Joints - ~~or Joints in~~

Joins in SQL - (combine 2 table).

An SQL Join clause combines records from two or more tables in a database. It creates a set that can be saved as a table. A Join is a means of combining fields from two tables by using values common to each.

* Types of Join -

1) Inner Join
   - Equi Join
   - Natural Join
   - Cross Join
2) Outer Join
   - Self outer Join
   - Right outer Join
   - full outer Join
3) Self Join

consider the follo tables employee and department where department_id is a primary key and employee.department_id is a foreign key.

Employee.

Lastname	department_id
Ram	31
Amit	33
Amol	33
Vijay	34
Sagar	34
Mayur	NULL

Mayur has not been assigned to any dept.

Department

department_id	department_name
31	comp
33	elex
34	mech
35	civil

Civil department currently has no employee.

Inner Join
equi join
natural join

(1) Inner Join -

An inner join is the most common join operation we in applications and represents default join type. Inner Join creates a new result table by combining common values of two tables. (A and B)

Based upon the join predicate. The query compares each row of A with each row of B to find all pairs of rows which satisfies the join predicate.

```
select *
from employee
    inner join department
on employee.department_id =
    department.department_id ;
```

employee.lastname	employee.department_id	department.department_name	department.department_id
Ram	31	Comp	31
Amit	33	elex	33
Amol	33	elex	33
Vijay	34	mech	34
sagar	34	mech	34 .

Equi - Join - An equi join is a specific type of comparator based join that uses equality comparisons in join predicate using other comparison operator (e.g. <) this qualifies a join as an equi join.

e.g. The query shown above is an ex. of equi. join. SQL provides an optional short hand notation for expressing equi. joins by using the 'using' construct.

```
select * from employee
    equi join department using (department id);
```

ii) Natural Join -

A natural join offers further specialization of equi join. The join predicate arises implicitely by comparing all columns in both tables that have same column name in the join table.

SQL> select * from employee
Natural join department; ↵

o/p :

department_id	employee. lastname	department. department_name
31	Ram	Comp
33	Amit	elex
33	Amol	elex
34	Vijay	mech
34	Sagar	mech

iii) cross join -

A cross join, cartitian join or product provides the foundation upon which all types of inner join operate. A cross join returns the cartisian product of the set of records from the 2 join tables.

1 to all

e.g. SQL> select *
from employee
cross join department; ↵

employee. lastname	employee. department_id	department. departmentname	department. department_id
Ram	31	comp	31
Amit	33	comp	31
Amol	33	comp	31
Vijay	34	comp	31
Sagar	34	comp	31
mayur	NULL	comp	31
Ram	31	elex	33
Amit	33	elex	33
Amol	33	elex	33
Vijay	34	elex	33
Sagar	34	elex	33
mayur	NULL	elex	33
Ram	31	mech	34
Amit	33	mech	34
Amol	33	mech	34
Vijay	34	mech	34
Sagar	34	mech	34
mayur	NULL	mech	34
Ram	31	civil	35
Amit	33	civil	35
Amol	33	civil	35
Vijay	34	civil	35
Sagar	34	civil	35
mayur	NULL	civil	35

The cross join does not apply any predicate to filter records from the join table.

Programmers can further filter the result by using where clause

(2) Outer Join -

An outer join does not require each record in the two join tables to have a matching record. The join table retains each record even if no other matching record exist.

i) Left outer join -

write all records of left table & write the match of right table with left table & if not found then write NULL in that position.

The result of a left outer join for table A & B always contains all the records of left table (A), even if the join condition does not find any matching record in right table (B).

e.g.

```
SQL> select *
     from employee
     left outer join department
     ON employee.department_id =
        department.department_id ;↵
```

o/p -

employee.lastname	employee.department_id	department.department_name	department.department_id
Ram	31	comp	31
Amit	33	elex	33
Amol	33	elex	33
Vijay	34	mech	34
Sagar	34	mech	34
Mayur	NULL	NULL	NULL

ii) Right outer join -

In right outer join every row from the right table (B) will appear in the join table atleast one, if no matching row from left table (A)

exist, null will appear in columns from A for those records that have no matching in A.

```
SQL> select *
     from employee
     right outer join department
     ON employee.department_id =
        department.department_id ;↵
```

employee. lastname	employee. department_id	department. department_name	department. department_id
Ram	31	comp	31
Amit	33	elex	33
Amol	33	elex	33
Vijay	34	mech	34
Sagar	34	mech	34
NULL	NULL	civil	35

iii) Full outer join -

A full outer join combines the result of left & right join. The join table will contain all records from both sides and fill in NULL's for missing matches on either side.

e.g.

```
SQL> select *
     from employee
     full outer join department
     ON employee.department_id =
        department.department_id ;↵
```

ployee.	employee.	department.	department
lastname	department_id	department_name	department_id
Ram	31	comp	31
Amit	33	elex	33
Amol	33	elex	33
vijay	34	mech	34
Sagar	34	mech	34
Mayur	NULL	NULL	NULL
NULL	NULL	civil	35