Autonomous University of Zacatecas

Academic Unit of Electrical Engineering

Software Engineering Academic Program

Group: 5B - Semester: 2022-5º

Practice Number: 15

Using Views

Date: 22/November/2022

Professor:

Aldonso Becerra Sánchez.

Student:

Cristian Omar Alvarado Rodríguez.

Contents

Using Views

November 22th, 2022

1 Introduction

A view is nothing more than a stored query in order to use it as many times as you want. A view does not contain data but the SELECT statement needed to create the view, that ensures that the data is consistent when using the data stored in the tables. As a result, views use very little disk space.

Views are used to:

• Make complex queries easier, as they allow you to split the query into multiple subqueries (each of which is simpler than the original).

• Provide tables with complete data, format results, or perform calculations on the original data

• Provide personalized and more understandable forms of data.

• Hide the intrinsic storage of the database and achieve a greater independence of the data with respect to the rest of the elements of the database.

• Restrict access to original data.

• Be used as data cursors in procedural languages (such as PL/SQL)

There are two types of views:

• **Simple:** They form a single table and do not contain grouping functions. Their advantage is that they always allow you to perform DML operations on them. Characteristic: - They can only refer to one table in the database.

- Cannot contain functions

- They cannot contain any grouping clause

- They admit to perform DML operations on them (they are therefore used to modify the original table)

• **Complex:** They get data from multiple tables, can use grouping and other functions. They do not always allow DML operations.

DML on views:

DML statements (INSERT, UPDATE, or DELETE) can be executed on plain views.

- For complex views, rows can be deleted if the view:

- Does not contain grouping functions (like SUM or AVG).

- The GROUP BY clause is not used.

- The DISTINCT clause is not used.

- ROWNUM is not used

In addition, the data cannot be modified if there are columns that use expressions or functions (no matter how simple they were).

To add data (INSERT statement) it is also necessary to comply that all the fields with NOT NUL type restrictions of the tables are included.

In any case, the WITH READ ONLY clause commented above prohibits any DML statement on the view.

2 Practice objective

Use SQL SELECT statements for retrieving data from several sources using different operations and Views.

3 Developing

Activity 1: Read all the choices carefully because there might be more than one correct answer. Choose all the correct answers for each question.

Explain the reason for your answer.

CREATE SIMPLE AND COMPLEX VIEWS.

1. Which of these is a defining characteristic of a complex view, rather than a simple view?
(Choose one or more correct answers.)

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.

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 (choose the best answer):

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.

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? (Choose the best answer.) 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.

RETRIEVE DATA FROM VIEWS.

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? (Choose the best answer.)

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.

5. To add the number of columns selected by a view: (Choose the best answer.)

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.

6. The following statement is issued against the Oracle database. Which line will produce an error? (Choose the best answer.)

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.

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.

R = A simple view: They form a single table and do not contain grouping functions. Their advantage is that they always allow DML operations to be performed on them.
DML statements (INSERT, UPDATE, or DELETE) can be executed on plain views.

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.

R = Complex View: They obtain data from several tables, they can use grouping functions and of any other type. They do not always allow DML operations.

In complex views, rows can be deleted if the view:

Does not contain grouping functions (like SUM or AVG).

The GROUP BY clause is not used.

The DISTINCT clause is not used.

ROWNUM is not used.

In addition, the data cannot be modified if there are columns that use expressions or functions (no matter how simple they were).

3. How can constraints be created and enforced on views?

R = WITH CHECK OPTION. Makes that only the rows that are shown in the view can be added (INSERT) or modified (UPDATE). The constraint that follows this section is the name given to this CHECK OPTION type constraint.

WITH READ ONLY. Makes the view read-only. Allows you to record a name for this restriction.

4. On what principle does a view constraint operate?

R = Constraints operate at the view level and not at the table level.

5. What statement is used to alter the definition of a view?

R = OR REPLACE. If the view already existed, it changes it to the current one (in this way we can modify a previously created view).

6. How are views dropped?

R = To do this, the DROP VIEW command is used.

7. How can you create a view even if the table referenced does not exist?

R = The FORCE statement creates the view even though the data in the SELECT query does not exist

8. What statement is used to recompile or revalidate an existing view definition?

R = OR REPLACE. If the view already existed, it changes it to the current one (in this way we can modify a previously created view).

9. What is object dependency? About views and tables.

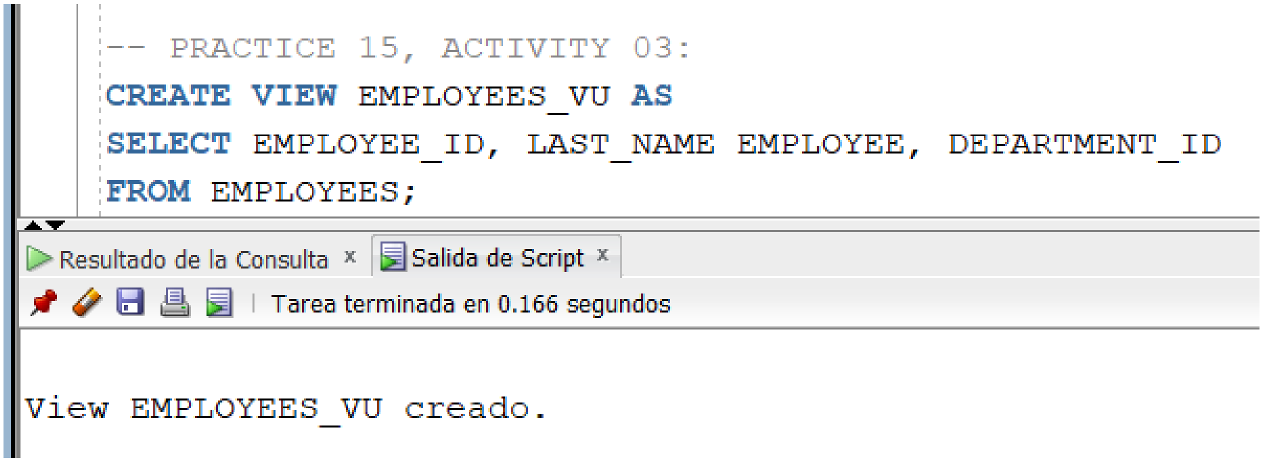
R =

Activity 3: This exercise must be performed in the local HR schema.

a) Problem 1:

In this exercise, you will see the effect of the set operators.

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, **see figure 1**.

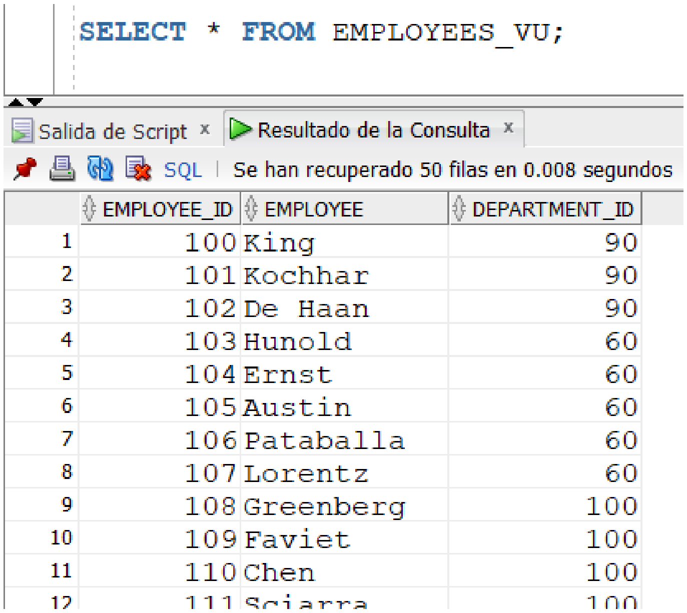


```
-- PRACTICE 15, ACTIVITY 03:
CREATE VIEW EMPLOYEES_VU AS
SELECT EMPLOYEE_ID, LAST_NAME EMPLOYEE, DEPARTMENT_ID
FROM EMPLOYEES;
```

```
Tarea terminada en 0.166 segundos

View EMPLOYEES_VU creado.
```

Figure 1: *Using and creating views in the HR schema.*

2. Confirm that the view works. Display the contents of the EMPLOYEES_VU view, **see figure 2**.



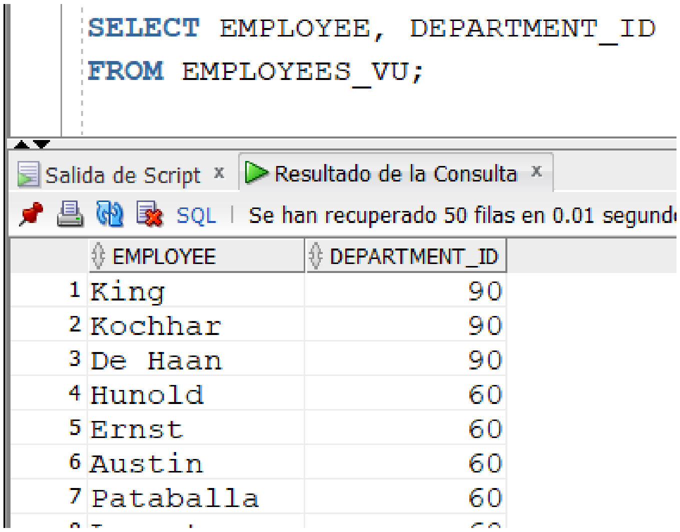
```sql
SELECT * FROM EMPLOYEES_VU;
```

Se han recuperado 50 filas en 0.008 segundos

	EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
1	100	King	90
2	101	Kochhar	90
3	102	De Haan	90
4	103	Hunold	60
5	104	Ernst	60
6	105	Austin	60
7	106	Pataballa	60
8	107	Lorentz	60
9	108	Greenberg	100
10	109	Faviet	100
11	110	Chen	100
12	111	Sciarra	100

Figure 2: *Using and creating views in the HR schema.*

3. Using your EMPLOYEES_VU view, write a query for the HR department to display all employee names and department numbers, **see figure 3**.



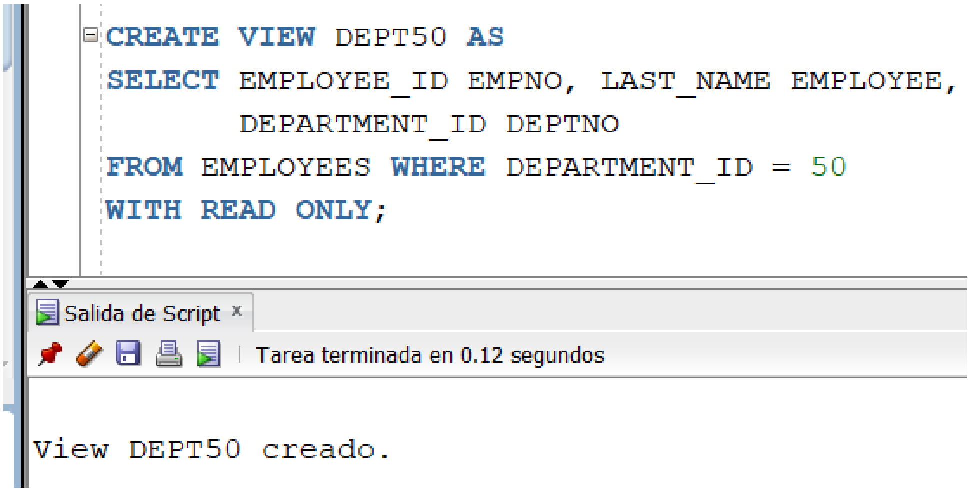
```sql
SELECT EMPLOYEE, DEPARTMENT_ID
FROM EMPLOYEES_VU;
```

Se han recuperado 50 filas en 0.01 segund

	EMPLOYEE	DEPARTMENT_ID
1	King	90
2	Kochhar	90
3	De Haan	90
4	Hunold	60
5	Ernst	60
6	Austin	60
7	Pataballa	60

Figure 3: *Using and creating views in the HR schema.*

4. 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, **see figure 4**.

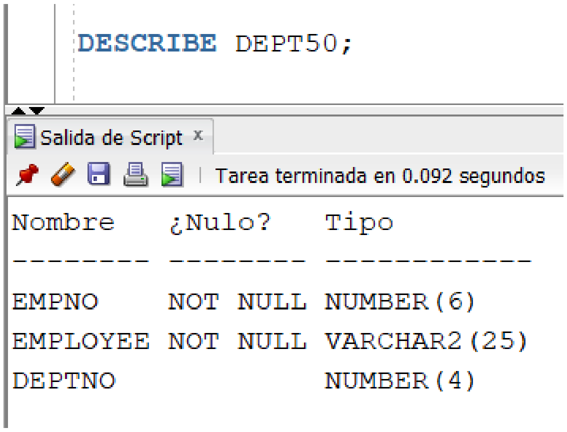


```
CREATE VIEW DEPT50 AS
SELECT EMPLOYEE_ID EMPNO, LAST_NAME EMPLOYEE,
       DEPARTMENT_ID DEPTNO
FROM EMPLOYEES WHERE DEPARTMENT_ID = 50
WITH READ ONLY;
```

```
View DEPT50 creado.
```

Figure 4: *Using and creating views in the HR schema.*

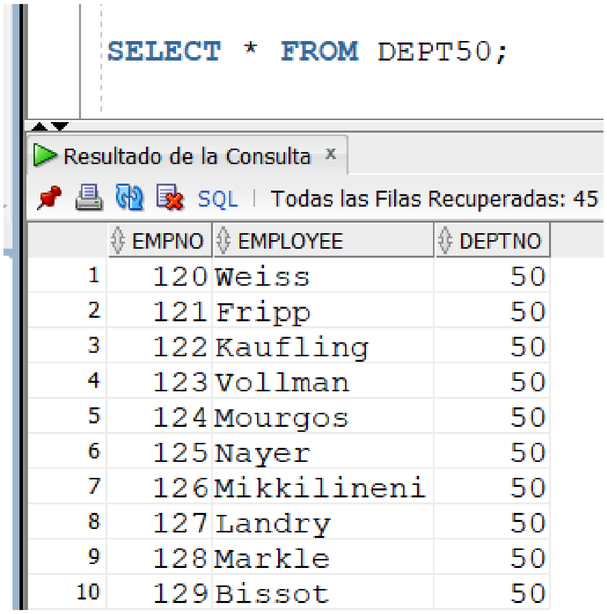
5. Display the structure and contents of the DEPT50 view, **see figure 5 and 6**.



```
DESCRIBE DEPT50;
```

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

Figure 5: *Using and creating views in the HR schema.*



```sql
SELECT * FROM DEPT50;
```

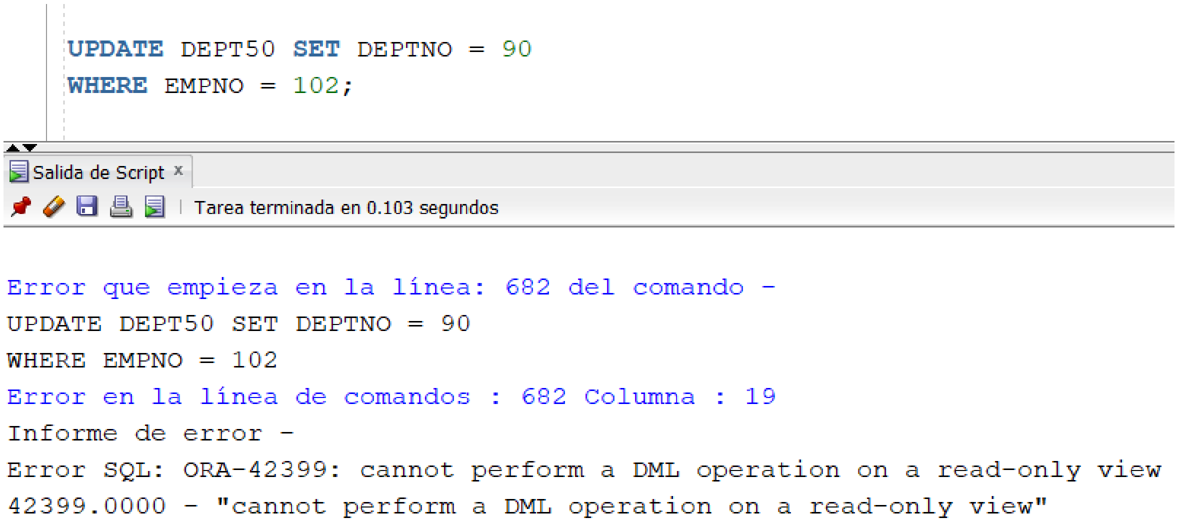
Todas las Filas Recuperadas: 45

	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

Figure 6: *Using and creating views in the HR schema.*

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

Explanation: In this case, it will flag an error since the view has a WITH READ ONLY read-only restriction, which means that DML operations cannot be executed on the view, **see figure 7**.



```sql
UPDATE DEPT50 SET DEPTNO = 90
WHERE EMPNO = 102;
```

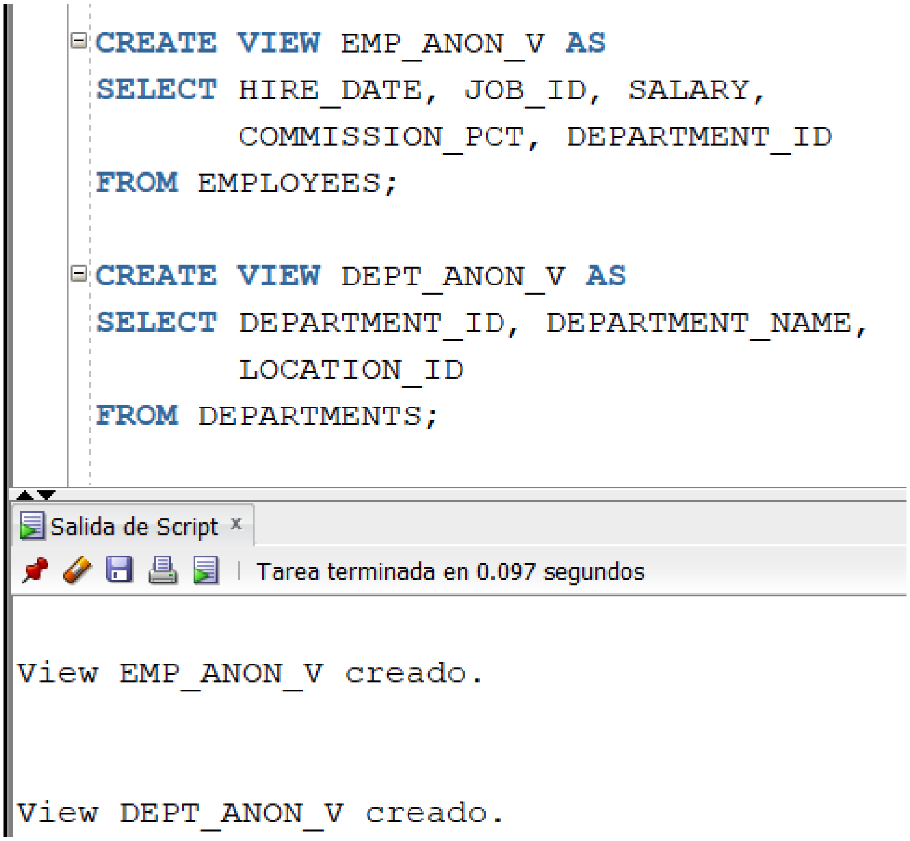
Tarea terminada en 0.103 segundos

```
Error que empieza en la línea: 682 del comando -
UPDATE DEPT50 SET DEPTNO = 90
WHERE EMPNO = 102
Error en la línea de comandos : 682 Columna : 19
Informe de error -
Error SQL: ORA-42399: cannot perform a DML operation on a read-only view
42399.0000 - "cannot perform a DML operation on a read-only view"
```

Figure 7: *Using and creating views in the HR schema.*

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, locarion_id), **see figure 8**.



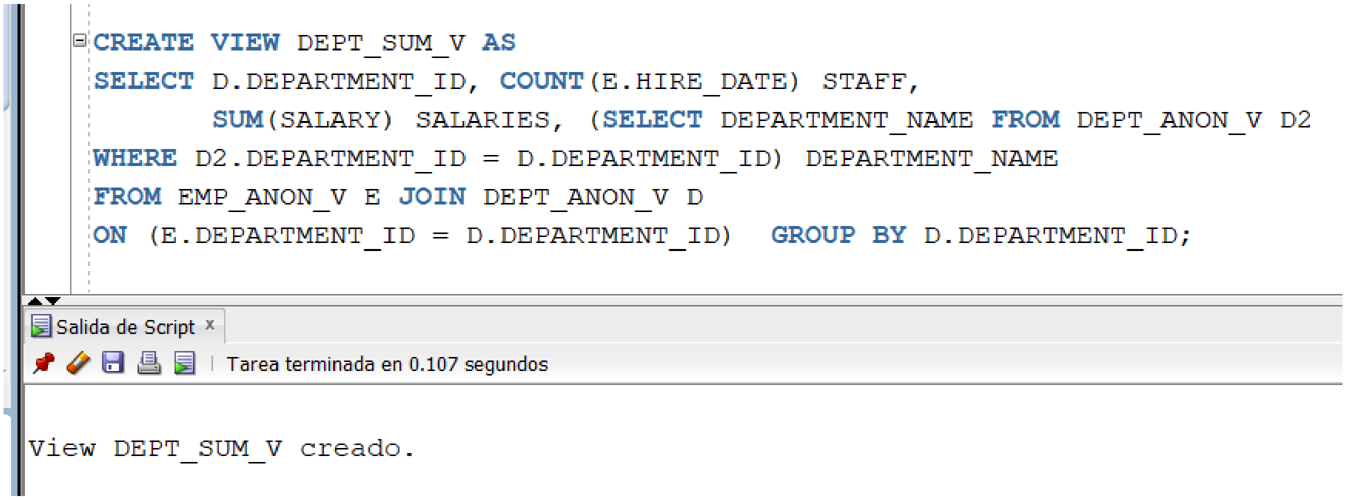
```
CREATE VIEW EMP_ANON_V AS
SELECT HIRE_DATE, JOB_ID, SALARY,
       COMMISSION_PCT, DEPARTMENT_ID
FROM EMPLOYEES;

CREATE VIEW DEPT_ANON_V AS
SELECT DEPARTMENT_ID, DEPARTMENT_NAME,
       LOCATION_ID
FROM DEPARTMENTS;
```

Tarea terminada en 0.097 segundos

```
View EMP_ANON_V creado.


View DEPT_ANON_V creado.
```

Figure 8: *Using and creating views in the HR schema.*

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, **see figure 9**.



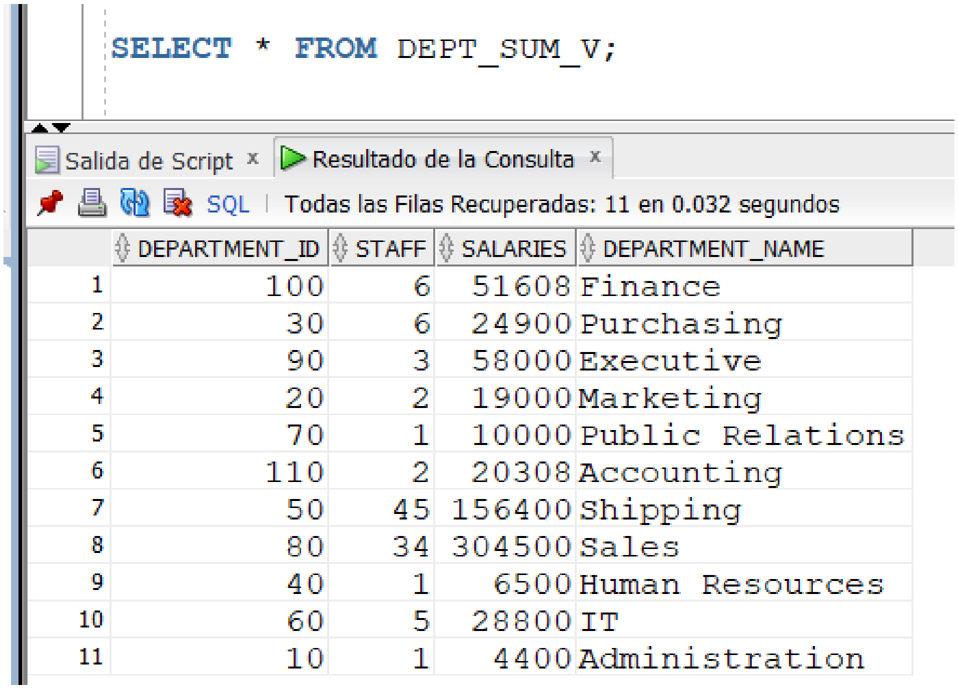
```
CREATE VIEW DEPT_SUM_V AS
SELECT D.DEPARTMENT_ID, COUNT(E.HIRE_DATE) STAFF,
       SUM(SALARY) SALARIES, (SELECT DEPARTMENT_NAME FROM DEPT_ANON_V D2
WHERE D2.DEPARTMENT_ID = D.DEPARTMENT_ID) DEPARTMENT_NAME
FROM EMP_ANON_V E JOIN DEPT_ANON_V D
ON (E.DEPARTMENT_ID = D.DEPARTMENT_ID)  GROUP BY D.DEPARTMENT_ID;
```

Tarea terminada en 0.107 segundos

```
View DEPT_SUM_V creado.
```

Figure 9: *Using and creating views in the HR schema.*

3. Query the resulting table, **see figure 10**.



```
SELECT * FROM DEPT_SUM_V;
```

Todas las Filas Recuperadas: 11 en 0.032 segundos

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

Figure 10: *Using and creating views in the HR schema.*

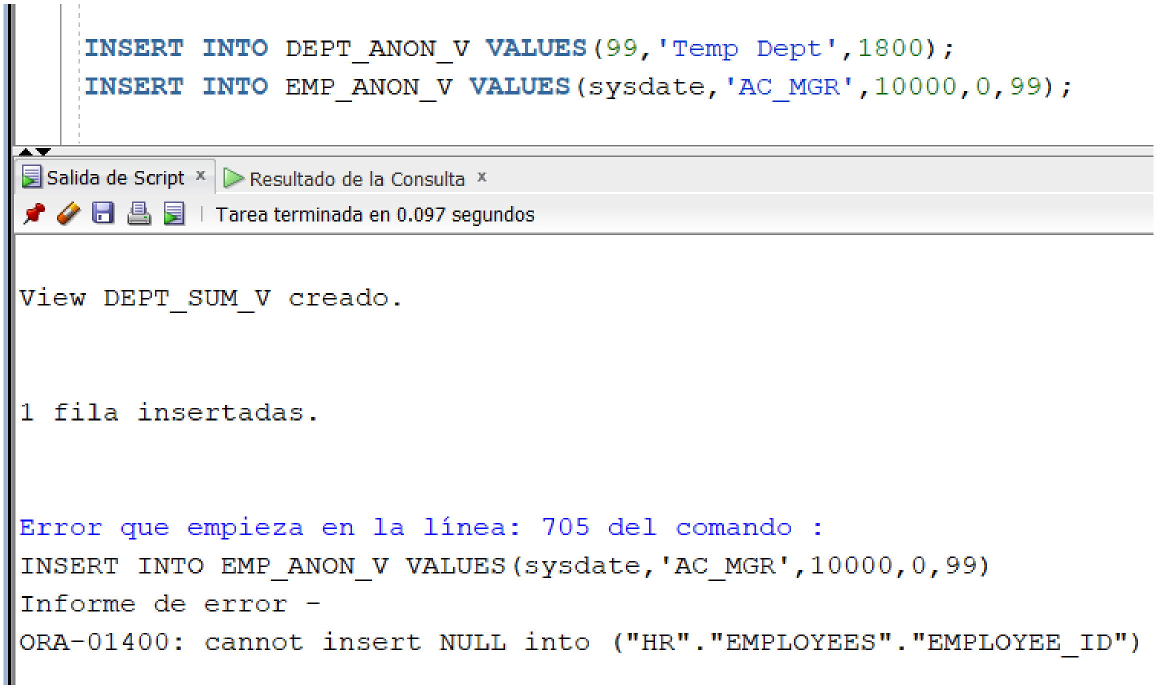
4. Execute these statements and show results.

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?

Explanation: I could insert a new row into the DEPT_ANON_V view with no problem but I couldn't do the same with the EMP_ANON_V view. These because the view does not receive the employee_id, and if the view does not receive it, it can't add new values because we cannot insert null value in the ID_EMPLOYEE column. **see figure 11**.



```
INSERT INTO DEPT_ANON_V VALUES(99,'Temp Dept',1800);
INSERT INTO EMP_ANON_V VALUES(sysdate,'AC_MGR',10000,0,99);
```

```
View DEPT_SUM_V creado.


1 fila insertadas.


Error que empieza en la línea: 705 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")
```

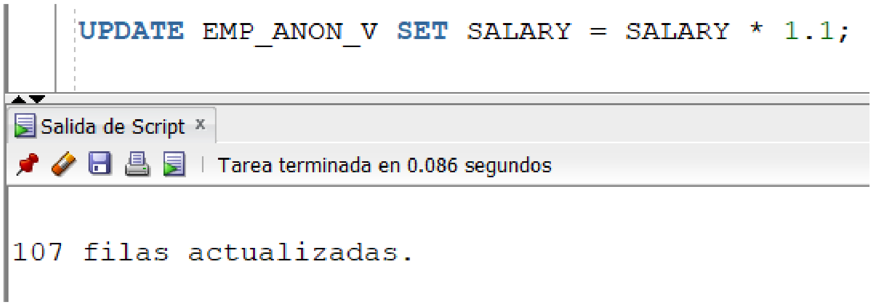
Figure 11: *Using and creating views in the HR schema.*

Try to update through it:

update emp_anon_v set salary=salary*1.1;

What is the result? Why?

Explanation: 107 values of the salary column were updated because there was no permission to prevent it, **see figure 12**.

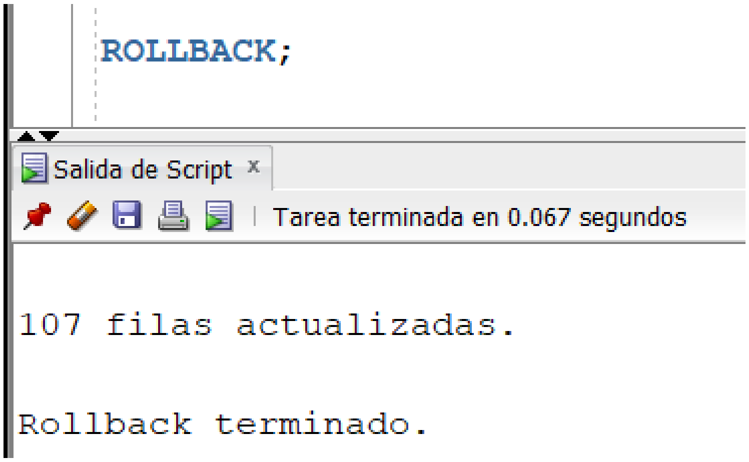


```
UPDATE EMP_ANON_V SET SALARY = SALARY * 1.1;
```

```
107 filas actualizadas.
```

Figure 12: *Using and creating views in the HR schema.*

Then roll back the changes:

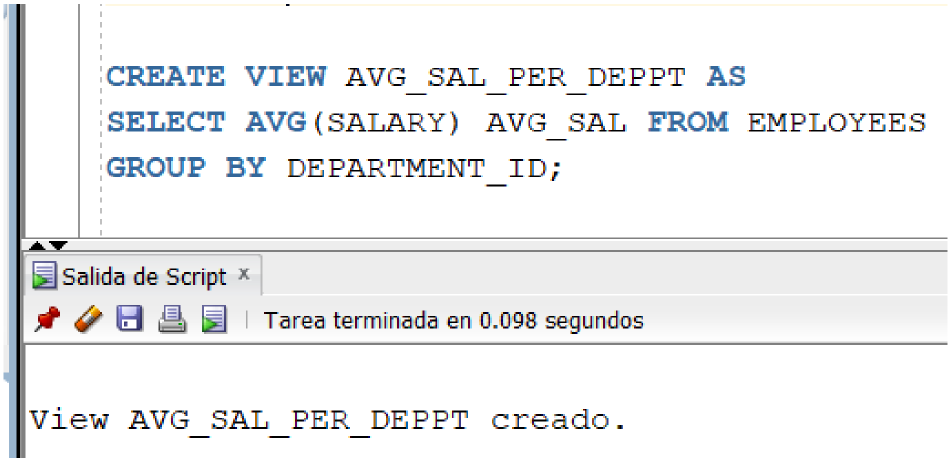


```
ROLLBACK;
```

```
107 filas actualizadas.

Rollback terminado.
```

Figure 13: *Using and creating views in the HR schema.*

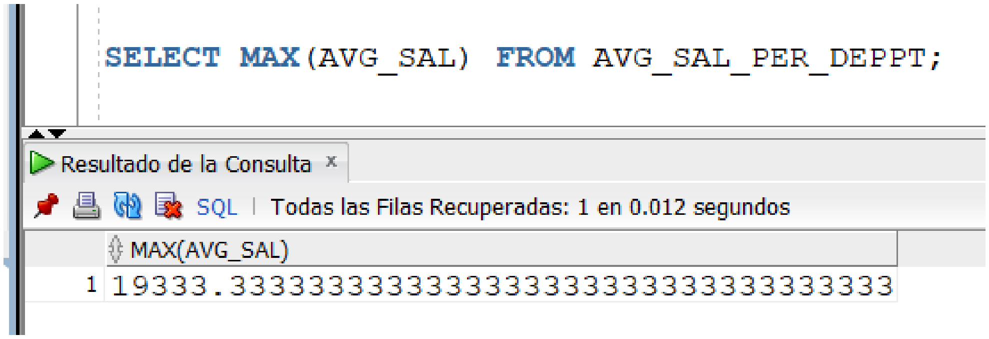
2. Find out the salary of the department with the highest average salary, by querying the EMPLOYEES table (use a subquery in the FROM clause to extract the average salary from employees): **see figure 14 and 15**.



```
CREATE VIEW AVG_SAL_PER_DEPPT AS
SELECT AVG(SALARY) AVG_SAL FROM EMPLOYEES
GROUP BY DEPARTMENT_ID;
```

```
Tarea terminada en 0.098 segundos

View AVG_SAL_PER_DEPPT creado.
```

Figure 14: *Using and creating views in the HR schema.*

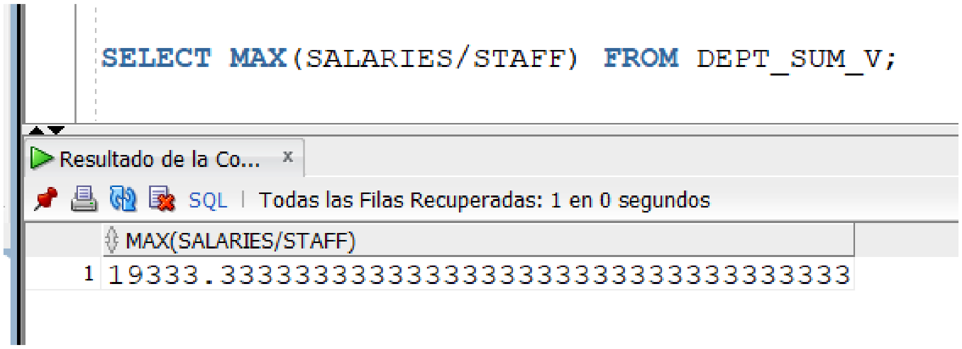


```
SELECT MAX(AVG_SAL) FROM AVG_SAL_PER_DEPPT;
```

```
Todas las Filas Recuperadas: 1 en 0.012 segundos

   MAX(AVG_SAL)
1  19333.3333333333333333333333333333333
```

Figure 15: *Using and creating views in the HR schema.*

and find the same information from the DEP_SUM_V view, which is a much simpler query: **see figure 16**.



```
SELECT MAX(SALARIES/STAFF) FROM DEPT_SUM_V;
```

Todas las Filas Recuperadas: 1 en 0 segundos

```
  MAX(SALARIES/STAFF)
1 19333.3333333333333333333333333333333333
```

Figure 16: *Using and creating views in the HR schema.*

3. Execute these statements and show results.

d) Problem 4:

From the following tables:

CREATE TABLE departments1 (

name CHAR(20) PRIMARY KEY,

office CHAR(50));

CREATE TABLE employees1 (

id SMALLINT PRIMARY KEY,

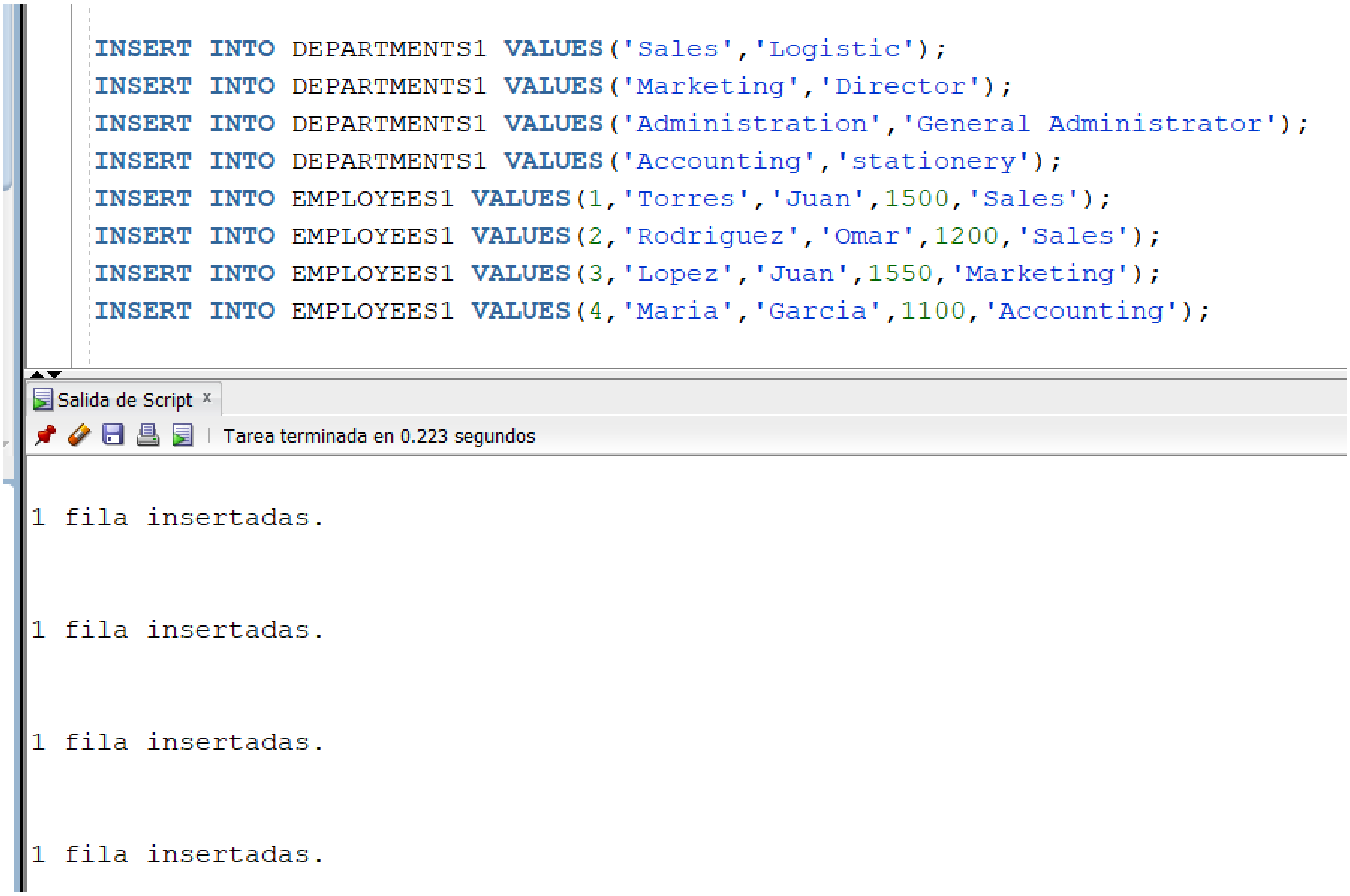
surname CHAR(40),

name CHAR(20),

salary INTEGER,

dep CHAR(20) REFERENCES departments1);

1. Insert some data in both tables, **see figure 17**.



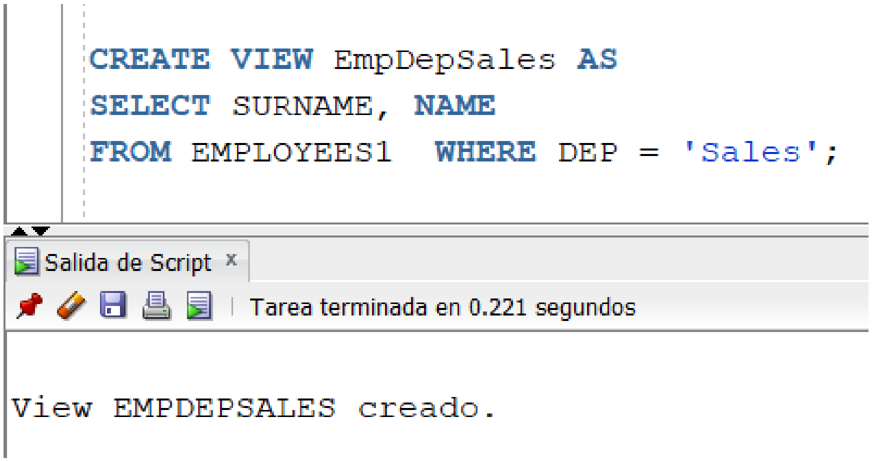
```
INSERT INTO DEPARTMENTS1 VALUES('Sales','Logistic');
INSERT INTO DEPARTMENTS1 VALUES('Marketing','Director');
INSERT INTO DEPARTMENTS1 VALUES('Administration','General Administrator');
INSERT INTO DEPARTMENTS1 VALUES('Accounting','stationery');
INSERT INTO EMPLOYEES1 VALUES(1,'Torres','Juan',1500,'Sales');
INSERT INTO EMPLOYEES1 VALUES(2,'Rodriguez','Omar',1200,'Sales');
INSERT INTO EMPLOYEES1 VALUES(3,'Lopez','Juan',1550,'Marketing');
INSERT INTO EMPLOYEES1 VALUES(4,'Maria','Garcia',1100,'Accounting');
```

Tarea terminada en 0.223 segundos

```
1 fila insertadas.


1 fila insertadas.


1 fila insertadas.


1 fila insertadas.
```

Figure 17: *Using and creating views in the HR schema.*

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, **see figure 18**.

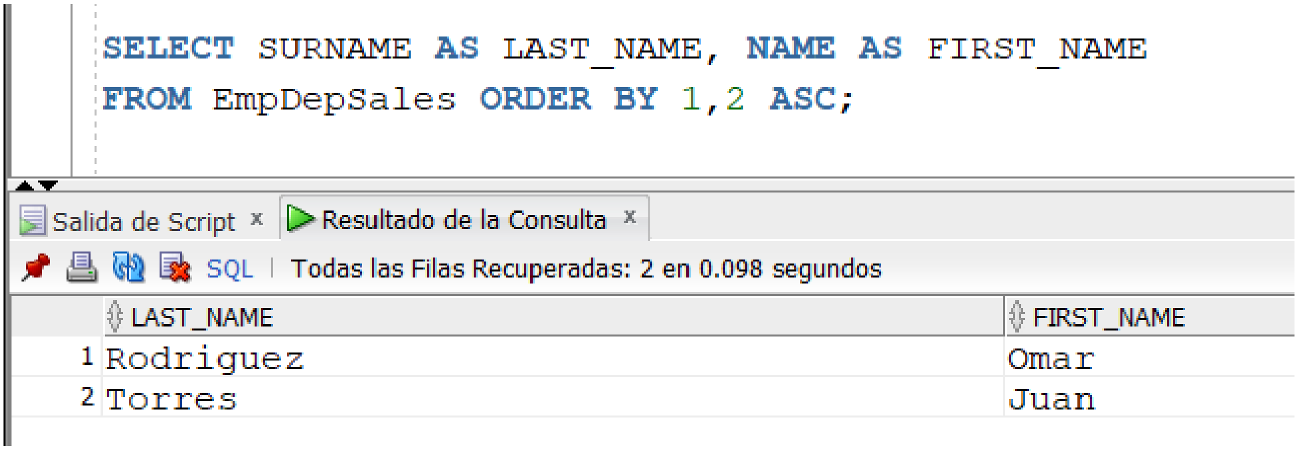


```sql
CREATE VIEW EmpDepSales AS
SELECT SURNAME, NAME
FROM EMPLOYEES1  WHERE DEP = 'Sales';
```

```
View EMPDEPSALES creado.
```

Figure 18: *Using and creating views in the HR schema.*

3. List the surname and first name of the employees of the department "Sales" alphabetically, **see figure 19**.

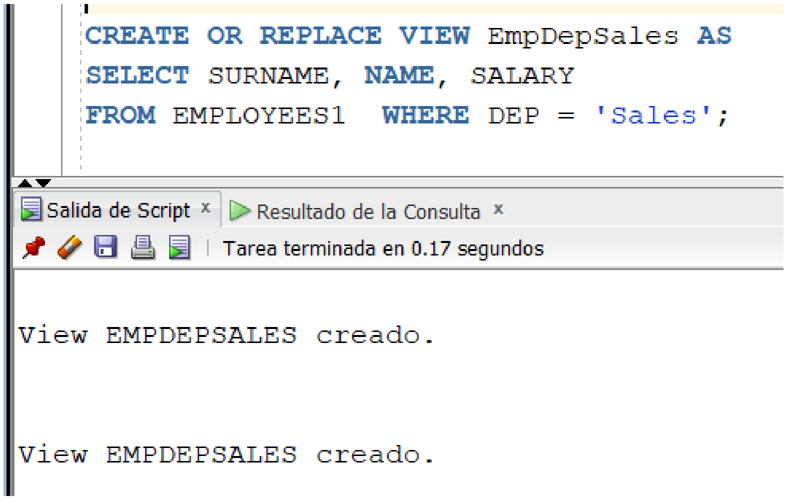


```sql
SELECT SURNAME AS LAST_NAME, NAME AS FIRST_NAME
FROM EmpDepSales ORDER BY 1,2 ASC;
```

	LAST_NAME	FIRST_NAME
1	Rodriguez	Omar
2	Torres	Juan

Figure 19: *Using and creating views in the HR schema.*

4. Alter the view EmpDepSales to also include the salary, **see figure 20**.



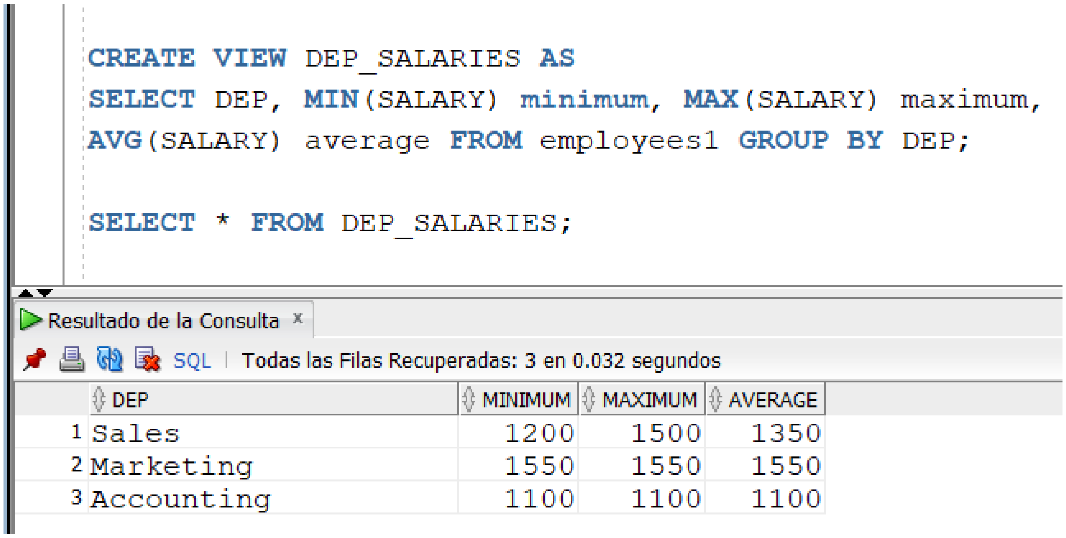
```
CREATE OR REPLACE VIEW EmpDepSales AS
SELECT SURNAME, NAME, SALARY
FROM EMPLOYEES1  WHERE DEP = 'Sales';
```

Tarea terminada en 0.17 segundos

```
View EMPDEPSALES creado.


View EMPDEPSALES creado.
```

Figure 20: *Using and creating views in the HR schema.*

5. Create a view that displays the minimum, maximum and average salaries of the employees of each department, **see figure 21**.



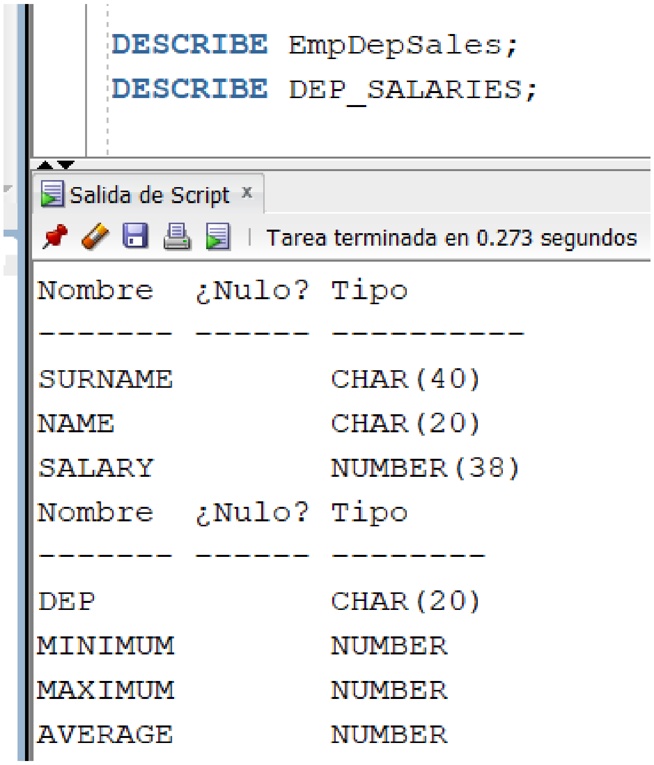
```
CREATE VIEW DEP_SALARIES AS
SELECT DEP, MIN(SALARY) minimum, MAX(SALARY) maximum,
AVG(SALARY) average FROM employees1 GROUP BY DEP;

SELECT * FROM DEP_SALARIES;
```

Todas las Filas Recuperadas: 3 en 0.032 segundos

	DEP	MINIMUM	MAXIMUM	AVERAGE
1	Sales	1200	1500	1350
2	Marketing	1550	1550	1550
3	Accounting	1100	1100	1100

Figure 21: *Using and creating views in the HR schema.*

6. Describe each view, **see figure 22**.



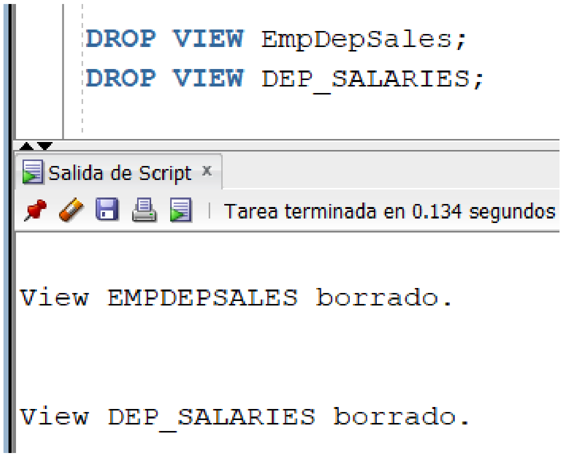
```
DESCRIBE EmpDepSales;
DESCRIBE DEP_SALARIES;
```

Tarea terminada en 0.273 segundos

```
Nombre   ¿Nulo? Tipo
-------- ------ ----------
SURNAME         CHAR(40)
NAME            CHAR(20)
SALARY          NUMBER(38)
Nombre   ¿Nulo? Tipo
-------- ------ --------
DEP             CHAR(20)
MINIMUM         NUMBER
MAXIMUM         NUMBER
AVERAGE         NUMBER
```

Figure 22: *Using and creating views in the HR schema.*

7. Drop each view created, **see figure 23**.



```
DROP VIEW EmpDepSales;
DROP VIEW DEP_SALARIES;
```

Tarea terminada en 0.134 segundos

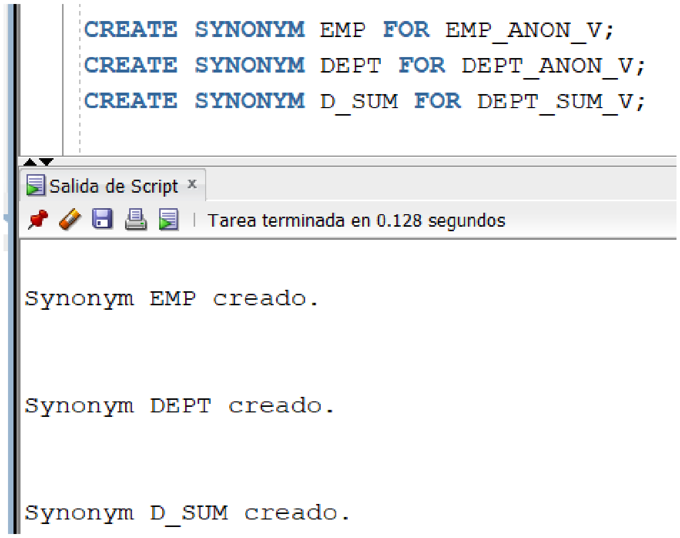
```
View EMPDEPSALES borrado.


View DEP_SALARIES borrado.
```

Figure 23: *Using and creating views in the HR schema.*

e) Problem 5:

1. Connect to your database as user HR.

2. Create synonyms for the three views created in Exercise b, problem 2: **see figure 24**.



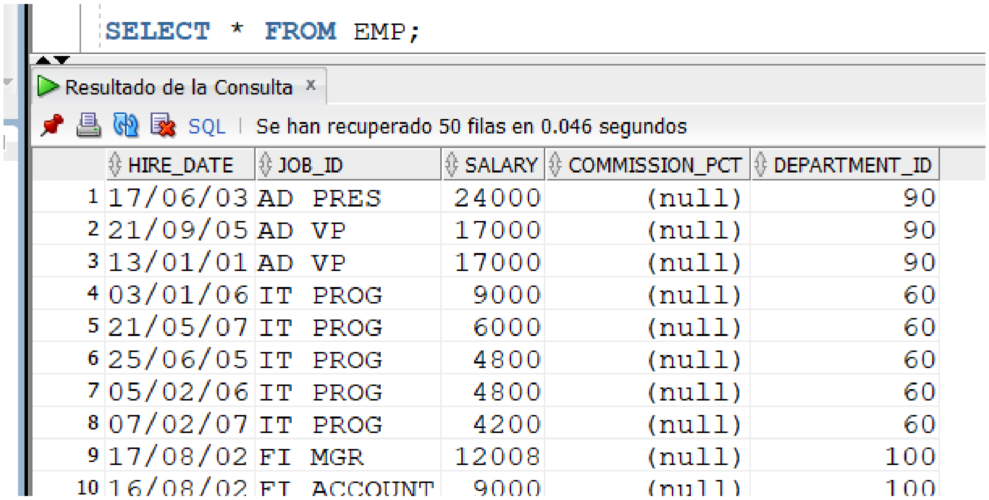
```
CREATE SYNONYM EMP FOR EMP_ANON_V;
CREATE SYNONYM DEPT FOR DEPT_ANON_V;
CREATE SYNONYM D_SUM FOR DEPT_SUM_V;
```

Tarea terminada en 0.128 segundos

```
Synonym EMP creado.


Synonym DEPT creado.


Synonym D_SUM creado.
```

Figure 24: *Using and creating views in the HR schema.*

3. Confirm that the synonyms are identical to the underlying object: **see figure 25,26 and 27**.

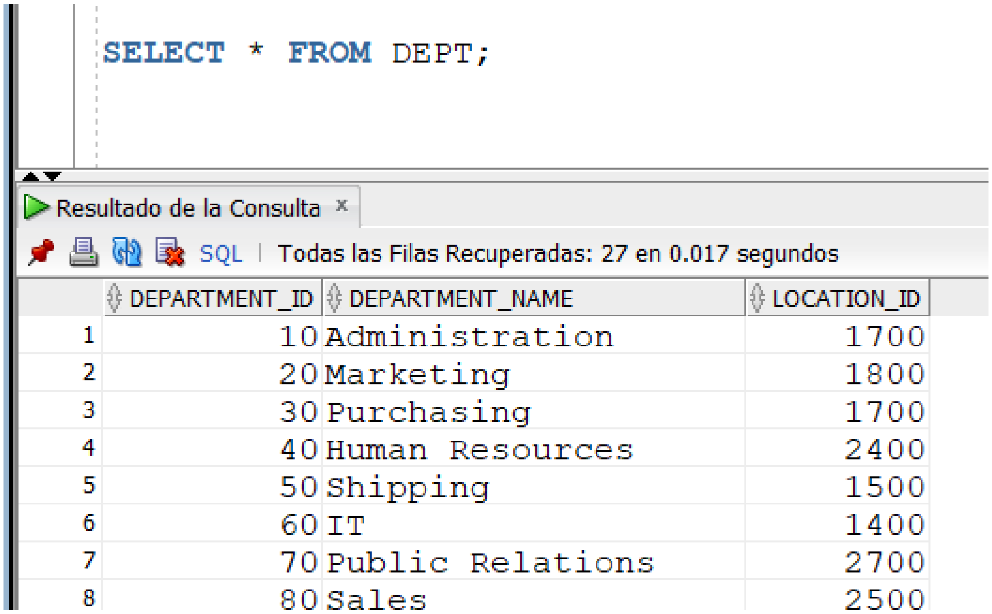


```
SELECT * FROM EMP;
```

Se han recuperado 50 filas en 0.046 segundos

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

Figure 25: *Using and creating views in the HR schema.*



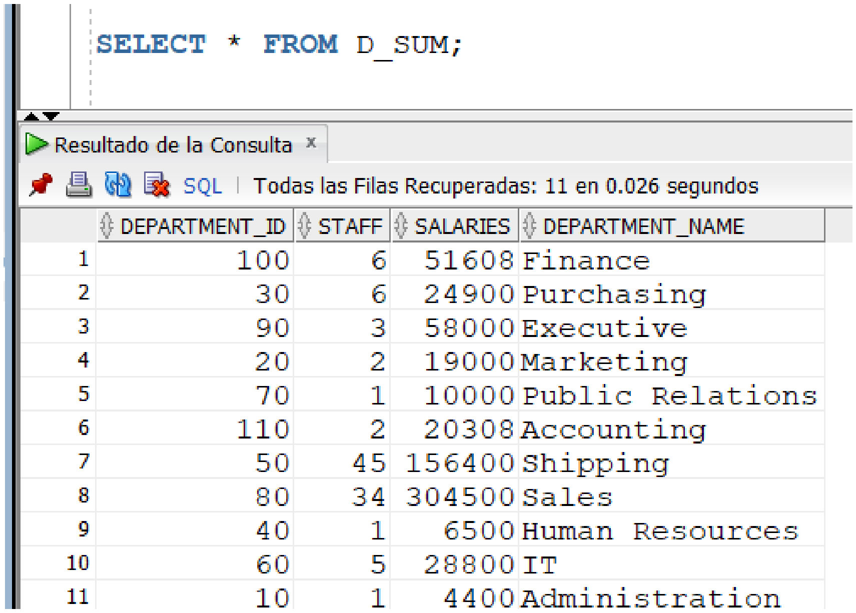
```sql
SELECT * FROM DEPT;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 27 en 0.017 segundos

	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

Figure 26: *Using and creating views in the HR schema.*



```sql
SELECT * FROM D_SUM;
```

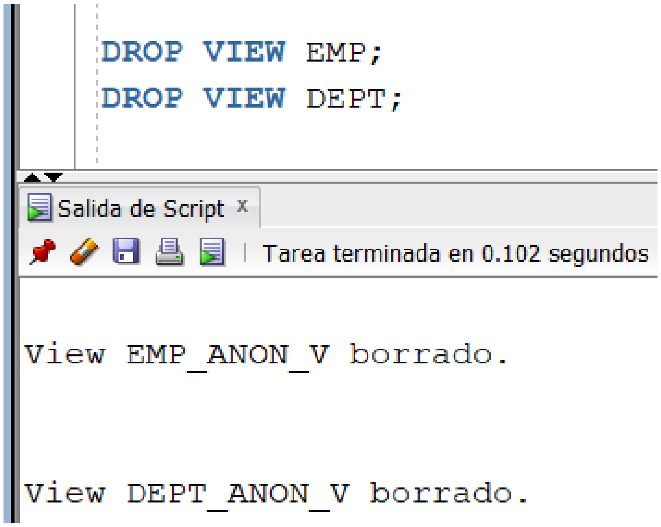
Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 11 en 0.026 segundos

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

Figure 27: *Using and creating views in the HR schema.*

4. Drop two of the views, **see figure 28**.

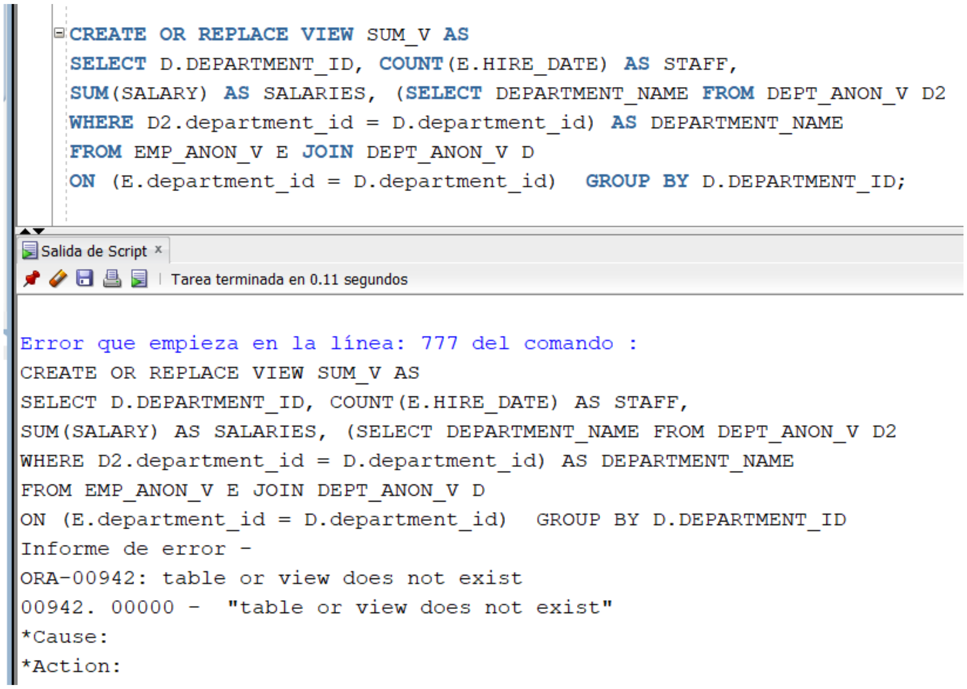


```
DROP VIEW EMP;
DROP VIEW DEPT;
```

```
View EMP_ANON_V borrado.


View DEPT_ANON_V borrado.
```

Figure 28: *Using and creating views in the HR schema.*

5. Query the complex view that is based on the dropped views, **see figure 29**.

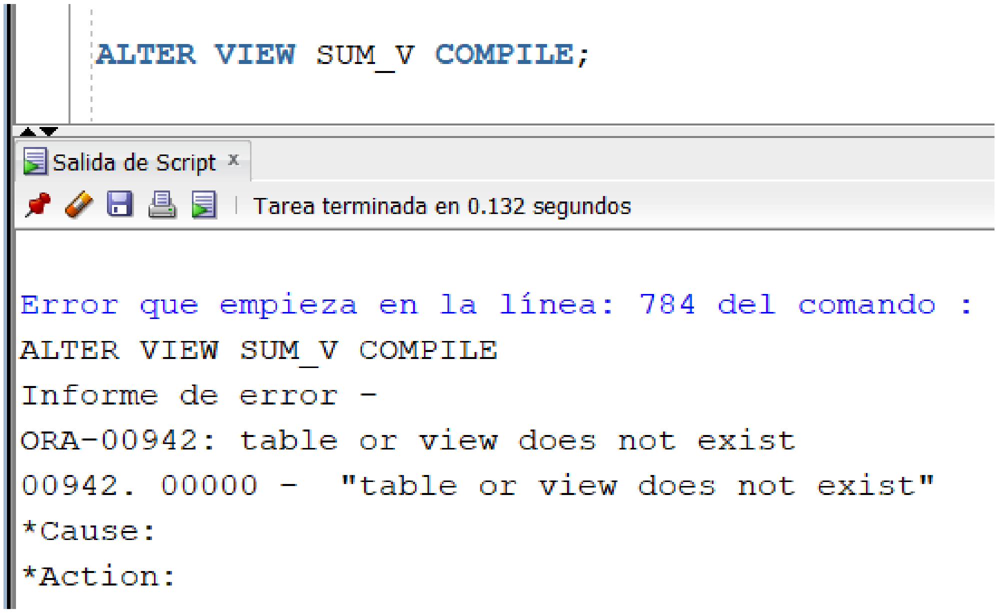


```
CREATE OR REPLACE VIEW SUM_V AS
SELECT D.DEPARTMENT_ID, COUNT(E.HIRE_DATE) AS STAFF,
SUM(SALARY) AS SALARIES, (SELECT DEPARTMENT_NAME FROM DEPT_ANON_V D2
WHERE D2.department_id = D.department_id) AS DEPARTMENT_NAME
FROM EMP_ANON_V E JOIN DEPT_ANON_V D
ON (E.department_id = D.department_id)  GROUP BY D.DEPARTMENT_ID;
```

```
Error que empieza en la línea: 777 del comando :
CREATE OR REPLACE VIEW SUM_V AS
SELECT D.DEPARTMENT_ID, COUNT(E.HIRE_DATE) AS STAFF,
SUM(SALARY) AS SALARIES, (SELECT DEPARTMENT_NAME FROM DEPT_ANON_V D2
WHERE D2.department_id = D.department_id) AS DEPARTMENT_NAME
FROM EMP_ANON_V E JOIN DEPT_ANON_V D
ON (E.department_id = D.department_id)  GROUP BY D.DEPARTMENT_ID
Informe de error -
ORA-00942: table or view does not exist
00942. 00000 -  "table or view does not exist"
*Cause:
*Action:
```

Figure 29: *Using and creating views in the HR schema.*

6. Attempt to recompile the broken view, **see figure 30**.

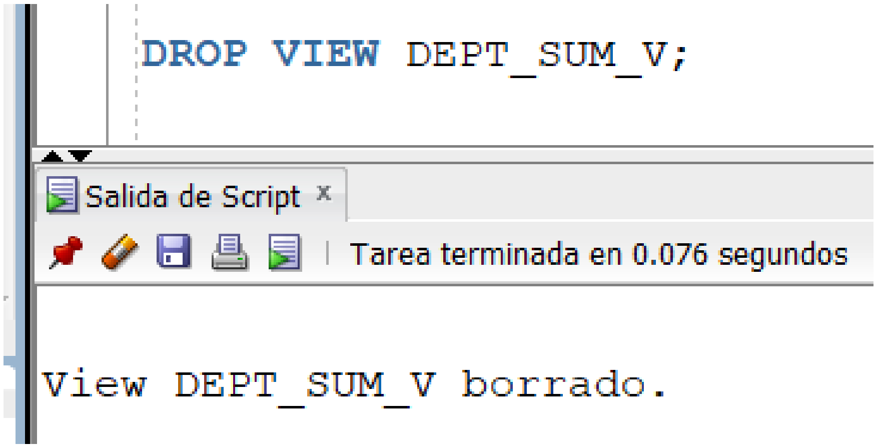


```
ALTER VIEW SUM_V COMPILE;
```

```
Error que empieza en la línea: 784 del comando :
ALTER VIEW SUM_V COMPILE
Informe de error -
ORA-00942: table or view does not exist
00942. 00000 -  "table or view does not exist"
*Cause:
*Action:
```

Figure 30: *Using and creating views in the HR schema.*

7. Drop the DEP_SUM_V view, **see figure 31**.

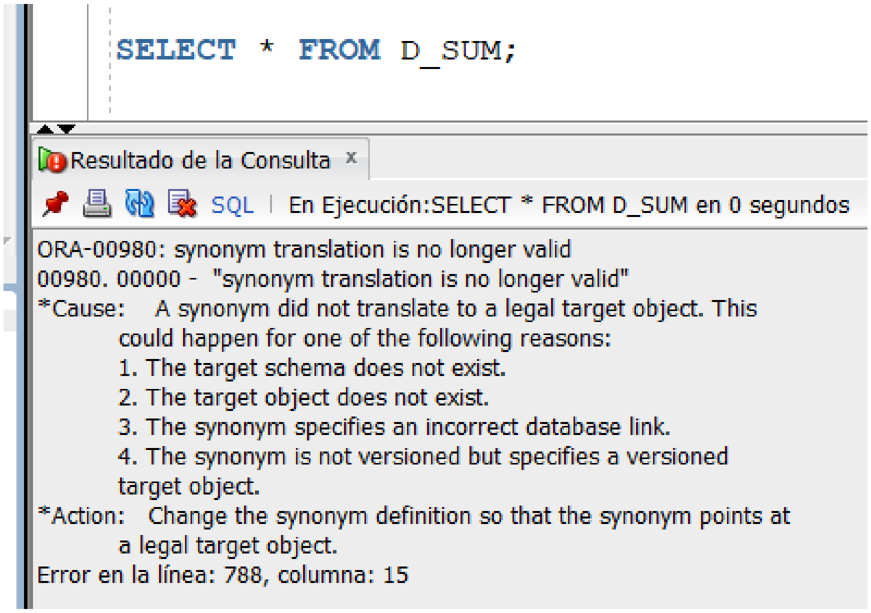


```
DROP VIEW DEPT_SUM_V;
```

```
View DEPT_SUM_V borrado.
```

Figure 31: *Using and creating views in the HR schema.*

8. Query the synonym for a dropped view, **see figure 32**.



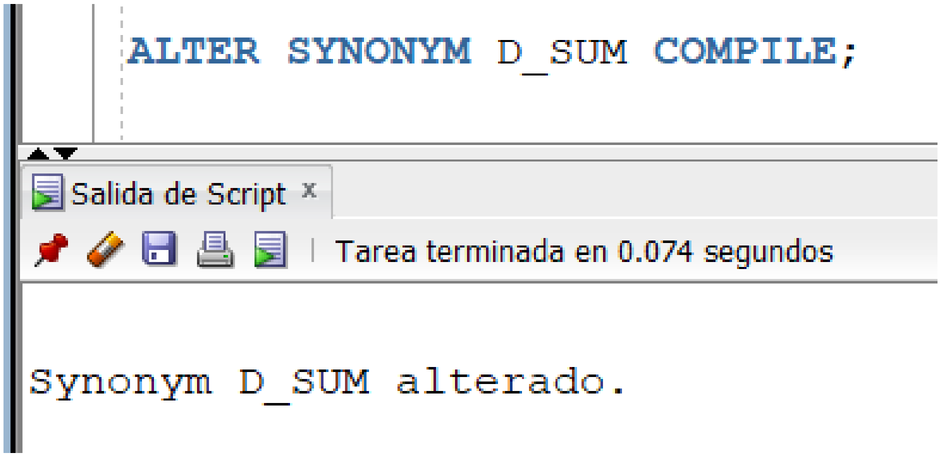
```
SELECT * FROM D_SUM;
```

En Ejecución:SELECT * FROM D_SUM en 0 segundos

```
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: 788, columna: 15
```

Figure 32: *Using and creating views in the HR schema.*

9. Recompile the broken synonym, **see figure 33**.



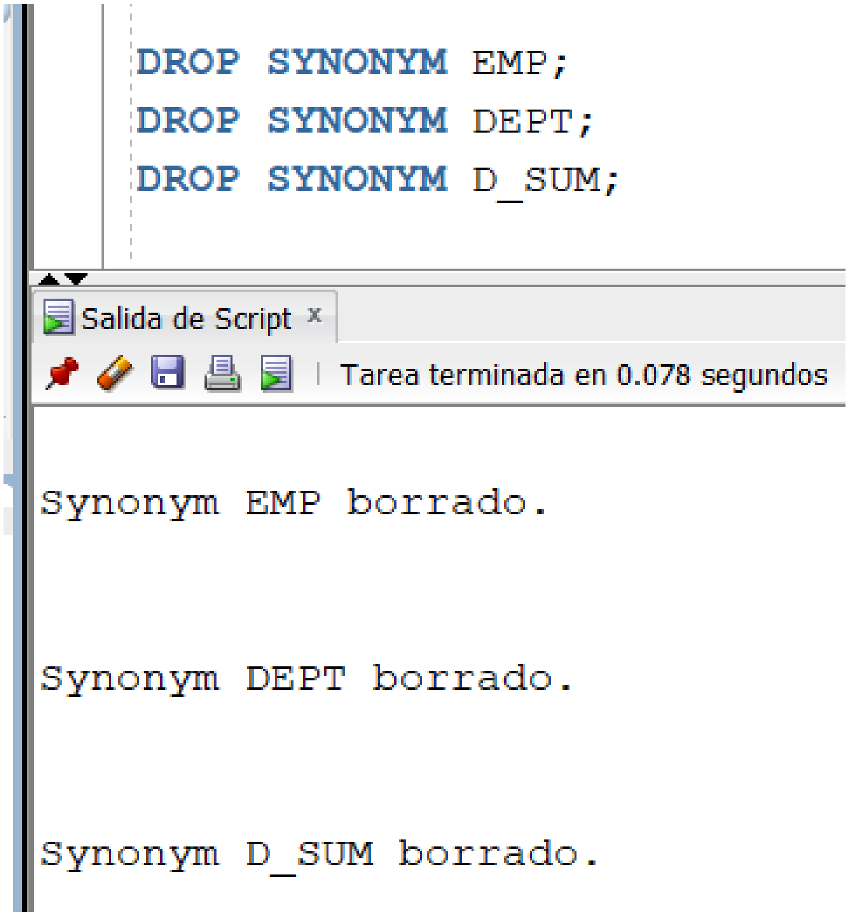
```
ALTER SYNONYM D_SUM COMPILE;
```

Tarea terminada en 0.074 segundos

```
Synonym D_SUM alterado.
```

Figure 33: *Using and creating views in the HR schema.*

10. Tidy up by dropping the synonyms, **see figure 34**.



```
DROP SYNONYM EMP;
DROP SYNONYM DEPT;
DROP SYNONYM D_SUM;
```

Tarea terminada en 0.078 segundos

```
Synonym EMP borrado.


Synonym DEPT borrado.


Synonym D_SUM borrado.
```

Figure 34: *Using and creating views in the HR schema.*

4 Pre-assessment

In this section you will find the Pre-assessment.

Criteria to be evaluate	Does it comply?	(%)
COMPLIES WITH THE REQUESTED FUNCTIONALITY	YES	
HAS THE CORRECT INDENTATION	YES	
HAS AN EASY WAY TO ACCESS THE PROVIDED FILES	YES	
HAS A REPORT WITH IDC FORMAT	YES	
REPORT INFORMATION IS FREE OF SPELLING ERRORS	YES	
DELIVERED IN TIME AND FORM	YES	
IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED)	YES	100%

5 Conclusion

The projection of the data within a relational database is stored in the table in the form of rows and columns. Projections are the first items identified during query execution. They are the selected columns within a table for which a query has been designed. Projections are mentioned in the first part of the SQL query, that is, the SELECT statement. After identifying the projections within the query frame, the next step would be to identify the rows that are relevant to the query. Filters are mentioned within the WHERE clause of the query and will identify the rows to be included in the results, the latter is called a selection.

This practice number 15 helped me practice the uses of the SELECT statement for data retrieval and projection. Finally, something important to mention is that the SQL language allows the projection and selection of data to meet the reporting needs that a programmer, developer or end user may need.