Agenda

```
Subquery
View
```

Subquery

Single Row Subquery

```sql
-- display employee with max sal.
SELECT * FROM emp WHERE sal = (SELECT MAX(sal) FROM emp);

-- display employee with second highest sal
SELECT sal FROM emp ORDER BY sal DESC LIMIT 1,1;
SELECT * FROM emp WHERE sal = (SELECT sal FROM emp ORDER BY sal DESC LIMIT 1,1);

-- display employee with third highest sal
SELECT DISTINCT sal FROM emp ORDER BY sal DESC LIMIT 2,1;
SELECT * FROM emp WHERE sal = (SELECT DISTINCT sal FROM emp ORDER BY sal DESC
LIMIT 2,1);

-- display all emps with dept same as that of KING
SELECT  deptno FROM emp WHERE ename = "KING";
SELECT * FROM emp WHERE deptno = (SELECT  deptno FROM emp WHERE ename = "KING");
SELECT * FROM emp WHERE deptno = (SELECT  deptno FROM emp WHERE ename = "KING")
and ename!="KING";
```

MultiRow Subquery

```
- Inner query return multiple rows
- We have to use ANY,ALL and IN opertors for comparision
```

```sql
-- find all emps having sal more that all salesman
SELECT sal FROM emp WHERE job = "SALESMAN";

SELECT * FROM emp WHERE sal > ALL(SELECT sal FROM emp WHERE job = "SALESMAN");
--sal > 1600 AND sal>1250 AND sal>1500

-- converting to single row subquery
--SELECT * FROM emp WHERE sal >(SELECT MAX(sal) FROM emp WHERE job = "SALESMAN");

-- find emps with sal greater than any of emp in dept 20;
SELECT sal FROM emp WHERE deptno = 20;
SELECT * FROM emp WHERE sal> ANY(SELECT sal FROM emp WHERE deptno = 20);
```

```
SELECT * FROM emp WHERE sal> ANY(SELECT sal FROM emp WHERE deptno = 20) AND
deptno!=20;

-- display the depts having employees
SELECT deptno FROM emp;
SELECT * FROM dept WHERE deptno = ANY(SELECT deptno FROM emp);
SELECT * FROM dept WHERE deptno IN (SELECT deptno FROM emp);

-- display the depts having no employees
SELECT * FROM dept WHERE deptno NOT IN (SELECT deptno FROM emp);
```

Corelated Subquery

```
-- display the depts having employees
SELECT * FROM dept WHERE deptno = ANY(SELECT DISTINCT deptno FROM emp);
--dept-10 ->3 rows
--dept-20 ->14 rows
--dept-30 ->14 rows
--dept-40 ->14 rows

SELECT * FROM dept d WHERE d.deptno = ANY(SELECT e.deptno FROM emp e);

SELECT * FROM dept d WHERE d.deptno = ANY(SELECT e.deptno FROM emp e WHERE
e.deptno = d.deptno );
--dept-10 -> 3 rows
--dept-20 -> 5 rows
--dept-30 -> 6 rows
--dept-40 -> 0 rows

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

Sub-query in Projection

```
-- display no of employees dept wise and also the total no of emp
-- 10 -> 3 14
-- 20 ->  5 14

SELECT COUNT(empno) AS total_count FROM emp;
SELECT deptno,COUNT(empno) emp_count FROM emp GROUP BY deptno;

SELECT deptno,COUNT(empno) emp_count, (SELECT COUNT(empno) FROM emp) AS
total_count FROM emp GROUP BY deptno;
```

Subquery in FROM clause

```
--display emp and category of emp as Rich in sal >2500 and POOR if sal<=2500 using
case
SELECT ename,CASE
WHEN sal>2500 THEN "RICH"
WHEN sal<=2500 THEN "POOR"
END AS category
FROM emp;

-- display count() of emp based on above category
SELECT category FROM
(SELECT CASE
WHEN sal>2500 THEN "RICH"
WHEN sal<=2500 THEN "POOR"
END AS category
FROM emp) AS emp_category;

SELECT category,COUNT(category) FROM
(SELECT CASE
WHEN sal>2500 THEN "RICH"
WHEN sal<=2500 THEN "POOR"
END AS category
FROM emp) AS emp_category
GROUP BY category;
```

SubQuery in DML Operations

```
-- add the emp JOHN having sal 4000 in operations dept
SELECT deptno FROM dept WHERE dname = "OPERATIONS";

INSERT INTO emp(ename,sal,deptno)
VALUES("JOHN",4000,(SELECT deptno FROM dept WHERE dname = "OPERATIONS"));

-- update the comm as 100 where dept is operations
UPDATE emp SET comm = 100 WHERE deptno = (SELECT deptno FROM dept WHERE dname =
"OPERATIONS");

-- delete the employees from operations department
DELETE FROM emp WHERE deptno = (SELECT deptno FROM dept WHERE dname =
"OPERATIONS");


INSERT INTO emp(ename,sal,deptno) VALUES("JOHN",6000,40);
-- remove employee having highest salary

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

SQL Performance

```
SELECT @@optimizer_switch;

-- display the depts having employees
EXPLAIN FORMAT = JSON SELECT * FROM dept WHERE deptno = ANY(SELECT deptno FROM
emp);
-- 5.30

EXPLAIN FORMAT = JSON SELECT * FROM dept d WHERE d.deptno = (SELECT DISTINCT
e.deptno FROM emp e WHERE e.deptno = d.deptno);
-- 0.65
```

VIEW

```
- Simple View
- Complex View
```

```
--display ename,sal,category of emp
--2500> RICH AND <=2500 POOR
SELECT ename,sal,CASE
WHEN sal> 2500 THEN "RICH"
WHEN sal<=2500 THEN "POOR"
END AS category
FROM emp;

CREATE VIEW v_emp_category AS
SELECT ename,sal,CASE
WHEN sal> 2500 THEN "RICH"
WHEN sal<=2500 THEN "POOR"
END AS category
FROM emp;

SHOW TABLES;
SHOW FULL TABLES;

SELECT * FROM v_emp_category;
SELECT ename,category FROM v_emp_category;

-- display category and count of emps
SELECT category, COUNT(ename) FROM v_emp_category GROUP BY category;

-- display category and total sal spent on that category
SELECT category, SUM(sal) FROM v_emp_category GROUP BY category;
```

```
SHOW CREATE TABLE books;
SHOW CREATE TABLE v_emp_category;
SHOW CREATE VIEW books; -- error
SHOW CREATE VIEW v_emp_category;

--Alter the view
ALTER VIEW v_emp_category AS
SELECT ename,sal,CASE
WHEN sal>= 3000 THEN "RICH"
WHEN sal>=2000 AND sal<3000 THEN "MIDDLE"
WHEN sal<2000 THEN "POOR"
END AS category
FROM emp;

-- Delete the View
DROP VIEW v_emp_category;
```

```
CREATE VIEW v_emp_sal AS SELECT empno,ename,sal FROM emp;

CREATE VIEW v_emp_category AS SELECT empno,ename,sal FROM emp WHERE sal>2000;

CREATE VIEW v_emp_da AS SELECT empno,ename,sal*0.5 As DA FROM emp;

CREATE VIEW v_job_summary AS SELECT
job,COUNT(empno),MAX(sal),MIN(sal),AVG(sal),SUM(sal) FROM emp GROUP BY job;

ALTER VIEW v_job_summary AS SELECT job,COUNT(empno) count,MAX(sal) max,MIN(sal)
min,AVG(sal) avg,SUM(sal) sum FROM emp GROUP BY job;

INSERT INTO v_emp_sal VALUES(1,"abc",1000);

INSERT INTO v_emp_category VALUES(2,"cde",1800);
SELECT * FROM v_emp_category;
SELECT * FROM emp;
ALTER VIEW v_emp_category AS SELECT empno,ename,sal FROM emp WHERE sal>2000 WITH
CHECK OPTION;
INSERT INTO v_emp_category VALUES(3,"efg",1900); -- error
INSERT INTO v_emp_category VALUES(3,"efg",2100);

INSERT INTO v_emp_da VALUES(4,"hij",500); -- error
```

```
-- Can we create a view from another view
CREATE VIEW v_emp_cat2 AS SELECT empno,ename,sal FROM v_emp_category;

SELECT * FROM v_emp_cat2;
```

```
DROP VIEW v_emp_category;

SELECT * FROM v_emp_cat2; -- error
```

Example

```
-- display empno,ename,deptno and dname.
SELECT e.empno,e.ename,d.deptno,d.dname FROM emp e INNER JOIN dept d ON
e.deptno=d.deptno;

-- CREATE A VIEW FOR above query.
CREATE VIEW v_emp_dept AS SELECT e.empno,e.ename,d.deptno,d.dname FROM emp e INNER
JOIN dept d ON e.deptno=d.deptno;

-- dipslay all emps from ACCOUNTINg dept(using view and without view)
 SELECT e.empno,e.ename,d.deptno,d.dname FROM emp e INNER JOIN dept d ON
e.deptno=d.deptno WHERE d.dname="ACCOUNTING";

 SELECT * FROM v_emp_dept WHERE dname = "ACCOUNTING";
```