Contents

[Hotel Data 2](#_Toc129117135)

[Comments: 2](#_Toc129117136)

[Assessment Instructions 2](#_Toc129117137)

[Part 1: ERD 2](#_Toc129117138)

[Part 2: Define the Database 3](#_Toc129117139)

[Part 3: Manage the Data 4](#_Toc129117140)

[Part 4: Query the Database 5](#_Toc129117141)

[Submit Your Work 6](#_Toc129117142)

[Criteria 7](#_Toc129117143)

# Hotel Data

See CSV files

### Comments:

* In the database, a reservation can include multiple rooms reserved by the same guest and for the same dates. There are two examples in the table above.
* Adults are guests whose age is 18 or older.
* Children are guests under the age of 18.
* Start date represents the first evening the guest will stay at the hotel.
* End date represents the day the guest plans to check out.

# Assessment Instructions

## Part 1: ERD

Using the data presented above, create an ERD that represents the database.

Specifications for the ERD:

1. The ERD should be created separately from the RDBMS. (E.g., Draw.io; shapes in Google Docs, Word, or PowerPoint; or paper and pencil)
2. The design should be in second normal form.
3. All tables and columns should be named appropriately so that it is easy to map data to the database.
4. The ERD should specify the data type and field size (where appropriate) for each column.
5. The ERD should indicate the nullability of all non-key fields.
6. The ERD should identify primary keys and foreign keys in each table.
7. The ERD should identify all relationships between tables.

Save the ERD as a file named *YourName-HotelERD.jpg* in your GitHub Classroom repository for this assessment. If you choose to use paper and pencil, submit a clear scan or image that is easy to read.

When you have completed the ERD, submit it to your instructor for review, make any necessary changes based on feedback, and resubmit. Once the instructor has approved the ERD, you are ready to move to the next part.

## Part 2: Define the Database

Using your ERD as an example, create a SQL script that performs the following steps:

1. ~~Create a new database named~~*~~YourNameHotelDB~~*~~.~~
2. Drops the existing database, if it exists, so that the script can rebuild a database with the same name.
3. Creates the tables with all appropriate fields, data types, and keys, using the structure indicated in the ERD.
   * **Tip**: Create the primary tables first, followed by the tables with foreign keys.

The script should run from beginning to end without errors.

Name the file *YourName-HotelDB.sql* and save it to your GitHub Classroom repository for this assessment.

## Part 3: Manage the Data

Create a separate SQL script to manage data in the tables.

First, using the data provided in the tables above, create the required SQL statements to populate the tables with the data above.

1. Include your name, address, and phone number in the first record of the table for guests. You may use a fictitious address and phone number, as long as you use your own name.
2. As with the tables, add data to the primary tables before adding data to the tables with foreign keys.

Second, after adding all of the data above, create SQL statements that will delete Jeremiah Pendergrass and his reservations from the database.

1. Deleting data should start with records that reference Jeremiah Pendergrass using a foreign key and then delete the record from the guest table as the last step.
2. The scripts should only delete records related to Jeremiah Pendergrass and his reservations. They should not delete any room data.

The script should run from beginning to end without errors.

Name the file *YourName-HotelData.sql* and save it to your GitHub Classroom repository for this assessment.

You are encouraged to use Excel or another spreadsheet application to generate csv files that you can import into MySQL tables. You can then use MySQL to generate the Insert statements to include in your script.

## Part 4: Query the Database

Create a third SQL script to retrieve the following data from the database.

1. Write a query that returns
   1. a list of reservations that end in July 2023,
   2. including the name of the guest, the room number(s), and the reservation dates.
2. Write a query that returns a list of
   1. all reservations for rooms with a jacuzzi,
   2. displaying the guest's name, the room number, and the dates of the reservation.
3. Write a query that returns
   1. all the rooms reserved for a specific guest,
   2. including the guest's name, the room(s) reserved, the starting date of the reservation, and how many people were included in the reservation.
   3. (Choose a guest's name from the existing data.)
4. Write a query that returns
   1. a list of rooms, reservation ID, and per-room cost for each reservation.
   2. The results should include all rooms, whether or not there is a reservation associated with the room.
5. Write a query that returns
   1. all rooms with a capacity of three or more and
   2. that are reserved on any date in April 2023.
6. Write a query that returns
   1. a list of all guest names and the number of reservations per guest,
   2. sorted starting with the guest with the most reservations and then by the guest's last name.
7. Write a query that
   1. displays the name, address, and phone number of a guest based on their phone number.
   2. (Choose a phone number from the existing data.)

For each query, include:

* The request from this assignment as a comment above the query, including the number
* The query itself
* The results of the query in a comment under the query

Name the file *YourName-HotelQueries.sql* and save it to your GitHub Classroom repository for this assessment.

Submit Your Work

Files

* *YourName-HotelERD.jpg*
* *YourName-HotelDB.sql*
* *YourName-HotelData.sql*
* *YourName-HotelQueries.sql*

## Criteria

All worth 5 marks

|  |  |  |
| --- | --- | --- |
|  | ***Area*** | ***Criteria*** |
| 1 | **ERD** | Second Normal Form - The database design reasonably meets the requirements for second normal form. |
| 2 | Primary Keys - Each table has an appropriate primary key. |
| 3 | Relationships - Relationships between tables are defined correctly, with appropriate foreign keys. |
| 4 | Field Settings - Each field uses a reasonably appropriate data type and field size. |
| 5 | **Database Creation** | Delete Database - The script deletes the database before rebuilding it. |
| 6 | Tables and Fields - The script creates tables and fields that meet the design presented in the ERD. |
| 7 | **Data Script** | Adds data to tables - The script successfully populates all tables and fields, using the data provided. |
| 8 | Primary Keys - Primary key values are appropriate. |
| 9 | Delete Record - The script successfully deletes the requested record. |
| 10 | **Queries** | Query 1 - There is a query that returns a list of reservations that end in July 2023, including the name of the guest, the room number(s), and the reservation dates. |
| 11 | Query 2 - There is a query that returns a list of all reservations for rooms with a jacuzzi, displaying the guest's name, the room number, and the dates of the reservation. |
| 12 | Query 3 - There is a query that returns all the rooms reserved for a specific guest, including the guest's name, the room(s) reserved, the starting date of the reservation, and how many people were included in the reservation. |
| 13 | Query 4 - There is a query that returns a list of rooms, reservation ID, and per-room cost for each reservation. The results should include all rooms, whether or not there is a reservation associated with the room. |
| 14 | Query 5 - There is a query that returns all the rooms accommodating at least three guests and that are reserved on any date in April 2023. |
| 15 | Query 6 - There is a query that returns a list of all guests and the number of reservations per guest, sorted starting with the guest with the most reservations and then by the guest's last name. |
| 16 | Query 7 - There is a query that displays the name, address, and phone number of a guest based on their phone number. |
| 17 | **Code Style** | All code is written and formatted to make it easy for another developer to read, including appropriate indents, object names, and comments. |