**UMBC**

**IS620 Database Application Development**

Spring 2019

Group Project

**Event Management System**

**Description**

Events R Us Inc. (ERU), is a fictitious company that manages events like conferences, weddings, etc. ERU has sought your team’s help, and has signed a contract with your team to get help in organizing their event operations, using an Oracle database.

You are to provide a management system where customers book events in a specific hotel. Events are birthdays, weddings, conferences, etc. A customer making a reservation can reserve one hall, however, the same customer can make multiple reservations if needed (e.g. a conference event that requires 3 small halls, and 1 large hall results in 4 reservations).

You will code PL/SQL procedures and functions with a number of operations that ERU will be using to perform the day-to-day business with events that take place in hotel rooms, reservations to events, cancelations, etc.

**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.

|  |  |  |
| --- | --- | --- |
| **Room type** | **Capacity** | **Price** |
| Small Hall | 100 | $500 |
| Medium Hall | 250 | $1000 |
| Large Hall | 500 | $2000 |

1. The daily rates of each room type and their capacities (max number of people) are as follows:
2. You should make event reservations for consecutive days for a specific room type.
3. 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.
4. There are no taxes charged to customers. ERU made a deal with the IRS not to collect taxes from guests (to make things easier for the implementation)
5. Event Room hall types: small hall, medium hall, large hall.
6. Customers pay their invoices when the event is over.
7. 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.
8. The services that are offered by all ERU events are the same. They include:

|  |  |
| --- | --- |
| **Service Type** | **Amount** |
| Breakfast | $10 per person |
| Lunch | $20 per person |
| DJ | $500 per event |
| Singer | $2000 per event |
| Pop band | $10000 per event |

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

**Individual tasks**: The operations and reports listed 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. Minimum parameters are Address, Phone, Event Room Types available etc. Have in mind that you will need more parameters.
  2. Find a hotel: Provide as input the address of the hotel and return its hotel ID
  3. Display hotel info: Given a hotel ID, display all information about that hotel
  4. Add Event Room: Given a hotel ID, add a new room for a specific event to that hotel. The room types are: Small Hall, Medium hall, Large hall.
  5. Report Hotels and Event Rooms In State: Given a state, display event room information of all hotels in that particular state. Include total capacity per event room type per hotel.
  6. Show available rooms by type: Given a hotel ID, display the count of all available rooms by room type.
* Member 2:

1. Make an event reservation: Input parameters: Hotel ID, guest’s name, start date, end date, event type, date of reservation, number of people attending, etc. Output: event reservation ID (this is called confirmation code in real-life). NOTE: Only one person can make an event reservation. However, the same person can make multiple reservations. Event types: Birthday, Wedding, Conference, Workshop, Hackathon, University Admission, etc. Also make sure that the reserved hall has capacity that can hold the number of people attending. For example, for a conference of 500 people, a customer must reserve 2 medium halls and a large hall for each day of the conference, usually 3 consecutive days.
2. Find an event reservation: Input is a person’s name and date, hotel ID. Output is event reservation ID
3. Cancel an event: Input the event reservationID and mark the reservation as cancelled (do NOT delete it)
4. ShowCancelations: Print all canceled events in the event management system. Show event reservation ID, hotel name, location, event type, room type, dates.
5. Change an event Date: Input the event reservation ID and change event start and end date, if there is availability in the same or larger room type for the new date interval

* Member 3:

1. Change an event RoomType: Input the reservation ID and change reservation room type if there is availability for the new room type during the reservation’s date interval
2. Show specific event: Given an event type (Birthday, Wedding, etc.) display all events of that type in all hotels along with the address of the hotel and the date of the event.
3. Show events by person: Given a person’s name, find all events under that name
4. Total 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 event and service type. Include discounts.
5. Event Invoice: Input: Event reservationID Output:
   * + Name of person that reserved the event
     + Event room number(s), 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

* Member 4:

1. Add a service to an event: Input: Event reservationID, specific service. Add the service to the event for a particular date. Multiple services are allowed on a reservation for the same date. For meals make sure to multiply the amount by the number of people attending the event.
2. Reservation Services Report: Input the event reservation ID and display all services on this reservation. Also print the number of attendees of the event. 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. Services Income Report: Given a hotelID, calculate and display income from all services in all reservations in that hotel.
5. Income by State Report: Input is a specific state. Print total income from all events as follows: Each output line should contain information of a specific event ID (income from room type, income from services, total income of this event). At the end a grand total of all events income. 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 all necessary input parameters. This means that you need to have another program that calls your procedures and functions.
* No hardwired information is allowed in the project, especially IDs. No IDs should be provided manually. Use sequences for keys.
* You will need to call other member’s functions to find hotel IDs (based on address, etc.)
* 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
* Teams with fewer members: If your team has 4 members (either by design, or 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.

**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.

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. That 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.