VIEWS:

A view is a virtual table. A view consists of rows and columns just like a table.

The difference between a view and a table is that views are definitions built on top of other tables (or views), and do not hold data themselves.

If data is changing in the underlying table, the same change is reflected in the view. A view can be built on top of a single table or multiple tables. It can also be built on top of another view.

Views offer the following advantages:

1. Ease of use

2. Space savings

3. Additional data security

View WITH CHECK OPTION:

The WITH CHECK OPTION clause prevents users from updating or inserting rows that are not visible through the view.

Updateable View:

You can modify the data of an underlying base table through a view, as long as the following conditions are true:

- Any modifications, including UPDATE, INSERT, and DELETE statements, must reference columns from only one base table.

- The columns being modified in the view must directly reference the underlying data in the table columns. The columns cannot be derived in any other way, such as through the following:

- An aggregate function: AVG, COUNT, SUM, MIN, MAX, GROUPING, STDEV, ...

- A computation. The column cannot be computed from an expression that uses other columns. Columns that are formed by using the set operators UNION, CROSSJOIN, EXCEPT, and INTERSECT amount to a computation and are also not updatable.

- The columns being modified are not affected by GROUP BY, HAVING, or DISTINCT clauses.

- TOP is not used anywhere in the SELECT statement of the view together with the WITH CHECK OPTION clause.

-- Inline Views (Derived Tables):

An inline view is a SELECT statement in the FROM clause.

=====================================================================

-- VIEWS: Create, Alter, Drop, with check option Statements.

-- Creating a VIEW:

CREATE VIEW viewName

AS

SELECT col1, col2, ... FROM tableName

WHERE condition

[WITH CHECK OPTION];

-- Altering a VIEW:

ALTER VIEW viewName

AS

SELECT col1, col2, ... FROM tableName

WHERE condition;

-- Dropping a VIEW:

DROP VIEW viewName;

-- Example1: Select students from dept 10.

CREATE VIEW dept10Students AS

SELECT \*

FROM student

WHERE dno=10;

CREATE VIEW dept10Students AS

SELECT \*

FROM student

WHERE dno=10

WITH CHECK OPTION;

-- Example2: Select department wise summary.

CREATE VIEW deptSummary (deptName, totalStudents, avgCGPA, maxCGPA, minCGPA)

AS

SELECT DName, COUNT(\*), AVG(CGPA), MAX(CGPA), MIN(CGPA)

FROM dept JOIN student ON dept.dno=student.dno

GROUP BY DName;

-- Example3: Select students with a cgpa higher than the average cgpa.

CREATE VIEW StudentsAboveAvg AS

SELECT Rolllno, FirstName, CGPA

FROM student

WHERE CGPA > (SELECT AVG(CGPA) FROM student);

===================================================================

--To Create NEW Table with data from Existing Table (SELECT-INTO Query):

SELECT \* INTO newTable FROM table1;

--To Create NEW empty table with identical structure of existing table:

SELECT \* INTO newtable FROM table1 WHERE 1=0;

SELECT col1, col2, col3 INTO newTable FROM table1

WHERE condition....

--Copy rows FROM table1 into table2 (INSERT-SELECT Query):

INSERT INTO table2

SELECT \* FROM table1;

INSERT INTO table2

SELECT col1, col3, ... FROM table1 WHERE condition..

--The TRUNCATE TABLE Statement

TRUNCATE TABLE table\_name;

---------------------------------------