Cartesian product (X)/cross joint

Cartesian Product is denoted by X symbol. Lets say we have two relations R1 and R2 then the cartesian product of these two relations (R1 X R2) would combine each tuple of first relation R1 with the each tuple of second relation R2.

SQL

Cartesian product (X) example

Table a and Table b as shown below

```
mysql> select * from a;
+---------+-------+
| Name    | val   |
+---------+-------+
| vishal  |    11 |
| ram     |    22 |
+---------+-------+
2 rows in set (0.00 sec)

mysql> select * from b;
+----------+
| name     |
+----------+
| ram      |
| vikrant  |
+----------+
2 rows in set (0.00 sec)
```

Mysql query –

Select * from a,b;

Select * from a cross join b;

```
mysql> select * from a,b;
+---------+------+---------+
| Name    | val  | name    |
+---------+------+---------+
| vishal  |   11 | ram     |
| ram     |   22 | ram     |
| vishal  |   11 | vikrant |
| ram     |   22 | vikrant |
+---------+------+---------+
4 rows in set (0.00 sec)
```

Degree of cartesion product is 3 and cardinality is 4=(2 rows of a X 2 rows of b)

SQL

Join – Join is used to fetch data from two or more tables, which is joined to appear as single set of data. It is used for combining column from two or more tables by using values common to both tables.

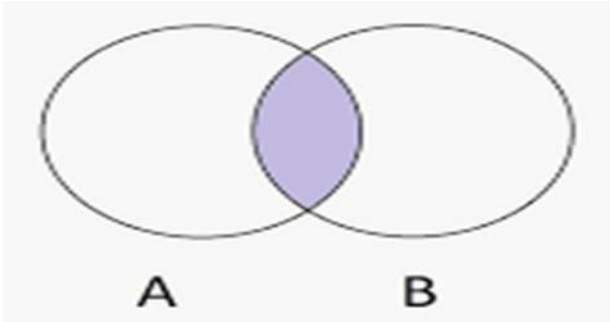
Types of JOIN

Following are the types of JOIN that we can use in SQL:

- Inner
- Outer
- Left
- Right

INNER Join or EQUI Join⋈

This is a simple JOIN in which the result is based on matched data as per the equality condition specified in the SQL query.



SQL

INNER Join or EQUI Join example

Table a and Table b as shown below

```
mysql> select * from a;
+---------+-------+
| Name    | val   |
+---------+-------+
| vishal  |    11 |
| ram     |    22 |
+---------+-------+
2 rows in set (0.00 sec)

mysql> select * from b;
+----------+
| name     |
+----------+
| ram      |
| vikrant  |
+----------+
2 rows in set (0.00 sec)
```

Mysql query –

Select course.student_name from couse , student where course.student_name=student.stude nt_name;

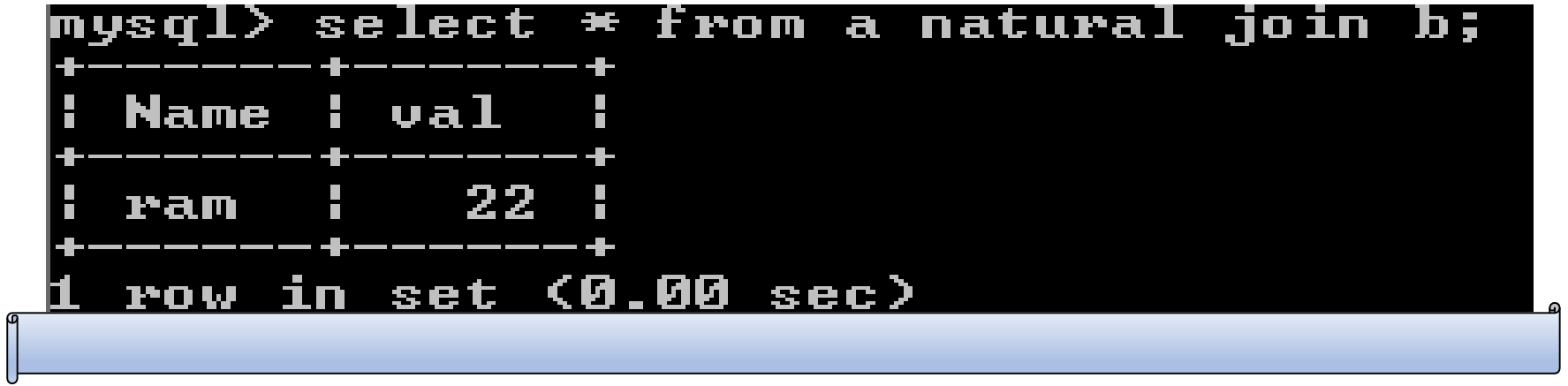
Select a.name from a inner join b where a.name=b.name;

```
mysql> select a.name from a inner join b where a.name=b.name;
+------+
| name |
+------+
| ram  |
+------+
```

Natural JOIN(⋈)

Natural Join is a type of Inner join which is based on column having same name and same datatype present in both the tables to be joined.E.g.

Select * from a natural join b;



LEFT Outer Join ⟕

The left outer join returns a resultset table with the matched data from the two tables and then the remaining rows of the left table and null from the right table's columns. E.g.

```
mysql> select * from a;
+---------+-------+
| Name    | val   |
+---------+-------+
| vishal  |    11 |
| ram     |    22 |
+---------+-------+
2 rows in set (0.00 sec)

mysql> select * from b;
+----------+
| name     |
+----------+
| ram      |
| vikrant  |
+----------+
2 rows in set (0.00 sec)
```

Mysql query –

Select * from a left outer join b on (a.name=b.name);

```
mysql> select * from a left outer join b on (a.name=b.name);
+-------+-----+------+
| Name  | val | name |
+-------+-----+------+
| vishal |  11 | NULL |
| ram    |  22 | ram  |
+-------+-----+------+
2 rows in set (0.02 sec)
```

RIGHT Outer Join

The right outer join returns a resultset table with the matched data from the two tables being joined, then the remaining rows of the right table and null for the remaining left table's columns.E.g.

```
mysql> select * from a;
+---------+-------+
| Name    | val   |
+---------+-------+
| vishal  |    11 |
| ram     |    22 |
+---------+-------+
2 rows in set (0.00 sec)

mysql> select * from b;
+----------+
| name     |
+----------+
| ram      |
| vikrant  |
+----------+
2 rows in set (0.00 sec)
```

Mysql query –

Select * from a right outer join b on (a.name=b.name);

```
mysql> select * from a right outer join b on (a.name=b.name);
+------+------+---------+
| Name | val  | name    |
+------+------+---------+
| ram  |   22 | ram     |
| NULL | NULL | vikrant |
+------+------+---------+
2 rows in set (0.00 sec)
```

Full Outer Join

The full outer join returns a resultset table with the matched data of two table then remaining rows of both left table and then the right table.E.g.

```
mysql> select * from a;
+---------+-------+
| Name    | val   |
+---------+-------+
| vishal  |    11 |
| ram     |    22 |
+---------+-------+
2 rows in set (0.00 sec)

mysql> select * from b;
+----------+
| name     |
+----------+
| ram      |
| vikrant  |
+----------+
2 rows in set (0.00 sec)
```

Mysql query –

Select * from a left outer join b on (a.name=b.name) union Select * from a right outer join b on (a.name=b.name) ;

```
mysql> select * from a left outer join b on(a.name=b.name) union select * from a
 right outer join b on (a.name=b.name);
+--------+------+---------+
| Name   | val  | name    |
+--------+------+---------+
| vishal |   11 | NULL    |
| ram    |   22 | ram     |
| NULL   | NULL | vikrant |
+--------+------+---------+
3 rows in set (0.01 sec)
```