft identification number, start time, arrival time, duration, price, and number of seats available.
4. Create a query that prompts for the identification number of a passenger and returns detailed information about that passenger's reservations.
5. Create a query that identifies the most preferred customers (passengers). Preferred customers are the ones who have spend at least \$2,000 in airline tickets during the current year.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: flights, passengers, routes, airlines, etc. These forms allow the user to add, update, and delete information about flights, passengers, routes, airlines, etc.
3. Create a form that allows the user to browse through the information saved in this database about the airlines. Use textboxes to present the following for each airline: identification number, name, address, contact information, total number of aircrafts, and total number of employees. Insert a subform that presents detailed information about the aircrafts owned by the selected airline. Insert another subform that presents detailed information about the routes run by the selected airline. Insert a command button that,

when clicked-on, presents a list with detailed information about the aircrafts purchased during the current month.

4. Create a form that allows the user to browse through the information saved in this database about the routes. Use textboxes to present the following information about the selected route: identification number, origin, and destination. Insert a subform that presents detailed information about the flights that run the selected route.
5. Create a form that allows the users to browse through the information saved in the database about flights. Insert a subform that presents detailed information about the reservations made on the selected flight. Insert a textbox that presents the average number of passengers on the selected flight. Insert a textbox that presents the average annual earnings on the selected flight. Insert a textbox that presents the busiest time period for the selected flight.
6. Create a form that allows the user to browse through the information saved in the database about the aircrafts. For each aircraft, present the following: identification number, brand name, model, capacity, purchase price and date, and the name of the airline that owns it. Insert a subform that presents detailed information about the flights that use the selected aircraft. Insert a textbox that presents the average percentage usage of the aircraft's capacity.

Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

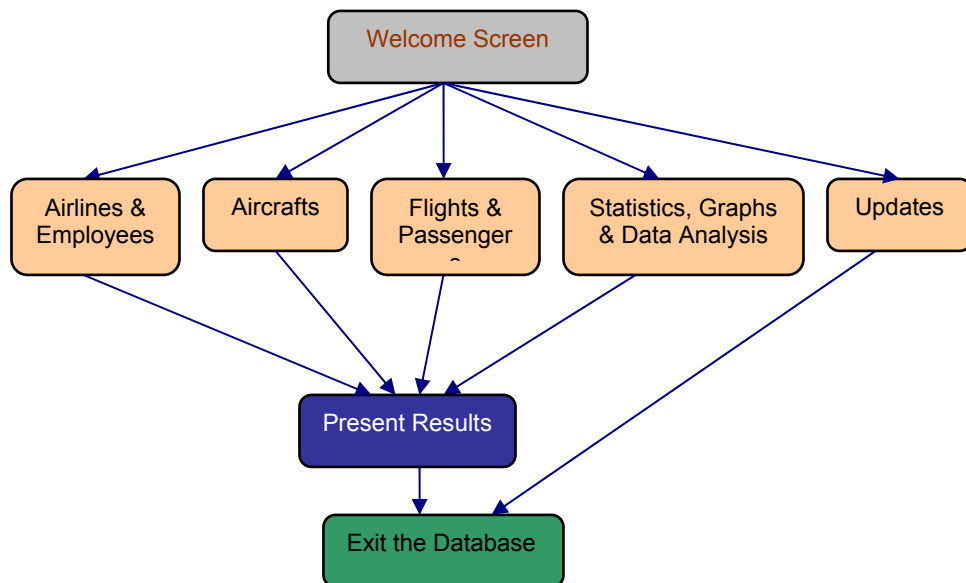
Reports:

1. Create the following financial summary reports:
 - a. Total revenues from airplane ticket sales per airline during the current year.
 - b. Total expenses from salaries and aircraft purchases per airline during the current year.
 - c. Total earnings per airline.
 - d. Total revenues per route during the current year.
 - e. Total revenues per aircraft during the current year.
 - f. Total revenues per flight during the current year.
 - g. Overall expenses, revenues, and earnings.
2. Use the chart wizard to plot the following:
 - a. Total monthly revenues per airline during the last twelve months.
 - b. Annual revenues per airline during the last five years.
 - c. Total number of passengers per route per month during the last twelve months.
 - d. Average percentage usage of capacity per aircraft per month during the last twelve months.
 - e. Average percentage usage of aircraft capacity per airline.
 - f. Average percentage usage of aircraft capacity per route.

- g. Average percentage usage of aircraft capacity per flight.
3. Report details about the most preferable and least preferable routes.
4. Report details about the most preferable and least preferable flights.
5. Report details about the most preferable and least preferable airline.
6. Create a report that prompts for a flight identification number and presents details about reservations made on that flight.
7. Use the label wizard to prepare labels with the address of each airline.
8. Report details about the aircrafts purchased during the current year.

Visual Basic.NET Application Development

This database application can be used by TFAA employees and managers, airline employees and managers, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Airlines & Employees: Users browse this part of the database to learn about airlines and their employees.

Aircrafts: Users browse this part of the database to learn about the specifications of the aircrafts used by different airlines, new aircrafts purchased, aircraft capacity usage, etc.

Flights and Passengers: Users browse this part of the database to make a reservation, compare prices and schedules, etc.

Statistics, Graphs & Data Analysis: Users browse this part of the database to identify trends in the following: the revenues generated by airline, total number of passengers per route, percentage usage of aircraft capacity, etc.

Update: This form allows the user to add/delete/update the information kept in this database about flights, airlines, routes, aircrafts, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a TFAA employee, an employee or a manager of an airline, the database administrator, etc. The user should have a login name and a password to be able to access the system. The TFAA employees use the system to identify the following: revenues generated by airline; total number of passengers by route, flight, and airline; new aircraft purchases; aircraft average capacity usage; etc. The airline employees use the system in order to learn about their position in the market, identify profitable routes, etc. The database administrator can have access to the update forms. Users are allowed to update their personal information.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

CRC is a local car rental company. Recently they have extended their activity by increasing the number of cars carried of different sizes and models. As their business is growing, CRC feels the need of better managing the information they carry about customers, cars, reservations, etc. They are interested in creating a database decision support system that would help with the following issues: identifying the number of cars by size and model to carry in their inventory, facilitating the process of making reservations, managing the data on reservations, etc.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Car*: The main attributes are identification number, model, size, purchase price, purchase date, etc.
2. *Customer*: The main attributes are identification number, name, address, telephone number, e-mail, birthday, gender, current balance, etc.
3. *Employee*: The main attributes are identification number, name, address, birthday, gender, education, position with the company, salary, etc.

Note that when a customer makes a reservation, the following is recorded: reservation date and time, pick-up date and time, return date and time, pick-up location, return location, payment amount, payment date, etc. The company keeps track of unsatisfied customer requests.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following set of queries helps with financial analysis:
 - a. Present the total revenues generated during the current month.
 - b. Present the total revenues generated by model during the current year.
 - c. Present the total number of new customers in the current month.
 - d. Present the total number of reservations by model during the current year.
 - e. Present the cars bought during the current year. For each car, present the following: identification number, model, size, and price.

- f. Present the total amount of money spent for buying new cars during the current year.
2. Create a query that prompts for a date and lists details about the pick-ups scheduled for that date.
3. Create a query that prompts for a date and lists details about the returns scheduled for that date.
4. Create a query that prompts for a date and returns the total number of cars available on that date. Group the information by model.
5. Present for each car model the average percentage use (e.g., Average % use of model X = (Total nr. of days being rented cars of model X)/(365*Total nr. of cars of model X)).
6. Calculate the average duration of a reservation. Group the information by model.
7. Find the five most preferable car models. Find the least preferable car model.
8. Present the average demand per month.
9. Find the busiest time period.
10. Present the average percentage use of the most preferable car models during the busiest period of the year.
11. Calculate the percentage of unsatisfied customer requests.
12. Create a query that prompts for an employee identification number and returns a list of reservations made by this employee.
13. Create a query that lists the names of the customers who have made at least one late payment.
14. Create a query that lists the identification number and name of the customers who have returned the rental cars late.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: cars, customers, reservations, etc. These forms allow the user to add, update, and delete information about cars, customers, reservations, etc.
3. Create a form that allows the user to select from a combo box a pick-up date. Insert a subform that presents a list of cars that are available on the selected date. For each car, present the following: identification number, model, production year, and daily (rental) rate.
4. Create a form that allows the user to browse through the information carried in this database about the cars. Insert a textbox that presents the average percentage use of the selected car. Insert a subform that presents the current reservations of the selected car. For each reservation, present the following: identification number; pick-up date and time; return date and time; identification number, name, and address of the customer who made this reservation; daily rate; payment amount; and payment date. Insert a command button that, when clicked-on, returns a list of the five most preferable car models.
5. Create a form that allows the user to browse through the information stored in this database about customers. Insert a subform that presents detailed information about

the reservations made by the selected customer. Highlight the reservations that have a late payment or a late return.

6. Create a form that enables users to access different financial reports. One could build such a form using command buttons. When the user clicks on a particular command button, one of the reports (1.a, 1.b, ..., 1.e) is generated.
7. Create a form that provides access to different statistical indicators of business performance. One could build such a form using command buttons. When the user clicks-on a particular command button, one of the reports (2.a, 2.b, ..., 2.f) is generated.

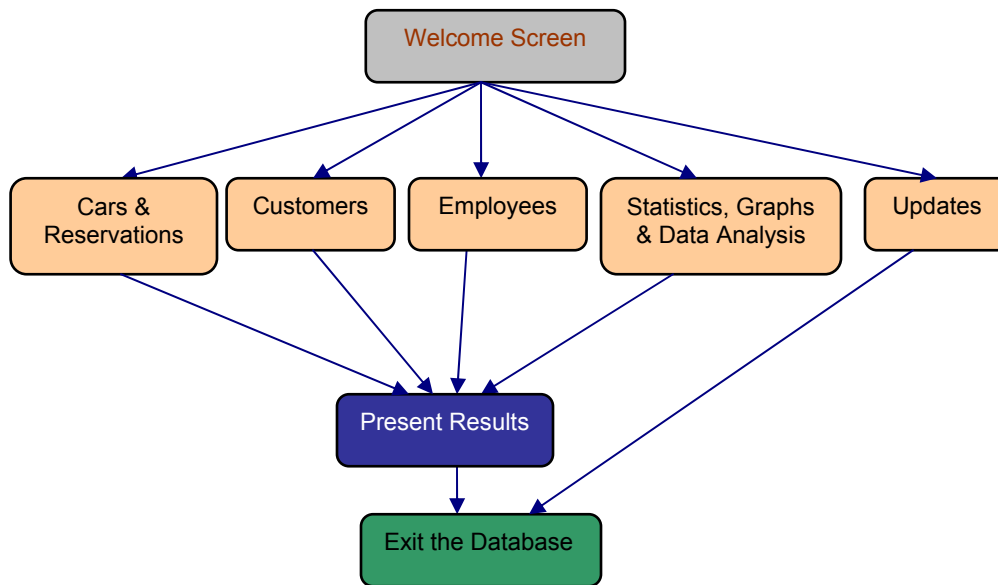
Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create the following financial summary report:
 - a. Report annual revenues generated from rentals. Group this information by car model.
 - b. Report annual expenses in purchasing new cars. Group this information by car model.
 - c. Report the total annual expenses in salaries.
 - d. Report the annual earnings of the company. Group this information by car model.
 - e. Report monthly earnings during the current year. Highlight the most profitable month.
2. Create the following reports that present important business performance indicators:
 - a. Report the total number of day-reservations per month during the current year.
 - b. Report the total number of day-reservations per car model during the current year.
 - c. Report the percentage use of each car model.
 - d. Report details about the unsatisfied customer requests during the current year. Group this information by date. Present the percentage of unsatisfied customer requests.
 - e. Report the total number of new customers per month during the current year.
 - f. Report the total number of customers per year during the last five years.
3. Use the label wizard to prepare labels with the address of each customer.

Visual Basic.NET Application Development

This database application can be used by the CRC employees and managers, customers, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Cars & Reservations: Users browse this part of the database to make a reservation, update a reservation, identify cars that are preferred by the customers, etc.

Customers: This part of the database presents information about customers.

Employees: This part of the database presents information about employees.

Statistics, Graphs & Data Analysis: Users browse this part of the database to identify trends in the revenues generated by model; total number of rentals per model; percentage usage of a car model; etc.

Update: This form allows the user to add/delete/update the information kept in this database about cars, customers, reservations, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a customer, an employee, or a manager of CRC, the database administrator, etc. The user should have a login name and a password to be able to access the system. The employees use the system to identify the following: revenues generated by car model; total number of rentals by model; percentage of unsatisfied customer requests; new car purchases; average percentage use of a particular model; etc. Customers use the system to make a reservation, update a reservation, update their personal information, etc. The database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

The Human Resources Department at Shands Hospital is concerned with managing the nursing staff. The department is facing challenges in deciding about hiring new employees, distributing employees into different departments, deciding about the FTE (full time equivalent) rates across departments, etc.

The Human Resource Department is interested in developing and implementing a human resource management system that would help with the following: maintaining data about employees, deciding about hiring new employees and transferring employees between departments, managing financial reports, etc.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Applicants*: The main attributes are identification number, name, address, telephone number, birthday, gender, employment history, degree, position applied for, etc.
2. *Clinic*: The main attributes are identification number, name, address, total number of employees, etc.
3. *Department*: The main attributes are identification number, name, address, description, total number of employees, etc.
4. *Employee*: The main attributes are identification number, name, address, telephone number, birthday, gender, employment history, degree, FTE, salary, etc.
5. *Position*: The main attributes are identification number, name, degree required, other qualifications, years of experience required, description of duties, etc.
6. *Request*: The main attributes are identification number, date, description, position, total number of employees to be hired, employee type (intern, part-time, or full-time employee), etc.

Note the following: (a) Each clinic consists of a number of departments; (b) *Employment history* is a multi-value attribute of the entity type *Employee*; (c) When an applicant is hired by a department (for a particular position), the following information is recorded: identification number, date of hire, employee type (intern, part-time, or full-time employee) etc; (d) Departments make requests about hiring employees; (e) Departments fire employees or an employee decides to leave.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. Create a query that presents the requests submitted by all the departments about hiring new employees. Group the information by department.
2. Present the total number of interns, full-time, and part-time employees hired during the current year. Group this information by department.
3. Create a query that prompts for the name of a department and returns a list with detailed information about the employees of the department.
4. Create a query that prompts for the name of a degree and lists detailed information about the employees who have earned that degree.
5. Create a query that prompts for the name of a position and lists detailed information about the applicants who have applied for that position.
6. Create an append query that enables the user to append the information about an applicant (who has been recently hired) to the table that has the information about the employees.
7. Present detailed information about the employees who are at least 60 years old. These employees potentially will be retiring soon. This information is important to plan for hiring new employees.
8. Present detailed information about the employees who were fired or left their jobs during the current year.
9. The following queries help with the financial analysis of the hospital:
 - a. Create a query that presents the total amount of money paid in salaries in each clinic.
 - b. Create a query that presents the average FTE by clinic.
 - c. Create a query that presents the total amount of money paid in salaries in each department.
 - d. Create a query that presents the average FTE by department. Sort this information by FTE.
 - e. Create a query that presents the average amount of money paid for a particular position.
 - f. Create a query that presents the additional amount of money paid in salaries during the current year because of hiring new employees.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: employees, applicants, position, requests, etc. These forms allow the user to add, update, and delete information about employees, applicants, position, requests, etc.
3. Create a form that allows the user to browse through the information kept in the database about requests submitted by different departments. Insert a subform that presents detailed information about the applicants who have the qualifications required by the selected request for employment.

4. Create a form that allows the user to browse through the information saved in this database about each department. Insert textboxes that present the following: the average FTE and the deviation of the selected department from this FTE, the clinic that the department belongs to, and the hospital. Insert a subform that presents a list of employees working in the selected department. For each employee, present the following: identification number, name, qualification, position, FTE, salary, and years of experience. Insert a subform that presents details about the requests submitted by the selected department.
5. Create a form that allows the user to browse through the details saved in the database about each position. Insert a subform that presents details about the applications received for the selected position. Insert a subform that presents details about the requests made from different departments about the position.
6. Create a report that allows the user to access different financial reports (reports 1.a, 1.b, ... ,1.d) prepared by the Human Resources Department.

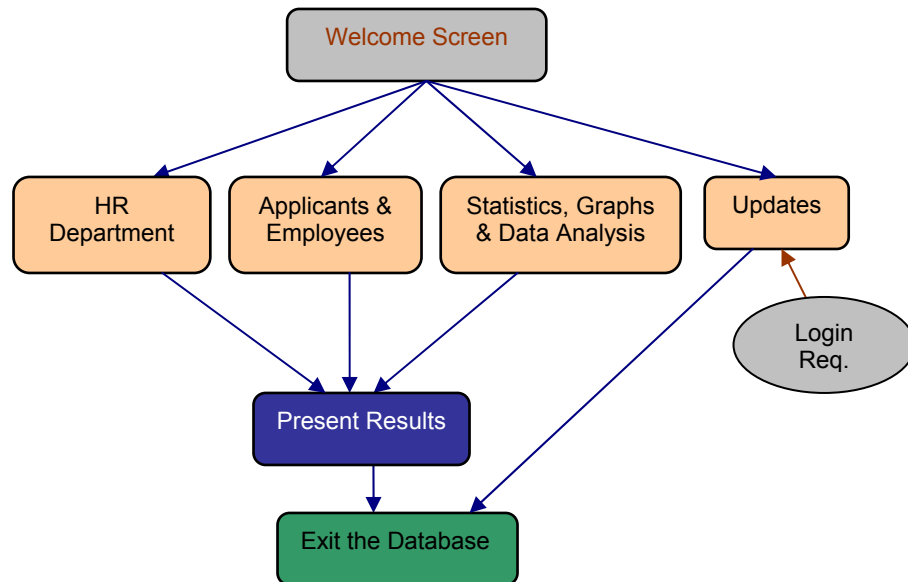
Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create the following financial reports:
 - a. Report the total annual increase in salary expenses from hiring new employees. Group this information by department.
 - b. Report the annual salary expenses for each clinic.
 - c. Report the deviation of each department, each clinic, and the hospital from the average FTE.
2. Use the chart wizard to plot the following:
 - a. Total number of applications for each available position.
 - b. Total number of employees hired per month during the last year.
 - c. Average FTE by department.
 - d. Average FTE by position.
 - e. For each clinic, plot the annual average FTE during the last five years.
 - f. Total number of new positions opened per year in the last five years.
3. Report details about the requests submitted by different departments. Group the information by department.
4. Report details about the positions filled during the current year.
5. Report details about the employees who were fired or quit their job during the current year.
6. Report details about: the most preferred position, the largest department, the best department to work for (the department with the least number of job resignations), and the largest clinic (in terms of total number of employees).

Visual Basic.NET Application Development

This database application can be used by the Human Resources Department employees, applicants, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

HR Department: Human Resources Department employees visit this part of the database in order to calculate average FTE, check financial reports, analyze departmental requests, post announcements about new positions, etc.

Customers: This part of the database presents information about customers.

Applicants & Employees: This part of the database presents information about employees and applicants.

Statistics, Graphs & Data Analysis: Users browse this part of the database to identify trends in the increase in annual salary expenses by department, the total number of employees hired during the current year, etc.

Update: This form allows the user to add/delete/update the information kept in this database about applicants, employees, positions, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be an applicant, an employee or a manager of the HR department, the database administrator, etc. The user should have a login name and a password to be able to access the system. The applicants access the database to learn about new job openings or to update their personal information. The managers use the system to identify the following: average FTE by department, total number of job openings in the departments, total number of employees hired, etc. The database administrator can have access to the update forms.

Case Study 46 ■ Human Resources Management at Shand's Hospital

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

“Clean-and-Go” is a local shop that provides the following services to the community: coin washers and dryers, drop-off services, and dry cleaning. This business has ten full-time employees. They currently rent their building.

The company is interested in building a database information system that would enable them to do the following: keep detailed information about daily collections, monthly expenses and revenues; manage employees’ schedules; manage equipment maintenance schedule; etc.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Cleaning Supplies*: The main attributes are name, description of usage, current inventory, safety stock level, etc.
2. *Customer*: The main attributes are identification number, name, address, e-mail, telephone number, credit card information, current balance, etc.
3. *Employee*: The main attributes are identification number, name, address, gender, date of employment, position hired for, schedule, etc.
4. *Equipment*: The main attributes are identification number, brand name, purchase date and purchase price, type, maintenance schedule, etc.
5. *Service*: The main attributes are identification number, name, description, rate charged, duration, etc.
6. *Supplier*: The main attributes are identification number, name, address, telephone number, current balance, etc.

Note the following: (a) *Schedule* is multi-value attribute of the entity type *Employee*. (b) *Maintenance schedule* is a multi-value attribute of the entity type *Equipment*. (c) The following items are classified as cleaning supplies: detergents, soaps, hangers, etc. (d) Customers pay for the services they get. When a customer is provided service, the following information is recorded: date, amount charged, description, satisfaction with the service, etc. (e) Suppliers offer equipments and cleaning supplies. When equipments or cleaning supplies are purchased, the following information is recorded: transaction identification number and date, quantity purchased, amount due, due date, description, delivery date, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following set of queries will help with the financial analysis of the company:
 - a. Create a query that calculates the annual expenses from equipment maintenance.
 - b. Create a query that calculates the annual expenses from cleaning supplies.
 - c. Create a query that presents the annual expenses from purchasing new equipment.
 - d. Create a query that calculates the annual expenses from salaries and rent.
 - e. Create a query that calculates the annual revenues from services provided to customers. Group this information by service type.
 - f. Create a query that presents the annual earnings of the company.
2. The following set of queries helps to analyze the progress of the business:
 - a. Create a query that presents the total number of new customers during the current year.
 - b. Create a query that presents the total number of service transactions during the current month.
 - c. Create a query that presents details about customer satisfaction with the service provided.
 - d. Create a query that presents the total number of equipment owned by the company. Group this information by equipment type.
3. Create a cross-tab query to present the weekly working schedule of the employees.
4. Create a cross-tab query to present the weekly maintenance schedule of the equipment.
5. The company has regulations about the amount of time (in terms of months) that equipment should be in use. Replacement time differs by equipment type. Create a query that presents the amount of time equipment has been used. Present the number of time period in terms of months.
6. Create a query that presents the average monthly usage per equipment (busy time/total time).
7. Create a query that presents details about the cleaning supplies for which the current inventory is below the safety stock level.
8. Create a query that presents details about the cleaning supply that has been used the most.
9. Create a query that presents details about the service that has been used the most.
10. Create a query that prompts for the name of a supplier and returns a list of products they offer.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: employees, services, equipments, cleaning supplies, etc. These forms allow the user to add, update, and delete information about employees, services, equipment, cleaning supplies, etc.

3. Create a form that allows the user to access different financial summary reports (reports 1.a, 1.b, ..., 1.d).
4. Create a form that allows the user to access different reports that provide indicators of business performance.
5. Create a form that allows the user to choose a particular date from a combo box. Insert a subform that presents details about customer related transactions performed in the selected date. Insert a textbox that presents the total revenues generated from these transactions.
6. Create a form that allows the user to browse through the information saved in this database about the equipment. Insert a subform that presents details about the maintenance services performed on the selected equipment. Insert a subform that presents details about the maintenances scheduled for the next month. Insert a command button that, when clicked-on, presents details about the activities (for example, if the equipment is a washing machine, present the total number of washing loads) performed on the selected equipment during the current month.
7. Create a form that allows the user to choose the name of a particular customer from a list box. Insert a subform that presents details about items the customer has given for dry-cleaning that need to be returned. Insert a textbox that presents the total amount due.
8. Create a form that allows the user to choose the identification number and name of a supplier from a combo-box. Insert a subform that presents details about the supplier and services/equipments purchased from the selected supplier. Insert a textbox that presents the amount of money due. Insert a command button that, when clicked-on presents details about the orders placed with the suppliers that have not been satisfied yet.

Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

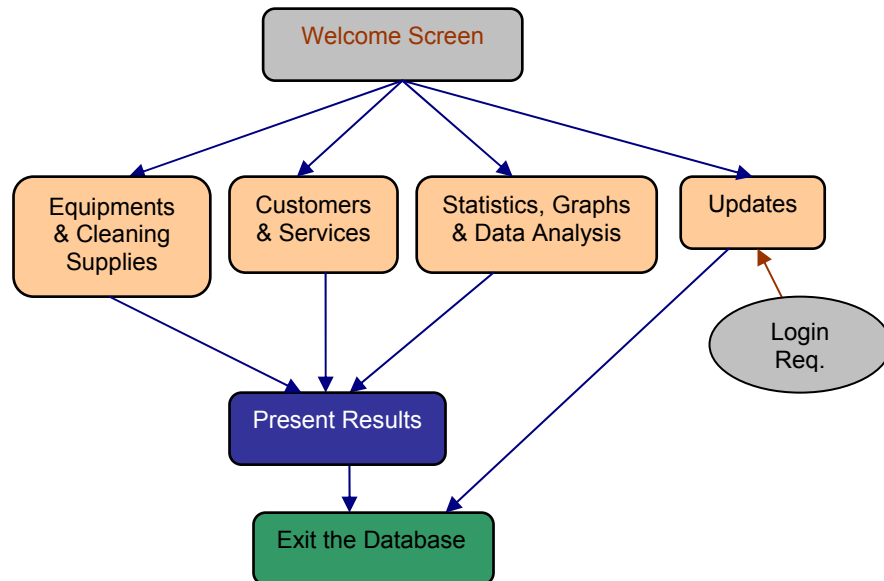
Reports

1. Create the following financial reports:
 - a. Present the annual revenues generated from services provided.
 - b. Present the annual expenses from the following: rent, salaries, equipment maintenance, purchasing new equipment, and cleaning supplies.
 - c. Present the annual earnings of the company.
 - d. Report the total expenses, total revenues, and total earnings during the current month.
2. Report the average daily usage, for each equipment, during the current year.
3. Report the maintenance schedule during the current month.
4. Report the weekly schedule of the employees.
5. Use the chart wizard to plot the following:
 - a. Total number of customers per year during the last five years.

- b. Monthly expenses for cleaning supplies during the last twelve months.
 - c. Total number of service transactions per month during the last twelve months.
 - d. Monthly expenses for maintenance during the last twelve months.
6. Report the most requested type of service.

Visual Basic.NET Application Development

This database application can be used by the employees, the manager, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the four options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Equipment & Cleaning Supplies: This part of the database keeps information about equipment and cleaning supplies. The employees browse the database to learn about the maintenance schedule of particular equipment, the inventory level of cleaning supplies, etc.

Customers & Services: This part of the database carries information about customers and services provided.

Statistics, Graphs & Data Analysis: Users browse this part of the database to identify trends in: the annual expenses, the annual revenues, the total number of new customers, etc.

Update: This form allows the user to add/delete/update the information kept in this database about equipment, services, customers, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a customer, an employee or a manager, the database administrator, etc. The user should have a login name and a password to be able to access the system. The customers

access the database to learn about the services offered and price charged or to update their personal information. The managers use the system to identify total expenses, total revenues, maintenance schedule, etc. The database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

A local movie theater is interested in building a database information system. The database will keep the information about the customers, sales, etc. The system will enable the management to analyze the progress of the business, create financial reports, etc.

An important aspect of this project is to develop a web application that will allow the customers to check the schedule of movies and purchase tickets on-line.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrity constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Customer*: The main attributes are identification number, name, address, telephone number, e-mail, etc.
2. *Employee*: The main attributes are identification number, name, address, date of hire, employment history, salary, etc.
3. *Movie*: The main attributes are identification number, title, year of production, awards won, description, actors, location, etc.
4. *Showroom*: The main attributes are location, name, capacity, etc.
5. *Producer*: The main attributes are identification number, name, address, contact information, current balance, etc.

Note the following: (a) *Awards won* and *Actors* are multi-value attributes of the entity type *Movie*. (b) *Employment history* is a multi-value attribute of the entity type *Employee*. (c) A movie is played in at least one showroom and it is played at least once a day. When a movie is assigned to a showroom, the following information is recorded: the timetable and the total number of tickets available per show. (d) When a customer purchases tickets for a movie, the following is recorded: ticket number, date, unit price, amount paid, and number of tickets purchased. The number of tickets purchased reduces the total number of tickets available. (e) Some of the customers choose to become members in order to buy tickets on-line. For members, the following additional information is required: login name and password. (f) When the movie theater purchases movies from a producer, the following information is recorded: transaction number, purchase price, purchase date, payment due date, and amount due.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following set of queries helps with financial analysis:
 - a. Create a query that presents the monthly revenues from ticket sales, the monthly expenses from salaries, the monthly expenses from purchasing new movies, and the monthly earnings.
 - b. Create a query that presents the total revenues, total expenses, and earnings during the current year.
2. Create a query that lists the five best movies of the current year. This classification is based on the number of awards won.
3. List the five most expensive movies of the current year.
4. Create a query that presents the total number of tickets sold per movie. Sort the information in descending order of the total number of tickets.
5. Create a query that presents the average capacity usage of a showroom during the current year.
6. Create a query that lists the one hundred most preferable customers.
7. Create a query that presents detailed information about the producer with whom the theater did the most business during the current year.
8. Create a query that prompts for a date and returns the movie schedule for the selected date and the total number of available tickets per show.
9. Create a query that prompts for the name of a movie and returns the weekly schedule of the selected movie and the total number of available tickets per show.
10. Create a query that prompts for the name of a customer and returns details about the ticket purchases made by the selected customer during the current month.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: employees, movies, customers, movie timetables, etc. These forms allow the user to add, update, and delete information about employees, movies, customers, movie timetables, etc.
3. Create a form that enables the user to browse through the financial reports created.
4. Create a form that would enable the user to select a movie title from a combo box. Insert a subform that presents the weekly timetable of the selected movie and the total number of available tickets per show. Insert textboxes to present the following information about the selected movie: production year, name of the producer, and a description of the movie. Insert a command button that, when clicked-on, returns a list with details about the awards won by the selected movie. Insert a command button that, when clicked-on, returns details about the actors who perform in the movie. Insert a command button that allows the user to purchase a ticket. When the user clicks-on this button (a) the user sign-in form opens. If the user is not a member, the user registration form opens. (b) Next, the data entry form opens to enable the user to purchase a ticket.
5. Create a form that allows the user to browse through the information saved in this database about actors. Insert a subform that presents details about the movies performed in by the selected actor. Insert a subform that presents details about the awards won by the selected actor.

6. Create a form that allows the user to browse through the information saved in this database about producers. Insert a subform that presents details about the movies purchased during the current year from the selected producer.
7. Create a form that allows the user to browse through the information saved in the database about customers. Insert a subform that presents details about the ticket purchases of the selected customer during the last month. Insert textboxes that present the following information: the total amount of money the selected customer has spent during the current month; the total revenues generated from ticket sales during the current month; and the total revenues from ticket sales during the current year. Insert a command button that, when clicked-on, returns details about the most preferred customers.

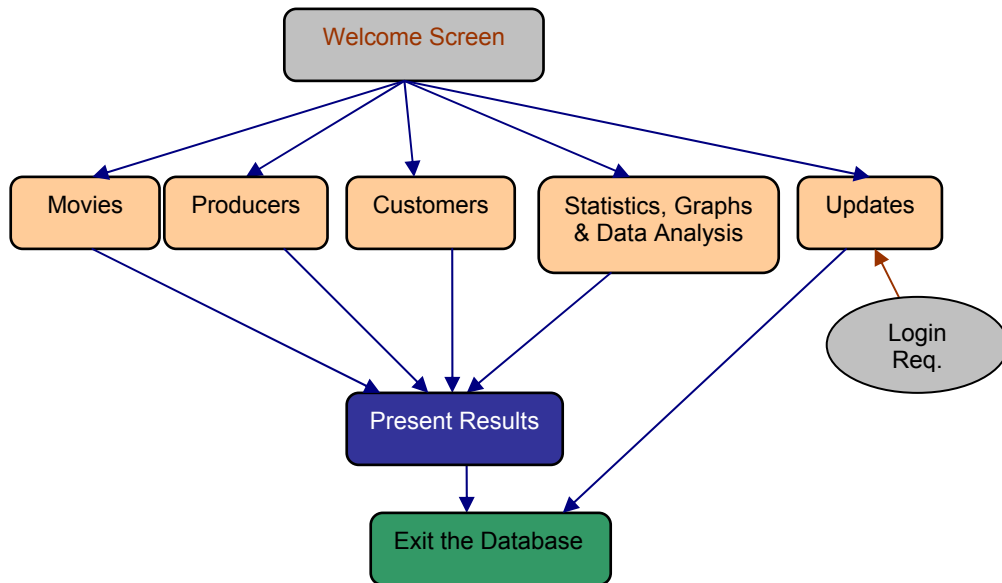
Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Create the following financial reports:
 - a. Report details about monthly expenses of the movie theater.
 - b. Report details about monthly revenues of the movie theater.
 - c. Create a summary report of annual expenses, revenues, and earnings of the movie theater.
 - d. Report details about the transactions performed during the current year with the movie producers. Group the information by producer name.
2. Use the chart wizard to plot the following:
 - a. The total number of tickets sold by movie.
 - b. The average monthly use of each showroom.
 - c. The total number of tickets sold per month during the last twelve months.
 - d. The total number of movies purchased from each producer.
 - e. The total number of new movies per month during the last twelve months.
 - f. The total number of new members per month during the last twelve months.
 - g. The monthly earnings during the last twelve months.
 - h. The annual earnings during the last five years.
 - i. The annual expenses from purchasing movies during the last five years.
3. Report details about the five producers with whom the theater did most of its business during the current year.
4. Report details about the one hundred most preferred customers.
5. Report details about the movies that won an award in the current year.
6. Report details about the actors who won an award in the current year.

Visual Basic.NET Application Development

This database application can be used by the employees and the managers of the movie theater, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Movies: This part of the database keeps information about movies shown in the theater. The customers browse the database to learn about the movies schedule, to learn about new movies, etc.

Producers: This part of the database presents information about the producers.

Customers: This part of the database presents information about the customers.

Statistics, Graphs & Data Analysis: Users browse this part of the database to identify trends in the annual expenses, the annual revenues, the total number of new members, etc.

Update: This form allows the user to add/delete/update the information kept in this database about movies, producers, customers, ticket sales, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a customer, an employee or a manager, the database administrator, etc. The user should have a login name and a password to be able to access the system. The customers access the database to learn about the movie schedule and new movies, to purchase tickets, or to update their personal information. The managers use the system to identify total expenses, total revenues, employee schedules, etc. The database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

“Shop Electronics” is a company that offers electronics products and services to the local community. The company offers a variety of products, such as television sets, computers, DVD players, CD players, telephones, digital cameras, printers, etc. The company offers the following services: part replacements and maintenance for all the products, installation of computer programs, etc.

The company is planning to extend its business in the cities nearby. The first step of this process is to create a database information management system that would enable the managers to better manage the inventories, keep details about sales transactions, keep information about the customers, etc. The second step in this process is to create a web application that would provide customers with details about the products and services offered and enable customers to purchase products on-line. Finally, after the customers in the cities nearby become familiar with the products and services offered, the company is planning to open electronics shops there.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Customer*: The main attributes are identification number, name, address, telephone number, e-mail, current balance, etc.
2. *Employee*: The main attributes are identification number, name, address, telephone number, employment date, position, employment history, login name, password, etc.
3. *Product*: The main attributes are identification number, name, type, brand name, year of production, manufacturer, inventory level, safety stock level, etc.
4. *Supplier*: The main attributes are identification number, name, address, contact information, current balance, etc.

Note the following: (a) A customer may choose to become a member. Members can have access to parts of the database. For each member, the following additional information is required: login name, password and shopping card. (b) *Shopping card* is a multi-valued attribute of the entity type *Member*. This attribute consists of item ordered and order date. (c) There are different types of products, such as computers, television sets, digital cameras, etc. Each product type has specific attributes that distinguish it from the rest of the products. For example, the following information is specific to computers: processor speed, processor description, hard drive, CD speed, modem speed, RAM, etc. (d) When a product is sold to a customer, the following information is recorded: transaction number, date, quantity sold, unit price, amount due, and due date. (e) When an order for a particular product is sent to a supplier, the following information is recorded: order number, date, quantity ordered, unit price, amount due, payment date, and delivery date.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

Queries:

1. The following set of queries helps with financial analysis:
 - a. Calculate total expenses, revenues, and earnings during the current month.
 - b. Calculate the annual expenses, revenues, and earnings.
 - c. Calculate earnings by product type.
 - d. Calculate earnings by service type.
2. The following queries present important business performance indicators:
 - a. The total number of new customers during the current month.
 - b. The total number of products sold during the current month. Group this information by product type.
 - c. The average annual inventory by product type.
 - d. The five product types that sold the most during the current year.
 - e. Present details about late deliveries of the current month.
3. Create a query that prompts for the identification number of a product type and returns details about the orders placed during the current month for this product type.
4. Create a query that prompts for the identification number of a customer and returns details about the purchases that the customer made during the current month.
5. Create a query that prompts for a brand name and returns details about the product types of the selected brand name.
6. Create a query that prompts for the brand name of a computer, processor speed, and RAM, and returns details about the computers held in the inventory that satisfy these specifications.
7. Create a query that presents details about the products for which the current inventory level is at or below the safety stock level.
8. Create a query that presents details about the ten most preferred customers.
9. Create a query that presents details about the ten most reliable suppliers. An indicator of a reliable supplier is delivering products on time.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: employees, products, customers, orders, suppliers, etc. These forms allow the user to add, update, and delete information about employees, products, customers, orders, suppliers, etc.
3. Create a form that allows the user to browse through the financial reports created.
4. Create a form that allows the user to browse through the business performance reports created.

5. Create a form that allows the user to browse through the information saved in the database about different product types. Insert a subform that presents details about orders placed and orders received during the last month for the selected product type. Insert a subform that presents details about the current month's sales of the selected product type. Insert textboxes that present the following information: (a) total number of products sold during the current month and the current year, and (b) revenues from sales during the current month and the current year.
6. Create a form that allows the user to browse through the information kept in the database about customers. Insert a subform that present details about the purchases made by the selected customer. Insert a subform that presents details about the services provided to the selected customer. Insert textboxes that present the following information: (a) the total amount paid by the selected customer for purchases made during the current month and current year, and (b) total amount paid by the selected customer for services.
7. Create a form that allows the user to browse through the information saved in the database about suppliers. Insert a subform that presents details about the products offered by the selected supplier. Insert a subform that presents details about the orders placed with the selected supplier that have not been delivered yet. Insert a command button that, when clicked-on, returns details about the late orders of the current supplier. Insert textboxes to present the following information: the total amount of money due, the total payments made during the current month, and the total payments made during the current year.

Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

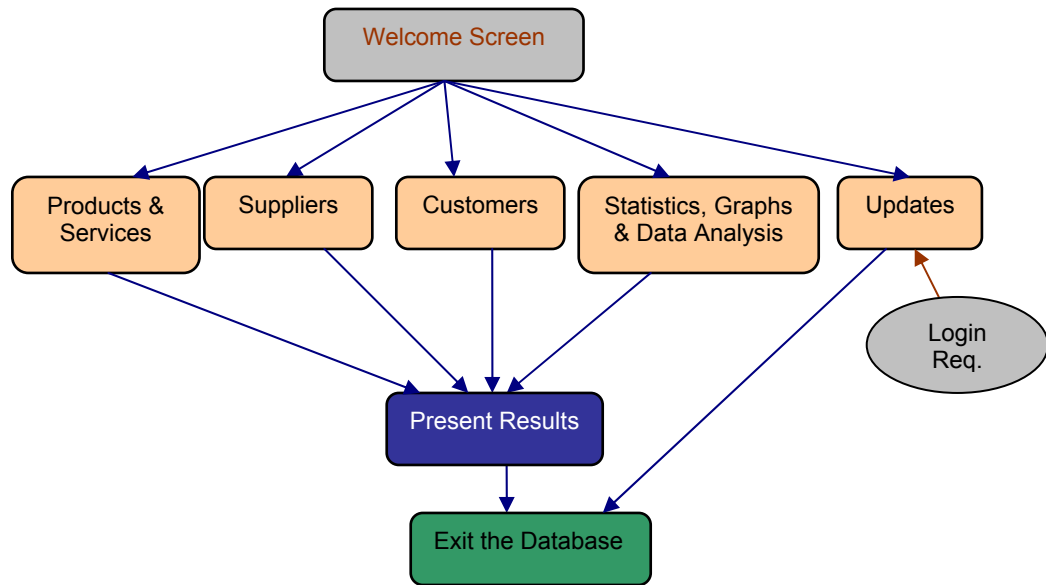
Reports

1. Create the following financial reports:
 - a. Report details about the current month's revenues, expenses, and earnings.
 - b. Report the annual revenues, expenses, and earnings of the store.
 - c. Report the annual earnings by product type.
 - d. Report the annual earnings by service type.
 - e. Report details about annual expenses from salaries.
2. Create the following business performance reports:
 - a. Report details about the products that sold the most during the current year.
 - b. Report details about the most reliable suppliers.
 - c. Report the annual average inventory level by product type.
 - d. Report details about the most preferable service type.
3. Use the chart wizard to plot the following:
 - a. The total number of customers per month during the last twelve months.
 - b. The quantity sold during the current year by product type.

- c. The annual earnings during the last five years.
 - d. The value of products ordered during the current year from each supplier.
4. Use the label wizard to create labels with the address of each customer.

Visual Basic.NET Application Development

This database application can be used by the employees and the managers of the electronics shop, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Products & Services: This part of the database keeps information about products and services offered by the electronics store. The customers browse the database to learn about products and services offered; prices, etc.

Suppliers: This part of the database carries information about suppliers.

Customers: This part of the database keeps information about customers.

Statistics, Graphs & Data Analysis: Users browse this part of the database to identify trends in: the annual expenses, the annual revenues, and total earnings by product and service; the total number of customers, etc.

Update: This form allows the user to add/delete/update the information kept in this database about products, suppliers, customers, orders, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a customer, an employee or a manager, the database administrator, etc. The user should have a login name and a password to be able to access the system. The customers

access the database to learn about the products and services provided, purchase products on-line, or update their personal information. The managers use the system to identify total expenses and total revenues; set orders with the suppliers; view the current inventory and safety stock level of a particular product; etc. The database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section.

Problem Description

“Mediterranean Food” is a wholesaler that provides a variety of Mediterranean food products. The owners of the company are originally from southern Italy. They receive orders from a number of companies that produce Mediterranean food products. Their suppliers are located in Italy, Greece, Turkey, etc. Their customers are companies located in the United States and Canada.

The company is interested in building a database information system that would help with the following: maintaining details about orders, customers, and suppliers; creating financial reports; analyzing the progress of the business; etc. Furthermore, the company is interested in creating a web application that would allow customers to place orders on-line, check the status of an order, etc.

Database Design

We present the main entity types of this database. For each entity type, we provide some of the corresponding attributes. Use this information in order to: (a) Build an Enhanced E-R diagram; (b) Transform the Enhanced E-R diagram to a relational database. Identify the primary key(s) and the foreign key(s) for each relation. Draw the relational integrality constraints; (c) For each of the relations created, indicate its normal form. If the relation is not in the 3NF, decompose it into 3NF relations.

1. *Customer*: The main attributes are identification number, name, address, contact information, e-mail, current balance, login name, password, shopping card, etc.
2. *Product*: The main attributes are identification number, name, brand name, description, manufacturer, expiration date, inventory level, safety stock level, etc.
3. *Supplier*: The main attributes are identification number, name, address, contact information, current balance, login name, password, etc.
4. *Employee*: The main attributes are identification number, name, address, telephone number, employment date, position, employment history, login name, password, etc.

Note the following: (a) When a customer places an order (submits an invoice), the following information is recorded: order number, date, quantity, unit price, amount due, payment amount, due date, payment date. (b) When an order is shipped to the customer, the following information is recorded: shipment number, shipment date, expected arrival date, actual arrival date, descriptions, etc. (c) When the company sets an order with a supplier, the following information is recorded: order number, date, quantity, unit price, amount due, payment amount, due date, payment date. (d) When a customer returns an item, the following information is recorded in the database: return number, date, description, etc.

Access Application Development

The following are some of the queries, forms, and reports one can create in order to increase the functionality of the database:

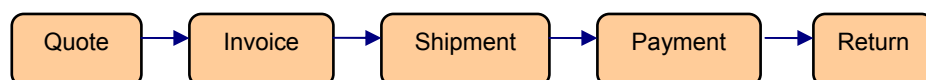
Queries:

1. The following set of queries helps with financial analysis:

- a. Present details about the current month’s revenues from sales.
 - b. Present details about the current month’s expenses.
 - c. Present the annual revenues, expenses, and earnings of the company.
 - d. Present the annual earnings by product.
 - e. Present the annual revenues by customer.
 - f. Present the annual expenses by supplier.
2. The following set of queries provides a number of business progress indicators:
- a. Present the total number of orders submitted and orders returned by the customers during the current month.
 - b. Present the total number of orders submitted to suppliers during the current month.
 - c. Present details about the total number of new customers during the current year.
 - d. Present the total number of late shipments during the current month.
 - e. Present details about the customers who have a negative account.
 - f. Present details about the five products that have sold the most.
 - g. Present details about the five most expensive products.
 - h. Present the current inventory level per product.
3. Create a query that presents details about the orders that have not been received yet.
4. Create a query that presents details about the items for which the current on-hand inventory is smaller than the safety stock level.
5. Create a query that presents the current inventory position of each product. The inventory position for a particular product is calculated as follows: on-hand inventory + orders not received yet from the suppliers - backorders.
6. Create a query that prompts for the identification number of a product and returns details about the orders placed for that product that have not been received yet.
7. Create a query that prompts for the identification number of a customer and returns details about the orders placed by the customer that have not been delivered yet.
8. Create a query that prompts for the identification number of a supplier and returns details about the orders placed with the supplier that have not arrived yet.

Forms:

1. Create a user sign-in form together with a registration form for new users.
2. Create the following data entry forms that are used for database administrative functions: products, orders, customers, shipments, suppliers, etc. These forms allow the user to add, update, and delete information about products, customers, employees, suppliers, etc.
3. The following presents the flow chart of the business activities of this company. Below, we present details about creating each part of the flow chart.



- a. Create a form that provides the customers with quotes about the items they plan to purchase. This form should provide details about the customer requesting a quote. Insert textboxes to present the current date and the name of the employee processing the quote. Insert a subform that presents the following details about the products for which the customer needs a quote: product identification number, name, unit price, quantity, and payment amount. Insert textboxes to present the following: the total payment amount (overall products), sales tax, and final payment due. Insert a command button that, when clicked on, allows the user to process the quote.
- b. The invoice form is similar to the quote form described above. The form presents details about the customer placing an order, the current date, and the employee preparing the invoice. Insert a subform that presents details about the items ordered through this invoice. Insert a command button that, when clicked on, allows the user to choose previous quotes to add to the current invoice. Insert textboxes to present the following: payment amount, sale tax, total amount due, and payment method (cash, check, or credit card). Insert a command button that, when clicked on, submits the invoice.
- c. The shipment form presents details about the orders shipped, such as name, address, and contact information of the customer; identification number of the invoice; items included in the order and corresponding quantities; shipment date, expected arrival date, and shipment costs. This form should allow the user to add/update a shipment.
- d. The payment form presents details about payments made by customers for the orders received. This form should allow the user to add/update a payment.
- e. Finally, the return form presents details about the items returned by customers. This form should allow the user to add/update a return.

Design a logo for this database. Insert this logo in the forms created above. Pick a background color for the forms and colors for the borders of the titles. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

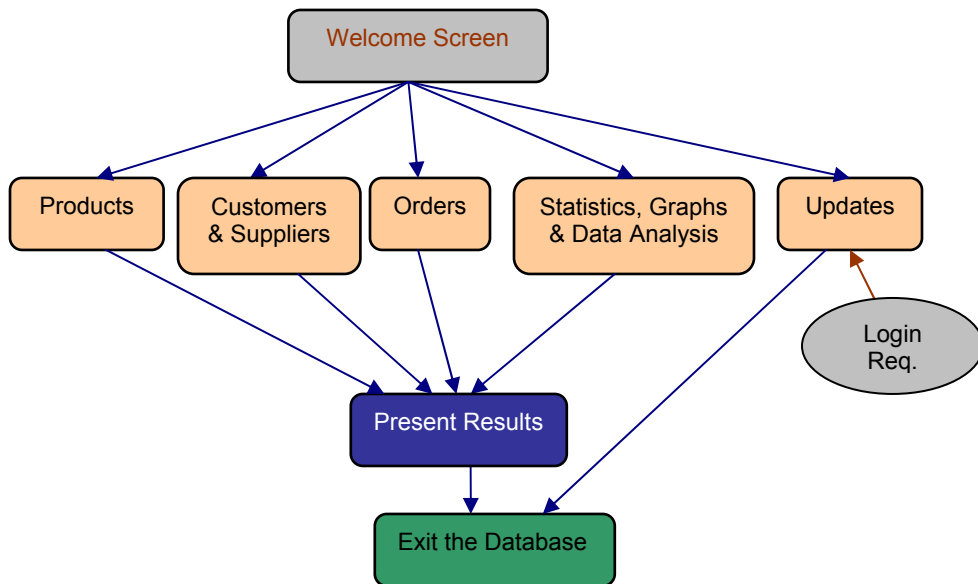
Reports

1. Create the following financial reports:
 - a. Report details about revenues and expenses for the current month. Calculate the total earnings.
 - b. Prepare a summary report of annual revenues, expenses, and earnings.
 - c. Present annual earnings by product.
 - d. Present annual revenues generated. Group this information by customer.
 - e. Present annual expenses. Group this information by supplier.
2. Create a report that prompts for shipment identification number and presents details about the selected shipment.
3. Report details about the quotes, invoices, and shipments submitted during the current month.
4. Report details about late shipments.

5. Use the chart wizard to plot the following:
 - a. The total number of late shipments per month during the last twelve months.
 - b. The total earnings per month during the last twelve months.
 - c. The total number of orders satisfied per month during the last twelve months.
 - d. The total number of new customers per year during the last five years.
 - e. Current on-hand inventory per product.
 - f. The total number of returns per month in the last twelve months.

Visual Basic.NET Application Development

This database application can be used by the employees and the managers, the customers, the database administrator, etc. In the following figure we present a tentative layout of the system.



In the welcome screen, the user can choose one of the five options presented. We give details about the forms or set of forms to be included in each option; however, you are encouraged to add other forms you find relevant. We suggest that the queries, forms, and reports already created in the Access Application Development section be included in here.

Products: This part of the database keeps information about products offered by the company. The managers browse the database to learn about the inventory level of a particular product, check the price of a product, etc.

Customers & Suppliers: This part of the database presents information about the customers and suppliers.

Orders: This part of the database allows the user to place an order, check the status of an order, etc.

Statistics, Graphs & Data Analysis: Users browse this part of the database to identify trends in: the annual expenses, the annual revenues, and total earnings by product; the total number of customers; etc.

Update: This form allows the user to add/delete/update the information kept in this database about products, suppliers, customers, orders, etc.

Web Extension

A user may access this database from personal computers at home or in the office. The user could be a customer, an employee or a manager, the database administrator, etc. The user should have a login name and a password to be able to access the system. The customers access the database to do the following: learn about the products and their prices; ask for quotes; submit invoices; make payments; update their personal information; etc. The managers use the system to identify total expenses and total revenues; place orders with the suppliers; view the current inventory and safety stock level of a particular product; etc. The database administrator can have access to the update forms.

Develop an ASP.NET web application that will enable the users to access the database and perform the activities described above. Your application will have forms similar to the ones described in the VB.NET Application section