--------LECTURE- 8--------

What is a Foreign Key?

A foreign key is like a bridge that connects two tables.

- **One table has the main ID (this is the parent).**
- **The other table has a copy of that ID (this is the child).**
- **That copy is called the foreign key.**

It makes sure the two tables stay connected correctly.

Imagine:

Table 1: Departments

dept_id	dept_name
10	HR
20	Sales

Table 2: Employees

emp_id	emp_name	dept_id
1	Rohan	10
2	Shilpa	20

Here:

- **departments.dept_id → primary key**
- **employees.dept_id → foreign key**

The foreign key says:

👉 **"Every employee must belong to a department that actually exists."**
If you try to insert employee with dept_id = 99, SQL will say:

❌ **"No such department!"**
This protects the data from mistakes.

Why Foreign Key is Useful

- **Keeps data correct**
- **Prevents wrong values**
- **Maintains relationships**
- **Helps with joins**
- **Very important in real databases and interviews**

🔦 Foreign Key Example in SQL

FOREIGN KEY (dept_id)
REFERENCES departments(dept_id)

This line creates the connection between the two tables.

🟫 What is CASCADE?

CASCADE means:

👉 **"If something happens in the parent table, automatically do the same in the child table."**
There are two common types:

1️⃣ ON DELETE CASCADE
2️⃣ ON UPDATE CASCADE

Think of:

- **Parent table = Departments**
- **Child table = Employees**

Employees belong to departments.
If the parent changes or gets removed, the child should follow — this is CASCADE.

🟤 1️⃣ ON DELETE CASCADE

👉 **If you delete a row in the parent table,**
SQL will automatically delete the matching rows in the child table.

Example:

Departments (Parent)	
dept_id	**dept_name**
10	HR

Employees (Child)

emp_name	dept_id
Rohan	**10**
Shilpa	**10**

If we run:
DELETE FROM departments WHERE dept_id = 10;
And the foreign key has ON DELETE CASCADE, then:

👉 **Both Rohan and Shilpa will be deleted automatically.**
Like:
"Department is gone → all its employees should also go."

🟤 2 ON UPDATE CASCADE

👉 **If you change (update) the parent table value,**
SQL will update the child table automatically.

Example:

Departments (before):

dept_id	dept_name
10	**HR**

Employees (before):

emp_name	dept_id
Rohan	**10**
Shilpa	**10**

If this query runs:

```
UPDATE departments
SET dept_id = 99
WHERE dept_id = 10;
```

With ON UPDATE CASCADE, employees table becomes:

emp_name	dept_id
Rohan	**99**
Shilpa	**99**

SQL says:
👉 **"If parent dept_id changed, child dept_id must also change."**

🤔 Why CASCADE is useful

- **Keeps data clean**
- **No broken relationships**
- **Saves time**
- **Avoids errors**