Table Joins and Indexes in SQL

Fetching data from multiple tables
Fetching data in faster way

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Joining

A join is a query that combines rows from two of more tables. In JOIN query more than one table are listed in FROM clause. MySQL provides various type of Joining :

1) *CROSS JOIN or CARTESIAN PRODUCT*
2) *EQUI-JOIN*
3) *NATURAL JOIN*

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Cross Join (Cartesian product)

- It return all possible concatenation of all rows from both table i.e. one row of First table is joined with all the rows of second table.
- Cartesian product join each row of one table with each row of another table. So if –
- First table have **6 rows** and second table have **4 rows** then total number of rows in output will be **6 x 4 = 24.**
- i.e. Total Number of Rows after Cartesian product(**Cardinality**) = **Cardinality of First Table X Cardinality of Second Table**

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Cross Join (Cartesian product)

COLOR	SHADES
RED	LIGHT
GREEN	CYAN
BLUE	SILVER

- Suppose I want a combination of all colors with all shades. In this case Cartesian product or cross join is used.
- **For Example**
  - **Select * from Shades,Color**
- **Output will contain 9 rows i.e. no. of rows in first table x no. of rows in second table**

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Cross Join (Cartesian product)

```
mysql> select * from color;
+--------+
| name   |
+--------+
| red    |
| yellow |
| green  |
+--------+
3 rows in set (0.00 sec)

mysql> select * from shades;
+--------+
| sname  |
+--------+
| light  |
| silver |
| golden |
+--------+
3 rows in set (0.01 sec)
```

```
mysql> select * from shades,color;
+--------+--------+
| sname  | name   |
+--------+--------+
| light  | red    |
| silver | red    |
| golden | red    |
| light  | yellow |
| silver | yellow |
| golden | yellow |
| light  | green  |
| silver | green  |
| golden | green  |
+--------+--------+
9 rows in set (0.00 sec)
```

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Cross Join (Cartesian product)

```
mysql> select concat(sname,' ',name) from shades,color;
+-------------------------+
| concat(sname,' ',name) |
+-------------------------+
| light red               |
| silver red              |
| golden red              |
| light yellow            |
| silver yellow           |
| golden yellow           |
| light green             |
| silver green            |
| golden green            |
+-------------------------+
9 rows in set (0.00 sec)

mysql>
```

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Equi-join

- *The join in which columns are compared for equality is called Equi-Join. A non-equi join specifies condition with non-equality operator. In equi-join we put(*) in the select list therefore the common column will appear twice in the output.*
- *To understand the output, lets take 2 table one for employee (contains employee detail with deptno) and another for department contains deptno and other department details.*

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Equi-join

Now we want to fetch details of employee along with its corresponding matching department. Like for 'alam' deptno is 10 so from dept table it should show deptno 10 details and so on

```
mysql> select * from emp;
+-------+----------+--------+--------+
| empno | ENAME    | DEPTNO | salary |
+-------+----------+--------+--------+
|     1 | alam     |     10 |  10300 |
|     2 | srijeeta |     20 |   6220 |
|     3 | bhaskar  |     30 |  11320 |
|     4 | emely    |     10 |  20500 |
|     5 | freddy   |     30 |  11320 |
|     7 | chanop   |     10 |  51100 |
|     8 | akshay   |     20 |  30700 |
|     9 | manish   |     20 |  46000 |
|    10 | nitin    |     20 |  78100 |
|    11 | naveen   |     20 |   9000 |
|    12 | Kirti    |     20 |   9000 |
|    13 | Gabbar   |     30 |  12100 |
|    14 | sunny    |     20 |   NULL |
+-------+----------+--------+--------+
13 rows in set (0.03 sec)

mysql> select * from dept;
+--------+------------+---------+
| deptno | dname      | dhead   |
+--------+------------+---------+
|     10 | Sales      | Ritika  |
|     20 | HR         | Ankit   |
|     30 | Production | Abuzair |
|     40 | IT         | Mesha   |
+--------+------------+---------+
4 rows in set (0.00 sec)
```

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Equi-join

```
mysql> select * from emp, dept where emp.deptno = dept.deptno;
+-------+----------+--------+--------+--------+------------+---------+
| empno | ENAME    | DEPTNO | salary | deptno | dname      | dhead   |
+-------+----------+--------+--------+--------+------------+---------+
|     1 | alam     |     10 |  10300 |     10 | Sales      | Ritika  |
|     2 | srijeeta |     20 |   6220 |     20 | HR         | Ankit   |
|     3 | bhaskar  |     30 |  11320 |     30 | Production | Abuzair |
|     4 | emely    |     10 |  20500 |     10 | Sales      | Ritika  |
|     5 | freddy   |     30 |  11320 |     30 | Production | Abuzair |
|     7 | chanop   |     10 |  51100 |     10 | Sales      | Ritika  |
|     8 | akshay   |     20 |  30700 |     20 | HR         | Ankit   |
|     9 | manish   |     20 |  46000 |     20 | HR         | Ankit   |
|    10 | nitin    |     20 |  78100 |     20 | HR         | Ankit   |
|    11 | naveen   |     20 |   9000 |     20 | HR         | Ankit   |
|    12 | Kirti    |     20 |   9000 |     20 | HR         | Ankit   |
|    13 | Gabbar   |     30 |  12100 |     30 | Production | Abuzair |
|    14 | sunny    |     20 |   NULL |     20 | HR         | Ankit   |
+-------+----------+--------+--------+--------+------------+---------+
13 rows in set (0.02 sec)

mysql>
```

Common Column appears twice in output

From the above query, we can observe that while doing equi-join we have to give equality condition on common column of both table so that it picks related records

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Equi-join

We can also give Table Alias i.e. nick name for table name and further we can use this name any where in query in place of table name. This is helpful when table name is of big length and we can shorten the query

```
mysql> select * from emp e, dept d where e.deptno = d.deptno;
+-------+---------+--------+--------+--------+------------+---------+
| empno | ENAME   | DEPTNO | salary | deptno | dname      | dhead   |
+-------+---------+--------+--------+--------+------------+---------+
|     1 | alam    |     10 |  10300 |     10 | Sales      | Ritika  |
|     2 | srijeeta|     20 |   6220 |     20 | HR         | Ankit   |
|     3 | bhaskar |     30 |  11320 |     30 | Production | Abuzair |
|     4 | emely   |     10 |  20500 |     10 | Sales      | Ritika  |
|     5 | freddy  |     30 |  11320 |     30 | Production | Abuzair |
|     7 | chanop  |     10 |  51100 |     10 | Sales      | Ritika  |
|     8 | akshay  |     20 |  30700 |     20 | HR         | Ankit   |
|     9 | manish  |     20 |  46000 |     20 | HR         | Ankit   |
|    10 | nitin   |     20 |  78100 |     20 | HR         | Ankit   |
|    11 | naveen  |     20 |   9000 |     20 | HR         | Ankit   |
|    12 | Kirti   |     20 |   9000 |     20 | HR         | Ankit   |
|    13 | Gabbar  |     30 |  12100 |     30 | Production | Abuzair |
|    14 | sunny   |     20 |   NULL |     20 | HR         | Ankit   |
+-------+---------+--------+--------+--------+------------+---------+
13 rows in set (0.00 sec)

mysql>
```

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Natural Join

- *The JOIN in which only one of the identical columns exists in called Natural Join. It is similar to Equi-join except that duplicate columns are eliminated in Natural join that would otherwise appear in Equi-Join.*
- *In natural join we specify the names of column to fetch in place of (*) which is responsible of appearing common column twice in output.*

Natural Join

See here, we are not giving *, like in equi-join but we are giving list of columns to fetch

```
mysql> select ename,salary,dhead from emp e,dept d where e.deptno=d.deptno;
+----------+--------+---------+
| ename    | salary | dhead   |
+----------+--------+---------+
| alam     |  10300 | Ritika  |
| srijeeta |   6220 | Ankit   |
| bhaskar  |  11320 | Abuzair |
| emely    |  20500 | Ritika  |
| freddy   |  11320 | Abuzair |
| chanop   |  51100 | Ritika  |
| akshay   |  30700 | Ankit   |
| manish   |  46000 | Ankit   |
| nitin    |  78100 | Ankit   |
| naveen   |   9000 | Ankit   |
| Kirti    |   9000 | Ankit   |
| Gabbar   |  12100 | Abuzair |
| sunny    |   NULL | Ankit   |
+----------+--------+---------+
13 rows in set (0.00 sec)

mysql>
```

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Natural Join

```
mysql> select empno,ename,deptno,dname,dhead from emp,dept where
    -> emp.deptno=dept.deptno;
ERROR 1052 (23000): Column 'deptno' in field list is ambiguous
mysql>
```

The reason of this error is – the deptno exists in both the table, so in this case if we are selecting or using only deptno then it becomes ambiguous from which table this deptno will be selected

To resolve this error, just qualify the common column by table name as **TableName.column name**

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Natural Join

```
mysql> select empno,ename,emp.deptno,dname,dhead from emp,dept where
    -> emp.deptno=dept.deptno;
+-------+----------+--------+------------+---------+
| empno | ename    | deptno | dname      | dhead   |
+-------+----------+--------+------------+---------+
|     1 | alam     |     10 | Sales      | Ritika  |
|     2 | srijeeta |     20 | HR         | Ankit   |
|     3 | bhaskar  |     30 | Production | Abuzair |
|     4 | emely    |     10 | Sales      | Ritika  |
|     5 | freddy   |     30 | Production | Abuzair |
|     7 | chanop   |     10 | Sales      | Ritika  |
|     8 | akshay   |     20 | HR         | Ankit   |
|     9 | manish   |     20 | HR         | Ankit   |
|    10 | nitin    |     20 | HR         | Ankit   |
|    11 | naveen   |     20 | HR         | Ankit   |
|    12 | Kirti    |     20 | HR         | Ankit   |
|    13 | Gabbar   |     30 | Production | Abuzair |
|    14 | sunny    |     20 | HR         | Ankit   |
+-------+----------+--------+------------+---------+
13 rows in set (0.00 sec)

mysql>
```

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Additional condition in joins

```
mysql> select empno,ename,emp.deptno,dname,dhead from emp,dept where
    -> emp.deptno=dept.deptno and dname='HR';
+-------+----------+--------+-------+-------+
| empno | ename    | deptno | dname | dhead |
+-------+----------+--------+-------+-------+
|     2 | srijeeta |     20 | HR    | Ankit |
|     8 | akshay   |     20 | HR    | Ankit |
|     9 | manish   |     20 | HR    | Ankit |
|    10 | nitin    |     20 | HR    | Ankit |
|    11 | naveen   |     20 | HR    | Ankit |
|    12 | Kirti    |     20 | HR    | Ankit |
|    14 | sunny    |     20 | HR    | Ankit |
+-------+----------+--------+-------+-------+
7 rows in set (0.03 sec)
```

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Joining Tables using **JOIN** clause of SQL SELECT

- Till now we have performed joining using traditional SQL method which is common to most of the RDBMS software now we will learn MySQL style of joining using JOIN clause
- MySQL support various options with JOIN
  - CROSS
  - NATURAL
  - ON
  - USING

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Cartesian product using JOIN

- Select * from shades JOIN color;
- Or
- Select * from shades CROSS JOIN color;

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Equi – Join using JOIN

- Select * from emp JOIN dept ON emp.deptno = dept.deptno;
- Select * from emp JOIN dept ON emp.deptno = dept.deptno where **salary>50000;**

Natural – Join using JOIN

- Select * from emp NATURAL JOIN dept

In NATURAL JOIN condition the join condition is not required it automatically joins based on the common column value