

Views in oracle

View:

- It is a logical table based on one or more tables or views.
- A view in practicality contains no data by itself.
- The tables upon which a view is based are called as base tables.
- Views can be created as object views or relational views.
- The object views support
 - Lob's
 - Object types
 - Ref's
 - Nested tables
 - Varrays
- Object view is a view of a user defined type, where each row contains objects, and each object with an object identifier.

Prerequisites:

- Should have create view or create any view system privilege.
- To create a subview, we need "under any view" system privilege or "under object" privilege on the superview.
- The owner of the schema should have select, insert, update or delete rows from all tables or views on which the view is based.
- The above privileges must be granted through privileges directly, rather than a role.

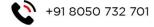
Syntax:

```
SQL> create [or replace] [{force/noforce}] view
                                                   viewname
 [(aliasname, [aliasname...])] As subquery [with [check option/read
 [constraint constraintname];
```

Or replace:

Specifies the view has to be replaced if already existing.

Force:











Dream, Create, Achieve,

FERILION LABS

Specifies the view has to be created even if the base table does not exist, which is default.

Alias name:

Specifies the name of an alias for an expression in the subquery.

With check option:

Specifies that only rows that would be retrieved by the subquery can only be inserted , updated or deleted.

Constraint:

Constraint_name specifies the name of the constraint as with check option or read only constraint.

With read only:

Specifies that rows must only be read from the base tables.

Restrictions:

- If a view has instead of trigger, then all sub views created on it must have instead of triggers, even if the views are inherently updatable.
- An alias cannot be specified when creating an object view.

Types of views:

- Simple views
- Complex views

Simple views:

Which contains a subquery that retrieves data from only one base table.

Complex views:

- Contains a subquery that can perform any of these actions.
 - Retrieving from multiple base tables.
 - Groups rows using a group by or distinct clause.
 - Contains a function call.











Dream, Create, Achieve.

Simple views - syntax:

SQL> create view employees as select empno "id number", ename name, sal "basic salary", job designation from emp;

Selecting data from a view:

SQL>select name, designation from employees; SQL>select "id number", name, to_char("basic salary",'99,99,999.99) monthly, "basic salary" * 12 annual from employees where "basic salary" > 2500;

Complex views:

SQL> create view empinfo as select e.empno employeeid, e.ename name, d.deptno departmentid, d.dname deapartmentname from emp e, dept d where d.deptno=e.deptno order by d.deptno;

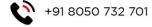
SQL> create view empgrades as select e.ename name, e.sal basic, s.grade grade from emp e, salgrade s where e.sal between s.losal and s.hisal order by s.grade;

SQL> create or replace empmanagers as select rownum serialno, initcap(e.ename)||' works under '||m.ename "employee and managers" from emp e, emp m where e.sal between s.losal and s.hisal order by s.grade;

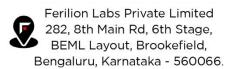
SQL>create or replace view empaccounts as select ename, deptno, sal monthly, sal * 12 annsal from emp where deptno=(select deptno, from dept where dname='accounting') order by annual;

SQL> create or replace view cummsum as select b.sal, sum(a.sal) cum sal from emp a, emp b where a.rowid<=b.rowid group by b.rowid, b.sal;

SQL> create or replace view orgdesignations as select job from emp where deptno=10 union select job from emp where deptno in(20, 30);











Data access using views:

- The steps or operations performed by the oracle server, when data is accessed using a view are
 - Retrieves the view definition from the data dictionary table user_views.
 - Checks the access privileges for the views base table.
 - Converts the view query into an equilent operation on the underlying base table or tables.

Views in data dictionary:

- Once the view has been created, we can query upon the data dictionary table called user_views to see the name and definition of the view.
- The text of the select statement that constitutes the view is stored in a long column.

SQL> select view_name, text from user_views;

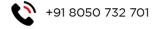
Modifying a view:

- Or replace option is used to modify an existing view with a new definition.
- A view can be altered without dropping, recreating, and regranting object privileges.
- The assigned column aliases in the create view clause, are listed in the same order as the columns in the subquery.

Creating views with columns declarations:

- When a view is being created, we can specify the names of the columns, that it can project, along with the view's definition.
- The view in this case totally hides the original names from the basic table.

```
SQL> create view deptsalsummary
Departmentname,
Minimumsalary,
Maxsalary,
Averagesalary,
Salarysum
)
As
Select d.dname, min(e.sal), max(e.sal), avg(e.sal), sum(e.sal) from emp
e, dept d where e.deptno=d.deptno group by d.dname;
```











Using views to create on the fly tables:

SQL>create view insertdept10 as select * from emp where deptno=10;

Creating a table with data using a view:

SQL>create table dept10 as select * from insertdept10; SQL>create view empgradeins as select ename, job, sal, grade from emp e, salgrade s where e.sal between s.losal and s.hisal; SQL> create table empgrades (employee, designation, basicsalary, grade) as select * from empgradeins;

Dropping a view:

- The drop view statement is used to remove a view permanently.
- Dropping a view has no effect on the tables upon which the view is created.
- Views or applications based on deleted views become invalid.
- We need drop any view privilege to remove the views from data dictionary.

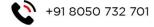
Syntax:

SQL> drop view viewname;

Inline views:

- An inline view is a subquery with an alias (called as correlation name), that can be used within a SQL statement.
- An inline view is not a schema object.
- An inline view in the from clause of a select statement defines a data source for the select statement.

SQL>select el.ename, el.sal, el.deptno, e2.maxsal from emp el, (select deptno, max(sal) maxsal from emp group by deptno) e2 where e1.deptno=e2.deptno and e1.sal < e2.maxsal;











Performing DML operations on a view:

Dml operations can be performed upon a table through views

But the views should be only simple views.

We cannot perform DML operations on complex views. Complex views are which are containing the following elements....

Rules to follow:

- Group functions.
- A group by clause.
- The ROWNUM Pseudo column.
- The DISTINCT key word.
- The columns defined by expressions.
- Data can be added through a view, unless it contains any of the above rules and there does not exist not null columns, and without default value.
- Many times, these transactions look consistent but it is not true, hence care should be taken when implementing them.

Using with check option clause:

- To ensure that DML on the view stays within the domain of THE VIEW we use the WITH CHECK option clause.
- Views make it possible to perform referential integrity checks.
- Using views, we can enforce constraints at database level.
- Using views we can protect the data integrity, but the use is very limited.
- Using views we can protect the data integrity, but the use is very limited.
- The with check option clause specifies that inserts, & updates performed through the view are not allowed to create rows, that the view cannot select.
- Views allow integrity constraints and data validation checks to be enforced on data being inserted or updated.

SQL> CREATE OR REPLACE VIEW EDEPT30 AS SELECT * FROM EMP WHERE DEPTNO=30 WITH CHECK OPTION CONSTRAINT EDept30chkview;











Dream. Create. Achieve.

SQL> CREATE OR REPLACE VIEW EMANAGE AS SELECT * FROM EMP WHERE JOB = 'MANAGER' WITH CHECK OPTION CONSTRAINT EManagerview;

Applying with read only option:

- By adding the WITH READ ONLY option we can ensure that NO DML operations are executed through view.
- An attempt to perform a DML operation results in oracle server error.

SQL> CREATE OR REPLACE VIEW EDptRead(EmpID, Name, Designation) AS SELECT Empno, Ename, Job FROM Emp WHERE Deptno=20 WITH READ ONLY;

