CS 539

Requirements Specification for Database Programming Project 1

Introduction

In this project, you will design and implement a relational database system to support the operations of an online travel reservation system. You will use HTML for the user interface, MySQL for the database server, and Java, Javascript, and JDBC for connectivity between the user interface and database server.

You will have to install your own virtual machine with a web server that will host your web
application as well as a MySQL server. Everything will be under the AWS (Amazon Web Services).

You are to work in **teams of four**.

Project Specification

As you probably know, there are a multitude of online travel reservation systems on the web. Two popular ones are expedia.com and orbitz.com. I suggest that you visit these web sites to get an understanding of the look-and-feel of a travel web site and how such a system is supposed to function.

The basic idea behind your on-line travel reservation system is that it will allow customers to use theweb to browse/search the contents of your database (at least that part you want the customer to see) and to make flight reservations over the web. Your web site should allow users to make both domestic and international reservations. It should also allow users to query the database for available flights (direct or **with stops**) between a pair of cities for a given date and "approximate" time.

Actual travel sites allow you to do a lot more than simply make flight reservations. For example, you can book a rental car or a hotel room. Due to time limitations, we will stick to flight reservations. Your database system must be based on the specifications and requirements that follow.

1 System Users

The users of your system will be the customers (passengers) that use your system to make a flight reservation, and the site's manager. You should assume that the computer knowledge of the users is limited, and thus your system must be easy to access and operate.

In the end, at the login page of your website, manager is able to log in to their work page where they can do their jobs by accessing all customer's booking informations, and customer will be able to log in to their user page to make reservations respectively.

2 Required Data

The data items required for the travel reservation database can be classified into six categories:
airlines, airports, flights, reservations, customers, and *employees*.

The above classification does not imply any particular table arrangement. You are responsible for arranging the data items into tables, determining the relationships among tables and identifying the key attributes. Finally, you should specify and enforce integrity constraints on the data, including referential integrity constraints.

You will first create an E-R diagram of your online travel reservation system before developing
your relational model.

2.1 Airlines Data

This category of data should include the following items:

1. Airline ID
2. Airline Name

Each airline has a two-letter ID. For example, the ID for American Airlines is AA, and the ID for United Airlines is UA.

2.1 Airport Data

This category of data should include the following items:

1. Airport ID
2. Airport Name

2.3 Flight Data

This category of data should include the following items:

1. Flight Number
2. Airline
3. Number of Seats
4. Stops (with arrive times and depart times)
5. Working days
6. Fares (discount based on perches date etc.)

flight is operated by an airline, has a flight number (unique only within that airline), makes a sequence of airport stops (with each stop having local arrival and departure times), and operates on a given set of days of the week. For example, UA Flight # 1 flies from Chicago O'Hare on Mondays and Thursdays, departing at 5pm and arrives into Paris at 6am the next day. It then departs Paris at 10am local time to arrive in Rome at 12pm, and so on. A flight also has an associated fare and fare restrictions, which include Advance Purchase (deeply discounted fares may require advance purchases of a 3, 7, 14, or 21 days) and Length of Stay (some round trip fares may have minimum or maximum stay requirements). Notice that the Fare of a flight is associated with the stops.

2.4 Reservation Data

This category of data should include the following items:

1. Reservation Number
2. Date
3. Passenger(s)
4. Legs (stops)
5. Total Fare
6. Fare Restrictions
7. Booking Fee
8. Customer Representative

A reservation has a unique number and is either for a single passenger or a set (maybe a group/family) of passengers. Each reservation is associated with a

sequence of direct flight segments called the legs of the reservation. Each leg has an associated from airport, to airport, flight number (along with its airline), departure date and time, special meal ordered, seat number, and class (economy/business/first). A reservation also has the following attributes: total fare, fare restrictions, and date when reservation made. For example, Mr. John Smith makes a reservation on 1/1/2010. He flies from Chicago on 1/10/2010 on flight UA #1 to Paris. After staying for a couple of days in Paris, Mr. Smith departs on 1/12/2010 from Paris on flight UA#3 to London, and so on.

A reservation also has an associated booking fee, which is how your company makes money, and an associated customer representative.

2.5 Customer Data

The items required for this category include:

1. Last Name
2. First Name
3. Address
4. City
5. State
6. Zip Code
7. Telephone
8. E-mail Address
9. Account Number
10. Account Creation Date
11. Credit Card Number
12. Preferences

A customer may partake in any number of flight reservation transactions, and may have one or more accounts from which to purchase tickets. Associated with each account is a reservation portfolio, indicating which reservations are held in that account. Customer preferences include aisle or window seat, preferred meal, etc.

2.6 Manager Data

This category of data should include the following:

1. Account number
2. Password
3. Employee number

3 User-Level Functionality

3.1 Manager-Level Functionality

The manager should be able to:

1. Add, Edit and Delete information for an customer
2. Obtain a sales report for a particular month
3. Produce a comprehensive listing of all flights
4. Produce a list of reservations by flight number or by customer name
5. Produce a summary listing of revenue generated by a particular flight, destination city, or customer
6. Determine which customer generated most total revenue
7. Produce a list of most active flights
8. Produce a list of all customers who have seats reserved on a given flight
9. Produce a list of all flights for a given airport
10. Produce a list of all flights whose arrival and departure times are on-time/delayed

3.2 Customer-Level Functionality

Customers should be thought of as online airline ticket buyers and should be able to easily browse your online travel reservation system on the web and make flight reservations. In particular, they should be able to make the following types of reservations:

- One-Way
- Round-Trip
- Multi-City
- Domestic or International
- Flexible Date/time

A customer should also be able to **cancel an existing reservation** and they should be able to retrieve the following information:

- A customer's current reservations
- Travel itinerary for a given reservation
- A history of all current and past reservations a customer has made

- Best-Seller list of flights

4 User Access Control

Your database system should provide controlled access to the data by distinguishing between the different types of users: manager and customers.

-
- A customer should not be allowed access to other customers' account information, or to any employee information.

5 User Interface

HTML and its successors provide facilities for creating pop-up and pull-down menus, value lists, input/output forms, labels and customized reports. You should make use of all of these capabilities, and in the process come up with a system that caters to users with only limited computer knowledge. The information you provide to customers should look professional and inviting.

6 Airline and Flight Database

Here is an online available database (Openflights): https://openflights.org/data.html
It provides a vast amount of data for you to test your finished system.
I recommend you do this project with some synthesized data by yourself before you finish the whole project. Then try to import the Openflights data and use it to test your data.
It is not mandatory to use the Openflights database, but make sure to provide me enough synthesized data in your database when I am testing your system. You will get up to 2% points deducted if I find too few data in your database to perform all the testing cases when I grade your projects.

Good luck!

Project 1 Schedule

1. March 2nd: ER Diagram of your system design. A .pdf file of your ER Diagram, it should reflect your design of your database scheme, you final project may not necessarily follow this ER Diagram because you make find improvements and may face problems that force you to change your design during your work. But the overall diagram has to be in high quality and overall structure has to be consistent with your final project.

2. March 16th: Web user interface. I want to see you have finish the web framework UI build, so you can focus on developing the SQL-related functions in the rest time. Which includes:
   a. A usable login-in page that I can create an account, login in as ordinary customer, login in as system manager
   b. System worker page when I login in as a system manager, this page is not needed to function well in terms of the SQL part, but the UI of the webpage has to be finished or partially finished. I just want to make sure you have finished the front-end part of your project. (It means it can even be a blank page with the simplest buttons, table and list shown there, if you decide not waste your time on HTML, CSS)
   c. Customer page when I login in as a customer, it may include different pages such as a search page, flight history page, my account page etc, it varies depends on how you design your website. Requirements are same as in b.

3. March 23th: All SQL queries you gonna use in your website development, and comment on each query to let me know in what case you need this query and what output result you want to get. This includes queries that is already implemented into your project and those that you plan to and want to implement into your project but you failed to do so due to time limitation and technique limitation (you may get extra credits if you show me some meaningful queries or high-quality queries that you failed to implemented in the end). The queries are not necessarily to be exactly same as in your final project submission, but at least 80% have to be consistent.
   For example: you want to check a user login to make sure the user inputs right user-name and a matched password, you need the query:
SELECT password from USER where username = ""

   4. March 30th: Final due of the whole project.

Note that you can submit your whole website source code by export it from eclipse, please make sure I can run your code on my local machine.