Laboratory 3

Views
Updateable, non-updateable, safe updateable

Development and Editing: Z. Garofalaki, A. Tsolakidis, P. Andritsos

Target

Familiarity with the use of views (VIEWS), types of views.

Tools

A. View management

A view is a virtual table that includes elements from one or more tables or other views of the database. The view has no "physical" existence, i.e. it does not exist as an array of stored elements. Nevertheless, the elements of the view directly reflect the changes made to the content of the tables on which it is based. The user manages the views like real tables (with some limitations). Their use is recommended in cases such as:

i. simplification of data access,
ii. data integrity,
iii. data independence,
iv. data security.

1. Creating **a view** is done with the following general command form.

```
create view view_name as select column1, column2, …. from table_name where condition;
```

2. The **creation of a view** named OPSI_P and containing the PID and P_DESCR columns of the PINAKAS table is done with the command:

```
create view OPSI_P as select PID, P_DESCR from PINAKAS ;
```

3. Deleting **a face** named OPSI_P is done with the command:

```
drop view OPSI_P;
```

B. Types of aspects

Views are virtual tables based on the result set of an SQL statement. They are defined as **updatable** when they allow users to update data by inserting, updating, or deleting (insert, update, delete). Changes to the updateable view are also reflected in the reference table. For updatable views, the reverse is also true: changes in the reference table are also reflected in the view. The conditions that an updateable view must meet are:

• Do not contain the terms distinct, order by, group by or having (select)
• Do not contain functions (min, max, avg, count(*), etc.) (select)
• Do not contain action results (select)
• Do not contain data from more than one table (from)
• Do not contain nested select (where)
• To contain all the columns of the reference table (select)

A **non** -updatable view is defined as a view that, due to its construction, does not allow users to update data through insertion, update or deletion (insert, update, delete). The conditions that make a view non-updatable are the omission of any of the conditions that an updateable view meets.

Updatable views can also be created as **safe updatable** or **guarded** (symmetric) views. Protected views meet all the requirements that updatable views do, but they do not allow users to update data via insert, update, or delete. Creating **a scrollable view** is done with the following general command form:

```
create view view_name as
select column1, column2, …. from table_name where condition
with check option?
```

Activities

Carry out the following activities. The command or commands required for the implementation of each step, as well as the result of its execution should be included in a deliverable file in text or screenshot format. The file or files with your answers should be compressed into a **xx_ÿÿÿÿÿ_EPONYMO.zip,** where: (a) xx is the number of the section you belong to (e.g. for group [02] MONDAY 12:00-13 :00, **xx = 02)** and (b) YYYYYY your Registration Number. This final file will be submitted to the e-class -> DATABASES II -> Assignments.

1. **Connect** to your system's MySQL using any of the above methods you wish.

2. Check if there is a DB with the name **personnel.**

3. Select the NW personnel to **use.**

4. Create the view named EMP_VIEW.

```
create view EMP_VIEW(e_ID, e_Name, e_Job, e_Dept, e_Comm)
as
select EMPNO, NAME, JOBNO, DEPTNO, COMM from EMP;
```

5. Display the contents of the view :

```
select * from EMP_VIEW;
```

6. New entries in the EMP table also affect the EMP_VIEW view. Try it insert, update, delete statements on the EMP table and display their effect on the EMP table and the EMP_VIEW view.

```
insert into EMP(EMPNO, NAME, JOBNO, DEPTNO, COMM) values (90, 'CLARKE', 100, 50, NULL)

select * from EMP;
select * from EMP_VIEW;
```

7. New entries in the EMP_VIEW view also affect the EMP table. Test insert, update, delete statements on EMP_VIEW view and display their effect on EMP table and EMP_VIEW view.

```
insert into EMP_VIEW(e_ID, e_Name, e_Job, e_Dept, e_Comm) values
(100, 'adams', 100, 60, null);

update EMP_VIEW set e_Job=200 where e_ID=100;

select * from EMP;
select * from EMP_VIEW;
```

8. Create a view that displays the employees of the SALES department

```
create view EMP_ON_SALES(e_ID, e_Name, e_Job, e_Dept, e_Comm) as select
EMPNO, NAME, JOBNO, DEPTNO, COMM from EMP
where DEPTNO in (select DEPTNO from DEPT where DNAME='SALES');

select * from EMP;
select * from EMP_ON_SALES;
```

9. Enter data into the EMP table:

```
insert into EMP(EMPNO, NAME, JOBNO, DEPTNO, COMM) values (110, 'NAVATHE', 100, 60, NULL);
```

10. Does the entry in **step 9** appear in the EMP table? Give her a reason your answer.

11. Does the data from the **step 9** record appear in the EMP_ON_SALES view? Justify your answer.

12. Enter data into the EMP_ON_SALES view:

```
insert into EMP_ON_SALES(e_ID, e_Name, e_Job, e_Dept, e_Comm) values
(120, 'ELMASRI', 100, 60, NULL);
```

13. Does the entry in **step 12** appear in the EMP table? Justify your answer.

14. Does the data from the **step 12** record appear in the EMP_ON_SALES view? Justify your answer.

15. Create the following view:

```
create view EMP_ON_SALES_S(e_ID, e_Name, e_Job, e_Dept, e_Comm)
as
select EMPNO, NAME, JOBNO, DEPTNO, COMM from EMP
where DEPTNO in(select DEPTNO from DEPT where DNAME='SALES') with check option;
```

16. Enter data into the EMP_ON_SALES_S view:

```
insert into EMP_ON_SALES_S(e_ID, e_Name, e_Job, e_Dept, e_Comm)
values (130, 'DATE', 100, 60, NULL);
```

17. Does the entry in **step 16** appear in the EMP table? Give reasons your answer.

18. Are the details of the **step 16** record displayed in the EMP_ON_SALES_S view? Justify your answer.

19. Create the following view:

```
create view EMP_DISTINCT_NAMES (NAME) as select distinct NAME from EMP order by NAME;
```

20. What type of view is EMP_DISTINCT_NAMES? Run the following command and justify its result:

```
insert into EMP_DISTINCT_NAMES values ('GREEN');
```

21. What type of views are the following and what does each of them contain:

```
DROP VIEW IF EXISTS GROUP_EMP; //optionally run to remove a face

create view GROUP_EMP(DEPT, COUNT_EMP, AVG_COMM) as select DEPTNO, count(*), avg(COMM) from
EMP group by DEPTNO;

create view EMP_DEPT_VIEW(EMPNO, NAME, JOBNO, DEPTNO, DNAME)
as select EMPNO, NAME, JOBNO, EMP.DEPTNO, DNAME
from EMP inner join DEPT on EMP.DEPTNO=DEPT.DEPTNO;

create view new_EMP_DEPT_VIEW(EMPNO, NAME, JOBNO, DEPTNO)
as select EMPNO, NAME, JOBNO, EMP.DEPTNO
from EMP inner join DEPT on EMP.DEPTNO=DEPT.DEPTNO;
```

22. Run the appropriate commands to demonstrate the type of faces that you created in **step 21.**

NW personnel

The tables contained in the personnel database should have the following structure and contents:

Columns Data type	
DEPT.DEPTNO, EMP.DEPTNO numeric(2)	
DNAME, JOB_DESCR varchar(24)	
LOC	char(23)
JOBCODE, JOBNO numeric(3)	
SAL, COMM	numeric(10,2)
EMPNO	numeric(4)
PROJECT.P_ID	int
PROJECT.P_NAME varchar(255)	

Table 1. Data types of tables EMP, JOB, DEPT

EMP

EMPNO	NAME	JOBNO	DEPTNO	COMM
10	CODD	100	50	
20	NAVATHE	200	50	450
30	ELMASRI	300	60	
40	DATE	100	50	

JOB

JOBCODE	JOB_DESCR	SAL
100	SALESMAN	2000
200	ANALYST	2000
300	DBA	3000

DEPT

DEPTNO	DNAME	LOC
50	SALES	ATHENS
60	ACCOUNTING	ATHENS
70	PAYROL	VOLOS

Figure 1. EMP, JOB, DEPT table data