VIEWS:

VIEW IS DB OBJECT IS CALLED SUBSET OF A TABLE. VIEW IS ALSO CALLED AS VIRTUAL TABLE BECAUSE IT DOESN'T STORE DATA AND IT DOESN'T OCCUPY ANY MEMORY.

VIWE IS CREATING BY USING "SELECT QUERY" FOR GETTING THE REQ.INFORMATION FROM TABLE (BASE TABLE).

TYPES OF VIEWS:

A USER CAN CREATE THE FOLLOWING TWO TYPES OF VIEWS ON BASE TABLES THOSE ARE,

1. SIMPLE VIEWS

2. COMPLEX VIEWS

1. SIMPLE VIEWS:

WHEN WE CREATE A VIEW TO ACCESS REQUIRED DATA FROM A SINGLE BASE TABLE IS CALLED AS SIMPLE VIEWS.

THROUGH A SIMPLE VIEW WE CAN PERFORM ALL DML (INSERT, UPDATE, DELETE) OPERATIONS ON BASE TABLE.

SYNTAX:

CREATE VIEW <VIEW NAME> AS SELECT * FROM <TN> [WHERE <CONDITION>];

EX1:

```
SQL> CREATE VIEW SV1 AS SELECT * FROM DEPT;

SQL> SELECT * FROM SV1;
```

DML OPERATIONS THROUGH A SIMPLE VIEW:

```
QL> INSERT INTO SV1 VALUES (50,'DBA','HYD');

SQL> UPDATE SV1 SET LOC='INDIA' WHERE DEPTNO=50;

SQL> DELETE FROM SV1 WHERE DEPTNO=50;
```

NOTE: WHENEVER WE PERFORM DML OPERATIONS ON VIEW INTERNALLY THE VIEW WILL PERFORM THOSE OPERATIONS ON BASE TABLE.HERE VIEW WILL ACT AS AN INTERFACE BETWEEN USER AND BASE TABLE.

USER <----------> <VIEW> <----------> BASE TABLE

EX2:

SQL> CREATE VIEW SV2 AS SELECT EMPNO, ENAME, JOB, SAL FROM EMP;

TESTING:

SQL> INSERT INTO SV2 VALUES (1122,'SAI','HR',8000); ---ALLOW

SQL> INSERT INTO SV2 VALUES (1122,'WARNER','SR.HR',9500); ----NOT ALLOW (EMPNO COLUMN IS PRIMARY KEY COLUMN IN EMP TABLE)

<u>WITH CHECK OPTION:</u>

IT IS A CONSTRAINT WHICH IS USED TO RESTRICT ROWS ON BASE TABLE THROUGH

A VIEW WHILE PERFORMING DML OPERATIONS.

EX:

SQL> CREATE VIEW SV3 AS SELECT * FROM TEST1 WHERE SAL=18000 WITH CHECK OPTION;

TESTING:

SQL> INSERT INTO SV3 VALUES (1025,'SCOTT',12000); ---NOT ALLOW

SQL> INSERT INTO SV3 VALUES (1025,'SCOTT',58000); ---NOT ALLOW

SQL> INSERT INTO SV3 VALUES (1025,'SCOTT',18000); --- ALLOWED

WITH READ ONLY:

IF WE CREATED A VIEW "WITH READ ONLY" CLAUSE THEN WE RESTRICT DML OPERATIONS.WE ALLOW "SELECT" AND "DESC" COMMANDS.

EX:

```
SQL> CREATE VIEW SV4 AS SELECT * FROM DEPT WITH READ
ONLY;
```

NOTE: NOW WE CANNOT PERFORM DML OPERATIONS THROUGH A VIEW ON BASE TABLE.

2.COMPLEX VIEWS: A VIEW IS CALLED AS COMPLEX VIEW,

I) WHEN WE CREATE ON MULTIPLE BASE TABLES.

II) WHEN WE CREATE A VIEW WITH AGGREGATIVE FUNCTIONS, GROUP BY, HAVING CLAUSES, SET OPERATORS, SUB-QUERY, DISTINCT KEY WORD.

COMPLEX VIEW ARE NOT ALWAYS SUPPORTS DML OPERATIONS.

EX1:

```
SQL> CREATE VIEW CV1 AS SELECT * FROM STUDENT S INNER
JOIN COURSE C
     ON S.CID=C.CID;
```

```
ERROR AT LINE 1:
ORA-00957: DUPLICATE COLUMN NAME
```

NOTE: WHEN WE CREATE A VIEW ON BASE TABLES THEN WE SHOULD NOT ALLOW DUPLICATE COLUMN NAMES.TO AVOID THIS PROBLEM THEN USE "USING" CLAUSE.

```
SQL> CREATE VIEW CV1 AS SELECT * FROM STUDENT S INNER
JOIN COURSE C USING(CID);
```

NOW WE CREATED A COMPLEX VIEW ON MULTIPLE TABLES.BUT NOT ALLOW DML OPERATIONS.

EX2:

```
SQL> CREATE VIEW CV2 AS
        SELECT * FROM EMP_HYD
        UNION
        SELECT * FROM EMP_CHENNAI;
```

> THE ABOVE COMPLEX VIEW CV2 IS NOT ALLOW DML OPERATIONS.

EX3:

```
SQL> CREATE VIEW CV3 AS SELECT DEPTNO, SUM(SAL) FROM
EMP GROUP BY DEPTNO;
```

```
ERROR AT LINE 1:
ORA-00998: MUST NAME THIS EXPRESSION WITH A COLUMN
ALIAS
```

NOTE: WHEN WE CREATE A VIEW WITH FUNCTION THEN WE MUST CREATE ALIAS NAME

FOR THOSE FUNCTIONS OTHERWISE ORACLE RETURNS AN ERROR.

EX:

```
SQL> CREATE VIEW CV3 AS SELECT DEPTNO, SUM(SAL) AS
SUMSAL FROM EMP
        GROUP BY DEPTNO;
```

> THE ABOVE COMPLEX VIEW CV3 NOT ALLOWED DML OPERATIONS.

EX4:

SQL> CREATE VIEW CV4 AS SELECT EMPNO, ENAME, SAL, D. DEPTNO, DNAME, LOC FROM EMP E INNER JOIN DEPT D ON E. DEPTNO=D.DEPTNO;

TESTING:

SQL> UPDATE CV4 SET SAL=500 WHERE EMPNO=7788; --- ALLOWED

SQL> DELETE FROM CV4 WHERE EMPNO=7782; ----ALOOWED

SQL> INSERT INTO CV4 VALUES (1122,'SAI',6000,10,'SAP','HYD'); ---NOT ALLOW

NOTE: GENERALLY COMPLEX VIEW ARE NOT ALLOWED TO PERFORM DML OPERATIONS BUT WE PERFORM UPDATE, DELETE OPERATIONS ON KEY PRESERVED TABLE (I.E PRIMARY KEY) SO THAT COMPLEX VIEWS ARE SUPPORTING DML OPERATION PARTIALLY.

FORCE VIEWS:

GENERALLY, VIEWS ARE CREATED BASED ON TABLES, BUT FORCE VIEWS ARE CREATE WITHOUT TABLES.

SYNTAX:

CREATE FORCE VIEW <VIEW NAME> AS SELECT * FROM <TN>;

EX:

SQL> CREATE FORCE VIEW FV1 AS SELECT * FROM TEST;

WARNING: VIEW CREATED WITH COMPILATION ERRORS.

TESTING:

SQL> SELECT * FROM FV1;

ERROR AT LINE 1:

ORA-04063: VIEW "SCOTT.FV1" HAS ERRORS

SQL> DESC FV1;

ERROR:

ORA-24372: INVALID OBJECT FOR DESCRIBE

> TO ACTIVATE A FORCE VIEW THEN WE SHOULD CREATE A TBALE WITH THE NAME

AS "TEST".

EX:

SQL> CREATE TABLE TEST (SNO INT, NAME VARCHAR2(10));

TABLE CREATED.

TESTING:

SQL> SELECT * FROM FV1; ----ACTIVATED

SQL> DESC FV1; ----ACTIVATED

NOTE: TO VIEW ALL VIEWS DETAILS IN ORACLE DB THEN WE USE THE FOLLOWING DATADICTIONAY IS "USER_VIEWS".

EX:

SQL> DESC USER_VIEWS;

SQL> SELECT VIEW_NAME FROM USER_VIEWS;

SYNTAX TO DROP A VIEW:

SQL> DROP VIEW <VIEW NAME>;

EX:

SQL> DROP VIEW SV1;

SQL> DROP VIEW CV1;

SQL> DROP VIEW FV1;

ADVANTAGES OF VIEWS:

1. IT IS PROVIDING SECURITY.IT MEANS THAT TO EACH USER CAN BE GIVEN PERMISSION TO ACCESS SPECIFIC COLUMNS & SPECIFIC ROWS FROM A TABLE.

2. IF DATA IS ACCESSED AND ENTERED THROUGH A VIEW,THE DB SERVER WILL CHECK DATA TO ENSURE THAT IT MEETS SPECIFIED INTERGRITY CONSTRAINTS RULES OR NOT.

3. QUERY SIMPLIFY IT MEANS THAT TO REDUCE COMPLEX QUERY.

DIFFERENCES BETWEEN SYNONYM AND VIEW:

SYNONYM	VIEW
1. IT IS A MIRROR OF TABLE.	1. IT IS A SUBSET OF TABLE.
2. CREATED ON A SINGLE TABLE.	2. CREATED ON MULTIPLE TABLES.
3. CREATE ON ENTIRE TABLE.	3. CREATED ON SPECIFIC ROWS AND SPECIFIC COLUMNS OF TABLE.
4. NOT SUPPORTS DATA ABSTRACTION.	4. SUPPORTING DATA ABSTRACTION MECHANISM. (HIDE DATA)