Agenda

- SubQuery

SubQuery

- A query inside another query is called as subquery
- we have two types of subquery
    - 1. single row subquery
        - In this the subquery/inner query returns single row
    - 2. multi rows subquery
        - In this the subquery return multiple rows
        - the output of inner query can be compared using ANY, ALL or IN operator.

Singlerow Subquery

```
--- display emp with highest salary
SELECT * FROM emp ORDER BY sal DESC LIMIT 1;

SELECT * FROM emp WHERE sal = MAX(sal);
-- cannot use group functions in where clause

SET @maxsal = (SELECT MAX(sal) FROM emp);
SELECT * FROM emp WHERE sal = @maxsal;

SELECT * FROM emp WHERE sal = (SELECT MAX(sal) FROM emp);
```

MultiRow SubQuery

```
-- find all emps having sal more than all salesman.
SELECT * FROM emp WHERE sal > ALL(SELECT sal FROM emp WHERE job =
"SALESMAN");
```

ANY vs IN operator

- ANY can be used in sub-queries only, while IN can be used with/without sub-queries.
- ANY can be used for comparision (<, >, <=, >=, =, or !=), while IN can be used only for equality comparision (=).
- Both operators are logically similar to OR operator.

ANY vs ALL operator

- Both can be used for comparision (<, >, <=, >=, =, or !=).
- Both are usable only with sub-queries.
- ANY is similar to logical OR, while ALL is similar to logical AND.

Corelated SubQuery

- By default inner query is executed for each row of the outer query.
- If no optimization settings are enabled, sub-queries are slower than joins.

```
-- display the depts in which emps exists
SELECT * FROM dept WHERE deptno IN (SELECT deptno FROM emp);
-- 10, ACC --> SELECT deptno FROM emp - 14 rows
-- 20, RES --> SELECT deptno FROM emp - 14 rows
-- 30, SAL --> SELECT deptno FROM emp - 14 rows
-- 40, OPS --> SELECT deptno FROM emp - 14 rows
```

- The sub-query execution can be speed-up if inner queries return/process less number rows.
- This is typically done by using WHERE clause in inner query.
- The WHERE clause in inner query may depend on current row of the outer query. This kind of query is called as "co-related sub-query".

```
SELECT * FROM dept WHERE deptno IN (SELECT DISTINCT deptno FROM emp);
-- 10, ACC --> SELECT DISTINCT deptno FROM emp - 3 rows
-- 20, RES --> SELECT DISTINCT deptno FROM emp - 3 rows
-- 30, SAL --> SELECT DISTINCT deptno FROM emp - 3 rows
-- 40, OPS --> SELECT DISTINCT deptno FROM emp - 3 rows

SELECT * FROM dept d WHERE d.deptno IN (SELECT e.deptno FROM emp e WHERE
e.deptno = d.deptno);
-- 10, ACC --> SELECT deptno FROM emp WHERE deptno=10 - 3 rows
-- 20, RES --> SELECT deptno FROM emp WHERE deptno=20 - 5 rows
-- 30, SAL --> SELECT deptno FROM emp WHERE deptno=30 - 6 rows
-- 40, OPS --> SELECT deptno FROM emp WHERE deptno=40 - 0 rows

SELECT * FROM dept d WHERE d.deptno = (SELECT DISTINCT e.deptno FROM emp e
WHERE e.deptno = d.deptno);
-- 10, ACC --> SELECT DISTINCT deptno FROM emp WHERE deptno=10 - 1 rows
-- 20, RES --> SELECT DISTINCT deptno FROM emp WHERE deptno=20 - 1 rows
-- 30, SAL --> SELECT DISTINCT deptno FROM emp WHERE deptno=30 - 1 rows
-- 40, OPS --> SELECT DISTINCT deptno FROM emp WHERE deptno=40 - 0 rows
```

Sub-query in projection

```
-- display number of emps in each dept along with total number of emps.
-- eg -> 10 - 3 - 14
SELECT deptno, COUNT(empno) FROM emp GROUP BY deptno;
```

```
SELECT deptno,COUNT(empno) empcount,(SELECT COUNT(empno) FROM emp) totalemp
FROM emp GROUP BY deptno;

SELECT deptno, CONCAT(COUNT(empno),"/",(SELECT COUNT(empno) FROM emp))
count FROM emp GROUP BY deptno;
```

Subquery in FROM Clause

- Inner-query can be written in FROM clause of SELECT statement.
- The output of the inner query is treated as a table (MUST give table alias) and outer query execute on that table.
- This is called as "Derived table" or "Inline view".

```
-- display empno, ename, sal and category of emp.
-- sal<= 2000 -> POOR ,sal > 2000 -> RICH

SELECT empno, ename, sal, IF(sal<=2000,"POOR","RICH") AS category FROM emp;

-- display no of emps in each category
SELECT category,COUNT(ename) FROM (SELECT empno, ename, sal,
IF(sal<=2000,"POOR","RICH") AS category FROM emp)AS empcategory GROUP BY
category;
```

Sub-query in DML

```
-- INSERT new emp with name 'JOHN' and sal=2000 in dept 'OPERATIONS'.
SELECT deptno FROM dept WHERE dname = 'OPERATIONS';

INSERT INTO emp(ename, sal, deptno) VALUES ('JOHN', 2000, (SELECT deptno
FROM dept
WHERE dname = 'OPERATIONS'));

-- give comm 100 to all emps in OPERATIONS dept.
UPDATE emp SET comm = 100 WHERE deptno =
(SELECT deptno FROM dept WHERE dname = 'OPERATIONS');

-- delete emps from OPERATIONS dept.
DELETE FROM emp WHERE deptno =
(SELECT deptno FROM dept WHERE dname = 'OPERATIONS');
```

- In MySQL, DML cannot be performed on the table from which inner query is selecting.

```
INSERT INTO emp(empno,ename,sal) VALUES(1000, 'JACK', 6000);
-- delete emp with max sal.
```

```
DELETE FROM emp WHERE sal = (SELECT MAX(sal) FROM emp);
```

SQL Performance

- In MySQL, query Performance is measured in terms of query cost. Lower the cost, better is performance.
- The cost of query depends on data in the table(s), MySQL version, server machine config, optimizer settings, etc.

```
SELECT @@optimizer_switch;
EXPLAIN FORMAT = JSON SELECT * FROM dept WHERE deptno IN (SELECT deptno
FROM emp);

EXPLAIN FORMAT = JSON SELECT * FROM dept d WHERE d.deptno = (SELECT
DISTINCT e.deptno FROM emp e WHERE e.deptno = d.deptno);
```