MySQL DML Commands - Joins and subqueries

By Prof. B.A.Khivsara

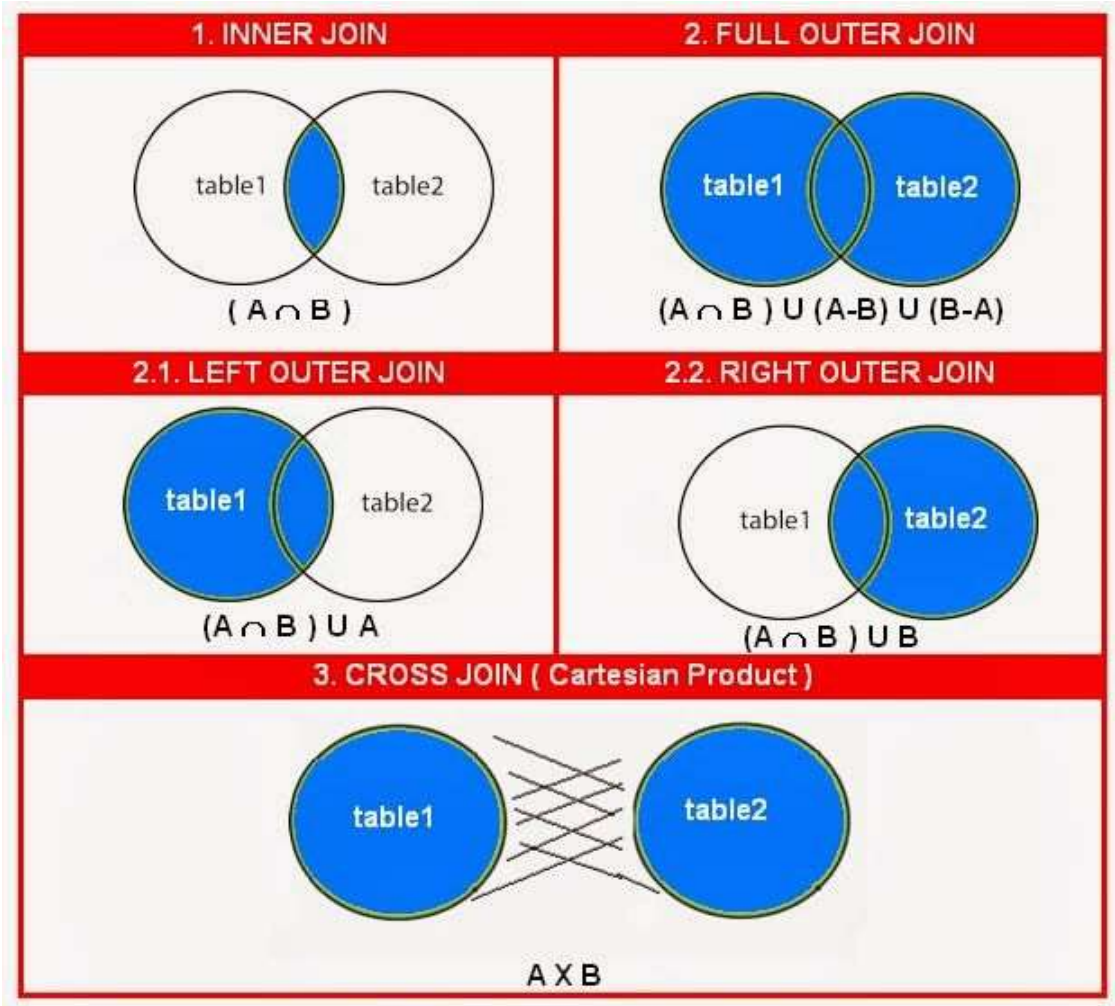
Design at least 10 SQL queries for suitable database application using SQL DML statements:

all types of Join,

Sub-Query and

View.

MySQL Joins


1. INNER JOIN
table1
table2
(A ∩ B)
2. FULL OUTER JOIN
table1
table2
(A ∩ B) U (A-B) U (B-A)
2.1. LEFT OUTER JOIN
table1
table2
(A ∩ B) U A
2.2. RIGHT OUTER JOIN
table1
table2
(A ∩ B) U B
3. CROSS JOIN (Cartesian Product)
table1
table2
A X B

MySQL Inner JOIN (Simple Join/Natural Join)

- The MySQL INNER JOIN is used to return all rows from multiple tables where the join condition is satisfied. It is the most common type of join.

- **Syntax 1:**

```
SELECT columns
FROM table1 ,table2
Where table1.column = table2.column;
```

- **Syntax 2:**

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

MySQL Inner JOIN Example

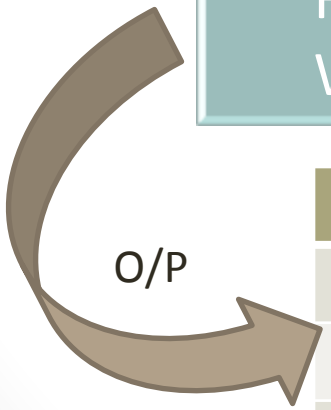
Stud_Info Table

RNo	Name	Address
1	Abhay	Nashik
2	Sarika	Pune
3	Riya	Nashik
4	Sachin	Manmad

Stud_Marks Table

RNo	Dbms	Toc
1	50	45
2	67	65
3	76	55
5	70	50

```
SELECT Stud_Info.Rno,Name,Dbms,Toc
FROM   Stud_Info,Stud_Marks
Where  Stud_Info.RNo= Stud_Marks.RNo
```



RNo	Name	Dbms	Toc
1	Abhay	50	45
2	Sarika	67	65
3	Riya	76	55

MySQL Left Outer Join

- The LEFT OUTER JOIN returns all rows from the left hand table specified in the ON condition and only those rows from the other table where the join condition is fulfilled.

- **Syntax:**

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

MySQL Left Outer Join Example

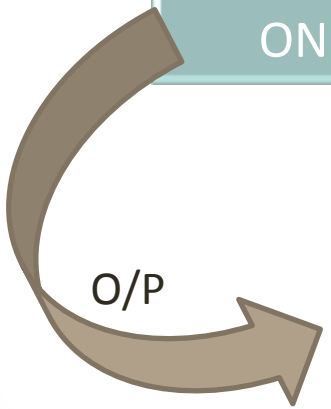
Stud_Info Table

RNo	Name	Address
1	Abhay	Nashik
2	Sarika	Pune
3	Riya	Nashik
4	Sachin	Manmad

Stud_Marks Table

RNo	Dbms	Toc
1	50	45
2	67	65
3	76	55
5	70	50

```
SELECT Stud_Info.Rno,Name,Dbms,Toc
FROM  Stud_Info LEFT  JOIN  Stud_Marks
ON   Stud_Info.RNo= Stud_Marks.RNo
```



RNo	Name	Dbms	Toc
1	Abhay	50	45
2	Sarika	67	65
3	Riya	76	55
4	Sachin	0	0

MySQL Right Outer Join

- The MySQL Right Outer Join returns all rows from the RIGHT-hand table specified in the ON condition and only those rows from the other table where he join condition is fulfilled.

- **Syntax:**

```
SELECT columns
FROM table1
RIGHT [OUTER] JOIN table2
ON table1.column = table2.column;
```

MySQL Right Outer Join Example

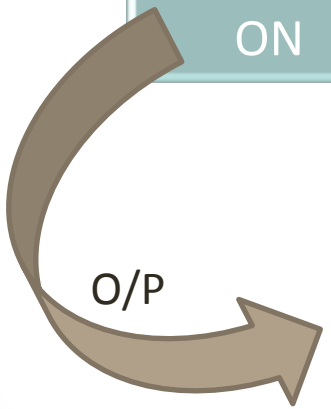
Stud_Info Table

RNo	Name	Address
1	Abhay	Nashik
2	Sarika	Pune
3	Riya	Nashik
4	Sachin	Manmad

Stud_Marks Table

RNo	Dbms	Toc
1	50	45
2	67	65
3	76	55
5	70	50

```
SELECT Stud_Info.Rno,Name,Dbms,Toc
FROM  Stud_Info RIGHT  JOIN  Stud_Marks
ON   Stud_Info.RNo= Stud_Marks.RNo
```



RNo	Name	Dbms	Toc
1	Abhay	50	45
2	Sarika	67	65
3	Riya	76	55
NULL	NULL	70	50

MySQL Right Outer Join Example

Stud_Info Table

RNo	Name	Address
1	Abhay	Nashik
2	Sarika	Pune
3	Riya	Nashik
4	Sachin	Manmad

Stud_Marks Table

RNo	Dbms	Toc
1	50	45
2	67	65
3	76	55
5	70	50

```
SELECT Stud_Marks.Rno,Name,Dbms,Toc
FROM  Stud_Info RIGHT  JOIN  Stud_Marks
ON   Stud_Info.RNo= Stud_Marks.RNo
```

O/P

RNo	Name	Dbms	Toc
1	Abhay	50	45
2	Sarika	67	65
3	Riya	76	55
5	NULL	70	50

MySQL Full Outer Join

- In MySQL Full out join is difficult as there is no simple command available for the same. But in Oracle SQL it’s possible.

SQL Full Outer Join Example

Stud_Info Table

RNo	Name	Address
1	Abhay	Nashik
2	Sarika	Pune
3	Riya	Nashik
4	Sachin	Manmad

Stud_Marks Table

RNo	Dbms	Toc
1	50	45
2	67	65
3	76	55
5	70	50

O/P After Full Outer Join

RNo	Name	Dbms	Toc
1	Abhay	50	45
2	Sarika	67	65
3	Riya	76	55
4	Sachin	0	0
5	NULL	70	50

MySQL CROSS JOIN

- A CROSS JOIN is such a join which specifies the complete cross product of two tables.
- For each record in the first table, all the records in the second table are joined, creating a potentially huge result set.
- This command has the same effect as leaving off the join condition, and its result set is also known as a Cartesian product.

- **Syntax**

```
SELECT Attr_list  FROM table_A    CROSS JOIN table_B;
```

Types of Subqueries

Single Row Sub Query: Sub query which returns single row output. They mark the usage of single row **comparison operators**, when used in WHERE conditions.

Multiple row sub query: Sub query returning multiple row output. They make use of multiple row comparison operators like **IN, ANY, ALL**. There can be sub queries returning multiple columns also.

Correlated Sub Query: Correlated subqueries depend on data provided by the outer query.This type of subquery also includes subqueries that use the **EXISTS** operator to test the existence of data rows satisfying specified criteria.

MySQL Subqueries- Single row Using Comparisons

Operator	Description
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
!=	Not equal to
<>	Not equal to
<=>	NULL-safe equal to operator

A subquery can be used before or after any of the comparison operators.

The subquery can return at most one value.

The value can be the result of an arithmetic expression or a column function.

MySQL Subqueries- Single Row Examples on Emp Table

Emp-id	Ename	City	Post	Salary
1	John	Nashik	Clerk	5000
2	Seema	Aurangabad	Developer	20000
3	Amita	Nagar	Manager	70000
4	Rakesh	Pune	Analyst	50000
5	Samata	Nashik	Tester	35000
6	Ankita	Chandwad	Developer	30000
7	Bhavika	Pune	Team-LR	50000
8	Deepa	Mumbai	CEO	90000
9	Nitin	Nagpur	Clerk	8000
10	Pooja	Pune	Analyst	45000

- **Display the information of employees, paid more than ‘pooja' from emp table**

```
Select *
from emp
 where salary  >
```

45000

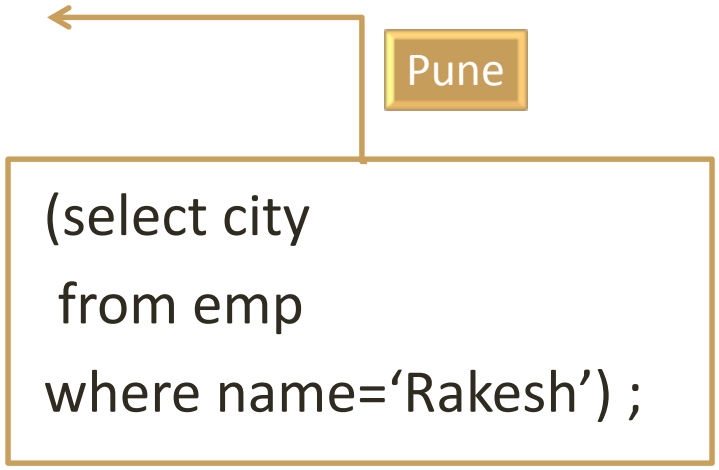
```
(select salary
 from emp
where name=‘Pooja’) ;
```

Output of Above Query

Emp-id	Ename	City	Post	Salary
3	Amita	Nagar	Manager	70000
4	Rakesh	Pune	Analyst	50000
7	Bhavika	Pune	Team-LR	50000
8	Deepa	Mumbai	CEO	90000

- **List the name of the employees, who live in the same city as of 'Rakesh'**

```
Select *
from emp
 where city =
```



Output of Above Query

Emp-id	Ename	City	Post	Salary
4	Rakesh	Pune	Analyst	50000
7	Bhavika	Pune	Team-LR	50000
10	Pooja	Pune	Analyst	45000

- **Display the information of employees, paid less salary than average salary throughout the company.**

```
Select *
from emp
 where salary  <
```

40300

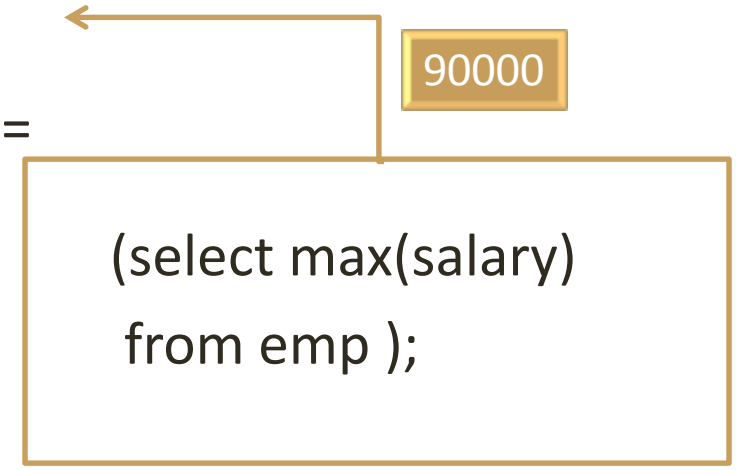
```
(select avg(salary)
 from emp ) ;
```

Output of Above Query

Emp-id	Ename	City	Post	Salary
1	John	Nashik	Clerk	5000
2	Seema	Aurangabad	Developer	20000
5	Samata	Nashik	Tester	35000
6	Ankita	Chandwad	Developer	30000
9	Nitin	Nagpur	Clerk	8000

- **Display the information of employees having maximum salary in company.**

```
Select *
from emp
 where salary  =
```



Output of Above Query

Emp-id	Ename	City	Post	Salary
8	Deepa	Mumbai	CEO	90000

MySQL Subqueries -Multiple rows with ALL, ANY, IN operator

[> ALL] More than the highest value returned by the subquery

[< ALL] Less than the lowest value returned by the subquery

[< ANY] Less than the highest value returned by the subquery

[> ANY] More than the lowest value returned by the subquery

[= ANY] Equal to any value returned by the subquery (same as IN)

MySQL Subqueries- Multiple Row Examples on Emp Table

Emp-id	Ename	City	Post	Salary	deptno
1	John	Nashik	Clerk	5000	10
2	Seema	Aurangabad	Developer	20000	20
3	Amita	Nagar	Manager	70000	20
4	Rakesh	Pune	Analyst	8000	10
5	Samata	Nashik	Tester	20000	10
6	Ankita	Chandwad	Developer	30000	30
7	Bhavika	Pune	Team-LR	50000	30
8	Deepa	Mumbai	CEO	90000	10
9	Nitin	Nagpur	Clerk	8000	30
10	Pooja	Pune	Analyst	45000	20

IN Example- Display the employee name ,salary and department no of those employees whose salary is the minimum salary of that department.

```
SELECT Ename, salary, deptno  FROM EMP
WHERE salary IN (5000,20000,8000)
    ( SELECT MIN(salary)
    FROM emp
    GROUP BY deptno )
```

Ename	Salary	deptno
John	5000	10
Seema	20000	20
Rakesh	8000	10
Samata	20000	10
Nitin	8000	30

Output of Above Query

>All Example- Display the employee name, salary and department no of those employees whose salary is higher than all developers salary.

```
SELECT Ename, salary, deptno  FROM EMP
WHERE salary > All (20000,30000)
                   ( SELECT salary
                   FROM emp
                   Where post='Developer')
```

Ename	Salary	deptno
Amita	70000	20
Bhavika	50000	30
Deepa	90000	10
Pooja	45000	20

Output of Above Query

<All Example- Display the employee name, salary and department no of those employees whose salary is lower than all developers salary.

```
SELECT Ename, salary, deptno  FROM EMP
WHERE salary <All (20000,30000)
    ( SELECT salary
    FROM emp
    Where post='Developer')
```

Ename	Salary	deptno
John	5000	10
Rakesh	8000	10
Nitin	8000	30

Output of Above Query

>Any Example- Display the employee name, salary and department no of those employees whose salary is higher than salary of any developers salary.

```
SELECT Ename, salary, deptno  FROM EMP
WHERE salary >ANY (20000,30000)
    ( SELECT salary
    FROM emp
    Where post='Developer')
```

Ename	Salary	deptno
Amita	70000	20
Ankita	30000	30
Bhavika	50000	30
Deepa	90000	10

Output of Above Query

<Any Example- Display the employee name, salary and department no of those employees whose salary is less than salary of developers salary.

```
SELECT Ename, salary, deptno  FROM EMP
WHERE salary <ANY
```

(20000,**30000**)

```
( SELECT salary
FROM emp
Where post='Developer')
```

Ename	Salary	deptno
John	5000	10
Seema	20000	20
Rakesh	8000	10
Samata	20000	10
Nitin	8000	30

Output of Above Query

Assignment

Cust_Info

C_Id	Cname	City
1	John	Nashik
2	Seema	Aurangabad
3	Amita	Nagar
4	Rakesh	Pune
5	Samata	Nashik
6	Ankita	Chandwad
7	Bhavika	Pune
8	Deepa	Mumbai
9	Nitin	Nagpur
10	Pooja	Pune

Acc_Info

C_Id	Acc_Type	Amount
1	Current	5000
2	Saving	20000
3	Saving	70000
4	Saving	50000
6	Current	35000
7	Loan	30000
8	Saving	50000
9	Saving	90000
10	Loan	8000
11	Current	45000

Assignment

- Show the cname, Acc_Type, amount information of customer who is having an saving account.
- Display the data using Natural, left and right join.
- Display the information of customers living in the same city as of ‘pooja’.
- Display the information of account, having less amount than average amount throughout the bank.
- Display the C_id having maximum amount in account.
- Display the amount and acc_type of those customers whose amount is the minimum amount of that Acc_type.
- Display the amount of those accounts whose amount is higher than amount of any saving account amount.

References

- https://www.javatpoint.com/mysql-join
- https://dev.mysql.com/doc/refman/5.7/en/join.html
- http://www.w3resource.com/mysql/advance-query-in-mysql/mysql-joins.php
- http://www.baskent.edu.tr/~tkaracay/etudio/ders/dbase/sql/pdfSQL/Subqueries.pdf
- https://www.tutorialspoint.com/sql_certificate/subqueries_to_solve_queries.htm