**MATERIALIZED VIEWS:**

* Oracle 8i introduced materialized views. Generally views doesn’t store any data where as materialized views are storing data.
* These views are used in data warehousing and handling by DBA .Materialized views also created from base tables.

**DIFFERENCES BETWEEN VIEW AND MATERIALIZED VIEW:**

|  |  |
| --- | --- |
| **VIEW** | **MATERIALIZED VIEW** |
| View does not store any Data. | Materialized view store  Data. |
| When we dropping base Table then view cannot be accessible | When we dropping base  Table then materialized  View can be accessible. |
| We can perform DML operations on view | We cannot perform DML  Operations on materialized view |

**Syntax:**

**CREATE MATERIALIZED VIEW**

**<VIEW NAME> AS SELECT \* FROM**

**<TABLE NAME>;**

**Ex:**

SQL> CREATE TABLE TEST (SNO INT, NAME VARCHAR2 (20));

SQL> CREATE MATERIALIZED VIEW MV1 AS SELECT \* FROM TEST;

**Error:**

Table 'TEST' does not contain a primary key constraint.

**NOTE:** When we create materialized view

On base table then base table should have

Primary key constraint otherwise oracle returns an error.

**Ex:**

SQL> CREATE TABLE TEST1 (SNO INT PRIMARY KEY, NAME VARCHAR2 (20));

SQL> INSERT INTO TEST1 VALUES (1,'A');

…………………………………………………;

SQL> CREATE VIEW V1 AS SELECT \* FROM TEST1;

View created.

SQL> CREATE MATERIALIZED VIEW MV1 AS SELECT \* FROM TEST1;

Materialized view created.

**Testing:**

SQL> UPDATE TEST1 SET NAME=LOWER (NAME);

* Here, base table (test1) and view (v1) table data is updated but materialized view (mv1) table data is not updated. if we want to update data in materialized View then we refresh materialized view by using the following syntax,

**Syntax:**

Exec dbms\_mview.refresh (‘materialized view name’);

Ex: Exec dbms\_mview.refresh (‘MV1’);

**ON DEMAND / ON COMMIT:**

IN ORACLE WE ARE REFRESHING MATERIALIZED VIEW IN TWO WAYS THOSE ARE,

**i) ON DEMAND:**

IT IS A DEFAULT REFERESHING METHOD.IN THIS METHOD WE ARE REFRESHING

MATERIALIZED VIEW BY USING "DBMS\_MVIEW" PROCEDURE.

**SYNTAX:**

EXECUTE DBMS\_MVIEW.REFRESH ('MVIEW NAME');

**EX:**

EXECUTE DBMS\_MVIEW.REFRESH ('MV1');

**ii) ON COMMIT:**

WE CAN REFRESHING A MATERIALIZED VIEW WITHOUT USING "DBMS\_MVIEW" BUT USING "ON COMMIT" METHOD.

**SYNTAX:**

CREATE MATERIALIZED VIEW <VIEW NAME>

REFRESH ON COMMIT

AS SELECT \* FROM <TN>;

**EX:**

Create table test2 (Eid int primary key,

Sal number (10));

**Ex:**

Create materialized view mv2 refresh on commit

As select \* from test2;

**Testing:**

Sql> insert into test2 values (101,25000);

Sql> commit;

Sql> select \* from mv2;

**Note:** if we want to view materialized views

Then we are using the following data dictionary is

“user\_mviews”.

**Ex:**

SQL> DESC USER\_MVIEWS;

SQL> SELECT MVIEW\_NAME FROM USER\_MVIEWS;

**SYNTAX TO DROP MATERIALIZED VIEW:**

SQL> DROP MATERIALIZED VIEW <MVIEW NAME>;

SQL> DROP MATERIALIZED VIEW MV1;

Materialized view dropped.

**FORCE VIEW:**

**SYNTAX:**

**CREATE FORCE VIEW <VIEW NAME> AS**

**SELECT \* FROM <TN>;**

**EX:**

**SQL> CREATE FORCE VIEW FV1 AS SELECT \* FROM TEST1;**

**Warning: View created with compilation errors.**

SQL> CREATE TABLE TEST1(SNO INT,NAME VARCHAR2(20));

Table created.