|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Database Management System Laboratory** | **Semester:** | **IV** |
| **Date of Performance:** | **25 / 01 / 2023** | **Batch No:** | **A – 2** |
| **Faculty Name:** | **Prof. Shila Dhande** | **Roll No.:** | **16014022050** |
| **Faculty Sign & Date:** |  | **Grade / Marks:** | **\_\_\_ / 25** |

**Experiment No.: 3**

**Title: Implementation of Database in SQL - DDL**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| **Aim:** Implementation of Database in SQL – DDL.  **Objective:** To define/modify database definitions with proper constraints. |

|  |
| --- |
| **COs to be achieved:** |
| **CO2:** Convert entity-relationship diagrams into relational tables, populate a relational  database and formulate SQL queries on the data Use SQL for creation and query the database.  **CO3:** Define and apply integrity constraints and improve database design using normalization techniques. |

|  |
| --- |
| **Books / Journals / Websites Referred:** |
| 1. G. K. Gupta:” Database Management Systems”, McGraw – Hill 2. Korth, Slberchatz, Sudarshan: “Database Systems Concept”, 6th Edition, McGraw Hill 3. Elmasri and Navathe, “Fundamentals of Database Systems”, 5th Edition, PEARSON Education. |

|  |
| --- |
| **Tools Required:** |
| * Postgresql Software |

|  |
| --- |
| **Theory:** |
| The set of relations in a database must be specifies to the system by means of a data definition language (DDL). The SQL DDL allows specification of not only a set of relations but also specific information about the relation including,   1. The schema for each relation 2. The domain of values associated with each attribute 3. The integrity constraints 4. The set of indices to be maintained for each relation 5. The security and authorization information for each relation 6. The physical storage structure of each relation on disk   **Syntax Create Table:**   * create table employee(ssn,fname varchar(10), mname varchar(10), lname varchar(10), desg varchar(20), gender varchar(5), addr varchar(20), bdate datetime, sal float,primary key(ssn)); * create table manages(ssn int, dept\_code int, start\_dt datetime, foreign key(ssn) references employee, foreign key(dept\_code) refrences department, key(ssn,dept\_code) ) on delete set null;primary   **Data Constraints:**  Business managers of the organization determine a set of rules that must be applied before the data is stored in the database. The application of such rules on raw data ensures **data integrity**.  **e.g.:** An employee belonging to Sales department cannot have salary higher than Rs. 1000.  An employee has a unique identification number.  **Applying Data Constraints:**  Oracle permits data constraints to be attached to table columns using SQL syntax. Constraints can be attached to table columns using,  Alter table, Unique Constraint Unique Constraint - At column level Syntax  **<ColumnName><Datatype>(<size>)**  **UNIQUE Unique Constraint- At table level**  **CREATE TABLE<TableName> (**  **<ColumnName><Datatype>(<size>)**  **<ColumnName><Datatype>(<size>)**  **<Columnname><Datatype>(<size>)**  **UNIQUE(<ColumnName1>, <ColumnName2>));** |

|  |
| --- |
| **Implementation Details (Problem Statement, Query and Screenshots of Results):** |
| **Problem Statement:**  Hotel Management System is a hotel reservation site script where site users will be able to search rooms availability with an online booking reservations system. Site users can also browse hotels, view room inventory, check availability, and book reservations in real-time. Site users enter check in date and check out date then search for availability and rates. After choosing the right room in the wanted hotel all booking and reservation process is done on the site and an SMS is sent to confirm the booking.  **Query:**   1. **Rooms Table –**   DROP TABLE IF EXISTS Rooms;  CREATE TABLE Rooms  (  room\_id int PRIMARY KEY,  room\_type text,  room\_no int NOT NULL,  room\_category text  );  INSERT INTO Rooms VALUES(1101, 'Studio Room', 100, 'AC');  INSERT INTO Rooms VALUES(1201, 'Single Room', 101, 'AC');  INSERT INTO Rooms VALUES(1300, 'Double Room', 102, 'Non-AC');  INSERT INTO Rooms VALUES(1400, 'Triple Room', 103, 'Non-AC');  SELECT \* FROM Rooms;  DROP TABLE Rooms;     1. **Booking Table –**   CREATE TABLE Booking  (  booking\_iD int PRIMARY KEY,  booking\_start\_date text NOT NULL,  booking\_end\_date DATE NOT NULL,  );  INSERT INTO Booking VALUES(9876, '2022-09-12' , '2022-09-17');  INSERT INTO Booking VALUES(9875, '2022-09-14' , '2022-09-22');  INSERT INTO Booking VALUES(9874, '2022-09-15' , '2022-09-20');  INSERT INTO Booking VALUES(9873, '2022-09-15' , '2022-09-21');  INSERT INTO Booking VALUES(9872, '2022-09-16' , '2022-09-23');  SELECT \* FROM BOOKING;  DROP TABLE BOOKING;     1. **Payment Table –**   CREATE TABLE Payment  (  pay\_id int PRIMARY KEY,  pay\_cust\_id int UNIQUE NOT NULL,  pay\_amount numeric NOT NULL  );  INSERT INTO Payment VALUES(4321 , 1234 , 4320.00);  INSERT INTO Payment VALUES(4322 , 1235 , 2543.20);  INSERT INTO Payment VALUES(4323 , 1236 , 3456.00);  INSERT INTO Payment VALUES(4324 , 1237 , 3290.70);  INSERT INTO Payment VALUES(4325 , 1238 , 10000.00);  SELECT \* FROM Payment;  DROP TABLE Payment;     1. **Services Table –**   CREATE TABLE Services  (  service\_id int PRIMARY KEY,  service\_type text,  service\_description text,  service\_name text,  room\_id int REFERENCES rooms(room\_id)  );  INSERT INTO Services VALUES(9201, 'Spa', 'Head and Back Massage', 'Thai Massage');  INSERT INTO Services VALUES(9202, 'Housekeeping', 'Room Cleaning', '-' );  INSERT INTO Services VALUES(9203, 'Laundry', '-' , 'Dry Cleaning');  INSERT INTO Services VALUES(9204, 'Cart Service', 'French Food service' , 'Ratatouille');  INSERT INTO Services VALUES(9205, '-' , '-' , '-' );  SELECT \* FROM Services;  DROP TABLE Services;     1. **Hotel Table –**   CREATE TABLE Hotel  (  hotel\_id int PRIMARY KEY NOT NULL,  hotel\_name text NOT NULL,  hotel\_address text NOT NULL,  hotel\_rent numeric,  hotel\_pin numeric,  hotel\_description text,  );  INSERT INTO Hotel VALUES(3001, 'Taj Hotel', 'The Taj Mahal Palace & Tower, Apollo Bunder', 4320.00, 400001, 'Welcome and enjoy your stay with taj hotel');  INSERT INTO Hotel VALUES(3002, 'Taj Hotel', 'The Taj Mahal Palace & Tower, Apollo Bunder', 2543.20, 400001, 'Welcome and enjoy your stay with taj hotel');  INSERT INTO Hotel VALUES(3003, 'Taj Hotel', 'The Taj Mahal Palace & Tower, Apollo Bunder', 3456.00, 400001, 'Welcome and enjoy your stay with taj hotel');  INSERT INTO Hotel VALUES(3004, 'Taj Hotel', 'The Taj Mahal Palace & Tower, Apollo Bunder', 3290.70, 400001, 'Welcome and enjoy your stay with taj hotel');  INSERT INTO Hotel VALUES(3005, 'Taj Hotel', 'The Taj Mahal Palace & Tower, Apollo Bunder', 10000.00, 400001,'Welcome and enjoy your stay with taj hotel');  SELECT \* FROM Hotel;  DROP TABLE Hotel;     1. **Customer Table –**   CREATE TABLE Customer  (  customer\_id int PRIMARY KEY NOT NULL,  customer\_name text NOT NULL,  customer\_mobile varchar[10] NOT NULL,  customer\_address text NOT NULL,  customer\_email text,  );  INSERT INTO Customer VALUES(7001; 'Ketaki', {1234567890}, 'Thane (W)', 'ketaki.mahajan@somaiya.edu');  INSERT INTO Customer VALUES(7002, 'Ibrahim', {9876543210}, 'Somaiya Hostel', 'mohd.mir@somaiya.edu');  INSERT INTO Customer VALUES(7003, 'Priyanshu', {2468013579}, 'Kalyan', 'priyanshu.m@somaiya.edu');  SELECT \* FROM Customer;  DROP TABLE Customer; |

|  |
| --- |
| **Post Lab Subjective / Objective Type Questions:** |
| 1. **Which command is used for removing a table and all its data from the database:**    1. **DROP Command**    2. **TRUNCATE Command**    3. **Both Commands** 2. **For the given ER model, using DDL command:**   **Write syntax to create CREATE Tables with all possible integrity constraints.**  **A small accounting firm wants a simple HR application that will help it to keep track of its employees, their positions, allowances, salary scales, and which company vehicles their employees drive. The application must keep track of all the positions at the firm, the employees filling these positions, the allowances for these positions, the salary scales for these positions, and the company vehicles assigned to these positions.**  Case Study 1 detailed E-R diagram  **-- Positions Table**  CREATE TABLE Positions (  PositionId INT PRIMARY KEY,  PositionName VARCHAR(255) NOT NULL,  PositionDescription TEXT,  Details TEXT  );  **-- Allowances Table**  CREATE TABLE Allowances (  AllowanceId INT PRIMARY KEY,  AllowanceName VARCHAR(255) NOT NULL,  AllowanceDescription TEXT,  Amount DECIMAL(10, 2) NOT NULL,  PositionId INT,  FOREIGN KEY (PositionId) REFERENCES Positions(PositionId)  );  **-- Employees Table**  CREATE TABLE Employees (  EmployeeId INT PRIMARY KEY,  SSNumber VARCHAR(15) UNIQUE NOT NULL,  LastName VARCHAR(255) NOT NULL,  FirstName VARCHAR(255) NOT NULL,  Gender CHAR(1),  DOB DATE,  PositionId INT,  FOREIGN KEY (PositionId) REFERENCES Positions(PositionId)  );  **-- Vehicles Table**  CREATE TABLE Vehicles (  VehicleId INT PRIMARY KEY,  VIN VARCHAR(17) NOT NULL,  RegistrationNo VARCHAR(255) NOT NULL,  Year INTEGER,  Make VARCHAR(255),  Model VARCHAR(255),  Color VARCHAR(255),  PositionId INT,  FOREIGN KEY (PositionId) REFERENCES Positions(PositionId)  );  **-- SalaryScales Table**  CREATE TABLE SalaryScales (  SalaryScalesCode INT PRIMARY KEY,  SalaryScalesName VARCHAR(255) NOT NULL,  SalaryScalesDescription TEXT,  MinimumSalary DECIMAL(10, 2) NOT NULL,  MaximumSalary DECIMAL(10, 2) NOT NULL,  PositionId INT,  FOREIGN KEY (PositionId) REFERENCES Positions(PositionId)  ); |

|  |
| --- |
| **Conclusion:** |
| In this experiment, we effectively converted entity-relationship diagrams into relational tables, populated a relational database, and proficiently devised SQL queries for data manipulation. This exercise enhanced our ability to construct well-organized databases and sharpened our proficiency in utilizing SQL for both database creation and querying. |

**Signature of faculty in-charge with Date:**