SUBQUERY

Definition

- A Query within a Query.
- A subquery is a SELECT statement that is embedded in a clause of another SELECT statement.
- You can place the subquery in a number of SQL clauses:

❑ WHERE clause

❑ HAVING clause

❑ FROM clause

In the syntax:

```
SELECT <COLUMN, ...>
FROM <TABLE>
WHERE EXPRESSION OPERATOR
   ( SELECT <COLUMN, ...>
     FROM <TABLE>
    WHERE <CONDITION>);
```

- Expression Operator could be equality operators or comparison operator such as =, >, <, >=, <=. It can also be a text operator such as "LIKE". The portion in **red** is considered as the "inner query", while the portion in **green** is considered as the "outer query".

EXAMPLE:

SQL> SELECT * FROM employee

WHERE sal > (SELECT sal FROM EMP WHERE name='TOM');

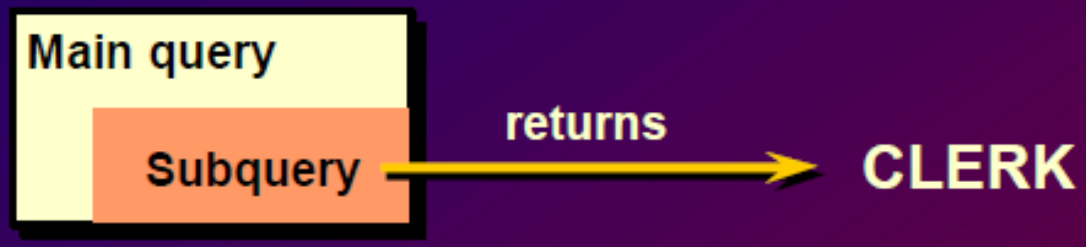
- A subquery is a query within another query. The outer query is called as main query and inner query is called as subquery.
- The subquery is often referred to as a nested SELECT, sub-SELECT, or inner SELECT statement.
- The subquery generally executes first, and its output is used to complete the query condition for the main or outer query.

Guidelines for using subquery

- Enclose subqueries in parentheses.
- Place subqueries on the right side of the comparison operator.
- Do not add an ORDER BY clause to a subquery.
- Use single-row operators with singlerow subqueries.
- Use multiple-row operators with multiple-row subqueries.

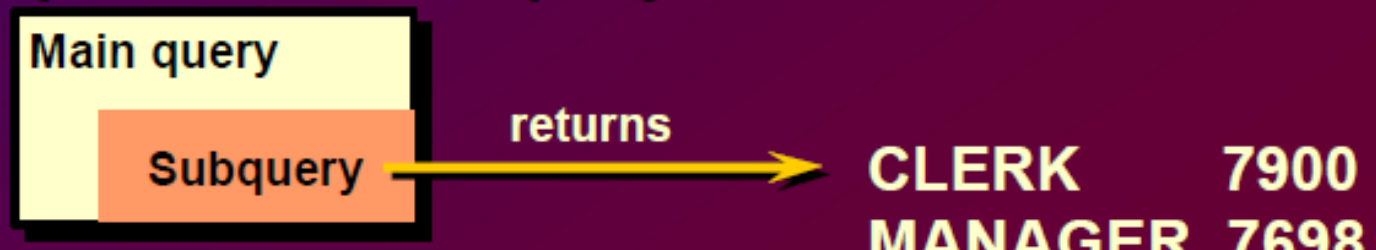
Types of Subqueries

- Single-row subquery



- Multiple-row subquery

- Multiple-column subquery



Types of Subqueries

Subquery	Description
Single-Row Subquery	Returns one row of results that consists of one column to the outer query
Multiple-Row Subquery	Returns more than one row of results to the outer query
Multiple-Column Subquery	Returns more than one column of results to the outer query
Correlated Subquery	References a column in the outer query. Executes the subquery once for every row in the outer query
Uncorrelated Subquery	Executes the subquery first and passes the value to the outer query

- **Single-Row Subqueries**

A *single -row subquery* is one that returns one row from the inner SELECT statement. This type of subquery uses a single -row operator.

The operators that can be used with single-row subqueires are =, >, >=, <, <=, and <>.

Single-Row Subqueries

- Return only one row
- Use single-row comparison operators

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

Example

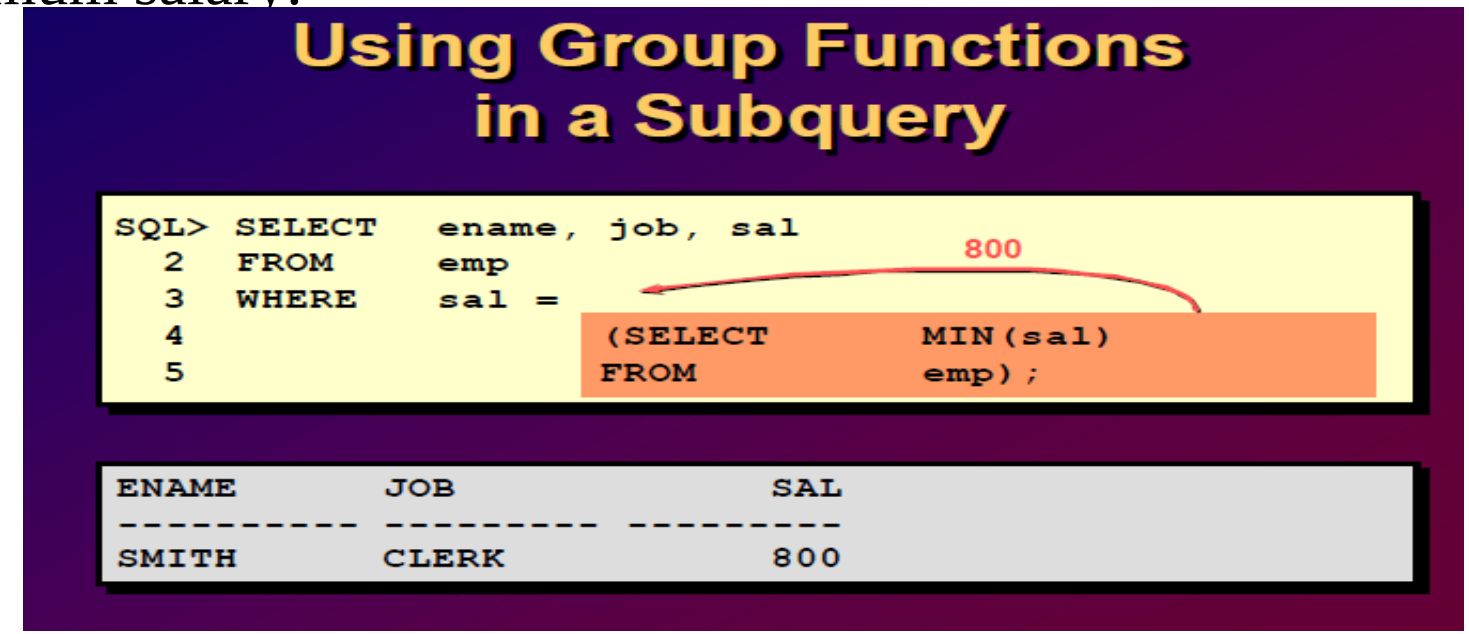
Display the employees whose job title is the same as that of employee 7369.

```
SQL> SELECT    ename, job
  2  FROM      emp
  3  WHERE     job =
  4                (SELECT  job
  5                 FROM    emp
  6                 WHERE   empno = 7369);
```

```
ENAME      JOB
---------- ---------
JAMES      CLERK
SMITH      CLERK
ADAMS      CLERK
MILLER     CLERK
```

- **Single row Subquery using group function**

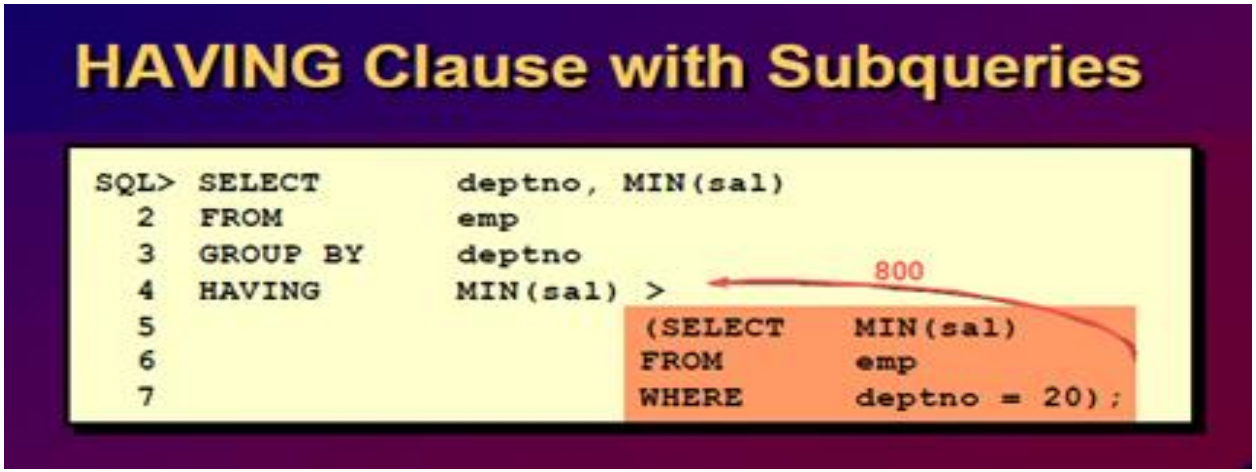
Consider an example of an employee table with employee name, job title, and salary of all employees whose salary is equal to the minimum salary.



The MIN group function returns a single value

▪ Having Clause in subqueries

Not only WHERE clause can be used in subqueries but also HAVING clause.



The SQL statement displays all the departments that have a minimum salary greater than that of department 20.

DEPTNO	MIN(SAL)
10	1300
30	950

SQL Subquery; INSERT Statement

Subquery can be used with INSERT statement to add rows of data from one or more tables to another table.

Lets try to group all the students who study Maths in a table 'maths_group'.

SYNTAX:

```
INSERT INTO <tablename>
            (column1, columns2,….)
                SELECT value1,value2,…
                    FROM … [rest of the query]
```

EXAMPLE:

```
INSERT INTO maths_group(id, name)
SELECT id, first_name || ' ' || last_name
FROM student_details WHERE subject= 'Maths‘;
```

SQL Subquery; Delete Statement

SYNTAX:

```
DELETE FROM <tablename>
                    WHERE (condition)
                              SELECT value1,value2,…
                                        FROM … [rest of the query]
```

EXAMPLE:

```
SQL>
SQL> DELETE FROM Empl1
  2     WHERE salary >
  3       (SELECT AVG(salary)
  4        FROM Empl1);

1 row deleted.

SQL>   select * from Empl1;

ID   FIRST_NAME LAST_NAME  START_DAT END_DATE      SALARY CITY
---- ---------- ---------- --------- --------- ---------- ----------
DESCRIPTION
---------------
01   Jason      Martin     25-JUL-96 25-JUL-06    1234.56 Toronto
Programmer
```

SQL Subquery; Update Statement

SYNTAX:

```
UPDATE  <tablename> SET  (column =(       SELECT value1,value2,…
                                          FROM … [rest of the query]
```

EXAMPLE:

```
SQL>
SQL>   UPDATE Empl1
  2      SET salary =
  3      (SELECT AVG(salary)
  4       FROM Empl1);

2 rows updated.

SQL>
SQL> select * from Empl1
  2  ;

ID   FIRST_NAME LAST_NAME   START_DAT END_DATE       SALARY CITY
---- ---------- ----------- --------- --------- ---------- ----------
DESCRIPTION
---------------
01   Jason      Martin      25-JUL-96 25-JUL-06    1784.67 Toronto
Programmer

05   Robert     Black       15-JAN-84 08-AUG-98    1784.67 Vancouver
Tester
```

Subqueries with EXISTS and with Not Exists

Example:

```
SELECT column1 FROM t1 WHERE EXISTS
                                (SELECT * FROM t2);
```

Example:

```
Select CustomerID, CompanyName
from customers as a
where not exists
(    select * from orders as b
            where a.CustomerID = b.CustomerID
                        and ShipCountry <> 'UK‘ );
```

- **Multiple-Row Subqueries**

Subqueries that return more than one row are called *multiple-row subqueries*. You use a multiple -row operator, instead of a single -row operator, with a multiple -row subquery. The multiple –row operator expects one or more values.

Multiple-Row Subqueries

- Return more than one row
- Use multiple-row comparison operators

Operator	Meaning
IN	Equal to any member in the list
ANY	Compare value to each value returned by the subquery
ALL	Compare value to every value returned by the subquery

```
SQL> SELECT    ename, sal, deptno
  2  FROM      emp
  3  WHERE     sal IN (SELECT    MIN(sal)
  4                    FROM     emp
  5                    GROUP BY deptno);
```

Example

Find the employees who earn the same salary as the minimum salary for departments.

The inner query is executed first, producing a query result containing three rows: 800, 950, 1300. The main query block is then processed and uses the values returned by the inner query to complete its search condition. In fact, the main query would look like the following to the Oracle Server:

```
SQL> SELECT   ename, sal, deptno
  2  FROM     emp
  3  WHERE    sal IN (800, 950, 1300);
```

Operator	Description
>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)

Figure 7-11 Descriptions of ALL and ANY operator combinations

Using ANY Operator in Multiple-Row Subqueries

The ANY operator compares a value to *each* value returned by a subquery.

The example displays employees whose salary is less than any clerk and who are not clerks. The maximum salary that a clerk earns is $1300. The SQL statement displays all the employees who are not clerks but earn less than $1300.

<ANY means less than the maximum.

>ANY means more than the minimum.

=ANY is equivalent to IN.

```
SQL> SELECT   empno, ename, job
  2  FROM     emp
  3  WHERE    sal < ANY
  4                        (SELECT    sal
  5                        FROM       emp
  6                        WHERE      job = 'CLERK')
  7  AND      job <> 'CLERK';
```

1300
1100
800
950

```
     EMPNO ENAME      JOB
--------- ---------- ---------
      7654 MARTIN     SALESMAN
      7521 WARD       SALESMAN
```

Using ALL Operator in Multiple-Row Subqueries

The ALL operator compares a value to *every* value returned by a subquery. The example displays employees whose salary is greater than the average salaries of all the departments.

The highest average salary of a department is $2916.66, so the query returns those employees whose salary is greater than $2916.66.

>ALL means more than the maximum and

<ALL means less than the minimum.

The NOT operator can be used with IN, ANY, and ALL operators

```
SQL> SELECT   empno, ename, job
  2  FROM     emp
  3  WHERE    sal > ALL
  4                   (SELECT        avg(sal)
  5                   FROM           emp
  6                   GROUP BY       deptno);
```

1566.6667
2175
2916.6667

```
     EMPNO ENAME      JOB
---------- ---------- ---------
      7839 KING       PRESIDENT
      7566 JONES      MANAGER
      7902 FORD       ANALYST
      7788 SCOTT      ANALYST
```

Using IN Operator in Multiple-Row Subqueries

The IN keyword treats each value as a member of a set and tests whether each row in the main query is a member of the set.

```
SELECT * FROM FinancialData
WHERE Code IN ( SELECT Code FROM
FinancialCodes  WHERE type = 'revenue' );
```

DIFFERENCE BETWEEN SINGLE ROW AND MULTIPLE ROW SUBQUERY

SINGLE ROW SUBQUERY	MULTIPLE ROW SUBQUERY
Subqueries that can return only one or zero rows to the outer statement are called **single-row subqueries**.	Subqueries that can return more than one row (but only one column) to the outer statement are called **multiple-row subqueries**.
Single-row subqueries are subqueries used with a comparison operator in a WHERE, or HAVING clause.	Multiple-row subqueries are subqueries used with an IN, ANY, or ALL clause.

Multiple-Column Subquery

- In multiple-column subqueries, rows in the subquery results are evaluated in the main query in pair-wise comparison. That is, column-to-column comparison and row-to-row comparison.
- For example, the following statement lists all items whose quantity and product id match to an item of order id 200.

```
SELECT order_id,  product_id, quantity
  FROM item WHERE (product_id, quantity) IN
( SELECT  product_id, quantity FROM item WHERE order_it =
200)AND order_id = 200;
```

Nested Subquery

Nested Subqueries

A subquery is nested when you are having a subquery in the where or having clause of another subquery.

Get the result of all the students who are enrolled in the same course as the student with ROLLNO 12.

```
Select *
From result
where rollno in (select rollno
                 from student
                 where courseid = (select courseid
                                   from student
                                   where rollno = 12));
```

Correlated subquery

Correlated Subquery

A **Correlated Subquery** is one that is executed after the outer query is executed. So correlated subqueries take an approach opposite to that of normal subqueries. The correlated subquery execution is as follows:

-The outer query receives a row.
-For each candidate row of the outer query, the subquery (the correlated subquery) is executed once.
-The results of the correlated subquery are used to determine whether the candidate row should be part of the result set.
-The process is repeated for all rows.

Correlated Subqueries differ from the normal subqueries in that the nested SELECT statement referes back to the table in the first SELECT statement.

To find out the names of all the students who appeared in more than three papers of their opted course, the SQL will be

```
Select name
from student A
Where 3 < (select count (*)
           from result b
           where b.rollno = a.rollno);
```

DIFFERENCE BETWEEN Non-Correlated AND Correlated SUBQUERY

Correlated SUBQUERY	Non-Correlated SUBQUERY
Correlated subquery is one whose value depends upon some variable that recieves its value in some qouter query.	Non correlated subquery first executes the inner most query .
Correlated query is evaluted in top-down manner.	Non corelated subquery is evaluated in a bottom-up-manner.
Correlated subquery improves the performance of SQL.	Non –correlated subquery needs to be evaluated repeatedly hence,it does not improves the efficiency or performance of SQL.

DIFFERENCE BETWEEN SUBQUERYAND Correlated SUBQUERY

SUBQUERY	Correlated SUBQUERY
The inner query is executed only once The inner query will get executed first and the output of the inner query used by the outer query.T	The outer query will get executed first and for every row of outer query, inner query will get executed. So the inner query will get executed as many times as no.of rows in result of the outer query. The outer query output can use the inner query output for comparison.
The inner query is not dependent on outer query.	The inner query and outer query dependent on each other
Ex: SELECT cust_name, dept_no FROM Customer WHERE cust_name IN (SELECT cust_name FROM Customer);	Ex: SELECT cust_name,dept_id FROM Cust WHERE cust_name in (SELECT cust_name FROM dept WHERE cust.dept_id=dept.dept_id);

THANK U