



FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES

CARRERA: Computación

ASIGNATURA: Programación Aplicada

NRO. PRÁCTICA: **TÍTULO PRÁCTICA:** Base de datos

OBJETIVO ALCANZADO:

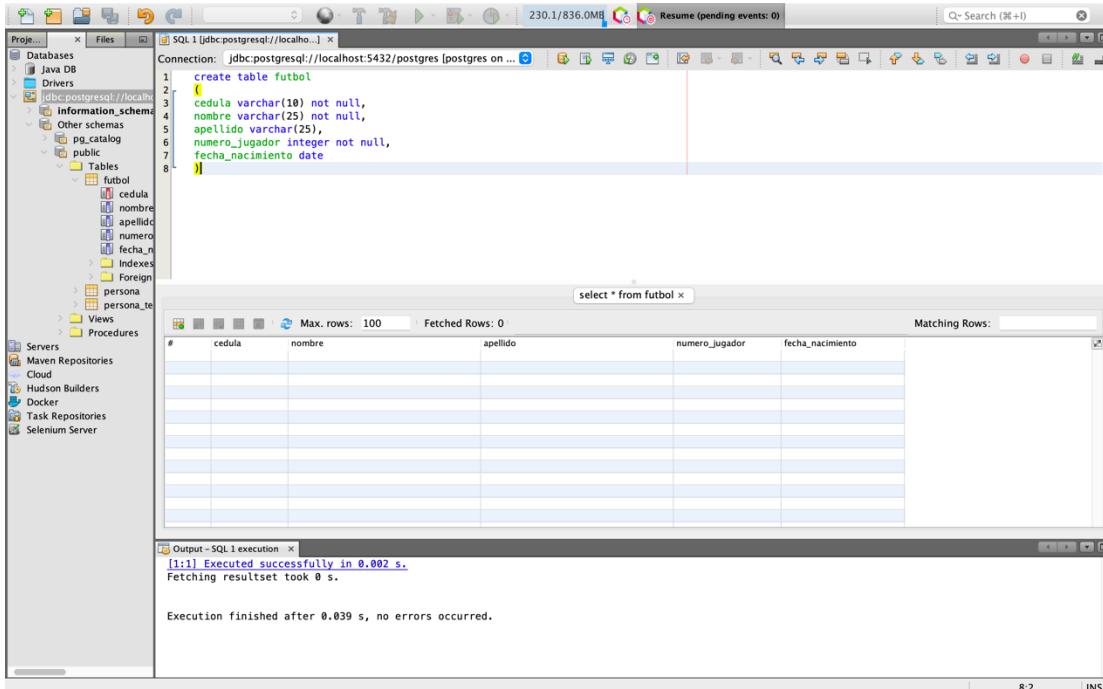
Identificar las sentencias SQL

Diseñar e Implementar códigos DDL, DML

Entender cada una de las características del uso de una Base de Datos

ACTIVIDADES DESARROLLADAS

- 2.1 Actividad: Realizar una base de datos para el equipo de futbol del curso.



The screenshot shows the MySQL Workbench interface. On the left, the database structure is visible, including the 'futbol' table under the 'Information_schema' schema. In the main panel, a SQL editor window displays the following code:

```

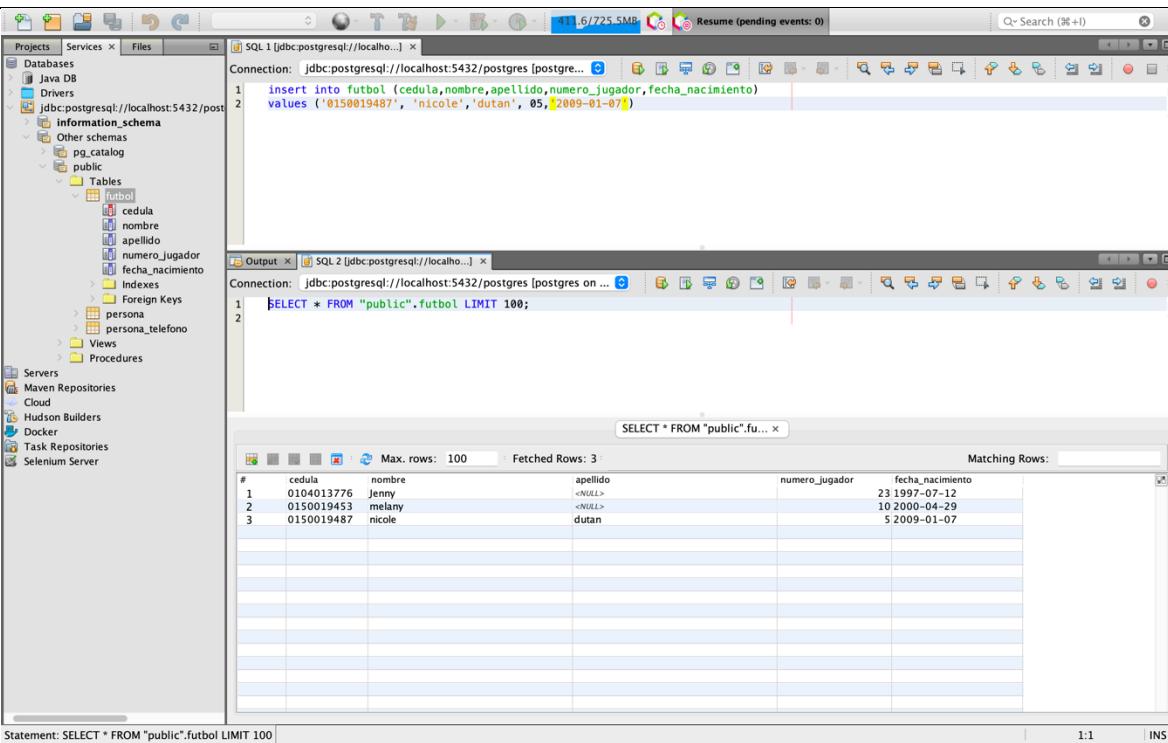
create table futbol
(
    razon_social varchar(10) not null,
    nombre varchar(25) not null,
    apellido varchar(25),
    numero_jugador integer not null,
    fecha_nacimiento date
);
  
```

Below the editor, a results grid shows columns: #, cedula, nombre, apellido, numero_jugador, and fecha_nacimiento. The output pane at the bottom shows the execution results:

```

[1:1] Executed successfully in 0.002 s.
Fetching resultset took 0 s.

Execution finished after 0.039 s, no errors occurred.
  
```



The screenshot shows the SQL Workbench/J interface. On the left, the database structure is visible, including the 'public' schema which contains the 'futbol' table. The 'futbol' table has columns: cedula, nombre, apellido, numero_jugador, and fecha_nacimiento. In the main pane, SQL Statement 1 shows the insertion of a new row into the 'futbol' table:

```
1 insert into futbol (cedula,nombre,apellido,numero_jugador,fecha_nacimiento)
2 values ('0150019487', 'nicole','dutan', 05,[2009-01-07])
```

SQL Statement 2 shows a query to select all rows from the 'futbol' table:

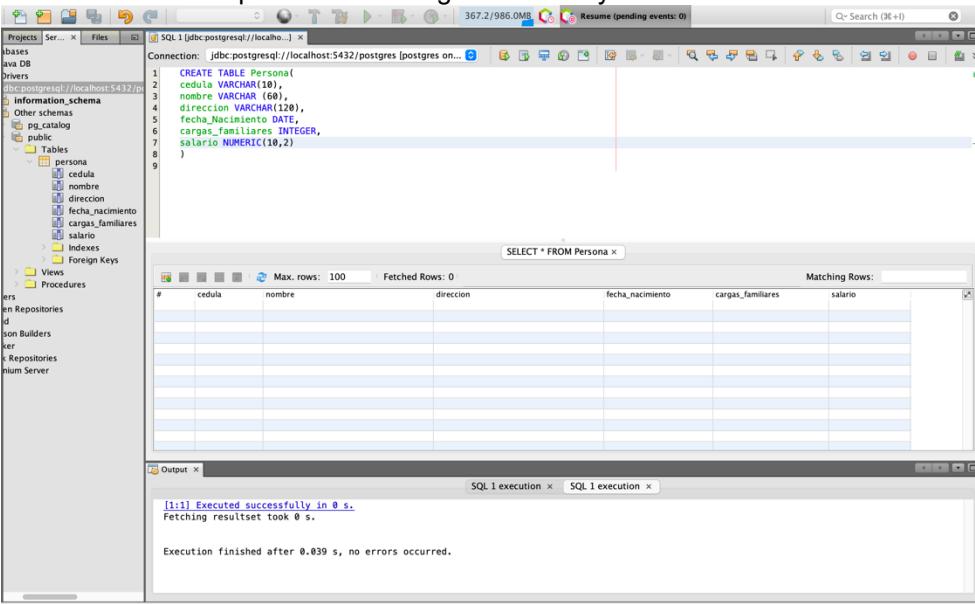
```
1 SELECT * FROM "public".futbol LIMIT 100;
```

The results of the SELECT query are displayed in a table:

#	cedula	nombre	apellido	numero_jugador	fecha_nacimiento
1	0104013776	Jenny	<NULL>		23 1997-07-12
2	0150019453	melyny	<NULL>		10 2000-04-29
3	0150019487	nicole	dutan		5 2009-01-07

2. 2.2 Crear la tabla PERSONA, con los siguientes datos:

- Cedula con una dimensión de 10 caracteres
- Nombre con una dimensión de 60 caracteres
- Dirección con una dimensión de 120 caracteres
- Fecha de nacimiento de tipo fecha
- Número de cargas familiares
- Salario de tipo real con 10 dígitos enteros y dos decimales



The screenshot shows the SQL Workbench/J interface. On the left, the database structure is visible, including the 'public' schema which contains the 'PERSONA' table. The 'PERSONA' table has columns: cedula, nombre, dirección, fecha_nacimiento, cargas_familiares, and salario. In the main pane, SQL Statement 1 shows the creation of the 'PERSONA' table:

```
1 CREATE TABLE Persona(
2     cedula VARCHAR(10),
3     nombre VARCHAR (60),
4     direccion VARCHAR(120),
5     fecha_Nacimiento DATE,
6     cargas_familiares INTEGER,
7     salario NUMERIC(10,2)
8 )
```

The results of the CREATE TABLE statement are displayed in a table:

#	cedula	nombre	direccion	fecha_nacimiento	cargas_familiares	salario

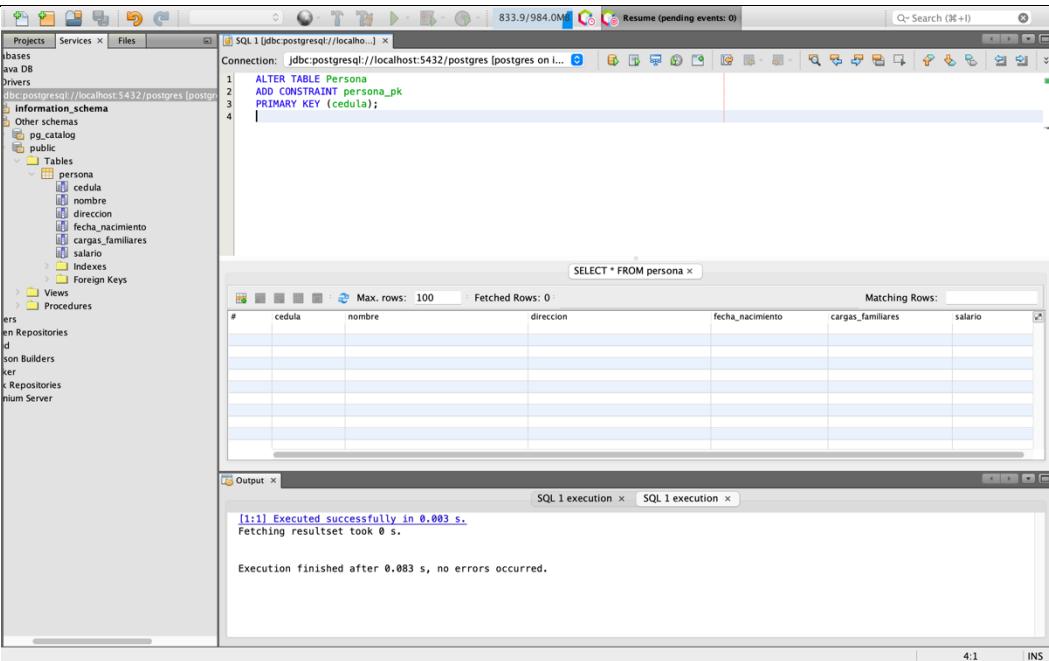
SQL Statement 2 shows the execution of the CREATE TABLE statement:

```
[1:1] Executed successfully in 0 s.
Fetching resultset took 0 s.
```

The message indicates that the execution was successful and took 0 seconds.

3. Modificar la tabla PERSONA de tal manera que el campo cedula sea clave primaria.

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



The screenshot shows the pgAdmin III interface. On the left, the object browser displays the database structure, including the 'information_schema' and 'public' schemas, and the 'persona' table under 'Tables'. In the center, a SQL editor window contains the following code:

```

1 ALTER TABLE Persona
2 ADD CONSTRAINT persona_pk
3 PRIMARY KEY (cedula);
4

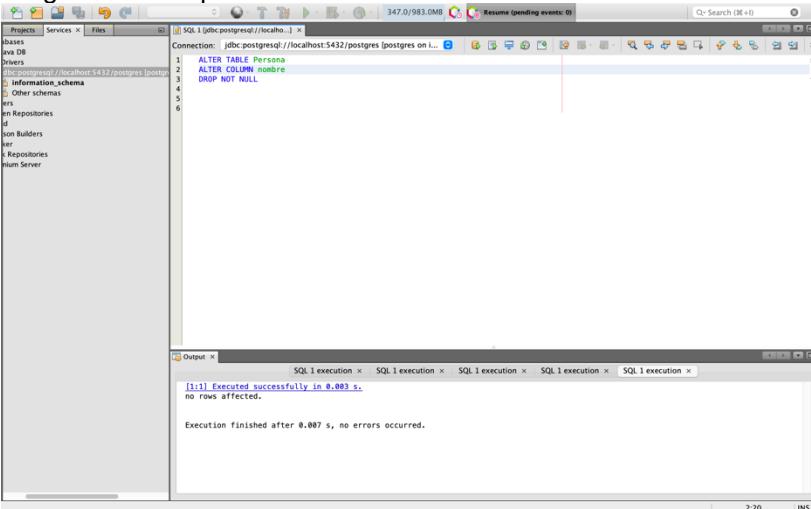
```

Below the SQL editor is a result grid titled "SELECT * FROM persona x" with columns: #, cedula, nombre, dirección, fecha_nacimiento, cargas_familiares, and salario. The output panel at the bottom shows the execution results:

[1:1] Executed successfully in 0.003 s.
Fetching resultset took 0 s.

Execution finished after 0.003 s, no errors occurred.

4. Modificar la tabla PERSONA de tal manera que los campos nombre, fecha de nacimiento sean obligatorios o requeridos



The screenshot shows the pgAdmin III interface. On the left, the object browser displays the database structure, including the 'information_schema' and 'public' schemas, and the 'persona' table under 'Tables'. In the center, a SQL editor window contains the following code:

```

1 ALTER TABLE Persona
2 ALTER COLUMN nombre
3 DROP NOT NULL;
4
5
6

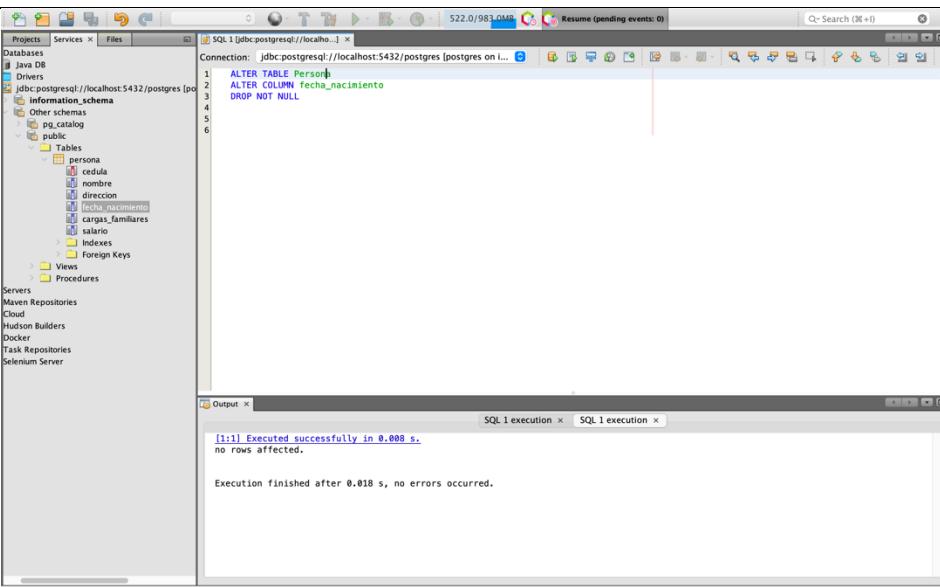
```

The output panel at the bottom shows the execution results:

[1:1] Executed successfully in 0.003 s.
no rows affected.

Execution finished after 0.007 s, no errors occurred.

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



```

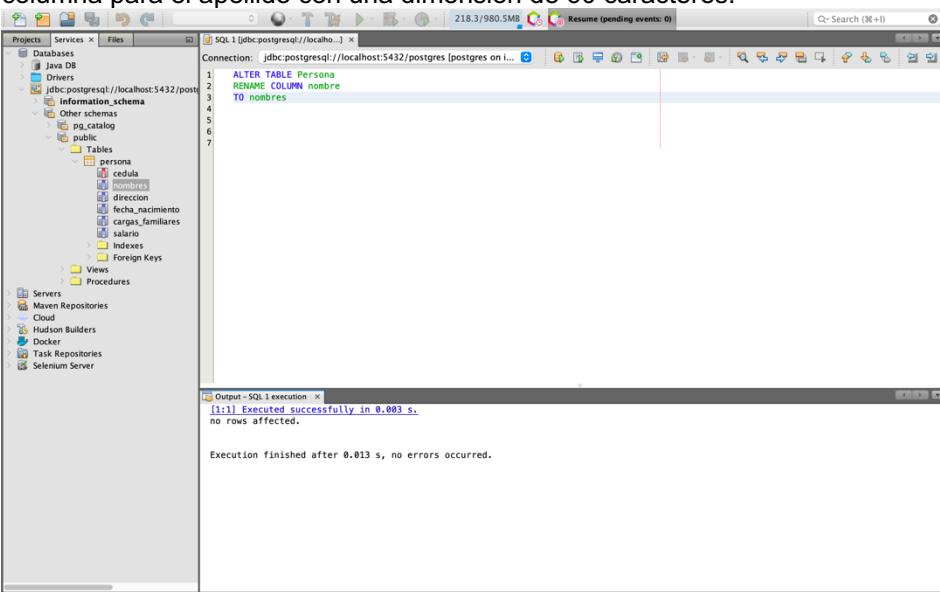
1: ALTER TABLE Persona
2:   ALTER COLUMN nombre
3:   DROP NOT NULL;
4:
5:
6:

```

[1:1] Executed successfully in 0.008 s.
no rows affected.

Execution finished after 0.018 s, no errors occurred.

5. Modificar la tabla PERSONA de tal manera la columna nombre cambie a nombres, y se añada una columna para el apellido con una dimensión de 60 caracteres.



```

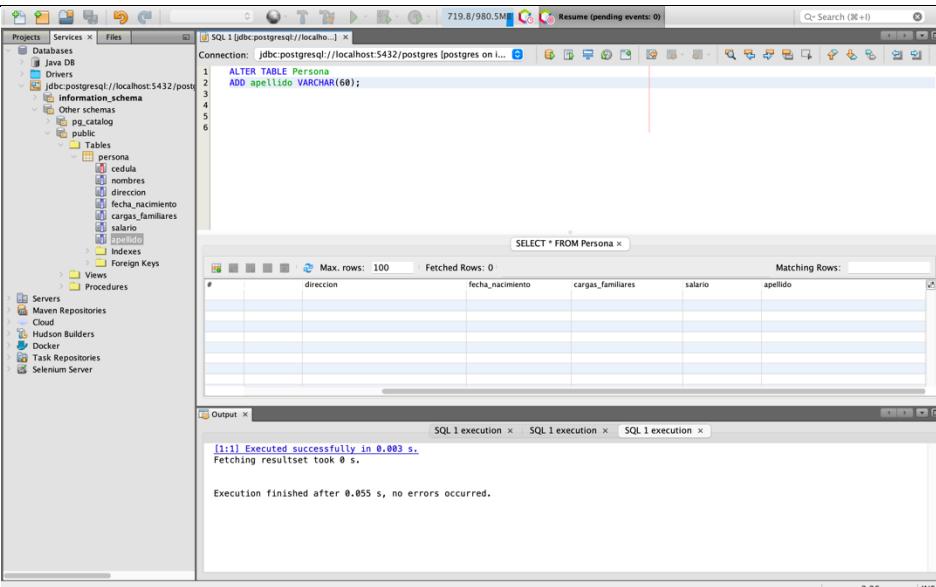
1: ALTER TABLE Persona
2:   RENAME COLUMN nombre
3:   TO nombres;
4:
5:
6:
7:

```

[1:1] Executed successfully in 0.003 s.
no rows affected.

Execution finished after 0.013 s, no errors occurred.

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



```

ALTER TABLE Persona
ADD apellido VARCHAR(60);

```

Output:

```

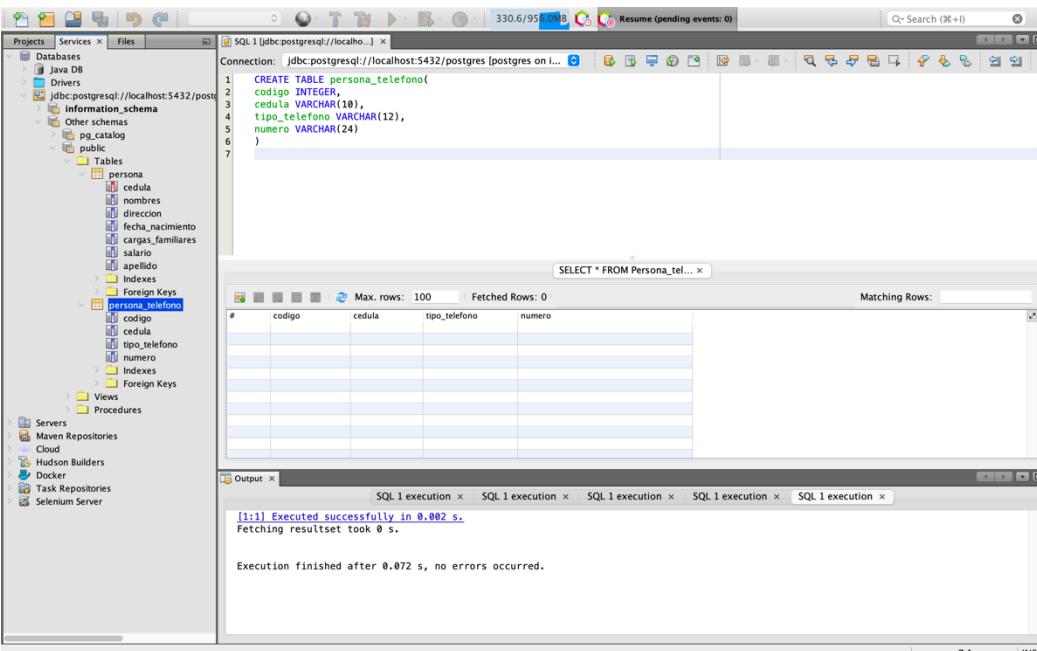
[1:1] Executed successfully in 0.003 s.
Fetching resultset took 0 s.

Execution finished after 0.055 s, no errors occurred.

```

6. Crear la tabla PERSONA_TELEFONO, con los siguientes datos:

- Código, de tipo entero
- Cedula, con una dimensión de 10 caracteres (es la referencia de la tabla persona)
- Tipo de teléfono, con una dimensión de 12 caracteres, aquí se guardarán valores como celular o convencional
- Número de teléfono, con una dimensión de 24 caracteres



```

CREATE TABLE persona_telefono{
1   codigo INTEGER,
2   cedula VARCHAR(10),
3   tipo_telefono VARCHAR(12),
4   numero VARCHAR(24)
5 }

```

Output:

```

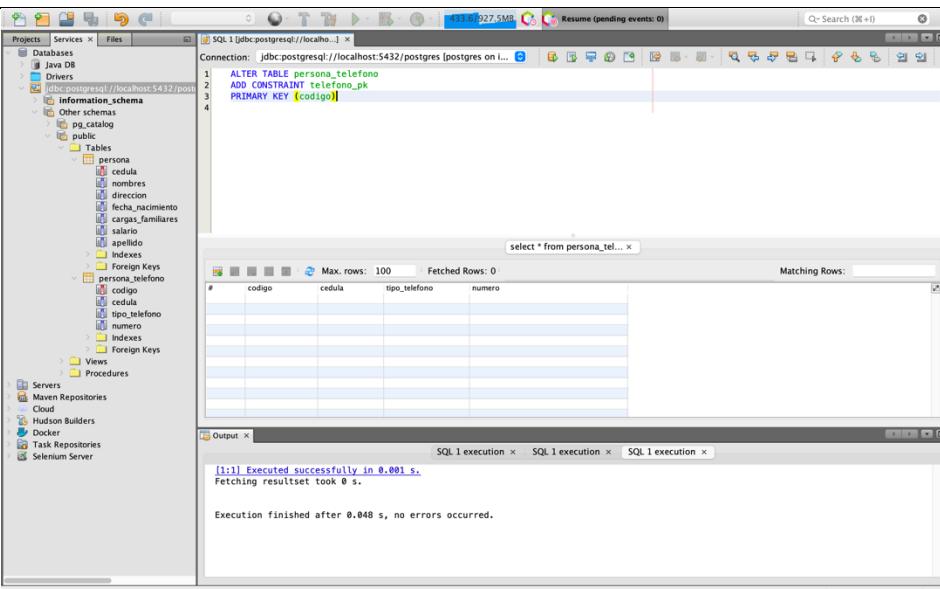
[1:1] Executed successfully in 0.002 s.
Fetching resultset took 0 s.

Execution finished after 0.072 s, no errors occurred.

```

7. Modificar la tabla PERSONA_TELEFONO de tal manera que el campo código sea clave primaria.

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



The screenshot shows the MySQL Workbench interface. On the left, the database structure is visible, including the `information_schema` and `public` schemas. In the center, the `SQL` editor contains the following SQL code:

```

1 ALTER TABLE persona_telefono
2 ADD CONSTRAINT telefono_pk
3 PRIMARY KEY (codigo);
4

```

Below the SQL editor, a results grid displays the structure of the `persona_telefono` table:

#	codigo	cedula	tipo_telefono	numero
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

The output panel at the bottom shows the execution results:

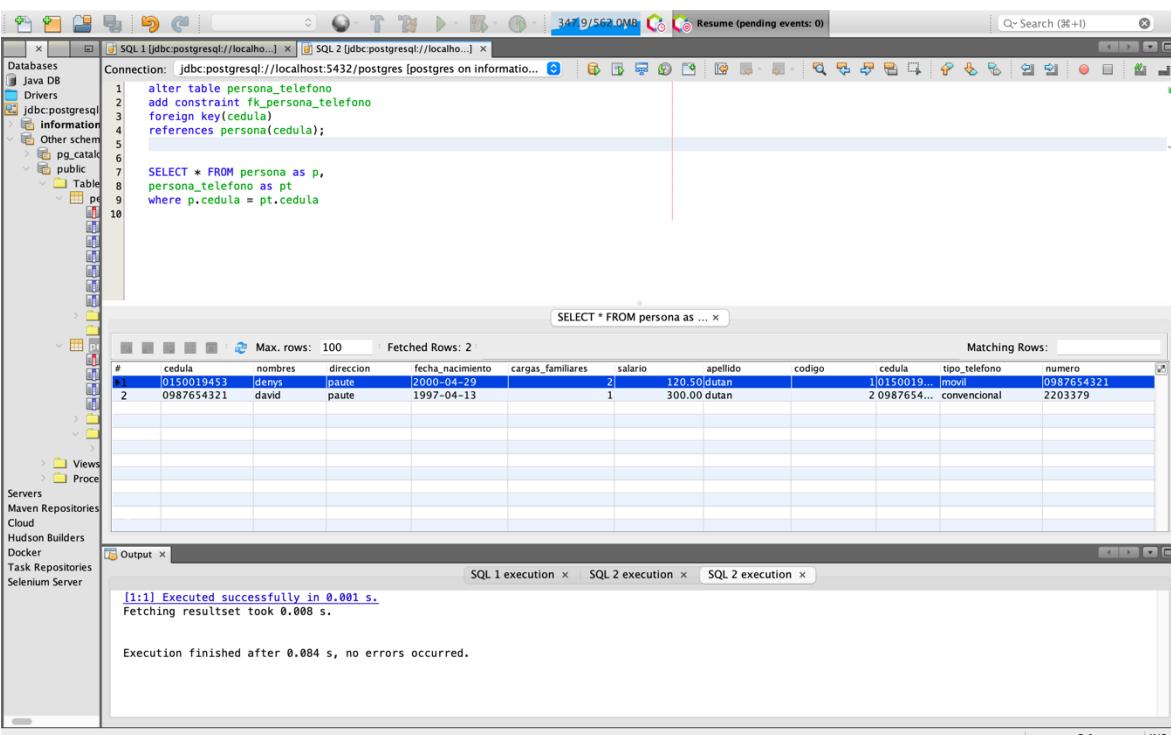
```

[1:1] Executed successfully in 0.001 s.
Fetching resultset took 0.5.

Execution finished after 0.048 s, no errors occurred.

```

8. Modificar la tabla PERSONA_TELEFONO de tal manera que se incluya la referencia a la tabla PERSONA por medio del campo cedula



The screenshot shows the MySQL Workbench interface. On the left, the database structure is visible, including the `information_schema` and `public` schemas. In the center, the `SQL` editor contains the following SQL code:

```

1 alter table persona_telefono
2 add constraint fk_persona_telefono
3 foreign key(cedula)
4 references persona(cedula);
5
6
7 SELECT * FROM persona as p,
8 persona_telefono as pt
9 where p.cedula = pt.cedula
10

```

Below the SQL editor, a results grid displays the data from the `persona` table:

#	cedula	nombres	direccion	fecha_nacimiento	cargas_familiares	salario	apellido	codigo	cedula	tipo_telefono	numero
1	0150019453	denys	pauete	2000-04-29	2	120.50	dutan	10150019...	movil	0987654321	
2	0987654321	david	pauete	1997-04-13	1	300.00	dutan	20987654...	convencional	2203379	

The output panel at the bottom shows the execution results:

```

[1:1] Executed successfully in 0.001 s.
Fetching resultset took 0.008 s.

Execution finished after 0.084 s, no errors occurred.

```

9. 2.3 SELECT de todos los registros de personas

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

The screenshot shows the SQL Workbench/J application interface. On the left is a tree view of the database schema, including 'Databases' (Java DB, JDBC PostgreSQL), 'Other schemas' (information_schema, pg_catalog, public), and 'Tables' (person). The 'Tables' section contains columns for cedula, nombres, dirección, fecha_nacimiento, cargas_familiares, salario, apellido, and Foreign Keys (persona_telefono). A query window on the right displays the results of the SQL command 'select * from persona'. The results show five rows of data:

#	dirección	fecha_nacimiento	cargas_familiares	salario	apellido	
1	Cuenca	1977-07-12	<NULL>	500.00	sanchez	
2	<NULL>	2009-02-17	<NULL>	800.00	dutan	
3	<NULL>	2000-08-17	<NULL>	720.00	dutan	
4	paupe	2000-04-29	<NULL>	2	120.50	dutan
5	paupe	1997-04-13		1	300.00	dutan

The output pane at the bottom shows the execution results:

```
[1:1] Executed successfully in 0.002 s.
Fetching resultset took 0 s.

Execution finished after 0.078 s, no errors occurred.
```

10. Selecciona los nombre y apellidos de las personas cuyo salario es mayor a 500 USD.

The screenshot shows the SQL Workbench/J application interface. On the left is a tree view of the database schema, including 'Databases' (Java DB, JDBC PostgreSQL), 'Other schemas' (information_schema, pg_catalog, public), and 'Tables' (person). The 'Tables' section contains columns for cedula, nombres, dirección, fecha_nacimiento, cargas_familiares, salario, apellido, and Foreign Keys (persona_telefono). A query window on the right displays the results of the SQL command 'select nombres, apellido from persona where salario > 500'. The results show two rows of data:

#	nombres	apellido
1	melany	dutan
2	leslie	dutan

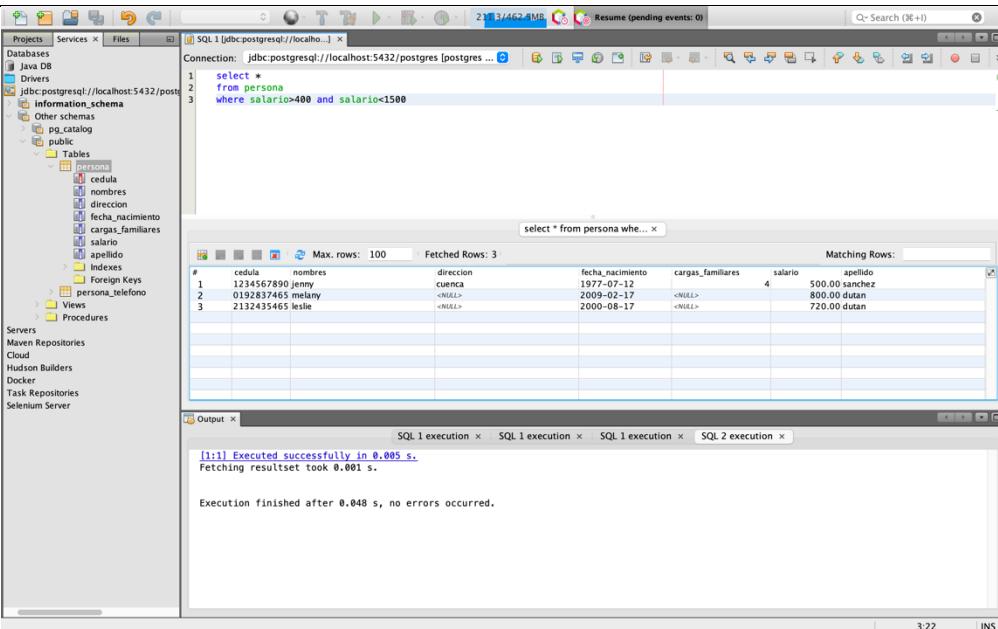
The output pane at the bottom shows the execution results:

```
[1:1] Executed successfully in 0.007 s.
Fetching resultset took 0 s.

Execution finished after 0.062 s, no errors occurred.
```

11. Selecciona todos los datos de las personas cuyo salario es mayor a 400 USD y menor a 1500.

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



The screenshot shows the MySQL Workbench interface. On the left, the database structure is visible, including the 'information_schema' and 'public' schemas. The 'public' schema contains a 'personas' table with columns: cedula, nombres, dirección, fecha_nacimiento, cargas_familiares, salario, apellido, and apellidos. A query is being run in the SQL editor:

```
1 select *
2   from persona
3  where salario>400 and salario<1500
```

The results grid shows three rows of data:

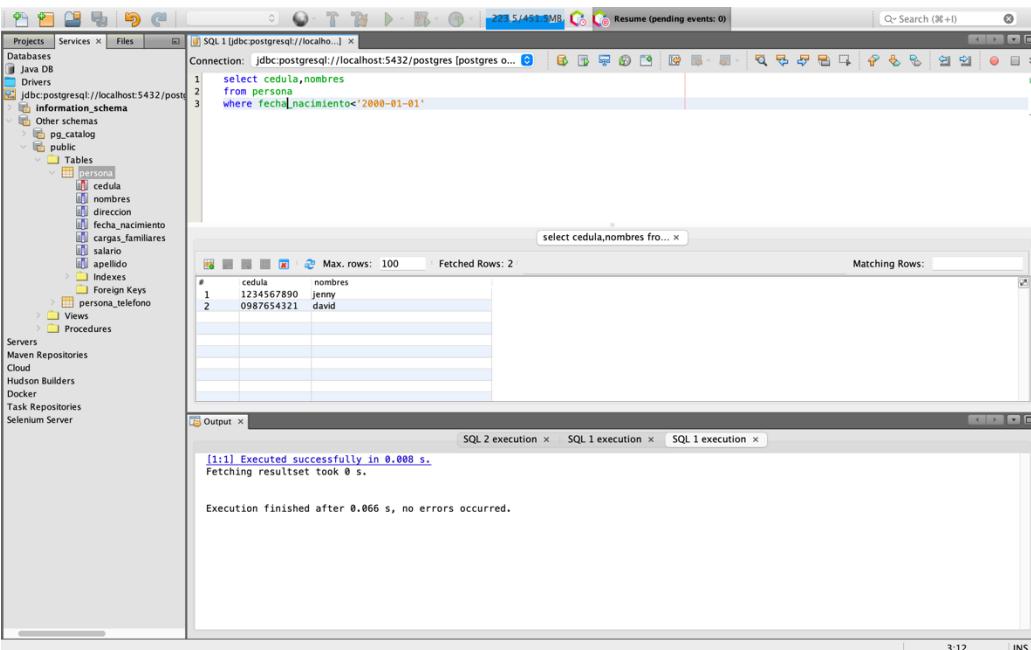
#	cedula	nombres	direccion	fecha_nacimiento	cargas_familiares	salario	apellido
1	1234567890	Jenny	cuenta	1977-07-12	<NULL>	500.00	sanchez
2	0192837465	melany	<NULL>	2009-02-17	<NULL>	800.00	dutan
3	2132435465	leslie	<NULL>	2000-08-17	<NULL>	720.00	dutan

The output window shows the execution details:

```
[1:1] Executed successfully in 0.005 s.
Fetching resultset took 0.001 s.

Execution finished after 0.048 s, no errors occurred.
```

12. Seleccione el número de cedula y nombres de las personas que hayan nacido antes del año 2000.



The screenshot shows the MySQL Workbench interface. On the left, the database structure is visible, including the 'information_schema' and 'public' schemas. The 'public' schema contains a 'personas' table with columns: cedula, nombres, dirección, fecha_nacimiento, cargas_familiares, salario, apellido, and apellidos. A query is being run in the SQL editor:

```
1 select cedula,nombres
2   from persona
3  where fecha_nacimiento<'2000-01-01'
```

The results grid shows two rows of data:

#	cedula	nombres
1	1234567890	Jenny
2	0987654321	david

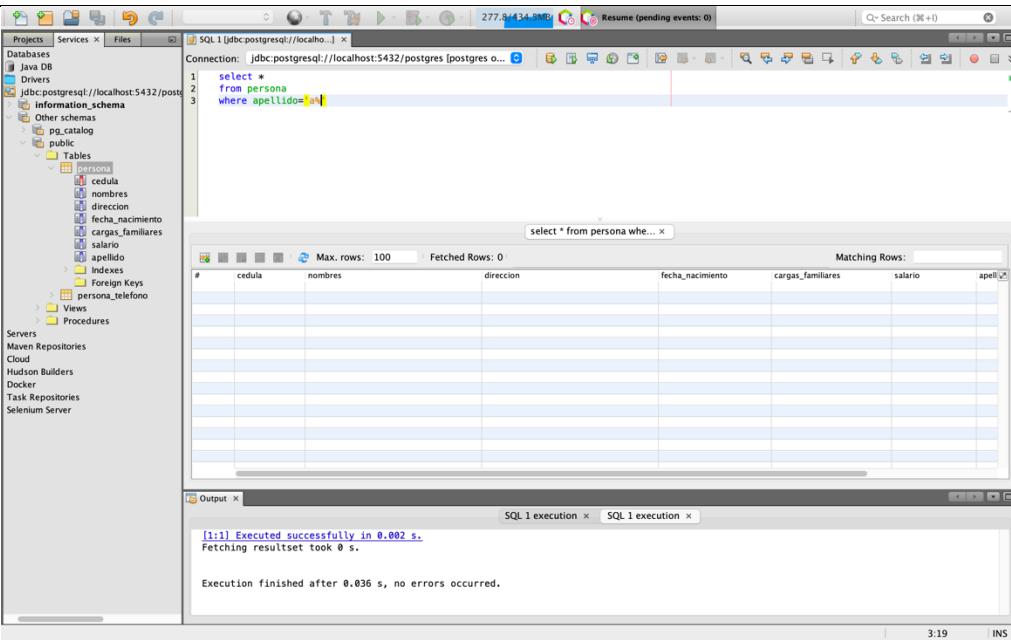
The output window shows the execution details:

```
[1:1] Executed successfully in 0.008 s.
Fetching resultset took 0 s.

Execution finished after 0.066 s, no errors occurred.
```

13. Seleccione los datos de las personas cuyo apellido empiece con la letra "a".

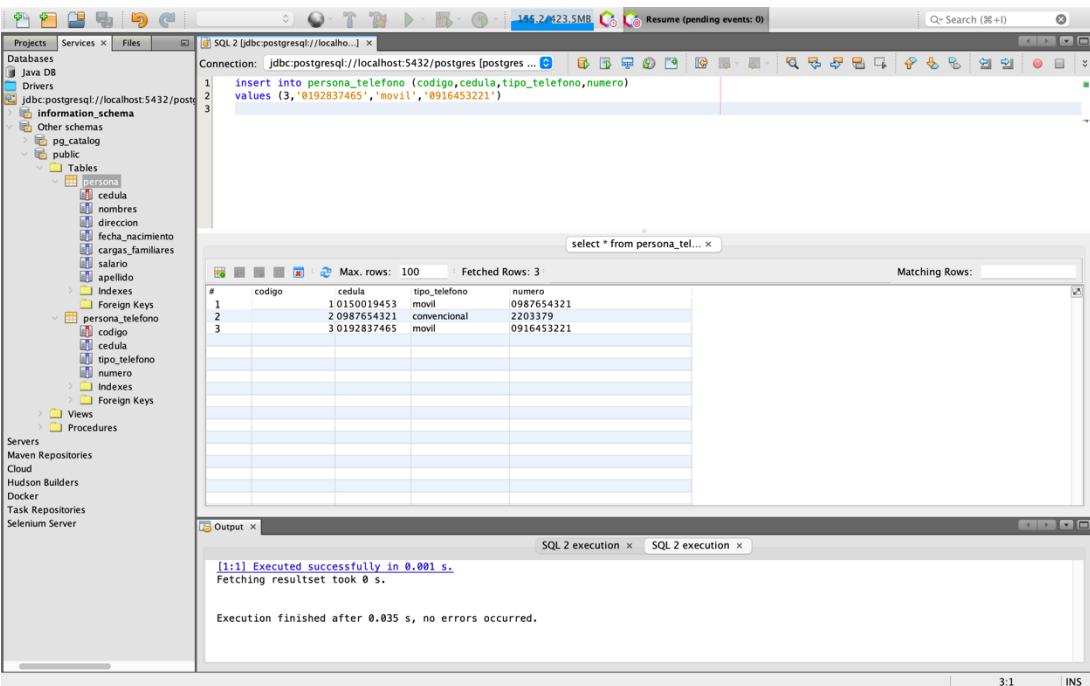
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



```
1 select *
2   from persona
3  where apellido like %a%
```

The screenshot shows the SQL Workbench interface. On the left is a tree view of the database schema, including 'information_schema', 'pg_catalog', and 'public' schemas, with 'Tables' expanded to show 'persona'. The main window contains a SQL editor with the above query and its results. The results grid has columns: #, cedula, nombres, direccion, fecha_nacimiento, cargas_familiares, salario, and apellido. The output panel at the bottom shows the execution message: '[1:1] Executed successfully in 0.002 s.'

14. De su conjunto de personas de su base de datos inserte uno número de teléfono para 3 de ellas, esto en la tabla persona_telefono.

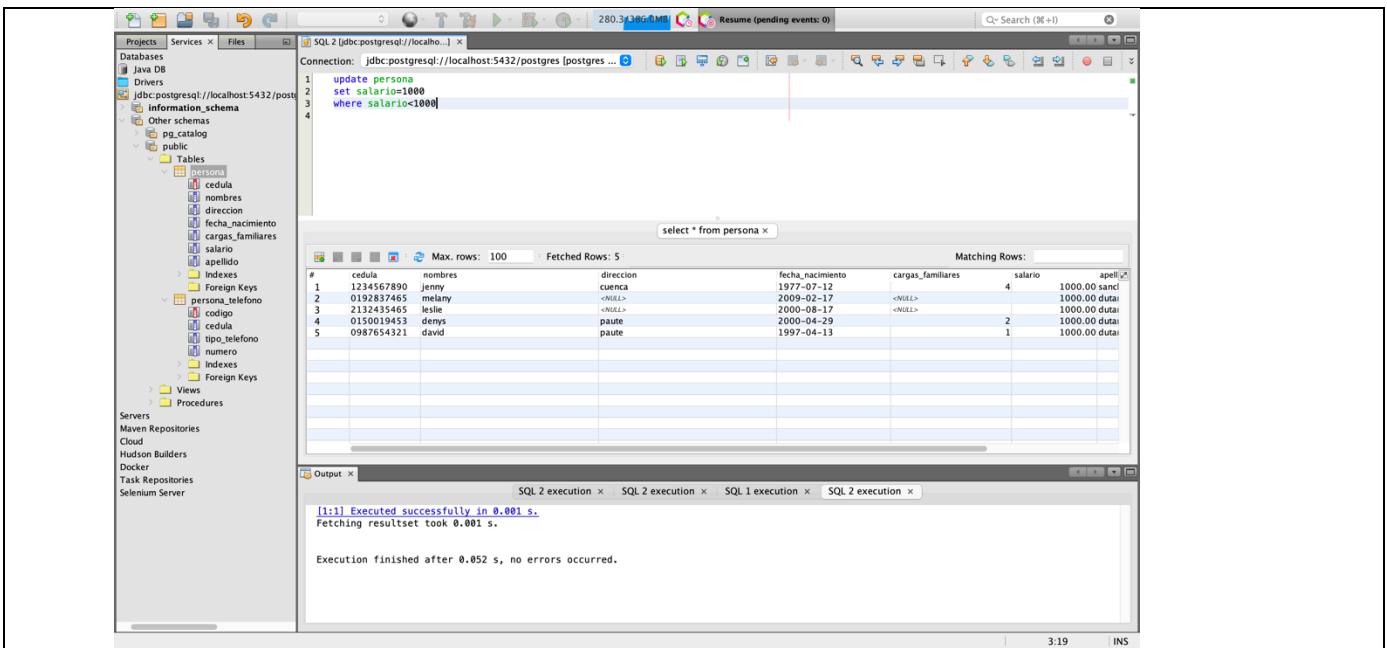


```
1 insert into persona_telefono (codigo,cedula,tipo_telefono,numero)
2 values (3,'0192837465','movil','0916453221')
3
```

The screenshot shows the SQL Workbench interface. On the left is a tree view of the database schema, including 'information_schema', 'pg_catalog', and 'public' schemas, with 'Tables' expanded to show 'persona' and 'persona_telefono'. The main window contains a SQL editor with the above query and its results. The results grid has columns: #, codigo, cedula, tipo_telefono, and numero. The output panel at the bottom shows the execution message: '[1:1] Executed successfully in 0.001 s.'

15. Cambie el salario a 1000 USD a aquellas personas cuyo salario sea menor a 1000.

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



The screenshot shows the SQL Workbench/J interface. In the top-left pane, the database structure is visible, including the `personas` table under the `public` schema. In the main SQL editor pane, the following UPDATE statement is executed:

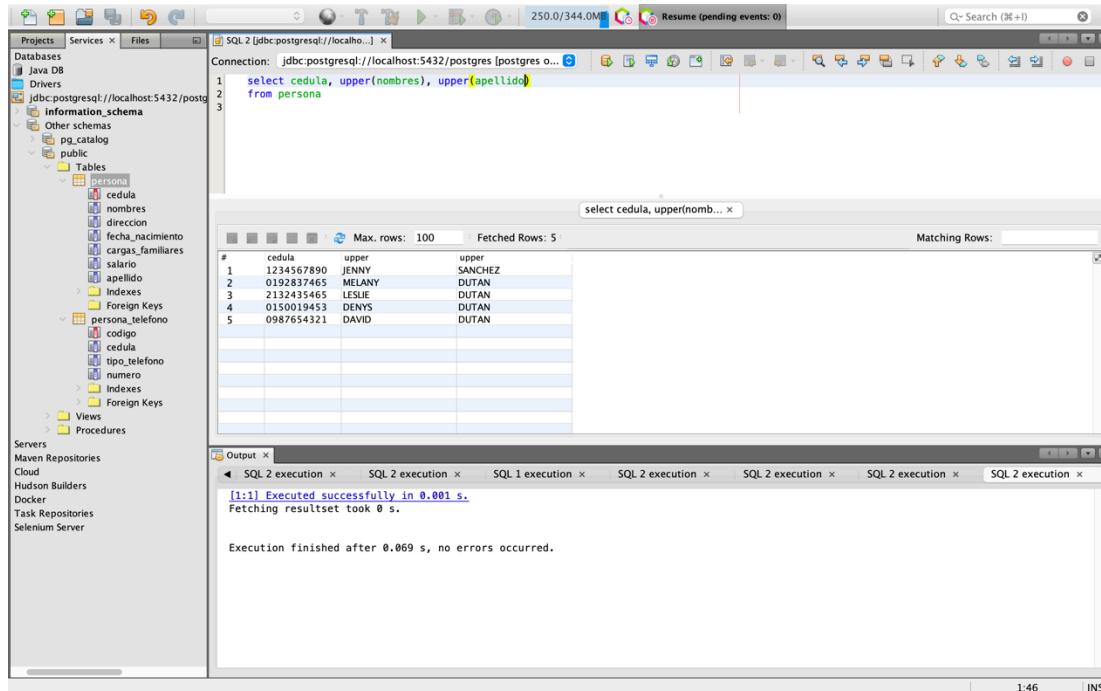
```
1 update persona
2 set salario=1000
3 where salario<1000
4
```

The results of the query are displayed in a table below:

#	Cedula	nombres	direccion	fecha_nacimiento	cargas_familiares	salario	apellido
1	1234567890	JENN	CURENA	1977-07-12	4	1000.00	sand
2	0192837465	MELANY	<NULL>	2009-02-17	<NULL>	1000.00	dutal
3	2132435465	leslie'	<NULL>	2000-08-17	<NULL>	1000.00	dutal
4	0150019453	DENYS	paute	2000-04-29	2	1000.00	dutal
5	0987654321	david	paute	1997-04-13	1	1000.00	dutal

In the bottom Output pane, the message indicates successful execution.

16. Cambie los nombres y apellidos de todas las personas a mayúsculas



The screenshot shows the SQL Workbench/J interface. In the top-left pane, the database structure is visible, including the `personas` table under the `public` schema. In the main SQL editor pane, the following SELECT statement is executed:

```
1 select cedula, upper(nombres), upper(apellido)
2 from persona
3
```

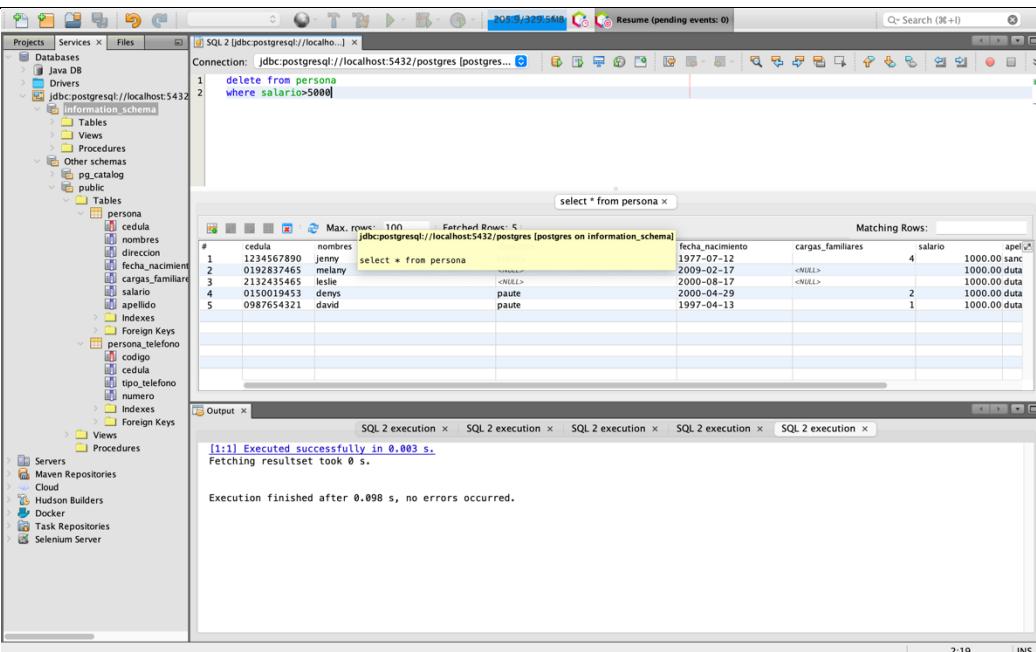
The results of the query are displayed in a table below:

#	Cedula	upper(nombres)	upper(apellido)
1	1234567890	UPPER	SANCHEZ
2	0192837465	MELANY	DUTAN
3	2132435465	LESIE	DUTAN
4	0150019453	DENYS	DUTAN
5	0987654321	DAVID	DUTAN

In the bottom Output pane, the message indicates successful execution.

17. Borre los registros de las personas cuyo salario sea mayor a 5000 USD.

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

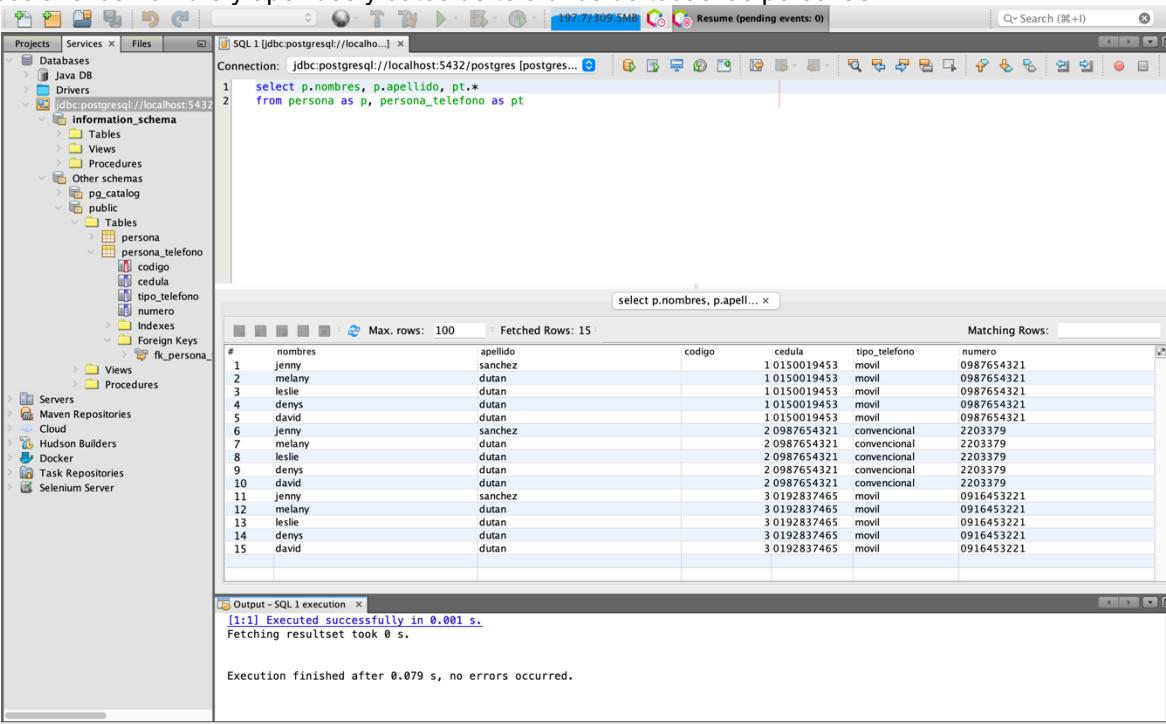


```
1 delete from persona
2 where salario>5000
```

#	cedula	nombres	apellido	fecha_nacimiento	cargas_familiares	salario
1	1234567890	jenny	sanchez	1977-07-12	4	1000.00
2	0192837465	melany	dutan	2009-02-17	<NULL>	1000.00
3	2132435465	leslie	dutan	2000-08-17	<NULL>	1000.00
4	0150019453	denys	paupe	2000-04-29	2	1000.00
5	0987654321	david	paupe	1997-04-13		1

18. 2.4 FOREIGN KEY

Seleccione los nombre y apellidos y datos de teléfonos de todas las personas.

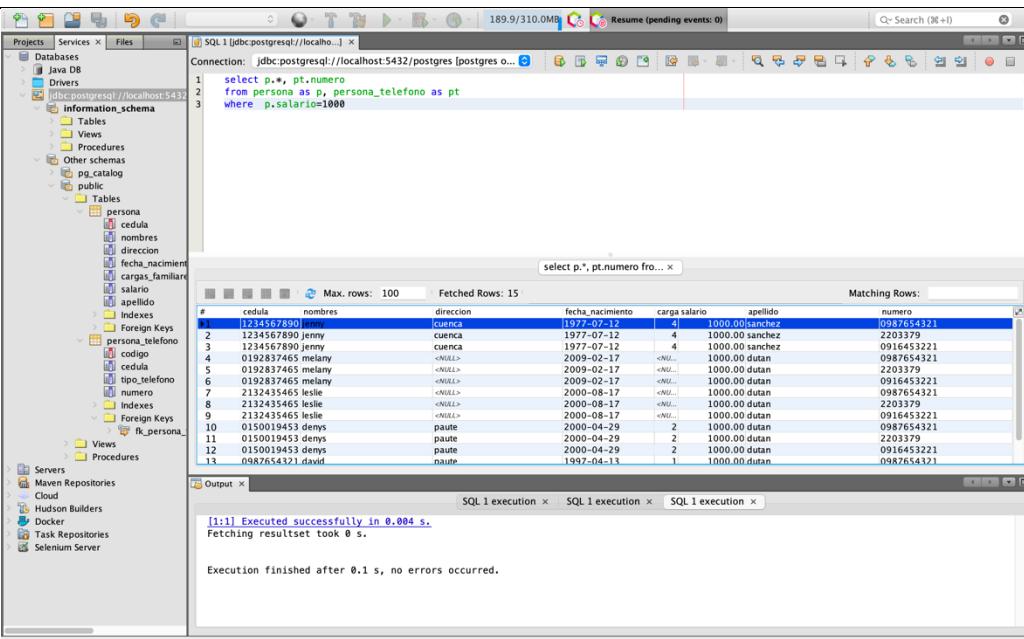


```
1 select p.nombres, p.apellido, pt.*
2 from persona as p, persona_telefono as pt
```

#	nombres	apellido	codigo	cedula	tipo_telefono	numero
1	jenny	sanchez	10150019453	movil	0987654321	
2	melany	dutan	10150019453	movil	0987654321	
3	leslie	dutan	10150019453	movil	0987654321	
4	denys	dutan	10150019453	movil	0987654321	
5	david	dutan	10150019453	movil	0987654321	
6	jenny	sanchez	20987654321	convenional	2203379	
7	melany	dutan	20987654321	convenional	2203379	
8	leslie	dutan	20987654321	convenional	2203379	
9	denys	dutan	20987654321	convenional	2203379	
10	david	dutan	20987654321	convenional	2203379	
11	jenny	sanchez	30192837465	movil	0916453221	
12	melany	dutan	30192837465	movil	0916453221	
13	leslie	dutan	30192837465	movil	0916453221	
14	denys	dutan	30192837465	movil	0916453221	
15	david	dutan	30192837465	movil	0916453221	

19. Seleccione los datos de las personas y sus números de teléfonos de aquellas personas que tiene un salario de 1000 USD.

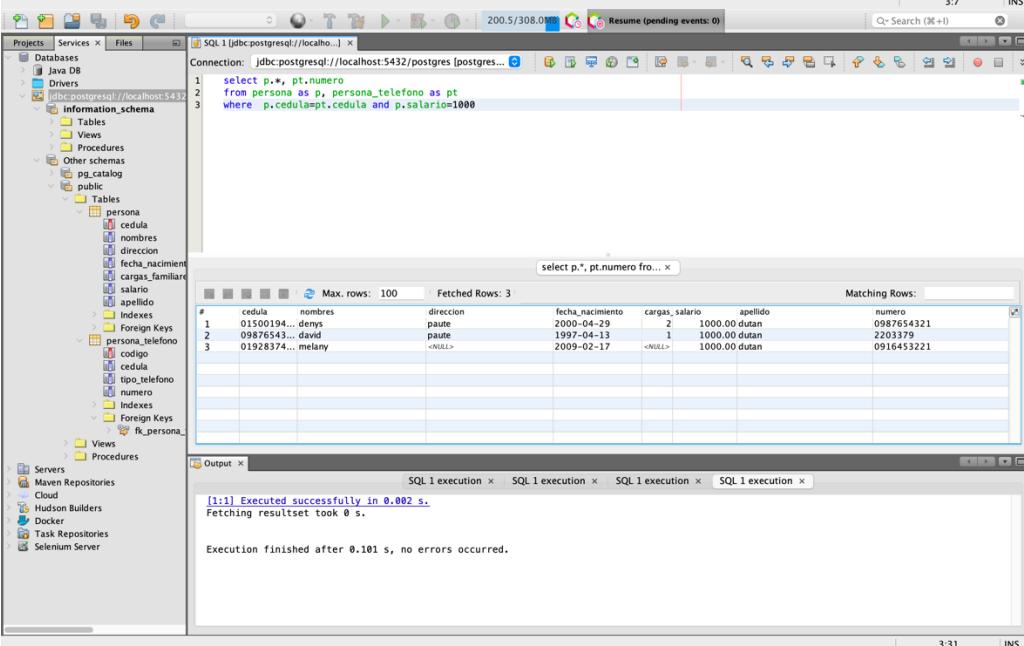
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



```
1 select p.*, pt.numero
2   from persona as p, persona_telefono as pt
3  where p.salario=1000
```

#	cedula	nombr... es	direccion	fecha_nacimiento	carga_salar... io	apellido	numero
1	1234567890	jenny	cuenca	1977-07-12	4	1000.00	sanchez
2	1234567890	jenny	cuenca	1977-07-12	4	1000.00	sanchez
3	0150019453	denys	<NULL>	2000-04-29	<NULL>	1000.00	dutan
5	0192837465	melany	<NULL>	2009-02-17	<NULL>	1000.00	dutan
6	0192837465	melany	<NULL>	2009-02-17	<NULL>	1000.00	dutan
7	2132435465	leslie	<NULL>	2000-08-17	<NULL>	1000.00	dutan
8	2132435465	leslie	<NULL>	2000-08-17	<NULL>	1000.00	dutan
9	0150019453	denys	pante	2000-04-29	2	1000.00	dutan
10	0150019453	denys	pante	2000-04-29	2	1000.00	dutan
11	0150019453	denys	pante	2000-04-29	2	1000.00	dutan
12	0987654321	david	pante	1997-04-13	1	1000.00	dutan
13	0987654321	david	pante	1997-04-13	1	1000.00	dutan

Execution finished after 0.1 s, no errors occurred.



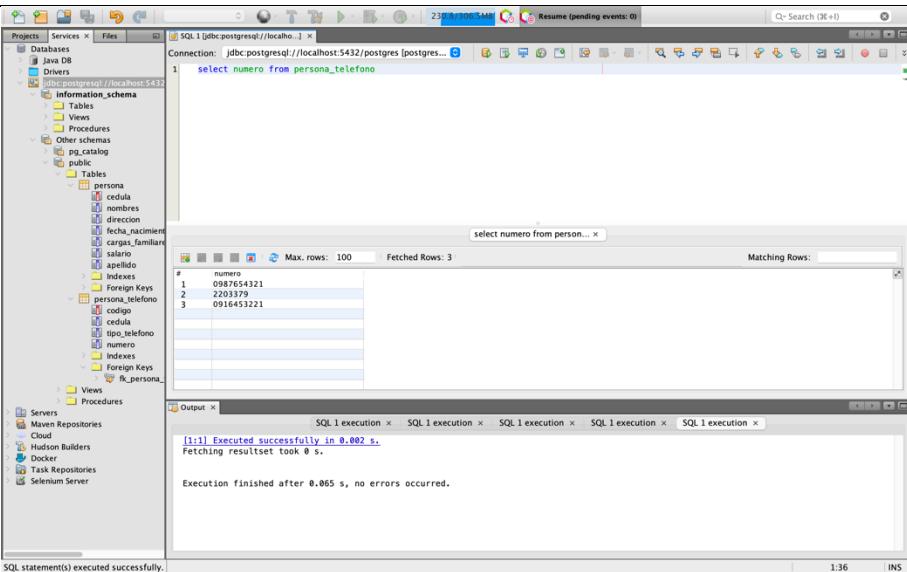
```
1 select p.*, pt.numero
2   from persona as p, persona_telefono as pt
3  where p.cedula like '0150019453%' and p.salario=1000
```

#	cedula	nombr... es	direccion	fecha_nacimiento	cargas_salar... io	apellido	numero
1	0150019453	denys	pante	2000-04-29	2	1000.00	dutan
2	0987654321	david	pante	1997-04-13	1	1000.00	dutan
3	0192837465	melany	<NULL>	2009-02-17	<NULL>	1000.00	dutan

Execution finished after 0.101 s, no errors occurred.

20. Seleccione los números de teléfonos que existen en su base de datos.

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

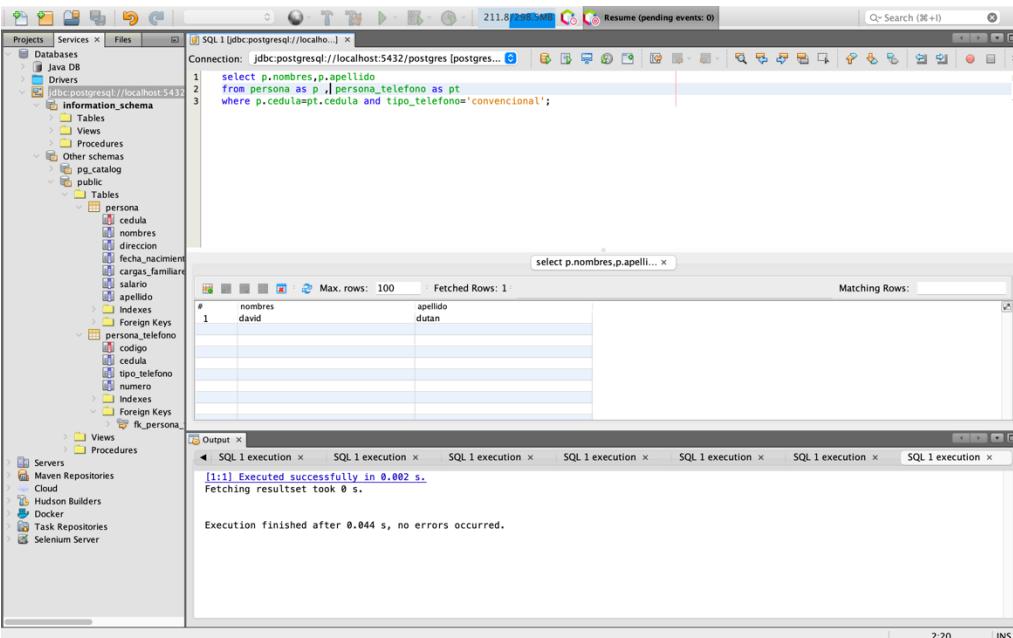


```
select numero from persona_telefono
```

#	numero
1	0987654321
2	2203379
3	0916453221

Execution finished after 0.065 s, no errors occurred.

21. Seleccione el nombre y apellido de las personas que tiene como teléfono un número convencional.



```
select p.nombres,p.apellido
from persona as p
inner join persona_telefono as pt
on p.cedula=pt.cedula
where pt.tipo_telefono='convencional';
```

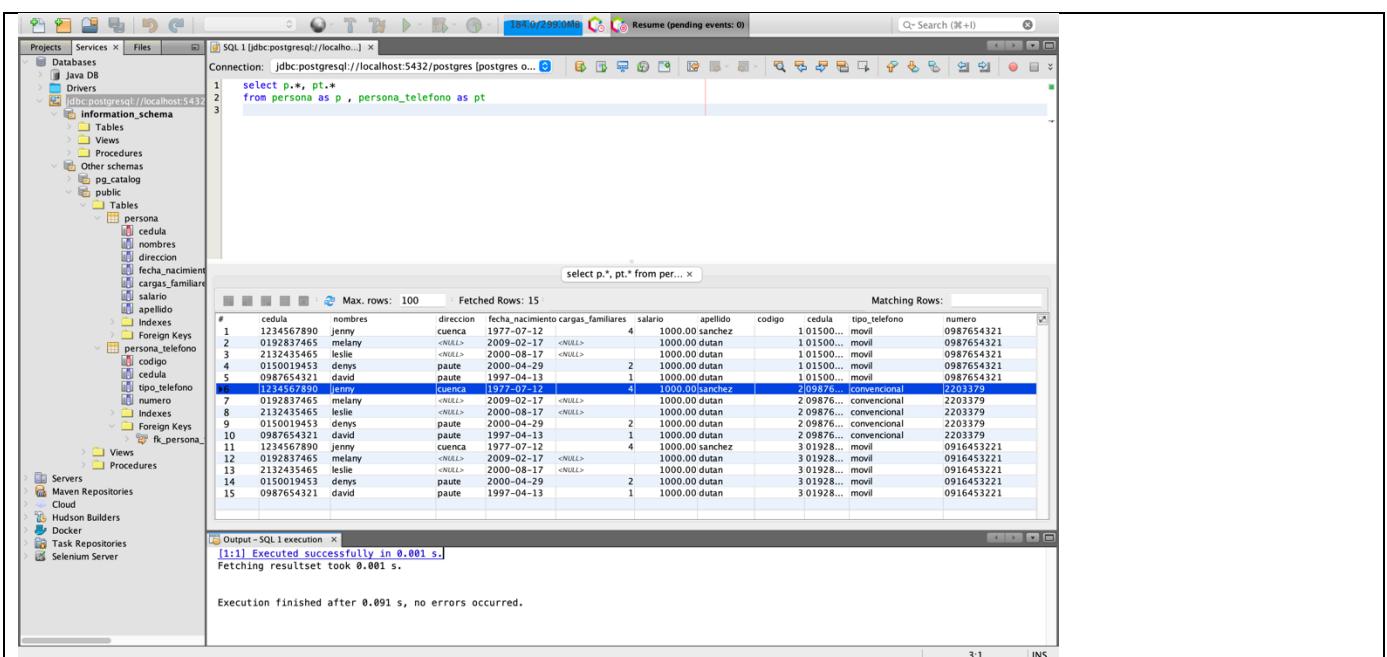
#	nombres	apellido
1	david	dutan

Execution finished after 0.044 s, no errors occurred.

22.

RESULTADO(S) OBTENIDO(S):

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



#	cedula	nombres	direccion	fecha_nacimiento	cargas_familiares	salario	apellido	codigo	cedula	tipo_telefono	numero
1	1234567890	jenny	Cuenca	1977-07-12	4	1000.00	sanchez	101500...	movil	0987654321	
2	0192837465	melany	<NULL>	2009-02-17	<NULL>	1000.00	dutan	101500...	movil	0987654321	
3	2132435465	leslie	<NULL>	2000-08-17	<NULL>	1000.00	dutan	101500...	movil	0987654321	
4	0150019453	denys	paupe	2000-04-29	2	1000.00	dutan	101500...	movil	0987654321	
5	0987654321	david	paupe	1997-04-13	1	1000.00	dutan	101500...	movil	0987654321	
6	1234567890	jenny	Cuenca	1977-07-12	4	1000.00	sanchez	209876...	convenional	2203379	
7	0192837465	melany	<NULL>	2009-02-17	<NULL>	1000.00	dutan	209876...	convenional	2203379	
8	2132435465	leslie	<NULL>	2000-08-17	<NULL>	1000.00	dutan	209876...	convenional	2203379	
9	0150019453	denys	paupe	2000-04-29	2	1000.00	dutan	209876...	convenional	2203379	
10	0987654321	david	paupe	1997-04-13	1	1000.00	dutan	209876...	convenional	2203379	
11	1234567890	jenny	Cuenca	1977-07-12	4	1000.00	sanchez	301928...	movil	0916453221	
12	0192837465	melany	<NULL>	2009-02-17	<NULL>	1000.00	dutan	301928...	movil	0916453221	
13	2132435465	leslie	<NULL>	2000-08-17	<NULL>	1000.00	dutan	301928...	movil	0916453221	
14	0150019453	denys	paupe	2000-04-29	2	1000.00	dutan	301928...	movil	0916453221	
15	0987654321	david	paupe	1997-04-13	1	1000.00	dutan	301928...	movil	0916453221	

CONCLUSIONES: En conclusión, se debe tener un entendimiento de como crear tablas con comandos de la base de datos de PostgreSQL. Al realizar esta practica con PostgreSQL también se debe haber entendido como insertar, seleccionar, editar, y borrar elementos de una tabla por medio de los comandos de PostgreSQL. Por ultimo se debe comprender como esta herramienta de base de datos acelera el proceso de archivar datos sin miedo a perder información.

RECOMENDACIONES:

Nombre de estudiante: _____ Denys Dutan _____

Firma de estudiante: 