Experiment No. 7

Environment: Microsoft Windows
Tools/ Language: Oracle

Objective: To implement the concept of views.

Theory:

Views:-
Logical data is how we want to see the current data in our database. Physical data is how this data is actually placed in our database.
Views are masks placed upon tables. This allows the programmer to develop a method via which we can display predetermined data to users according to our desire.
Views may be created for the following reasons:
1. The DBA stores the views as a definition only. Hence there is no duplication of data.
2. Simplifies Questionnaires.
3. Can be queried as a base table itself.
4. Provides data security.
5. Avoids data redundancy.

Creation of Views:-
Syntax:-
```
CREATE OR REPLACE VIEW viewname AS
SELECT columnname,columnname
FROM tablename
WHERE columnname=expression_list;
```

Renaming the columns of a view:-

Syntax:-
```
CREATE OR REPLACE VIEW viewname (newcolumnname, … ) AS
SELECT columnname….
FROM tablename
WHERE columnname=expression_list;
```

Selecting a data set from a view-

Syntax:-
```
SELECT columnname, columnname
FROM viewname
WHERE search condition;
```

Destroying a view-

Syntax:- `DROP VIEW viewname;`

Classification of Views

Views are of two types based on their behavior during data manipulation

Updatable Views

Updatable here signifies that one can insert, update and delete rows and data from view. (Actually all DML manipulations are reflected to *base* table)

1. It is created from single table
2. The view must include the PRIMARY KEY and NOT NULL columns of the table based upon which the view has been created.
3. No Aggregate function such as SUM, Count, AVG be used.
4. The view must not have any DISTINCT, GROUP BY or HAVING clause in it's definition.
5. Any of the selected output fields (of the view) must not use constants, strings or value expressions.
6. If the view you want to update is based upon another view, the later should be updatable.
7. The view must not have any SUBQUERIES in it's definitions

Non Updatable Views

Non Updatable View is the one in which no one can INSERT, UPDATE or DELETE i.e, no changes can be applicable.

1. It is created from more than one table i.e. using JOINS
2. Having aggregate function or GROUP BY, DISTINCT or HAVING be used it view Definition. (Even if view is of single table and have any of these clause then still it is non-updatable)
3. Not include all NOT NULL columns or PRIMARY KEYS.

Run the following script:

```
drop table Apply;
drop table Student;
drop table  College;
create table Student(sID int primary key, sName varchar2(10), GPA
real, sizeHS int, DoB date);
insert into Student values (123, 'Amy', 3.9, 1000, '26-JUN-96');
insert into Student values (234, 'Bob', 3.6, 1500, '7-Apr-95');
insert into Student values (345, 'Craig', 3.5, 500, '4-Feb-95');
insert into Student values (456, 'Doris', 3.9, 1000, '24-Jul-97');
insert into Student values (567, 'Edward', 2.9, 2000, '21-Dec-96');
insert into Student values (678, 'Fay', 3.8, 200, '27-Aug-96');
insert into Student values (789, 'Gary', 3.4, 800, '8-Oct-96');
insert into Student values (987, 'Helen', 3.7, 800, '27-Mar-97');
insert into Student values (876, 'Irene', 3.9, 400, '7-Mar-96');
insert into Student values (765, 'Jay', 2.9, 1500, '8-Aug-98');
insert into Student values (654, 'Amy', 3.9, 1000, '26-May-96');
insert into Student values (543, 'Craig', 3.4, 2000, '27-Aug-05');
commit;
```

EXAMPLES

Updatable View

Creating View named as STU_VIEW on Student Table:

```
create or replace view STU_VIEW as
        select GPA, sNAme, DoB from Student;
```

Display Records from View

```
Select * from STU_VIEW;
```

Now, try to update the view (*check DoB of sID 765 before running the query*)

```
Update STU_VIEW
Set DoB=Add_months(DoB,4)
Where sName='Jay';
```

Check both base table as well as view to see it changes are reflecting or not

```
Select * from STU_VIEW;
```

```
Select * from Student;
```

Non-Updatable View

Create **Non Updatable View:**

```
create or replace view sView as select sName, GPA from Student;
```

Display records:

```
select * from sView;
```

Add **record** to sView:

```
Insert into sView values ('Shikhar', 4.0);
```

Error Generated:
SQL Error: ORA-01400: cannot insert NULL into ("SYSTEM"."STUDENT"."SID")

This error is generated because view does not include **Primary Key** of **S**tudent **T**able.

View with check option

We have created a view bornAUG on Student Table with check option.

```
create or Replace view bornAUG
as    Select * from Student where EXTRACT(month from DoB)=8
      with check option;
```

Check the view using:

```
Select * from bornAUG;
```

Now, we try to insert an row:

```
insert into bornAUG values (546,'Ram',3.1,900,'15-Aug-1998');
```

Now , Again check both view bornAUG and Student have entry for Ram or not (*you will find him in both*):

```
Select * from bornAUG;
```

```
Select * from Student;
```

Now, try to insert a row for Raj who born in December:

```
insert into bornAUG values (547,'Raj',3.2,1500,'23-Dec-1998');


SQL Error: ORA-01402: view WITH CHECK OPTION where-clause
violation
01402. 00000 -  "view WITH CHECK OPTION where-clause violation"
```

Practical Assignment - 7

Department: Computer Engineering & Applications

Course: B.Tech. (CSE)

Subject: Database Management System Lab (BCSC 0802)

Year: 2nd **Semester:** 3rd

SQL Script for this Experiment

```
BEGIN
 FOR cur_rec IN (SELECT object_name, object_type
                 FROM user_objects
                WHERE object_type IN
                        ('TABLE',
                         'VIEW',
                         'PACKAGE',
                         'PROCEDURE',
                         'FUNCTION',
                         'SEQUENCE'
                        ))
 LOOP
    BEGIN
       IF cur_rec.object_type = 'VIEW' OR cur_rec.object_type = 'TABLE'
       THEN
          EXECUTE IMMEDIATE   'DROP '
                            || cur_rec.object_type
                            || ' "'
                            || cur_rec.object_name
                            || '" CASCADE CONSTRAINTS';
       ELSE
          EXECUTE IMMEDIATE   'DROP '
                            || cur_rec.object_type
                            || ' "'
                            || cur_rec.object_name
                            || '"';
       END IF;
    EXCEPTION
       WHEN OTHERS
       THEN
          DBMS_OUTPUT.put_line (  'FAILED: DROP '
                                || cur_rec.object_type
                                || ' "'
                                || cur_rec.object_name
                                || '"'
                               );
    END;
END LOOP;
END;
/
commit;
```

```
drop table Apply;
drop table Student;
drop table  College;
create table College(cName varchar2(10) primary key, state
varchar2(10), enrollment int);
create table Student(sID int primary key, sName varchar2(10), GPA
real, sizeHS int, DoB date);
create table Apply(sID int, cName varchar2(10), major
varchar2(20), decision char(1), primary key(sID, major, cName),
constraint sID_fk Foreign key(sID) references Student, constraint
cName_fk Foreign key(cName) references College);

delete from Student;
delete from College;
delete from Apply;

insert into Student values (123, 'Amy', 3.9, 1000, '26-JUN-96');
insert into Student values (234, 'Bob', 3.6, 1500, '7-Apr-95');
insert into Student values (345, 'Craig', 3.5, 500, '4-Feb-95');
insert into Student values (456, 'Doris', 3.9, 1000, '24-Jul-97');
insert into Student values (567, 'Edward', 2.9, 2000, '21-Dec-96');
insert into Student values (678, 'Fay', 3.8, 200, '27-Aug-96');
insert into Student values (789, 'Gary', 3.4, 800, '8-Oct-96');
insert into Student values (987, 'Helen', 3.7, 800, '27-Mar-97');
insert into Student values (876, 'Irene', 3.9, 400, '7-Mar-96');
insert into Student values (765, 'Jay', 2.9, 1500, '8-Aug-98');
insert into Student values (654, 'Amy', 3.9, 1000, '26-May-96');
insert into Student values (543, 'Craig', 3.4, 2000, '27-Aug-05');
insert into College values ('Stanford', 'CA', 15000);
insert into College values ('Berkeley', 'CA', 36000);
insert into College values ('MIT', 'MA', 10000);
insert into College values ('Cornell', 'NY', 21000);
insert into College values ('Harvard', 'MA', 50040);
insert into Apply values (123, 'Stanford', 'CS', 'Y');
insert into Apply values (123, 'Stanford', 'EE', 'N');
insert into Apply values (123, 'Berkeley', 'CS', 'Y');
insert into Apply values (123, 'Cornell', 'EE', 'Y');
insert into Apply values (234, 'Berkeley', 'biology', 'N');
insert into Apply values (345, 'MIT', 'bioengineering', 'Y');
insert into Apply values (345, 'Cornell', 'bioengineering', 'N');
insert into Apply values (345, 'Cornell', 'CS', 'Y');
insert into Apply values (345, 'Cornell', 'EE', 'N');
insert into Apply values (678, 'Stanford', 'history', 'Y');
insert into Apply values (987, 'Stanford', 'CS', 'Y');
insert into Apply values (987, 'Berkeley', 'CS', 'Y');
insert into Apply values (876, 'Stanford', 'CS', 'N');
insert into Apply values (876, 'MIT', 'biology', 'Y');
insert into Apply values (876, 'MIT', 'marine biology', 'N');
insert into Apply values (765, 'Stanford', 'history', 'Y');
insert into Apply values (765, 'Cornell', 'history', 'N');
insert into Apply values (765, 'Cornell', 'psychology', 'Y');
insert into Apply values (543, 'MIT', 'CS', 'N');
commit;
```

Student

sID	sName	GPA	sizeHS	DoB
123	Amy	3.9	1000	26-JUN-96
234	Bob	3.6	1500	7-Apr-95
345	Craig	3.5	500	4-Feb-95
456	Doris	3.9	1000	24-Jul-97
567	Edward	2.9	2000	21-Dec-96
678	Fay	3.8	200	27-Aug-96
789	Gary	3.4	800	8-Oct-96
987	Helen	3.7	800	27-Mar-97
876	Irene	3.9	400	7-Mar-96
765	Jay	2.9	1500	8-Aug-98
654	Amy	3.9	1000	26-May-96
543	Craig	3.4	2000	27-Aug-98

Apply

sID	cName	major	decision
123	Stanford	CS	Y
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	Y
234	Berkeley	biology	N
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
678	Stanford	history	Y
987	Stanford	CS	Y
987	Berkeley	CS	Y
876	Stanford	CS	N
876	MIT	biology	Y
876	MIT	marine biology	N
765	Stanford	history	Y
765	Cornell	history	N
765	Cornell	psychology	Y
543	MIT	CS	N

College

cName	state	enrollment
Stanford	CA	15000
Berkeley	CA	36000
MIT	MA	10000
Cornell	NY	21000
Harvard	MA	50040

Write SQL queries for the following:
Create the following views and run the update and delete commands as specified.
State reasons in case the view is non-updateable.

Q1. Create view WeakStudent on Student whose GPA is less than 3.7.

Q2. Create a view cView on college (containing all the columns) and rename the column cName as collegeName and enrollment as seats respectively.

Q3. Create view CSaccept having IDs and college of student who applied to CS major and their application is accepted.

Q4. Create view CSberkeley having IDs, name, GPA, sizeHS of those student who are accepted in CS at Berkeley and comes from High School that is greater than 500. (Instruction: *You are required to use view CSaccept in this query with Student table for joining*)

Q5. Display information about student in CSberk having GPA greater than 3.8.

Q6. Drop view CSaccept.

Q7. Display all student in CSberkeley.

Q8. Update GPA by 0.8 of students in view WeakStudent who having high school size greater than 1000.

Q9. Create a view AppCount which contains sID of Student and number of applications they filed. Name the column sID and NoOfApp.

Q10. Update NoOfApp so that sID 234 contains 4 applications.

Q11. Create a view StuName contain student name and their GPA. Is this View Updatable? If not specify why.

Q12. Create view studentHS having details of student comes from High School of size more than 1000 using *with check option.*

Q13. Try insert detail of a new student Ram with sID 999 having GPA 9.9 and sizeHS 9999 to newly created view studentHS. (*Comment is this view updatable?*)

Q14. Clerk realize he wrongly type sizeHS of Ramu as 9999 it is actually 999. Can you help him to update the value of sizeHS of Ramu.

Q15. Now, another boy registered to our system named Ramu with sID 998 having GPA 9.8 and sizeHS 989.

Pre-Experiment Questions:

1. What is the need of creating a view?
2. What are advantages that a view offers?

Post Experiment Questions:

1. What do you understand by non-updateable view?
2. List the cases in which a view is non-updateable?
3. Can create view on other views**?**
4. **Can we** drop view on which another view is constructed?