Video-15

Topics to cover:

- Joins in SQL
  - 5. CROSS JOIN
  - 6. SELF JOIN

5. CROSS JOIN: Produces the Cartesian product of both tables.
Every row from the first table combines with every row from the second.

Syntax:

```
SELECT column_list
FROM table1
CROSS JOIN table2;
```

Note: If 3 employees and 4 departments exist, result = 12 rows

🗂️ Example Tables

1. Students Table

```
CREATE TABLE Students (
        student_id INT PRIMARY KEY,
        student_name VARCHAR(50)
);
INSERT INTO Students (student_id, student_name) VALUES
(1, 'Alice'),
(2, 'Bob');
```

2. Courses Table

```
CREATE TABLE Courses (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(50)
);
INSERT INTO Courses (course_id, course_name) VALUES
(101, 'Math'),
(102, 'Science');
```

🔗 CROSS JOIN Example

```
SELECT student_id, student_name, course_name
FROM Students
CROSS JOIN Courses;
```

6. SELF JOIN: A self join is simply a regular join where a table is joined to itself. You must use aliases to differentiate the two instances of the same table.

Syntax:

```
SELECT a.column_list, b.column_list
FROM table_name a
JOIN table_name b
ON a.common_column = b.common_column;
```

```
CREATE TABLE employees (
    emp_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    manager_id INT,
    FOREIGN KEY (manager_id) REFERENCES employees(emp_id)
);

INSERT INTO employees (name, manager_id)
VALUES
    ('Amit', NULL),   -- Amit is the top-level manager (no manager)
    ('Priya', 1),     -- Priya reports to Amit
    ('Ravi', 1),      -- Ravi reports to Amit
    ('Sneha', 2),     -- Sneha reports to Priya
    ('Arjun', 3);     -- Arjun reports to Ravi

SELECT * FROM employees;
```

```
SELECT e1.emp_id, e1.name AS Employee, e2.name AS Manager
FROM employees e1
LEFT JOIN employees e2
ON e1.manager_id = e2.emp_id;
```