|  |  |
| --- | --- |
| **Name** | Sahil Shah |
| **UID No.** | 2021300115 |
| **Experiment No.** | 2 |

|  |  |
| --- | --- |
| **AIM:** | To create a database and populate using SQL commands (with constraints):   * Data Definition Language – Create, Alter, Drop, Rename, Truncate * Constraints – Not Null, Unique Key, Primary Key, Foreign Key, Check, Dropping a Constraint |
| **Program 1** | |
| **PROBLEM STATEMENT:** | To create a Hotel Database Management System on SQL and populate it using the SQL commands. |
| **THEORY:** | Structured Query Language (SQL), as we all know, is the database language using which we can perform certain operations on the existing database. We can also use this language to create a database. SQL uses certain commands like Create, Drop, Insert, etc. to carry out the required tasks.  These SQL commands are mainly categorized into four categories:   1. DDL – Data Definition Language 2. DQL – Data Query Language 3. DML – Data Manipulation Language 4. DCL – Data Control Language   Though many resources claim there to be another category of SQL clauses **TCL – Transaction Control Language**.  **DDL (Data Definition Language):**  DDL or Data Definition Language consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database. DDL is a set of SQL commands used to create, modify, and delete database structures but not data. These commands are normally not used by a general user, who should be accessing the database via an application.  List of DDL commands:   * **CREATE**: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers). * **DROP**: This command is used to delete objects from the database. * **ALTER:**This is used to alter the structure of the database. * **TRUNCATE:**This is used to remove all records from a table, including all spaces allocated for the records are removed. * **RENAME:**This is used to rename an object existing in the database.  1. **The SQL CREATE DATABASE Statement:**   The CREATE DATABASE statement is used to create a new SQL database.  **Syntax:**  CREATE DATABASE *databasename*;  **The SQL CREATE TABLE Statement:**  The CREATE TABLE statement is used to create a new table in a database.  **Syntax:**  CREATE TABLE *table\_name*( *column1 datatype*, *column2 datatype*, *column3 datatype*,    .... );   1. **The SQL DROP DATABASE Statement:**   The DROP DATABASE statement is used to drop an existing SQL database.  **Syntax:**  DROP DATABASE databasename;  **The SQL DROP TABLE Statement:**  The DROP TABLE statement is used to drop an existing table in a database.  **Syntax:**  DROP TABLE table\_name;   1. **SQL ALTER TABLE Statement:**   The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.  The ALTER TABLE statement is also used to add and drop various constraints on an existing table.   * **ALTER TABLE - ADD Column**   To add a column in a table, use the following syntax:  ALTER TABLE table\_name  ADD column\_name datatype;   * **ALTER TABLE - DROP COLUMN**   To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):  ALTER TABLE table\_name  DROP COLUMN column\_name;   * **ALTER TABLE - ALTER/MODIFY COLUMN**   To change the data type of a column in a table, use the following syntax:  ALTER TABLE table\_name  MODIFY column\_name datatype;   1. **SQL TRUNCATE TABLE:**   The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.  **Syntax:**  TRUNCATE TABLE table\_name; |
| **CODE:**  -- Creating the Database  CREATE DATABASE HOTEL;  USE HOTEL;  -- Creating the Table (Hotel)  CREATE TABLE Hotel (  HotelName varchar(255) NOT NULL,  ContactNumber int NOT NULL,  LocationStreetName varchar(255) NOT NULL,  LocationPincode int NOT NULL,  LocationCity varchar(255) NOT NULL,  HotelID int NOT NULL,  Rating int,  PRIMARY KEY (HotelID)  );  -- Inserting a Row into Table  INSERT INTO Hotel VALUES('Aparna', 999999999, 'Salisbury Road', 400987, 'Mumbai', 103, 4);  SELECT \* FROM Hotel;  -- Alter Table  ALTER TABLE Hotel  ADD HotelEmail varchar(255);  SELECT \* FROM Hotel;  -- Creating the Table (Room)  CREATE TABLE Room (  RoomNumber int,  RoomAvailability varchar(5) NOT NULL,  RoomSize varchar(50) NOT NULL,  RoomType varchar(50) NOT NULL,  PRIMARY KEY (RoomNumber),  HotelID int NOT NULL,  FOREIGN KEY (HotelID) REFERENCES Hotel(HotelID)  );  -- Inserting Rows into Table  INSERT INTO Room VALUES(237, 'NO', '2 persons', 'A.C', 103);  INSERT INTO Room VALUES(069, 'YES', '2 persons', 'Deluxe', 103);  INSERT INTO Room VALUES(235, 'YES', '1 person', 'A.C', 103);  INSERT INTO Room VALUES(123, 'YES', '4 persons', 'Non-A.C', 103);  INSERT INTO Room VALUES(420, 'YES', '3 persons', 'A.C', 103);  INSERT INTO Room VALUES(666, 'YES', '3 persons', 'A.C', 103);  SELECT \* FROM Room;  -- Truncate  TRUNCATE TABLE Room;  SELECT \* FROM Room;  -- Rename  RENAME TABLE Hotel TO HotelSahil;  -- Drop  DROP TABLE Room;  SELECT \* FROM Room;  **QUERIES:**  **Using Create, Use, Insert Into, Select commands:**      **Using Alter, Select command:**      **Using Create, Insert Into, Select Commands:**      **Using Truncate, Select command:**      **Using Rename command:**    **Using Drop command:** | |
| **CONCLUSION:**  In this experiment, I learned about the various MySQL commands like Create, Use, Insert Into, Alter, Rename and Alter, constraints like Primary Key, Foreign Key, Not Null, Unique, etc. to implement on MySQL Workbench. Using this knowledge, I created two tables – Hotel and Room and through them established the Foreign Key constraint, added data to them and performed the carious commands. | |