**Database Management System**

**Practical No : 9**

**Aim :**  i) To perform operations on views

ii) To perform operations on sequences

## **view**

* Views in SQL are considered as a virtual table. A view also contains rows and columns.
* To create the view, we can select the fields from one or more tables present in the database.
* A view can either have specific rows based on certain condition or all the rows of a table.

**Creating View**

CREATE or REPLACE VIEW view\_name

AS

SELECT column\_name(s)

FROM table\_name

WHERE condition

CREATE or REPLACE VIEW emp\_view

AS

SELECT \* FROM employees WHERE department\_id=90 and salary>1000;

The data fetched from SELECT statement will be stored in another object called emp\_view. We can use CREATE and REPLACE separately too, but using both together works better, as if any view with the specified name exists, this query will replace it with fresh data.

## **Force view creation**

FORCE keyword is used while creating a view, forcefully. This keyword is used to create a View even if the table does not exist. After creating a force View if we create the base table and enter values in it, the view will be automatically updated.

CREATE or REPLACE FORCE VIEW view\_name AS

SELECT column\_name(s)

FROM table\_name

WHERE condition;

## **Selecting a data set from a view**

select <column1>,<column2>,....<columnN> from <view\_name> ;

## **Update a view**

UPDATE view-name SET VALUE WHERE condition;

## **Destroying a view**

Drop view <view\_name>;

## **Read-only view :** We can create a view with read-only option to restrict access to the view.

CREATE or REPLACE FORCE VIEW view\_name AS

SELECT column\_name(s)

FROM table\_name

WHERE condition WITH read-only;

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

## **Sequences**

A sequence is a set of integers 1, 2, 3, ... that are generated in order on demand. Sequences are frequently used in databases because many applications require each row in a table to contain a unique value and sequences provide an easy way to generate them.

## **Creating sequences**

CREATE SEQUENCE sequence-name

START WITH initial-value

INCREMENT BY increment-value

MAXVALUE maximum-value

CYCLE | NOCYCLE;

* The **initial-value** specifies the starting value for the Sequence.
* The **increment-value** is the value by which sequence will be incremented.
* The **maximum-value** specifies the upper limit or the maximum value upto which sequence will increment itself.
* The keyword CYCLE specifies that if the maximum value exceeds the set limit, sequence will restart its cycle from the begining.
* And, NO CYCLE specifies that if sequence exceeds MAXVALUE value, an error will be thrown.

Let's start by creating a sequence, which will start from 1, increment by 1 with a maximum value of 999.

CREATE SEQUENCE seq\_1

START WITH 1

INCREMENT BY 1

MAXVALUE 999

CYCLE;

**Consider a relation/table: class(rollno, name)**

How to use it: INSERT INTO class VALUE(seq\_1.nextval, 'anu');

Whenever you use nextval the value of that sequence will be incremented/decremented by specified value.

## **Referencing a sequence**

Select <sequence\_name>.nextval from dual;

## **Altering sequences**

ALTER SEQUENCE sequence-name

START WITH initial-value

INCREMENT BY increment-value

MAXVALUE maximum-value

CYCLE | NOCYCLE;

## **Destroying a sequence**

Drop sequence <sequence\_name> ;

**Exercise**

1. The organization wants to display only the details of the employees those who are managers [Manager\_Record].
2. The organization wants to display only the details like empno, empname, deptno, deptname of the employees [Employee\_Detail].
3. The organization wants to display only the details like empno,empname,deptno,deptname of the all the employees except the managers and [NoManager].
4. Display all the views generated.
5. Execute the DML commands on the view created.
6. Find the departments of all mangers from Manager\_detail.
7. Find name along with department name from Employee\_detail.
8. Find Eno, and their corresponding dname from No\_Manager.
9. Add a column Address in Manager\_Record.
10. Change a column name Deptno to D\_ID in No\_Manager.
11. Change size of Empname column to 20 in Employee\_Detail
12. Drop a view.
13. Create a sequence to insert the data in table person(pid, name, age), which automatically takes the value of pid, which starts with 101 and incremented by 1, and the valid range for pid is 101-199.
14. Update the sequence created in the above question and increment the value with 2.
15. What is view?
16. What is Indexed view? How to create it?
17. What are partitioned views and distributed partitioned views?
18. What functions can a view be used to performed?
19. Describe the functionalities that views support.
20. What are the restrictions that views have to follow?
21. Explain the use of cache option in creation of a sequence.