**Types of Query**:

• Root Query

• Parent Query (or) Outer Query (or) Main Query

• Child Query (or) Inner Query (or) sub Query

**Root Query**:-

• The query which is not depend on any other query for its conditions value.

(or)

Independent query.

**Example**:-

Sql>Select ename,sal from emp Where Deptno=10;

**Parent Ouery**:-

• The query which depend on any other query for its condition value.

**Sub Query** :-

* The query which provides, conditional values to its parent query.
* A sub Query in the WHERE clause of a SELECT statement is called as NESTED SUBQUERY.
* A sub Query in the FROM clause of a SELECT statement is called as INLINE VIEW.
* A sub Query can be part of a column in the SELECT list.
* Sub Query can contain another sub query.
* Oracle imposes no limit on the number of sub query levels in the FROM clause of the Top-Level Query.
* Within the WHERE clause up to 255 sub queries can be nested.
* To make the statement easier for readability, qualify the columns in a Sub Query with the table name or table alias.

**Note**: parent query and sub query both are combined.

**SUB QUERY Principle**:

* Solve a problem by combining the two queries, placing one query inside the other Query.
* The inner query or the sub query returns a value that is used by the outer query upon the main query.

**Sub Query usage**:

* A Sub Query is a SELECT statement that is embedded in a clause of another SELECT statement.
* We can build powerful statements out of simple ones by using Sub Queries.
* They can be very useful when you need to select rows from a table with a condition that depends on the data in the table itself.

**Syntax**:

Sql>SELECT select\_list

FROM table

WHERE Column name/expr operator (

SELECT select\_list

FROM table

);

**Comparison conditions fall into two classes**:

* Single -row operators (>, =, >=, <, <>, <=)
* Multiple-row operators (IN, ANY, ALL).

**Types of sub query:**

**Single row sub Query**:

• These queries return only one row from the Inner SELECT statement.

**Multiple row sub query**:

* These queries return more than one row from the Inner SELECT statement.

**Guidelines for Using Subqueries**:

* Enclose Sub Queries in parentheses.
* Place Sub Queries on the right side of the comparison condition.
* The ORDER BY clause in the Sub Query is not needed unless you are performing top-analysis.
* Only one ORDER BY clause can be used for a SELECT statement.
* Two classes of comparison conditions are used in Sub Queries:
* Single-row operators
* Multiple-row operators.

**Simple Sub Query**:

• Sub Query processed first and processed completely.

**Co-related Sub Query**:

* In the Co-Related Sub Query a parent query will be executed first and based on the output of outer query the inner query executes.
* If Parent Query Returns N rows inner query executed for N times.

**Simple Sub Query with Single Row**:

Sql> Select Empno, Ename, Sal, Deptno

From Emp

Where Sal<(Select Sal

From Emp

Where Empno=7566);

Sql>Select Empno,Ename,Job

From EMP

Where Job = (Select Job

From Emp Where upper(Ename) = UPPER('ford')) ;

Sql> Select Empno,Ename, Hiredate,Sal,Job

From Emp

Where Hiredate>(Select Hiredate

From Emp

Where Ename = 'MILLER'

) Order by Sal;

Sql> Select Empno,Ename,Sal,Job From Emp

Where Deptno=(Select Deptno

From Dept Where Dname = 'SALES');

Sql> Select Empno,Ename,Sal,Job From Emp

Where Deptno=(Select Deptno

From Dept Where Loc='NEW YORK'

) And Job='CLERK';

Sql> Select Empno,Ename,Sal,Job

From Emp

Where Deptno=(Select Deptno From Dept Where Dname = 'SALES');

Sql> Select Empno,Ename,Sal,Comm,Sal+NVL(Comm,0) From Emp

Where Deptno = (Select Deptno From Dept Where Loc='CHICAGO');

Sql> Select \* From Emp

Where Deptno = (Select Deptno

From Emp Where Ename='FORD') AND

Job IN(Select job

From Emp Where Deptno=(Select Deptno

From Dept Where Dname = 'SALES'));

**Appling GRQUP function in SUB QUERIES**:

* The data from the main query can be displayed by using a Group Function in a Sub Query.
* As a Group Function returns a single row, the query passes the success state. The inner sub query should not have a GROUP by Clause in this scenario.

Examples:

Sql>SELECT ename,job,sal

FROM emp

WHERE sal = (SELECT Max(sal)

FROM emp

);

Sql> SELECT ename,job,sal

FROM emp

WHERE sal = (SELECT Min(sal) FROM emp);

Sql>SELECT ename,job,sal

FROM emp

WHERE sal>(SELECT Avg(sal) FROM emp);

Sql>SELECT ename,job,sal FROM emp

WHERE sal<(SELECT Variance (sal) FROM emp);

**Appling HAVING Clause With SUB QUERIES**:

* A Sub Query can also be applied in HAVING Clause.
* The Oracle server execute the sub query and the result are returned into the HAVING Clause of the Main Query.

**Example**:-

Sq|>SELECT deptno,Min(sal)

FROM emp

GROUP BY deptno

HAVING Min(sal)>(SELECT Min(sal)

FROM emp

WHERE deptno=20

);

Sql>SELECT job,Avg(sal)

FROM emp

GROUP BY job

HAVING Avg(sal) = (SELECT Min(Avg(sal))

FROM emp

GROUP BY job);

Sql> SELECT job,Avg(sal)

FROM emp

GROUP BY job

HAVING Avg(sal)<(SELECT Max(Avg(sal))

FROM emp

GROUP BY job);

**Sub Query Returns more than one row**:

* Subqueries that return more than one row are called **Multiple-row Sub Queries**.
* We use a multiple-row operator, instead of a Single-Row operator, with a Multiple -row Sub Query.
* The multiple-row operator expects one or more values.
* Use multiple-row comparison operators
* IN 🡪 Equal to any member in the list
* ANY/SOME 🡪 Compare value to each value returned by the Sub Query.
* ALL 🡪 Compare value to every value returned by the Sub Query

Sql>SELECT ename, sal, deptno FROM emp

WHERE sal IN(SELECT MAX(sal)

FROM emp

GROUP BY deptno);

Sql>SELECT ename,sal,job,deptno

FROM emp

WHERE sal IN(SELECT MAX(sal)

FROM emp

GROUP BY job);

Sql>SELECT ename,sal,deptno

FROM emp

WHERE sal IN(SELECT MIN(sal)

FROM emp

GROUP BY deptno);

**ANY Operator**:

Sql> SELECT ename,sal,deptno

FROM emp

WHERE sal <SOME(1250,1500,1600);

Sql>SELECT ename,sal,deptno

FROM emp

WHERE sal <SOME(SELECT sal FROM emp

Where Job='SALESMAN');

Sql>SELECT ename, sal,deptno, Job

FROM emp

WHERE sal <ANY(SELECT sal

FROM emp Where

Deptno in(20,30) And

Job<>'MANAGER');

Sql>Select Empno,Ename,Sal,Deptno

From emp

Where Sal>ANY(Select Sal

From Emp

where deptno=10 );

**Note:<ANY/SOME> means less than the Maximum value in the list.**

Sql> SELECT ename,sal,deptno

FROM emp

WHERE sal >SOME(1250,1500,1600);

Sql>SELECT ename,sal,deptno

FROM emp

WHERE sal >ANY(SELECT sal

FROM emp

Where Job='SALESMAN'

);

**Note:<ANY/SOME> means More than the Minimum value in the list.**

Sql> SELECT ename,Job,Deptno

FROM emp

WHERE sal =ANY(SELECT sal

FROM emp

Where Job='MANAGER'

);

**Note: ANY in this case it is equivalent to In operator.**

**All Operator**:

Sql> Select Ename,Job,Sal,Deptno

From Emp

Where Sal>ALL(Select Sal

From Emp

Where Deptno=30);

Sql> Select Ename,Job,Sal,Deptno From Emp

Where Sal>ALL(Select AVG(Sal)

From Emp

Group by Deptno);

**Note:>AlI 🡪it means more than the maximum in the list.**

Sql> Select Ename,Job,Sal,Deptno

From Emp

Where Sal<ALL(Select AVG(Sal)

From Emp

Group by Deptno);

**Note:>ALL 🡪 it means less than the maximum In the list.**

**Sub Query Returning multiple Columns:**

* In sub Queries multiple columns can be compared in the WHERE, clause, writing a compound WHERE clause using logical operator.

**Syntax**:

Sql>Select Column 1, Column2,....

From Table Name

Where

(Column a, Column b,…..) IN(Select Column a, Column B,……

From TableName

Where Condition

);

* The column comparisons in a multiple column sub Query can be.
  + Pair wise comparison.
  + Non pair wise comparison.
* Pair wise comparisons Each Candidate row in the SELECT statement,Must have both the same values associated with each column in the group.
* The Non pair wise comparison, the candidate row must match the multiple condition in the WHERE clause but the values are compared individually.

**Pair wise Comparison or compound WHERE clause based SubQuery**

Sql> Select Ename,Sal,Deptno

From Emp

Where (Deptno,Sal) IN(Select Deptno, Max(Sal)

From Emp

Group by Deptno)

And Deptno<>10;

**Non-Pair wise comparison or compound WHERE clause based Sub Query**

Sql>Select Ename,Sal,Deptno From Emp

Where Deptno IN(Select Deptno

From Emp

Group by Deptno) And

Sal IN(Select Max(Sal)

From Emp

Group by Deptno)

And Deptno<>10;

**Null Values in a Sub Query**:

* One of the values returned by the inner query is a null value, and hence the entire query returns no rows. The reason is that all conditions that compare a null value result in a null.
* So whenever NULL values are likely to be part of the results set of a Sub Query, do not use the NOT IN operator. The NOT IN operator is equivalent to <> ALL operator.

Sql> Select E.Ename

From Emp E

Where E.Empno IN(Select M.Mgr

From Emp M

);

**Appling Sub Query in From Clause**:

* A Sub Query in the from clause is equivalent to a view.
* The Sub Query in the from clause defines a Data source for that particular SELECT statement.

Sql>Select d.Deptno,d.Dname,v.cnt

From Dept d, (Select Deptno,Count(\*) cnt

From Emp Group by

Deptno) v Where d.Deptno=v.Deptno And cnt>3;

Sql>Select E.Ename,E.Sal,E.deptno,S.Salavg

From Emp E,(Select Deptno,AVG(Sal) Salavg

From Emp Group By Deptno) S Where E.Deptno=S.Deptno AND E.Sal>S.Salavg;

Sql> Select Deptno,SUM(Sal),

SUM(Sal)/Tol\_Sal\*100"Salary%

From Emp,(Select SUM(Sal) Tol\_Sal

From Emp

)

Group by Deptno,Tol\_Sal;

Sql> Select E.EmpCount,D.DeptCount From(Select COUNT(\*) EmpCount From Emp )E, (Select COUNT(\*) DeptCount

From Dept )D;

Sql> Select E.EmpCount,D.DeptCount,S.GradeCnt,

E.Empcount+D.DeptCount+S.GradeCnt TotalRecCnt

From(Select COUNT(\*) EmpCount

From Emp )E, (Select COUNT(\*) DeptCount

From Dept )D,

(Select COUNT(\*) GradeCnt

From Salgrade )S;

Sql>Select A.Deptno "department Number",

(A.NumEmp/B TotalCount)\*100 "%Employees",

(A.SalSum/B.TotalSal)\*100 "%Salary"

From(Select Deptno,

Count(\*) NumEmp,

SUM(Sal) SalSum

From Emp

Group By Deptno) A,

(Select COUNT(\*) TotalCount, SUM(Sal) TotalSal From Emp)B;

**Sub Select Statements**:

* These are SELECT statements declared as part of the SELECT list. Sql>Select Ename,Sal,

(Select SUM(Sal) From Emp

) "organizaton TolSal" From Emp;

Sql> Select Ename,Sal,

(Select MAX(Sal)

From Emp ) "organizaton Maximum Salary", (Select MIN(Sal)

From Emp ) "organizaton Lowest Salary" From Emp;

**Co-related Sub Queries**:

• It is another way of performing Queries upon the data with a

simulation of Joins.

**Syntax**

Sql> SELECT column1, column2, ... FROM table1 T\_Alias1

WHERE column1 operator (SELECT colum1, column2

FROM table2 t\_Alias2

WHERE T\_Alias1.Column Operator T\_Alias2.Column

);

**Steps Performed**

* Parent Query processed first.
* Passes the Qualified column value to the sub query WHERE clause,
* Get a candidate row (fetched by the outer query).
* Execute the inner query using the value of the candidate row.
* Use the values resulting from the inner query to qualify or disqualify the candidate.
* Repeat until no candidate row remains.

**Exists**:

* Returns true if inner query is success otherwise False.

Sql> Select Empno, Ename, Sal, Deptno

From Emp

Where Deptno = 10 AND

EXISTS

(Select COUNT(\*)

From Emp

Where Deptno=10

AND Job ='ANALYST' Group By Job

Having COUNT(\*)>5);

**Note**: In the case Group by clause become as dummy clause so without using

group by we can use having.

Sql> Select Empno,Ename,Sal,Deptno

From Emp

Where Deptno=30 AND

EXISTS

(Select COUNT(\*)

From Emp Where Deptno = 10

AND Job = 'SALESMAN'

Having COUNT(\*)>3);

Sql>Select Deptno,Dname

From Dept D

Where EXISTS(Select \*

From Emp E

Where D.Deptno=E.Deptno

);

Sql>Select Deptno,Dname

From Dept D

Where NOT EXISTS(Select \*

From Emp E

Where D.Deptno=E.Deptno);

Sql> Select P.Ename

From Emp P

Where EXISTS(Select \*

From Emp C

Where C.Empno=P.MGR

);

Sql> Select P.Ename

From Emp p

Where NOT EXISTS(Select \*

From Emp C

Where C.Empno = P.MGR

);

Sql> Select P.Ename From Emp P

Where EXISTS(Select \* From Emp C

Where C.MGR=P.Empno);

Sql> Select P.Ename From Emp P Where NOT EXISTS(Select \*

From Emp C Where C.MGR=P.Empno);

Sql>Select Empno,Ename,E.Deptno,Sal,MGR From Emp E

Where E.Sal>ANY(Select M.Sal

FROM Emp M Where M.Empno=E.MGR);