- **The Join Operation**

Join operator is used to retrieve data from multiple table or relations.

Syntax:- <tablename>⋈<tablenama>

There are various types of join in relational algebra.The join operation allows us to combine a selection and a Cartesian product into a single operation.

- **Natural Join**

It is a binary operator, written as (R S), where R and S are two relations. The result of this operation is the set of all tuple-combinations in R and S equal on their common attribute names. For example, natural join of two tables named Employee and Dept is (Employee ⋈Dept)

Employee

Name	ID	Dept_N
Hari	3411	Finance
Shalini	2242	Sales
Ganesh	3403	Finance
Hemant	2207	Sales

Dept

Dept_N	Manager
Finance	Ganesh
Sales	Hemant
Production	Charu

Employee ⋈ Dept

Name	ID	Dept_N	Manager
Hari	3411	Finance	Ganesh
Shalini	2242	Sales	Hemant
Ganesh	3403	Finance	Ganesh
Hemant	2207	Sales	Hemant

The natural join is the relational counterpart of logical AND, and has great importance. The natural join permits combination of relations associated by a foreign key. For example, in the previous table, a foreign key probably holds from Employee.Dept_N to Dept.Dept_N, and then the natural join of Employee and Dept combines every employee with their respective department. Note that this works because the foreign key holds between attributes with the same name.

- **Outer join**

An outer join is a type of join in RDBMS that returns all rows from one or both tables, even if there is no match in the other table. Unlike an inner join, which only includes rows with matching values in both tables, an outer join includes unmatched rows and fills in NULL values where no match exists.

 Types of outer join

o Left outer join

o Right outer join

o Full outer join

Example

Emp

Name	City
Tamil	Kvp
Selva	Chml
Kavi	Gopi
Kutty	Chennai

Emp_salary

Name	Dept	Salary
Tamil	CSE	25000
Selva	IT	23000
Durai	IT	20000
Kutty	CSE	30000

Natural Join

Emp ⋈ Emp_Salary

Name	City	Dept	Salary
Tamil	Kvp	CSE	25000
Selva	Chml	IT	23000
Kutty	Chennai	CSE	30000

Here, details of "kavi & durai" are lost. To avoid the loss of information, we can use outer join

- **Left Outer Join (or Left Join)**
- Returns all rows from the left table and the matching rows from the right table.
- If no match is found in the right table, NULL values are returned for columns from the right table.
- **SQL Syntax**:

```
SELECT columns
FROM table1
LEFT OUTER JOIN table2
ON table1.column = table2.column;
```

Emp ⟕ Emp_Salary

Name	City	Dept	Salary
Tamil	Kvp	CSE	25000
Selva	Chml	IT	23000
Kavi	Gopi	NULL	NULL
Kutty	Chennai	CSE	30000

- **Right Outer Join (or Right Join)**

- Returns all rows from the right table and the matching rows from the left table.
- If no match is found in the left table, NULL values are returned for columns from the left table.

- **SQL Syntax**:
  SELECT columns FROM table1
  RIGHT OUTER JOIN table2
  ON table1.column = table2.column;

Emp ⟖ Emp_Salary

Name	City	Dept	Salary
Tamil	Kvp	CSE	25000
Selva	Chml	IT	23000
Durai	NULL	IT	20000
Kutty	Chennai	CSE	30000

- **Full Outer Join (or Full Join)**

- Returns all rows from both tables, with matches where available.
- If no match exists in either table, NULL values are returned for the non-matching side.
- **SQL Syntax**:

  SELECT columns FROM
  table1 FULL OUTER JOIN table2
  ON table1.column = table2.column;

Emp ⟗ Emp_Salary

Name	City	Dept	Salary
Tamil	Kvp	CSE	25000
Selva	Chml	IT	23000
Kavi	Gopi	NULL	NULL
Kutty	Chennai	CSE	30000
Durai	NULL	IT	20000

NOTE :-YOU CAN REFER THIS NOTES AND ALSO PREPARE YOUR [redacted] NOTES FOR BETTER REFERENCE.ADD SOME EXAMPLES