

# **SET OPERATORS IN ORACLE**

## SET OPERATORS IN ORACLE

# **SET Operators:**

- These Operators Are Used to Combine Information of Similar DATA Type From One or More Than One Table.
- DATA Type of The Corresponding Columns in All The SELECT Statements Should Be Same.
- The Different Types of SET Operators Are
  - UNION Operator.
  - UNION ALL Operator.
  - INTERSECT Operator.
  - MINUS Operator.
- SET Operators Can Combine Two or More Queries Into One Result.
- The Result of Each SELECT Statement Can Be Treated As a Set, and SQL Set Operations Can Be Applied On Those Sets To Arrive At a Final Result.
- SQL Statements Containing SET Operations Are Referred To As a Compound Queries, SELECT Statement in a Compound Query is Referred To As a Compound Query.
- Set Operations Are Often Called Vertical joins, As The Result Combines Data From Two or More SELECT Based on Columns Instead of Rows.

## **The Generic Syntax:**

<Compound Query> { UNION | UNION ALL | MINUS | INTERSECT} <Component query>;

### **UNION:**

Combines The Results of Two SELECT Statements Into One Result Set Including The Duplicates, And Then Eliminates Any Duplicate Rows From That Resullt Set.

# **UNION ALL:**

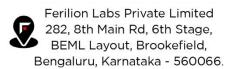
Combines The Result of Two SELECT Statements Into One Result Set Including The Duplicates.

## INTERSECT:

Returns Only Those Rows That Are Returned By Each of Two SELECT Statements.













## MINUS:

Takes The Result Set of One SELECT Statement, And Removes Those Rows That Are Also Returned By a Second SELECT Statement.

#### **Point of Concentration:**

- The Queries Are All Executed Independently But Their Output is Merged.
- Only Final Query Ends With a Semicolon.

# **Rules And Restrictions:**

- The Result Sets of Both The Queries Must Have The Same Number of Columns.
- The Data Type of Each Column in The Second Result Must Match The Data Type of Its Corresponding Column in The First Result Set.
- The Two SELECT Statements May Not Contain An ORDER BY Clause, The Final Result of The Entire SET Operation Can Be Ordered.
- The Columns Used For Ordering Must Be Defined Through The Column Number.

### **Illustrations:**

```
SQL> SELECT Empno, Ename, FROM Emp WHERE Deptno = 10
                                                           UNTON
      SELECT Empno, Ename FROM Emp WHERE Deptno = 30 ORDER BY 1;
SQL> SELECT Empno, Ename, Job FROM Emp WHERE Deptno = ( SELECT
Deptno FROM Dept WHERE Dname = 'SALES' )
                                                           UNION
SELECT Empno, Ename, Job FROM Emp WHERE Deptno = ( SELECT Deptno
FROM Dept WHERE Dname = 'ACCOUNTING' ) ORDER BY 1;
SQL> SELECT Empno, Ename FROM Emp WHERE Deptno=10
                              UNION ALL
         SELECT Empno, Ename FROM Emp WHERE Deptno=30 ORDER BY
SQL> SELECT Empno, Ename FROM Emp WHERE Deptno=10
                              INTERSECT
         SELECT Empno, Ename FROM Emp WHERE Deptno=30 ORDER BY
                                                                1;
SQL> SELECT Empno, Ename FROM Emp WHERE Deptno=10
                              MINUS
         SELECT Empno, Ename FROM Emp WHERE Deptno=30 ORDER BY 1;
SQL> SELECT Job FROM Emp WHERE Deptno=20
                              UNION
         SELECT Job FROM Emp WHERE Deptno=30;
SQL> SELECT Job FROM Emp WHERE Deptno=20
                              UNION ALL
         SELECT Job FROM Emp WHERE Deptno=30;
SQL> SELECT Job FROM Emp WHERE Deptno=20
                               INTERSECT
```











# Dream. Create. Achieve.

SELECT Job FROM Emp WHERE Deptno=30; SQL> SELECT Job FROM Emp WHERE Deptno=20 MINUS SELECT Job FROM Emp WHERE Deptno=30; SQL> SELECT ROWNUM, Ename FROM Emp WHERE ROWNUM<7 MINUS SELECT ROWNUM, Ename FROM Emp WHERE ROWNUM<6;

