Day 7

VIEW

- DCL Commands are GRANT and REVOKE.

```
CREATE
USER 'dmc'@'localhost'
IDENTIFIED BY 'dmc';

GRANT ALL ON *.* TO 'dmc'@'localhost' WITH GRANT OPTION;
```

```
REVOKE ALL ON *.* FROM 'dmc'@'localhost';
```

- View is SQL feature which is used to give controlled access to the columns for user.
- View is based on select query.
- Syntax:

```
CREATE VIEW view_name [(column_list)]
AS select_statement;
```

- Create copy of emp table;

```
CREATE TABLE employees
AS ( SELECT * FROM emp );

SELECT * FROM employees;
```

- Create view to get all the records from employees table.

```
CREATE VIEW view1
AS ( SELECT * FROM employees );

SHOW TABLES;
SHOW FULL TABLES;
DESC employees;
DESC view1;
```

- Insert record using view1

```
INSERT INTO view1
( empno, ename, sal, deptno)
VALUES
(  7935, 'Sandeep', 8000, 30 ); -- OK
```

- Fetch records using view1

```
SELECT * FROM view1;
SELECT * FROM employees;
```

- View do not store data/record rather it stores pointer of table which is used to store data.
- View contains only select queries.
- Create view using specific columns

```
CREATE VIEW view2
AS ( SELECT empno, ename, sal, deptno FROM employees );

SHOW FULL TABLES;
DESC view2;
SELECT * FROM view2;
```

- Insert record into view2

```
INSERT INTO view2
VALUES( 7936, 'Rahul', 5500, 20 );

SELECT * FROM view2;
SELECT * FROM employees;
```

- Create view with computed column

```
CREATE VIEW view3
AS
( SELECT empno, ename, deptno, sal, sal * 12 "Annual Salary" FROM
employees );

SHOW FULL TABLES;
DESC view3;
SELECT * FROM view3;
```

- Insert record into view3

```
INSERT INTO view3
VALUES( 7937, 'Ketan', 5000, 10, 60000 );
-- Column 'Annual Salary' is not updatable
```

- Create view using where clause

```
CREATE VIEW view4
AS
( SELECT empno, ename, deptno, sal FROM employees WHERE sal > 2000 );

SHOW FULL TABLES;
DESC view4;
SELECT * FROM view4;
```

- Insert record into view4

```
INSERT INTO view4
VALUES( 7937, 'Ketan', 1500, 10 );

SELECT * FROM view4;
SELECT * FROM employees;
```

- Create View using "WITH CHECK OPTION"

```
CREATE VIEW view5
AS
( SELECT empno, ename, deptno, sal FROM employees WHERE sal > 2000 )WITH
CHECK OPTION;

SHOW FULL TABLES;
DESC view5;
SELECT * FROM view5;
```

- Insert record into view5

```
INSERT INTO view5
VALUES( 7937, 'Ketan', 1500, 10 );
```

- Create view using group by

```
CREATE VIEW view6
AS
```

```
(SELECT subject, SUM(price) "Total price" FROM books GROUP BY subject );

SHOW FULL TABLES;
DESC view6;
SELECT * FROM view6;
```

- Insert record into view6

```
INSERT INTO view6
VALUES('C Sharp', 2000);
```

- Create view using joins

```
CREATE VIEW view7
AS
( SELECT e.ename, d.dname
  FROM emp e INNER JOIN dept d
  ON e.deptno = d.deptno
);

SHOW FULL TABLES;
DESC view7;
SELECT * FROM view7;
```

- Insert into view7

```
INSERT INTO view7
( ename, dname )
VALUES( 'ABC', 'QA'); -- Not OK
```

- Create joins using sub query

```
CREATE VIEW view8
AS
(
    SELECT * FROM emp
    WHERE sal > ALL( SELECT sal FROM emp WHERE job='SALESMAN')
);

SHOW FULL TABLES;
DESC view8;
SELECT * FROM view8;
```

- Insert record into view8;

```
INSERT INTO view8
( empno, ename, sal, deptno )
VALUES( 7935,'Sandeep',5000,30); --Not OK
```

- Drop view

```
DROP VIEW view1;
DROP VIEW view2;
DROP VIEW view3;
DROP VIEW view4;
DROP VIEW view5;
DROP VIEW view6;
DROP VIEW view7;
DROP VIEW view8;

-- DROP VIEW view1, view2, view3, view4, view5, view6, view7, view8;
```

Types of view

- There are 2 types of view:
  1. Simple view
     - It allows us to perform DML (INSERT,UPDATE,DELETE)operations
  2. Complex view
     - A view which contains computed column, group by, joins and sub queries in select query then it is called complex view.
  3. Complex view do not allows us to perform DML operations.

Advantages

```
1. If we write view on table then no need write SQL statement multiple
times.
In other words we can achive reusability.
2. View reduces complexity hence it is simple to use.
3. It allows to give access to limited columns as well as to perform
limited operations. In other words, we can achive security.
```

VIEW Security

```
CREATE VIEW view1
AS ( SELECT empno, ename, sal, deptno FROm employees);

SELECT user from user; -- Error
SELECT user from mysql.user; -- OK
```

```
use mysql;
SELECT DATABASE() FROM DUAL;
SELECT user from user;
```

- Create dmc user

```
CREATE USER 'divesd'@'localhost' IDENTIFIED BY 'divesd';
```

- Assign permission

```
GRANT SELECT, INSERT, UPDATE, DELETE
ON classwork.* TO 'divesd'@'localhost' WITH GRANT OPTION;
```

- Create dmc user

```
CREATE USER 'precat'@'localhost' IDENTIFIED BY 'precat';
```

- Use 'divesd'@'localhost' user

```
mysql -u divesd -pdivesd;
SELECT USER() FROM DUAL;
SHOW DATABASES;
USE classwork;
SELECT DATABASE() FROM DUAL;
SHOW TABLES;
```

- Assign permissions to precat.

```
mysql -u divesd -pdivesd;
GRANT SELECT ON classwork.view1 TO 'precat'@'localhost';

SHOW DATABASES;
USE classwork;
SHOW FULL TABLES;
DESC view1;
SELECT * FROM view1;

INSERT INTO view1
VALUES(7937,'Ketan',5000,20);
```

Temporary Table

- If we want to store result returned by select query temporarily then we should create Temporary table.
- It gets space on HDD.
- We can use Temporary table for current session only.
- If we logout or if connection fail then Temporary gets detroy.
- We can give same name to the base table and Temporary table. In this is preference will be given to Temporary table. But it is not recommended.
- Different can create Temporary table with same name but it is not accessible to each other.
- Create Temporary table

```
CREATE TEMPORARY TABLE tbl_name( );
```

```
mysql -u dac -pdac classwork;
CREATE TEMPORARY TABLE new_employees
LIKE  employees;

DESC new_employees;
SHOW TABLES;
INSERT INTO new_employees ( SELECT * FROM employees );
SELECT * FROM new_employees;
EXIT;
```

```
CREATE TEMPORARY TABLE new_employees
AS ( SELECT empno, ename, sal, sal * 12 Total FROM employees );

DESC new_employees;

SELECT SUM( Total ) FROM new_employees;
EXIT;
```

```
CREATE TEMPORARY TABLE new_employees
(
    id INT PRIMARY KEY,
    name VARCHAR( 50 )
);
INSERT INTO new_employees VALUES(1,'DAC');
INSERT INTO new_employees VALUES(1,'DMC');
```

```
CREATE TEMPORARY TABLE new_emp
AS SELECT * FROM emp;

CREATE TEMPORARY TABLE new_dept
AS SELECT * FROM dept;
```

```sql
SELECT e.ename, d.dname
FROM new_emp e INNER JOIN new_dept d
ON e.deptno = d.deptno;

CREATE VIEW view1
AS ( SELECT e.ename, d.dname
FROM new_emp e INNER JOIN new_dept d
ON e.deptno = d.deptno );
```