**Database Design: Using Postgresql and Web Tools to**

**Implement** **SILENT CAR AUCTION Database System**

This project is to use PostgreSQL and web to implement the RBMS application. In this case, you'll design a relational database for a monthly silent ear auction by a local bank. After your database design is completed and correct, you will create database tables and populate them with data. Then you will produce a form, five queries, a report, and a custom navigation pane. The form will record all bids for automobiles. The queries will display all newer automobiles that are available, all Chevrolets, and all Fords. Other queries will report the maximum bid and the winners and losers of each auction. The reports will sum­marize the available automobiles by year of manufacture. The custom navigation pane will allow access to all tables, forms, queries, and the report.

* BACKGROUND

A local bank, Merchant Savings Society, offers automobile loans to many types of customers, including some with shady credit histories. Subsequently, Merchant ends up repossessing an average of 25 cars per month. Once the cars are repossessed, Merchant sells them to individual buyers to recoup some of their losses. The cars are placed in a parking lot owned by Merchant; each car has the keys in the ignition. Potential buyers are invited to inspect the cars and start them, but they cannot drive the cars. To find out which cars are available each month, potential buyers subscribe to a mailing list. Buyers can look over the list of ears and decide if they want to inspect one. If the car looks good, the buyer puts in a bid to Merchant via e-mail. The highest bid wins the car. Merchant has been doing the paperwork for these transactions by hand, but management feels it's time to computerize the system because it's quite popular and many potential customers now participate. Because you are proficient in Microsoft Access, you have been hired to create a database for Merchant.

You must consider a number of parameters when designing the database for Merchant. When the bank repos­sesses a car and places it in the lot, bank employees note the location in the lot so they can find it easily. They also write down the year, type of car, mileage, and the VIN (vehicle identification number). Potential custom­ers' information is also kept in an Excel spreadsheet. Their name and address are recorded along with e-mail addresses. Finally, the bank needs to have some way of recording the bids that come in, Customers send in their bids on specific cars. The bids are recorded and the date is noted. A form would be a handy way to record these bids for each car.

Frequently, customers like to look at particular manufacturers' cars that are available. The bank would like to run some queries each month to list available Chevrolet and Ford cars. Also, customers are mostly interested in cars that are only a few years old, because they are the most economical. So, the bank would like a way of listing only cars that are two or three years old.

Of course, the bank needs to know the top bid for each auction, and would like to have a query that shows the top bid for each vehicle. Bank employees also would like to enter an Auto ID and see which customers won their bids and which lost for a specific car.

For advertising purposes, the bank would like a report of all available cars, grouped by year of manufacture. Finally, the bank would like a custom navigation pane to make it easy to work with the database.

**1. Creating the database (15 points)**

Use the SQL DDL statements to create the tables required for this project. Please also note that the tables are created in certain order such that by the time when a foreign key needs to be created, the corresponding primary key also need to be created.

1. First, determine the tables you need by listing the name of each table and the fields it should contain. Avoid data redundancy. Do not create a field if it could be created derived from other field in a query.
2. You'll need a transaction table. Avoid duplicating data.
3. You must mark the appropriate key field(s). Keep in mind that some tables might need a compound primary key to uniquely identify a record within a table.
4. Type records into the tables, using the names and addresses of your friends. Create at least 10 cus­tomer records.
5. Assume that 25 vehicles are available for bidding.
6. Make each customer bid on a vehicle. Have some customers bid on the same vehicle. Assume that all this data is for one month.
7. Appropriately limit the size of the text fields; for example, a zip code does not need the default set­ting of 255 characters.

FIGURE 4-4 Available Chevrolets query

**2. PostgreSQL Implementation (55 points)**

1. (7 points) Create a stored procedure called Available Fords. Your data will differ, but the output should resemble the following table.

Table 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Winners and Losers | | | | | |
| Auto ID | Location | Year | Type | Mileage  (000km) | Vin |
| 3 | L234 | 2006 | FORD F150 XL 6A ASBTW | 22 | 1FTRF12216NB66324 |
| 9 | C211 | 2005 | FORD FOCUS ZX4 4A ASB | 45 | 1FAFP34N45W254872 |
| 10 | L211 | 2005 | FORD F150 XL 4X2 6M ASBTW | 49 | 1FTRF122XSNB81984 |
| 12 | B196 | 2004 | FORD F150 OFFROAD 4X4 XCAB 8 | 60 | 1FTPX14504FA15423 |
| 23 | Z095 | 2002 | FORD EXPLORER EB 4X4 8A ASBW | 82 | 1FMDU74W02UB72009 |
| 24 | D135 | 2002 | FORD EXPLORER XLS 4X4 6A ASB | 110 | 1FMZU72E42UA13223 |
| 25 | F195 | 2002 | FORD F150XLT SPORT 4X2 6A A | 81 | 1FTRX17282NA98776 |

1. (7 points) Create a stored procedure called Max bid. This Query should display the Auto ID, Location, Year and Maximum Bid for each auto. Your output should resemble that in the following table, with different data.

Table 2

|  |  |  |  |
| --- | --- | --- | --- |
| Max Bid | | | |
| Auto ID | Location | Year | Maximum Bid |
| 1 | O765 | 2007 | $26,000.00 |
| 2 | C173 | 2007 | $14,500.00 |

1. (8 points) Create a stored procedure called Winners and Losers that prompts the user for an input. The input would be a specific Auto ID. Display the Auto ID, Last Name, Bid, Maximum Bid, and Won Bid?

Table 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Winners and Losers | | | | |
| Auto ID | Last Name | Bid | Maximum Bid | Won Bid? |
| 1 | Burham | $25,009.00 | $26,000.00 | No |
| 2 | Mattern | $26,000.00 | $26,000.00 | Yes |
| 3 | Meartz | $22,001.00 | $26,000.00 | No |

1. (8 points) Write two procedures to add tuples into the customer table and the available autos. As an example, you can use a procedure, e.g., **add\_customers** to add a tuple in the customer table, where **Customer ID, Last Name, First Name and etc.** are parameters of the procedure.
2. (15 points) If the **bid Date** is larger than one month (compare with the previous biding records of a certain auto), your program should perform the following tasks: (1) print a message indicating **bid\_date is overdue,** (b) if the bid by one customer is **less than the current max bid**, print another indicating message (c) if one month has passed since the first bid, the system should alert and update the database to describe which one win the bid. Use triggers to implement such update.
3. (10 points) You need to make your code user friendly by designing and displaying appropriate messages for all exceptions. For example, if someone wants to find the bid information of a customer but entered a non-existent customer id, your program should report the problem clearly.

**3. Interface (20 points)**

Implement a Web interactive interface using any program language. Your interface program should utilize as many of your database stored procedures/functions as possible.

**4. Documentation (10 points)**

Documentation consists of the following aspects:

1. Each procedure and function and every other object you create for your project needs to be explained clearly regarding its objective and usage.
2. Your code needs to be well documented with in-line comments.

**5. Hand-ins, Demo and Grading**

1. You will also need to submit your source code along with your documentation to the Blackboard.
2. It is required to demonstrate your project to the instructor using tuples created by the instructor.
3. The grading will be based on the quality of your code, the documentation and on how successful of your demo is.

FIGURE 4-4 Available Chevrolets query