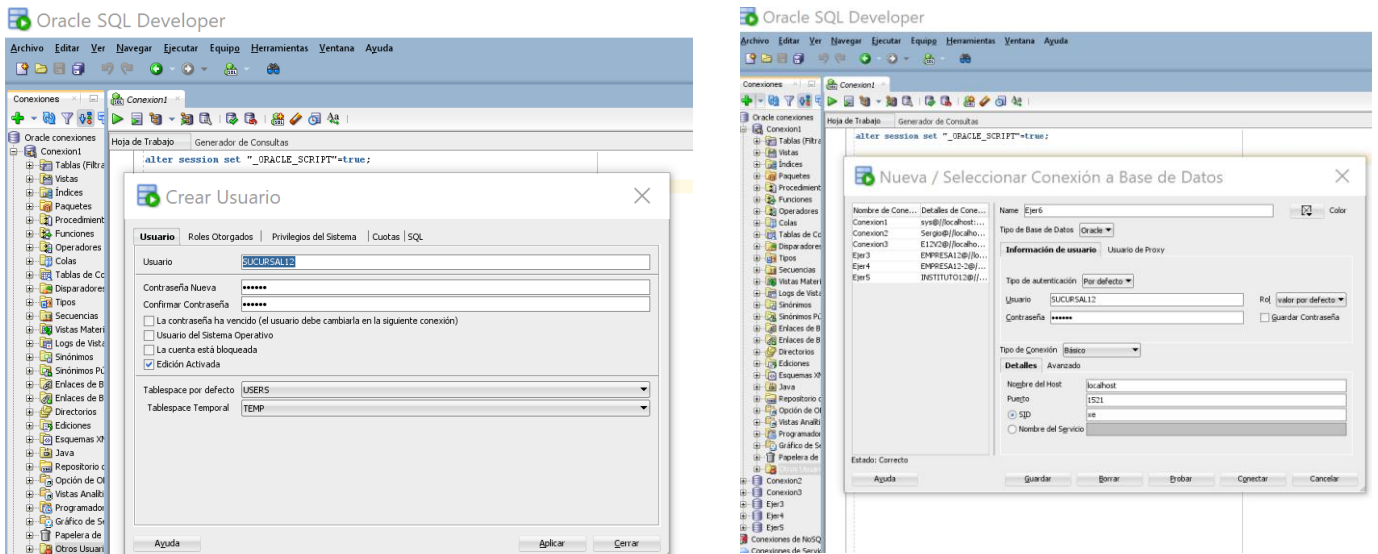
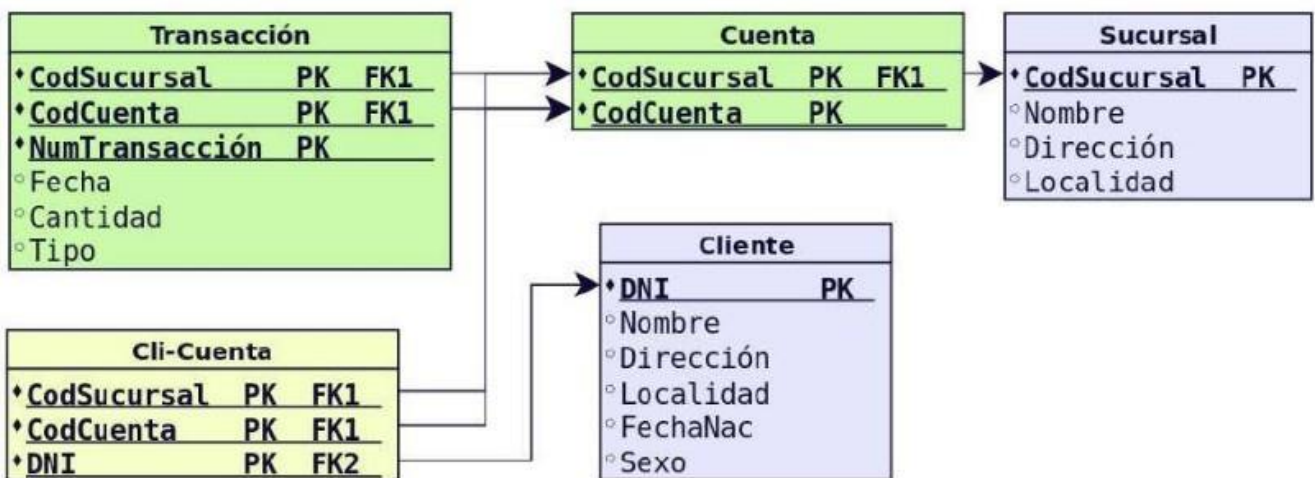


Operaciones DDL sobre tablas



1. Realiza el diseño físico para el siguiente modelo relacional. Asigna el tipo de datos que consideres más adecuado. Realiza el diseño sin restricciones.



Oracle SQL Developer

Archivo Editar Ver Navegar Ejecutar Equipo Herramientas Ventana Ayuda

Conexiones

Oracle conexiones

- Conexion1
- Conexion2
- Conexion3
- Ejer3
- Ejer4
- Ejer5
- Ejer6

Tablas (Filtrado)

- Vistas
- Índices
- Paquetes
- Procedimientos
- Funciones
- Operadores
- Colas
- Tablas de Colas
- Disparadores
- Tipos
- Secuencias
- Vistas Materializadas
- Logs de Vistas Materializadas
- Sinónimos
- Sinónimos Públicos
- Enlaces de Base de Datos
- Enlaces de Base de Datos
- Directorios
- Ediciones
- Esquemas XML
- Java
- Repositorio de Base de Datos
- Opción de OLAP
- Vistas Analíticas
- Programador
- Gráfico de Semántica R
- Papelera de Reciclaje
- Otros Usuarios

Conexiones de NoSQL de Oracle

Conexiones de Servicio Esquemas

Hoja de Trabajo Generador de Consultas

```
1 CREATE TABLE Sucursal (  
2   cod_sucursal number(10),  
3   nom_sucursal varchar2(20),  
4   direcc_sucursal varchar2(30),  
5   local_sucursal varchar2(20),  
6   CONSTRAINT PK_SUCURSAL PRIMARY KEY (cod_sucursal)  
7 );  
8  
9 CREATE TABLE Cliente (  
10  dni_cli varchar2(9),  
11  nom_cli varchar2(20),  
12  direcc_cli varchar2(30),  
13  local_cli varchar2(20),  
14  fecha_nac date,  
15  sexo varchar2(10),  
16  CONSTRAINT PK_CLIENTE PRIMARY KEY (dni_cli)  
17 );  
18  
19 CREATE TABLE Cuenta (  
20  cod_sucursal number(10),  
21  cod_cuenta number(10),  
22  CONSTRAINT PK_CUENTA PRIMARY KEY (cod_sucursal, cod_cuenta),  
23  CONSTRAINT FK_CUENTA FOREIGN KEY (cod_sucursal) REFERENCES Sucursal (cod_sucursal)  
24 );  
25  
26 CREATE TABLE Transaccion (  
27  cod_sucursal number(10),  
28  cod_cuenta number(10),  
29  num_transaccion number(10),  
30  fecha_transaccion date,  
31  cantidad_transaccion number(9,2),  
32  tipo_transaccion varchar2(10),  
33  CONSTRAINT PK_TRANSACCION PRIMARY KEY (cod_sucursal, cod_cuenta, num_transaccion),  
34  CONSTRAINT FK_TRANSACCION FOREIGN KEY (cod_sucursal, cod_cuenta) REFERENCES Cuenta (cod_sucursal, cod_cuenta)  
35 );  
36  
37 CREATE TABLE CliCuenta (  
38  cod_sucursal number(10),  
39  cod_cuenta number(10),  
40  dni_cli varchar2(9),  
41  CONSTRAINT PK_CLICUENTA PRIMARY KEY (cod_sucursal, cod_cuenta, dni_cli),  
42  CONSTRAINT FK_CLICUENTA FOREIGN KEY (cod_sucursal, cod_cuenta) REFERENCES Cuenta (cod_sucursal, cod_cuenta),  
43  CONSTRAINT FK_CLICUENTA_DNI FOREIGN KEY (dni_cli) REFERENCES Cliente (dni_cli)  
44 );
```

Salida de Script x

Tarea terminada en 0,147 segundos

Table SUCURSAL creado.

Table CLIENTE creado.

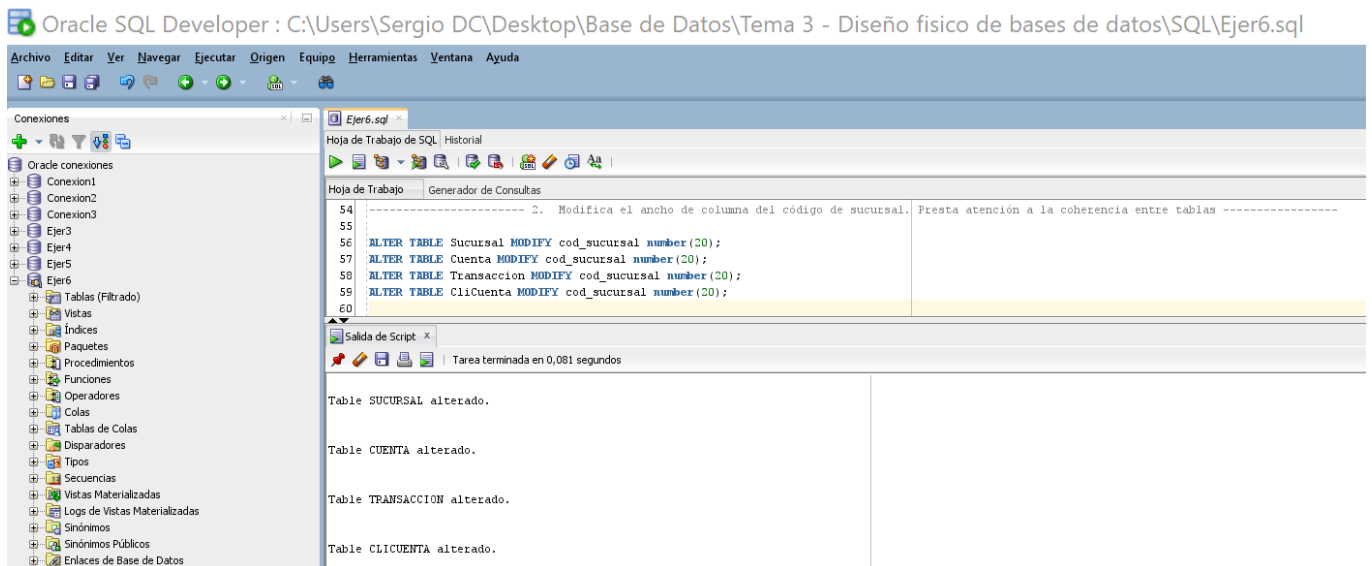
Table CUENTA creado.

Table TRANSACCION creado.

Table CLICUENTA creado.

2. Modifica el ancho de columna del código de sucursal. Presta atención a la coherencia entre tablas

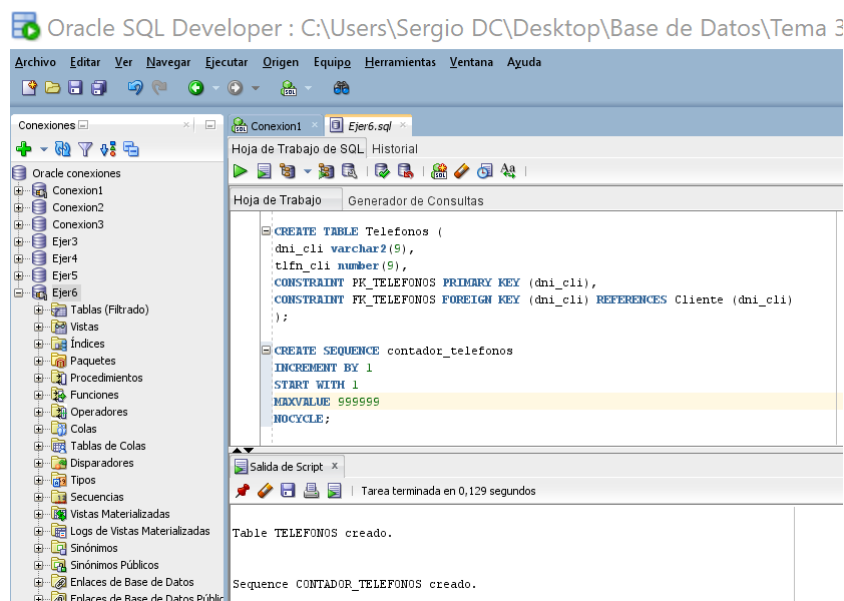
```
ALTER TABLE Sucursal MODIFY cod_sucursal number(20);  
ALTER TABLE Cuenta MODIFY cod_sucursal number(20);  
ALTER TABLE Transaccion MODIFY cod_sucursal number(20);  
ALTER TABLE CliCuenta MODIFY cod_sucursal number(20);
```



3. Añade las restricciones necesarias (utilizando etiquetas para cada restricción):

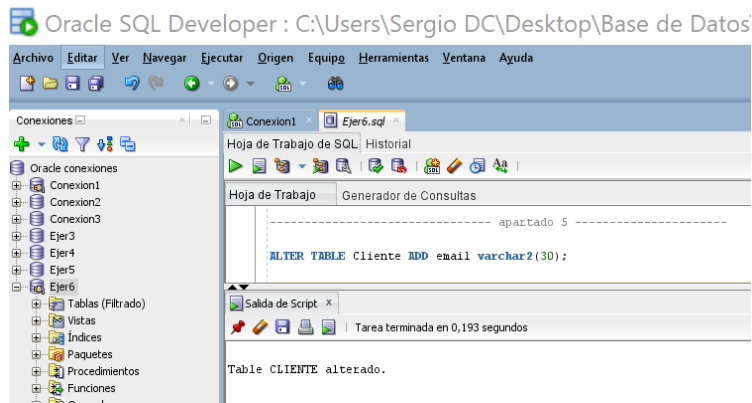
- Claves primarias
- Claves foráneas

4. Añade la tabla teléfonos y enlázalo a clientes, de tal forma que cada cliente pueda tener más de un teléfono asignado. Considerar teléfonos una entidad débil con clave compuesta de DNI y un contador de teléfonos de cada usuario.



5. Añade a la tabla cliente el campo “email”:

- Crearle la restricción necesaria para que compruebe el formato de un correo electrónico (algo@algo.com).



Pista: Esta sería la forma super técnica y restrictiva

```
SELECT * FROM people WHERE email NOT LIKE '%_@_%._%'
```

Next we will run following script which will only select the valid email address:

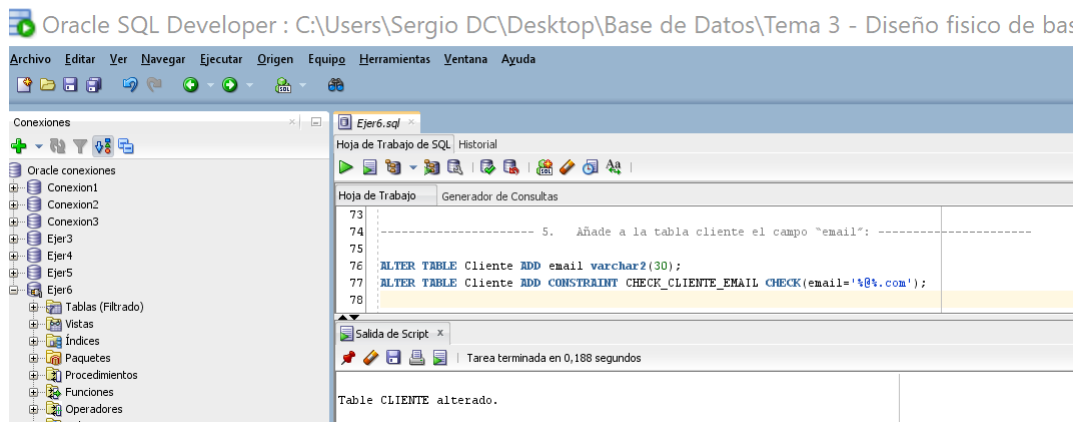
```
1 SELECT EmailAddress AS ValidEmail
2 FROM Contacts
3 WHERE EmailAddress LIKE '%_@_%._%'
4      AND PATINDEX('%[^a-z,0-9,@,.,_,\-%]', EmailAddress) = 0
5 GO
```

We can also use NOT condition in the WHERE clause and select all the invalid emails as well.

```
1 SELECT EmailAddress AS NotValidEmail
2 FROM Contacts
3 WHERE NOT EmailAddress LIKE '%_@_%._%'
4      AND PATINDEX('%[^a-z,0-9,@,.,_,\-%]', EmailAddress) = 0
5 GO
```

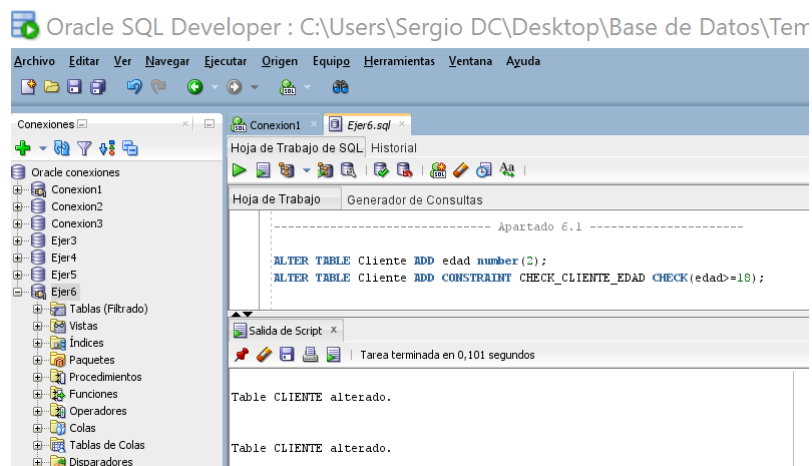
Pista para check format email

ALTER TABLE Cliente ADD CONSTRAINT CHECK_CLIENTE_EMAIL CHECK(email='%@%.com');



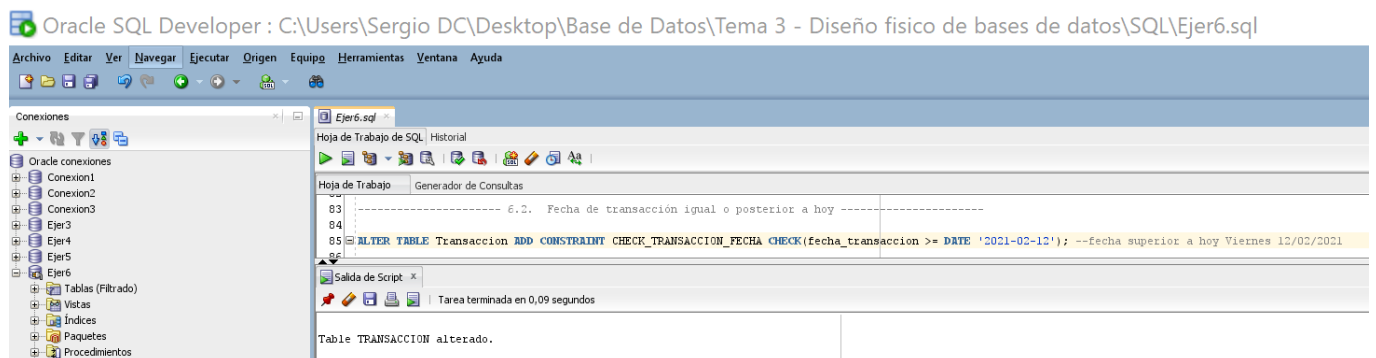
6. Crea los siguientes chequeos en las tablas (restricciones).

6.1. Edad mayor o igual a 18



6.2. Fecha de transacción igual o posterior a hoy

ALTER TABLE Transaccion ADD CONSTRAINT CHECK_TRANSACCION_FECHA CHECK(fecha_transaccion >= DATE '2021-02-12');



6.3. Fecha de nacimiento anterior a hoy.

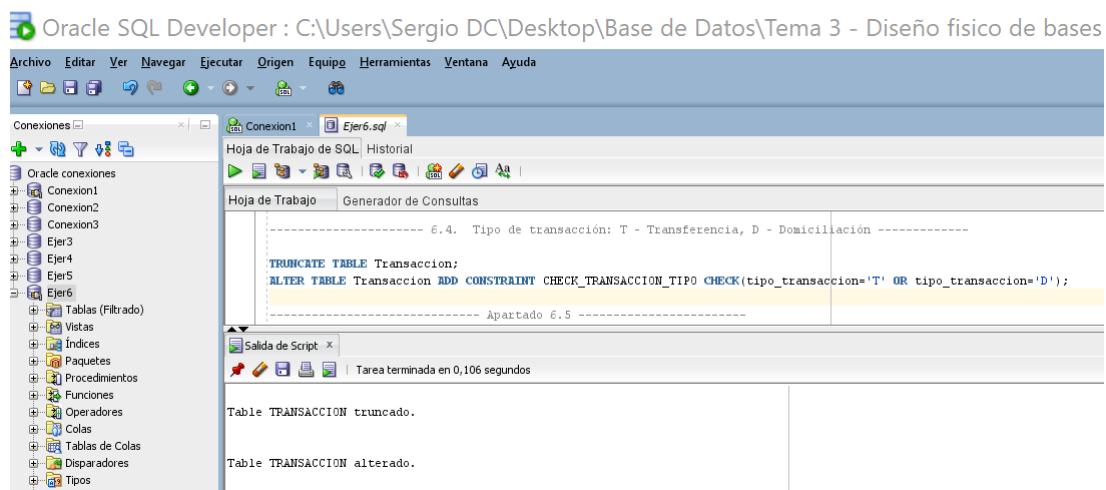
ALTER TABLE *Cliente* **ADD CONSTRAINT** *CHECK_CLIENTE_FECHANAC* **CHECK**(fecha_nac <= DATE '2021-02-12');



6.4. Tipo de transacción: T – Transferencia, D – Domiciliación

TRUNCATE TABLE *Transaccion*;

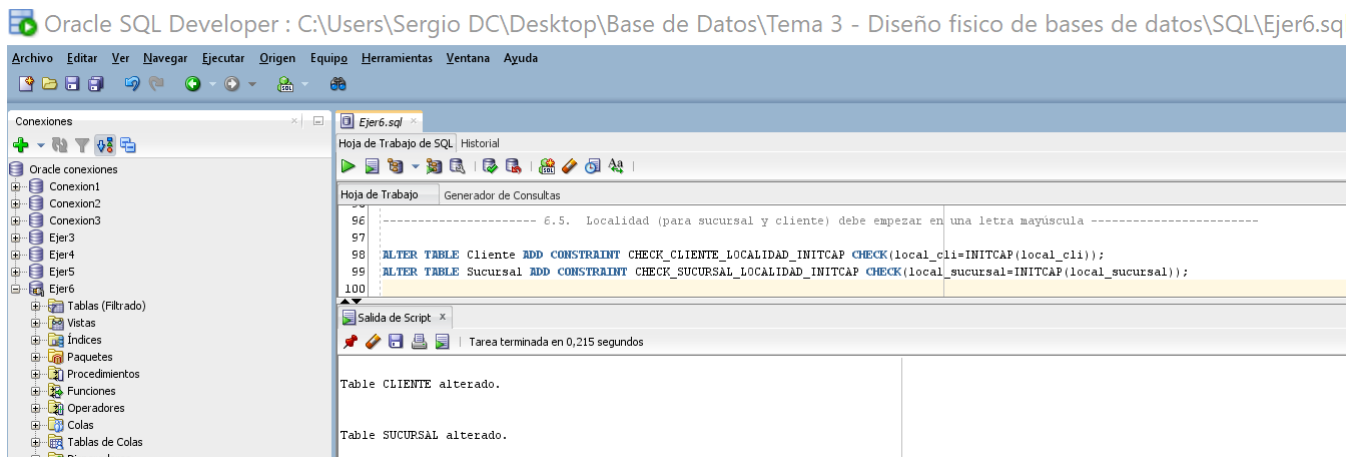
ALTER TABLE *Transaccion* **ADD CONSTRAINT** *CHECK_TRANSACCION_TIPO* **CHECK**(tipo_transaccion='T' OR tipo_transaccion='D');



6.5. Localidad (para sucursal y cliente) debe empezar en una letra mayúscula

ALTER TABLE *Cliente* **ADD CONSTRAINT** *CHECK_CLIENTE_LOCALIDAD_INITCAP* **CHECK**(local_cli=INITCAP(local_cli));

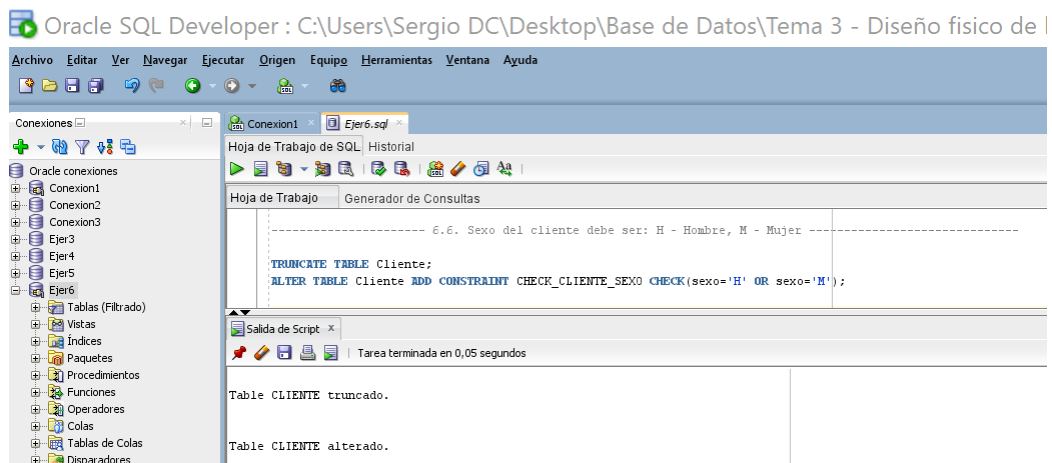
ALTER TABLE *Sucursal* **ADD CONSTRAINT** *CHECK_SUCURSAL_LOCALIDAD_INITCAP* **CHECK**(local_sucursal=INITCAP(local_sucursal));



6.6. Sexo del cliente debe ser: H – Hombre, M – Mujer

TRUNCATE TABLE *Cliente*;

ALTER TABLE *Cliente* **ADD CONSTRAINT** *CHECK_CLIENTE_SEXO* **CHECK**(sexo='H' OR sexo='M');

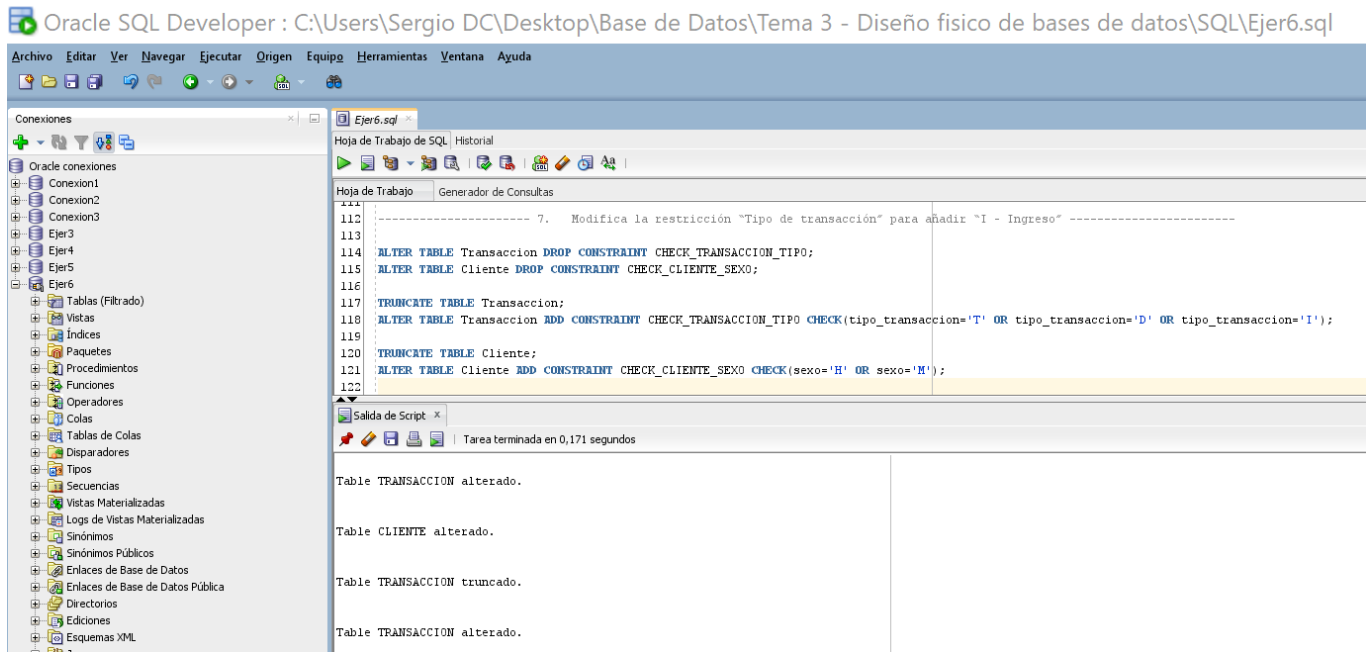


7. Modifica la restricción “Tipo de transacción” para añadir “I – Ingreso”

ALTER TABLE *Transaccion* **DROP CONSTRAINT** *CHECK_TRANSACCION_TIPO*;
ALTER TABLE *Cliente* **DROP CONSTRAINT** *CHECK_CLIENTE_SEXO*;

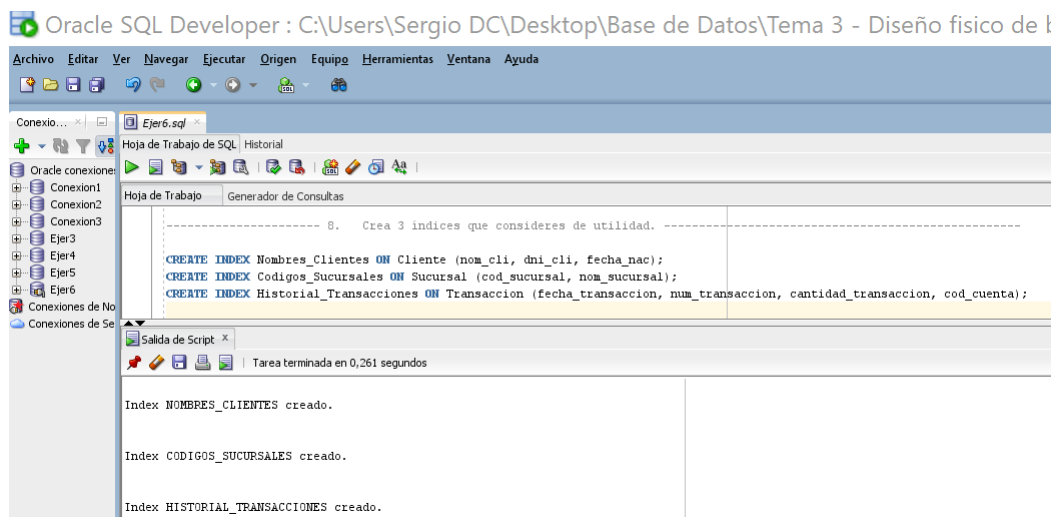
TRUNCATE TABLE *Transaccion*;
ALTER TABLE *Transaccion* **ADD CONSTRAINT** *CHECK_TRANSACCION_TIPO* **CHECK**(tipo_transaccion='T' OR tipo_transaccion='D' OR tipo_transaccion='I');

TRUNCATE TABLE *Cliente*;
ALTER TABLE *Cliente* **ADD CONSTRAINT** *CHECK_CLIENTE_SEXO* **CHECK**(sexo='H' OR sexo='M');



8. Crea 3 índices que consideres de utilidad.

CREATE INDEX *Nombres_Clientes* **ON** *Cliente* (*nom_cli*, *dni_cli*, *fecha_nac*);
CREATE INDEX *Codigos_Sucursales* **ON** *Sucursal* (*cod_sucursal*, *nom_sucursal*);
CREATE INDEX *Historial_Transacciones* **ON** *Transaccion* (*fecha_transaccion*, *num_transaccion*, *cantidad_transaccion*, *cod_cuenta*);



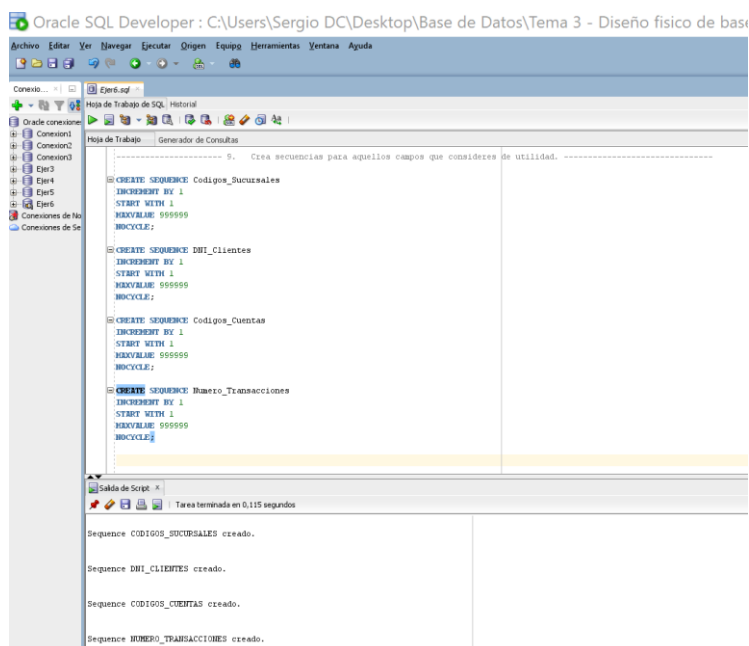
9. Crea secuencias para aquellos campos que consideres de utilidad.

```
CREATE SEQUENCE Codigos_Sucursales
INCREMENT BY 1
START WITH 1
MAXVALUE 999999
NOCYCLE;
```

```
CREATE SEQUENCE DNI_Clientes
INCREMENT BY 1
START WITH 1
MAXVALUE 999999
NOCYCLE;
```

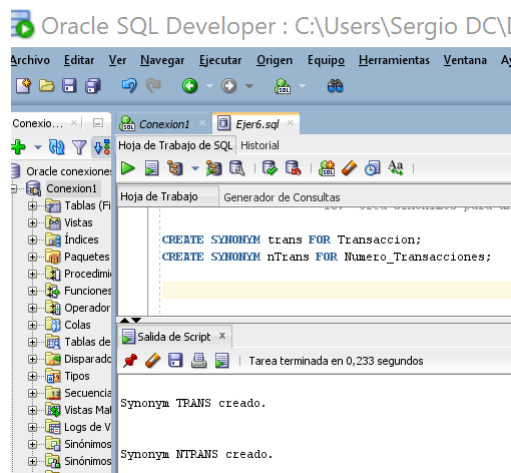
```
CREATE SEQUENCE Codigos_Cuentas
INCREMENT BY 1
START WITH 1
MAXVALUE 999999
NOCYCLE;
```

```
CREATE SEQUENCE Numero_Transacciones
INCREMENT BY 1
START WITH 1
MAXVALUE 999999
NOCYCLE;
```



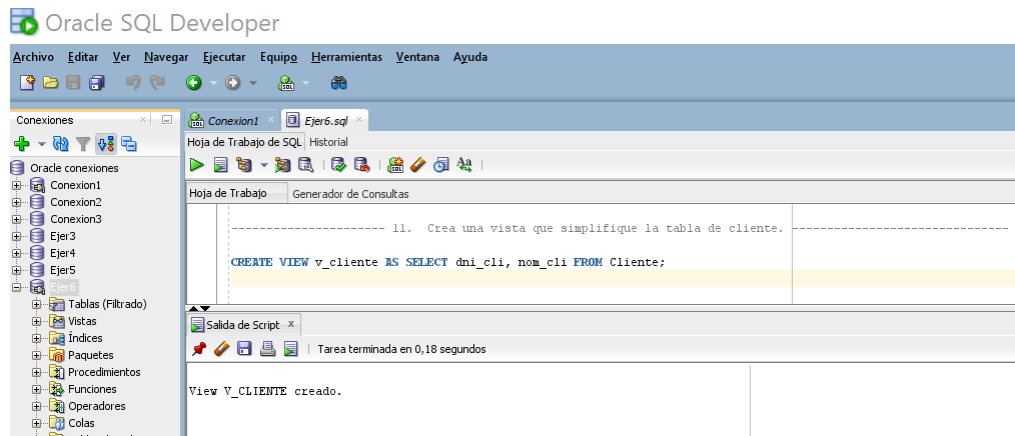
10. Crea sinónimos para una tabla y una secuencia.

```
CREATE SYNONYM trans FOR Transaccion;
CREATE SYNONYM nTrans FOR Numero_Transacciones;
```



11. Crea una vista que simplifique la tabla de cliente.

CREATE VIEW v_cliente AS SELECT dni_cli, nom_cli FROM Cliente;



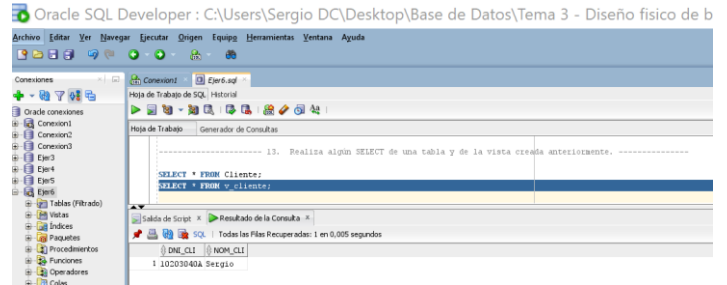
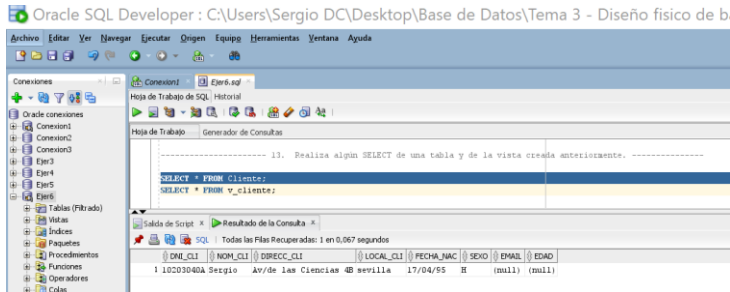
12. Realiza una inserción para cada tabla.

INSERT INTO Sucursal (cod_sucursal, nom_sucursal, direcc_sucursal, local_sucursal)
VALUES(10000, 'sevilla este', 'Av/de las Ciencias N55', 'sevilla');
INSERT INTO Cliente (dni_cli, nom_cli, direcc_cli, local_cli, fecha_nac, sexo) VALUES('10203040A',
'Sergio', 'Av/de las Ciencias 4B', 'sevilla', '17041995', 'H');
INSERT INTO Cuenta (cod_sucursal, cod_cuenta) VALUES(10000, 25053006);
INSERT INTO Transaccion (cod_sucursal, cod_cuenta, num_transaccion, fecha_transaccion,
cantidad_transaccion, tipo_transaccion) VALUES(10000, 25053006, 11111, '13022021', 10.50, 'T');
INSERT INTO CliCuenta (cod_sucursal, cod_cuenta, dni_cli) VALUES(10000, 25053006,
'10203040A');
INSERT INTO Telefonos (dni_cli, tlf_n_cli) VALUES('10203040A', 684367181);



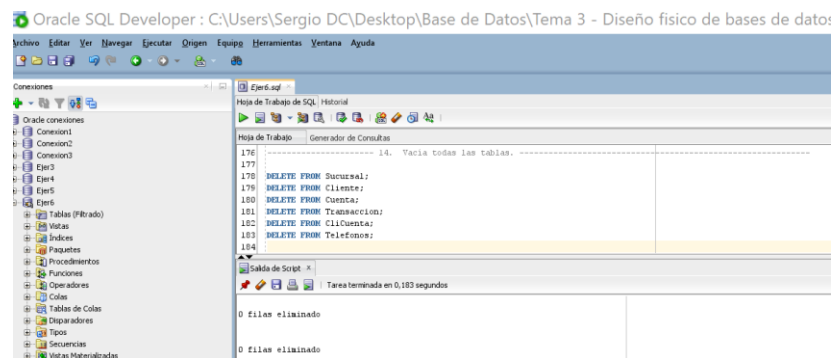
13. Realiza algún SELECT de una tabla y de la vista creada anteriormente.

```
SELECT * FROM Cliente;  
SELECT * FROM v_cliente;
```



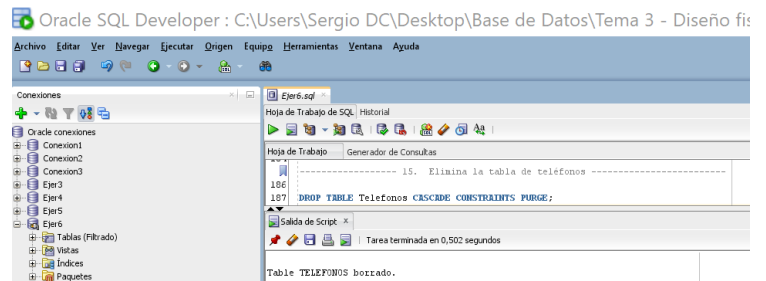
14. Vacía todas las tablas.

```
DELETE FROM Sucursal;  
DELETE FROM Cliente;  
DELETE FROM Cuenta;  
DELETE FROM Transaccion;  
DELETE FROM CliCuenta;  
DELETE FROM Telefonos;
```



15. Elimina la tabla de teléfonos

```
DROP TABLE Telefonos CASCADE CONSTRAINTS PURGE;
```



NOTA: Comenta a que se deben las posibles dificultades que aparecen en la realización de los ejercicios y explica como las has resuelto.