Performing a Manual Refresh

- DBMS_MVIEW.REFRESH(mv_name, refresh type);
  - ’F’ = FORCE, ‘C’ = COMPLETE
- DBMS_SNAPSHOT.REFRESH(mv_name, refresh type);

Materialized Views - Syntax

```
CREATE MATERIALIZED VIEW view-name
BUILD [IMMEDIATE | DEFERRED]
REFRESH [FAST | COMPLETE | FORCE ]
ON [COMMIT | DEMAND ]
[[ENABLE | DISABLE] QUERY REWRITE]
AS
select_statement;
```

BUILD IMMEDIATE [default]: Populates the MV immediately
BUILD DEFFERED: Populates the MV on the first refresh request

REFRESH FAST: MV is populated with changes to data only (need a materialized view log for this)
REFRESH COMPLETE: MV is truncated and repopulated
REFRESH FORCE [default]: A fast refresh is attempted, otherwise it is complete refreshed

ON COMMIT: MV is refreshed whenever a commit has occurred on a base table
ON DEMAND [default]: MV is refreshed manually or on schedule

ENABLE QUERY REWRITE: Allows query optimizer to use query rewrite
DISABLE QUERY REWRITE [default]: Prevents query optimizer to use query rewrite

FAST REFRESH

- Fast refresh updates the Materialized View with only the changes made to the base table, rather than truncating and re-populating
- To do a fast refresh you need to create a Materialized View log which tracks changes made to the base tables

```
CREATE MATERIALIZED VIEW LOG ON table_name;
```

- You must have CREATE TABLE privileges on the master table
- Fast refresh can only be performed on Simple MVs

COMPLETE AND FORCE REFRESH

- A COMPLETE REFRESH will clear the entire MV Table and populate it again

- A FORCE REFRESH will attempt to do a FAST REFRESH and revert to a COMPLETE refresh if this fails

ON COMMIT VS ON DEMAND

- ON COMMIT will refresh your MV whenever a COMMIT is performed following a DML operation on any dependent table

- ON DEMAND will refresh the MV on a specified schedule

START WITH SYSDATE NEXT SYSDATE + interval in days

ON PREBUILT TABLE

- Converts a pre-existing table into a Materialized View
- The pre-existing table must be a replica of an existing table or created via CTAS
- Table and MV are both separate objects

```
CREATE MATERIALIZED VIEW view_name
ON PREBUILT TABLE
BUILD [IMMEDIATE | DEFERRED]
REFRESH [FAST | COMPLETE | FORCE ]
ON [COMMIT | DEMAND ]
[[ENABLE | DISABLE] QUERY REWRITE]
AS
select_statement;
```

The view name must be the same name as the pre-existing table

The columns used in the select statement must be the same as per the pre-existing table

Simple vs Complex MVs

In general a Materialized View can be considered Complex when it has been constructed using:

- A CONNECT BY clause
- An INTERSECT, UNION ALL or MINUS Set Operation
- The DISTINCT or UNIQUE keyword
- Aggregation Functions and Group By Clauses
- Table Joins