**DATE= ‘YYYY-MM-DD’**

**Schema Overview**

**Before running the queries, ensure your tables are set up as follows:**

**sql**

**Copy code**

**CREATE TABLE Student (**

**s\_id INT PRIMARY KEY,**

**Drive\_id INT,**

**T\_id INT,**

**s\_name VARCHAR(100),**

**CGPA DECIMAL(3,2),**

**s\_branch VARCHAR(50),**

**s\_dob DATE,**

**FOREIGN KEY (Drive\_id) REFERENCES PlacementDrive(Drive\_id) ON DELETE CASCADE,**

**FOREIGN KEY (T\_id) REFERENCES Training(T\_id) ON DELETE CASCADE**

**);**

**CREATE TABLE PlacementDrive (**

**Drive\_id INT PRIMARY KEY,**

**Pcompany\_name VARCHAR(100),**

**package DECIMAL(10,2),**

**location VARCHAR(100)**

**);**

**CREATE TABLE Training (**

**T\_id INT PRIMARY KEY,**

**Tcompany\_name VARCHAR(100),**

**T\_Fee DECIMAL(10,2),**

**T\_date DATE**

**);**

**Queries**

**Find the Student details and Placement details using NATURAL JOIN.  
sql  
Copy code  
SELECT \***

**FROM Student**

**NATURAL JOIN PlacementDrive;**

**Find all the student details with company\_name who have conducted the same drive.  
sql  
Copy code  
SELECT s.\*, pd.Pcompany\_name**

**FROM Student s**

**JOIN PlacementDrive pd ON s.Drive\_id = pd.Drive\_id;**

**List all the Student name and Student branch of Student having package 5 LPA.  
sql  
Copy code  
SELECT s.s\_name, s.s\_branch**

**FROM Student s**

**JOIN PlacementDrive pd ON s.Drive\_id = pd.Drive\_id**

**WHERE pd.package = 500000;  -- Assuming 5 LPA is 500000**

**List all the student names and company\_name having T\_fee more than 20000.  
sql  
Copy code  
SELECT s.s\_name, t.Tcompany\_name**

**FROM Student s**

**JOIN Training t ON s.T\_id = t.T\_id**

**WHERE t.T\_Fee > 20000;**

**Display all training details attended by “Shantanu” in the year 2011.  
sql  
Copy code  
SELECT t.\***

**FROM Training t**

**JOIN Student s ON t.T\_id = s.T\_id**

**WHERE s.s\_name = 'Shantanu' AND YEAR(t.T\_date) = 2011;**

**List the total number of companies who conducted training before 2015.  
sql  
Copy code  
SELECT COUNT(DISTINCT Tcompany\_name) AS TotalCompanies**

**FROM Training**

**WHERE T\_date < '2015-01-01';**

**List the student names with company ‘Microsoft’ and location ’Thane’.  
sql  
Copy code  
SELECT s.s\_name**

**FROM Student s**

**JOIN PlacementDrive pd ON s.Drive\_id = pd.Drive\_id**

**WHERE pd.Pcompany\_name = 'Microsoft' AND pd.location = 'Thane';**

**Find the names of all Students who joined ‘Microsoft’ training in 2015.  
sql  
Copy code  
SELECT s.s\_name**

**FROM Student s**

**JOIN Training t ON s.T\_id = t.T\_id**

**WHERE t.Tcompany\_name = 'Microsoft' AND YEAR(t.T\_date) = 2015;**

**Create a view showing the Student and Training details.  
sql  
Copy code  
CREATE VIEW StudentTrainingDetails AS**

**SELECT s.s\_id, s.s\_name, s.CGPA, t.Tcompany\_name, t.T\_Fee, t.T\_date**

**FROM Student s**

**JOIN Training t ON s.T\_id = t.T\_id;**

1. **Perform Manipulation on simple view - Insert, Update, Delete, Drop View.**

**Insert (Note: This may not work as inserting into views is limited; you might need to insert into the underlying tables instead):  
sql  
Copy code  
INSERT INTO Student (s\_id, Drive\_id, T\_id, s\_name, CGPA, s\_branch, s\_dob)**

**VALUES (11, 2, 1, 'New Student', 8.5, 'Computer Science', '2000-06-15');**

**Update (Modify details in the view, which affects the underlying tables):  
sql  
Copy code  
UPDATE Student**

**SET CGPA = 9.0**

**WHERE s\_name = 'Shantanu';**

**Delete (Remove records; this will delete from the underlying table):  
sql  
Copy code  
DELETE FROM Student**

**WHERE s\_name = 'Old Student';**

**Drop View:  
sql  
Copy code  
DROP VIEW StudentTrainingDetails;**

**Important Notes**

* **Referential Integrity: Ensure that foreign keys are respected when inserting or deleting records.**
* **View Manipulations: Views are often read-only if they involve multiple tables or complex queries. Simple updates may only work if the view is based on a single table without aggregates or groupings.**
* **Date Functions: The YEAR() function is used to extract the year from a date in MySQL.**

**Make sure to adjust any conditions based on your actual data. If you run into any errors or need further explanations, feel free to ask!**

**Table Schema (for Reference)**

sql

Copy code

CREATE TABLE Employee (

    Emp\_id INT PRIMARY KEY,

    Dept\_id INT,

    Emp\_fname VARCHAR(50),

    Emp\_lname VARCHAR(50),

    Emp\_Position VARCHAR(50),

    Emp\_salary DECIMAL(10, 2),

    Emp\_JoinDate DATE

);

CREATE TABLE Dept (

    Dept\_id INT PRIMARY KEY,

    Dept\_name VARCHAR(50),

    location VARCHAR(50)

);

CREATE TABLE Project (

    Proj\_id INT PRIMARY KEY,

    Dept\_id INT,

    Proj\_Name VARCHAR(50),

    Proj\_Location VARCHAR(50),

    Proj\_cost DECIMAL(10, 2),

    Proj\_year INT

);

**Queries**

**Find Employee details and Department details using NATURAL JOIN.**sql  
Copy code  
SELECT \*

FROM Employee

NATURAL JOIN Dept;

**Find the emp\_fname, Emp\_position, location, Emp\_JoinDate who have the same Dept\_id.**sql  
Copy code  
SELECT Emp\_fname, Emp\_position, location, Emp\_JoinDate

FROM Employee e

JOIN Dept d ON e.Dept\_id = d.Dept\_id;

**Find the Employee details, Proj\_id, and Project cost for employees whose Project location is not ‘Hyderabad’.**sql  
Copy code  
SELECT e.\*, p.Proj\_id, p.Proj\_cost

FROM Employee e

JOIN Project p ON e.Dept\_id = p.Dept\_id

WHERE p.Proj\_Location <> 'Hyderabad';

CORRECT

SELECT e.Emp\_id, e.Emp\_fname, e.Emp\_lname, COUNT(p.Proj\_id) AS Project\_Count

FROM Employee e

JOIN Project p ON e.Dept\_id = p.Dept\_id

WHERE p.Proj\_Location <> 'Hyderabad'

GROUP BY e.Emp\_id, e.Emp\_fname, e.Emp\_lname;

**Find Department Name, employee name, and Emp\_position for projects in the year 2020.**sql  
Copy code  
SELECT d.Dept\_name, e.Emp\_fname, e.Emp\_Position

FROM Employee e

JOIN Dept d ON e.Dept\_id = d.Dept\_id

JOIN Project p ON d.Dept\_id = p.Dept\_id

WHERE p.Proj\_year = 2020;

**Display emp\_position and Dept\_name for employees with Project cost > 30000.**sql  
Copy code  
SELECT e.Emp\_Position, d.Dept\_name

FROM Employee e

JOIN Dept d ON e.Dept\_id = d.Dept\_id

JOIN Project p ON d.Dept\_id = p.Dept\_id

WHERE p.Proj\_cost > 30000;

**Find the names of all Projects that started in the year 2015.**sql  
Copy code  
SELECT Proj\_Name

FROM Project

WHERE Proj\_year = 2015;

**List the Dept\_name where the number of employees is 10.**sql  
Copy code  
SELECT d.Dept\_name

FROM Dept d

JOIN Employee e ON d.Dept\_id = e.Dept\_id

GROUP BY d.Dept\_id, d.Dept\_name

HAVING COUNT(e.Emp\_id) = 10;

**Display the total number of employees who joined any project before 2009.**sql  
Copy code  
SELECT COUNT(DISTINCT e.Emp\_id) AS total\_employees

FROM Employee e

JOIN Project p ON e.Dept\_id = p.Dept\_id

WHERE p.Proj\_year < 2009;

**Create a view showing the employee and Department details.**sql  
Copy code  
CREATE VIEW emp\_dept\_view AS

SELECT e.Emp\_id, e.Emp\_fname, e.Emp\_lname, e.Emp\_Position, e.Emp\_salary, e.Emp\_JoinDate, d.Dept\_name, d.location

FROM Employee e

JOIN Dept d ON e.Dept\_id = d.Dept\_id;

1. **Perform manipulation on the view: Insert, Update, Delete, Drop View.**

* **Insert**: You cannot directly insert into a view that includes joins or aggregates. For simple views, this is possible, but with some restrictions.

**Update**:  
sql  
Copy code  
UPDATE emp\_dept\_view

SET Emp\_salary = Emp\_salary + 5000

WHERE Emp\_id = 1;

**Delete**:  
sql  
Copy code  
DELETE FROM emp\_dept\_view

WHERE Emp\_id = 2;

* **Drop View**:  
  sql  
  Copy code  
  DROP VIEW emp\_dept\_view;