SQL assignment(module 4)

Q1 Statement to create the Contact table.

```
Ans CREATE TABLE Contact (
  ContactID INT PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Email VARCHAR(100),
   PhoneNumber VARCHAR(15),
   Address VARCHAR(255)
);
```

Q2 Statement to create the Employee table.

```
Ans. CREATE TABLE Employee (
   EmployeeID INT PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Email VARCHAR(100),
   PhoneNumber VARCHAR(15),
   JobTitle VARCHAR(50),
   Department VARCHAR(50),
   Salary DECIMAL(10,2),
   HireDate DATE
);
```

Q3 Statement to create the ContactEmployee table HINT: Use DATE as the datatype for ContactDate. It allows you to store the date in this format: YYYY-MM-DD (i.e., ‘2014-03-12’ for March 12, 2014).

```
Ans. CREATE TABLE ContactEmployee (
   ContactEmployeeID INT PRIMARY KEY,
   ContactID INT,
   EmployeeID INT,
   ContactDate DATE,
   Notes VARCHAR(255),
   FOREIGN KEY (ContactID) REFERENCES Contact(ContactID),
   FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);
```

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 empid = 106;
SELECT * FROM Employee WHERE empid = 106;
```

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

```
Ans. UPDATE Company
```

```
SET comp_name = 'Urban Outfitters'

WHERE companyid = 6;

select * FROM company WHERE companyid = 6;
```

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

-- HINT: Use the primary key of the ContactEmployee table to specify the correct record to remove.

```
Ans. DELETE FROM ContactEmployee

WHERE contactempid = 8;

select * FROM contactEmployee;
```

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 CONCAT (e.first_name, ' ' ,e.last_name) as employeeName, c.comp_name, ce.contactdate

FROM contactEmployee ce, Employee e, company c

WHERE ce.empid = e.empid

 AND c.companyid = c.companyid

 AND c.comp_name = 'Toll brothers';
```

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

Ans. LIKE is used in SQL to search for a specific pattern in a column — often with wildcard characters.

The two most common wildcards are % and _.

1) The % (percent) wildcard

- Represents zero, one, or many characters.
- It’s like saying “anything can go here.”

Examples:

Query	Meaning
WHERE Name LIKE 'A%'	Finds all names starting with A (e.g., Anna, Ajay, Amit)
WHERE Name LIKE '%n'	Finds all names ending with n (e.g., John, Rohan)
WHERE Name LIKE '%an%'	Finds all names that contain “an” anywhere (e.g., Anita, Ranjan)

---

2) The _ (underscore) wildcard

- Represents exactly one character.

Examples:

Query	Meaning
WHERE Name LIKE '_im'	Finds all 3-letter names with ‘im’ at the end (e.g., Kim, Tim)
WHERE Name LIKE 'A_'	Finds all 2-letter names starting with A (e.g., An, Al)
WHERE Name LIKE 'a'	Finds all 3-letter names with ‘a’ in the middle (e.g., Sam, Raj)

Q9) Explain normalization in the context of databases.

Ans. Normalization is a process in database design that organizes data to reduce redundancy and improve data integrity. The main objective of normalization is to structure a database in such a way that each piece of data is stored only once, and data dependencies are logical. This makes data storage more efficient, consistent, and easier to maintain.

Normalization Process (Normal Forms)

Normalization is usually done in steps, called **Normal Forms (NFs)**. Each normal form has specific rules that must be followed to make the database more organized.

1. First Normal Form (1NF)

- Each column should contain only **atomic (single)** values — no repeating groups or arrays.
- Each record must be unique.

2. Second Normal Form (2NF)

- The table must be in **1NF**.
- There should be **no partial dependency**, meaning that every non-key attribute must depend on the **whole primary key**, not just part of it.

3. Third Normal Form (3NF)

- The table must be in **2NF**.
- There should be **no transitive dependency**, meaning that non-key attributes should depend **only on the primary key** and not on other non-key attributes.

4. Boyce-Codd Normal Form (BCNF)

- BCNF is a stronger version of 3NF.

- It removes remaining anomalies by ensuring that **for every functional dependency A → B, A must be a super key**.
- It handles cases where 3NF still allows certain types of dependency problems.
-

5 . fifth normal form (5NF)

- Fifth Normal Form addresses **join dependencies**4ensuring that data doesn't
- get duplicated when tables are rejoined after decomposition.

10) What does a join in MySQL mean?

Ans. A **JOIN** combines rows from two or more tables into a single result set based on a related column. It lets you answer questions that need data from multiple tables — for example, “which employees contacted which company?” or “show orders with customer names.

Why we use joins :

To combine data	JOINs allow you to get data from multiple tables in a single query.
To avoid data duplication	Instead of storing the same data in one table, we keep related data separately and use JOINs when needed.
To make queries more powerful	You can get detailed, meaningful results (like “Employee name with Company name”) easily.

To follow normalization	Data stays clean and organized across different tables, and JOINs connect them when necessary.

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

Ans. **1) DDL – Data Definition Language**
DDL commands are used to **define, create, and change the structure** of the database — like tables, schemas, or views.
These commands affect the **structure**, not the data inside the table.

Main DDL Commands:

CREATE

ALTER

DROP

TRUNCATE

RENAME

2) DML – Data Manipulation Language
DML commands are used to manage or modify data stored inside tables.
It deals with data, not the structure.

Main DML Commands:

INSERT

Update

Delete

Select

3) DCL – Data Control Language
DCL commands are used to control access and permissions to the database.
They decide who can read, modify, or delete data.

Main DCL Commands:

GRANT

REVOKE

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

Ans. The JOIN clause in MySQL is used to combine data from two or more tables based on a related column between them (usually a foreign key).

It helps us to get complete information that is stored in different tables.
Since data in databases is often split into multiple related tables, JOIN connects them logically.

Join Type		
1. INNER JOIN	Returns only the matching rows from both tables (common data).	Show employees who have a matching company ID in the Company table.
2. LEFT JOIN (or LEFT OUTER JOIN)	Returns all rows from the left table and matching rows from the right table. If no match, shows NULL.	Show all employees, even those who don’t belong to any company.

Join Type		
3. RIGHT JOIN (or RIGHT OUTER JOIN)	Returns all rows from the right table and matching rows from the left table. If no match, shows NULL.	Show all companies, even if no employees are assigned.
4. FULL JOIN (or FULL OUTER JOIN)	Returns all rows from both tables, matching or not.	Show all employees and all companies (available in some SQL versions, not directly in MySQL).
5. CROSS JOIN	Returns every possible combination of rows from both tables.	If one table has 3 rows and another has 2, the result will have 3×2 = 6 rows.