Autonomous University of Zacatecas

Academic Unit of Electrical Engineering

Academic Program of Software Engineering

Database Systems Laboratory II Practice 2 - DDL

Professor:
Aldonso Becerra Sánchez

Student:
Daniel Alejandro Morales Castillo

1 Introduction

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.

The DDL (Data definition Language) is a language provide by the database management system and it allows the programmers do fundamental works handling of SQL statements. This practice works for review some theoretical concepts

During the week we saw topics such as constraints and alter tables, these topics are fundamental when creating our databases, our tables and this practice will surely help us to learn more about it

2 Development

Activity 1

Explain the reason for your answer. Categorize the Main Database Objects

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. Explain the reason for your answer.

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

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

A. It will be an orphaned table, without a schema.
B. The creation will fail.
C. It will be in the SYS schema.
D. It will be in the schema of the user creating it.
E. It will be in the PUBLIC schema.

My answer: A AND D

When the user going to create tables need a connection to a data base, a data base connextion has an user and the user is the schema owner so the table will be created in the user schema because you are working with a connection with that user. So if you have not a connection create table will fail, it depends of the connection

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.

A. Index
B. PL/SQL stored procedure
C. Synonym
D. Table
E. View

My answer: B

Objects in a database and namespace and I found that the index, table, synonym, and view share the same namespace, so these objects are not in the same namespace with PL / SQL stored procedure

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

A. create table “WHERE” (col1 date);
B. create table “mincase” (col1 date);
C. create table 1var (col1 date);
D. create table var1 (col1 date);
E. create table delete (col1 date);

My answer: A AND C

I chose A because you can’t create a table using (”) as the first character and if you don’t use (”), then” WHERE ”is a reserved word, so it can’t be used it is like a name. About the letter C is because the first character of the name of a table must be a letter so you can’t create a table called 1var.

List the Data Types that Are Available for Columns

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

A. BLOB
B. CHAR
C. LONG
D. NUMBER
E. RAW
F. VARCHAR2

My answer: A,C,D,E AND F

The char data type has a fixed length, the others handle a variable length, and wide, do your research to answer the question

5. Study these statements: create table tab1 (c1 number(1), c2 date),alter session set $\mathrm{nls}_{d}ate_{f}ormat =' dd-mm-yy'; insertintotab1values(2.2,'29-07-09');, Willtheinsertsucceed?(Choosethebestanswer)$

A. The insert will fail because the 8.8 is too long.
B. The insert will fail because the ’29-07-09’ is a string, not a date.
C. The insert will fail for both reasons A and B.
D. The insert will succeed.

My answer:D

All the passed sentences would work fine because they are all correct, at the time of table creation I have no problem, in the alternate session it is fine and when I do the insertion of the values it’s ok because both values were sending correctly.

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

A. CHAR

B. FLOAT

C. INTEGER

D. STRING

My answer: D

the data type "String" does not exist, when you need to use a data type to represent text or words, we can use char or varchar2 and it is common to use them, float and integer are data types to use with numbers and Researched about Oracle and String datatypes not showing up as an intern type of data.

7. Consider this statement: create table t1 as select * from employees where 9=4; What will be the result? (Choose the best answer.)

A. There will be an error because of the impossible condition.

B. No table will be created because the condition returns FALSE.

C. The table T1 will be created but no rows inserted because the condition returns FALSE.

D. The table T1 will be created and every row in EMPLOYEES inserted because the condition returns a NULL as a row filter

My answer: C

The table t1 will be created by the subquery because the sentence is fine and the condition is somewhat strange but not bad, so the condition will be false and the table will be created but nothing will be inserted since it lacks logic

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.)

A. The new table will have no constraints, because constraints are not copied when creating tables with a subquery.

B. All the constraints on TAB will be copied to NEWTABLE.

C. Primary key and unique constraints will be copied but not check and not null constraints.

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

E. All constraints will be copied, except foreign key constraints.

My answer: D

I did tests with the creation of tables by subqueries and when the tables were created I used the sentence "describe table" to see the columns and constraints as primary keys and not null, we know that a primary key is unique and it is

not null, but when I created tables with subqueries the primary key constraint it didn't show up but the non-null constraint on other columns did. Then you can see that all the constraints are copied, except the primary and unique keys apparently

Explain How Constraints Are Created at the Time of Table Creation

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

A. CHECK
B. NOT NULL
C. PRIMARY KEY
D. UNIQUE

My answer: C AND D

Indexes are used to search and find something in a table quickly, then we can see that in the constraints primary and unique key are used to identify something by the attribute then it has the constraint, so when we are looking for a record by indexing the primary key and unique constraint can identify the exact record

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)

A. The whole transaction will be rolled back.

B. The second statement will be rolled back completely, and the first will be committed.

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

D. Only the one update that caused the violation will be rolled back; everything else will be committed.

E. Only the one update that caused the violation will be rolled back; everything else will remain uncommitted.

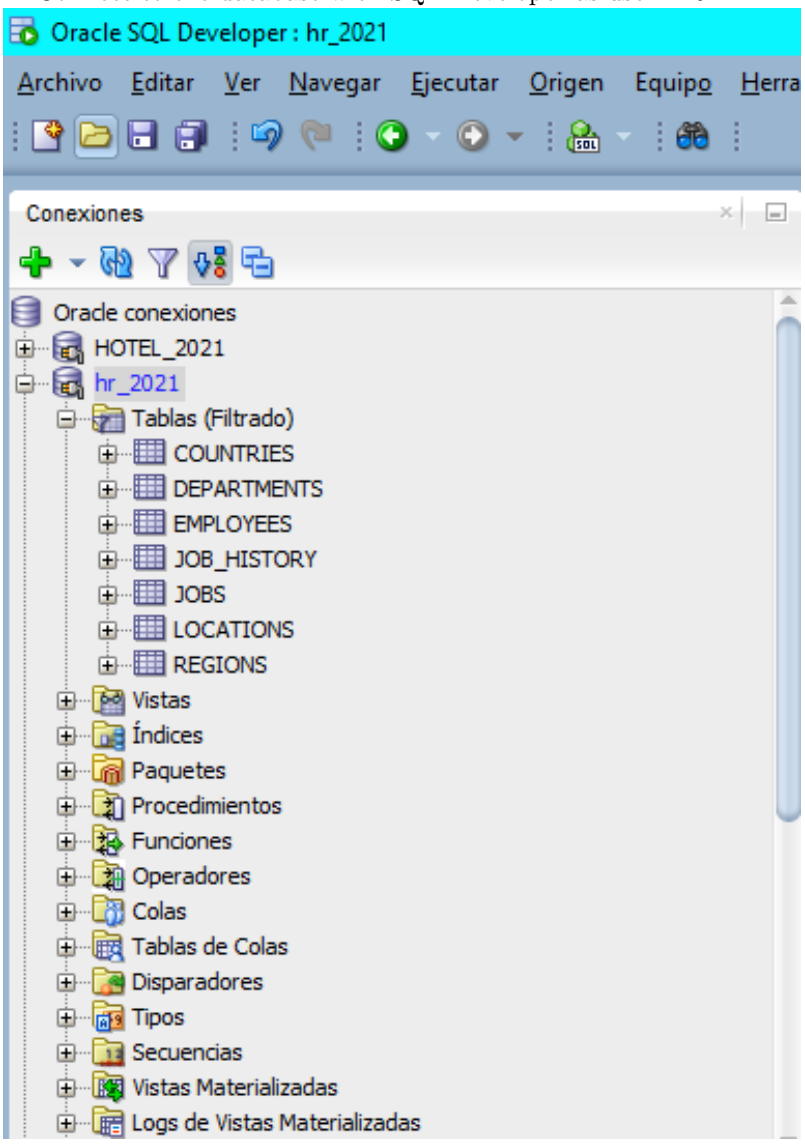
My answer: B

If two statements will be executed and the first will be successful and the second one does not think that only the former will be fully engaged because the The sentences will be executed in order, the second cannot be executed because will show an error when violating restrictions.

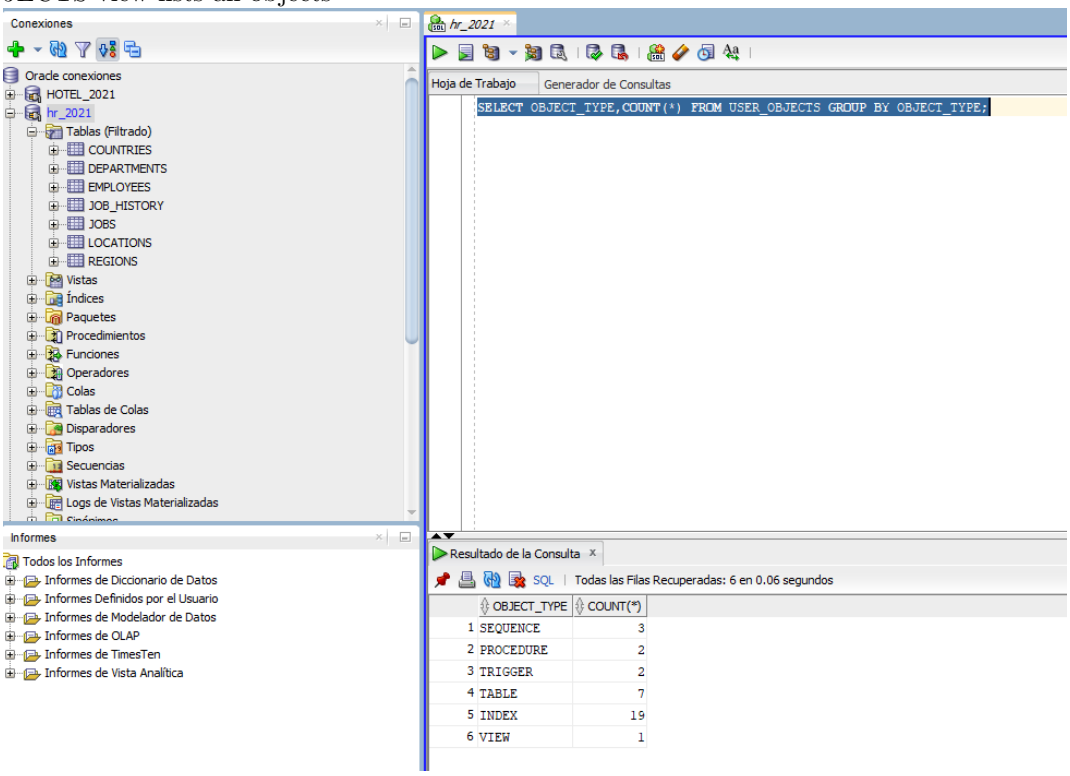
Activity 2:

EXECUTE THE FOLLOWING SENTENCES. INCLUDE AN OUTPUT IMAGE FOR EACH ONE.

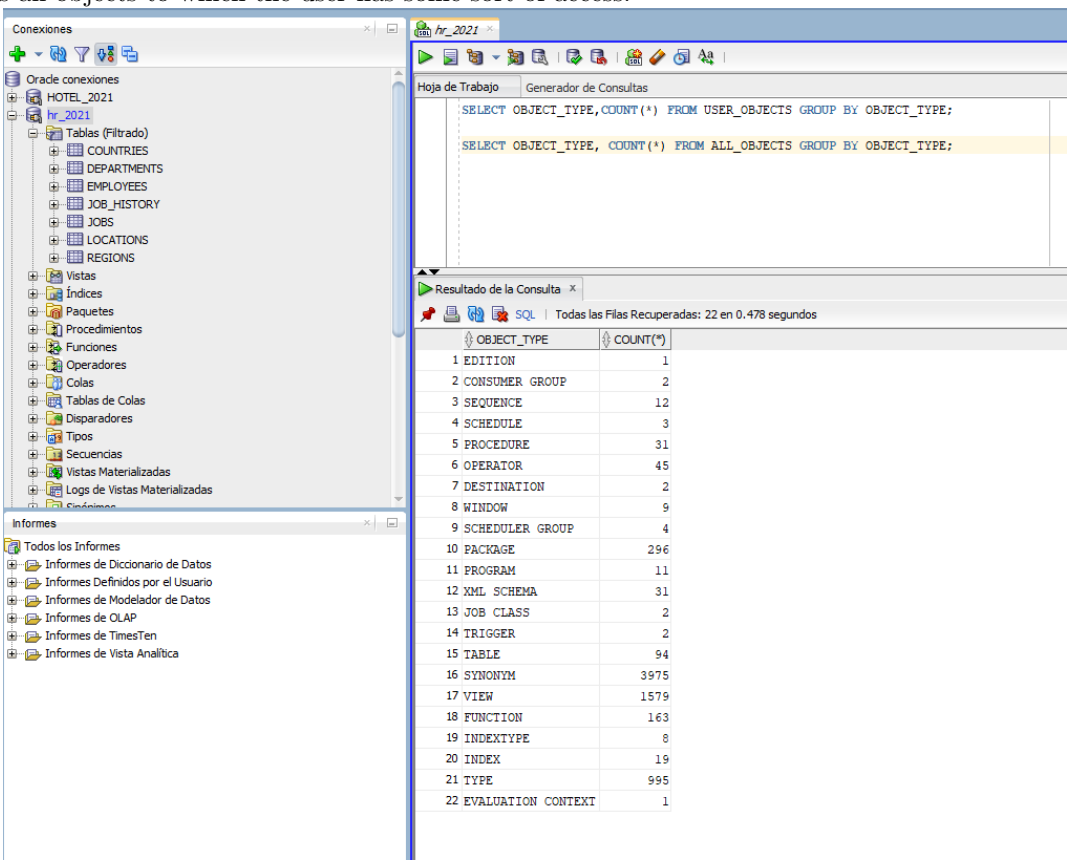
Determine What Objects Are Accessible to Your Session
1. Connect to the database with SQL Developer as user HR.



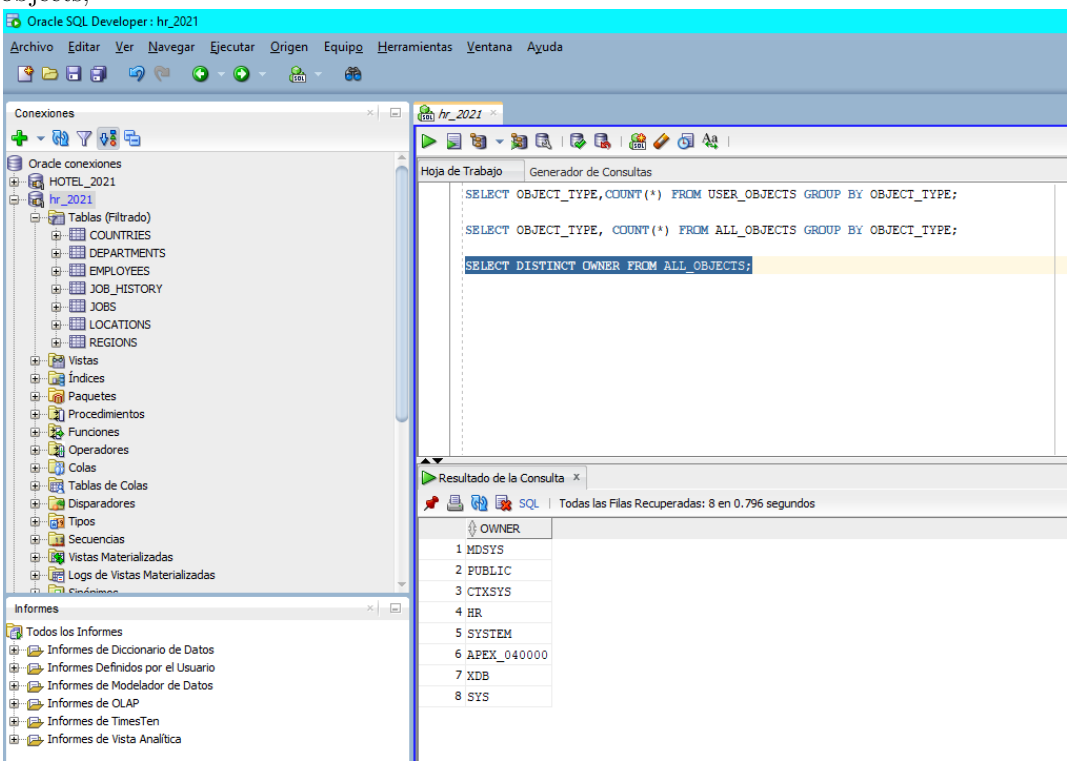
2. Determine how many objects of each type are in the HR schema: select object type,count(*) from user objects group by object type; The USER OBJECTS view lists all objects



3. Determine how many objects in total HR has permissions on: select object type,count(*) from all objects group by object type; The ALL OBJECTS view lists all objects to which the user has some sort of access.



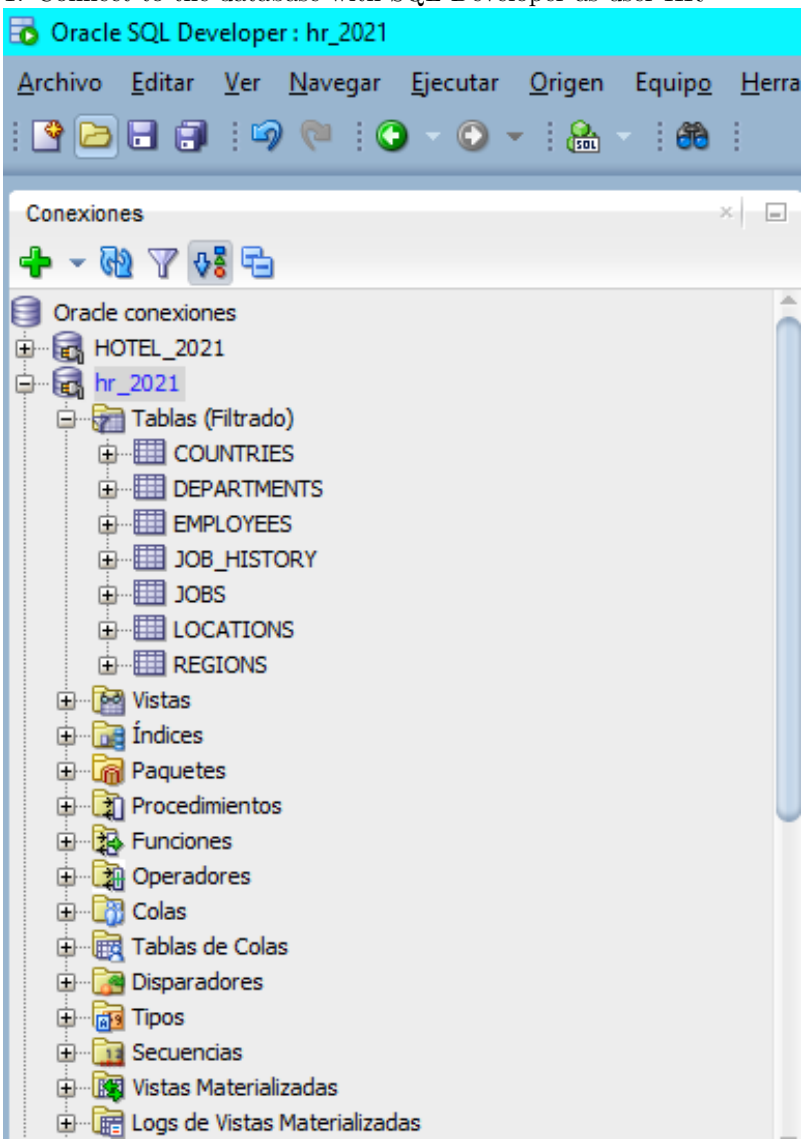
4. Determine who owns the objects HR can see: select distinct owner from all objects;



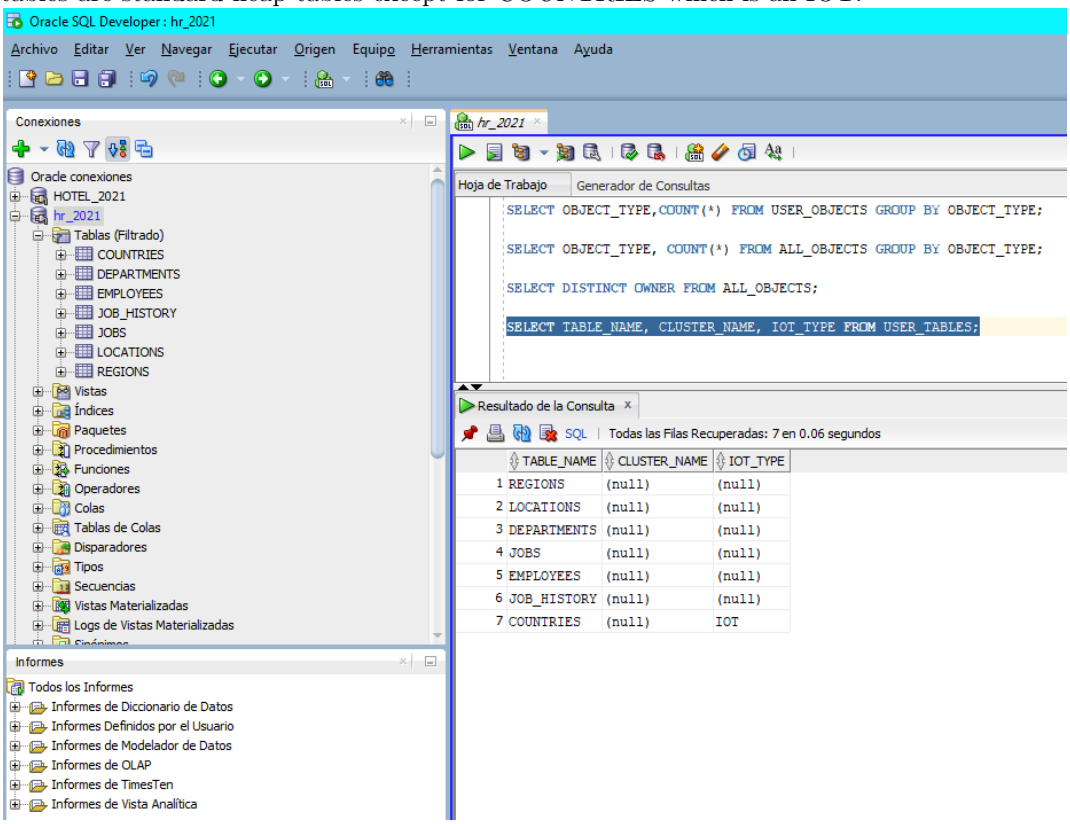
Investigate Table Structures

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

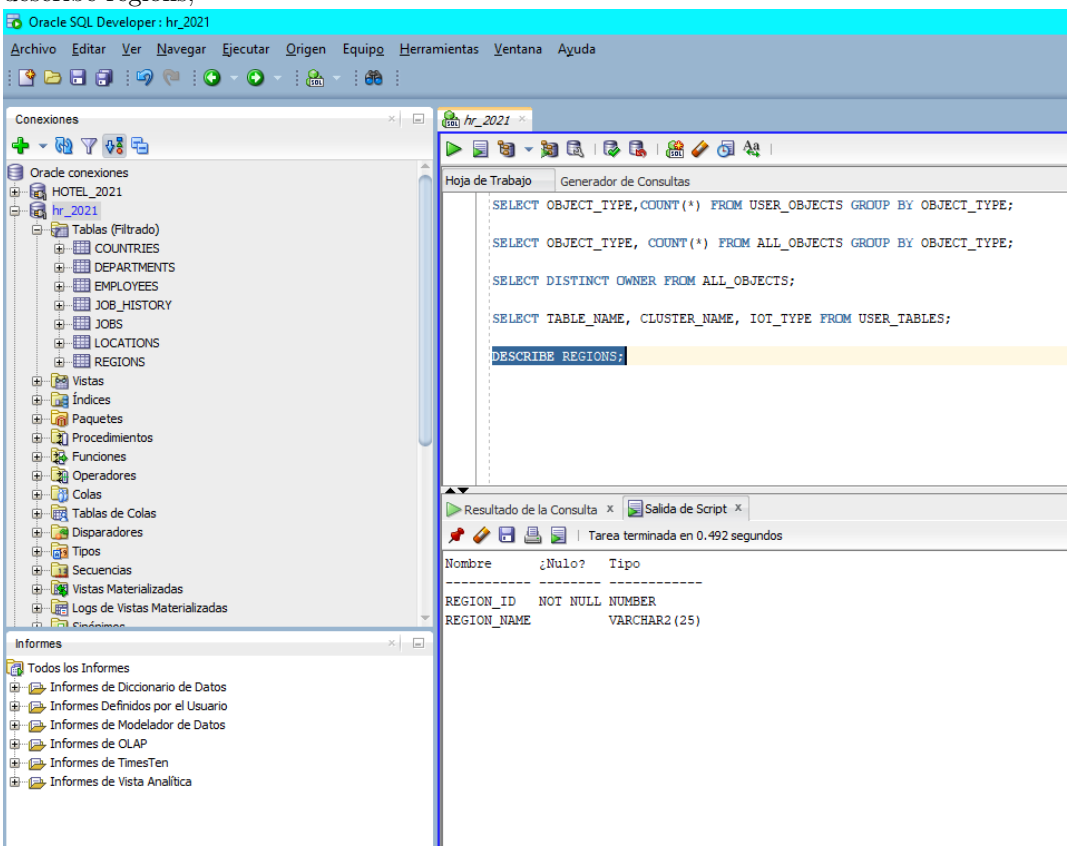
1. Connect to the database with SQL Developer as user HR



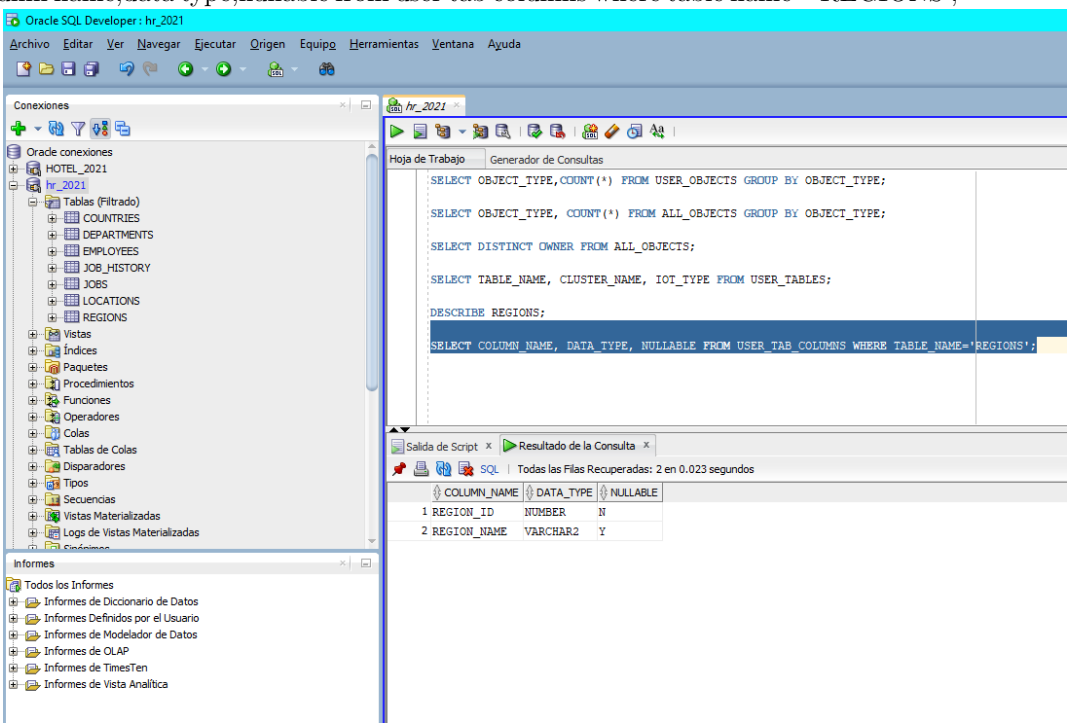
2. Determine the names and types of tables that exist in the HR schema: select table name,cluster name,iot type from user tables; Clustered tables and index organized tables (IOTs) are advanced table structures. In the HR schema, all tables are standard heap tables except for COUNTRIES which is an IOT.



3. Use the DESCRIBE command to display the structure of a table:
describe regions;



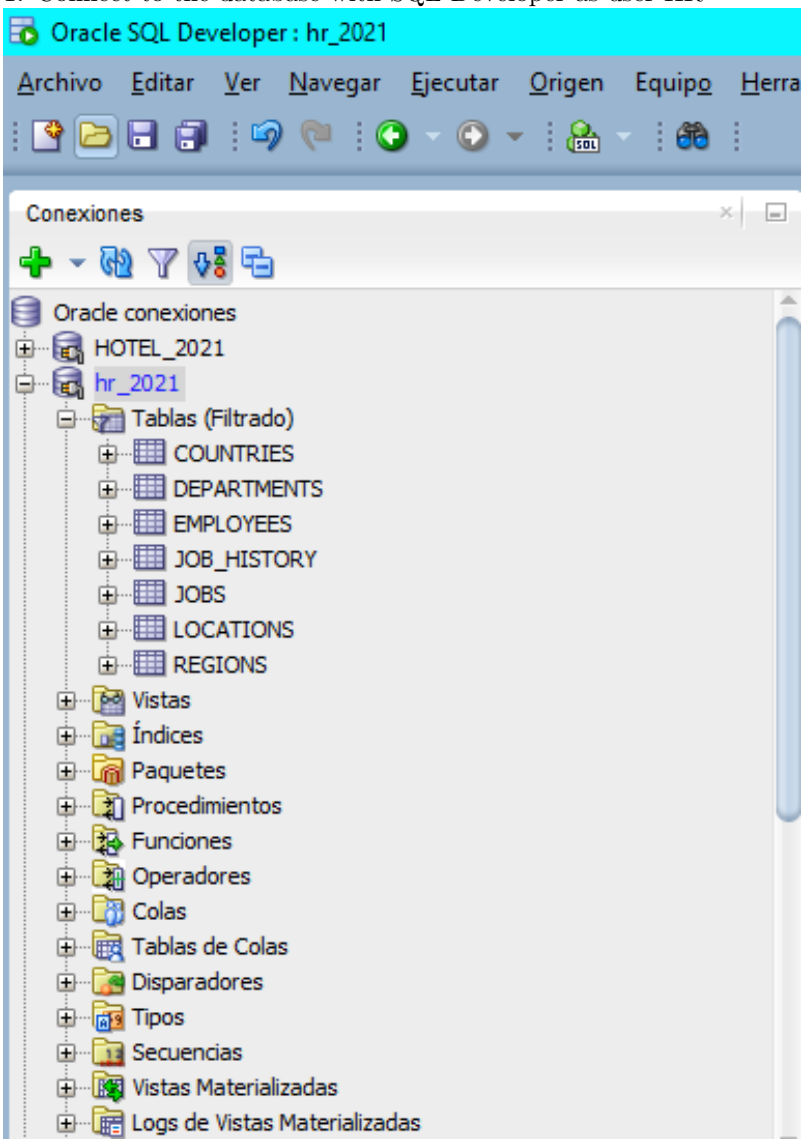
4. Retrieve similar information by querying a data dictionary view: select column name,data type,nullable from user tab columns where table name=’REGIONS’;



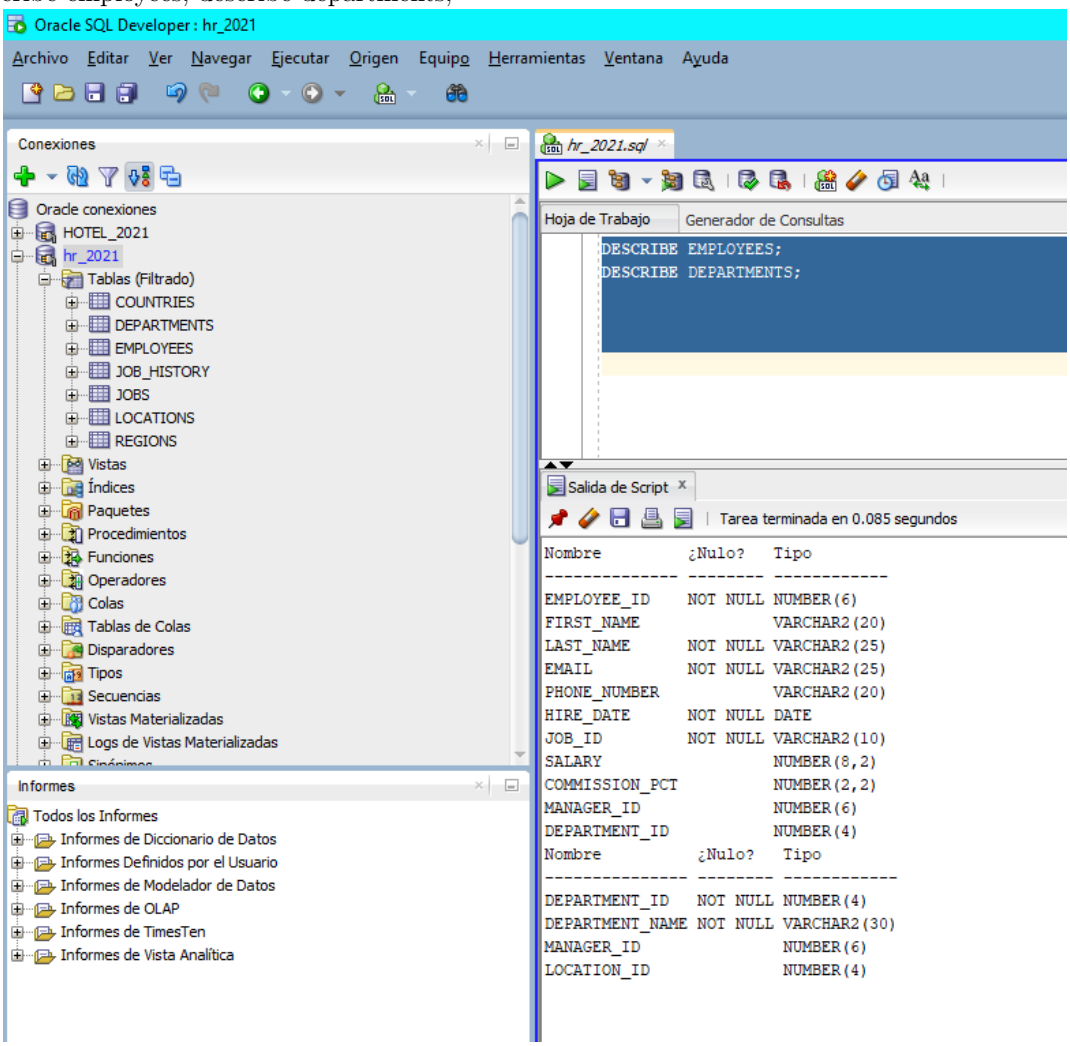
Investigate the Data Types in the HR schema

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

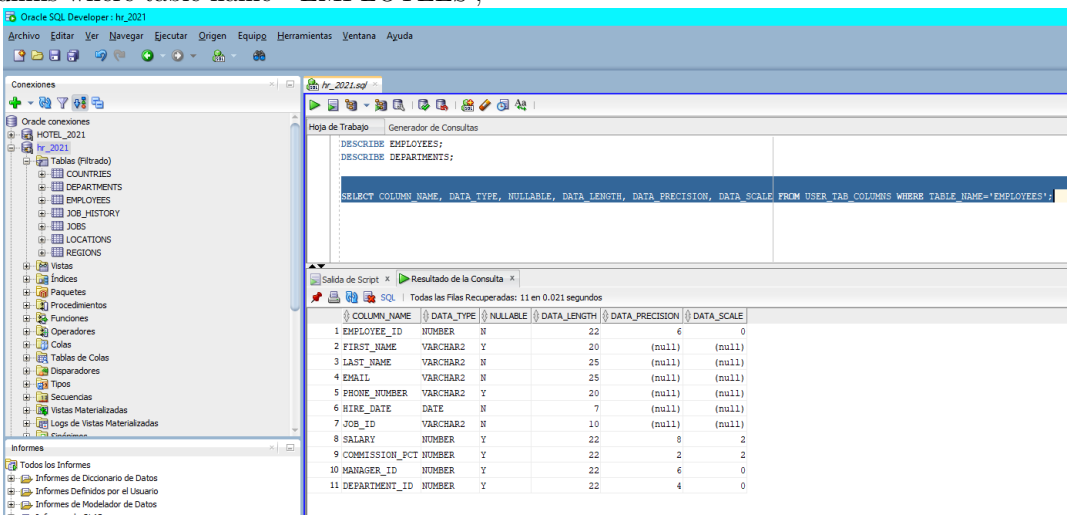
1. Connect to the database with SQL Developer as user HR



2. Use the DESCRIBE command to show the data types in some tables: describe employees; describe departments;



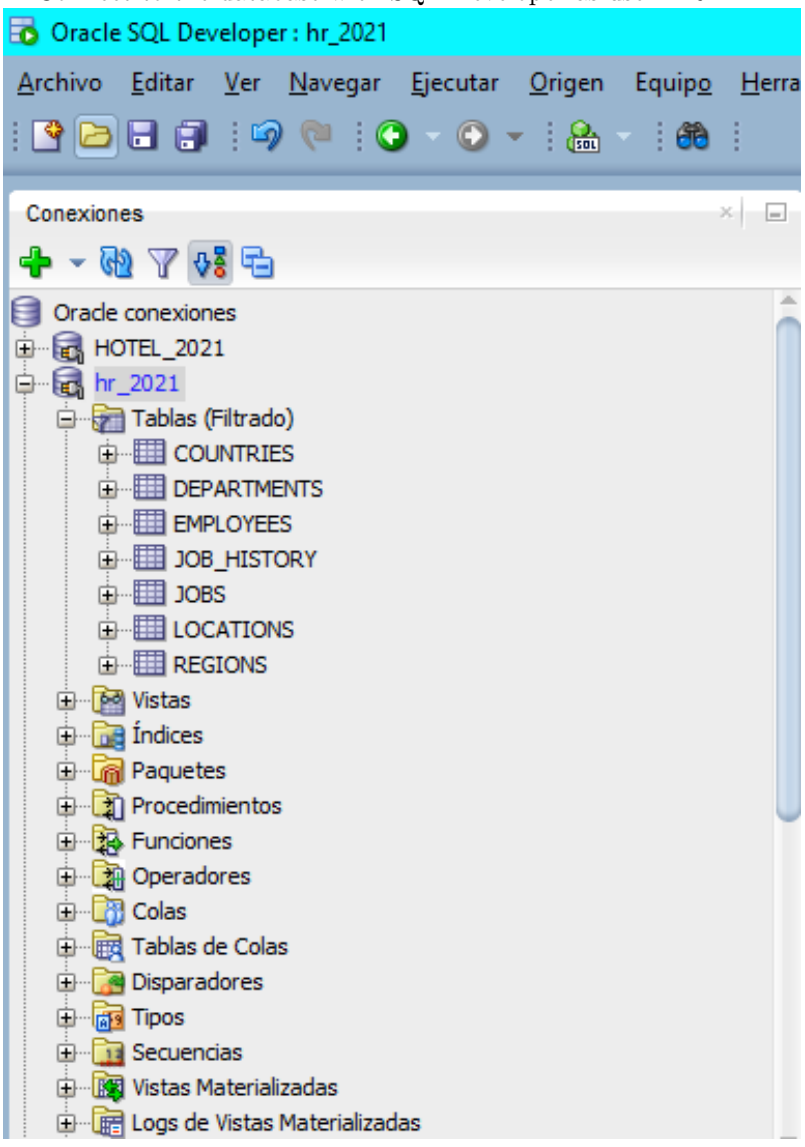
3. Use a query against a data dictionary view to show what columns make up the EMPLOYEES table, as the DESCRIBE command would: select column name,data type,nullable,data length,data precision, data scale from user tab columns where table name=’EMPLOYEES’;



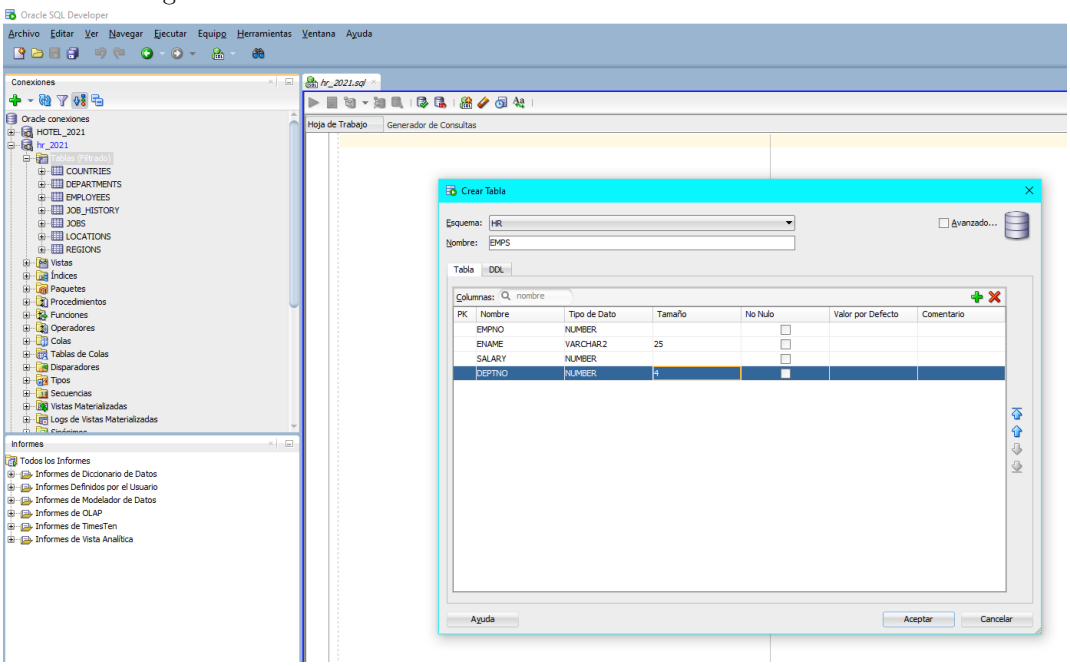
Create Tables

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.

1. Connect to the database with SQL Developer as user HR



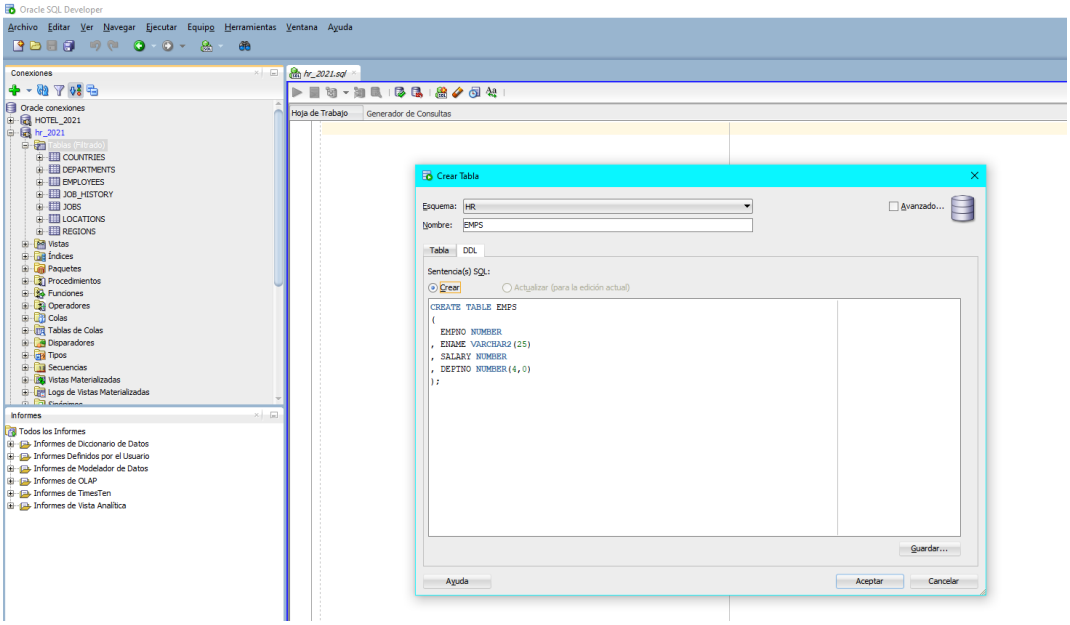
2. Right-click the Tables branch of the navigation tree, and click New Table
3. Name the new table EMPS, and use the Add Column button to set it up as in the following illustration:



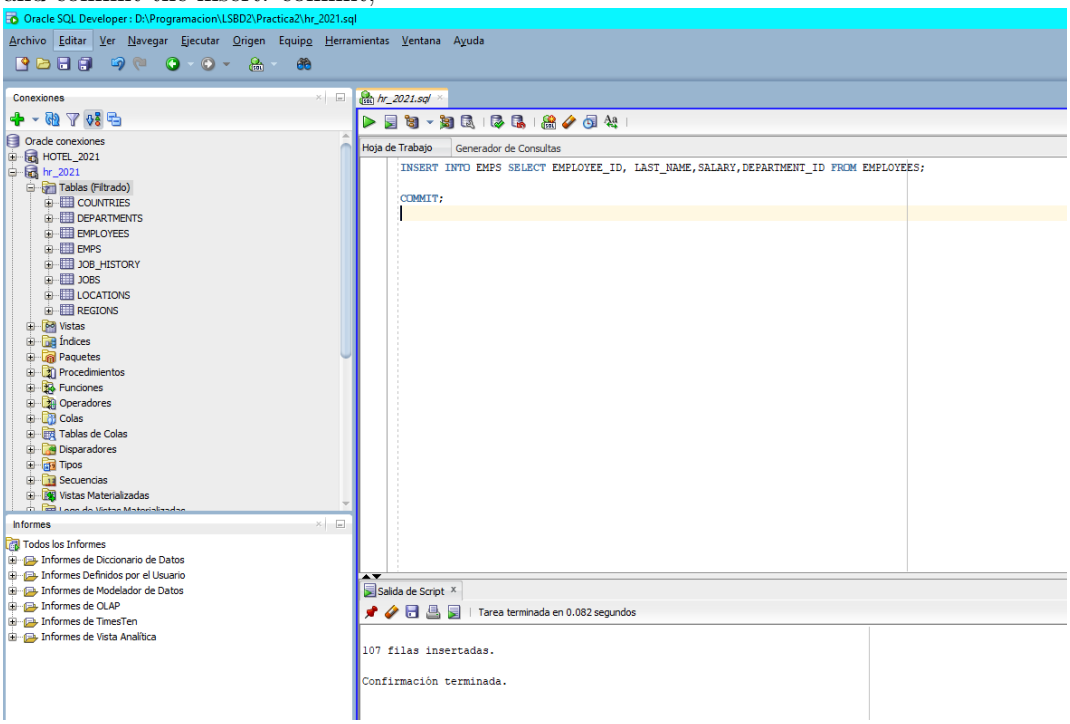
4. Click the DDL tab to see if the statement that has been constructed. It should look like this:

CREATE TABLE EMPS(
EMPNO NUMBER,
ENAME VARCHAR2(25),
SALARY NUMBER,
DEPTNO NUMBER(4, 0));

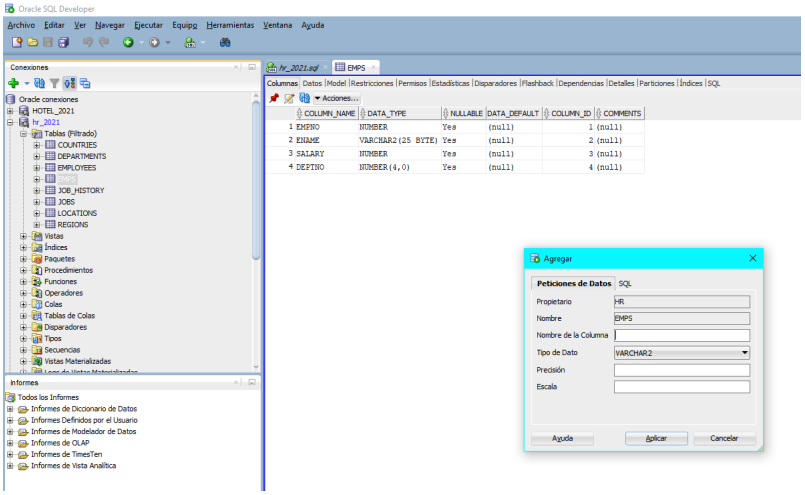
Return to the Table tab (as in the preceding illustration) and click OK to create the table



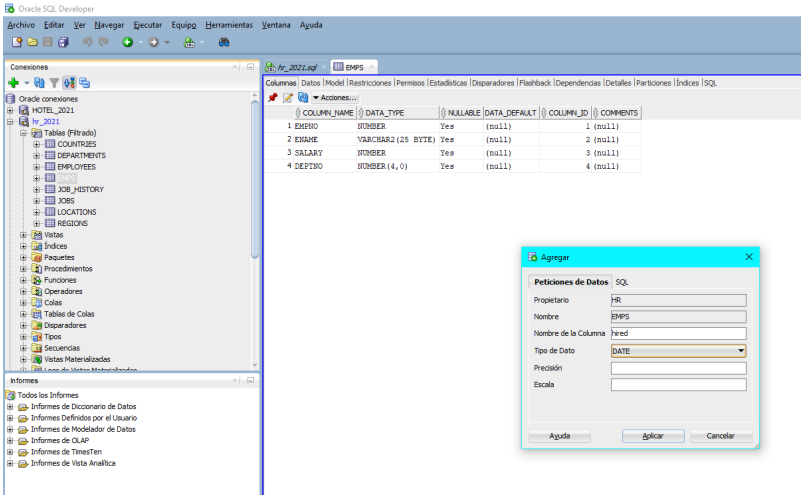
5. Run this statement:
insert into emps select employee id,last name, salary,department id from employees;
and commit the insert: commit;



6. Right-click the EMPS table in the SQL Developer navigator, click Column and Add.

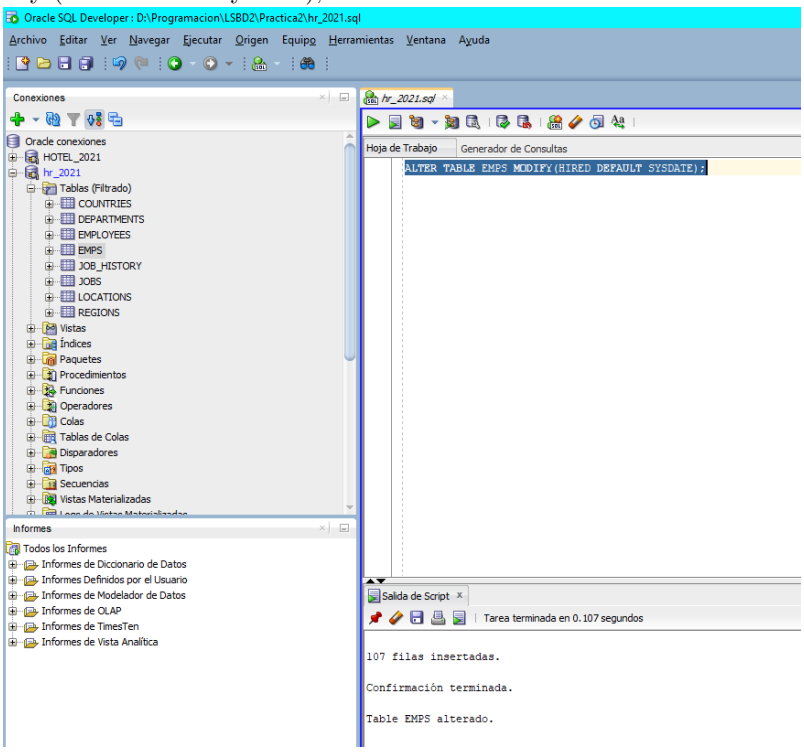


7. Define a new column HIRED, type DATE, as in the following illustration below; and click Apply to create the column.

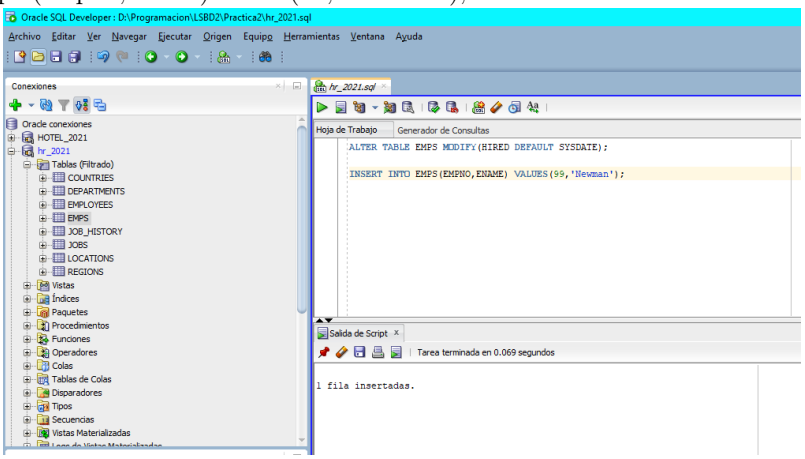


8. Connect to the database as HR. I was already connected as HR

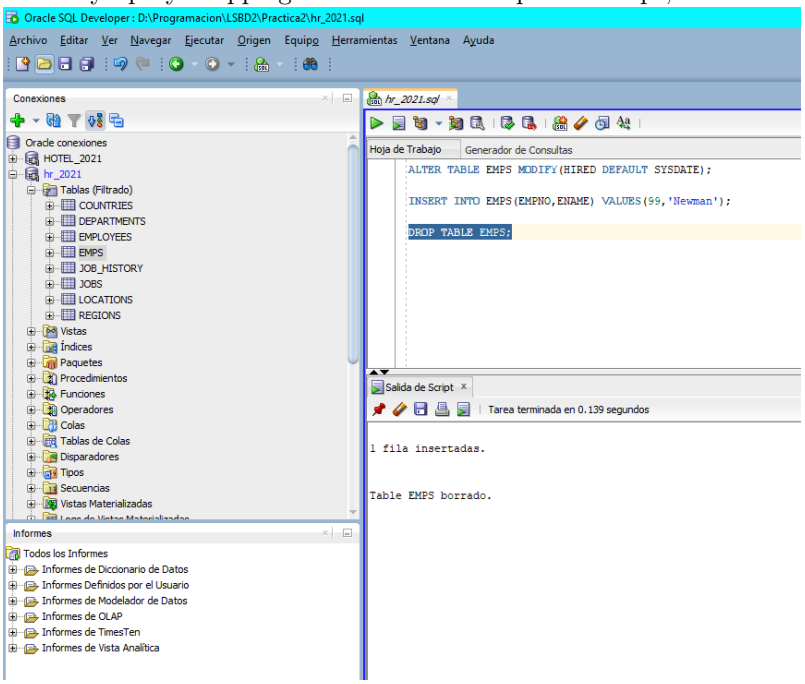
9. Define a default for HIRED column in the EMPS table: alter table emps modify (hired default sysdate);



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: insert into emps (empno,ename) values(99,'Newman');



11. Tidy up by dropping the new table: drop table emps;

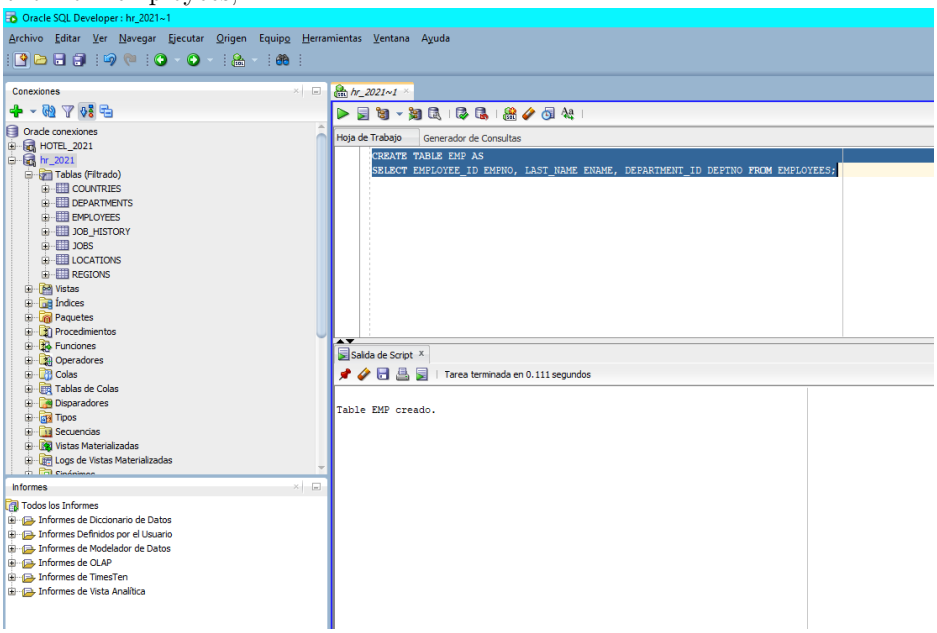


Work with Constraints

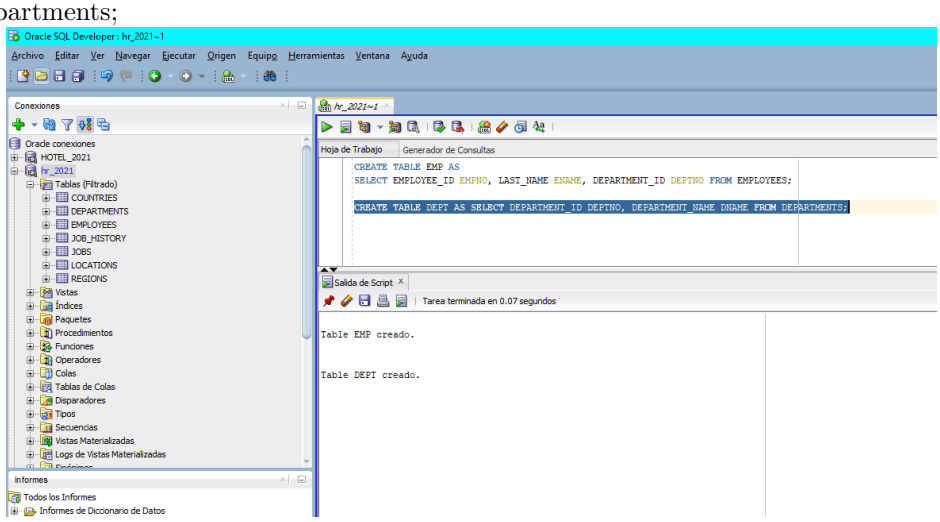
SQL Developer to create tables, add constraints, and demonstrate their use.

1. Connect to the database with SQL Developer as user HR
I was already as user HR

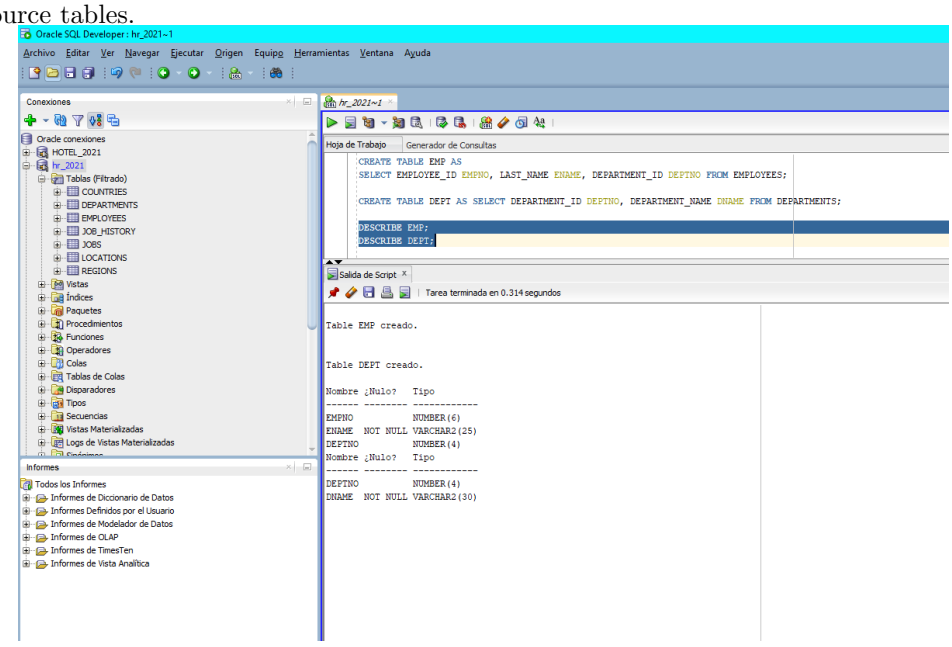
2. Create a table EMP as a copy of some columns from EMPLOYEES: create table emp as select employee id empno, last name ename, department id deptno from employees;



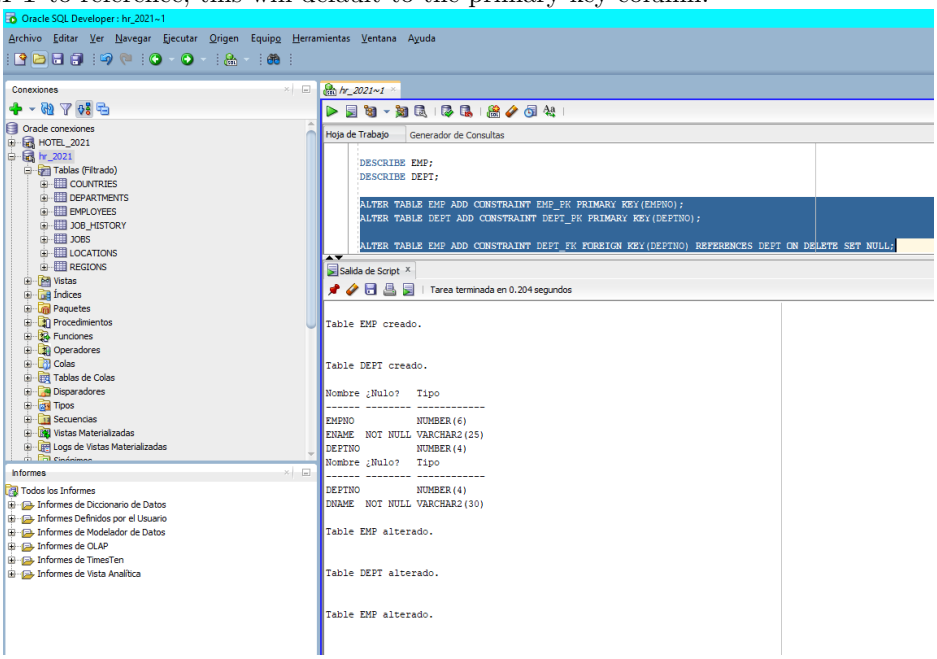
3. Create a table DEPT as a copy of some columns from DEPARTMENTS: create table dept as select department id deptno, department name dname from departments;



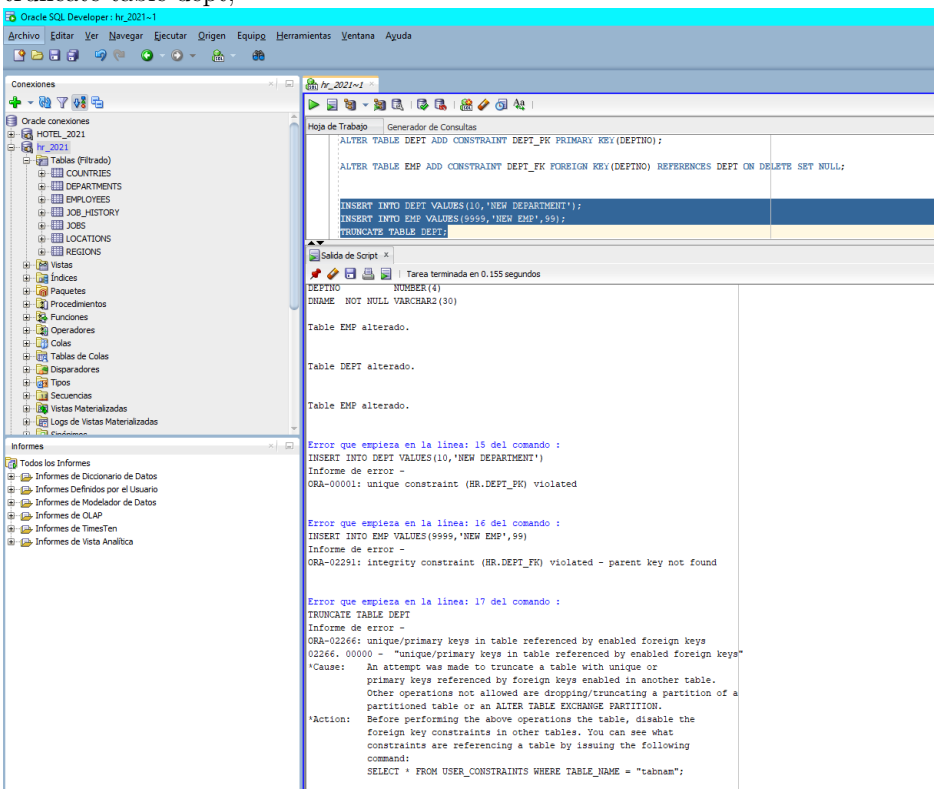
4. Use DESCRIBE to describe the structure of the new tables. Note that the not null constraint on ENAME and DNAME has been carried over from the source tables.



5. Add a primary key constraint to EMP and to DEPT and a foreign key constraint linking the tables: 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; The preceding last constraint does not specify which column of DEPT to reference; this will default to the primary key column.



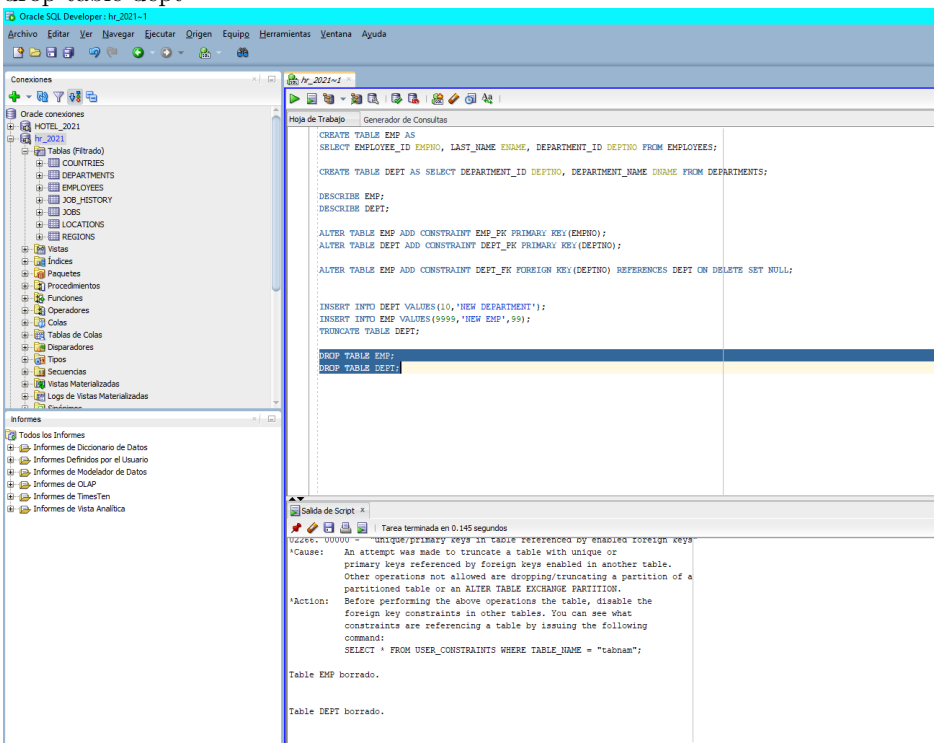
6. Demonstrate the effectiveness of the constraints by trying to insert data that will violate them: 17 insert into dept values(10,'New Department');
insert into emp values(9999,'New emp',99);
truncate table dept;



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

drop table emp;

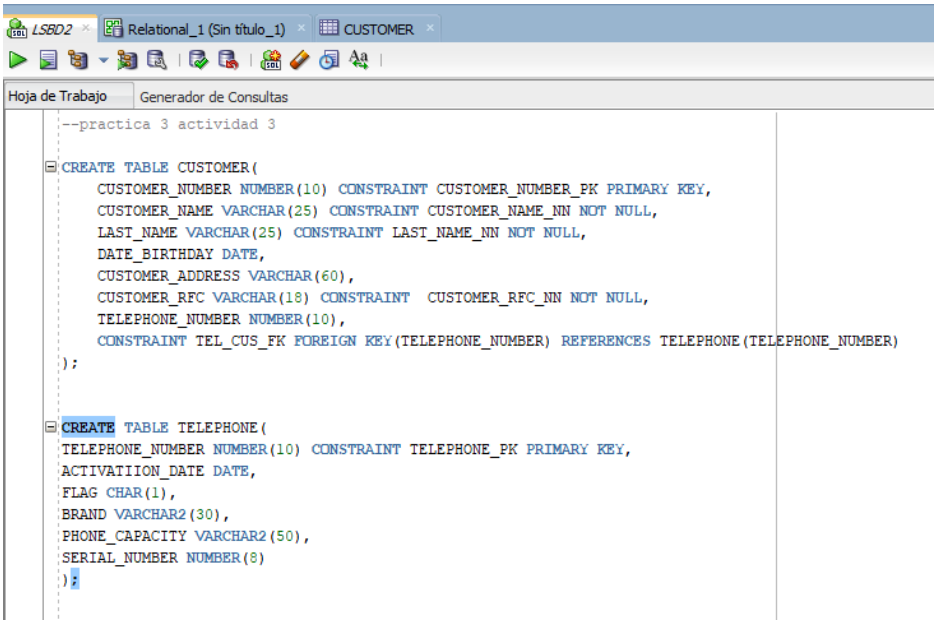
drop table dept



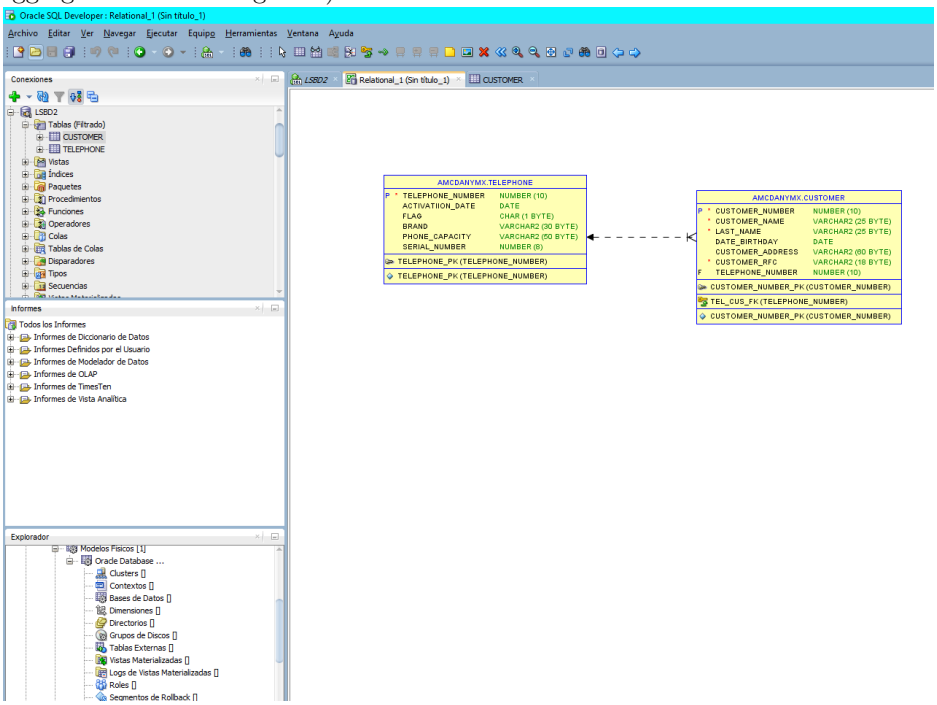
Activity 3:

the section that describes the Work developed in the following activities. Consider this simple analysis of a call record system in a local telephone company: A subscriber is identified by a customer number and also has a name, last name, date of birth, address, rfc, 1 or 2 references (friends or relatives) and possibly one or more telephones. A telephone is identified by its number, which must be a 10-digit integer beginning with 55, an activation date, and a flag ('A' or 'I') for whether it is active. Inactive telephones are not assigned to a subscriber; active telephones are. These subscribers are associated to physical telephones, and also have a brand, capacity (memory, cpu, display, camera, . . .) and serial number. Besides, all telephones are engaged with a forced plan: 6, 12, 18 or 24 months. It is necessary to store the start and final date of it (when this is hired). For every call, it is necessary to record the time it started and the time it finished.

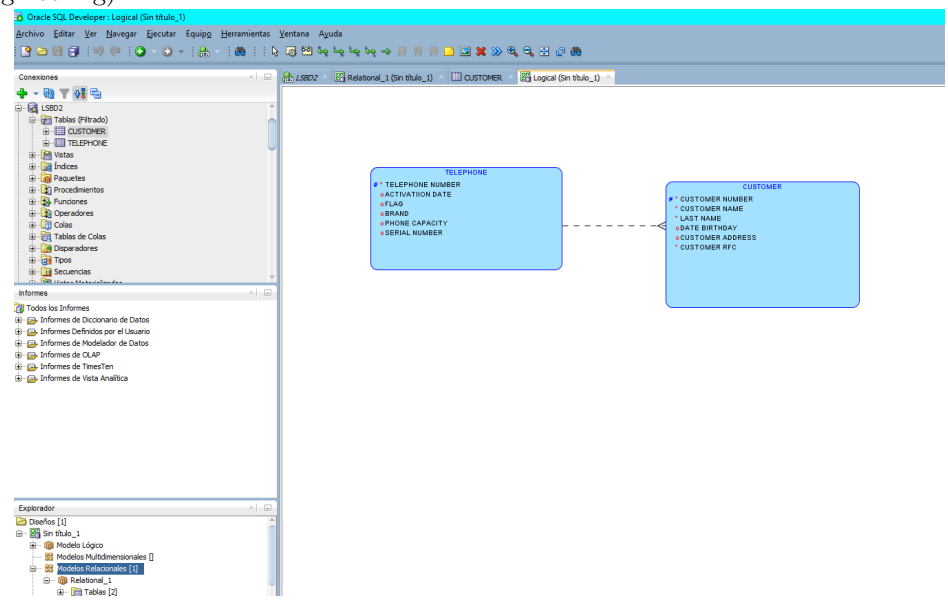
- Create the necessary tables.
- Generate the constraints and defaults that can be used to implement this system.



• Generate the corresponding relational model using SQL Data Modeler (dragging the tables using GUI).



• Generate the corresponding logical model using SQL Data Modeler (reverse engineering).



Activity 4:

Write the section that describes the Work developed in the following activities. 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.

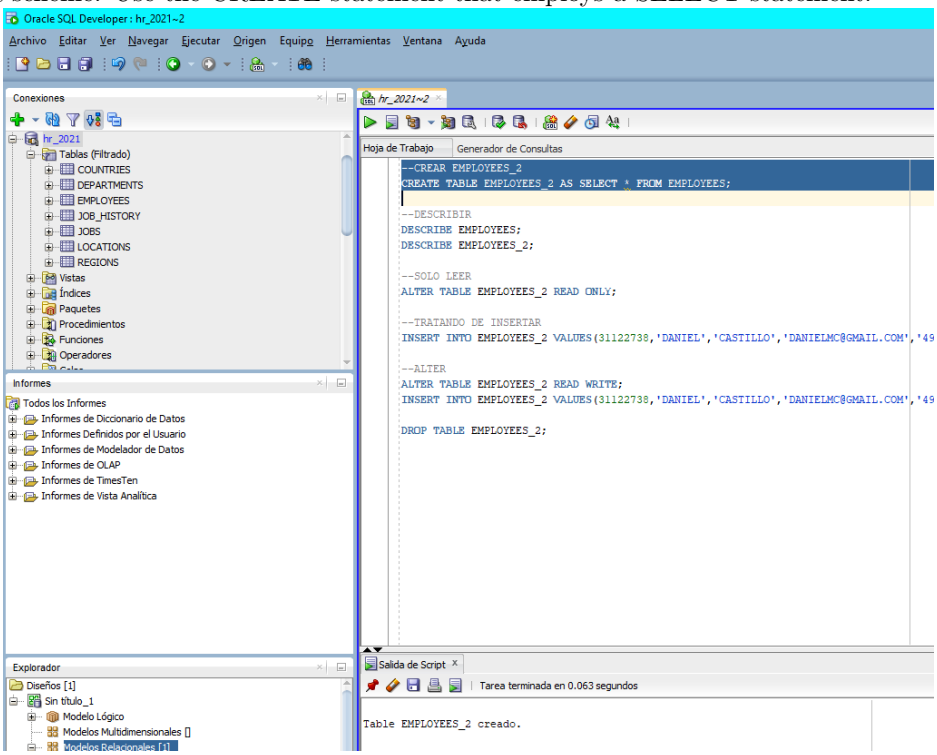
Answer: Of course, constraints help because when you move one record to another table there is a problem, as we saw in other activities, the primary key does not happen with the same restriction, we would add it with a restriction to have a Good and well structured human resources table, and thus we can also have better control if these situations happen

Activity 5:

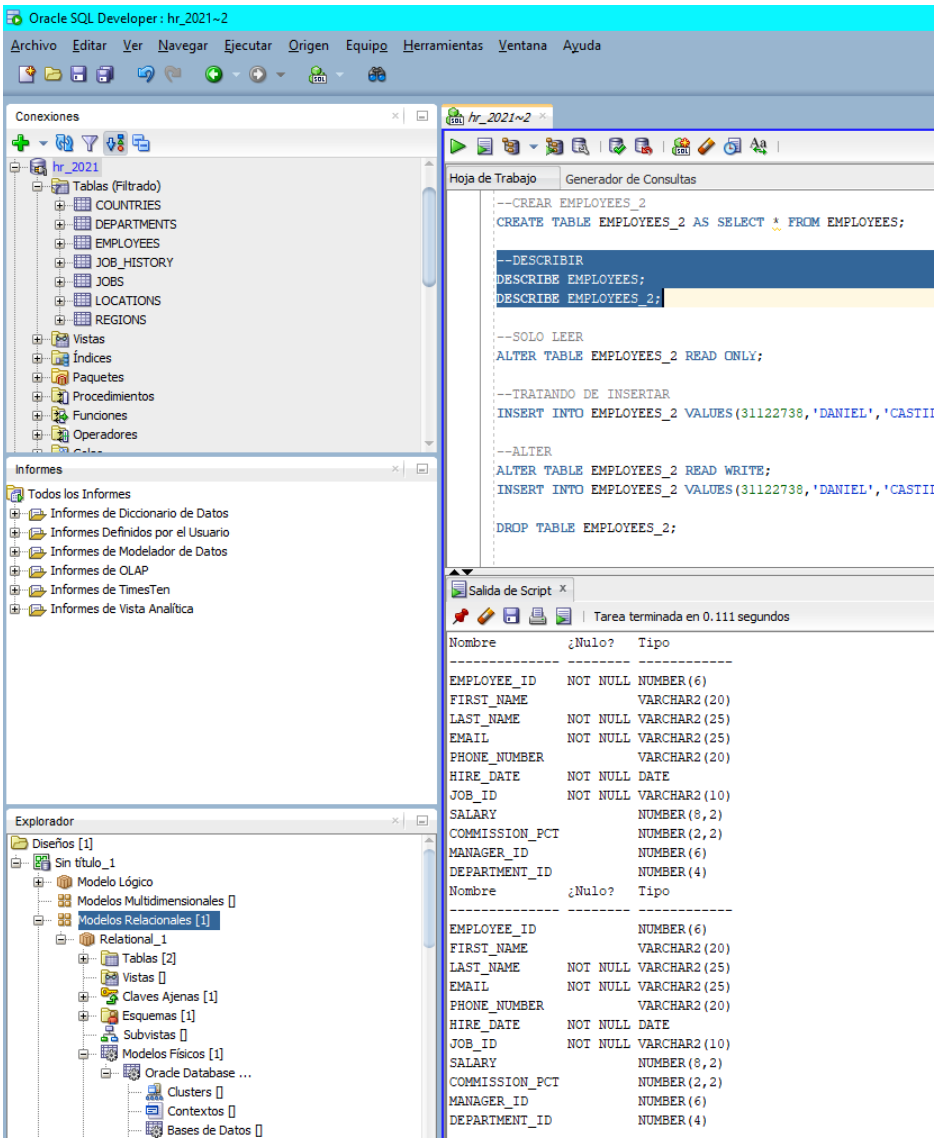
Write the section that describes the Work developed in the following activities.

Carry out the following steps (capture an image for each statement output):

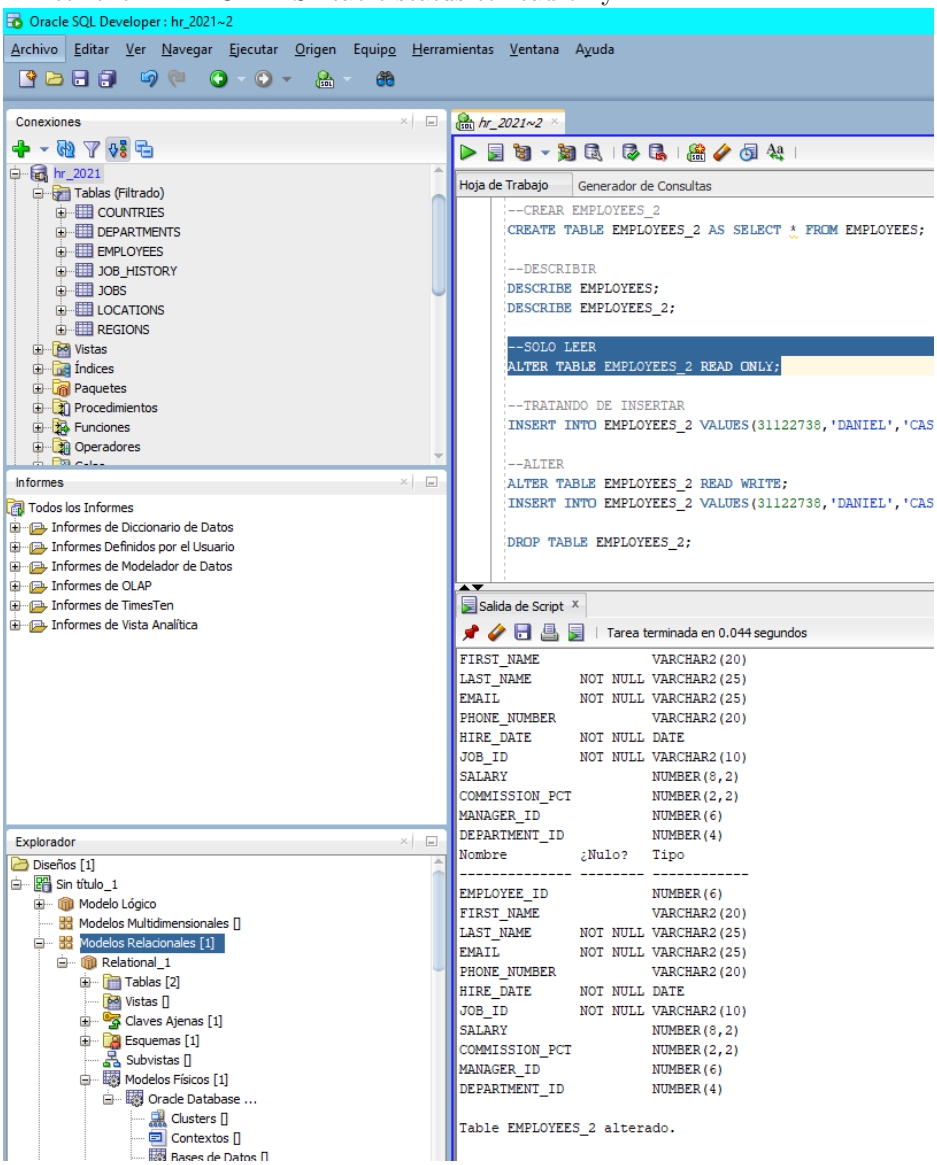
• Create the EMPLOYEES2 table based on the EMPLOYEES table from HR scheme. Use the CREATE statement that employs a SELECT statement.



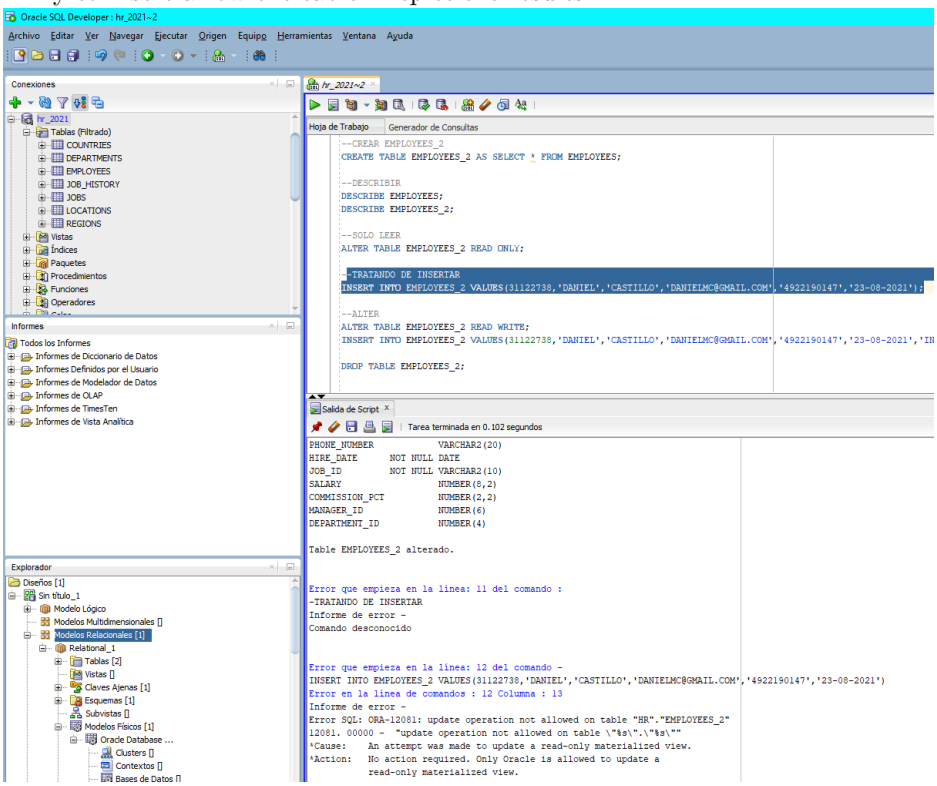
• Describe the table structure.


Oracle SQL Developer : hr_2021~2
Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda
Conexiones
hr_2021
Tablas (Filtrado)
COUNTRIES
DEPARTMENTS
EMPLOYEES
JOB_HISTORY
JOBS
LOCATIONS
REGIONS
Vistas
Índices
Paquetes
Procedimientos
Funciones
Operadores
Informes
Todos los Informes
Informes de Diccionario de Datos
Informes Definidos por el Usuario
Informes de Modelador de Datos
Informes de OLAP
Informes de TimesTen
Informes de Vista Analítica
Explorador
Diseños [1]
Sin título_1
Modelo Lógico
Modelos Multidimensionales []
Modelos Relacionales [1]
Relational_1
Tablas [2]
Vistas []
Claves Ajenas [1]
Esquemas [1]
Subvistas []
Modelos Físicos [1]
Oracle Database ...
Clusters []
Contextos []
Bases de Datos []
hr_2021~2
Hoja de Trabajo
Generador de Consultas
--CREAR EMPLOYEES_2
CREATE TABLE EMPLOYEES_2 AS SELECT * FROM EMPLOYEES;
--DESCRIBIR
DESCRIBE EMPLOYEES;
DESCRIBE EMPLOYEES_2;
--SOLO LEER
ALTER TABLE EMPLOYEES_2 READ ONLY;
--TRATANDO DE INSERTAR
INSERT INTO EMPLOYEES_2 VALUES(31122738,'DANIEL','CASTII
--ALTER
ALTER TABLE EMPLOYEES_2 READ WRITE;
INSERT INTO EMPLOYEES_2 VALUES(31122738,'DANIEL','CASTII
DROP TABLE EMPLOYEES_2;
Salida de Script
Tarea terminada en 0.111 segundos
Nombre ¿Nulo? Tipo
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(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID NUMBER(6)
DEPARTMENT_ID NUMBER(4)
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(8,2)
COMMISSION_PCT NUMBER(2,2)
MANAGER_ID NUMBER(6)
DEPARTMENT_ID NUMBER(4)

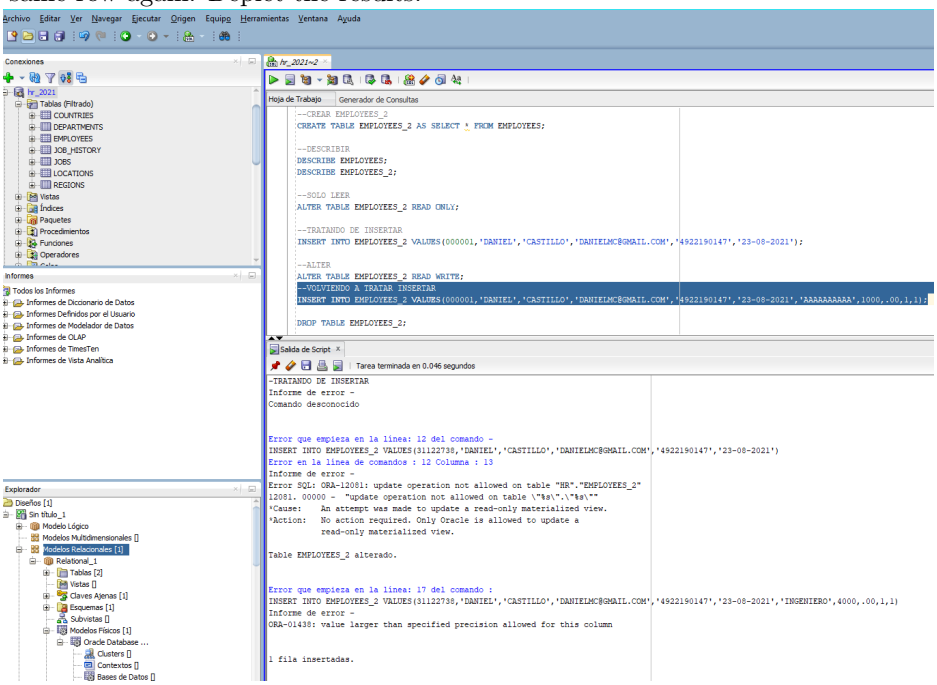
- Alter the EMPLOYEES2 table status to read-only.



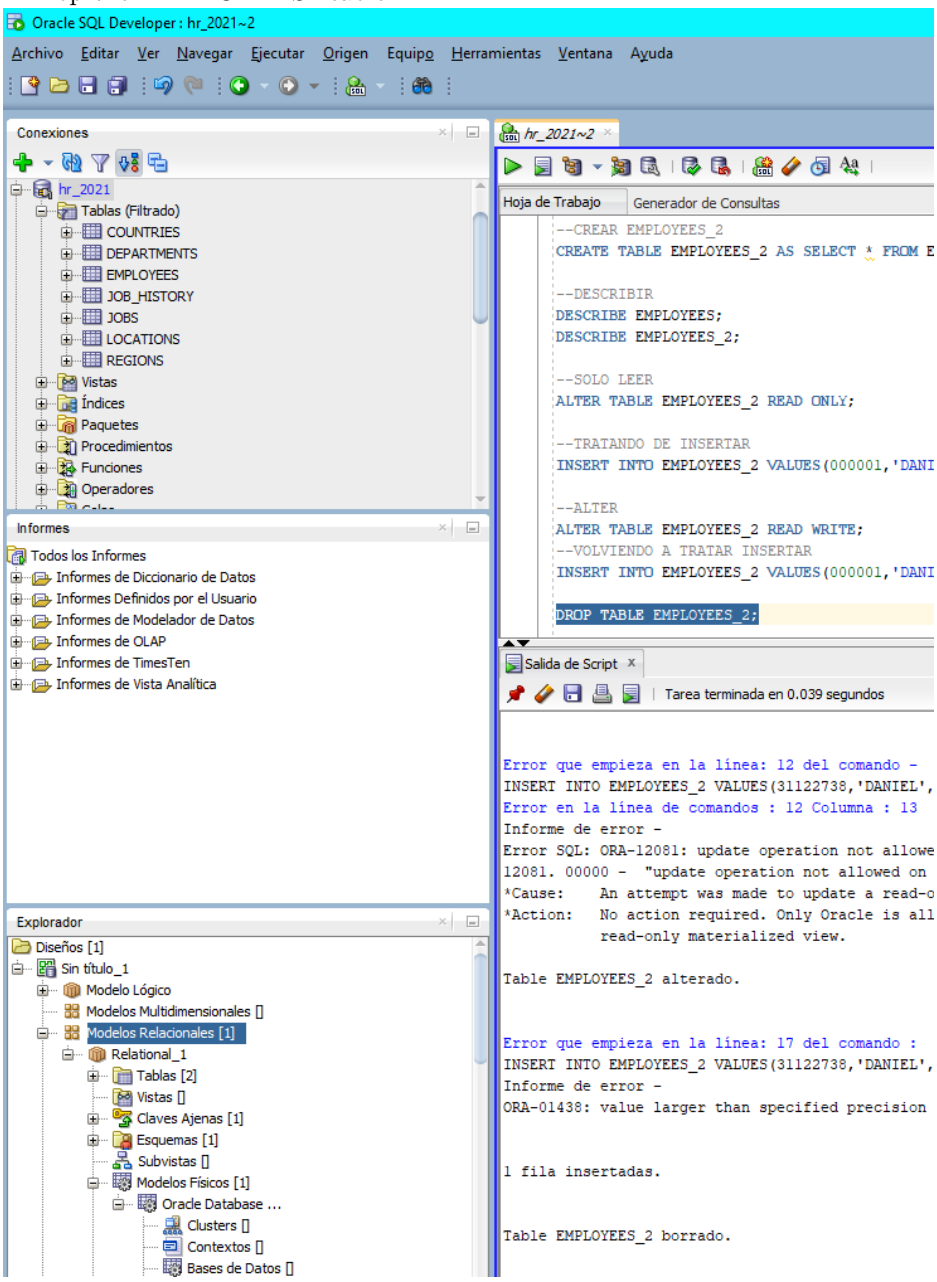
• Try to insert a row the table. Depict the results.



• Revert the EMPLOYEES2 table to the write status. Now, try to insert the same row again. Depict the results.

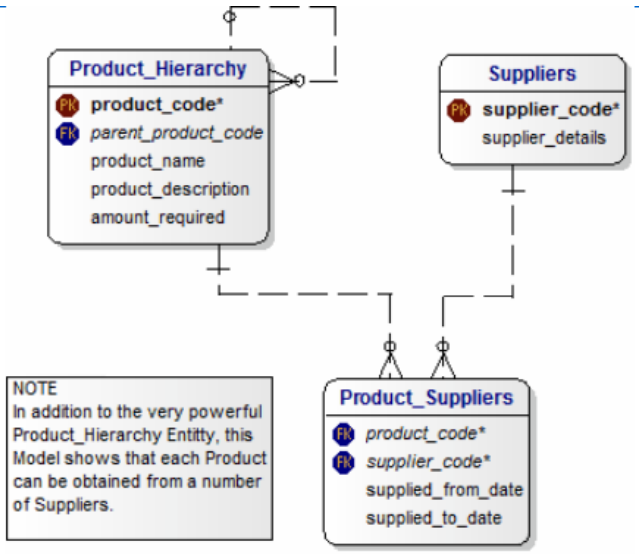


• Drop the EMPLOYEES2 table.

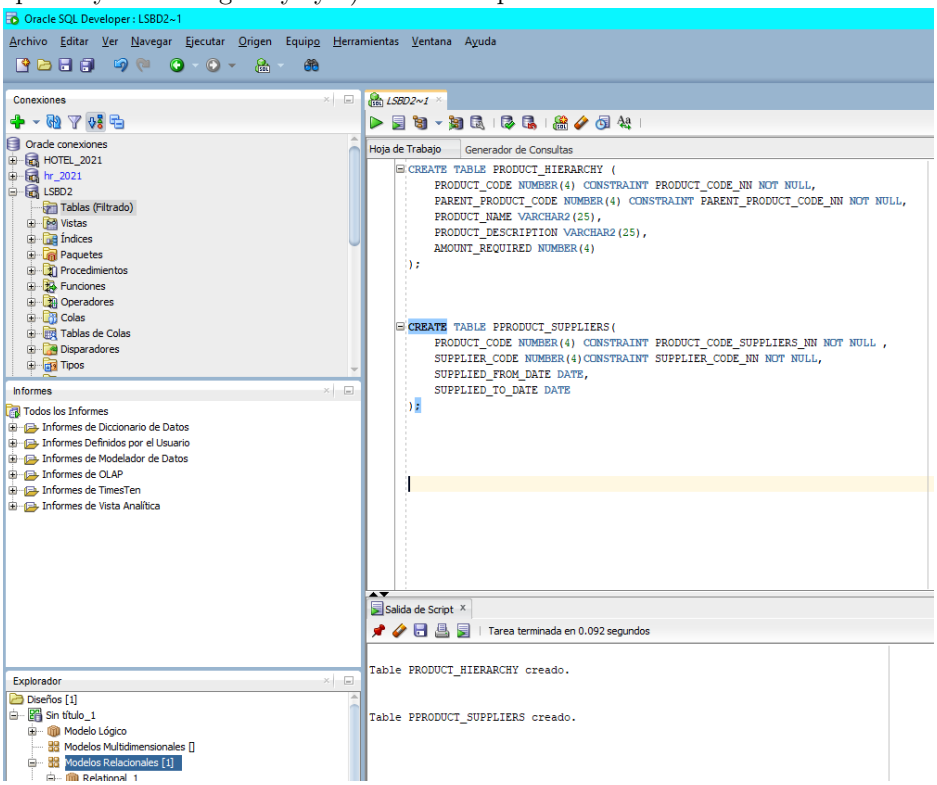


Activity 6:

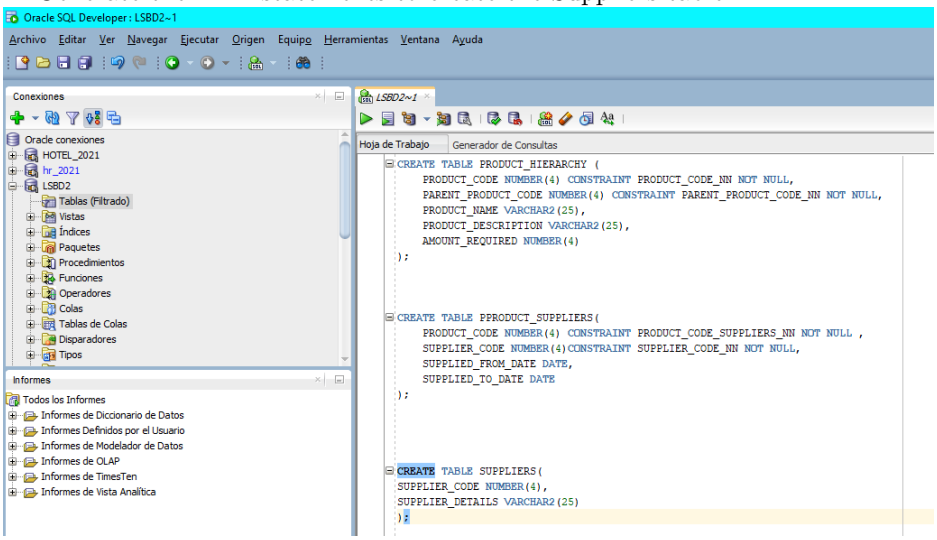
Write the section that describes the Work developed in the following activities. Taking into account the following diagram.



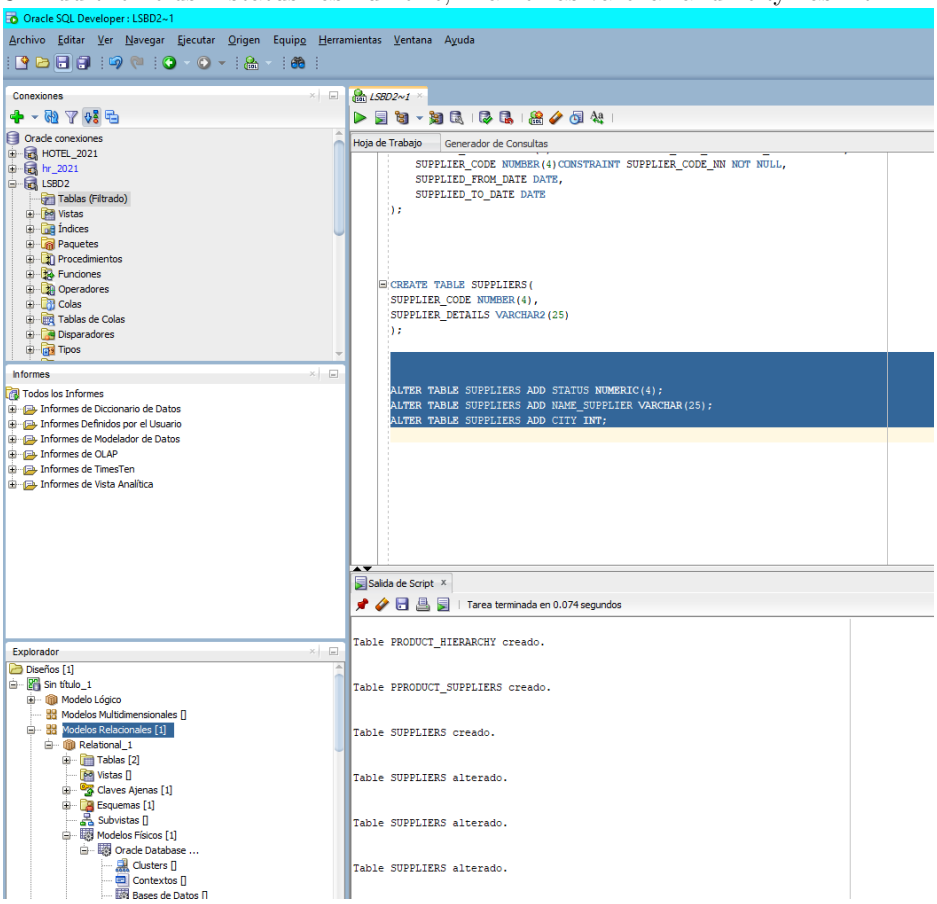
1. Generate the DDL statements to create only the isolated tables of ProductHierarchy and ProductSuppliers (add constraints here in CREATE, don‘t add primary and foreign keys yet). Use descriptive names in constraints.



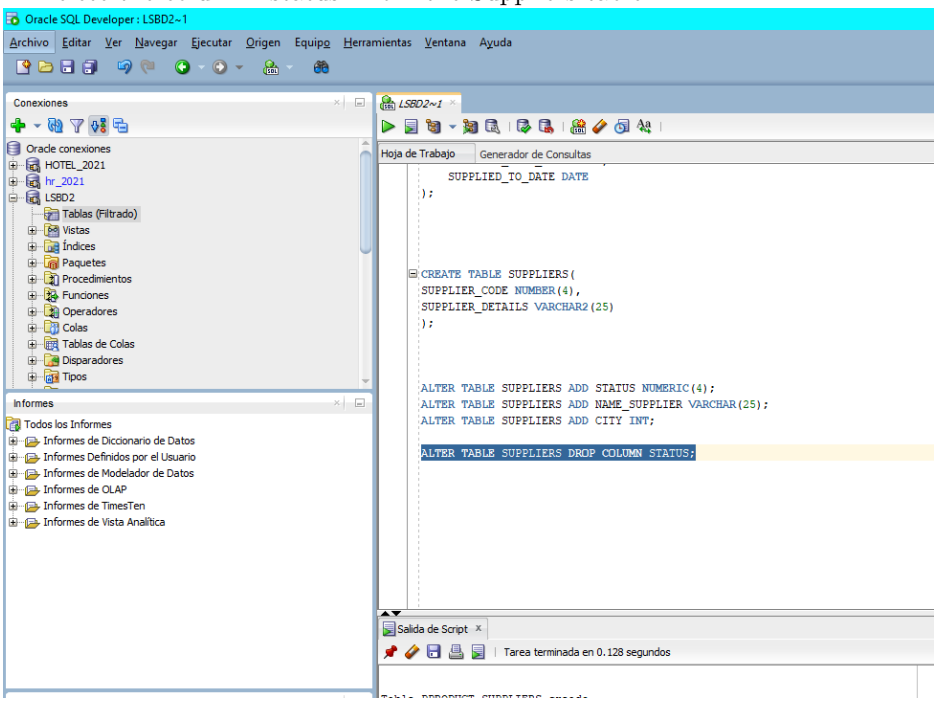
2. Generate the DDL statements to create the Suppliers table.



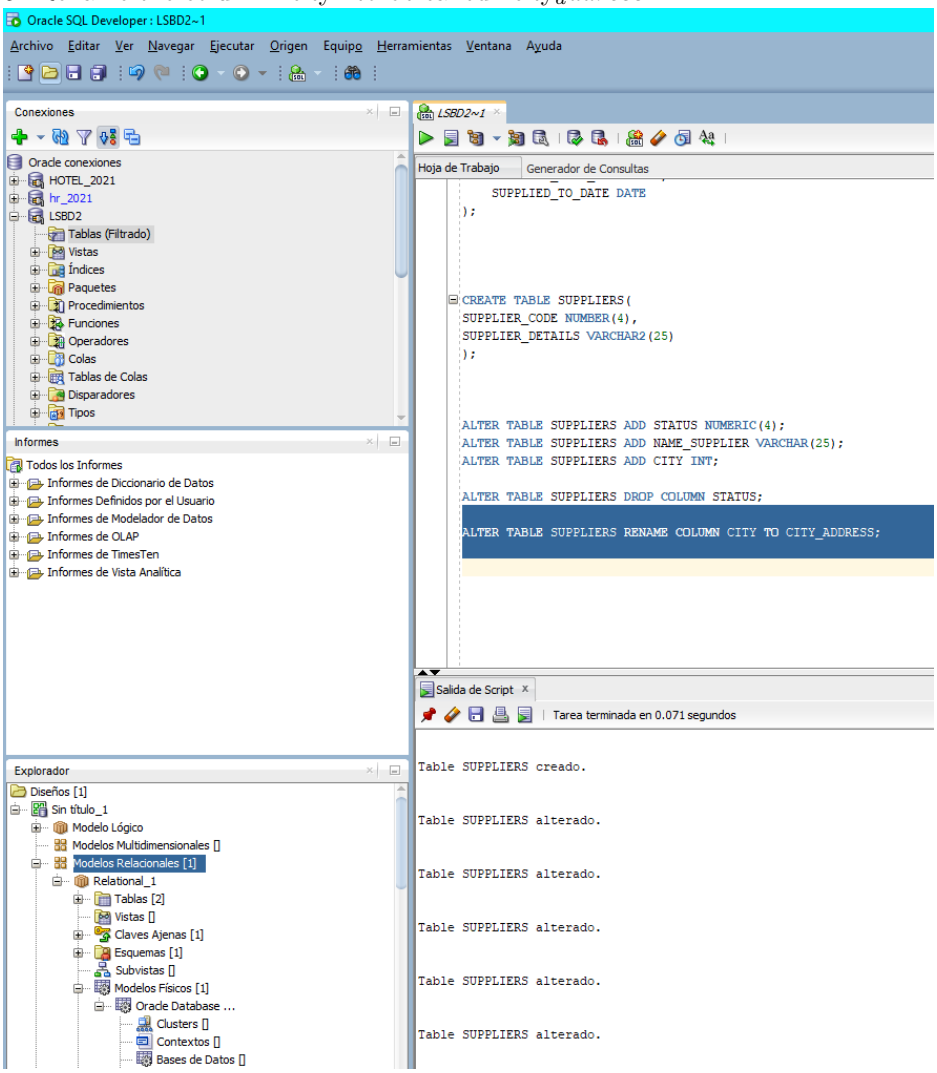
3. Add the fields: “status” as numeric, ”name” as varchar and “city” as int.



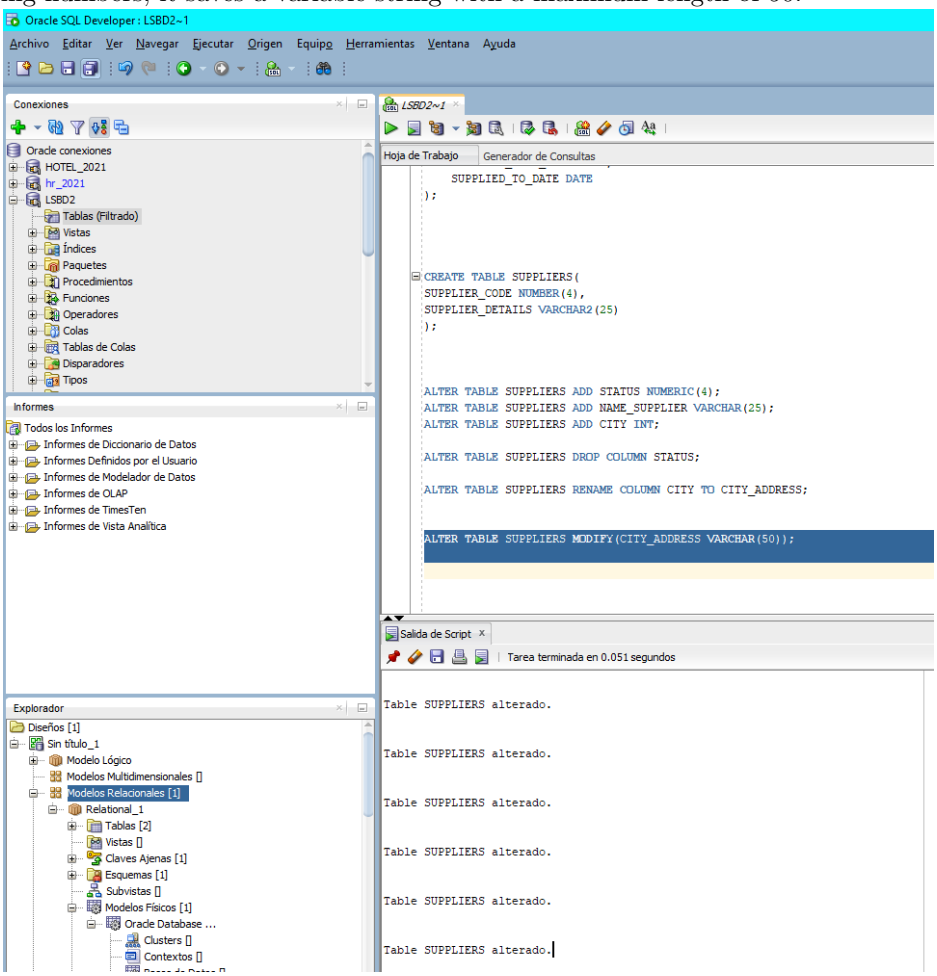
4. Delete the column “status” from the Suppliers table.



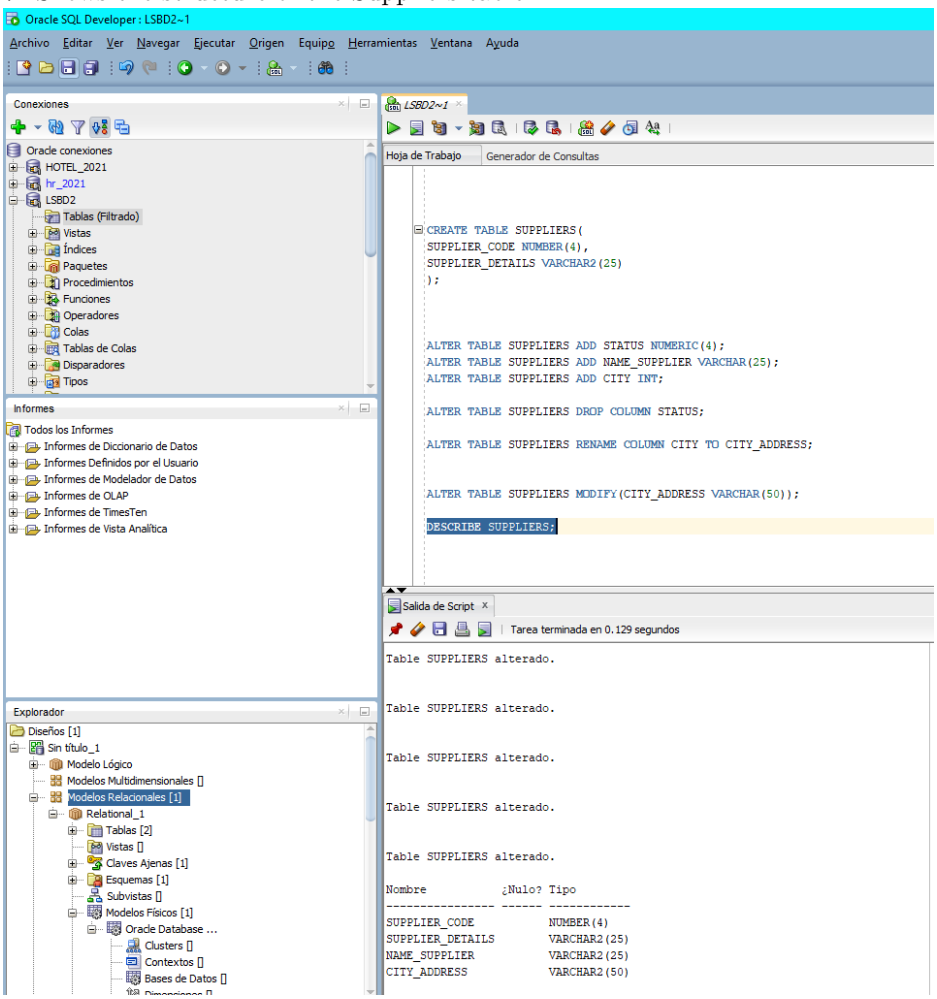
5. Rename the column "city" to be called "city$_address$".



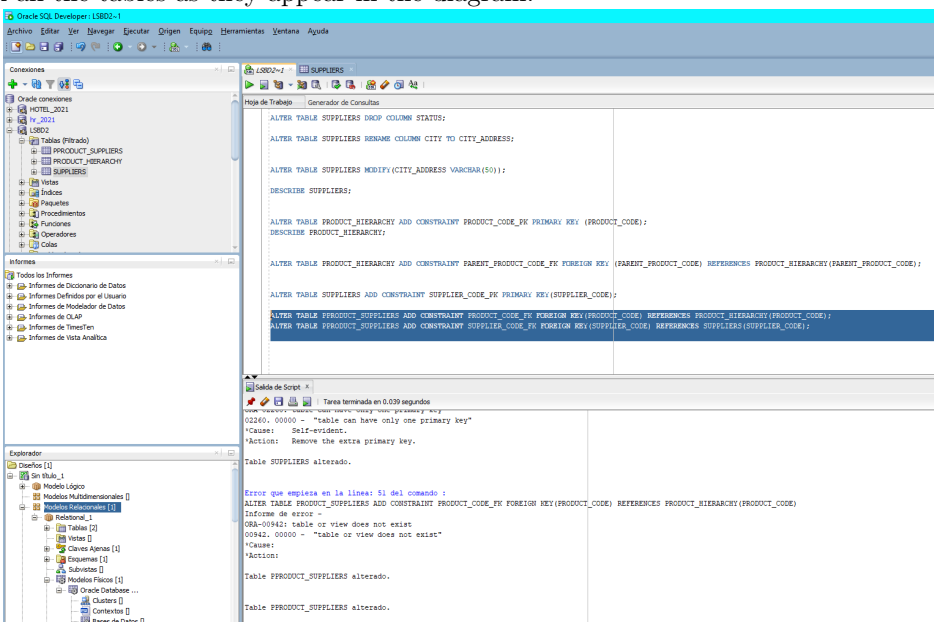
6. Modify the data type of the “cityaddress” column, so that instead of saving numbers, it saves a variable string with a maximum length of 50.



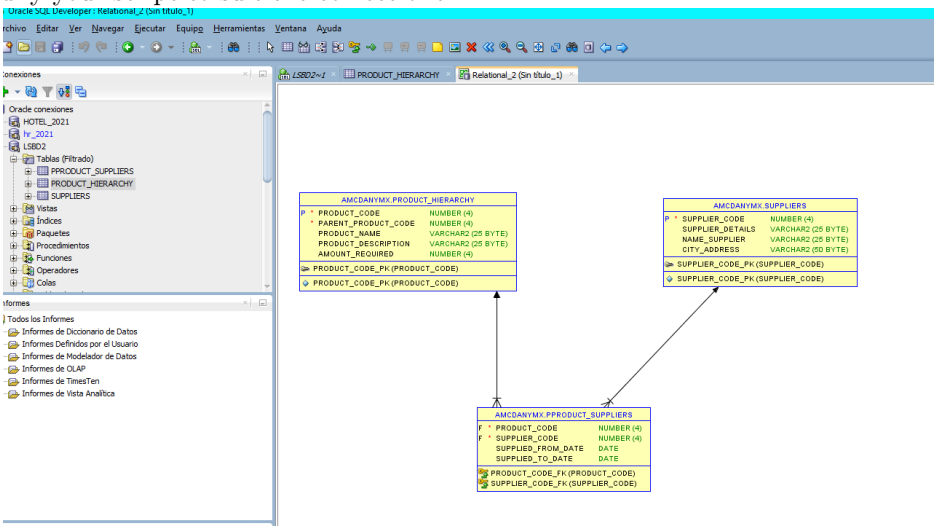
7. Shows the structure of the Suppliers table.



8. Generate, with the ALTER statements, all the necessary instructions to link all the tables as they appear in the diagram.

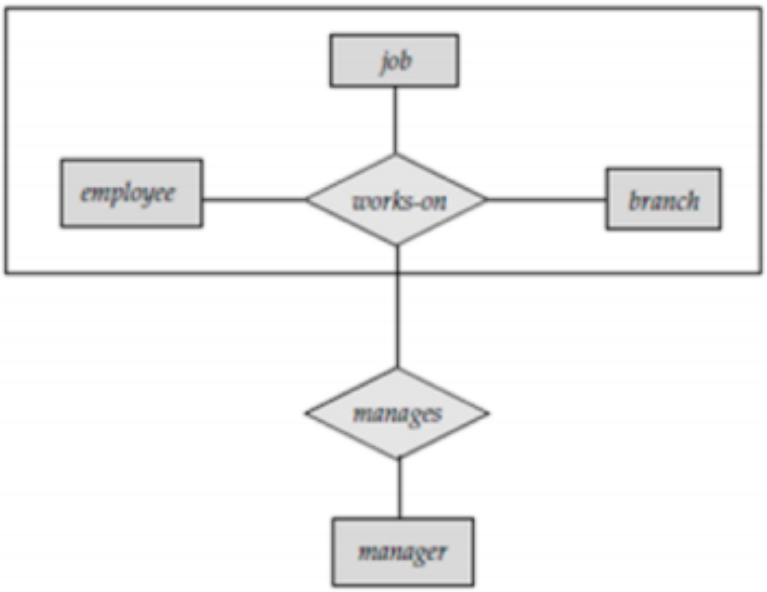


9. Obtain the relational diagram and compare the results. If necessary, modify your script to suit the correct one.

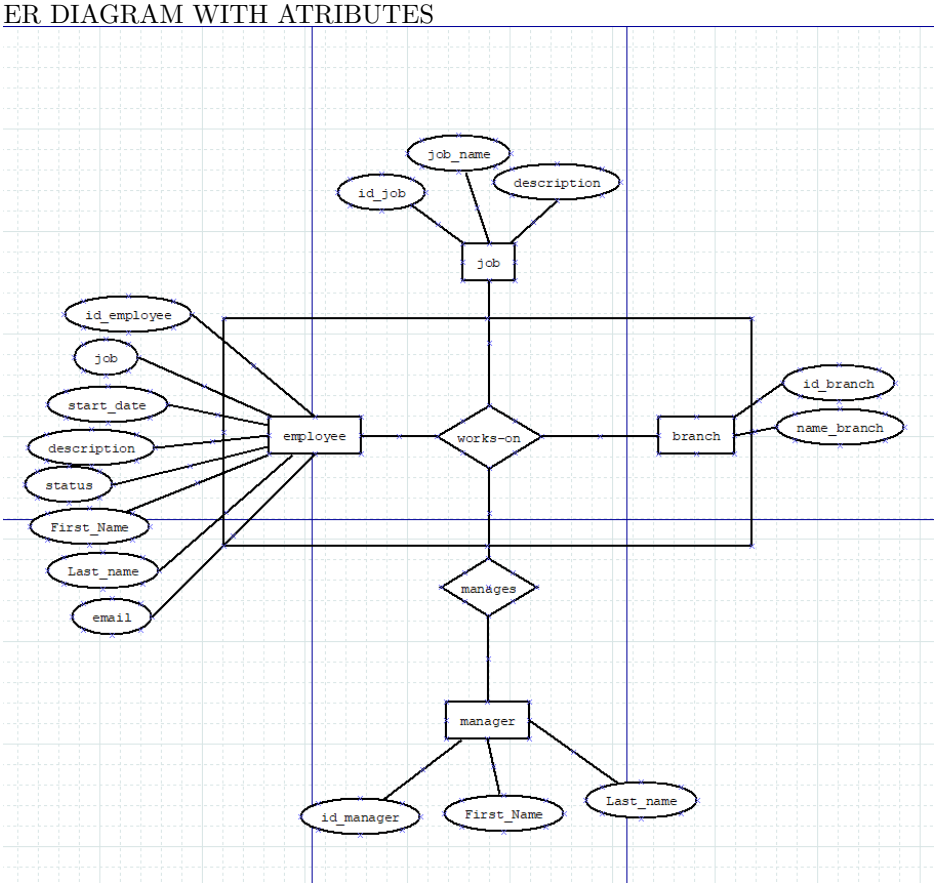


Activity 7:

Write the section that describes the Work developed in the following activities. Complete the following ER diagram with the corresponding attributes. The main idea is: an employee (with their personal data) works on a specific job, which initiated in a particular date/hour with a description and status. This job is carried out in a branch, in which the task is located. Thus, a supervisor manages the entire task with a specific date of assignment



ER DIAGRAM WITH ATRIBUTES

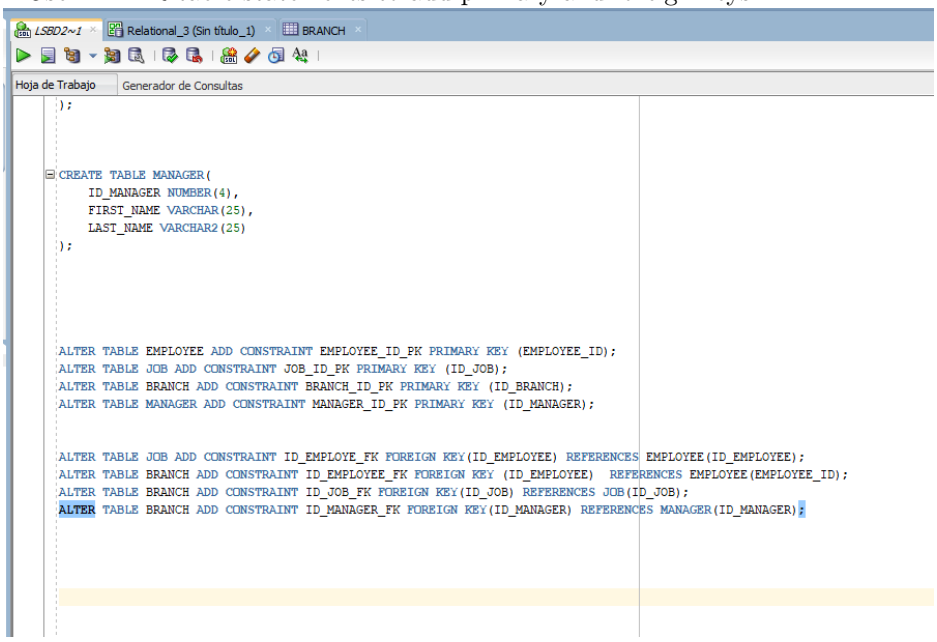


Using the following E-R diagram. Generate the corresponding DDL statements. Don‘t forget to consider:

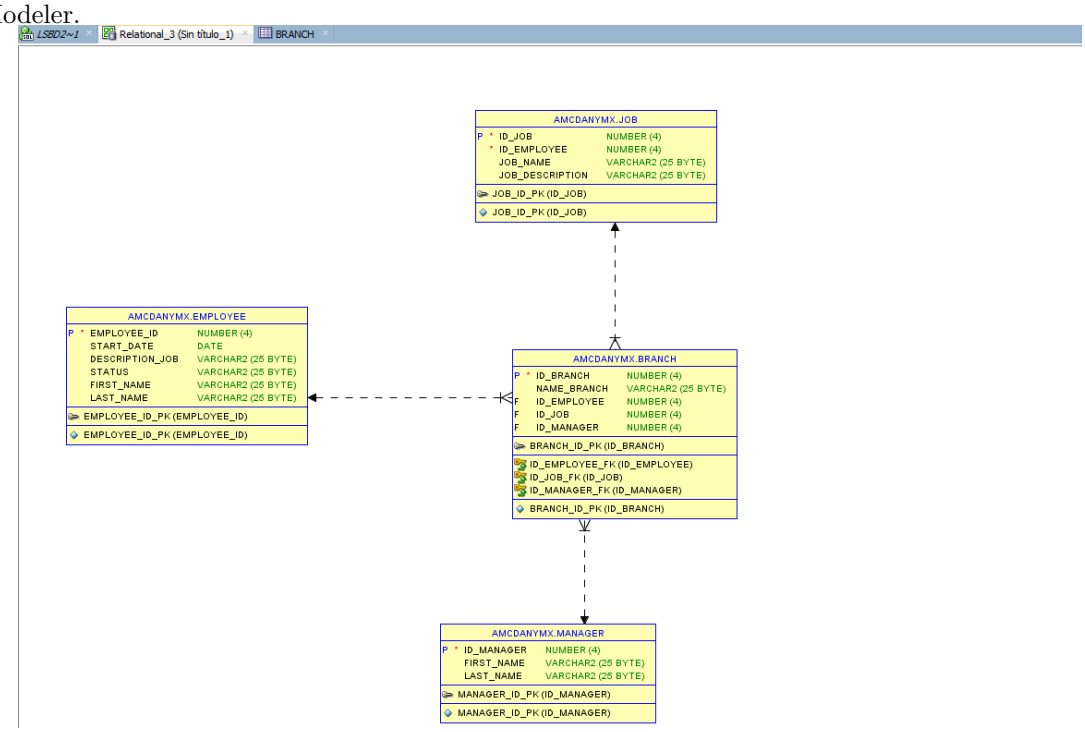
• CREATE table statements are the first to be applied. In this case, the basic constraints must be performed here: not null, default, ...

```
--ACTIVIDAD 7

CREATE TABLE EMPLOYEE(
    EMPLOYEE_ID NUMBER(4) CONSTRAINT EMPLOYEE_ID_NN NOT NULL,
    START_DATE DATE,
    DESCRIPTION_JOB VARCHAR2(25),
    STATUS VARCHAR(25),
    FIRST_NAME VARCHAR(25),
    LAST_NAME  VARCHAR(25)
);

CREATE TABLE JOB(
    ID_JOB NUMBER(4) CONSTRAINT ID_JOB__NN NOT NULL,
    ID_EMPLOYEE NUMBER(4) CONSTRAINT ID_EMPLOYEE_NN NOT NULL,
    JOB_NAME VARCHAR(25),
    JOB_DESCRIPTION VARCHAR2(25)
);


CREATE TABLE BRANCH(
ID_BRANCH NUMBER(4),
NAME_BRANCH VARCHAR2(25),
ID_EMPLOYEE NUMBER(4),
ID_JOB NUMBER(4),
ID_MANAGER NUMBER(4)
);



CREATE TABLE MANAGER(
    ID_MANAGER NUMBER(4),
    FIRST_NAME VARCHAR(25),
    LAST_NAME VARCHAR2(25)
);
```

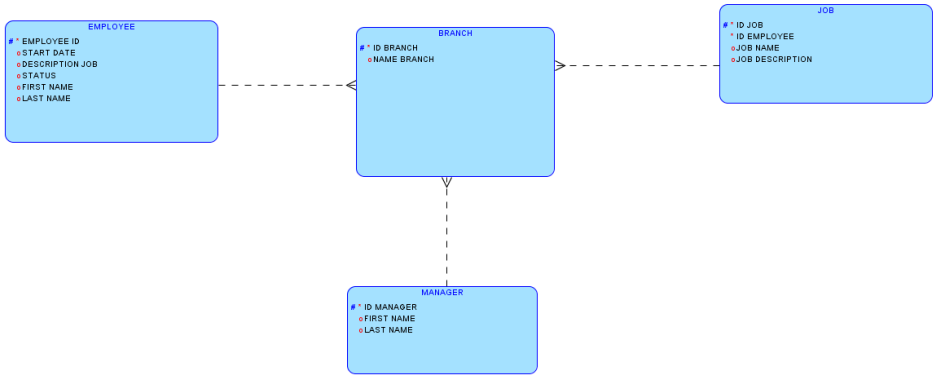
• Use ALTER table statements to add primary and foreign keys



• Generate automatically the corresponding relational model using Data Modeler.



- Generate the logical diagram in SQL Data Modeler.

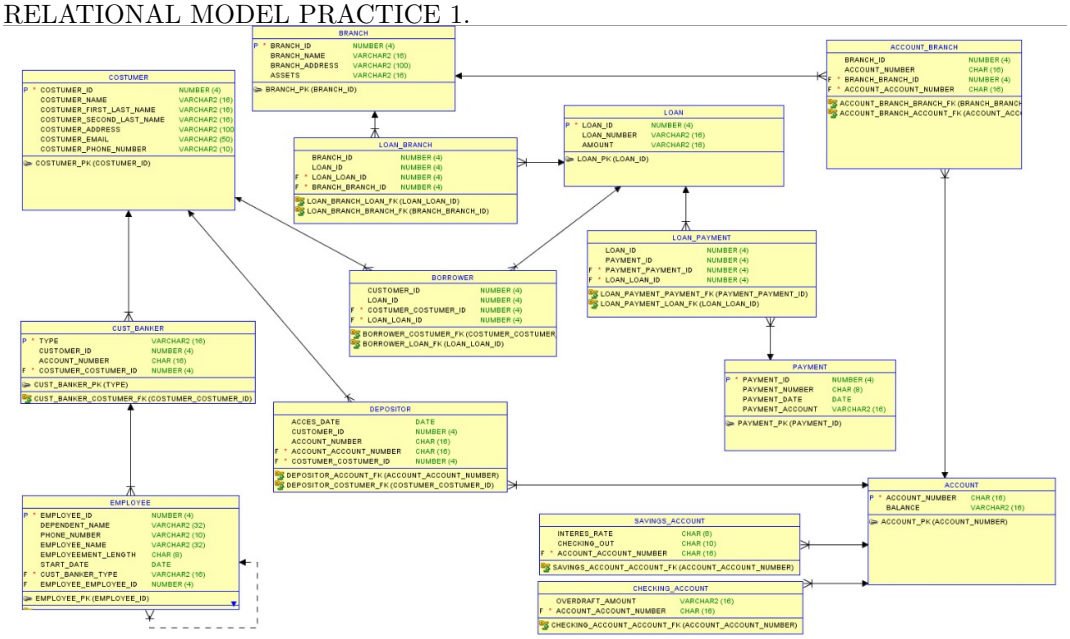


Activity 8:

Write the section that describes the Work developed in the following activities

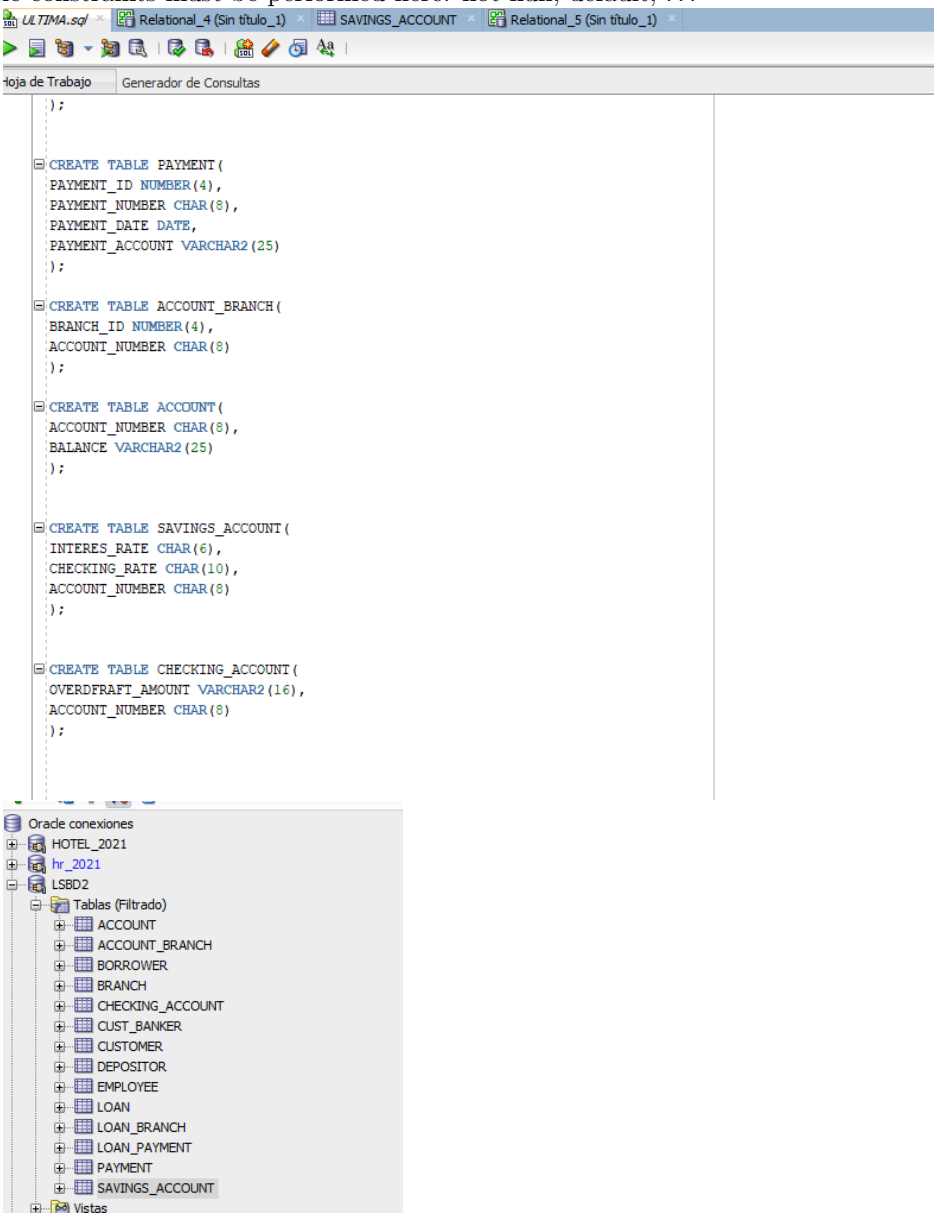
Taking into account your final E-R diagram of activity 2 of practice 1 (Bank scenario).

RELATIONAL MODEL PRACTICE 1.



Generate the corresponding DDL statements. Don‘t forget to consider:

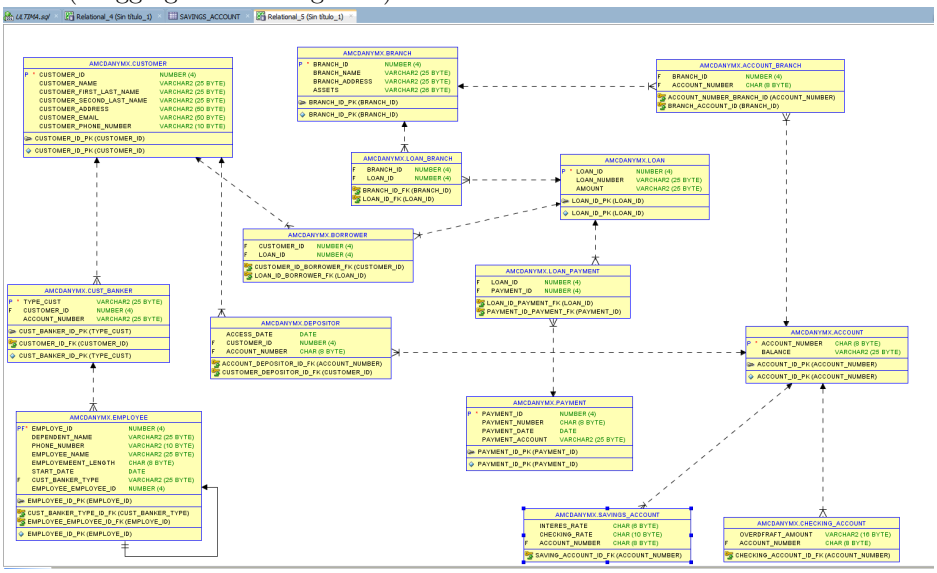
• CREATE table statements are the first to be applied. In this case, the basic constraints must be performed here: not null, default, . . .



• Use ALTER table statements to add primary and foreign keys.

```
ALTER TABLE CUSTOMER ADD CONSTRAINT CUSTOMER_ID_PK PRIMARY KEY(CUSTOMER_ID);
ALTER TABLE BRANCH ADD CONSTRAINT BRANCH_ID_PK PRIMARY KEY(BRANCH_ID);
ALTER TABLE LOAN ADD CONSTRAINT LOAN_ID_PK PRIMARY KEY(LOAN_ID);
ALTER TABLE CUST_BANKER ADD CONSTRAINT CUST_BANKER_ID_PK PRIMARY KEY(TYPE_CUST);
ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE_ID_PK PRIMARY KEY(EMPLOYE_ID);
ALTER TABLE PAYMENT ADD CONSTRAINT PAYMENT_ID_PK PRIMARY KEY(PAYMENT_ID);
ALTER TABLE ACCOUNT ADD CONSTRAINT ACCOUNT_ID_PK PRIMARY KEY(ACCOUNT_NUMBER);


ALTER TABLE CUST_BANKER ADD CONSTRAINT CUSTOMER_ID_FK FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID);
ALTER TABLE EMPLOYEE ADD CONSTRAINT CUST_BANKER_TYPE_ID_FK FOREIGN KEY(CUST_BANKER_TYPE) REFERENCES CUST_BANKER(TYPE_CUST);
ALTER TABLE LOAN_BRANCH ADD CONSTRAINT BRANCH_ID_FK FOREIGN KEY(BRANCH_ID) REFERENCES BRANCH(BRANCH_ID);
ALTER TABLE LOAN_BRANCH ADD CONSTRAINT LOAN_ID_FK FOREIGN KEY(LOAN_ID) REFERENCES LOAN(LOAN_ID);
ALTER TABLE BORROWER ADD CONSTRAINT CUSTOMER_ID_BORROWER_FK FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID);
ALTER TABLE BORROWER ADD CONSTRAINT LOAN_ID_BORROWER_FK FOREIGN KEY(LOAN_ID) REFERENCES LOAN(LOAN_ID);
ALTER TABLE LOAN_PAYMENT ADD CONSTRAINT LOAN_ID_PAYMENT_FK FOREIGN KEY(LOAN_ID) REFERENCES LOAN(LOAN_ID);
ALTER TABLE LOAN_PAYMENT ADD CONSTRAINT PAYMENT_ID_PAYMENT_FK FOREIGN KEY(PAYMENT_ID) REFERENCES PAYMENT(PAYMENT_ID);
ALTER TABLE DEPOSITOR ADD CONSTRAINT CUSTOMER_DEPOSITOR_ID_FK FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID);
ALTER TABLE DEPOSITOR ADD CONSTRAINT ACCOUNT_DEPOSITOR_ID_FK FOREIGN KEY(ACCOUNT_NUMBER) REFERENCES ACCOUNT(ACCOUNT_NUMBER);
ALTER TABLE SAVINGS_ACCOUNT ADD CONSTRAINT SAVING_ACCOUNT_ID_FK FOREIGN KEY(ACCOUNT_NUMBER) REFERENCES ACCOUNT(ACCOUNT_NUMBER);
ALTER TABLE CHECKING_ACCOUNT ADD CONSTRAINT CHECKING_ACCOUNT_ID_FK FOREIGN KEY(ACCOUNT_NUMBER) REFERENCES ACCOUNT(ACCOUNT_NUMBER);

ALTER TABLE EMPLOYEE ADD CONSTRAINT EMPLOYEE_EMPLOYEE_ID_FK FOREIGN KEY(EMPLOYE_ID) REFERENCES EMPLOYEE(EMPLOYE_ID);

ALTER TABLE ACCOUNT_BRANCH ADD CONSTRAINT BRANCH_ACCOUNT_ID FOREIGN KEY(BRANCH_ID) REFERENCES BRANCH(BRANCH_ID);
ALTER TABLE ACCOUNT_BRANCH ADD CONSTRAINT ACCOUNT_NUMBER_BRANCH_ID FOREIGN KEY(ACCOUNT_NUMBER) REFERENCES ACCOUNT(ACCOUNT_NUMBER);
```

• Generate automatically the corresponding relational model using Data Modeler (dragging tables using GUI).



• Compare the results with practice 1

we can see that the relational models of the bank scenario, of practice 1 and this are practically the same

3 PRE-EVALUATION

Practices pre-Assessment for Database Systems Laboratory II Pre-Assessment
PRACTICE 2 carried out by student

1 COMPLIES WITH THE REQUESTED FUNCTIONALITY
YES

4 HAS THE CORRECT INDENTATION
YES

6 HAS AN EASY WAY TO ACCESS THE PROVIDED FILES
YES

7 HAS A REPORT WITH IDC FORMAT
YES

8 REPORT INFORMATION IS FREE OF SPELLING ERRORS
YES

9 DELIVERED IN TIME AND FORM
YES

10 IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED)
YES,100 percent

4 Conclusion

The entity relationship diagrams are important to have a correct development of the database logic, and can be seen better in the relational models. this practice helped me to learn something else and review the topic again.

We know that entity-relationship models are very important for the realization of databases and we have to pay close attention to them.

The issue of restrictions is very important just like the others in the database, we can manipulate them in different ways to create them either on the same line in CREATE or after having created the fields of the tables

speaking now of the ALTERs, I think it is a very useful command within the sql language, as it allows you to quickly modify things in your tables and create constraints outside of them

5 Extra-notes

A tablet with images and script of the practice is attached