



given next

Great! Now we continue your syllabus in order.

Your next topic is:

★ Topic 5: Revisiting Foreign Keys (01:58:41)

I will explain **very simple**, then give **practice questions**.

◆ What is a Foreign Key?

A **foreign key** connects two tables.

Example:

Table: departments

nginx

dept_id	dept_name
1	IT
2	HR
3	Finance

Table: employees

nginx

emp_id	name	dept_id	salary
1	Rahul	1	50000
2	Mina	2	45000
3	Arjun	3	60000

Here:

`employees.dept_id` → **foreign key referencing** `departments.dept_id`.

◆ Foreign Key Syntax

sql

```
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,
    name VARCHAR(50),
    dept_id INT,
    salary INT,
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);
```

⭐ Practice (Foreign Keys)

Use these tables:

departments

```
nginx

dept_id | dept_name
1       | IT
2       | HR
3       | Sales
```

employees

```
nginx

emp_id | name      | dept_id | salary | city
1       | Rahul     | 1        | 50000  | Kolkata
2       | Mina      | 2        | 35000  | Delhi
3       | Arjun     | 1        | 60000  | Mumbai
4       | Sneha     | 3        | 40000  | Pune
5       | Rohan     | 2        | 30000  | Kolkata
```

📝 Homework – Foreign Keys

Q1. Write SQL to create the `departments` table with `dept_id` as PRIMARY KEY.

Q2. Write SQL to create the `employees` table with `dept_id` as a FOREIGN KEY referencing `departments(dept_id)`.

Q3. Insert all rows for the `departments` table (3 rows).

Q4. Insert all rows for the `employees` table (5 rows).

Q5. Try inserting this row:

```
ini  
  
emp_id = 6  
name = "Kabir"  
dept_id = 5  
salary = 45000  
city = "Chennai"
```

Will it work? Why/why not?

⭐ Next Topic After You Finish This

👉 Cascading Foreign Keys (02:06:35)

Reply when ready:

"Here are my answers for Q1–Q5"

I will check and then move to the next topic!



given

Great, Supriya!

We continue with your next topic from the syllabus:

⭐ Topic 6: Cascading Foreign Keys (02:06:35)

I'll explain very simply, then give practice questions.

◆ What is CASCADE in SQL?

When you delete or update a row in the **parent table**, CASCADE automatically updates/deletes matching rows in the **child table**.

★ Types of Cascade

1. ON DELETE CASCADE

If you delete a department → all employees in that department will also be deleted.

2. ON UPDATE CASCADE

If you update dept_id in departments → it automatically updates employees.dept_id also.

◆ Example Tables

Departments (parent)

nginx

dept_id	dept_name
1	IT
2	HR
3	Sales

Employees (child)

nginx

emp_id	name	dept_id	salary
1	Rahul	1	50000
2	Mina	2	35000
3	Arjun	1	60000

◆ Creating table with CASCADE

sql

```
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,
    name VARCHAR(50),
    dept_id INT,
    salary INT,
    FOREIGN KEY (dept_id)
        REFERENCES departments(dept_id)
        ON DELETE CASCADE
```

```
    ON UPDATE CASCADE
);
```

★ What happens with CASCADE?

If you run:

```
sql
DELETE FROM departments WHERE dept_id = 1;
```

Automatically deleted:

Rahul and Arjun (because they belong to IT dept)

★ Homework – Cascading Foreign Keys

Use these tables:

departments

```
nginx
dept_id | dept_name
1       | IT
2       | HR
3       | Marketing
```

employees

```
nginx
emp_id | name      | dept_id | salary
1       | Rahul     | 1        | 50000
2       | Mina      | 2        | 35000
3       | Arjun     | 1        | 60000
4       | Sneha     | 3        | 40000
```

✍ Q1. Write SQL to create `employees` table with

ON DELETE CASCADE and ON UPDATE CASCADE .

Q2. If you run this query:

sql

```
DELETE FROM departments WHERE dept_id = 1;
```

Which employees will get deleted automatically?

Q3. If you change the department ID:

sql

```
UPDATE departments SET dept_id = 10 WHERE dept_id = 3;
```

What will happen to employees related to department 3?

Q4. Insert this row into employees:

ini

```
emp_id = 5
name = "Kabir"
dept_id = 10
salary = 55000
```

Will it work? (Think about updated dept_id)

Q5. What is the difference between:

- ON DELETE CASCADE
 - ON UPDATE CASCADE
-

Reply When Done:

"Here are my answers for Cascade Q1–Q5"

Then I'll give you the next topic:

👉 ALTER Command (02:12:30)



given

Great Supriya!

We continue your syllabus exactly in order.

Next topic is:

⭐ Topic 7: ALTER Command (02:12:30)

I will explain simply → then give practice questions.

◆ ALTER TABLE – What it does?

ALTER TABLE is used to change the structure of an existing table.

You can use it to:

- ✓ Add a column
 - ✓ Drop a column
 - ✓ Rename a column
 - ✓ Change datatype
 - ✓ Rename the table
-

⭐ 1. Add a column

sql

```
ALTER TABLE employees  
ADD email VARCHAR(100);
```

⭐ 2. Drop a column

sql

```
ALTER TABLE employees  
DROP COLUMN email;
```

★ 3. Rename a column

sql

```
ALTER TABLE employees  
RENAME COLUMN name TO full_name;
```

★ 4. Change datatype

sql

```
ALTER TABLE employees  
MODIFY salary DECIMAL(10,2);
```

★ 5. Rename table

sql

```
ALTER TABLE employees  
RENAME TO staff;
```

★ Practice Table (Use this):

employees

nginx

emp_id	name	dept_id	salary	city
1	Rahul	1	50000	Kolkata
2	Mina	2	35000	Delhi
3	Arjun	1	60000	Mumbai

4		Sneha		3		40000		Pune
5		Rohan		2		30000		Kolkata

Homework – ALTER TABLE

Q1. Add a new column `email` `VARCHAR(100)` to `employees`.

Q2. Add a new column `age` `INT`.

Q3. Change datatype of `salary` to `DECIMAL(10,2)`.

Q4. Rename column `name` to `full_name`.

Q5. Drop the column `city`.

Q6. Rename table `employees` to `company_staff`.

Q7. Add a new column `joining_date` `DATE`.

Q8. Modify `dept_id` datatype from `INT` to `VARCHAR(10)`.

Q9. Delete column `age` from the table.

After you finish:

Send:

"Here are my answers for ALTER Q1–Q9"

Then I will give the next topic:

 CHANGE and MODIFY Commands (02:15:06)



given

Perfect Supriya!

We continue your SQL syllabus **exactly in order**.

Your next topic is:

⭐ Topic 8: CHANGE and MODIFY Commands (02:15:06)

These are used to **change columns** in an existing table.

I will explain **very simple**, then give **homework**.

◆ 1. MODIFY Command

Used to **change only datatype** of a column.

Example:

```
sql

ALTER TABLE employees
MODIFY salary DECIMAL(10,2);
```

- ✓ Keeps the same column name
 - ✓ Only datatype or constraints change
-

◆ 2. CHANGE Command

Used to **rename + change datatype** together.

Example:

```
sql

ALTER TABLE employees
CHANGE name full_name VARCHAR(100);
```

Here:

- `name` → old column name
 - `full_name` → new column name
 - `VARCHAR(100)` → new datatype
- ✓ CHANGE can do rename + datatype change
 - ✓ MODIFY can only change datatype
-

⭐ Example Table (use this for practice)

employees

nginx

emp_id	name	salary	city
1	Rahul	50000	Kolkata
2	Mina	35000	Delhi
3	Arjun	60000	Mumbai
4	Sneha	40000	Pune



Homework – CHANGE & MODIFY

Q1. Change datatype of `city` to `VARCHAR(50)` using `MODIFY`.

Q2. Change `name` column to `full_name VARCHAR(80)` using `CHANGE`.

Q3. Change datatype of `salary` to `INT` using `MODIFY`.

Q4. Rename `salary` to `monthly_salary` with datatype `DECIMAL(10,2)` using `CHANGE`.

Q5. Modify `emp_id` to datatype `BIGINT`.

Q6. Rename `city` column to `location` with datatype `VARCHAR(100)`.



After you finish:

Send:

"Here are my answers for CHANGE/MODIFY Q1–Q6"

Then your next topic will be:



TRUNCATE Command (02:20:33)



given

Great Supriya!

Now we move to the next topic from your syllabus:

⭐ Topic 9: TRUNCATE Command (02:20:33)

Simple explanation → then practice questions.

◆ What is TRUNCATE?

`TRUNCATE` is used to delete all rows from a table **very fast**.

- ✓ Deletes all data
 - ✓ Does NOT delete the table
 - ✓ Resets AUTO_INCREMENT to 1
 - ✓ Cannot be rolled back (in MySQL with default settings)
-

⭐ Syntax

sql

```
TRUNCATE TABLE employees;
```

◆ TRUNCATE vs DELETE

Feature	DELETE	TRUNCATE
Deletes rows	Yes	Yes
WHERE condition	✓ Yes	✗ No
Auto-increment reset	✗ No	✓ Yes
Speed	Slow	Very fast
Rollback	Sometimes	Usually cannot

⭐ Example Table (use this for practice)

`orders`

yaml

order_id	customer	amount	city
1	Rahul	500	Kolkata
2	Mina	1200	Delhi
3	Arjun	900	Pune
4	Sneha	700	Mumbai



Homework – TRUNCATE Command

Q1. Write SQL to delete all data from `orders` table using TRUNCATE.

Q2. After TRUNCATE, if you insert a new row:

ini

```
order_id = auto increment
customer = "Kabir"
amount = 1500
city = "Chennai"
```

What will the new `order_id` be?

Q3. What is the difference between:

sql

```
DELETE FROM orders;
TRUNCATE TABLE orders;
```

Q4. Can you use TRUNCATE with a WHERE clause? Why?

Q5. If `orders` table has a foreign key referencing it,
will TRUNCATE work?



After you finish:

Send:

"Here are my answers for TRUNCATE Q1–Q5"

Then your next topic will be:

👉 JOINS in SQL (02:24:00)

