



Universidad Autónoma de Zacatecas

ACADEMIC UNIT OF ELECTRICAL ENGINEERING

Software Engineering Academic Program

Group: 5B - Semester: 2022-5^o

Practice Number: 02

Practice Name: DDL

DATE: 25/AUGUST/2022

Professor:

Aldonso Becerra Sánchez.

Student:

Cristian Omar Alvarado Rodríguez.

Índice

1. Introduction	3
2. Practice objective	4
3. Developing	4
4. Pre-assessment	32
5. Conclusion	32

DDL

August 25th, 2022

1. Introduction

Data Definition Language (DDL) is a subset of SQL. It is a language for describing data and their relationships in a database. You can generate DDL in a database object script to:

- Keep a snapshot of the database structure.
- Set up a test system where the database acts like the production system, but contains no data.
- Produce templates for new objects that you can create based on existing ones.

DDL statements are used to describe a database, to define its structure, to create its objects, and to create the subobjects of the table. You can make changes to a rule set after you create it.

Today's database industry embeds DDL in any formal language that describes data. However, it is considered a subset of SQL (Structured Query Language). SQL often uses normal English imperative verbs as sentences to implement modifications to the database. Therefore, DDL does not appear as a different language in an SQL database, but it does define changes to the database schema.

2. Practice objective

The objective of this practice is to be able to practice the use of DDL statements.

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. If necessary, investigate some answers

1. If a table is created without specifying a schema, in which schema will it be? (Choose the best answer). Challenge question.

R = D) It will be in the schema of the user creating it.

Table will be created in the schema of the current user since the schema was not specified, this happens because sometimes the schema is not specified since it would be ambiguous.

2. Several object types share the same namespace, and therefore cannot have the same name in the same schema. Which of the following object types is not in the same namespace as the others? (Choose the best answer). Challenge question.

R = A) Index.

Indices can be: non-unique (values can be repeated) or unique (values cannot be duplicated). By default, if the index type is not specified, a non-unique one is created.

3. Which of these statements will fail because the table name is not legal? (Choose two answers.)

R = C) create table 1var (col1 date); **and E)** create table delete (col1 date);

In option C it will be wrong since the name of the table must start with a letter and in option E it

would also mark an error because the name of the table is a reserved word.

4. Which of the following data types are variable length? (Choose all correct answers.)

R = VARCHAR2, NUMBER, LONG, RAW and BLOB.

5. Study these statements:

```
create table tab1 (c1 number(1), c2 date);
alter session set nls_date_format='dd-mm-yy';
insert into tab1 values (2.2,'29-07-09');
```

Will the insert succeed? (Choose the best answer).

Will the insert succeed? (Choose the best answer)

R = D) The insert will succeed.

The statement will be executed correctly since there are no errors in the writing of the statement.

6. Which of the following is not supported by Oracle as an internal data type? (Choose the best answer.)

R = D) STRING.

String is not an oracle database data type.

7. Consider this statement:

```
create table t1 as select * from employees where 9=4;
```

What will be the result? (Choose the best answer.)

R = C) The table T1 will be created but no rows inserted because the condition return FALSE.

8. When a table is created with a statement such as the following:`create table newtable as select * from oldtable;` will there be any constraints on the new table? (Choose the best answer.)

R = D) Check and not null constraints will be copied but not unique or primary key.

9. Which types of constraint require an index? (Choose all that apply.)

R = PRIMARY KEY and UNIQUE.

10. A transaction consists of two statements. The first succeeds, but the second (which updates several rows) fails partway through because of a constraint violation. What will happen? (Choose the best answer). Challenge question.

R = C) The second statement will be rolled back completely, and the first will remain uncommitted.

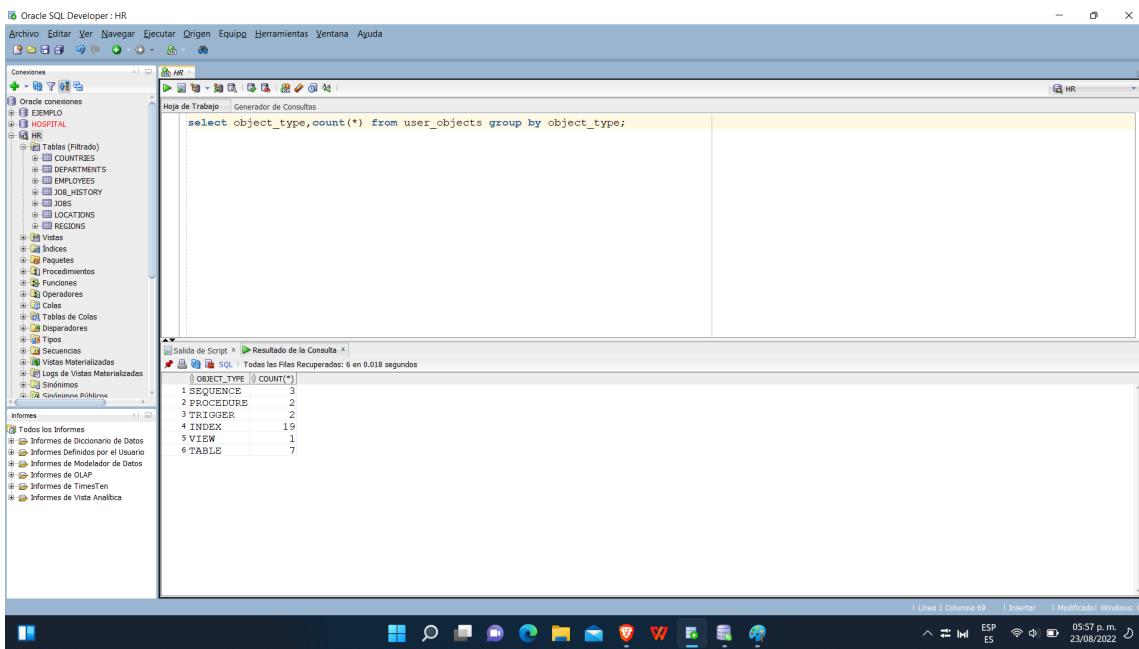
Activity 2: Write the section that describes the Work developed in the following activities.

Determine What Objects Are Accessible to Your Session

2. Determine how many objects of each type are in the HR schema. **Figure 1** shows the number of types of objects that the HR schema has.

3. Determine how many objects in total HR has permissions on; See **figure 2**.

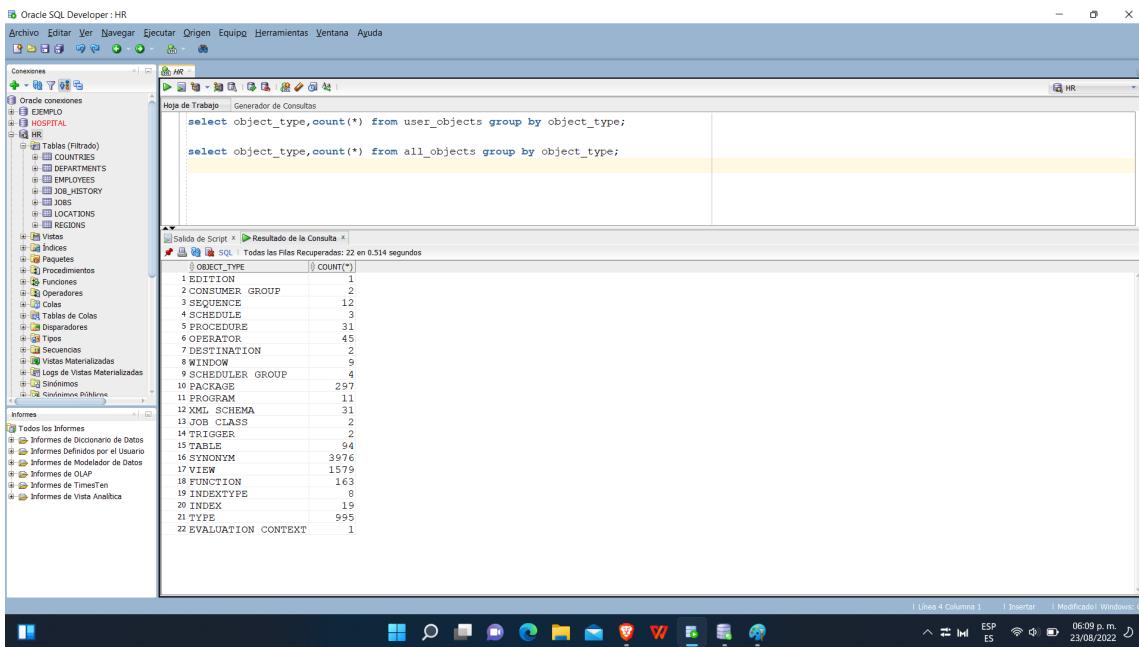
4. Determine who owns the objects HR can see. In **Figure 3** you can see the owners of the objects that HR can see.



```
select object_type, count(*) from user_objects group by object_type;
```

OBJECT_TYPE	COUNT(*)
SEQUENCE	3
PROCEDURE	2
TRIGGER	2
INDEX	19
VIEW	1
TABLE	7

Figura 1: Number of HR schema objects



OBJECT_TYPE	COUNT(*)
EDITION	1
CONSUMER GROUP	2
SEQUENCE	12
SCHEDULE	3
PROCEDURE	31
OPERATOR	45
DESTINATION	2
WINDOW	9
SCHEDULER GROUP	4
PACKAGE	297
PROGRAM	11
ROLE	31
SCHEMA	2
JOB CLASS	2
TRIGGER	2
TABLE	94
SYNONYM	3976
VIEW	1579
FUNCTION	165
INDEXTYPE	8
INDEX	19
TYPE	995
EVALUATION CONTEXT	1

Figura 2: HR object permissions

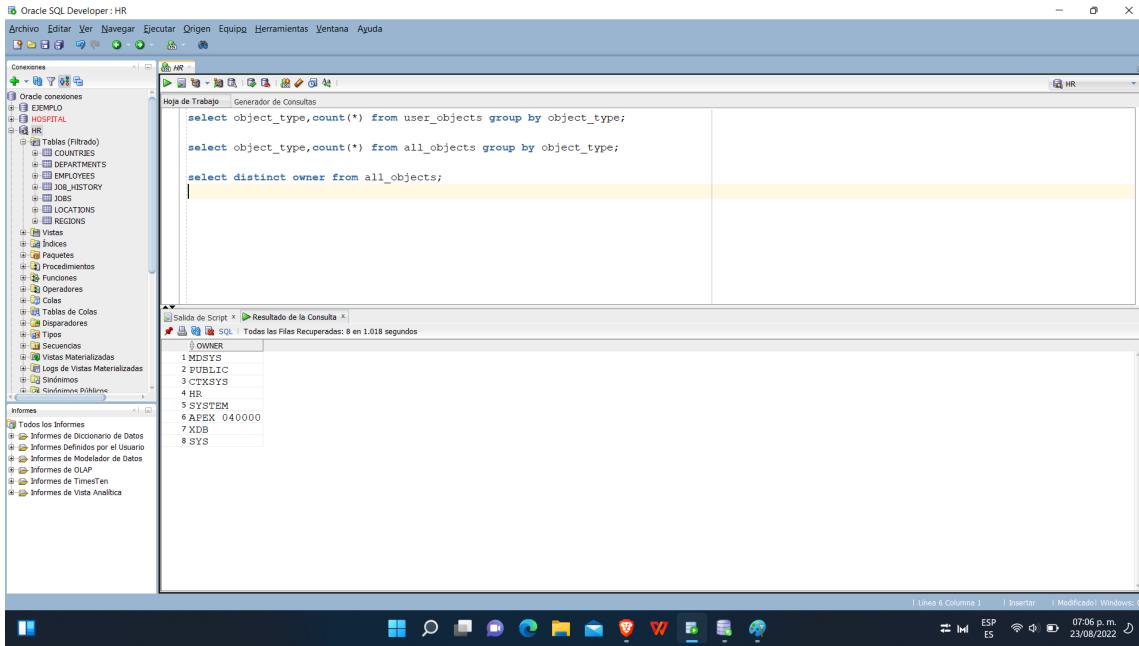
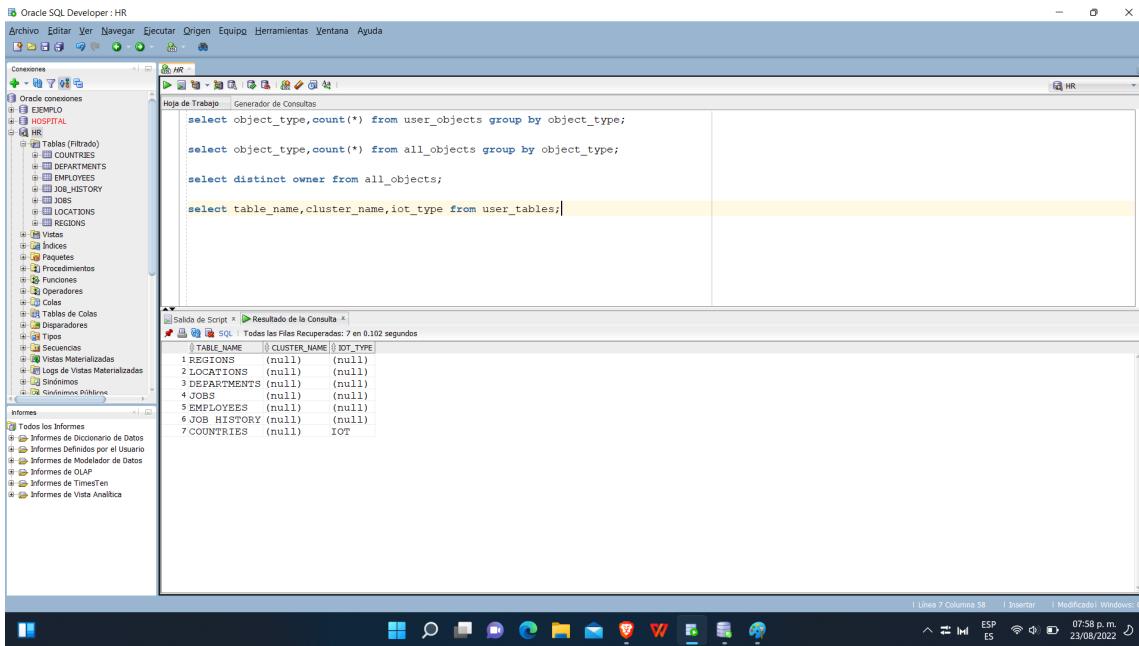


Figura 3: *Owner of the objects that HR can see.*

Investigate Table Structures

In this exercise, query various data dictionary views as user HR to determine the structure of a table.

2. Determine the names and types of tables that exist in the HR schema. Clustered tables and index organized tables (IOTs) are advanced table structures. In the HR schema, all tables are standard heap tables except for COUNTRIES which is an IOT; see **figure 4**.
3. Use the DESCRIBE command to display the structure of a table. **Figure 5** shows how the HR schema tables are described.
4. Retrieve similar information by querying a data dictionary view. **Figure 6** shows what was done at this point.



```

select object_type,count(*) from user_objects group by object_type;

select object_type,count(*) from all_objects group by object_type;

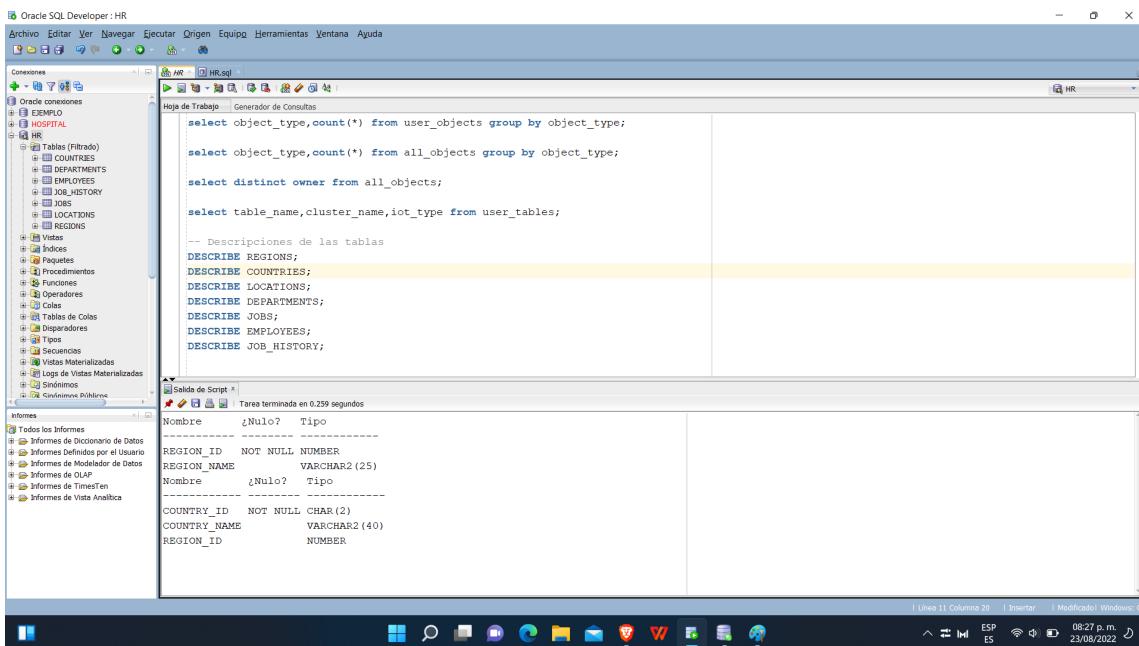
select distinct owner from all_objects;

select table_name,cluster_name,iot_type from user_tables;

```

TABLE_NAME	CLUSTER_NAME	TYPE
1 REGIONS	(null)	ITOT
2 LOCATIONS	(null)	
3 DEPARTMENTS	(null)	
4 JOBS	(null)	
5 EMPLOYEES	(null)	
6 JOB_HISTORY	(null)	
7 COUNTRIES	(null)	ITOT

Figura 4: Name and types of HR schema tables.



```

select object_type,count(*) from user_objects group by object_type;

select object_type,count(*) from all_objects group by object_type;

select distinct owner from all_objects;

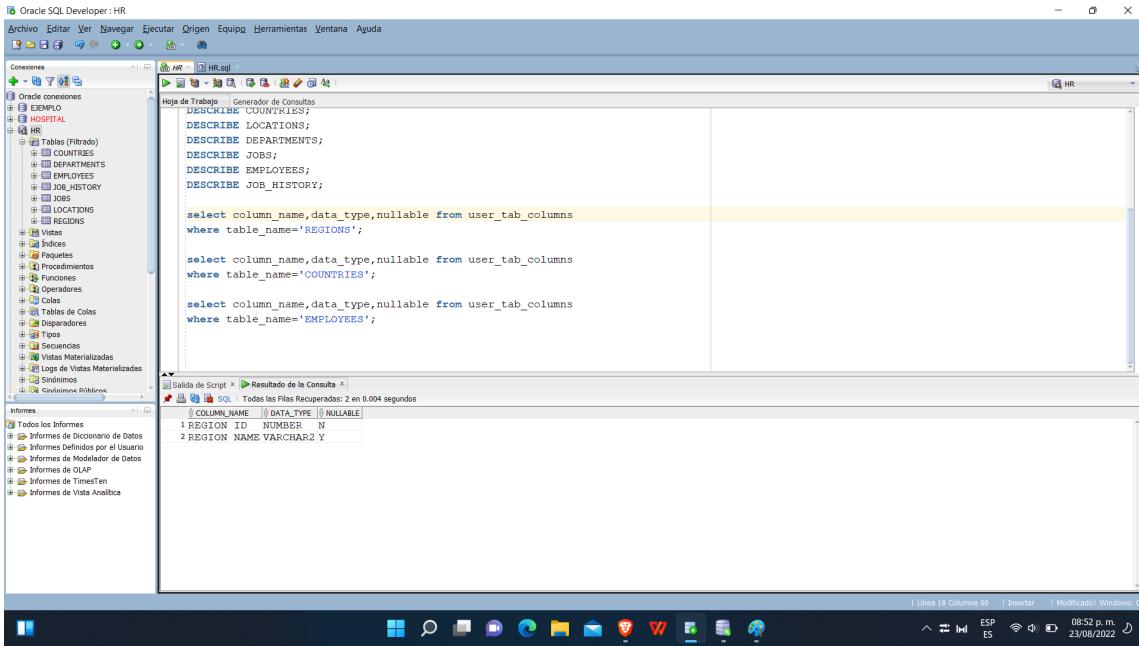
select table_name,cluster_name,iot_type from user_tables;

-- Descripciones de las tablas
DESCRIBE REGIONS;
DESCRIBE COUNTRIES;
DESCRIBE LOCATIONS;
DESCRIBE DEPARTMENTS;
DESCRIBE JOBS;
DESCRIBE EMPLOYEES;
DESCRIBE JOB_HISTORY;

```

Nombre	¿Nulo?	Tipo
REGION_ID	NOT NULL NUMBER	
REGION_NAME	VARCHAR2(25)	
Nombre	¿Nulo?	Tipo
COUNTRY_ID	NOT NULL CHAR(2)	
COUNTRY_NAME	VARCHAR2(40)	
REGION_ID	NUMBER	

Figura 5: Description of the HR schema tables.



```

OrACLE SQL Developer : HR
Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda
Conexiones | HR | HR.sql | 
+ Oracle conexiones
  + EXAMPIO
    + HOSPITAL
      + HR
        + Tablas (Filtrado)
          + COUNTRIES
          + DEPARTMENTS
          + EMPLOYEES
          + JOB_HISTORY
          + JOBS
          + LOCATIONS
          + REGIONS
        + Índices
        + Paquetes
        + Procedimientos
        + Funciones
        + Desencadenadores
        + Colas
        + Tablas de Colas
        + Disparadores
        + Tipos
        + Símbolos
        + Logos de Vista Materializadas
        + Logos de Vista Materializadas
        + Síndromos
        + Confidencias Privadas
informes | 
+ Todos los Informes
  + Informes de Diccionario de Datos
  + Informes Definidos por el Usuario
  + Informes del Analizador de Datos
  + Informes de OLAP
  + Informes de Tiempo
  + Informes de Vista Analítica
  
```

Hojas de Trabajo - Generador de Consultas

```

DESCRIBE COUNTRIES;
DESCRIBE LOCATIONS;
DESCRIBE DEPARTMENTS;
DESCRIBE JOBS;
DESCRIBE EMPLOYEES;
DESCRIBE JOB_HISTORY;

select column_name,data_type,nullable from user_tab_columns
where table_name='REGIONS';

select column_name,data_type,nullable from user_tab_columns
where table_name='COUNTRIES';

select column_name,data_type,nullable from user_tab_columns
where table_name='EMPLOYEES';

```

Salida de Script x > Resultado de la Consulta x

COLUMN_NAME	DATA_TYPE	NULLABLE
1 REGION_ID	NUMBER	N
2 REGION_NAME	VARCHAR2	Y

1 Linea 18 Columna 60 | Inserción | Modificado | Windows: CF | 08:52 p.m. 23/08/2022

Figura 6: Description of the HR schema tables.

Investigate the Data Types in the HR schema

In this exercise, find out what data types are used in the tables in the HR schema, using two techniques.

2. Use the DESCRIBE command to show the data types in some tables:

describe employees;

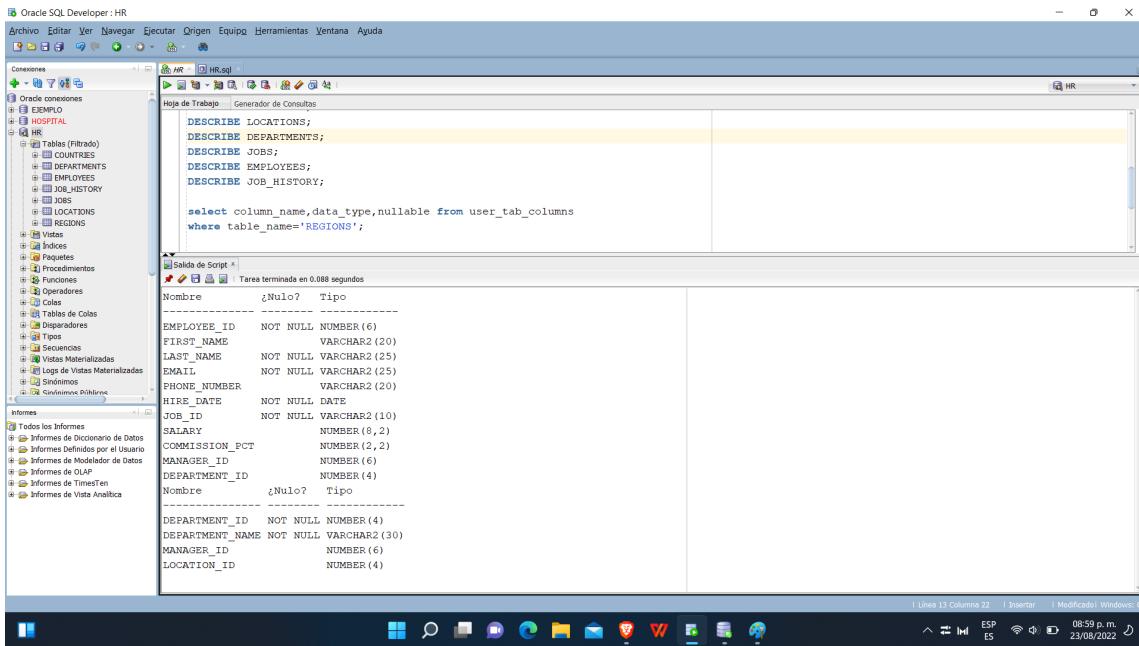
describe departments;

see **figure 7**.

3. Use a query against a data dictionary view to show what columns make up the EMPLOYEES table, as the DESCRIBE command would:

select column_name,data_type,nullable,data_length,data_precision, data_scale from user_tab_columns
where table_name='EMPLOYEES';

see **figure 8**.



```

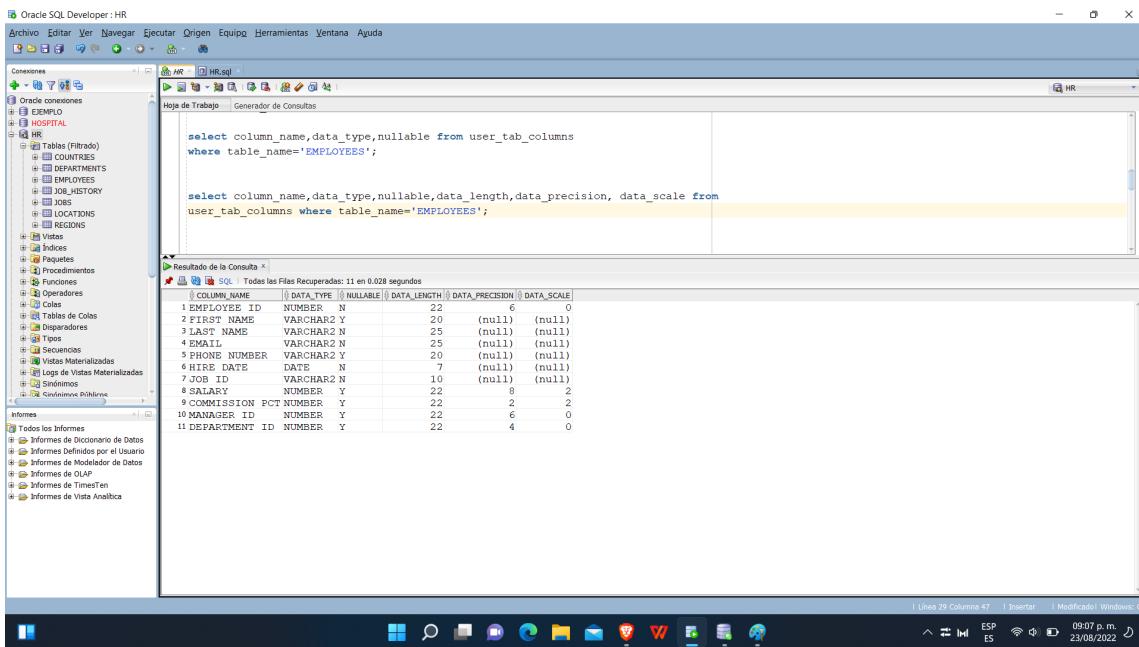
DESCRIBE LOCATIONS;
DESCRIBE DEPARTMENTS;
DESCRIBE JOBS;
DESCRIBE EMPLOYEES;
DESCRIBE JOB_HISTORY;

select column_name,data_type,nullable from user_tab_columns
where table_name='REGIONS';

-----[Output]-----
Nombre ¿Nulo? Tipo
-----[Output]-----
EMPLOYEE_ID NOT NULL NUMBER(6)
FIRST_NAME VARCHAR2(20)
LAST_NAME NOT NULL VARCHAR2(25)
EMAIL NOT NULL VARCHAR2(25)
PHONE_NUMBER VARCHAR2(20)
HIRE_DATE NOT NULL DATE
JOB_ID NOT NULL VARCHAR2(10)
SALARY NUMBER(6,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID NUMBER(6)
DEPARTMENT_ID NUMBER(4)
Nombre ¿Nulo? Tipo
-----[Output]-----
DEPARTMENT_ID NOT NULL NUMBER(4)
DEPARTMENT_NAME NOT NULL VARCHAR2(30)
MANAGER_ID NUMBER(6)
LOCATION_ID NUMBER(4)

```

Figura 7: Description of the employee and department tables of the HR schema.



```

select column_name,data_type,nullable from user_tab_columns
where table_name='EMPLOYEES';

-----[Output]-----
select column_name,data_type,nullable,data_length,data_precision, data_scale from
user_tab_columns where table_name='EMPLOYEES';

-----[Output]-----
COLUMN_NAME DATA_TYPE NULLABLE DATA_LENGTH DATA_PRECISION DATA_SCALE
1 EMPLOYEE_ID NUMBER N 22 6 0
2 FIRST_NAME VARCHAR2 Y 20 (null) (null)
3 LAST_NAME VARCHAR2 N 25 (null) (null)
4 EMAIL VARCHAR2 N 25 (null) (null)
5 PHONE_NUMBER VARCHAR2 Y 20 (null) (null)
6 HIRE_DATE DATE N 7 (null) (null)
7 JOB_ID VARCHAR2 N 10 (null) (null)
8 SALARY NUMBER X 22 8 2
9 COMMISSION_PCT NUMBER Y 22 2 2
10 MANAGER_ID NUMBER Y 22 6 0
11 DEPARTMENT_ID NUMBER Y 22 4 0

```

Figura 8: Employee table description.

Create Tables

In this exercise, use SQL Developer to create a heap table, insert some rows with a subquery, and modify the table. Do some more modifications with SQL*Plus, then drop the table.

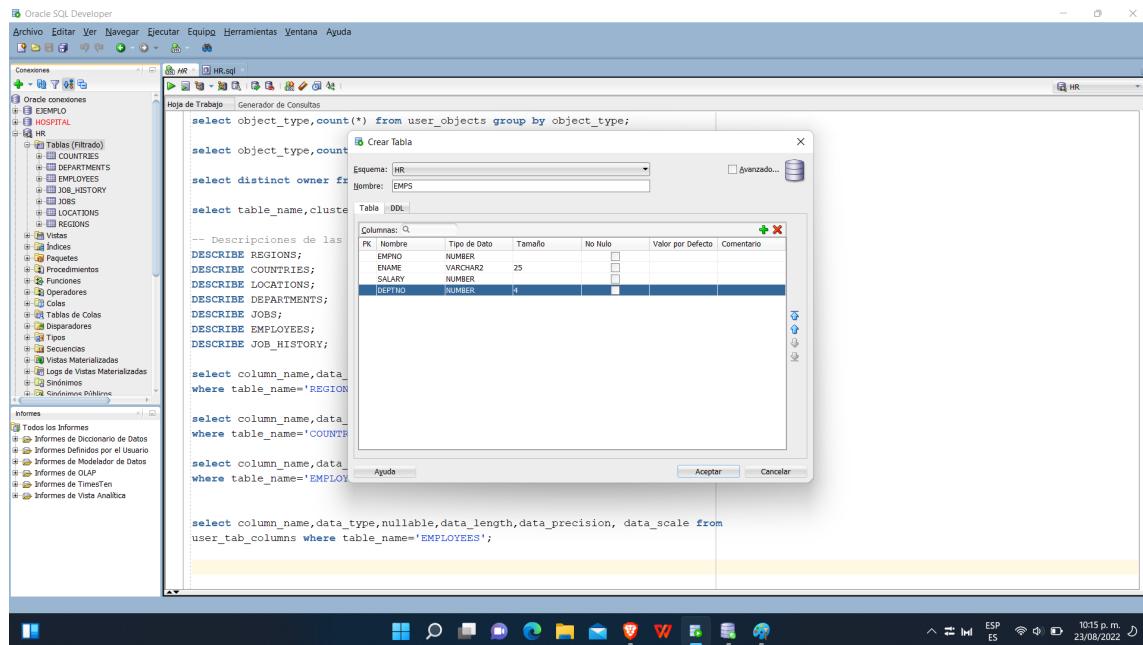
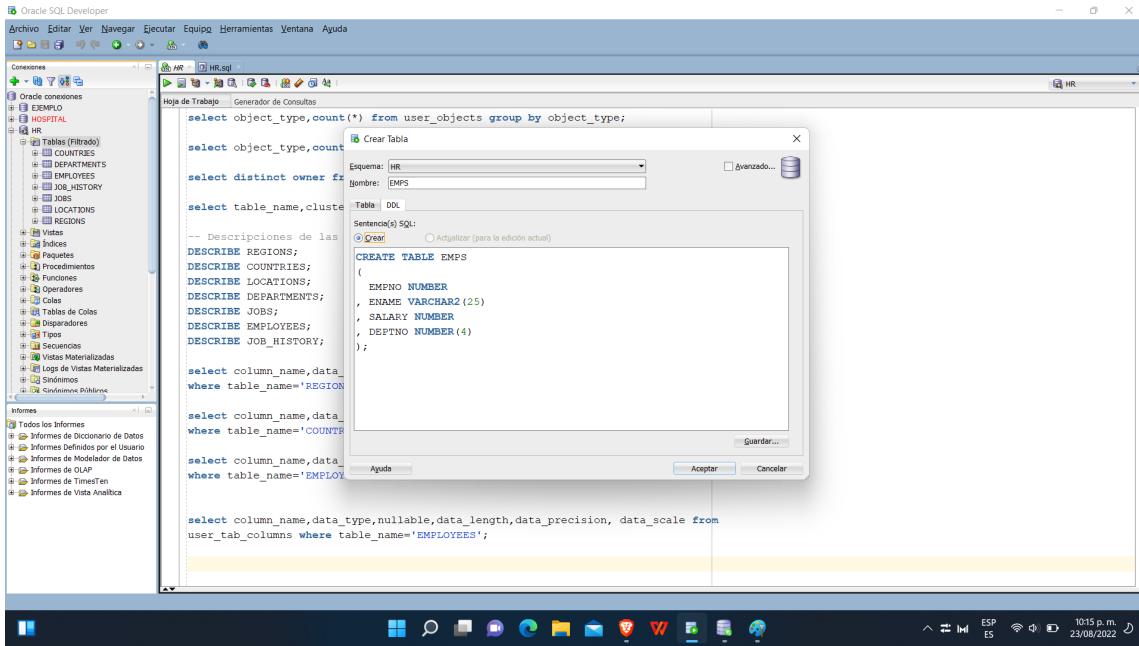
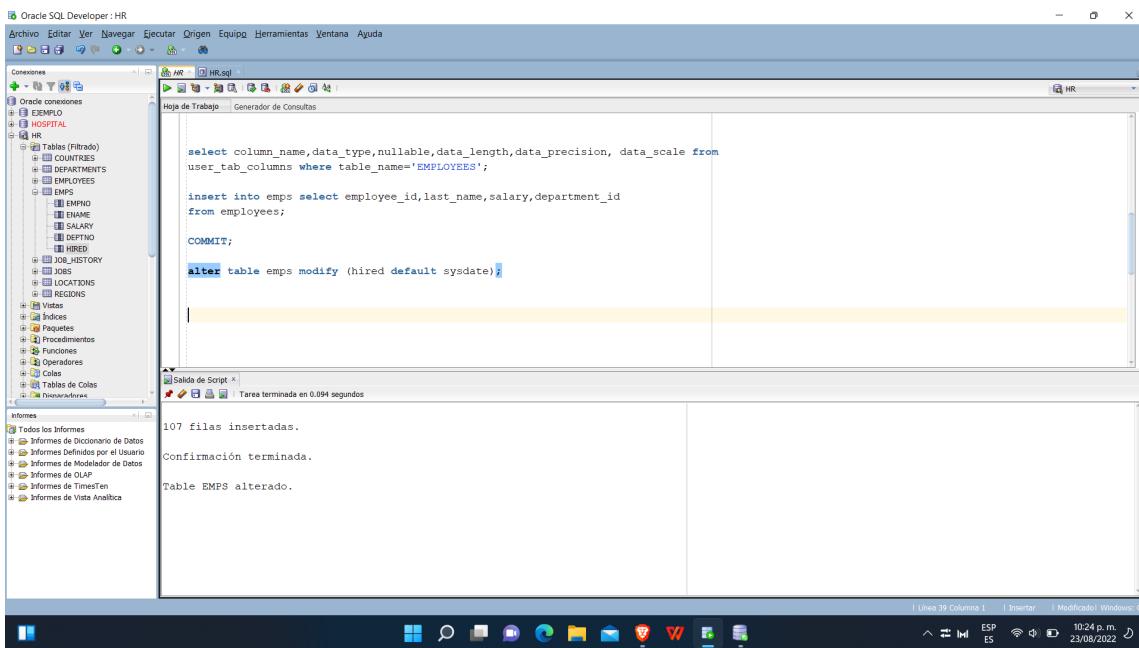
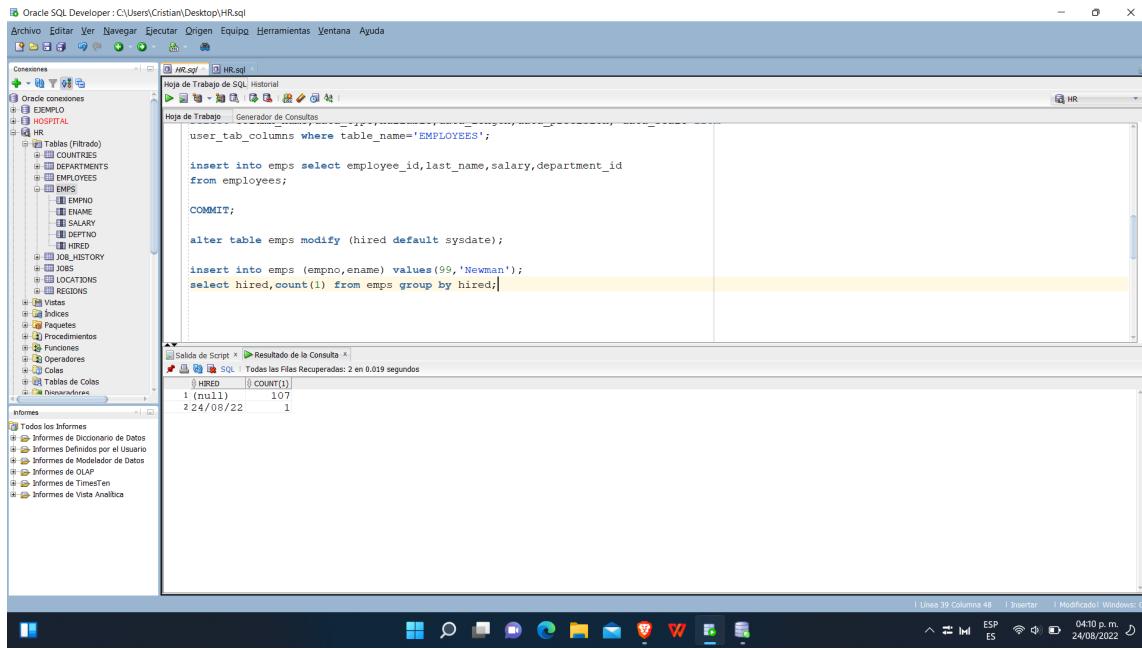


Figura 9: *Create Tables.*

In **figure 11**, a default value was established for the HIRED column, said default value is the current date and time of the system.

Figura 10: *Create Tables DDL.*Figura 11: *A default value is set to the HIRED column.*

10. Insert a row without specifying a value for HIRED and check that the new row does have a HIRED date but that the other rows do not.



The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema, including the HR schema with its tables: COUNTRIES, DEPARTMENTS, EMPLOYEES, and EMPS. The EMPS table is expanded, showing columns: EMPNO, ENAME, SALARY, DEPTNO, HIRED, and JOB_HISTORY. The central workspace contains a SQL script named HR.sql:

```

user_tab_columns where table_name='EMPLOYEES';

insert into emps select employee_id,last_name,salary,department_id
from employees;

COMMIT;

alter table emps modify (hired default sysdate);

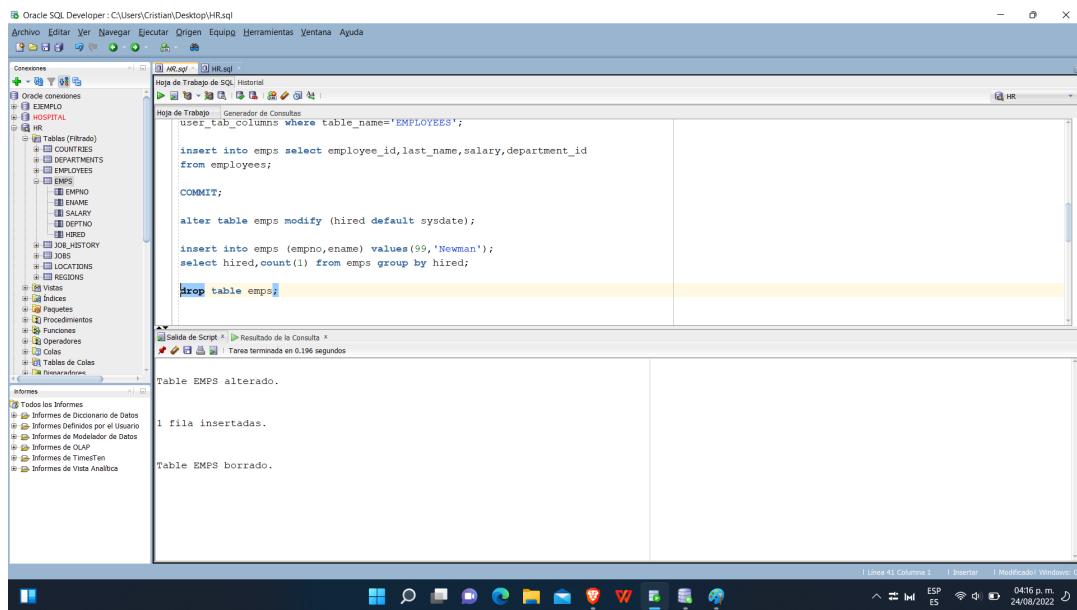
insert into emps (empno,ename) values(99,'Newman');
select hired,count(*) from emps group by hired;

```

The bottom pane shows the results of the last query:

Hired	Count(*)
1 (null)	107
22/08/2022	1

Figura 12: Check which column has a date.



The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema, including the HR schema with its tables: COUNTRIES, DEPARTMENTS, EMPLOYEES, and EMPS. The EMPS table is expanded, showing columns: EMPNO, ENAME, SALARY, DEPTNO, HIRED, and JOB_HISTORY. The central workspace contains a SQL script named HR.sql:

```

user_tab_columns where table_name='EMPLOYEES';

insert into emps select employee_id,last_name,salary,department_id
from employees;

COMMIT;

alter table emps modify (hired default sysdate);

insert into emps (empno,ename) values(99,'Newman');
select hired,count(*) from emps group by hired;

drop table emps;

```

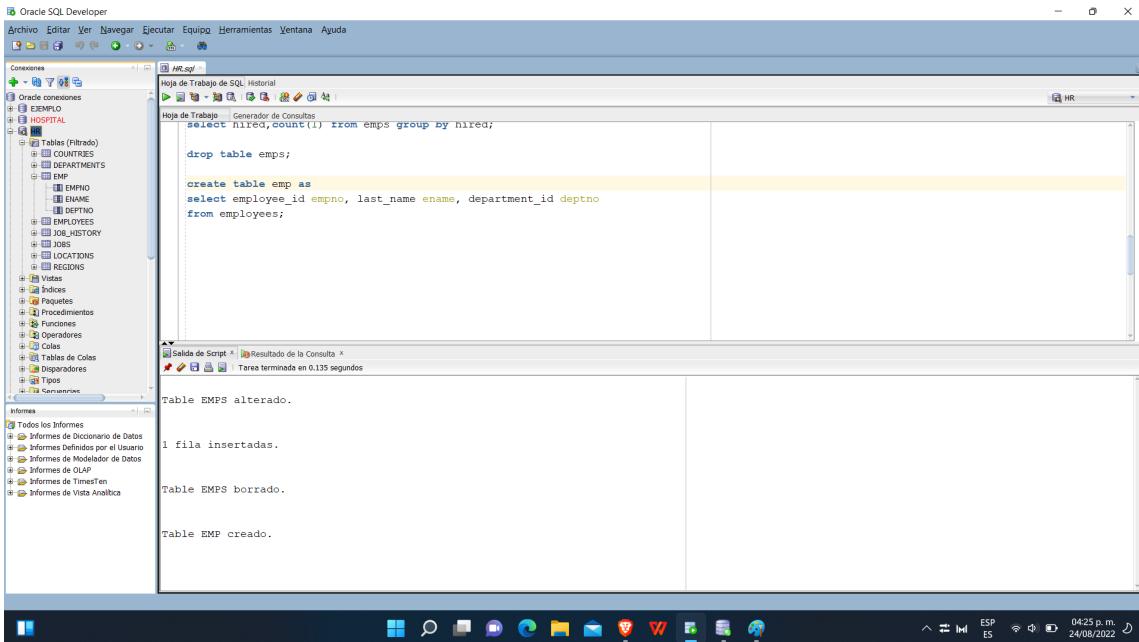
The bottom pane shows the results of the last command:

Table EMPS altered.
1 fila insertadas.
Table EMPS borrado.

Figura 13: The EMPS table was deleted.

Work with Constraints.

2. Create a table EMP as a copy of some columns from EMPLOYEES. See **figure 14**.



The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the HR schema with various objects like Tables, Views, and Procedures. The main area contains a SQL worksheet titled 'Hoja de Trabajo - Generador de Consultas' with the following script:

```

select hired, count(*) from emps group by hired;

drop table emps;

create table emp as
select employee_id empno, last_name ename, department_id deptno
from employees;

```

Below the worksheet, the 'Salida de Script' and 'Resultado de la Consulta' panes show the execution results:

- Table EMPS alterado.
- 1 fila insertadas.
- Table EMPS borrado.
- Table EMP creado.

The status bar at the bottom right indicates the task was completed in 0.135 seconds at 04:25 p.m. on 24/08/2022.

Figura 14: *Creation of the EMP table in the HR schema.*

3. Create a table DEPT as a copy of some columns from DEPARTMENTS. See **figure 15**.

4. Use DESCRIBE to describe the structure of the new tables. Note that the not null constraint on ENAME and DNAME has been carried over from the source tables. See **figure 16**.

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a SQL script named 'HR.sql' is open. The code creates three tables: 'emp', 'emp', and 'dept'. The 'emp' table is created by selecting columns from the 'employees' table. The 'dept' table is created by selecting department_id, department_name, and dname from the 'departments' table. The 'dept' table is then described, showing columns: DEPTNO (NUMBER(4)), DNAME (NOT NULL VARCHAR2(30)), and Nombre (Tipo). The 'dept' table is also described again, showing columns: DEPTNO (NUMBER(4)), DNAME (NOT NULL VARCHAR2(30)), and Nombre (Tipo). The 'emp' table is described, showing columns: EMPNO (NUMBER(6)), ENAME (NOT NULL VARCHAR2(25)), and DEPTNO (NUMBER(4)). The 'dept' table is described again, showing columns: DEPTNO (NUMBER(4)), DNAME (NOT NULL VARCHAR2(30)), and Nombre (Tipo).

```

drop table emp;

create table emp as
select employee_id empno, last_name ename, department_id deptno
from employees;

create table dept as
select department_id deptno, department_name dname from departments;

DESCRIBE DEPT;
DESCRIBE EMP;

SELECT column_name, data_type
FROM user_tab_columns
WHERE table_name = 'DEPT';

SELECT column_name, data_type
FROM user_tab_columns
WHERE table_name = 'EMP';

```

Figura 15: *Creation of the DEPT table by making a copy of the DEPARTMENTS table of the HR schema.*

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a SQL script named 'HR.sql' is open. The code describes the 'dept' and 'emp' tables. The 'dept' table is described, showing columns: DEPTNO (NUMBER(4)), DNAME (NOT NULL VARCHAR2(30)), and Nombre (Tipo). The 'emp' table is described, showing columns: EMPNO (NUMBER(6)), ENAME (NOT NULL VARCHAR2(25)), and DEPTNO (NUMBER(4)).

```

CREATE TABLE dept AS
SELECT department_id deptno, department_name dname FROM departments;

DESCRIBE DEPT;
DESCRIBE EMP;

SELECT column_name, data_type
FROM user_tab_columns
WHERE table_name = 'DEPT';

SELECT column_name, data_type
FROM user_tab_columns
WHERE table_name = 'EMP';

```

Figura 16: *Description of the DEPT and EMP tables of the HR schema.*

5. Add a primary key constraint to EMP and to DEPT and a foreign key constraint linking the tables. See **figure 17 and 18**.

The screenshot shows the Oracle SQL Developer interface. In the center, there is a SQL worksheet titled 'AR.sql' containing the following SQL code:

```

create table dept as
select department_id deptno, department_name dname from departments;

DESCRIBE DEPT;
DESCRIBE EMP;

alter table emp add constraint emp_pk primary key (empno);
alter table dept add constraint dept_pk primary key (deptno);
alter table emp add constraint
dept_fk foreign key (deptno) references dept on delete set null;

```

Below the code, the output shows the results of the executed statements:

```

Table EMP altered.

Table DEPT altered.

Table EMP altered.

```

Figura 17: *Restrictions on EMP and DEPT tables.*

The screenshot shows the Oracle SQL Developer interface with the 'EMP' table selected in the left sidebar. In the central pane, the 'Restricciones' tab is active, displaying the following table of constraints:

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE	VALIDATED	GENERATED	BAD	RELY	LAS
1 DEPT FK	Foreign Key	(null)	HR	DEPT	DEPT FK	SET NULL	ENABLED	NOT DEFERRABLE	VALIDATED	USER NAME	(null)	(null)	24/08/2022
2 EMP PK	Primary Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED	USER NAME	(null)	(null)	24/08/2022
3 SYS_C007408	Check	"ENAME" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED	GENERATED	NAMES	(null)	24/08/2022

Figura 18: *Restrictions on the EMP table.*

6. Demonstrate the effectiveness of the constraints by trying to insert data that will violate them.

The screenshot shows the Oracle SQL Developer interface with the HR schema selected. In the central workspace, a script is being run:

```

alter table emp add constraint emp_pk primary key (empno);
alter table dept add constraint dept_pk primary key (deptno);
alter table emp add constraint dept_fk foreign key (deptno) references dept on delete set null;

insert into dept values(10,'New Department');
insert into emp values(9999,'New emp',99);
truncate table dept;

```

The output window displays three error messages:

- Error que empieza en la linea: 61 del comando :
insert into dept values(10,'New Department')
Informe de error -
ORA-00001: unique constraint (HR.DEPT_PK) violated
- Error que empieza en la linea: 62 del comando :
insert into emp values(9999,'New emp',99)
Informe de error -
ORA-02291: integrity constraint (HR.DEPT_FK) violated - parent key not found
- Error que empieza en la linea: 63 del comando :
truncate table dept
Informe de error -
ORA-02266: unique/primary keys in table referenced by enabled foreign keys
02266. 00000 - "unique/primary keys in table referenced by enabled foreign keys"

Figura 19: Demonstrate the effectiveness of the constraints.

7. Tidy up by dropping the tables. Note that this must be done in the correct order.

The screenshot shows the Oracle SQL Developer interface with the HR schema selected. A script is being run to drop the tables:

```

dept_fk Foreign key (deptno) references dept on delete set null;

insert into dept values(10,'New Department');
insert into emp values(9999,'New emp',99);
truncate table dept;

DROP TABLE EMP;
DROP TABLE DEPT;

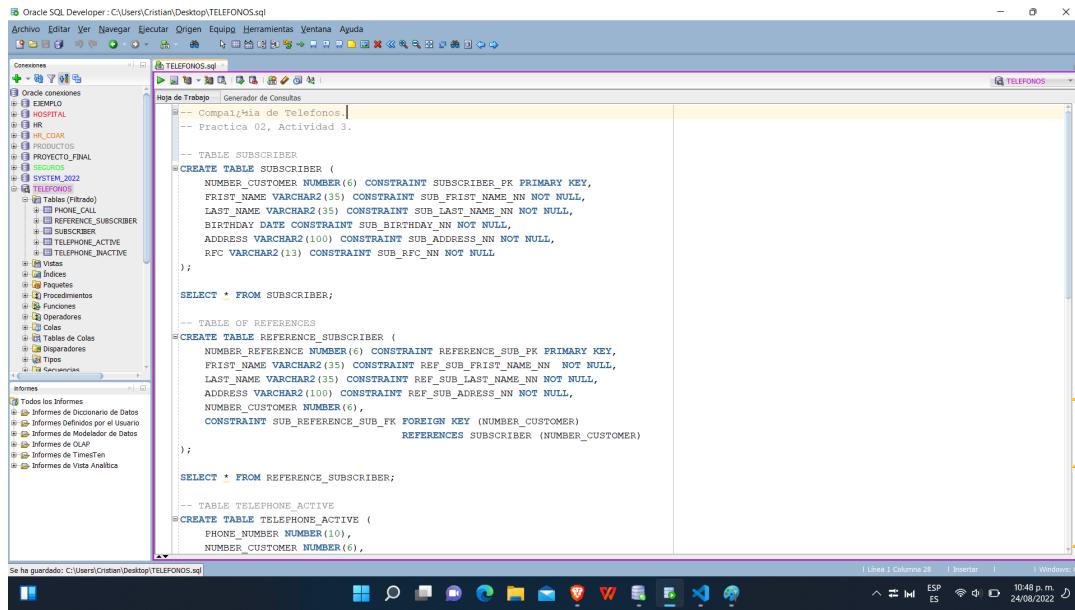
```

The output window shows the results of the table drops:

- Table EMP borrado.
- Table DEPT borrado.

Figura 20: EMP and DEPT tables were deleted.

Activity 3: In this activity we are asked to carry out the necessary SQL sentences to create the necessary tables for the problem described, for this the following sentences were carried out:



```

-- Oracle SQL Developer : C:\Users\Kristian\Desktop\TELEFONOS.sql
Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda
Conexiones TELEFONOS.sql Hoja de Trabajo Generador de Consultas
-- Compañia de Telefonos
-- Práctica 02, Actividad 3.

-- TABLE SUBSCRIBER
CREATE TABLE SUBSCRIBER (
    NUMBER_CUSTOMER NUMBER(6) CONSTRAINT SUBSCRIBER_PK PRIMARY KEY,
    FIRST_NAME VARCHAR2(35) CONSTRAINT SUB_FIRST_NAME_NN NOT NULL,
    LAST_NAME VARCHAR2(35) CONSTRAINT SUB_LAST_NAME_NN NOT NULL,
    BIRTHDAY_DATE CONSTRAINT SUB_BIRTHDAY_NN NOT NULL,
    ADDRESS VARCHAR2(100) CONSTRAINT SUB_ADDRESS_NN NOT NULL,
    RFC VARCHAR2(13) CONSTRAINT SUB_RFC_NN NOT NULL
);

SELECT * FROM SUBSCRIBER;

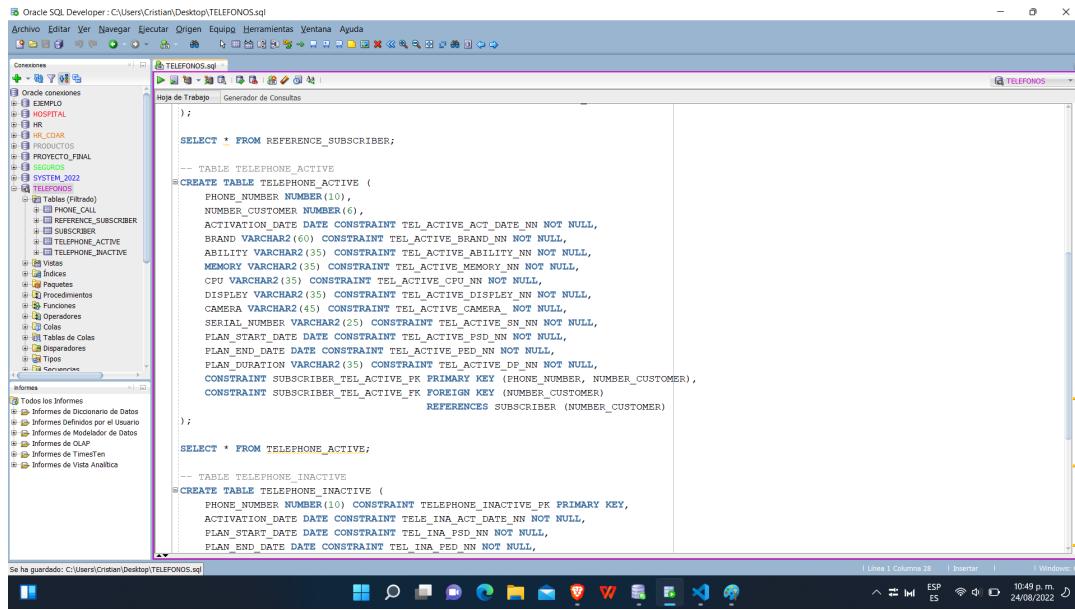
-- TABLE OF REFERENCES
CREATE TABLE REFERENCE_SUBSCRIBER (
    NUMBER_REFERENCE NUMBER(6) CONSTRAINT REFERENCE_SUB_PK PRIMARY KEY,
    FIRST_NAME VARCHAR2(35) CONSTRAINT REF_SUB_FIRST_NAME_NN NOT NULL,
    LAST_NAME VARCHAR2(35) CONSTRAINT REF_SUB_LAST_NAME_NN NOT NULL,
    ADDRESS VARCHAR2(100) CONSTRAINT REF_SUB_ADDRESS_NN NOT NULL,
    NUMBER_CUSTOMER NUMBER(6),
    CONSTRAINT SUB_REFERENCE_SUB_FK FOREIGN KEY (NUMBER_CUSTOMER)
        REFERENCES SUBSCRIBER (NUMBER_CUSTOMER)
);

SELECT * FROM REFERENCE_SUBSCRIBER;

-- TABLE TELEPHONE_ACTIVE
CREATE TABLE TELEPHONE_ACTIVE (
    PHONE_NUMBER NUMBER(10),
    NUMBER_CUSTOMER NUMBER(6)
);

```

Figura 21: *Sql statements activity three*



```

-- Oracle SQL Developer : C:\Users\Kristian\Desktop\TELEFONOS.sql
Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda
Conexiones TELEFONOS.sql Hoja de Trabajo Generador de Consultas
-- TABLE TELEPHONE_ACTIVE
CREATE TABLE TELEPHONE_ACTIVE (
    PHONE_NUMBER NUMBER(10),
    NUMBER_CUSTOMER NUMBER(6),
    ACTIVATION_DATE DATE CONSTRAINT TEL_ACTIVE_ACT_DATE_NN NOT NULL,
    BRAND VARCHAR2(60) CONSTRAINT TEL_ACTIVE_BRAND_NN NOT NULL,
    ABILITY VARCHAR2(35) CONSTRAINT TEL_ACTIVE_ABILITY_NN NOT NULL,
    MEMORY VARCHAR2(35) CONSTRAINT TEL_ACTIVE_MEMORY_NN NOT NULL,
    CPU VARCHAR2(35) CONSTRAINT TEL_ACTIVE_CPU_NN NOT NULL,
    DISPLAY VARCHAR2(35) CONSTRAINT TEL_ACTIVE_DISPLAY_NN NOT NULL,
    CAMERA VARCHAR2(45) CONSTRAINT TEL_ACTIVE_CAMERA_NN NOT NULL,
    SERIAL_NUMBER VARCHAR2(25) CONSTRAINT TEL_ACTIVE_SN_NN NOT NULL,
    PLAN_START_DATE DATE CONSTRAINT TEL_ACTIVE_PSD_NN NOT NULL,
    PLAN_END_DATE DATE CONSTRAINT TEL_ACTIVE_PED_NN NOT NULL,
    PLAN_DURATION VARCHAR2(35) CONSTRAINT TEL_ACTIVE_DP_NN NOT NULL,
    CONSTRAINT SUBSCRIBER_TEL_ACTIVE_FK PRIMARY KEY (PHONE_NUMBER, NUMBER_CUSTOMER),
    CONSTRAINT SUBSCRIBER_TEL_ACTIVE_FK FOREIGN KEY (NUMBER_CUSTOMER)
        REFERENCES SUBSCRIBER (NUMBER_CUSTOMER)
);

SELECT * FROM TELEPHONE_ACTIVE;

-- TABLE TELEPHONE_INACTIVE
CREATE TABLE TELEPHONE_INACTIVE (
    PHONE_NUMBER NUMBER(10) CONSTRAINT TELEPHONE_INACTIVE_PK PRIMARY KEY,
    ACTIVATION_DATE DATE CONSTRAINT TEL_INA_ACT_DATE_NN NOT NULL,
    PLAN_START_DATE DATE CONSTRAINT TEL_INA_PSD_NN NOT NULL,
    PLAN_END_DATE DATE CONSTRAINT TEL_INA_PED_NN NOT NULL,
    CONSTRAINT SUBSCRIBER_TEL_INACTIVE_FK FOREIGN KEY (PHONE_NUMBER, NUMBER_CUSTOMER)
        REFERENCES SUBSCRIBER (NUMBER_CUSTOMER)
);

```

Figura 22: *Sql statements activity three*

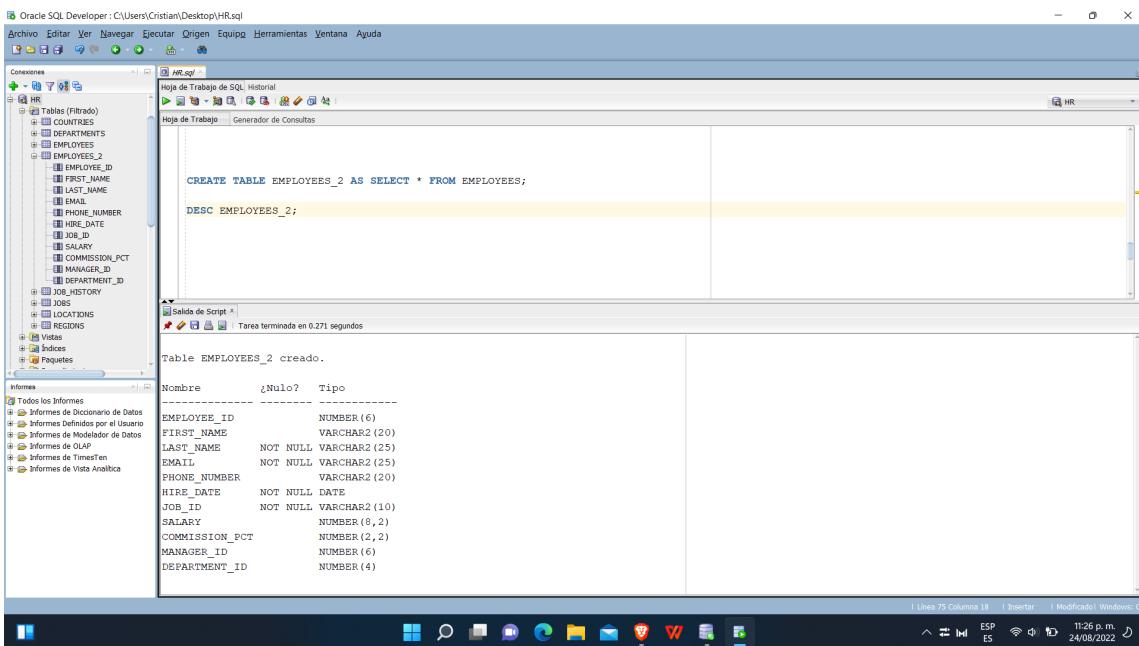
Activity 4: Propose a response to the following scenario issue:

- You are designing table structures for a human resources application. The business analysts have said that when an employee leaves the company, his employee record should be moved to an archive table. Can constraints help? Explain the reasons.

R = In this case, the restrictions could cause a problem because if the employee table is related to other tables, important information could be deleted from other tables because a cascade delete could occur, but in the same way the action could be carried out satisfactorily.

Activity 5:

- Create the EMPLOYEES_2 table based on the EMPLOYEES table from HR schema. Use the CREATE statement that employs a SELECT statement.
- Describe the table structure



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connexiones' sidebar shows a connection to the 'HR' schema. The main workspace has two tabs: 'Hoja de Trabajo de SQL: Historial' and 'Hoja de Trabajo - Generador de Consultas'. The 'Generador de Consultas' tab contains the following SQL code:

```

CREATE TABLE EMPLOYEES_2 AS SELECT * FROM EMPLOYEES;
DESC EMPLOYEES_2;

```

Below the code, the 'Salida de Script' panel displays the message: 'Table EMPLOYEES_2 creado.' (Table EMPLOYEES_2 created.)

Nombre	¿Nulo?	Tipo
EMPLOYEE_ID		NUMBER (6)
FIRST_NAME		VARCHAR2 (20)
LAST_NAME	NOT NULL	VARCHAR2 (25)
EMAIL	NOT NULL	VARCHAR2 (25)
PHONE_NUMBER		VARCHAR2 (20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2 (10)
SALARY		NUMBER (6,2)
COMMISSION_PCT		NUMBER (2,2)
MANAGER_ID		NUMBER (6)
DEPARTMENT_ID		NUMBER (4)

Figura 23: Description of the table EMPLOYEES_2

- Alter the EMPLOYEES_2 table status to read-only. See **figure 24**.

- Try to insert a row the table. Depict the results. See **figure 25**.
- Revert the EMPLOYEES_2 table to the write status. Now, try to insert the same row again. Depict the results. see **figure 26**.

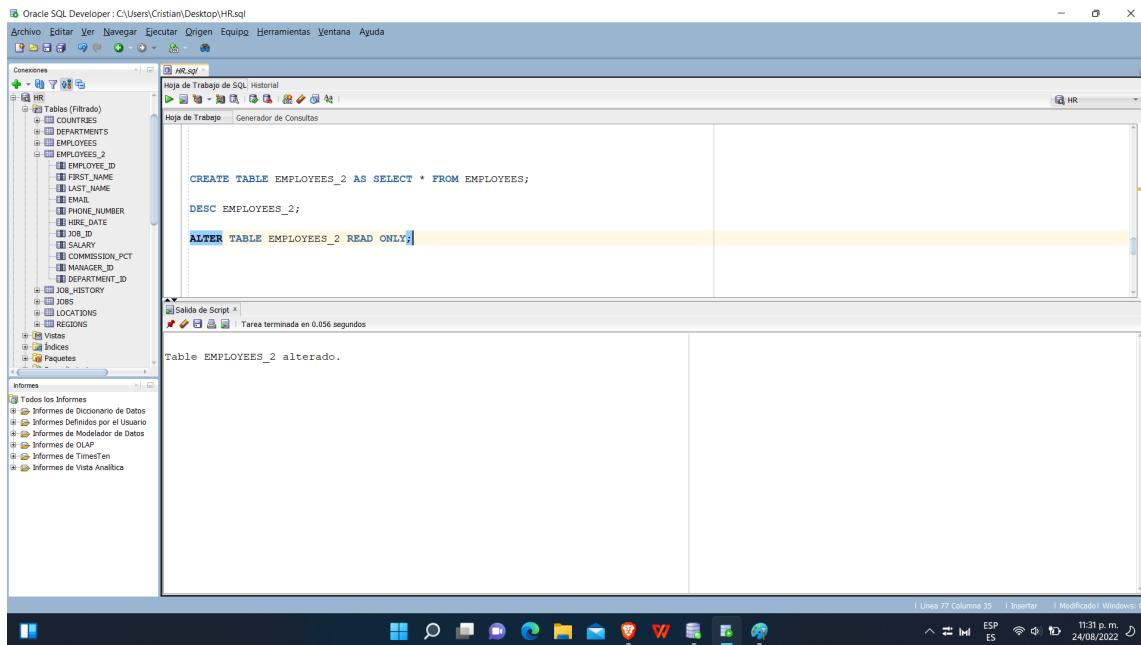


Figura 24: *Changed the state of the table to read-only.*

The screenshot shows the Oracle SQL Developer interface. In the central workspace, there is a SQL script window titled 'Hoja de Trabajo' containing the following SQL code:

```

CREATE TABLE EMPLOYEES_2 AS SELECT * FROM EMPLOYEES;
DESC EMPLOYEES_2;
ALTER TABLE EMPLOYEES_2 READ ONLY;
ALTER TABLE EMPLOYEES_2 ADD AGE DATE;
  
```

Below the script window, the 'Salida de Script' (Script Output) pane displays an error message:

```

Error que empieza en la linea: 79 del comando :
ALTER TABLE EMPLOYEES_2 ADD AGE DATE
Informe de error -
ORA-12081: update operation not allowed on table "HR"."EMPLOYEES_2"
12081. 00000 -  "update operation not allowed on table \"%s\".\"%s\""
*Cause:  An attempt was made to update a read-only materialized view.
*Action: No action required. Only Oracle is allowed to update a
read-only materialized view.
  
```

The status bar at the bottom right indicates the date and time as 24/08/2022 11:37 p.m.

Figura 25: Attempt to modify table EMPLOYEES_2.

The screenshot shows the Oracle SQL Developer interface. In the central workspace, there is a SQL script window titled 'Hoja de Trabajo' containing the following SQL code:

```

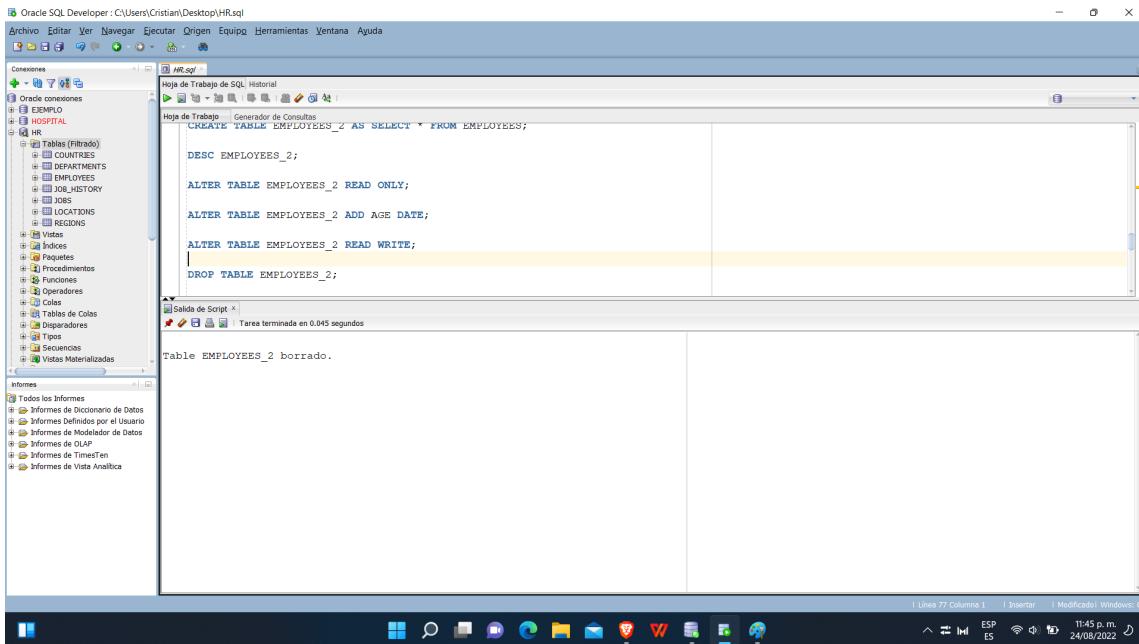
CREATE TABLE EMPLOYEES_2 AS SELECT * FROM EMPLOYEES;
DESC EMPLOYEES_2;
ALTER TABLE EMPLOYEES_2 READ ONLY;
ALTER TABLE EMPLOYEES_2 ADD AGE DATE;
ALTER TABLE EMPLOYEES_2 READ WRITE;
  
```

Below the script window, the 'Salida de Script' (Script Output) pane displays the message 'Table EMPLOYEES_2 alterado.' (Table EMPLOYEES_2 altered).

The status bar at the bottom right indicates the date and time as 24/08/2022 11:42 p.m.

Figura 26: Making the table write-only

- Drop the EMPLOYEES_2 table. See **figure 27**.

Figura 27: Deleted table *EMPLOYEES_2*

Activity 6:

1. Generate the DDL statements to create only the isolated tables of Product_Hierarchy and Product_Suppliers (add constraints here in CREATE, don't add primary and foreign keys yet). Use descriptive names in constraints. See **figure 28**.
2. Generate the DDL statements to create the Suppliers table.
3. Add the fields: "status" as numeric, "name" as varchar and "city" as int. See **figure 29 and 30**.

```

Hoja de Trabajo Generador de Consultas
-- Practica 02, Actividad 6

CREATE TABLE PRODUCT_HIERARCHY(
    PRODUCT_CODE CHAR(8),
    PARENT_PRODUCT_CODE CHAR(10),
    PRODUCT_NAME VARCHAR2(20) CONSTRAINT PRO_HI_PN_NN NOT NULL,
    PRODUCT_DESCRIPTION VARCHAR2(40) CONSTRAINT PRO_HIE_PD_NN NOT NULL,
    AMOUNT_REQUIRED NUMBER(6) CONSTRAINT PRO_HI_AR_NN NOT NULL
        CONSTRAINT PR_AM_CK CHECK (AMOUNT_REQUIRED > 0)
);

SELECT * FROM PRODUCT_HIERARCHY;

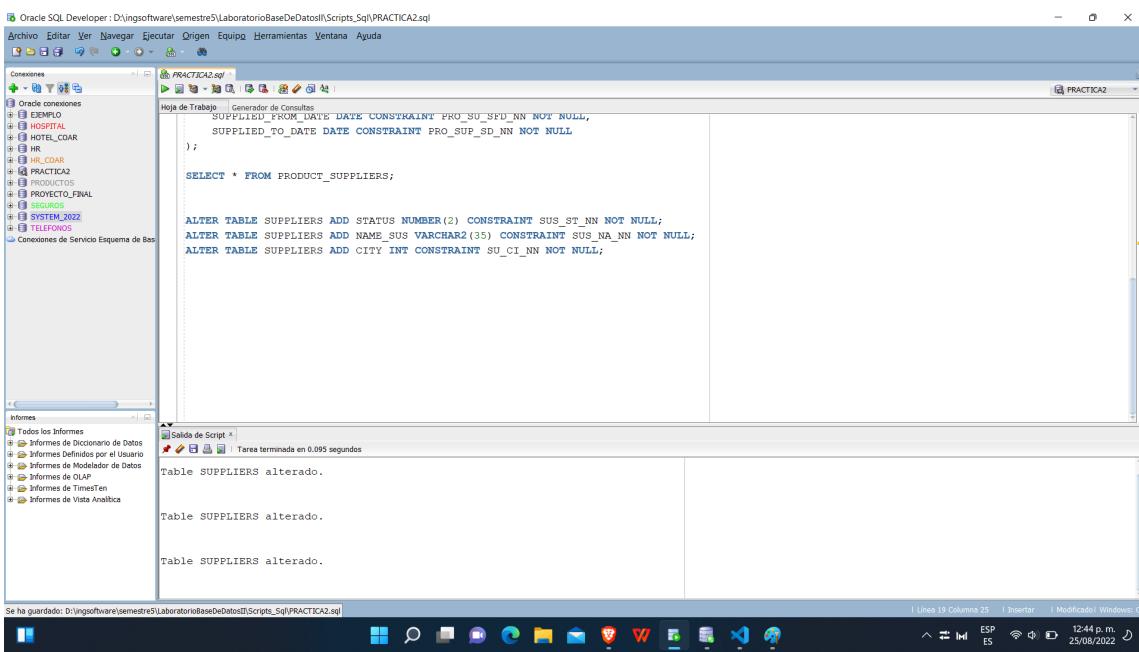
CREATE TABLE SUPPLIERS (
    SUPPLIER_CODE CHAR (8),
    SUPPLIER_DETAILS VARCHAR2 (30) CONSTRAINT SUPPLIER_SD_NN NOT NULL
);

SELECT * FROM SUPPLIERS;

CREATE TABLE PRODUCT_SUPPLYERS(
    PRODUCT_CODE CHAR (8),
    SUPPLIER_CODE CHAR (8),
    SUPPLIED_FROM_DATE DATE CONSTRAINT PRO_SU_SFD_NN NOT NULL,
    SUPPLIED_TO_DATE DATE CONSTRAINT PRO_SUP_SD_NN NOT NULL
);

SELECT * FROM PRODUCT_SUPPLYERS;

```

Figura 28: *DDL statements for creating tables*


The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar:** Includes standard options like Archivo, Editar, Ver, Navegar, Ejecutar, Origen, Equipo, Herramientas, Ventana, Ayuda.
- Left Sidebar (Conexiones):** Shows various database connections including EJEMPLO, HOSPITAL, PROYECTO_FINAL, and PRACTICA2.
- Central Area (Hoja de Trabajo):** Displays the DDL code for modifying the SUPPLIERS table:

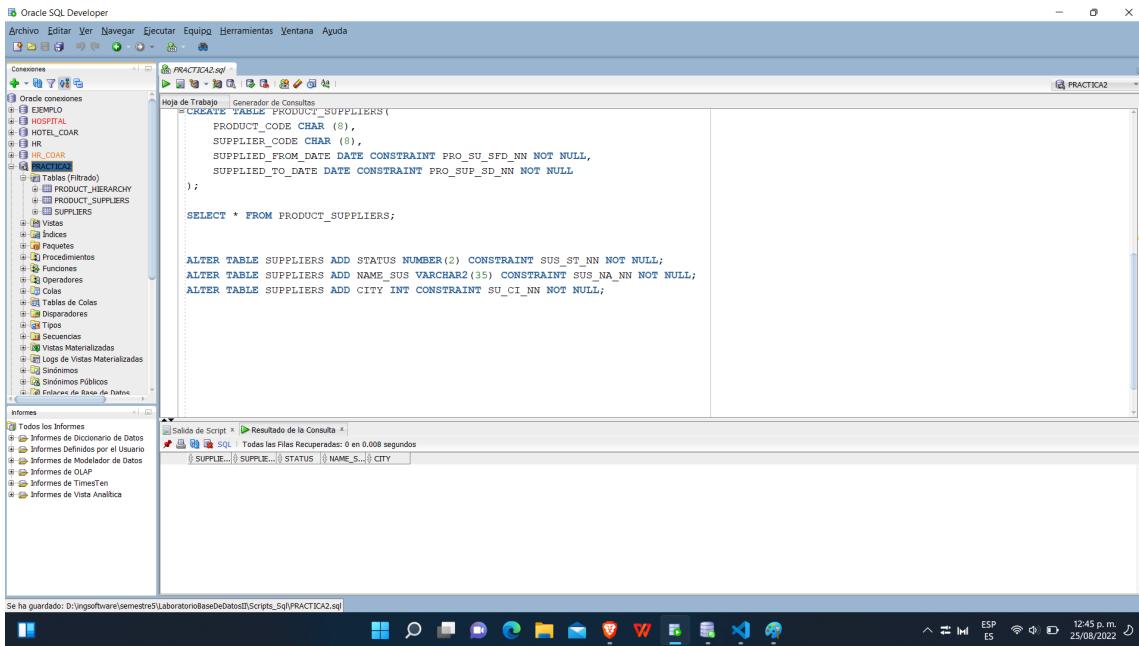

```

ALTER TABLE SUPPLIERS ADD STATUS NUMBER(2) CONSTRAINT SUS_ST_NN NOT NULL;
ALTER TABLE SUPPLIERS ADD NAME_SUS VARCHAR2(35) CONSTRAINT SUS_NA_NN NOT NULL;
ALTER TABLE SUPPLIERS ADD CITY INT CONSTRAINT SU_CI_NN NOT NULL;
      
```
- Bottom Area (Salida de Script):** Shows the execution results:


```

Table SUPPLIERS alterado.
Table SUPPLIERS alterado.
Table SUPPLIERS alterado.
      
```
- Bottom Status Bar:** Shows the path "Se ha guardado: D:\ingsoftware\semestre5\LaboratorioBaseDeDatos\Scripts_Sql\PRACTICA2.sql", the date "25/08/2022", and the time "12:44 p. m."

Figura 29: *Modifying the suppliers table*

Figura 30: *Modifying the suppliers table*

4. Delete the column “status” from the Suppliers table.
5. Rename the column `çity` to be called `çity_address`.
6. Modify the data type of the “city_address” column, so that instead of saving numbers, it saves a variable string with a maximum length of 50.
7. Shows the structure of the Suppliers table.

Figure 31 shows the sentences made for the previous points.

8. Generate, with the ALTER statements, all the necessary instructions to link all the tables as they appear in the diagram. See **figure 32**.
9. Obtain the relational diagram and compare the results. If necessary, modify your script to suit the correct one. See **figure 33**.

Oracle SQL Developer : D:\ingsoftware\semestre5\LaboratorioBaseDeDatosII\Scripts_Sql\PRACTICA2.sql

Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda

Conexiones | PRACTICA2.sql | PRACTICA2

Hojas de Trabajo - Generador de Consultas

```

PRODUCT_CODE CHAR(8),
SUPPLIER_CODE CHAR(8),
SUPPLIED_FROM_DATE DATE CONSTRAINT PRO_SU_SFD_NN NOT NULL,
SUPPLIED_TO_DATE DATE CONSTRAINT PRO_SUP_SD_NN NOT NULL
);

SELECT * FROM PRODUCT_SUPPLIERS;

ALTER TABLE SUPPLIERS ADD STATUS NUMBER(2) CONSTRAINT SUS_ST_NN NOT NULL;
ALTER TABLE SUPPLIERS ADD NAME_SUS VARCHAR2(35) CONSTRAINT SUS_NA_NN NOT NULL;
ALTER TABLE SUPPLIERS ADD CITY INT CONSTRAINT SU_CI_NN NOT NULL;

ALTER TABLE SUPPLIERS DROP COLUMN STATUS;
ALTER TABLE SUPPLIERS RENAME COLUMN CITY TO CITY_ADDRESSES;
ALTER TABLE SUPPLIERS MODIFY CITY_ADDRESSES VARCHAR2(50);

SELECT * FROM SUPPLIERS;

DESC SUPPLIERS;

```

Informes | Salida de Script | Resultado de la Consulta | Tareas terminadas en 0.492 segundos

Table SUPPLIERS alterado.

Nombre	NotNull?	Tipo
SUPPLIER_CODE		CHAR(8)
SUPPLIER_DETAILS	NOT NULL	VARCHAR2(30)
NAME_SUS	NOT NULL	VARCHAR2(35)
CITY_ADDRESSES	NOT NULL	VARCHAR2(50)

I Línea 41 Columna 16 | Inserir | Modificado | Windows: CF | 01:02 p.m. 25/08/2022

Figura 31: *Modifying the suppliers table*

Oracle SQL Developer : D:\ingsoftware\semestre5\LaboratorioBaseDeDatosII\Scripts_Sql\PRACTICA2.sql

Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda

Conexiones | PRACTICA2.sql | PRACTICA2

Hojas de Trabajo - Generador de Consultas

```

SELECT * FROM SUPPLIERS;
SELECT * FROM PRODUCT_HIERARCHY;
SELECT * FROM PRODUCT_SUPPLYERS;

-- ADD PRIMARY KEY'S
ALTER TABLE PRODUCT_HIERARCHY ADD CONSTRAINT
    PRO_HIERARCHY_PK PRIMARY KEY (PRODUCT_CODE);

ALTER TABLE SUPPLIERS ADD CONSTRAINT SUPPLIERS_PK PRIMARY KEY (SUPPLIER_CODE);

-- ADD FOREIGN KEY'S
ALTER TABLE PRODUCT_HIERARCHY ADD CONSTRAINT
    PRO_HIERARCHY_FK FOREIGN KEY (PARENT_PRODUCT_CODE)
    REFERENCES PRODUCT_HIERARCHY (PRODUCT_CODE);

ALTER TABLE PRODUCT_SUPPLYERS ADD CONSTRAINT PROS_FK FOREIGN KEY (PRODUCT_CODE)
    REFERENCES PRODUCT_HIERARCHY (PRODUCT_CODE);

ALTER TABLE PRODUCT_SUPPLYERS ADD CONSTRAINT SUS_PROS_FK FOREIGN KEY (SUPPLIER_CODE)
    REFERENCES SUPPLIERS (SUPPLIER_CODE);

```

Salida de Script | Resultado de la Consulta | Tareas terminadas en 0.094 segundos

Table PRODUCT_SUPPLYERS alterado.

Table PRODUCT_SUPPLYERS alterado.

I Línea 46 Columna 1 | Inserir | Windows: CF | 01:43 p.m. 25/08/2022

Figura 32: *Add primary and foreign keys*

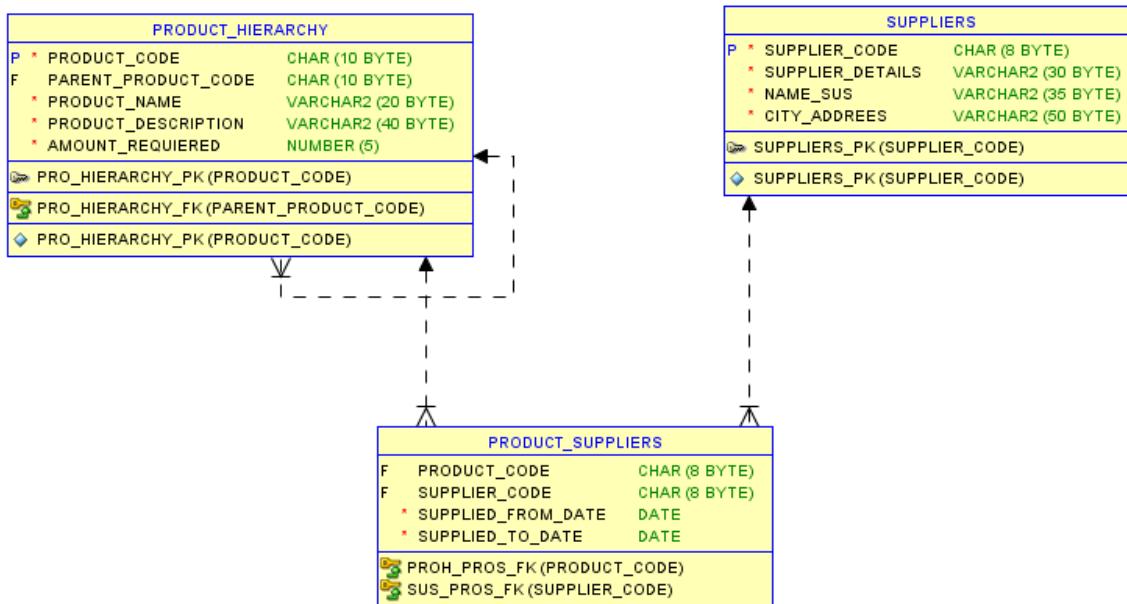


Figura 33: Relational Diagram of activity six

Activity 7:

The following **figures 34 and 35** show the DDL statements for creating the tables based on the ER diagram of the activity.

Figure 36 shows the statements that were used to add the primary key and foreign key constraints to the tables.

The screenshot shows the Oracle SQL Developer interface. On the left is the 'Object Navigator' pane, which lists several schemas: EJEMPLO, EJEMPLO2, and EJEMPLO3. The 'EJEMPLO2' schema is expanded, showing tables like BRANCH, EMPLOYEE, JOB, and WORKS_ON, along with various objects such as sequences, triggers, and views. The main workspace is titled 'Hoja de Trabajo - Generador de Consultas' and contains the following DDL code:

```

-- Práctica 02, Actividad 7

--CREATE TABLE EMPLOYEE (
    EMPLOYEE_ID NUMBER(6),
    FIRST_NAME VARCHAR2(35) CONSTRAINT EMPLOYEE_FIRST_NAME_NN NOT NULL,
    LAST_NAME VARCHAR2(35) CONSTRAINT EMPLOYEE_LAST_NAME_NN NOT NULL,
    EMAIL VARCHAR2(55) CONSTRAINT EMPL_EMAIL_NN NOT NULL CONSTRAINT EMPLOYEE_EMAIL_UK UNIQUE,
    PHONE_NUMBER VARCHAR2(10) CONSTRAINT EMPL_PHONE_NUMBER_NN NOT NULL,
    HIRE_DATE DATE CONSTRAINT EMPL_HIRE_DATE_NN NOT NULL,
    SALARY NUMBER(8,2) CONSTRAINT EMPL_SALARY_NN NOT NULL
);

SELECT * FROM EMPLOYEE;

CREATE TABLE JOB (
    JOB_ID NUMBER(4),
    JOB_NAME VARCHAR2(35) CONSTRAINT JOB_JNAME_NN NOT NULL,
    MIN_SALARY NUMBER(6) CONSTRAINT JOB_MIN_SALARY_NN NOT NULL,
    MAX_SALARY NUMBER(6) CONSTRAINT JOB_MAX_SALARY_NN NOT NULL
);

SELECT * FROM JOB;

CREATE TABLE BRANCH (
    BRANCH_ID NUMBER(4),
    BRANCH_NAME VARCHAR2(45) CONSTRAINT BRANCH_BNAME_NN NOT NULL,
    BRANCH_LOCATION VARCHAR2(55) CONSTRAINT BRANCH_BLOCATION_NN NOT NULL
);

SELECT * FROM BRANCH;

CREATE TABLE WORKS_ON (
    EMPLOYEE_ID NUMBER(6),
    JOB_ID NUMBER(4),
    BRANCH_ID NUMBER(4),
    START_DATE DATE CONSTRAINT WORKS_ON_SDATE_NN NOT NULL,
    END_DATE DATE CONSTRAINT WORKS_ON_EDATE_NN NOT NULL
);

SELECT * FROM WORKS_ON;

```

At the bottom of the code area, it says 'Se ha guardado: D:\ingsoftware\semestre5\LaboratorioBaseDeDatos\Scripts_Sql\EJEMPLO2.sql'.

Figura 34: *DDL statements for creating tables.*

This screenshot is similar to Figure 34, showing the Oracle SQL Developer interface with the 'EJEMPLO2' schema selected in the Object Navigator. The main workspace contains the following DDL code:

```

CREATE TABLE BRANCH (
    BRANCH_ID NUMBER(4),
    BRANCH_NAME VARCHAR2(45) CONSTRAINT BRANCH_BNAME_NN NOT NULL,
    BRANCH_LOCATION VARCHAR2(55) CONSTRAINT BRANCH_BLOCATION_NN NOT NULL
);

SELECT * FROM BRANCH;

CREATE TABLE WORKS_ON (
    EMPLOYEE_ID NUMBER(6),
    JOB_ID NUMBER(4),
    BRANCH_ID NUMBER(4),
    START_DATE DATE CONSTRAINT WORKS_ON_SDATE_NN NOT NULL,
    END_DATE DATE CONSTRAINT WORKS_ON_EDATE_NN NOT NULL
);

SELECT * FROM WORKS_ON;

CREATE TABLE MANAGER (
    MANAGER_ID NUMBER(6),
    FIRST_NAME VARCHAR2(35) CONSTRAINT MNG_FRIST_NAME_NN NOT NULL,
    LAST_NAME VARCHAR2(35) CONSTRAINT MNG_LAST_NAME_NN NOT NULL,
    EMAIL VARCHAR2(55) CONSTRAINT MNG_EMAIL_NN NOT NULL,
    PHONE_NUMBER VARCHAR2(10) CONSTRAINT MNG_PHONE_NUMBER_NN NOT NULL
);

SELECT * FROM MANAGER;

CREATE TABLE MANAGES (
    DATE_ASSIGNMENTDATE DATE,
    MANAGER_ID NUMBER(6),
    START_DATE DATE
);

SELECT * FROM MANAGES;

```

At the bottom of the code area, it says 'Se ha guardado: D:\ingsoftware\semestre5\LaboratorioBaseDeDatos\Scripts_Sql\EJEMPLO2.sql'.

Figura 35: *DDL statements for creating tables.*

```
-- ADD PRIMARY KEYS
ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE_PK PRIMARY KEY (EMPLOYEE_ID);
ALTER TABLE JOB ADD CONSTRAINT JOB_PK PRIMARY KEY (JOB_ID);
ALTER TABLE BRANCH ADD CONSTRAINT BRANCH_PK PRIMARY KEY (BRANCH_ID);
ALTER TABLE WORKS_ON ADD CONSTRAINT WORKS_ON_PK PRIMARY KEY (START_DATE);
ALTER TABLE MANAGER ADD CONSTRAINT MANAGER_PK PRIMARY KEY (MANAGER_ID);
ALTER TABLE MANAGES ADD CONSTRAINT MANAGES_PK PRIMARY KEY (DATE_ASSIGNMENTDATE);

-- ADD FOREIGN KEYS
ALTER TABLE WORKS_ON ADD CONSTRAINT EMPL_WORKS_ON_FK FOREIGN KEY (EMPLOYEE_ID)
    REFERENCES EMPLOYEE(EMPLOYEE_ID);

ALTER TABLE WORKS_ON ADD CONSTRAINT JOB_WORKS_ON_FK FOREIGN KEY (JOB_ID)
    REFERENCES JOB(JOB_ID);

ALTER TABLE WORKS_ON ADD CONSTRAINT BRANCH_WORKS_ON_FK FOREIGN KEY (BRANCH_ID)
    REFERENCES BRANCH(BRANCH_ID);

ALTER TABLE MANAGES ADD CONSTRAINT MANAGES_MANAGER_FK FOREIGN KEY (MANAGER_ID)
    REFERENCES MANAGER(MANAGER_ID);

ALTER TABLE MANAGES ADD CONSTRAINT MANAGES_WORKS_ON_FK FOREIGN KEY (START_DATE)
    REFERENCES WORKS_ON(START_DATE);
```

Figura 36: *Adding constraints, primary keys and foreign keys.*

Figures 37 and 38 show the reaction diagrams and the logic diagram generated with SQL Data Modeler.

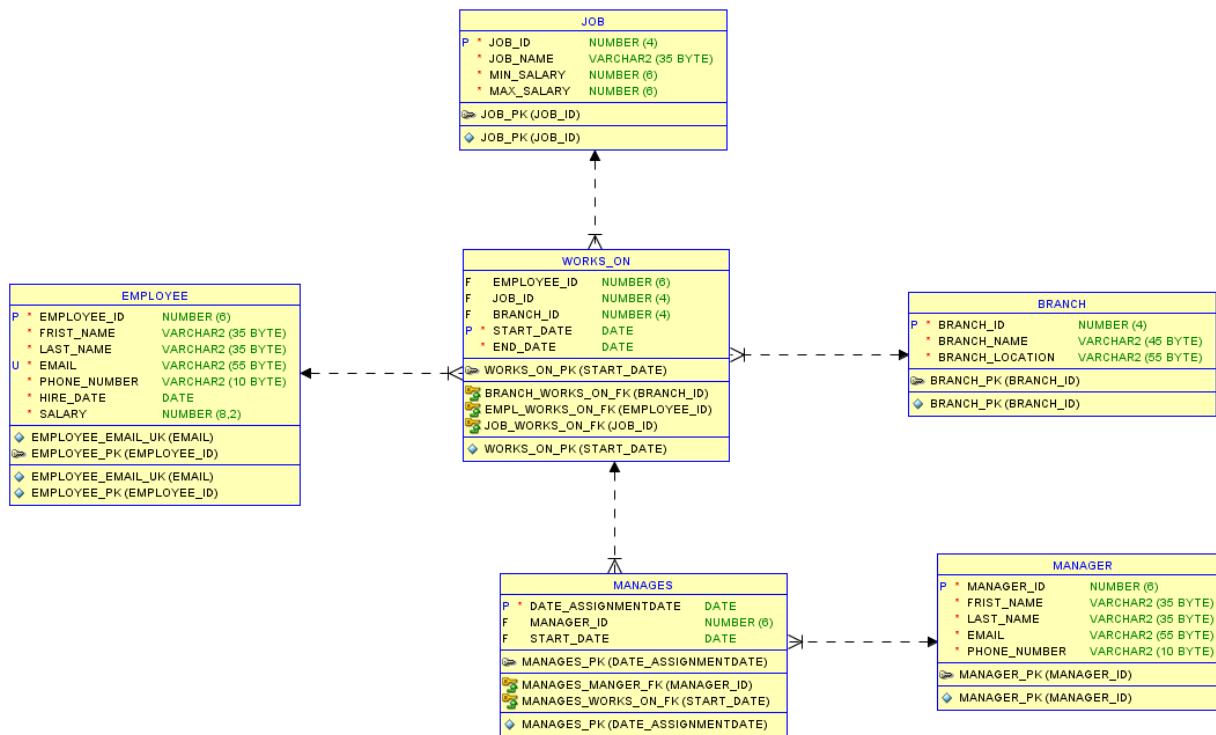


Figura 37: Relational model of activity seven.

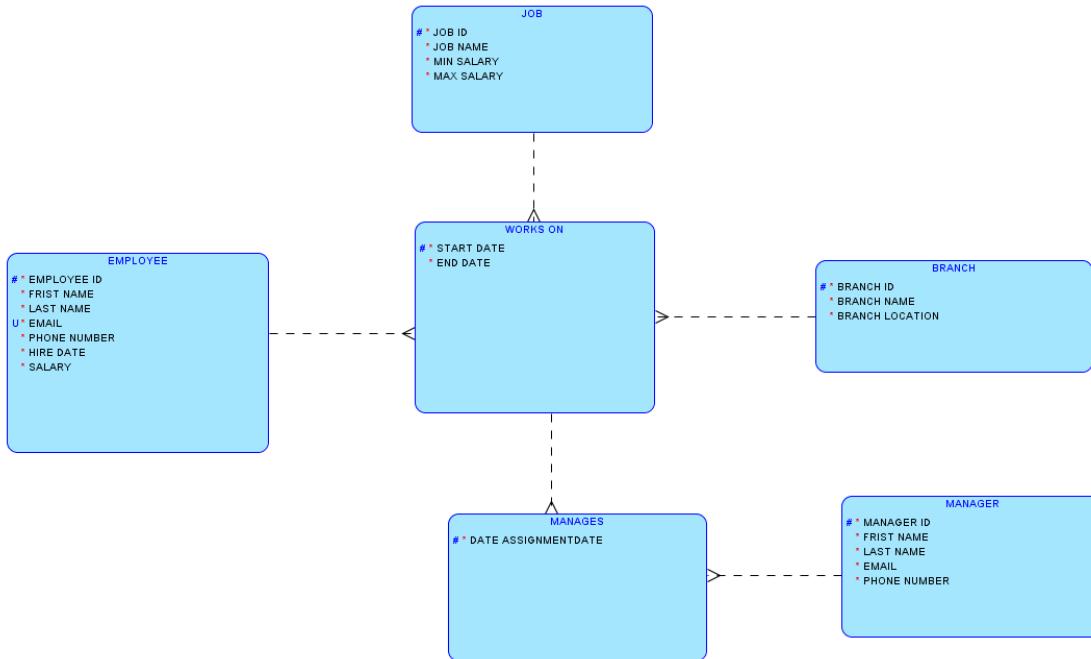


Figura 38: Reverse engineered logic model.

Activity 8: This activity number eight is not finished.

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)	NO 95 %

5. Conclusion

The Oracle DDL language is transcendental in the handling of SQL statements at the level of both administrator and database programmer, since it allows the definition of database schemes regardless of the platform used to generate it.

This practice was very important for me since it helped me to remember and reinforce my knowledge in the use of DDL statements.