**Database Design: Using MySQL and PHP to**

**Implement the Leasing luxury database System**

This project is to use MySQL and PHP to implement the RBMS application. In this case, you will design a relational database for a business that leases designer handbags. After your database design is completed and correct, you will create database tables and populate them with data. Customers will rent the handbags on your website. So you will produce a website will display the handbags available for rent by designer type and with some queries and reports. The queries will address the following questions: What handbags are available by a specified designer? How many handbags from each designer are in inventory? How many days has each customer leased handbags? Your reports will display the handbags available for rent by designer type and the amount of money each customer has spent leasing different handbags.

1.BACKGROUND

Consumers love luxurious products, but they often cannot afford to buy luxury. Your Aunt Mabel has started a company to address this problem; her new company leases expensive handbags to women. Aunt Mabel owns a large number of designer handbags, and several years ago she began to lend them out to friends for special occasions. After several friends offered to pay her for the “loan” of the handbag, Mabel realized that she could offer a valuable service to women who want an expensive designer handbag for an evening out, a wedding, or other formal event. Leasing Luxury, or LL as it is commonly known, operates in the Atlanta area. The company model is simple: The daily cost of renting a handbag is set at a certain amount, depending on the designer. For example, a Coach bag rents for $9 per day, shipping included. Customers can keep a bag as long as they like, but they continue to pay the daily fee. Optional insurance costs an extra $1 per day; the insurance covers any damage to the expensive bags.

You have been hired to create a database to keep track of the available handbags, the customers who register for the service, and their bag rentals. The leasing payments are handled manually at this point. You have several goals for the database. First, you need to keep track of all the handbags that Aunt Mabel owns and that are available for leasing. She categorizes the bags by *designer*, *type of bag*, *color*  for each bag.

Customer essentials need to be recorded as well. Aunt Mabel conducts the business on paper and via telephone, but she wants to move all work to the computer and Internet as soon as possible. Eventually, she would like customers to book their requests via a Web site.When customers register for the service, they provide their e-mail address, regular mailing address, and credit card number. (All transactions are conducted with credit cards.) Your database must record this leasing information for the handbags, along with the date rented, date returned, and whether the customer purchases damage insurance.

As customers request handbags, Aunt Mabel and the other workers at LL would like to be able to fill out a form that lists each available bag and includes fields for the customer ID and leasing dates. Mabel would like to be able to enter this information directly into the database.

Customers often visit the website to find what handbags are available by different designers. Mabel would like to be able to enter a designer’s name and then see a list of available bags by that designer.

Mabel would like to understand the rental habits of her best customers, who she defines as the customers that keep the bags the longest time. Therefore, she would like to be able to calculate how long customers keep the bags, sorted by the longest duration to the shortest. She would also like to know how many bags she has for each type of designer. This information will be used for marketing purposes.

**1. Building 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 will need by listing the name of each table and the fields it

should contain. Avoid data redundancy. Do not create a field if it can be created by a calculated

field in a query.

(2)You will need transaction tables (Rentals). Think about what business events occur with each customer’s transactions. Avoid duplicating data.

(3)You must mark the appropriate primary key field(s) or foreign key field(s) for each table.

Keep in mind that each table might need a compound primary key to uniquely identify a record within a table.

(4) Enter at least 10 records for handbags from four different designers.

(5) Enter records for at least eight customers, including their names, addresses, telephone numbers,

e-mail addresses, and fictional credit card numbers. Enter your own name and information as an

additional customer.

(6)Each handbag should be rented at least once. Each customer should rent a handbag twice.Appropriately limit the size of the text fields; for example, a telephone number does not need

the default length of 255 characters.

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

You need to write SQL queries, stored procedures/functions, and triggers to implement this project. The following requirements and functionalities need to be implemented.

1. (7 points) Write a stored procedure Bags by Manufacturer that inputs for the designer name and then lists the name of the bag and its color. In the example shown in table 1, the manufacturer name is Coach. Your output should resemble that shown in table 1, although your data will be different.

Table 1 Bags by Manufacturer

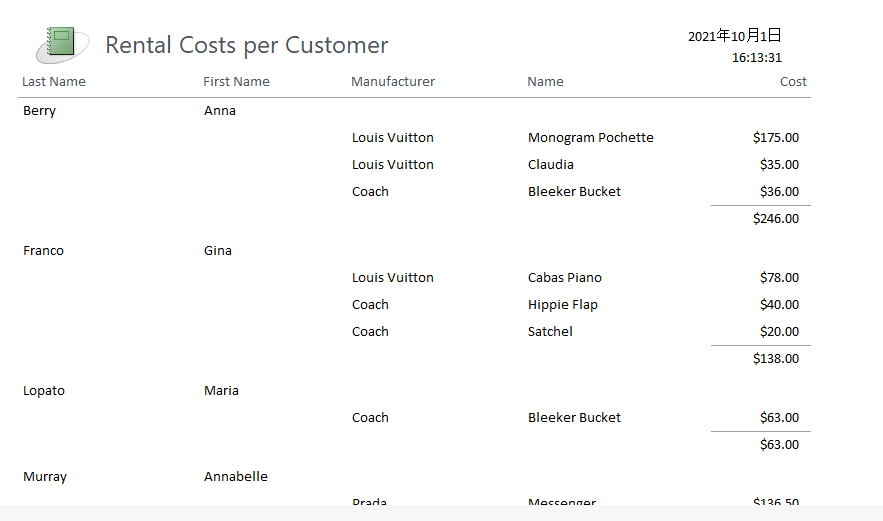
| **Bags by Manufacturer** | | |
| --- | --- | --- |
| **Name** | **Color** | **Manufacturer** |
| Satchel | Camel | Coach |
| Hippie Flap | Green | Coach |
| Bleeker Bucket | Blue | Coach |

1. (8 points) Write a stored procedure called Best Customers that calculates and adds the rental times of all handbags by each customer. The query should include columns for Last Name, First Name, Address, Telephone, and Total Length of rental. Your output should look like that in table 2, although your data will be different.

Table 2 Best Customers

| **Best Customers** | | | | |
| --- | --- | --- | --- | --- |
| **Last Name** | **First Name** | **Address** | **Telephone** | **Total Length of Rentals** |
| Zern | Joan | 58 W. Central Ave | 404-675-0091 | 40 |
| Murray | Annabelle | 59 W. Central Ave | 404-998-3928 | 31 |
| Smith | Patricia | 1700 E. Lincoln Ave | 404-765-3342 | 28 |
| Berry | Anna | 9 Pleasant Way | 404-887-4673 | 28 |
| Franco | Gina | 1012 Peachtree St | 404-887-2342 | 14 |
| Pao | Jill | 89 Orchard | 404-887-9238 | 13 |
| Lopato | Maria | 5490 West 5th | 404-234-8876 | 7 |
| Quinn | Sally | 54 Oak Ave | 404-987-3427 | 1 |

1. (7 points) Write a procedure to calculate and list the amount of money each customer is spending. The amounts will account for the number of days the bag has been rented. For example, you can use a procedure, say **report\_customer\_amount(customer\_id)**, for this operation.



1. (8 points) Write two procedures add tuples into the Rentals table and the bags table. As an example, you can use a procedure, say **add\_rentals** to add a tuple in the Rentals table, where **Customer ID,Bag ID,Date\_Rented,Date\_Returned etc** are parameters of the procedure. Note that **Date Rented** should be the current time (use current\_timestamp).
2. (15points) If a bag is returned, your program should perform the following tasks: (1) print a message indicating the total length of rental and the total amount of money, (2) make the bag is available to rent again. Use triggers to implement the tasks.
3. (10points) 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 purchases 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 PHP 、JavaScript etc . Your interface program should utilize as many of your MySQL 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.