**UMBC**

**IS420 Database Application Development**

Fall 2020

Group Project

**Hotel Management System**

**Description**

UMBC Hotels Inc. (U.Hotels), is a fictitious hotel company. U.Hotels is in need of upgrading its hotel management system and has sought your help. Your team has been contacted by U.Hotels and you have signed a contract to help them organize their office operations, using an Oracle database. You will code PL/SQL procedures and functions with a number of operations that U.Hotels will be using to perform the day-to-day business with rooms, reservations etc.

U.Hotels is a typical hotel chain. Customers make reservations for specific dates and get specific services offered by hotel (nights they stay, hotel restaurant, etc.) which they pay for when they check out.

**Database Design**

There are several types of information to be stored in the database on hotels in different cities operated by U.HOTELS, customers, reservations, cancellations etc. You are free to make your own design in the database and create your own tables. The following list outlines the minimum information you should capture in the database. Feel free to add more tables and attributes if necessary to support the office operations more effectively.

The database you will create will have information on hotels, such as Address, Phone, Room Types available, etc.

Customer information includes Name, Address, Phone, Credit Card, etc.

The most vital part of the database should be information on reservations. It should include Client Name, Dates, Rate, Room Type, etc.

**Important Assumptions**

To make this project manageable and within time limits we have to make several assumptions. The purpose of these assumptions is to give specific details to certain situations to guide your coding efforts.

|  |  |
| --- | --- |
| **Month** | **Room Rate** |
| Sep - Apr | $100 (single), $200 (double), $500 (suite), $1000 (conf. room) |
| May - Aug | $300 (single), $500 (double), $900 (suite), $5000 (conf. room) |

1. The rates of each hotel room fluctuate according to peak season or off-season. Use the rate in the table provided for each room depending on the season.
2. You should make reservations for consecutive days for a specific Room type.
3. Guests can check-in with or without reservations.
4. If a reservation is made 2 months in advance or more, the customer gets a 10% discount on the rate. Otherwise, the customer has to pay full rate.
5. There are no taxes charged to customers. U.HOTELS made a deal with the IRS not to collect taxes from guests (to make things easier for the implementation)
6. Room types: single room 1-bed, double-room 2-beds, suites, conference rooms
7. Guests pay their bill (invoice) when they check-out.
8. A guest can reserve and stay in multiple rooms at the same time (e.g. a large family) and must pay for all reserved rooms.
9. The services that are offered by all U.HOTELS are the same. They include:
   1. Restaurant services (assume $20 per person per meal)
   2. Pay-per-view movies (assume $5 per movie)
   3. Laundry services (assume $10 per time – regardless of number of items)
10. Guests can check out earlier (before the last day they reserved a room) or later (they can stay longer if there is room availability)
11. Guests cannot reserve rooms that span across months with different rates. You do not have to implement this, but do not enter reservations for such cases.

The actual tasks for the hotel management system are given below.

**Individual tasks**: The operations and reports below are to be implemented by team members individually. Some tasks are simpler than others, therefore, different members may have different number of tasks in order to have a balanced distribution among team members. Individual tasks will be graded by uploading them on Blackboard as individual members.

**Integration tasks**: The entire team must work together only for integration of the individual tasks. This will take more time than one might anticipate because there will be conflicts and different assumptions among team members that would need to be resolved. You must set aside at least 10 days for integration of the various components of the project. Integration will be graded during the demo.

* Member 1:
  1. Add a new hotel: Create a new hotel with appropriate information about the hotel as input parameters, including name, street, city, state, phone, etc. For simplicity, each newly created hotel must have 50 single rooms, 20 double, 5 suites, and 2 conference rooms.
  2. Find a hotel: Provide as input the address of the hotel and return its hotel ID
  3. Add room to hotel: Given a hotel ID add a specific type of room to it with an input number of instances. For example, add 10 double rooms, or add 2 suites.
  4. Sell existing hotel: Sell a hotel by providing its hotel ID. Mark it as sold, do not delete the record, and print all sold hotel information. Show hotel ID, location, etc.
  5. Report Hotels In State: Given a state, display name, address, phone number, and number of available rooms along with room type of each hotel in that particular state.
* Member 2:
  1. Make a reservation: Input parameters: Hotel ID, guest’s name, start date, end date, room type, date of reservation, etc. Output: reservation ID (this is called confirmation code in real-life). NOTE: Only one guest per reservation. However, the same guest can make multiple reservations. Also, make sure that there is availability of that room type before a reservation is made.
  2. Find a reservation: Input is guest’s name and date, and hotel ID. Output is reservation ID
  3. Cancel a reservation: Input the reservationID and mark the reservation as cancelled (do NOT delete it)
  4. ShowCancelations: Print all canceled reservations in the hotel management system. Show reservation ID, hotel name, location, guest name, room type, dates
* Member 3:
  1. Change a reservationDate: Input the reservation ID and change reservation start and end date, if there is availability in the same room type for the new date interval
  2. Change a reservationRoomType: Input the reservation ID and change reservation room type if there is availability for that room type during the reservation’s date interval
  3. Show single hotel reservations: Given a hotel ID, show all reservations for that hotel
  4. Show single guest reservations: Given a guest name, find all reservations under that name
  5. Monthly Income Report: Calculate and display total income from all sources of all hotels. Totals must be printed by month, and for each month by room type, service type. Include discounts.
* Member 4:
  1. Add a service to a reservation: Input: ReservationID, specific service. Add the service to the reservation for a particular date. Multiple services are allowed on a reservation for the same date.
  2. Reservation Services Report: Input the reservation ID and display all services on this reservation. Print “no services for this reservation” if none exists.
  3. Show Specific Service Report: Input the service name, and display information on all reservations that have this service in all hotels
  4. Total Services Income Report: Given a hotelID, calculate and display income from all services in all reservations in that hotel.
* Member 5:
  1. Show available rooms by type: Given a hotel ID, display the count of all available rooms by room type.
  2. Room Checkout Report: Input: ReservationID Output:
     + Guest name
     + Room number, rate per day and possibly multiple rooms (if someone reserved several rooms)
     + Services rendered per date, type, and amount
     + Discounts applied (if any)
     + Total amount to be paid
  3. Income By State Report: Input is state. Print total income from all sources of all hotels by room type and service type in the given state. Include discounts.
* GUI: There is no Graphical User Interface (GUI) for this project. You need to create PL/SQL procedures and functions that carry out the tasks identified above. Each task will be a separate PL/SQL stored procedure or function.
* Input/output: For tasks that require input parameters, you need to call the corresponding PL/SQL procedure or function and pass to it the input parameters. This means that you need to have another program that calls your procedures and functions.
* Do not hard-code ID values for primary keys. Use sequences instead. Primary keys should be automatically generated based on sequences, otherwise the penalty is 10% off of the final project grade.
* It is fine to call procedures created by other members of your team to get information that you need for your own procedure. For example, if you need a hotelID, you can call the Find a Hotel function which returns the hotel ID, and then you can use that ID in your procedure/function.
* How to speed up your work: First start by writing and completing the simple tasks. Make sure that you are DEBUGGING your code:
  1. First thing to do in each procedure/function is to print out the values of the input parameters. This way you know that they were passed correctly, before you start working on the main part of the procedure/function.
  2. Occasionally within the procedure/function print out the values of variables, just to make sure your procedure is progressing correctly.
  3. Always use EXCEPTIONs to explain what went wrong. This will definitely speed up the implementation time. In addition, EXCEPTIONS ARE REQUIRED for every single procedure and function. There will be points taken off for missing exceptions.
* Experiment in your own Oracle account. Make sure your code works well and then transfer it to the Team Oracle account for integration with the other team members’ code. Team accounts will be created in due time.
* Teams with fewer members: If your team has 4 or fewer members (due to members dropping the course) then you must do the following:
  1. Let me know if your team has 3 or fewer members right away
  2. If your team has 4 members, the tasks of the fifth member must be distributed and allocated to the surviving team members. You need to let me know that this is the case, so that I will take it into consideration when assigning a grade to you.

Once the above stored procedures/functions have been created and tested, each team needs to write a driver program, which calls the above procedures. There will be a scenario given to the class that describes which procedures should be called including parameters. The scenario will be given in English, you will need to translate it into PL/SQL procedure/function calls with the necessary parameters. A short scenario is given below.

**Sample scenario**:

1. Add a hotel:
   1. Add a new hotel called H1 in New York, NY
   2. Add a new hotel called H2 in Baltimore, MD
   3. Add a new hotel called H3 in San Francisco, CA
2. Find a hotel:
   1. Find the hotel ID for the hotel H3
   2. Find the hotel ID for the hotel H2
3. Add a room:
   1. Add 5 double rooms to H2
   2. Add 2 suites to H2
   3. Add 10 double rooms to H1
4. Sell H1
5. Reservations:
   1. Make a reservation at hotel H2 by John Smith from Aug 1 – Aug 10 for a suite
   2. Make a reservation at hotel H1 (already sold)
6. Add services:
   1. For reservation in item 5a add restaurant services for each day
   2. For reservation in item 5a add 1 pay-per-view movie for the first day
   3. For reservation in item 5a add laundry service for one day (of your choice)
7. Checkout:
   1. Show the checkout/invoice list of John Smith

A complete scenario will be given later in the semester to the entire class. Each team will take the scenario and write a driver program by calling the appropriate procedures that carry out the items mentioned in the scenario. The driver program is a script of PL/SQL anonymous code that implements a given scenario. A test scenario will be given earlier, and a final scenario will be given about a week before the final demonstrations.

**Project Grading**: The project is divided into several deliverables that are part of the project and they are due throughout the semester. Most deliverables are individual work, whereas a few are group deliverables. In essence, each student needs to take ownership of specific components, work individually on them, and then work with all team members to integrate his/her components together into a single working project.

* Demo: Towards the end of the semester there will be a demo of the entire project. All team members are required to participate and demo their own components. Each student will present his or her individual procedures in front of the Professor, the TA, and the Lab.
* The demo will take place during Lab time.
* The demo is the final deliverable, and it represents the highest weight of the project. In that deliverable individual components make up 80% of its grade, and the team/integration components make up 20% of the grade.
* If a team member does not fulfill his or her own procedures, the entire team will not have a complete product, therefore, integration penalty will be applied to all members of the team