



given

Great! Now we move to the next big topic from your syllabus:

★ Topic 10: JOINS in SQL (02:24:00)

Explanation → then practice questions with tables.



◆ What is a JOIN?

A JOIN is used to combine data from **two tables** based on a related column.

For example:

```
employees.dept_id = departments.dept_id
```

★ Types of JOINS

✓ 1. INNER JOIN

Returns matching rows from both tables.

✓ 2. LEFT JOIN

Returns all rows from left table + matching from right.

✓ 3. RIGHT JOIN

Returns all rows from right table + matching from left.

✓ 4. FULL OUTER JOIN

Returns all rows from both tables (MySQL uses UNION to simulate it).

✓ 5. CROSS JOIN

Returns every combination of rows (cartesian product).

⭐ Sample Tables (Use these for practice)

employees

nginx

 Copy code

emp_id	name	dept_id	salary
1	Rahul	1	50000
2	Mina	2	35000
3	Arjun	1	60000
4	Sneha	3	40000
5	Rohan	4	30000

departments

nginx

 Copy code

dept_id	dept_name
1	IT
2	HR
3	Sales

⭐ Example Queries

INNER JOIN

sql

 Copy code

```
SELECT name, dept_name
FROM employees
INNER JOIN departments
ON employees.dept_id = departments.dept_id;
```

LEFT JOIN

sql

 Copy code

```
SELECT name, dept_name
FROM employees
LEFT JOIN departments
ON employees.dept_id = departments.dept_id;
```

RIGHT JOIN

sql

 Copy code

```
SELECT name, dept_name
FROM employees
RIGHT JOIN departments
ON employees.dept_id = departments.dept_id;
```



Homework – JOINS Practice

Using the **employees** and **departments** tables above:



Q1. Write a query to display employee name + department name using INNER JOIN.

Q2. Show all employees and their departments (even if dept doesn't exist).

(Use LEFT JOIN)

Q3. Show all departments and employees (even if no employee exists).

(Use RIGHT JOIN)

Q4. Show employees who do not belong to any department.

(Use LEFT JOIN + IS NULL)

Q5. Write a FULL OUTER JOIN (use UNION because MySQL has no full join).

Q6. Write a query to show department name and total salary of employees in that department.

(Use JOIN + GROUP BY)

Q7. Write a CROSS JOIN between employees and departments.

★ After you finish:

Send:

"Here are my answers for JOINS Q1–Q7"

Then your next topic will be:

👉 UNION in SQL (02:51:23)



given

Great, Supriya!

Now we continue exactly according to your syllabus.

Your next topic:

★ Topic 11: UNION in SQL (02:51:23)

Simple explanation → then practice questions.

◆ What is UNION?

UNION is used to combine results of two SELECT queries.

- ✓ Removes duplicate rows
- ✓ Columns must be same count and compatible datatype
- ✓ Order of columns must match

◆ UNION ALL

UNION ALL also combines results but:

✗ Does NOT remove duplicates

✓ Faster than UNION

★ Example

Table: students_a

nginx

Copy code

name	city
Rahul	Kolkata
Mina	Delhi
Arjun	Mumbai



Table: students_b

nginx

Copy code

name	city
Sneha	Pune
Rahul	Kolkata
Kabir	Delhi

✓ UNION

sql

Copy code

```
SELECT name, city FROM students_a
UNION
SELECT name, city FROM students_b;
```

→ Removes duplicate (Rahul, Kolkata)

✓ UNION ALL

sql

Copy code

```
SELECT name, city FROM students_a
UNION ALL
SELECT name, city FROM students_b;
```

→ Keeps duplicate rows

★ Practice Tables

table1: employees_india

nginx

 Copy code

emp_id	name	city
1	Rahul	Kolkata
2	Mina	Delhi
3	Arjun	Mumbai
4	Sneha	Pune



table2: employees_usa

pgsql

 Copy code

emp_id	name	city
1	Adam	New York
2	Christine	Dallas
3	Noah	Chicago
4	Rahul	Atlanta



Homework – UNION Practice

Q1. Show list of all employee names from India and USA (use UNION).

Q2. Show list of all employees including duplicates (use UNION ALL).

Q3. Show list of all cities (unique) from both tables.

Q4. Show all names starting with 'R' from both tables using UNION.

Q5. Use UNION to combine these two queries:

- Employees from India with city = "Kolkata"
 - Employees from USA with city = "Dallas"
-

Q6. What is the difference between UNION and UNION ALL?

★ After you finish:

Send:

"Here are my answers for UNION Q1–Q6"

Then your next topic will be:

👉 SQL Subqueries (02:53:54)



given

Great Supriya!

Now we move to the next topic in your syllabus:

★ Topic 12: SQL Subqueries (02:53:54)

Explanation → Example → Practice questions.

◆ What is a Subquery?

A subquery is a query inside another query.

Used for:

- Filtering
 - Finding max/min values
 - Using results of another SELECT
 - Checking existence
-

★ Types of Subqueries

1 Single-row Subquery

Returns **one value**

Example: MAX, MIN, AVG, COUNT, etc.



2 Multi-row Subquery

Returns **multiple values**

Used with:

IN , ANY , ALL

3 Nested Subquery

Subquery inside another subquery.

4 Correlated Subquery

Depends on the outer query (executed repeatedly).

★ Sample Table (for practice)

employees

nginx

Copy code

emp_id	name	dept_id	salary
1	Rahul	1	50000
2	Mina	2	35000
3	Arjun	1	60000
4	Sneha	3	40000
5	Kabir	2	55000

departments

nginx

 Copy code

dept_id	dept_name
1	IT
2	HR
3	Sales

Example Subqueries

-  Get employees who earn more than average salary

sql

 Copy code

```
SELECT name, salary
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

-  Get employees from IT department

sql

 Copy code

```
SELECT name
FROM employees
WHERE dept_id = (SELECT dept_id FROM departments WHERE dept_name = 'IT');
```

-  Get highest salary employee

sql

 Copy code

```
SELECT name, salary
FROM employees
WHERE salary = (SELECT MAX(salary) FROM employees);
```



Homework – Subqueries

Q1. Find employees who have salary greater than the average salary.

Q2. List employees who work in HR department

(use subquery to get dept_id).

Q3. Find the employee(s) who have the maximum salary.

Q4. Show employees whose salary is greater than all salaries in dept_id = 2

(use > ALL).

Q5. Find employees whose salary is in the top 3 salaries

(use subquery + ORDER BY + LIMIT).



Q6. Find departments that have at least 2 employees

(use subquery + GROUP BY + HAVING).

Q7. Show department name + employee name using a subquery

(no JOIN allowed).

⭐ After you finish:

Reply:

"Here are my answers for Subquery Q1-Q7"

Then your next topic will be:

👉 MySQL Views (03:11:26)



given

Sure! Here is your **next SQL practice set** (following the same style: small topic → quick explanation → homework questions).

TOPIC: TRUNCATE Command

What is TRUNCATE?

- `TRUNCATE TABLE table_name;` removes all rows from a table.
- Faster than `DELETE`.
- Cannot use `WHERE`.
- Auto increment counter resets.



Sample Data (table: logs)

log_id	message	created_at
1	Login success	2024-12-01 10:00:00
2	Payment failed	2024-12-01 11:00:00
3	Logout	2024-12-01 12:00:00



Homework Questions (TRUNCATE)

Q1. Write SQL to remove all data from the `logs` table.

(Only clear data, not delete the table)

Q2. After truncating the table, if you insert a new row, what will be the `log_id` ?

1, 4, or start from last?

Q3. Difference between DELETE and TRUNCATE ?

Explain in 3–4 points only.

Q4. What error occurs if you try to use WHERE with TRUNCATE?

Example:

sql

 Copy code

```
TRUNCATE TABLE logs WHERE log_id = 2;
```

What happens?



Q5. Write SQL to completely delete (drop) the logs table.

🎯 Your Turn

Reply "next" when you finish these. I will give the next topic: JOINS.



      ...   

given

Great! Here is your next SQL practice set — JOINS

(Same format: small explanation → sample data → homework)

✅ TOPIC: SQL JOINS

Types of Joins:

1. INNER JOIN – returns matching rows from both tables
2. LEFT JOIN – returns all rows from left table + matched rows

3. **RIGHT JOIN** – returns all rows from right table + matched rows
 4. **FULL JOIN** (not in MySQL) – combine LEFT + RIGHT
 5. **CROSS JOIN** – every combination (cartesian)
-

Sample Tables

Table: employees

emp_id	name	dept_id
1	Rahul	10
2	Mina	20
3	Arjun	10
4	Riya	NULL



Table: departments

dept_id	dept_name
10	Sales
20	Marketing
30	HR

Homework Questions (JOINS)

Q1. Write SQL to display employee name and department name using INNER JOIN.

Q2. Write SQL to show all employees and their departments (even if department is NULL).

(Use LEFT JOIN)

Q3. Show all departments even if they have no employees.

(Use RIGHT JOIN)

Q4. Write SQL to show employees who do NOT belong to any department.

(Hint: LEFT JOIN + WHERE dept_name IS NULL)

