

# Joins Assignment 2

---

## ◆ 1. Left Join vs Right Join

Feature	LEFT JOIN	RIGHT JOIN
Definition	Returns all rows from the <b>left table</b> and matching rows from the right. Non-matching rows from right are NULL.	Returns all rows from the <b>right table</b> and matching rows from the left. Non-matching rows from left are NULL.
Priority Table	Left table is preserved.	Right table is preserved.
Usage	Commonly used (more intuitive).	Less commonly used (same result can often be obtained by swapping tables in a LEFT JOIN).

### Example:

-- Left Join

```
SELECT E.name, D.dept_name
FROM Employee E
LEFT JOIN Department D ON E.dept_id = D.dept_id;
```

-- Right Join

```
SELECT E.name, D.dept_name
FROM Employee E
RIGHT JOIN Department D ON E.dept_id = D.dept_id;
```

---

## ◆ 2. ON Clause vs USING Clause

Feature	ON Clause	USING Clause
Syntax	JOIN ... ON condition	JOIN ... USING(column)
Flexibility	Can join on different column names or complex conditions.	Only works when the column names are <b>same</b> in both tables.
Explicitness	More verbose, requires fully qualified column names.	Simpler, avoids repetition if column names match.
Example	sql SELECT E.name, D.dept_name FROM Employee E JOIN Department D ON E.dept_id = D.dept_id;	sql SELECT E.name, D.dept_name FROM Employee JOIN Department USING(dept_id);

---

### ◆ 3. Natural Join vs Inner Join

Feature	NATURAL JOIN	INNER JOIN
Definition	Automatically joins tables based on all columns with the <b>same name</b> and compatible data types.	Joins tables based on a specified condition using ON or USING.
Condition	Implicit (database engine finds common columns).	Explicit (developer defines join condition).
Control	Less control; may join on unintended columns if multiple names match.	Full control over which columns to join.
Example	sql SELECT * FROM Employee NATURAL JOIN Department;	sql SELECT * FROM Employee INNER JOIN Department ON Employee.dept_id = Department.dept_id;

---

### ◆ 4. Inner Join vs Left Join

Feature	INNER JOIN	LEFT JOIN
Definition	Returns only rows with <b>matching values</b> in both tables.	Returns all rows from the <b>left table</b> , even if no match in the right (with NULLs).
Result Size	Smaller or equal to either table (only matched rows).	Always includes all rows from the left table.
Null Handling	Non-matching rows are excluded.	Non-matching rows from right are filled with NULL.
Example	sql SELECT E.name, D.dept_name FROM Employee E INNER JOIN Department D ON E.dept_id = D.dept_id;	sql SELECT E.name, D.dept_name FROM Employee E LEFT JOIN Department D ON E.dept_id = D.dept_id;