# Joins

Trishla Shah

trishla.shah@smu.ca

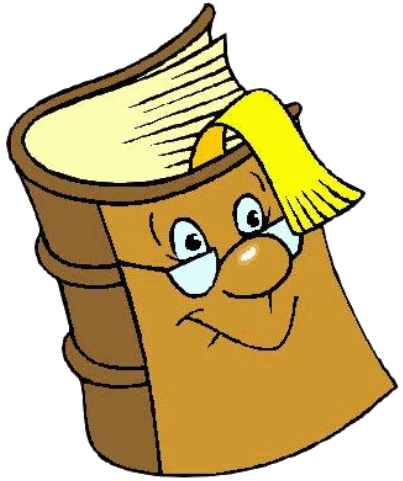
# Join

- Join is a special form of cross product of two tables.

- In Cartesian product we join a tuple of one table with the tuples of the second table. But in join there is a special requirement of relationship between tuples.

# Example

- For example if there is  a relation STUDENT  and a relation  BOOK then it may be required to know that how many  books have been issued to any particular student. Now in  this case the primary key of STUDENT that is stId is a foreign key in BOOK table through which the join can be made.



# SQL Joins

- SQL  joins are used to get data from two or more tables bases on relationship  between some of the columns in tables.

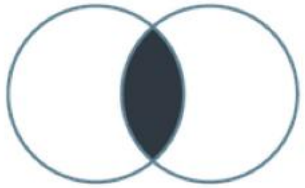
- In most of the cases we will use primary  key of first table and foreign key of  secondary table to get data from tables

# Overview of Join Types

- Inner Join
- Left Join or Left Outer Join
- Right Join or Right Outer Join
- Full Outer
- Self Join

# Overview of Join Types

- Inner Join: When you use an inner join to combine tables, the result is a table that contains values that have matches in both tables.



Inner Join

- Left Join: When you use a left join to combine tables, the result is a table that contains all values from the left table and corresponding matches from the right table.

Left Join

# Overview of Join Types

- Right Join: When you use a right join to combine tables, the result is a table that contains all values from the right table and corresponding matches from the left table.

Right Join

- Full Outer Join: When you use a full outer join to combine tables, the result is a table that contains all values from both tables.

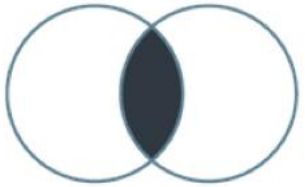
Full outer Join

# Tables for examples

| EMPNO | ENAME | JOB | MGR | DEPTNO |
|---|---|---|---|---|
| 111 | Trishla | Analyst | 444 | 10 |
| 222 | Khilan | Clerk | 333 | 20 |
| 333 | Kaushik | Manager | 111 | 10 |
| 444 | Chaitali | Engineer | 222 | 40 |

| DEPTNO | DNAME | LOC |
|---|---|---|
| 10 | INVENTORY | Halifax |
| 20 | FINANCE | Ottawa |
| 30 | HR | Toronto |

# Inner Join

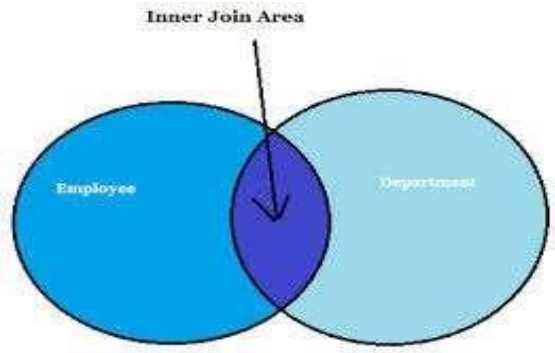



- Clause is used to combine rows from two or more tables,  based on a common field between them.
- INNER JOIN creates a new result table by combining  column values of two tables (table1 and table2) based  upon the join-condition.
- Keyword used is INNER JOIN or simply JOIN.

# Syntax

```
SELECT table1.column1, table2.column2...
FROM table1 INNER JOIN table2
ON  table1.common_field = table2.common_field;
```

# Example

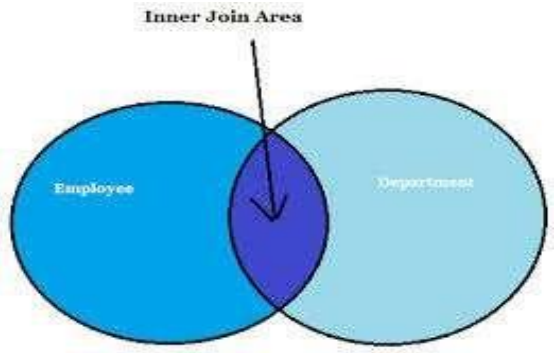



| EMPNO | ENAME | JOB | MGR | DEPTNO |
|---|---|---|---|---|
| 111 | Trishla | Analyst | 444 | 10 |
| 222 | Khilan | Clerk | 333 | 20 |
| 333 | Kaushik | Manager | 111 | 10 |
| 444 | Chaitali | Engineer | 222 | 40 |

| DEPTNO | DNAME | LOC |
|---|---|---|
| 10 | INVENTORY | Halifax |
| 20 | FINANCE | Ottawa |
| 30 | HR | Toronto |

```
SELECT  emp.ename,emp.job,dept.dname,dept.loc
FROM EMPLOYYE emp  INNER JOIN DEPARTMENT dept
ON emp.deptno = dept.deptno;
```

# Example - Result




| ENAME | JOB | DNAME | LOC |
|---|---|---|---|
| Trishla | Analyst | INVENTORY | Halifax |
| Khilan | Clerk | FINANCE | Ottawa |
| Kaushik | Manager | INVENTORY | Halifax |

# Left Join/Left Outer Join


- LEFT JOIN returns all rows from the left table (table1),  even if there are no matches in the right table(table2).

- If there are no matches found in right table the result is  null in the right table.

- Keyword used is LEFT JOIN.

# Syntax

```
SELECT table1.column1, table2.column2...
FROM table1
LEFT JOIN table2 ON table1.common_field =  table2.common_field;
```

# Example

| ENAME | JOB | DNAME | LOC |
|---|---|---|---|
| Trishla | Analyst | Inventory | Halifax |
| Khilan | Clerk | Finance | Ottawa |
| Kaushik | Manager | Inventory | Halifax |
| Chaitali | Engineer | NULL | NULL |

```
SELECT emp.ename,emp.job,dept.dname,dept.loc from EMPLOYYE emp  LEFT JOIN
DEPARTMENT dept ON emp.deptno = dept.deptno;
                                                    OR
SELECT emp.ename,emp.job,dept.dname,dept.loc from EMPLOYYE emp ,
DEPARTMENT dept WHERE emp.deptno(+) =dept.deptno;
```

# Right Join/Right Outer Join


- RIGHT JOIN returns all rows from the right table (table2),  even if there are no matches in the left table(table1).

- If there are no matched found in let table the result is null  in the left table.

- Keyword used is RIGHT JOIN

# Syntax

```
SELECT table1.column1, table2.column2...  FROM table1
RIGHT JOIN table2 ON table1.common_field =  table2.common_field;
```

# Example

| ENAME | JOB | DNAME | LOC |
|---|---|---|---|
| Trishla | Analyst | Inventory | Halifax |
| Khilan | Clerk | Finance | Ottawa |
| Kaushik | Manager | Inventory | Halifax |
| NULL | NULL | HR | Toronto |

```
SELECT emp.ename,emp.job,dept.dname,dept.loc from EMPLOYYE emp  RIGHT JOIN DEPARTMENT
dept ON emp.deptno =dept.deptno;
                                                    OR
SELECT emp.ename,emp.job,dept.dname,dept.loc from EMPLOYYE emp,
DEPARTMENT dept WHERE emp.deptno = dept.deptno(+);
```

# Full Join


- The SQL FULL JOIN combines the results of both left and right outer joins.
- The joined table will contain all records from both tables, and fill in NULLs for missing matches on either side.
- Keyword used is FULL JOIN. In some data bases it is also known as FULL OUTER JOIN.

# Syntax

```
SELECT table1.column1, table2.column2...  FROM table1
FULL JOIN table2 ON table1.common_field =  table2.common_field;
```

# Example

| ENAME | JOB | DNAME | LOC |
|---|---|---|---|
| Trishla | Analyst | Inventory | Halifax |
| Khilan | Clerk | Finance | Ottawa |
| Kaushik | Manager | Inventory | Halifax |
| NULL | NULL | HR | Toronto |
| Chaitali | Engineer | NULL | NULL |

```
SELECT emp.ename,emp.job,dept.dname,dept.loc from EMPLOYYE emp  FULL JOIN
DEPARTMENT dept ON emp.deptno =dept.deptno;
```

# Self Join

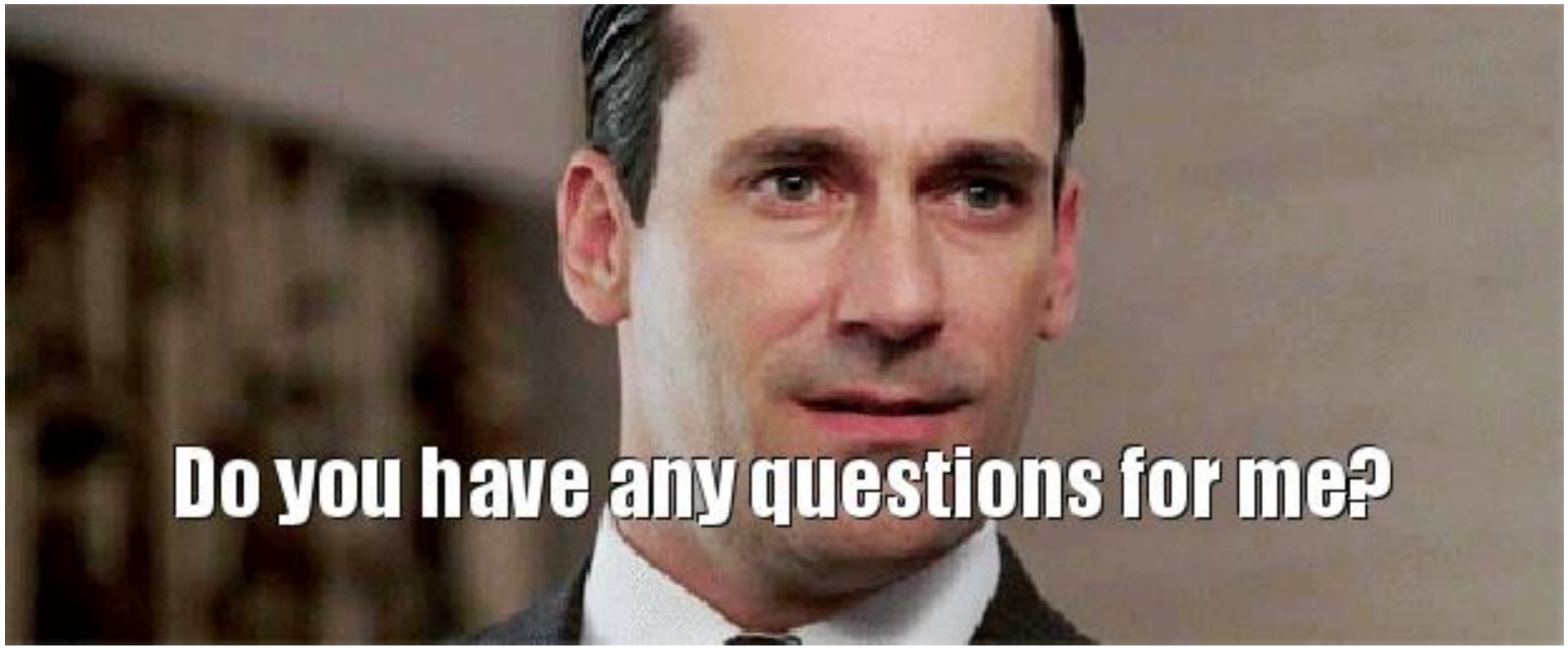
- SELF JOIN is used to join a table to itself as if the table were two tables

**Syntax** -> SELECT a.column_name, b.column_name... FROM table1 a, table1 b WHERE a.common_field = b.common_field

# Example

| ENAME | JOB | MANAGER |
|---|---|---|
| Trishla | Analyst | Chaitali |
| Khilan | Clerk | Kaushik |
| Kaushik | Manager | Ramesh |
| Chaitali | Engineer | Khilan |

```
SELECT emp1.ename, emp1.job, emp2.ename as manager  FROM EMPLOYEE
emp1, EMPLOYEE emp2 where emp1.mgr =  emp2.empno;
```


Do you have any questions for me?