

Practice 15: Using Views

Jesús Manuel Juárez Pasillas

November 23, 2021

1 Introduction

A view is an alternative to displaying data from multiple tables; it is like a virtual table that stores a query. This allows us to consult one or more tables, making complex queries so that they become easy to consult, it also allows us to restrict access to the data, among other things.

2 Development

2.1 Activity 1:

Read all the choices carefully because there might be more than one correct answer.

1. Which of these is a defining characteristic of a complex view, rather than a simple view?

A. Joining two tables

B. Naming the view's columns with column aliases

C. Restricting the selection of rows with a WHERE clause

D. Performing an aggregation

E. Restricting the projection by selecting only some of the table's columns

A complex view allows the union of two or more tables, while a simple view only allows one table.

2. Consider these three statements:
create view v1 as select department_id,department_name,last_name from departments join employees using (department_id);
select department_name,last_name from v1 where department_id=20;
select d.department_name,e.last_name from departments d, employees e where d.department_id=e.department_id and d.department_id=20;
The first query will be quicker than the second because

A. The view has already done the work of joining the tables.

- B. The view uses ISO standard join syntax, which is faster than the Oracle join syntax used in the second query.
- C. The view is precompiled, so the first query requires less dynamic compilation than the second query.
- D. **There is no reason for the first query to be quicker.**

Using views does not make queries faster as it does the same processing as a normal query.

3. Study this view creation statement:

```
create view dept30 as
select department_id,employee_id,last_name from employees
where department_id=30 with check option;
What might make the following statement fail?
update dept30 set department_id=10 where employee_id=114;
```

 - A. Unless specified otherwise, views will be created as WITH READ ONLY.
 - B. The view is too complex to allow DML operations.
 - C. **The WITH CHECK OPTION will reject any statement that changes the DEPARTMENT_ID.**
 - D. The statement will succeed.

The WITH CHECK OPTION option does not allow modifications to the fields that affect the records returned by the view.

4. There is a simple view SCOTT.DEPT_VIEW on the table SCOTT.DEPT. This insert fails with an error:

```
SQL> insert into dept_view values('SUPPORT','OXFORD');
insert into dept_view values('SUPPORT','OXFORD') * ERROR at line 1:
ORA-01400: cannot insert NULL into ("SCOTT"."DEPT"."DEPTNO")
What might be the problem?
```

 - A. The INSERT violates a constraint on the detail table.
 - B. **The INSERT violates a constraint on the view.**
 - C. The view was created as WITH READ ONLY.
 - D. The view was created as WITH CHECK OPTION.

A view restriction is violated, either because the view does not include all the non-null values and therefore does not allow inserting values, or because the view is a complex view which does not allow DML operations.

5. To add the number of columns selected by a view:
 - A. Add more columns to the underlying table.

B. Issue the alter view statement.

C. Use a correlated subquery in conjunction with the view.

D. Drop and re-create the view with references to select more columns.

To add columns that a view queries, it is necessary to replace the previous view.

6. The following statement is issued against the Oracle database. Which line will produce an error?

A. create view EMP_VIEW_01

B. as select E.EMPID, E.LASTNAME, E.FIRSTNAME, A.ADDRESS

C. from EMPLOYEE E, EMPL_ADDRESS A

D. where E.EMPID = A.EMPID

E. with check option;

F. This statement contains no errors.

The statement does not contain any errors since it correctly uses all parts of a view.

2.2 Activity 2:

Propose an answer to the following issues:

1. What is a simple view? How does it differ from a complex view? Which view allows the user to insert data into the view's underlying table? Explain.

- **A simple view is a logical representation of a table, it does not contain group functions and can perform DML operations, a complex view is a logical representation of one or more tables, it can contain group functions, and very few allow DML operations. Simple views always allow DML operations, complex views do not always.**

2. What is a complex view? What are the rules that determine when a complex view can be used to modify data in an underlying table? Explain.

- **A complex view is a logical representation of one or more tables, it can contain group functions, and very few allow DML operations. You cannot delete a row if the view contains Group Functions, a GROUP BY clause, or the DISTINCT command. You cannot modify data in the view if it contains any of the above conditions, columns defined by expressions, or the ROWNUM pseudo column. Data cannot be aggregated if the view contains any of the above conditions or any NOT NULL columns not included by the view (base table).**

3. How can constraints be created and enforced on views?
 - **With the WITH CHECK OPTION options (Specifies that only the rows accessible to the view can be inserted or updated) and with WITH READ ONLY (Ensures that no DML operation can be performed on this view).**
4. On what principle does a view constraint operate?
 - **The read-only principle or whether modifications can be made.**
5. What statement is used to alter the definition of a view?
 - **CREATE OR REPLACE VIEW (view name) AS...**
6. How are views dropped?
 - **DROP VIEW (name of the view to delete) ;.**
7. How can you create a view even if the table referenced does not exist?
 - **Using the FORCÉ clause**
8. What statement is used to recompile or revalidate an existing view definition?
 - **CREATE OR REPLACE VIEW.**
9. What is object dependency? About views and tables.
 - **It is when one or more views refer to one or more tables. The views depend on the tables.**

2.3 Activity 3:

This exercise must be performed in the local HR schema.

a) Problem 1.

1. The staff in the HR department wants to hide some of the data in the EMPLOYEES table. Create a view called EMPLOYEES_VU based on the employee numbers, employee last names, and department numbers from the EMPLOYEES table. The heading for the employee name should be EMPLOYEE.
 - **Create view Employees_Vu as
Select employee_id, last_name employee, department_id from
Employees;**
2. Confirm that the view works. Display the contents of the EMPLOYEES_VU view.

- **Select * from Employees_Vu;**

	EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
1	207	Perez	(null)
2	100	King	90
3	101	Kochhar	90
4	102	De Haan	90
5	103	Hunold	60
6	104	Ernst	60
7	105	Austin	60
8	106	Pataballa	60
9	107	Lorentz	60
10	108	Greenberg	100
11	109	Faviet	100
12	110	Chen	100

Figure 1: Confirm that the view works.

- Using your EMPLOYEES_VU view, write a query for the HR department to display all employee names and department numbers.

- **Select employee,department_id from Employees_Vu;**

	EMPLOYEE	DEPARTMENT_ID
1	Perez	(null)
2	King	90
3	Kochhar	90
4	De Haan	90
5	Hunold	60
6	Ernst	60
7	Austin	60
8	Pataballa	60
9	Lorentz	60
10	Greenberg	100
11	Faviet	100
12	Chen	100

Figure 2: Employee names and department numbers.

- Department 50 needs access to its employee data. Create a view named DEPT50 that contains the employee numbers, employee last names, and department numbers for all employees in department 50. You have been

asked to label the view columns EMPNO, EMPLOYEE, and DEPTNO. For security purposes, do not allow an employee to be reassigned to another department through the view.

- **Create view DEPT50 as**
Select employee_id empno, last_name employee, department_id deptno from Employees where department_id = 50 with check option;

5. Display the structure and contents of the DEPT50 view.

- **Describe Dept50;**

Nombre	¿Nulo?	Tipo
-----	-----	-----
EMPNO	NOT NULL	NUMBER(6)
EMPLOYEE	NOT NULL	VARCHAR2(25)
DEPTNO		NUMBER(4)

Figure 3: Structure of the DEPT50 view.

- **Select * from Dept50;**

	EMPNO	EMPLOYEE	DEPTNO
1	120	Weiss	50
2	121	Fripp	50
3	122	Kaufling	50
4	123	Vollman	50
5	124	Mourgos	50
6	125	Nayer	50
7	126	Mikkilineni	50
8	127	Landry	50
9	128	Markle	50
10	129	Bissot	50
11	130	Atkinson	50
12	131	Marlow	50

Figure 4: contents of the DEPT50 view.

6. Test your view. Attempt to reassign Matos to department 80. What is the result?

- **Update Dept50 set deptno=80 where employee='Matos';**

It does not update the data due to the **WITH CHECK OPTION** clause.

b) **Problem 2.**

1. Create views on the EMPLOYEES (emp_anon_v) and DEPARTMENT (dept_anon_v) tables that queries representative personal information (hire_date, job_id, salary, commission_pct, department_id; department_id, department_name, location_id) :
 - **Create view emp_anon_v as**
 Select hire_date, job_id, salary, commission_pct, department_id
 from Employees ;
 - **Create view dept_anon_v as**
 Select department_id, department_name, location_id from
 Departments;
2. Create a complex view that will join and aggregate (sum salary grouped by department's id and name [salaries], count the number of employees [staff]) the two simple views. Name the view dep_sum_v. Note that there is no reason not to have views of views.
 - **Create view dep_sum_v as**
 Select department_id, department_name, sum(salary) salaries,
 count(*) staff from Emp_anon_v join Dept_anon_v using
 (department_id) group by department_id, department_name;
3. Query the resulting table:
 - **Select * from Dep_sum_v;**

	DEPARTMENT_ID	DEPARTMENT_NAME	SALARIES	STAFF
1	100	Finance	51608	6
2	50	Shipping	156400	45
3	70	Public Relations	10000	1
4	30	Purchasing	24900	6
5	90	Executive	58000	3
6	10	Administration	4400	1
7	110	Accounting	20308	2
8	40	Human Resources	6500	1
9	20	Marketing	19000	2
10	60	IT	28800	5
11	80	Sales	304500	34

Figure 5: Query the resulting table.

c) **Problem 3.**

1. Insert a new department through the DEPT_ANON_V view and attempt to insert an employee through EMP_ANON_V:
 insert into DEPT_ANON_V values(99,'Temp Dept',1800);
 insert into EMP_ANON_V values(sysdate,'AC_MGR',10000,0,99);
 What is the result? Why?

- ```
1 fila insertadas.
```
- Error que empieza en la línea: 34 del comando :
- ```
insert into EMP_ANON_V values(sysdate,'AC_MGR',10000,0,99)
```
- Informe de error -
- ```
ORA-01400: cannot insert NULL into ("HR"."EMPLOYEES"."EMPLOYEE_ID")
```

Try to update through it:  
 update emp\_anon\_v set salary=salary\*1.1;  
 What is the result? Why?

- [illegible]

and find the same information from the `DEP_SUM_V` view, which is a much simpler query:

- [illegible]

d) **Problem 4.**

- 8



- **insert into departments1 values('IT','Office\_2');**
  - **insert into employees1 values(1,'Perez','Juan',1234,'Sales');**
  - **insert into employees1 values(2,'Felix','Maria',5678,'IT');**
  - **insert into employees1 values(3,'Hernandez','Pedro',9101,'Sales');**
2. Create a view called EmpDepSales that contains the surname and first name of the employees of the department "Sales" renaming those properties as "last\_name" and "first\_name". Use alias out of the subquery.
- **create view EmpDepSales**  
**(last\_name,first\_name) as**  
**Select surname,name from Employees1 where dep='Sales';**

| LAST_NAME   | FIRST_NAME |
|-------------|------------|
| 1 Perez     | Juan       |
| 2 Hernandez | Pedro      |

Figure 9: called EmpDepSales.

3. List the surname and first name of the employees of the department "Sales" alphabetically.
- **Select last\_name,first\_name from EmpDepSales order by last\_name,first\_name;**

| LAST_NAME   | FIRST_NAME |
|-------------|------------|
| 1 Hernandez | Pedro      |
| 2 Perez     | Juan       |

Figure 10: Surname and first name from EmpDepSales.

4. Alter the view EmpDepSales to also include the salary.
- **create or replace view EmpDepSales**  
**(last\_name,first\_name,salary) as**  
**Select surname,name,salary from Employees1 where dep='Sales';**

| LAST_NAME   | FIRST_NAME | SALARY |
|-------------|------------|--------|
| 1 Perez     | Juan       | 1234   |
| 2 Hernandez | Pedro      | 9101   |

Figure 11: Include the salary.

5. Create a view that displays the minimum, maximum and average salaries of the employees of each department.

- create view EmpSalary (maxsal,minsal,avgsal)as  
Select max(salary),min(salary),avg(salary) from Employees1  
group by dep;

|   | MAXSAL | MINSAL | AVGSAL |
|---|--------|--------|--------|
| 1 | 9101   | 1234   | 5167.5 |
| 2 | 5678   | 5678   | 5678   |

Figure 12: Minimum, maximum and average salaries.

6. Describe each view.

- Describe EmpDepSales;

| Nombre     | ¿Nulo? | Tipo       |
|------------|--------|------------|
| LAST_NAME  |        | CHAR(40)   |
| FIRST_NAME |        | CHAR(20)   |
| SALARY     |        | NUMBER(38) |

Figure 13: Describe EmpDepSales.

- Describe EmpSalary;

| Nombre | ¿Nulo? | Tipo   |
|--------|--------|--------|
| MAXSAL |        | NUMBER |
| MINSAL |        | NUMBER |
| AVGSAL |        | NUMBER |

Figure 14: Describe EmpSalary.

7. Drop each view created.

- Drop view EmpDepSales;
- Drop view EmpSalary;

e) **Problem 5.**

1. Connect to your database as user HR.
2. Create synonyms for the three views created in Exercise b, problem 2:
  - Create synonym emp for emp\_anon\_v;
  - Create synonym dept for dept\_anon\_v;
  - Create synonym dep\_sum for dep\_sum\_v;
3. Confirm that the synonyms are identical to the underlying object:

- Select \* from emp;

|    | ⚡ HIRE_DATE | ⚡ JOB_ID   | ⚡ SALARY | ⚡ COMMISSION_PCT | ⚡ DEPARTMENT_ID |
|----|-------------|------------|----------|------------------|-----------------|
| 1  | 10/09/07    | AD_PRES    | 8000     | (null)           | (null)          |
| 2  | 17/06/03    | AD_PRES    | 24000    | (null)           | 90              |
| 3  | 21/09/05    | AD_VP      | 17000    | (null)           | 90              |
| 4  | 13/01/01    | AD_VP      | 17000    | (null)           | 90              |
| 5  | 03/01/06    | IT_PROG    | 9000     | (null)           | 60              |
| 6  | 21/05/07    | IT_PROG    | 6000     | (null)           | 60              |
| 7  | 25/06/05    | IT_PROG    | 4800     | (null)           | 60              |
| 8  | 05/02/06    | IT_PROG    | 4800     | (null)           | 60              |
| 9  | 07/02/07    | IT_PROG    | 4200     | (null)           | 60              |
| 10 | 17/08/02    | FI_MGR     | 12008    | (null)           | 100             |
| 11 | 16/08/02    | FI_ACCOUNT | 9000     | (null)           | 100             |

Figure 15: Select \* from emp.

- Select \* from dept;

|    | ⚡ DEPARTMENT_ID | ⚡ DEPARTMENT_NAME | ⚡ LOCATION_ID |
|----|-----------------|-------------------|---------------|
| 1  | 10              | Administration    | 1700          |
| 2  | 20              | Marketing         | 1800          |
| 3  | 30              | Purchasing        | 1700          |
| 4  | 40              | Human Resources   | 2400          |
| 5  | 50              | Shipping          | 1500          |
| 6  | 60              | IT                | 1400          |
| 7  | 70              | Public Relations  | 2700          |
| 8  | 80              | Sales             | 2500          |
| 9  | 90              | Executive         | 1700          |
| 10 | 100             | Finance           | 1700          |
| 11 | 110             | Accounting        | 1700          |

Figure 16: Select \* from dept.

- **Select \* from dep\_sum;**

|    | DEPARTMENT_ID | DEPARTMENT_NAME  | SALARIES | STAFF |
|----|---------------|------------------|----------|-------|
| 1  | 100           | Finance          | 51608    | 6     |
| 2  | 50            | Shipping         | 156400   | 45    |
| 3  | 70            | Public Relations | 10000    | 1     |
| 4  | 30            | Purchasing       | 24900    | 6     |
| 5  | 90            | Executive        | 58000    | 3     |
| 6  | 10            | Administration   | 4400     | 1     |
| 7  | 110           | Accounting       | 20308    | 2     |
| 8  | 40            | Human Resources  | 6500     | 1     |
| 9  | 20            | Marketing        | 19000    | 2     |
| 10 | 60            | IT               | 28800    | 5     |
| 11 | 80            | Sales            | 304500   | 34    |

Figure 17: Select \* from dep\_sum.

4. Confirm that the synonyms work (even to the extent of producing the same errors) by running the statements in b and c against the synonyms instead of the views:
  - **insert into dept values(99,'Temp Dept',1800);**
  - **insert into emp values(sysdate,'AC\_MGR',10000,0,99);**
  - **update emp set salary=salary\*1.1;**
  - **rollback;**
5. Drop two of the views:
  - **Drop view emp\_anon\_v;**
  - **Drop view dept\_anon\_v;**
6. Query the complex view that is based on the dropped views:
  - **Select \* from dep\_sum;**

```
ORA-04063: view "HR.DEP_SUM_V" has errors
04063. 00000 - "%s has errors"
*Cause: Attempt to execute a stored procedure or use a view that has
 errors. For stored procedures, the problem could be syntax errors
 or references to other, non-existent procedures. For views,
 the problem could be a reference in the view's defining query to
 a non-existent table.
 Can also be a table which has references to non-existent or
 inaccessible types.
*Action: Fix the errors and/or create referenced objects as necessary.
Error en la línea: 103, columna: 15
```

Figure 18: Complex view that is based on the dropped views.

7. Attempt to recompile the broken view:
8. Drop the DEP\_SUM\_V view:
  - **Drop view dep\_sum\_v;**
9. Query the synonym for a dropped view:
  - **Select \* from dep\_sum;**

```
ORA-00980: synonym translation is no longer valid
00980. 00000 - "synonym translation is no longer valid"
*Cause: A synonym did not translate to a legal target object. This
 could happen for one of the following reasons:
 1. The target schema does not exist.
 2. The target object does not exist.
 3. The synonym specifies an incorrect database link.
 4. The synonym is not versioned but specifies a versioned
 target object.
*Action: Change the synonym definition so that the synonym points at
 a legal target object.
Error en la línea: 108, columna: 15
```

Figure 19: Query the synonym for a dropped view.

10. Recompile the broken synonym:
11. Tidy up by dropping the synonyms:
  - **Drop synonym emp;**
  - **Drop synonym dept;**
  - **Drop synonym dep\_sum;**

### 3 Pre-Assessment:

- Practices pre-Assessment for Database Systems Laboratory II

| Practice                                              | Pre-Assessment |
|-------------------------------------------------------|----------------|
| COMPLIES WITH THE REQUESTED FUNCTIONALITY             | X              |
| HAS THE CORRECT INDENTATION                           | X              |
| HAS AN EASY WAY TO ACCESS THE PROVIDED FILES          | X              |
| HAS A REPORT WITH IDC FORMAT                          | X              |
| REPORT INFORMATION IS FREE OF SPELLING ERRORS         | X              |
| DELIVERED IN TIME AND FORM                            | X              |
| IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED) | 100%           |

Table 1: Pre-Assessment.

### 4 Conclusion:

Views are very useful for complex queries, since it only has to be done once and only query the view that gets the data you want. In addition, they also work very well for us to restrict access to data by placing restrictions on the view so that they cannot modify or only this in read-only mode.