# SQL

Assignment

Create 4 table

|  |
| --- |
| 1. **Create Company Table** |
| CREATE TABLE Company (  CompanyID INT PRIMARY KEY AUTO\_INCREMENT,  CompanyName varchar (50),  Street varchar (50),  City varchar (50),  State varchar (50),  zip varchar (12),  ) |

|  |
| --- |
| **2. Create Contact table** |
| CREATE TABLE Contact(  ContectID INT PRIMARY KEY AUTO\_INCREMENT,  CompanyID INT ,  FOREIGN KEY (CompanyID) REFERENCES company(CompanyID),  FirstName varchar (45),  lastName varchar(45),  Street varchar(45),  City varchar(45),  State varchar(2),  Zip varchar(10),  isMain boolean,  Email varchar(45),  Phone varchar(12)  ); |
| 1. **Create ContactEmployee Table** |
| CREATE TABLE ContactEmployee (  ContactEmployeeID int PRIMARY KEY AUTO\_INCREMENT,  ContactID int,  EmployeeID int,  FOREIGN KEY (EmployeeID) REFERENCES employee(EmployeeID),  ContactDate Date ,  Description varchar(50)  ) |
| 1. **Create Employee Table** |
| CREATE TABLE Employee (  EmployeeID int PRIMARY KEY AUTO\_INCREMENT,  Firstname varchar (50),  Lastname varchar (50),  Salary decimal(10,2)  HireDate Date ,  JobTitle varchar(50),  EmailID varchar(50),  Phone varchar(12),  ) |

Q1) Statement to create the Contact table.

Ans: CREATE TABLE Contact(

ContectID INT PRIMARY KEY AUTO\_INCREMENT,

CompanyID INT ,

FOREIGN KEY (CompanyID) REFERENCES company(CompanyID),

FirstName varchar (45),

lastName varchar(45),

Street varchar(45),

City varchar(45),

State varchar(2),

Zip varchar(10),

isMain boolean,

Email varchar(45),

Phone varchar(12)

);

Q2)Statement to create the Employee table.

Ans:CREATE TABLE Employee(

EmployeeID int PRIMARY KEY AUTO\_INCREMENT,

FirstName varchar(45),

LastName varchar(45),

Salary decimal(10,2),

HireDate date,

JobTitle varchar(25),

Email varchar(45),

Phone varchar(12)

);

Q3)Statement to create the ContactEmployee table.

Ans:CREATE TABLE ContactEmployee(

ContactEmployeeID int PRIMARY KEY AUTO\_INCREMENT,

ContactID int,

EmployeeID int,

ContactDate date,

Description varchar(100)

);

Q4)In the Employee table, the statement that changes Lesley Bland’s phone number to 215-555-8800.

Ans: UPDATE employee SET Phone='215-555-8800' WHERE FirstName='Lesley' AND LastName='Bland';

Q5)In the Company table, the statement that changes the name of “Urban Outfitters, Inc.” to “Urban Outfitters”.

Ans:UPDATE company SET CompanyName='Urban Outfitters' WHERE CompanyID=1;

Q6)In ContactEmployee table, the statement that removes Dianne Connor’s contact event with Jack Lee (one statement).

Ans:DELETE FROM ContactEmployee WHERE EmployeeID = (SELECT EmployeeID FROM Employee WHERE Name = 'Dianne Connor') AND ContactID = (SELECT ContactID FROM Contact WHERE Name = 'Jack Lee');

Q7)Write the SQL SELECT query that displays the names of the employees that have contacted Toll Brothers (one statement). Run the SQL SELECT query in MySQL Workbench. Copy the results below as well.

Ans:

SELECT employee.FirstName, employee.LastName FROM employee

JOIN contactemployee on employee.EmployeeID = contactemployee.EmployeeID

JOIN contact on contactemployee.ContactID = contact.ContectID

JOIN company on contact.CompanyID = company.CompanyID

WHERE company.CompanyName='Toll Brothers';

Q8)What is the significance of “%” and “\_” operators in the LIKE statement?

Ans: 1. % (Percentage Sign):

Significance: The % wildcard represents zero or more characters.

Usage: It is used to match any sequence of characters (including no characters).

2. \_ (Underscore):

Significance: The \_ wildcard represents exactly one character.

Usage: It is used to match any single character at a specific position.

Q9)Explain normalization in the context of databases.

Ans: Normalization in the context of databases is the process of organizing data to reduce redundancy (duplicate data) and improve data integrity. The goal is to structure the data so that it is efficient to store, update, and retrieve, and to avoid anomalies (errors) during these operations.

Key Points of Normalization:

a. Reduce Duplication: By breaking down large tables into smaller, related tables, you minimize the repetition of data.

b. Organize Data Logically: Data is grouped into tables based on their relationships. This logical organization makes it easier to understand and work with the data.

c. Improve Data Integrity: With normalization, each piece of data is stored in one place, making it easier to update without errors.

Q10)What does a join in MySQL mean?

Ans: In MySQL, a join is an operation that allows you to combine data from two or more tables based on a related column between them. Joins are used to retrieve related data stored across multiple tables, making it easier to query and work with that data as if it were in a single table.

Q11) What do you understand about DDL, DCL, and DML in MySQL?

Ans:1)DDL (Data Definition Language):

- DDL commands are used to define, alter, and manage the structure of database objects like tables, indexes, and schemas. Examples CREATE,ALTER and DROP.

2)DCL (Data Control Language):

-DCL commands are used to control access to data within the database. These commands manage permissions and control who can access or manipulate data.

Examples GRANT and REVOKE.

3. DML (Data Manipulation Language):

- DML commands are used to manipulate the data stored within the database tables. These commands allow you to insert, update, delete, and retrieve data.

Examples SELECT,UPDATE,DELETE and INSERT.

Q12)What is the role of the MySQL JOIN clause in a query, and what are some common types of joins?

Ans: Role of the MySQL JOIN Clause:

a. Combining Data: The JOIN clause is essential for combining data from multiple tables, enabling you to perform complex queries that involve multiple sources of information.

b. Retrieving Related Data: It allows you to retrieve related data stored across different tables by linking them through a common column, typically a key column like ID.

c. Querying Relationships: JOINs are used to query the relationships between tables, such as customers and their orders, employees and their departments, or students and their courses.

Types of Joins:

1.Inner Join :- join common column from both tables.

2.Left Join :- join left table and common column from both tables.

3.Right Join :- join right table and common column from both tables

4.Full Join :- join both tables