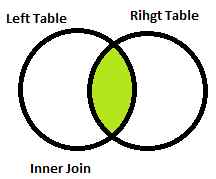
**JOINS**  
The process of combining data from more than one table using a single select statement is called joining. Within a relational database, the data will not be stored in a single table. Hence a join is needed when we need to combine data from more than one table.  
  
I have two tables to implement joins, Mas\_Employee and Mas\_Department.  
  
  
  
Joining in SQL Server is classified as follows:

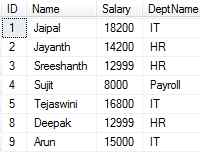
* Inner Join
* Outer Join: Left Outer Join, Right Outer Join, Full Outer Join
* Cross Join
* Self Join

The basic **syntax** to implement joins is as in the following:

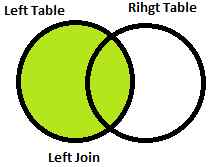
1. **Select** {Column\_List}
2. **From** {Main\_Table}
3. JoinType {Refernce\_Table}
4. **On** {Condition}

**INNER JOIN**  
It returns only the matching rows from the various tables. By default a join is an Inner Join. In the following example the output (the employee ID 6, 7 Employee details) are not shown, because the DeptId 5, 4 do not exist in the Mas\_Department.  
  
  
  
**Example:**Display Employee Details along with department name.

1. **Select** E.**Name**, E.Salary, D.DeptName
2. **From** Mas\_Employee E -- E is an alias name for Mas\_Employee
3. Join Mas\_Department D -- D is an alias name for Mas\_Department
4. **On** E.DeptId = D.DeptId

**Output**  
  
  
  
**OUTER JOIN:** Outer Joins are again divided into the following 3 types.

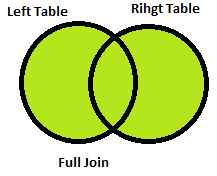
1. Left Join or Left Outer Join
2. Right Join or Right Outer Join
3. Full Join or Full Outer Join

**Left Outer Join**  
It returns all the matching rows and non-matching rows from the left table. A Left Outer Join can also be called a Left Join.  
  
  
  
**Example**

1. **Select** E.**Name**, E.Salary, D.DeptName
2. **From** Mas\_Employee E
3. Left Join Mas\_Department D
4. **On** E.DeptId = D.DeptId

**Output**  
  
  
  
**Right Outer Join**  
It returns all the matching rows and non-matching rows from the right table. A Right Outer Join can also be called a Right Join.  
  
**Example**

1. **Select** E.**Name**, E.Salary, D.DeptName
2. **From** Mas\_Employee E
3. Right Join Mas\_Department D
4. **On** E.DeptId = D.DeptId

**Output**  
  
  
  
**Full Outer Join**  
It returns all the matching rows and non-matching rows from both tables. A Full Outer Join can also be called a Full Join.  
  
  
  
**Example**

1. **Select** E.**Name**, E.Salary, D.DeptName
2. **From** Mas\_Employee E
3. **Full** Join Mas\_Department D
4. **On** E.DeptId = D.DeptId

**Output**  
  
  
  
**CROSS JOIN**  
The join statement without any join condition is called a Cross Join. The result of this Cross Join is called a Cartesian Product. In other words, if Mas\_Employee has 9 rows and Mas\_Department has 3 rows then generate 9 X 3 = 27 rows in the output.  
  
**Example:** Display Employee Details by associating every employee with every department.

1. **Select** E.**Name**, E.Salary, D.DeptName
2. **From** Mas\_Employee E
3. Right Join Mas\_Department D

**SELF JOIN**  
The join statement that joins a table to itself is called a Self Join.  
  
**Example:**Display Employee Details along with his/her manager's name.

1. **Select** E.ID, E.**Name**, E.Sal, E.Manager, M.**Name**
2. **From** Employee E
3. Join Employee M
4. **On** E.Manager=M.ID

**Note:**A rule to be followed when writing a join statement is the number of join conditions must be one less than the number of tables on which you join.