# Joins

Whenever we need to retrieve data from 2 or more table combined we use joins. A join with the condition omitted forms cartesian product.

## *Types Of Joins*

There are many types of joins in SQL but the ones proprietary to Oracle are as discussed in the lectures are as follows

- **Equi-join**
- **Non-equijoin**
- **Outer join**
- **Self join**

## *Equi-Joins*

A join in which primary and the foreign key are checked for equality condition resulting in only the rows where primary and foreign keys are equal is called Equi-Join.

Example

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers,Orders
Where Customers.CustomerID = Orders.CustomerID
```

## *Non Equi-Joins*

A join in which primary and the foreign key are checked for some condition other than equality such as (>,<,<=,<=,Between etc) resulting in the rows where primary and foreign keys have some relation other than equality is called Non-Equi-Join.

Example

```
SELECT Customers. cartvalue, Orders. cartvalue
FROM Customers,Orders
Where Customers.cartvalue > Orders.cartvalue
```

# *Outer Joins*

In outer joins we can also see rows which do not meet the conditions. There are 2 main types of outer Joins.

**Left outer join and Right outer Join.**

In **left outer join** we receive the Whole Left table regardless of the condition and the right table with only those rows which satisfy the condition.

Example

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers,Orders
Where Customers.CustomerID = Orders.CustomerID(+)
```

In **Right outer join** we receive the Whole Right table regardless of the condition and the Left table with only those rows which satisfy the condition.

Example

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers,Orders
Where Customers.CustomerID (+)= Orders.CustomerID
```

# *Self Joins*

When we join a table with itself using Alias it is called self join.

Example

```
SELECT a.emp_id AS "Emp_ID",a.emp_name AS "Employee Name",
b.emp_id AS "Supervisor ID",b.emp_name AS "Supervisor Name"
FROM employee a, employee b
WHERE a.emp_supv = b.emp_id;
```

# *SQL Joins*

- **Cross joins**
- **Natural joins**
- **Full or two sided outer joins**

# *Cross Joins*

Genrates the Cross product.

Example

**SELECT ***

**From emp**

**Cross Join dept**

# *Natural Joins*

The NATURAL JOIN clause is based on all columns in the two tables that have the same name. It selects rows from the two tables that have equal values in all matched columns. If the columns having the same names have different data types, an error is returned.Example

**SELECT department_id,location_id, city**

**FROM departments**

**NATURAL JOIN locations;**

# *Full Joins*

The SQL FULL JOIN combines the results of both left and right outer joins.

The joined table will contain all records from both the tables and fill in NULLs for missing matches on either side.

**SELECT dept.deptno, emp.ename**

**FROM dept,emp**

**FULL JOIN emp**

**ON dept. deptno = emp.deptno;**