

University of Ottawa

Faculty of Engineering School of Electrical Engineering and Computer Science

Course Project

<i>Course</i>	CSI2132	Databases I
<i>Academic year</i>	2023-24	
<i>Semester</i>	Winter	
<i>Instructor</i>	Verena Kantere	
<i>Announced</i>	24.01.2024	
<i>Submission of 1st deliverable</i>	16.02.2024	
<i>Submission of 2nd deliverable</i>	31.03.2024	

Groups of 1-3 students may submit common project deliverables

e-Hotels

Five of the most well-known hotel chains, with hotels in more than 14 different locations in North America, have decided to collaborate and develop an application that will allow their customers to easily book rooms in their hotels, seeing room availability in real time. You are requested to develop the database and the application that allows the above.

For every hotel chain we need to know the address of its central offices, the number of its hotels, contact email addresses and phone numbers. The hotels of hotel chains are categorized (e.g. 1-star up to 5-star). For each hotel we need to know the number of rooms, the address of the hotel and contact email and phone numbers for this hotel. For the rooms in a hotel, we need to know their price, all amenities (e.g. TV, air condition, fridge etc), the capacity of the room (e.g. single, double etc), if they have sea view or mountain view, if they can be extended (e.g. adding one more bed) and if there are any problems/damages in the room. For customers we need to store their full name, address and a type of ID, e.g. SSN/SIN/driving licence, the date of their registration into our system. For employees of the hotels, we need to store their full name, address and SSN/SIN. The employees may have various roles/positions in a hotel. Every hotel needs to have a manager. The customers can search for and book rooms through the online application for specific dates. When they check-in the hotel, their room booking is transformed to renting and they can also pay for this renting. The employee that does the check-in for a customer is responsible for transforming the room booking to renting. A customer may

present physically at a hotel without a booking and directly ask for a room. In this case the employee at the hotel can do the renting of the room right away without prior booking. We need to store in the database the history of both bookings and rentings (archives), but we do not need to store the history of payments. Information about an old (archived) room booking/renting must exist in the database, even if information about the room itself or the customer does not exist in the database anymore. We should be able to delete from our database hotel chains, hotels and rooms. We cannot have in the database information about a room without having in the database the information about the corresponding hotel (i.e. the hotel in which the room belongs too). In the same way, we cannot have in the database information about a hotel without having in the database the information about the corresponding hotel chain (i.e. the hotel chain in which the hotel belongs too).

You are requested to do the following:

- 1a. (10%) **ER diagram**: Create the ER diagram that corresponds to the above description.
- 1b. (8%) **Relational database schema**: Create the relational database schema that corresponds to your ER diagram.
- 1c. (7%) **Integrity constraints**: Define the necessary constraints that will ensure the correctness of the database to be created according to your relational database schema. These are primary keys, referential integrity constraints, domain and attribute constraints and user-defined constraints. Be inventive with the definition of user-defined constraints.
- 2a. (10%) **Database implementation**: Implement the database according to your relational database schema and the constraints that you have defined.
- 2b. (5%) **Database population**: Insert in your database data for each one of the 5 hotel chains. Each one of them has at least 8 hotels, which belong to at least 3 categories. Two of the hotels at least should be in the same area. Each hotel should have at least 5 rooms of different capacity. Populate your database with enough data to be able to showcase the execution of queries/triggers/views.
- 2c. (10%) **Database queries**: Implement at least 4 queries of your choice on your database. Implement at least 1 query with aggregation and at least 1 with a nested query.
- 2d. (10%) **Database modifications**: Create the necessary SQL modifications (use queries and especially triggers): Your database should allow insert, delete and update operations of data in your database according to the referential integrity constraints, and moreover, to the user-defined constraints, which you have defined. Implement at least 2 triggers of your choice for this purpose.
- 2e. (5%) **Database indexes**: Implement at least 3 indexes on the relations of your database and justify why you have chosen these indexes: explain what type of queries and data updates you are expecting on your database and how these indexes are useful to accelerate querying of the database.
- 2f. (5%) **Database views**: Implement 2 views. View 1: the first view is the number of available rooms per area. View 2: the second view is the aggregated capacity of all the rooms of a specific hotel. You are welcome to implement more views of your choice.
- 2g. (30%) **Web application**: Design and implement an appropriate User Interface, through which a user will be able to see the available rooms by giving different, multiple and combinations of criteria in order to choose the room that he/she is interested in and book it or rent it. These criteria should be: the dates (start, end) of booking or renting, the room capacity, the area, the hotel chain, the category of the hotel, the total number of rooms in the hotel, the price of the rooms. The user should be able to see the available choices when he/she changes the value of any of these criteria.
 The User Interface should allow the insert/delete/update of all information related to customers, employees, hotels and rooms.
 The user can be either a customer (who will use the interface in order to search for rooms and do bookings) or a hotel employee (who will use the interface to either turn a booking to

renting when a customer checks in the hotel, or do directly a renting when a customer presents physically to the hotel). An employee should be able to insert a customer payment for a renting through the interface.

The user should be able to see in the User Interface the two specific SQL Views implemented in (2f).

The User Interface should be user friendly, meaning that the user is not required to know SQL. All information should be presented to the user through appropriately designed forms. You should use appropriate elements, like drop-down lists, radio buttons etc.

Note: If you decide not to implement the User Interface as described above, but you implement directly on the DBMS the described modifications and queries, you will still receive 40% of the grade of this requirement, i.e. 12% of the total grade of the project instead of 30%.

For the development of your application we suggest that you use the following: PostgreSQL/MySQL for the implementation of the database, Apache Tomcat for client-server communication, PHP or Java for the server side of the application and HTML for the client side of the application. Nevertheless, you are allowed to choose other technologies, too.

DELIVERABLES

1st Deliverable

Please submit a report that includes your answers to requirements 1-3 of the above list. Thus, your report should include the following:

1. The ER diagram. Please include a brief justification.
2. The relational database schema. Please include a brief justification.
3. The constraints that you have defined. Please include a brief justification for each constraint.

2nd Deliverable

Please submit a .zip file that includes the following:

1. A report that includes the following:
 - a. The DBMS and the programming languages that you have used in your implementation of the application.
 - b. Specific steps to guide someone to install your applications
 - c. A list with the DDLs that create your database
2. Your SQL code that supports all the functionalities in your application
3. All the code that is necessary for running your application
4. A video presenting your project. Details for the creation and the content of the video are given in the following.
5. A pdf file with Table 1 filled in (shown at the end of this document).

Video presentation

Create a video file .mp4 that is not more than 30MB and the time length of it is around 10 minutes and at most 15 minutes. You can use any software you want in order to create the video. The video should record your screen and your voice while you are presenting and describing your project.

In the video you are asked to give the following information in the following order:

1. Mention the software technologies that you have used for the creation of your project.
2. Show briefly the relational database schema you have implemented and comment on any differences of this schema and the one you have submitted as part of Deliverable 1.

3. Comment briefly on the major integrity constraints that you have implemented. Justify their choices and show the respective SQL code.
4. Show and describe briefly the data with which you have populated your database. For example, mention how you created the data and what is the size in number of rows per major relation.
5. Show the execution of some SQL queries on your database.
6. Show the execution of some modifications on your database which fire a trigger. Show the SQL code of the trigger and explain how it implements a specific user-defined constraint.
7. Show the SQL code and comment on the indexes you have implemented on your database.
8. Show the SQL code and comment on the views you have implemented on your database.
9. Show the User Interface you have implemented. Show briefly some of the functionalities of the interface: e.g. you can show the insertion/deletion/update of data, you can show the customer and the employee view and explain how they differ. Furthermore, show how a new booking or renting is made, and show how a booking is transformed to a renting when the customer checks in.

Please fill in the following table with a start timestamp in your video for every one of the above requirements. If you have not implemented one of the requirements, note this in the start timestamp column with the text 'not implemented'.

Table 1 Contents of the video

Requirement	Start timestamp
1	
2	
3	
4	
5	
6	
7	
8	
9	