# Ch-14: Joining in SQL Interface

A **JOIN** combines rows from **two or more tables** based on a related column between them (usually a **primary key–foreign key** relationship).

## 1. Cartesian Join (Cross Join)

**Description:**

- Joins **every row** from the first table with **every row** from the second table.
- If Table A has *m* rows and Table B has *n* rows, result = $m \times n$ rows.

**Syntax:**

```
SELECT * FROM Student, Marks;
-- OR
SELECT * FROM Student CROSS JOIN Marks;
```

**Example:**

**Student Table:**

| RollNo | Name |
|---|---|
| 101 | Ravi |
| 102 | Neha |

**Marks Table:**

| RollNo | Subject | Marks |
|---|---|---|
| 101 | Math | 90 |
| 102 | English | 85 |

**Result:**

All combinations of rows ($2 \times 2 = 4$ rows).

## 2. Equi-Join

**Description:**

- Combines rows based on a **common column** using = operator.
- Returns **matching records** from both tables.

**Syntax:**

```
SELECT Student.Name, Marks.Subject, Marks.Marks
FROM Student, Marks
WHERE Student.RollNo = Marks.RollNo;
```

**Example:**

**Student Table:**

| RollNo | Name |
|---|---|
| 101 | Ravi |
| 102 | Neha |

**Marks Table:**

| RollNo | Subject | Marks |
|---|---|---|
| 101 | Math | 90 |
| 102 | English | 85 |
| 103 | Science | 70 |

**Result:**
Only matching rows (based on RollNo):

| Name | Subject | Marks |
|---|---|---|
| Ravi | Math | 90 |
| Neha | English | 85 |

## 3. Natural Join

**Description:**

- Automatically joins tables based on **all columns with the same name** and compatible data types.
- Removes duplicate columns in result.

**Syntax (depends on RDBMS support):**

```
SELECT *
FROM Student
NATURAL JOIN Marks;
```

**Note:**

- Not supported in all databases (e.g., MySQL requires specific configurations).
- Recommended to use INNER JOIN USING() if NATURAL JOIN is not available.

**Alternative in MySQL:**

```
SELECT *
FROM Student
JOIN Marks
USING (RollNo);
```

## Comparison Table of Joins

| Join Type | Returns | Duplicate Columns | Condition Needed |
|---|---|---|---|
| **Cartesian Join** | All combinations of rows | Yes | □ No |
| **Equi-Join** | Matching rows based on condition | Yes | □ Yes |
| **Natural Join** | Matching rows based on same column name | □ No | □ Auto-detects |

## INNER JOIN (Preferred Equi Join Style)

```
SELECT Student.Name, Marks.Subject, Marks.Marks
FROM Student
INNER JOIN Marks
ON Student.RollNo = Marks.RollNo;
```

Same as equi-join but **more readable and standard**.