MySQL JOINS

Joining Tables

Join – It is a query that combines rows from two or more tables based on a condition.

Joining – The function of combining data from multiple tables is called joining.

Ways of joining table:-
- Unrestricted join
- Restricted join

(i) UNRESTRICTED JOIN

Here, two tables are joined without any binding condition. The result produced is called the cartesian product.

A cartisan product is formed when:
- A join condition is omitted.
- A join condition is invalid.
- All rows in the first table are joined to all rows in the second table.

Syntax:

```
Select * from n1 , n2 ;
```

eg:- `Select * from t1 ;`

ID	NAME
1	A
2	B
3	C

Select * from t2;

ID	DEPT
2	HR
3	Security
5	Sales

Select * from t1 , t2 ;

ID	NAME	ID	DEPT
1	A	2	HR
1	A	3	Security
1	A	5	Sales
2	B	2	HR
2	B	3	security
2	B	5	sales
3	C	2	HR
3	C	3	Security
3	C	5	sales.

(ii) RESTRICTED JOIN.

When two tables are joined by adding a condition (WHERE clause), it will no longer be the cartesian product. It will be called restricted join or join.

Select * from t_1;

ID	NAME
1	A
2	B
3	C

Select * from t_2;

ID	DEPT
2	HR
3	Security
5	Sales

Select t_1.ID, NAME, DEPT from t_1, t_2 where t_1.ID = t_2.ID;

ID	NAME	DEPT
2	B	HR
3	C	Security.

Qualified names.

Qualified names are used to avoid ambiguity in case the columns have similar names.

<table name>.<Column name>

TABLE ALIASES

A table alias is a temporary lable given along with table name in FROM clause.

Advantages :-

- It allows us to combine two or more columns.
- It is also useful when the column names are big or not readable.

Syntax:

Select <alias name>.<column name> from <table name> <alias> where <condition>;

Eg:- Select e.ename, d.dname, d.deptno from emp_details e, dept d where e.dept no = d.deptno;

ename	dname	deptno
AMIR	ACCOUNTING	10
MITA	ACCOUNTING	10
SMITH	RESEARCH	20
MAHADEVAN	RESEARCH	20
ANOOP	RESEARCH	20
ANYA	SALES	30
SETH	SALES	30

eg:- Select m.name, n.dept from t1 m, t2 n where m.id = n.id;

NAME	DEPT
BINA	SALES
PAYAL	ACCOUNTING
JATIN	SALES

ADDITIONAL SEARCH CONDITIONS IN JOINS

Ques. Display details like department number, department name, employee number, employee name, job and salary. And order the rows by employee number with department number. These details should be only for employees earning atleast ₹ 1500 and of SALES department.

```
Select e.deptno, dname, empno, ename, job,
sal from empl e, dept d where e.deptno =
d.deptno and dname = 'SALES' and sal >= 1500
order by e.deptno, empno;
```

deptno	dname	empno	ename	job	sal
30	SALES	8499	ANYA	SALESMAN	1600.00
30	SALES	8698	BINA	MANAGER	2850.00
30	SALES	8844	KULDEEP	SALESMAN	1500.00

JOINING MORE THAN TWO TABLES.

Sometimes we need to extract information from more than two tables. In this case, you need to join more than two tables. This can be achieved by specifying all the required table names in FROM clause and by providing all join conditions using AND operator.

Example:

Select ENAME, DNAME, JOB, SAL, GRADE FROM EMPL E, DEPT D, SALARYGRADE S where E.DEPTNO = D.DEPTNO and SAL between LOSAL and HISAL ORDER BY E.DEPTNO, EMPNO;

ename	dname	job	sal	grade
AMIR	ACCOUNTING	PRESIDENT	5000	5
SHIAVNSH	ACCOUNTING	MANAGER	2450	4
MITA	ACCOUNTING	MANAGER	1300	2
SMITH	RESEARCH	CLERK	800	1
MAHADEVAN	RESEARCH	CLERK	2985	4
ANOOP	RESEARCH	MANAGER	1100	1
SCOTT	RESEARCH	CLERK	3000	4
FAKIR	RESEARCH	ANALYST	3000	4
ANYA	SALES	ANALYST	1600	3
SETH	SALES	SALESMAN	1250	2
MOMIN	SALES	SALESMAN	1250	2
BINA	SALES	MANAGER	2850	4
KULDEEP	SALES	SALESMAN	1500	3
JATIN	SALES	CLERK	950	1

EQUI JOIN

The join, in which columns are compared for equality, is called Equi-join.

Example:-

```
Select * from empl, dept where empl.deptno
= dept.deptno;
```

ename	Sal	deptno	deptno	dname
SMITH	800	20	20	RESEARCH
ANYA	1600	30	30	SALES
SETH	1250	30	30	SALES
BINA	2850	30	30	SALES

NON-EQUI JOIN

A non-equi join is a query that specified same relationship other than equality between the columns.

Example:

```
Select ename, job, sal, grade from empl E,
salarygrade S where sal between Losal and Hisal
and job = 'ANALYST';
```

ename	job	sal	grade
SCOTT	ANALYST	3000	4
FAKIR	ANALYST	3000	4

NATURAL JOIN

A natural join in which only one of the identical columns (coming from joined tables) exists, is called natural join.

Select empl.*, dname, loc from empl, dept where empl.deptno = dept.deptno;

ename	sal	deptno	dname	loc
SMITH	800	20	RESEARCH	CHENNAI
ANYA	1600	30	SALES	KOLKATA
SETH	1250	30	SALES	KOLKATA
BINA	2850	30	SALES	NEW DELHI
AMIR	5000	10	ACCOUNTING	NEW DELHI

JOINING TABLES USING JOIN CLAUSES OF SQL SELECT

CROSS JOIN

Cross join (or cartesian product) is a very basic type of join that simply matches each row from one table to every row from another table.

Query using CROSS JOIN:—

Select * from empl CROSS JOIN dept;

NATURAL JOIN

Natural join clause removes one of the duplicate columns from the output.

Syntax:-

```
Select * from <table> NATURAL JOIN <table 2>;
```

Example:-

```
Select * from empl NATURAL JOIN dept;
```

Note ⇒ With natural join clause, join condition does not need to be specified.

FILTERING OUTPUT

Filtration of output can be done with the help of WHERE clause.

LEFT JOIN

When using LEFT JOIN all rows from the first table will be returned whether there are matches in the second table or not. For unmatched rows of first table, NULL is shown in columns of second table.

Syntax:-

```
Select <select-list> from <table 1> LEFT JOIN
<table 2> ON <joining-condition>;
```

RIGHT JOIN

Right join works just like left join but the table order is reversed. All rows from the second table are going to be returned whether or not there are matches in the first table.

Syntax:

Select <select - table> from <table 1> RIGHT JOIN <table 2> ON <joining - condition>;

PERFORMING SET OPERATIONS ON RELATIONS

3 set operators in SQL :-

(i) Union
(ii) Intersect
(iii) Minus

1) UNION

It allows manipulation of results returned by two or more queries by combining the results of each query into a single result set.

Select statement UNION select statement ;

Rules of using UNION operator.

i. Both the SELECT statement must be UNION compatible i.e., the select - lists (column - lists) of SELECT statement must have same number of columns having similar data types and order.

For instance, the following union query is invalid:

```
Select ecode, ename FROM branch 1 UNION
SELECT ename, ecode FROM branch 2;
```

It is because the order of columns is different. Also, the following union is invalid because the SELECT statements do not contain compatible select-lists.

```
SELECT ecode, ename FROM branch 1 UNION
SELECT ecode, ename FROM branch 2;
```

ii. The order by clause can occur only at the end of the UNION statement. It can't be used within the individual queries that make the UNION statement.

iii. The GROUP BY and HAVING clauses are allowed only within individual queries. These clauses cannot be used to affect the final result set.

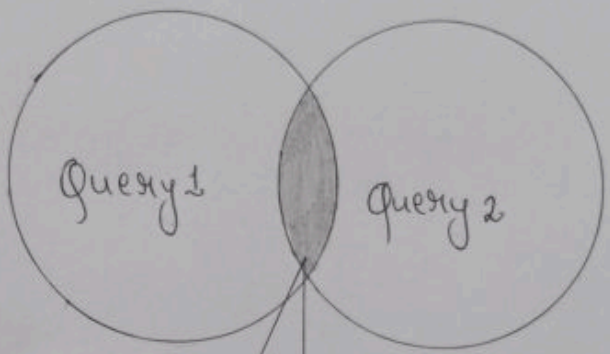
UNION ALL

UNION ALL returns combined all (including duplicate) enteries of two tables.

Syntax:

```
SELECT <col_name> from <table 1>
UNION ALL SELECT <col_name> from <table 2>;
```

2. INTERSECT Operator



INTERSECT operator returns common rows from both left & right tables.

Syntax:

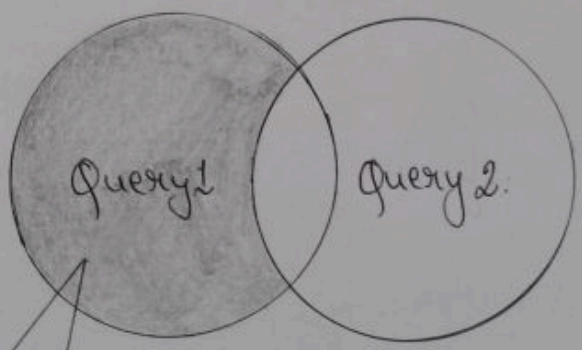
<SELECT query 1> INTERSECT <SELECT query 2>;

Example:

```
SELECT Distinct V.Name from Vendor V INNER JOIN
customer C ON V.Name = C.Name ;
```

NOTE ⇒ MySQL supports no INTERSECT thus the second way of doing INTERSECT operation is INNER JOIN.

3. MINUS operator.



MINUS operation returns the rows that are in the first table not in the second table.

Syntax:

SELECT * from table1 MINUS SELECT * from table 2;