```
-- Department Table (4 columns)

CREATE TABLE Department (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(50),
    location VARCHAR(50),
    manager_id INT
);

-- Employee Table (7 columns)
CREATE TABLE Employee (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(50),
    job_title VARCHAR(50),
    salary DECIMAL(10,2),
    hire_date DATE,
    dept_id INT,
    manager_id INT,
    FOREIGN KEY (dept_id) REFERENCES Department(dept_id)
);

-- Emp_Personal_Detail Table (8 columns)
CREATE TABLE Emp_Personal_Detail (
    detail_id INT PRIMARY KEY,
    emp_id INT,
    gender CHAR(1),
    dob DATE,
    phone VARCHAR(15),
    email VARCHAR(50),
    city VARCHAR(50),
    marital_status VARCHAR(20),
```

```
    FOREIGN KEY (emp_id) REFERENCES Employee(emp_id)
);

SELECT * FROM Emp_Personal_Detail;
```

Q1. Find The Highest Paid Employee In Each Department Along With Department Name.

```
SELECT E.EMP_ID, E.EMP_NAME, E.SALARY, D.DEPT_NAME
FROM EMPLOYEE E
JOIN DEPARTMENT D ON E.DEPT_ID = D.DEPT_ID
WHERE E.SALARY = (SELECT MAX(E2.SALARY) FROM EMPLOYEE E2
WHERE E2.DEPT_ID = E.DEPT_ID);
```

Q2. List Employees Who Earn More Than Their Manager's Salary.

```
SELECT E.EMP_ID, E.EMP_NAME, E.SALARY, M.EMP_NAME AS MANAGER_NAME, M.SALARY AS
MANAGER_SALARY
FROM EMPLOYEE E
JOIN EMPLOYEE M ON E.MANAGER_ID = M.EMP_ID
WHERE E.SALARY > M.SALARY;
```

Q3. Show Department(S) Where The Average Salary Of Employees Is Greater Than The Overall Company Average Salary.

```
SELECT D.DEPT_ID, D.DEPT_NAME, AVG(E.SALARY) AS AVG_DEPT_SALARY
FROM EMPLOYEE E
JOIN DEPARTMENT D ON E.DEPT_ID = D.DEPT_ID
GROUP BY D.DEPT_ID, D.DEPT_NAME
HAVING AVG(E.SALARY) > (SELECT AVG(SALARY) FROM EMPLOYEE);
```

Q4. Find Employees Whose Age Is More Than 30 And Are Working In The Same City As Their Department's Location.

```
SELECT E.EMP_ID, E.EMP_NAME, TIMESTAMPDIFF(YEAR, PD.DOB, CURDATE()) AS AGE, PD.CITY, D.LOCATION
FROM EMPLOYEE E
JOIN EMP_PERSONAL_DETAIL PD ON E.EMP_ID = PD.EMP_ID
JOIN DEPARTMENT D ON E.DEPT_ID = D.DEPT_ID
WHERE TIMESTAMPDIFF(YEAR, PD.DOB, CURDATE()) > 30 AND PD.CITY = D.LOCATION;
```

Q5. Display Employees Who Are The Only Employee In Their Department (Single Employee Department).

```
SELECT * FROM EMPLOYEE E
WHERE DEPT_ID IN (SELECT DEPT_ID FROM EMPLOYEE GROUP BY DEPT_ID
HAVING COUNT(*) = 1);
```

Q6. Show Employees With Duplicate City Entries In Emp_personal_detail Table (I.e., More Than One Employee From The Same City).

```
SELECT * FROM EMP_PERSONAL_DETAIL EPD
WHERE CITY IN ( SELECT CITY FROM EMP_PERSONAL_DETAIL
GROUP BY CITY
HAVING COUNT(*) > 1);
```

Q7. Display Employees With Email Ids That Do Not Follow Proper Format.

```
SELECT EMP_ID, EMAIL FROM EMP_PERSONAL_DETAIL
WHERE EMAIL NOT REGEXP '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$';
```

Q8. Write A Query To Display Employees Along With Their Rank Of Salary Within Department.

```
SELECT EMP_ID, EMP_NAME, DEPT_ID, SALARY,
RANK() OVER (PARTITION BY DEPT_ID ORDER BY SALARY DESC) AS SALARY_RANK
FROM EMPLOYEE;
```

Q9. Find Employees Whose Salary Is Within 10% Of The Highest Salary In The Company.

```
SELECT EMP_ID, EMP_NAME,  SALARY
FROM EMPLOYEE
WHERE SALARY >= (SELECT MAX(SALARY) * 0.9 FROM EMPLOYEE);
```

Q10. List Departments Where All Employees Are Single

```
SELECT D.DEPT_ID, D.DEPT_NAME
FROM DEPARTMENT D
JOIN EMPLOYEE E ON D.DEPT_ID = E.DEPT_ID
JOIN EMP_PERSONAL_DETAIL PD ON E.EMP_ID = PD.EMP_ID
GROUP BY D.DEPT_ID, D.DEPT_NAME
HAVING COUNT(*) = SUM(CASE WHEN PD.MARITAL_STATUS = 'SINGLE' THEN 1 ELSE 0 END);
```

Q11. Difference Between INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN With Example.

```
# 1. INNER JOIN = Returns Only Matching Rows From Both Tables.
SELECT E.EMP_NAME, D.DEPT_NAME
FROM EMPLOYEE E
INNER JOIN DEPARTMENT D ON E.DEPT_ID = D.DEPT_ID;
```

2. LEFT JOIN = Returns All Rows From The Left Table And Matching Rows From The Right. Non Matching Rows Show Null.

```
SELECT E.EMP_NAME, D.DEPT_NAME
FROM EMPLOYEE E
LEFT JOIN DEPARTMENT D ON E.DEPT_ID = D.DEPT_ID;
```

3. RIGHT JOIN = Returns All Rows From The Right Table And Matching Rows From The Left.

```
SELECT E.EMP_NAME, D.DEPT_NAME
FROM EMPLOYEE E
RIGHT JOIN DEPARTMENT D ON E.DEPT_ID = D.DEPT_ID;
```

4. FULL JOIN = Not Supported Directly In MySQL But Can Be Simulated.

Returns All Rows From Both Tables, Matching Where Possible. Use Union.

Q12. How To Find The Second Highest Salary In A Table?

```
SELECT DISTINCT SALARY
FROM EMPLOYEE
ORDER BY SALARY DESC
LIMIT 1 OFFSET 1;
```

Q13. What Is The Difference Between Where And Having Clause?

1) WHERE Clause

Used To Filter Rows Before Any Grouping Or Aggregation.

Cannot Use Aggregate Functions Like SUM(), AVG(), Etc.

2) HAVING Clause

Used To Filter Groups After Group By Has Been Applied.

Can Use Aggregate Functions Like COUNT(), MAX(), Etc.

Q14. Write A Query To Find Duplicate Records In A Table.

```
SELECT EMP_NAME, DEPT_ID, COUNT(*) AS DUPLICATE_COUNT
FROM EMPLOYEE
GROUP BY EMP_NAME, DEPT_ID
HAVING COUNT(*) > 1;
```

Q15. Write A Query To Find Employees Who Have The Same Salary As Another Employee.

```
SELECT E1.EMP_ID, E1.EMP_NAME, E1.SALARY
FROM EMPLOYEE E1
JOIN EMPLOYEE E2
ON E1.SALARY = E2.SALARY
AND E1.EMP_ID <> E2.EMP_ID;
```