Views

Views

- A view is a tailored presentation of the data contained in one or more tables or other views.
- A view takes the output of a query and treats it as a table.
- Therefore, a view can be thought of as a stored query or a *virtual table*.

Storage of Views

- Unlike a table, a view is *not allocated any storage space*, *nor* does a view actually *contain data*.
- Rather, a view is *defined by a query* that extracts or *derives data from the tables* that the view references.
- These tables are called **base tables**.
- Base tables can in turn be actual tables or can be views themselves
- A view requires no storage other than storage for the definition of the view (the stored query) in the data dictionary.

Views and Tables

- Because views are derived from tables, they have many similarities:
  - You can define views with up to 1000 columns, just like a table.
  - You can query views, and with some restrictions you can update, insert into, and delete from views.
  - All operations performed on a view actually affect data in some base table of the view and are subject to the integrity constraints and triggers of the base tables.

Use of Views

- **Provide an additional level of table security by restricting access to a predetermined set of rows or columns of a table**
- **Hide data complexity**

  A single view can be defined with a join, which is a collection of related columns or rows in multiple tables.
- **Simplify statements for use**

  Views allow users to select information from multiple tables.
- **Present the data in a different perspective from that of the base table**

  The columns of a view can be renamed without affecting the base tables
- **Isolate applications from changes in definitions of base tables**

  For example, if a view's defining query references three columns of a four column table, and a fifth column is added to the table, then the view's definition is not affected, and all applications using the view are not affected.
- **Save complex queries**

  A query can perform extensive calculations with table information. By saving this query as a view, you can perform the calculations each time the view is queried.

Oracle and Views

- Oracle stores a *view's definition in the data dictionary* as the text of the query that defines the view.
- When you reference a view in a SQL statement, Oracle:
  1. *Merges* the *statement* that references the view *with* the *query that defines the view*
  2. *Parses* the merged statement
  3. *Executes* the statement

Creating Views

- You can create views using the `CREATE VIEW` statement. Each view is defined by a query that references tables, materialized views, or other views.
- The following statement creates a view on a subset of data in the `emp` table:

```
CREATE VIEW sales_staff AS
SELECT empno, ename, deptno
FROM emp
WHERE deptno = 10
WITH CHECK OPTION CONSTRAINT sales_staff_cnst;
```

- Furthermore, the CHECK OPTION creates the view with the constraint (sales_staff_cnst) that INSERT and UPDATE statements issued against the view cannot result in rows that the view cannot select.
- In the Above example one can Insert/Update for rows where deptno = 10 only. When one tries insert/update for the rows whose deptno value <> 10 then the statement gives an error.
- If one doesn't specify the CHECK OPTION then changes could be made in any other rows irrespective of the deptno.

Join Views

- A **join view** is defined as a view that has more than one table or view in its `FROM` clause (a **join**) and that does not use any of these clauses: `DISTINCT, AGGREGATION, GROUP BY, START WITH, CONNECT BY, ROWNUM,` and set operations `(UNION ALL, INTERSECT,` and so on).
- An **updatable join view** is a join view that involves two or more base tables or views, where `UPDATE, INSERT,` and `DELETE` operations are permitted.
- In order to be inherently updatable, a view cannot contain any of the following constructs:
  - A set operator
  - A `DISTINCT` operator
  - An aggregate or analytic function
  - A `GROUP BY`, `ORDER BY`, `CONNECT BY`, or `START WITH` clause
  - A collection expression in a `SELECT` list
  - A subquery in a `SELECT` list
  - Joins (with some exceptions).
  - Views that are not updatable can be modified using `INSTEAD OF` triggers.

Creating Join Views

- You can also create views that specify more than one base table or view in the `FROM` clause. These are called **join views**.
- The following statement creates the `division1_staff` view that joins data from the `emp` and `dept` tables:

```
CREATE VIEW division1_staff AS
SELECT ename, empno, job, dname
FROM emp, dept
WHERE emp.deptno IN (10, 30)
AND emp.deptno = dept.deptno;
```

- An **updatable join view** is a join view where `UPDATE`, `INSERT`, and `DELETE` operations are allowed.

Updatable Join Views

- An updatable join view (also referred to as a **modifiable join view**) is a view that contains more than one table in the top-level `FROM` clause of the `SELECT` statement, and is not restricted by the `WITH READ ONLY` clause.

Rule	Description
General Rule	•Any `INSERT`, `UPDATE`, or `DELETE` operation on a join view can modify only one underlying base table at a time.
`UPDATE` Rule	•All updatable columns of a join view must map to columns of a **key-preserved table**. If the view is defined with the `WITH CHECK OPTION` clause, then all join columns and all columns of repeated tables are non-updatable.
`DELETE` Rule	•Rows from a join view can be deleted as long as there is exactly one key-preserved table in the join. If the view is defined with the `WITH CHECK OPTION` clause and the key preserved table is repeated, then the rows cannot be deleted from the view.
`INSERT` Rule	•An `INSERT` statement must not explicitly or implicitly refer to the columns of a **non-key preserved table**. If the join view is defined with the `WITH CHECK OPTION` clause, `INSERT` statements are not permitted.