mysql> use joints

Database changed

mysql> show tables;

+------------------+

| Tables\_in\_joints |

+------------------+

| departments |

| employees |

+------------------+

2 rows in set (0.01 sec)

mysql> select \* from departments;

+---------------+-----------------+

| department\_id | department\_name |

+---------------+-----------------+

| 1 | HR |

| 2 | IT |

| 3 | Finaace |

+---------------+-----------------+

3 rows in set (0.00 sec)

mysql> select \* from employees;

+--------+-------------+---------------+

| emp\_id | emp\_name | department\_id |

+--------+-------------+---------------+

| 1 | John Doe | 1 |

| 2 | Jane Smith | 2 |

| 3 | Bob Johnson | 1 |

| 4 | Ram John | 2 |

| 5 | Ram John | 4 |

+--------+-------------+---------------+

5 rows in set (0.00 sec)

1. INNER JOIN

mysql> select \* from employees inner join departments on employees.department\_id = departments.departm

ent\_id;

+--------+-------------+---------------+---------------+-----------------+

| emp\_id | emp\_name | department\_id | department\_id | department\_name |

+--------+-------------+---------------+---------------+-----------------+

| 1 | John Doe | 1 | 1 | HR |

| 2 | Jane Smith | 2 | 2 | IT |

| 3 | Bob Johnson | 1 | 1 | HR |

| 4 | Ram John | 2 | 2 | IT |

+--------+-------------+---------------+---------------+-----------------+

4 rows in set (0.00 sec)

1. LEFT JOIN

mysql> select \* from employees left join departments on employees.department\_id = departments.departme

nt\_id;

+--------+-------------+---------------+---------------+-----------------+

| emp\_id | emp\_name | department\_id | department\_id | department\_name |

+--------+-------------+---------------+---------------+-----------------+

| 1 | John Doe | 1 | 1 | HR |

| 2 | Jane Smith | 2 | 2 | IT |

| 3 | Bob Johnson | 1 | 1 | HR |

| 4 | Ram John | 2 | 2 | IT |

| 5 | Ram John | 4 | NULL | NULL |

+--------+-------------+---------------+---------------+-----------------+

5 rows in set (0.00 sec)

1. RIGHT JOIN

mysql> select \* from employees right join departments on employees.department\_id = departments.departm

ent\_id;

+--------+-------------+---------------+---------------+-----------------+

| emp\_id | emp\_name | department\_id | department\_id | department\_name |

+--------+-------------+---------------+---------------+-----------------+

| 3 | Bob Johnson | 1 | 1 | HR |

| 1 | John Doe | 1 | 1 | HR |

| 4 | Ram John | 2 | 2 | IT |

| 2 | Jane Smith | 2 | 2 | IT |

| NULL | NULL | NULL | 3 | Finaace |

+--------+-------------+---------------+---------------+-----------------+

5 rows in set (0.00 sec)

mysql> select \* from employees right join departments on departments.department\_id = employees.department\_id;

+--------+-------------+---------------+---------------+-----------------+

| emp\_id | emp\_name | department\_id | department\_id | department\_name |

+--------+-------------+---------------+---------------+-----------------+

| 3 | Bob Johnson | 1 | 1 | HR |

| 1 | John Doe | 1 | 1 | HR |

| 4 | Ram John | 2 | 2 | IT |

| 2 | Jane Smith | 2 | 2 | IT |

| NULL | NULL | NULL | 3 | Finaace |

+--------+-------------+---------------+---------------+-----------------+

5 rows in set (0.00 sec)

Assigments

1. Basic Inner Join

- Objective: Use an inner join to combine data from two tables.

- Assignment:

Write a query to select the first name, last name, and department name for all employees who belong to a department. Use the employees and departments tables.

SELECT e.first\_name, e.last\_name, d.department\_name

FROM employees e

INNER JOIN departments d ON e.department\_id = d.department\_id;

mysql> select \* from employees;

+--------+------------+-----------+---------------+

| emp\_id | first\_name | last\_name | department\_id |

+--------+------------+-----------+---------------+

| 1 | Reena | Singh | 101 |

| 2 | Mohan | Yadav | 101 |

| 3 | Rizwan | Mohd | 102 |

| 4 | Quisid | Jadav | 104 |

| 5 | Sunita | Morya | 103 |

| 6 | Sahil | Gupta | 103 |

+--------+------------+-----------+---------------+

6 rows in set (0.00 sec)

mysql> select \* from departments;

+---------------+-----------------+

| department\_id | department\_name |

+---------------+-----------------+

| 101 | Hr |

| 102 | Finance |

| 103 | IT |

+---------------+-----------------+

3 rows in set (0.00 sec)

mysql> select e.first\_name, e.last\_name, d.department\_name from employees as e inner join departments

as d on e.department\_id = d.department\_id;

+------------+-----------+-----------------+

| first\_name | last\_name | department\_name |

+------------+-----------+-----------------+

| Reena | Singh | Hr |

| Mohan | Yadav | Hr |

| Rizwan | Mohd | Finance |

| Sunita | Morya | IT |

| Sahil | Gupta | IT |

+------------+-----------+-----------------+

5 rows in set (0.00 sec)

2. Left Join (Returning All Records from Left Table)

- Objective: Use a left join to get all records from the left table, even if there is no match in the right table.

- Assignment:

Write a query to display all employees and their department names. Show NULL for employees without a department.

SELECT e.first\_name, e.last\_name, d.department\_name

FROM employees e

LEFT JOIN departments d ON e.department\_id = d.department\_id;

mysql> select e.first\_name, e.last\_name, d.department\_name from employees as e left join departments a

s d on e.department\_id = d.department\_id;

+------------+-----------+-----------------+

| first\_name | last\_name | department\_name |

+------------+-----------+-----------------+

| Reena | Singh | Hr |

| Mohan | Yadav | Hr |

| Rizwan | Mohd | Finance |

| Quisid | Jadav | NULL |

| Sunita | Morya | IT |

| Sahil | Gupta | IT |

+------------+-----------+-----------------+

3. Right Join (Returning All Records from Right Table)

- Objective: Use a right join to get all records from the right table, even if there is no match in the left table.

- Assignment:

Write a query to display all departments and their employees. Show NULL for departments with no employees.

SELECT d.department\_name, e.first\_name, e.last\_name

FROM departments d

RIGHT JOIN employees e ON d.department\_id = e.department\_id;

mysql> select d.department\_name , e.first\_name , e.last\_name from departments as d right join employees as e on d.department\_id = e.department\_id;

+-----------------+------------+-----------+

| department\_name | first\_name | last\_name |

+-----------------+------------+-----------+

| Hr | Reena | Singh |

| Hr | Mohan | Yadav |

| Finance | Rizwan | Mohd |

| NULL | Quisid | Jadav |

| IT | Sunita | Morya |

| IT | Sahil | Gupta |

+-----------------+------------+-----------+

6 rows in set (0.00 sec)

4. Full Outer Join (Simulated)

- Objective: MySQL does not support FULL OUTER JOIN, so simulate it using a combination of LEFT JOIN and RIGHT JOIN.

- Assignment:

Write a query to display all employees and all departments, showing NULL for unmatched rows.

SELECT e.first\_name, e.last\_name, d.department\_name

FROM employees e

LEFT JOIN departments d ON e.department\_id = d.department\_id

UNION

SELECT e.first\_name, e.last\_name, d.department\_name

FROM employees e

RIGHT JOIN departments d ON e.department\_id = d.department\_id;

mysql> select e.first\_name , e.last\_name , d.department\_name from employees as e left join departments as d on e.department\_id = d.department\_id union select e.first\_name, e.last\_name, d.department\_name from employees as e right join departments as d on e.department\_id = d.department\_id;

+------------+-----------+-----------------+

| first\_name | last\_name | department\_name |

+------------+-----------+-----------------+

| Reena | Singh | Hr |

| Mohan | Yadav | Hr |

| Rizwan | Mohd | Finance |

| Quisid | Jadav | NULL |

| Sunita | Morya | IT |

| Sahil | Gupta | IT |

+------------+-----------+-----------------+

6 rows in set (0.01 sec)

5. Self Join (Joining a Table with Itself)

- Objective: Use a self join to combine data from the same table.

- Assignment:

Write a query to find employees who have the same manager. Use a self join on the employees table.

SELECT e1.first\_name AS Employee, e2.first\_name AS Manager

FROM employees e1

JOIN employees e2 ON e1.manager\_id = e2.employee\_id;

6. Join with Multiple Conditions

- Objective: Use a join with multiple conditions.

- Assignment:

Write a query to select the employee name and department name for employees who are either in the 'HR' or 'IT' department and have a salary greater than 50,000.

SELECT e.first\_name, e.last\_name, d.department\_name

FROM employees e

INNER JOIN departments d ON e.department\_id = d.department\_id

WHERE d.department\_name IN ('HR', 'IT')

AND e.salary > 50000;

mysql> select e.first\_name , e.last\_name, d.department\_name from employees as e inner join departments as d on e.department\_id = d.department\_id where d.department\_name in("Hr", "IT") and e.salary>50000;

+------------+-----------+-----------------+

| first\_name | last\_name | department\_name |

+------------+-----------+-----------------+

| Sunita | Morya | IT |

| Sahil | Gupta | IT |

+------------+-----------+-----------------+

2 rows in set (0.00 sec)

7. Using JOIN with GROUP BY

- Objective: Use a join with a GROUP BY clause.

- Assignment:

Write a query to find the total salary paid to employees in each department.

SELECT d.department\_name, SUM(e.salary) AS total\_salary

FROM employees e

INNER JOIN departments d ON e.department\_id = d.department\_id

GROUP BY d.department\_name;

mysql> select d.department\_name , sum(e.salary) as Total\_salary from employees as e inner join departments as d on e.department\_id = d.department\_id group by d.department\_name;

+-----------------+--------------+

| department\_name | Total\_salary |

+-----------------+--------------+

| Hr | 55000 |

| Finance | 50000 |

| IT | 175000 |

+-----------------+--------------+

3 rows in set (0.00 sec

8. Join with `ORDER BY`

- Objective: Use a join with sorting.

- Assignment:

Write a query to display the employee names and their department names, ordered by employee salary in descending order.

SELECT e.first\_name, e.last\_name, d.department\_name, e.salary

FROM employees e

INNER JOIN departments d ON e.department\_id = d.department\_id

ORDER BY e.salary DESC;

mysql> select e.first\_name , e.last\_name , d.department\_name , e.salary from employees as e inner join

departments d on e.department\_id = d.department\_id order by e.salary desc;

+------------+-----------+-----------------+--------+

| first\_name | last\_name | department\_name | salary |

+------------+-----------+-----------------+--------+

| Sunita | Morya | IT | 90000 |

| Sahil | Gupta | IT | 85000 |

| Rizwan | Mohd | Finance | 50000 |

| Mohan | Yadav | Hr | 30000 |

| Reena | Singh | Hr | 25000 |

+------------+-----------+-----------------+--------+

5 rows in set (0.00 sec)

9. Join with Aliases

- Objective: Use table aliases in joins for easier reference.

- Assignment:

Write a query to display the employee names along with their department names, **using aliases for the tables.**

SELECT emp.first\_name, emp.last\_name, dept.department\_name

FROM employees AS emp

INNER JOIN departments AS dept ON emp.department\_id = dept.department\_id;

mysql> select emp.first\_name, emp.last\_name, dept.department\_name from employees as emp inner join dep

artments as dept on emp.department\_id = dept.department\_id;

+------------+-----------+-----------------+

| first\_name | last\_name | department\_name |

+------------+-----------+-----------------+

| Reena | Singh | Hr |

| Mohan | Yadav | Hr |

| Rizwan | Mohd | Finance |

| Sunita | Morya | IT |

| Sahil | Gupta | IT |

+------------+-----------+-----------------+

5 rows in set (0.00 sec)

10. Join with Aggregation and `HAVING` Clause

- Objective: Use a join with aggregation and a `HAVING` clause to filter grouped results.

- Assignment:

Write a query to display departments where the average salary of employees is greater than 60,000.

SELECT d.department\_name, AVG(e.salary) AS avg\_salary

FROM employees e

INNER JOIN departments d ON e.department\_id = d.department\_id

GROUP BY d.department\_name

HAVING AVG(e.salary) > 60000;

mysql> select d.department\_name , avg(e.salary) as Avg\_salary from employees e inner join departments

d on e.department\_id = d.department\_id group by d.department\_name having avg(e.salary)>60000;

+-----------------+------------+

| department\_name | Avg\_salary |

+-----------------+------------+

| IT | 87500.0000 |

+-----------------+------------+

1 row in set (0.01 sec)

mysql>