
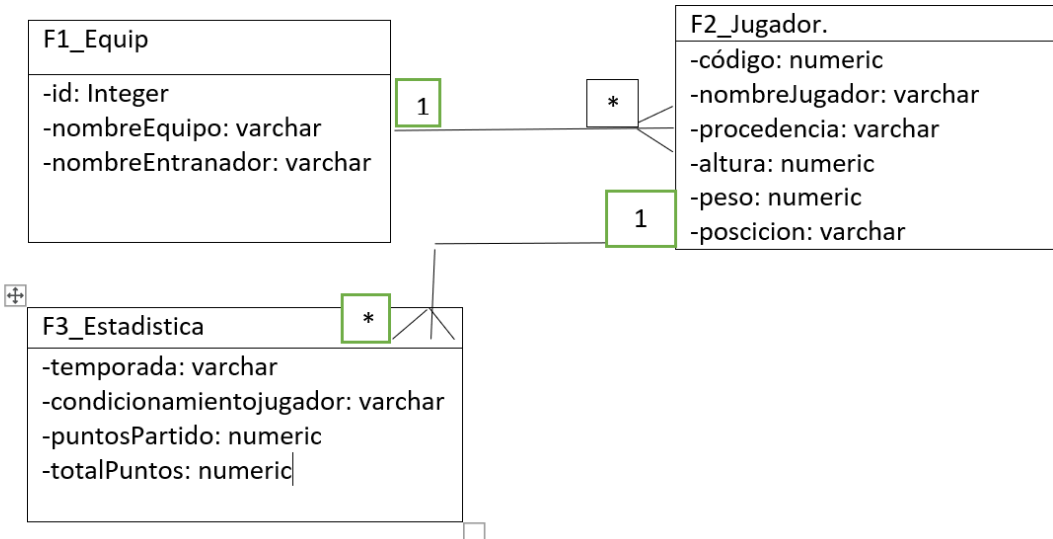
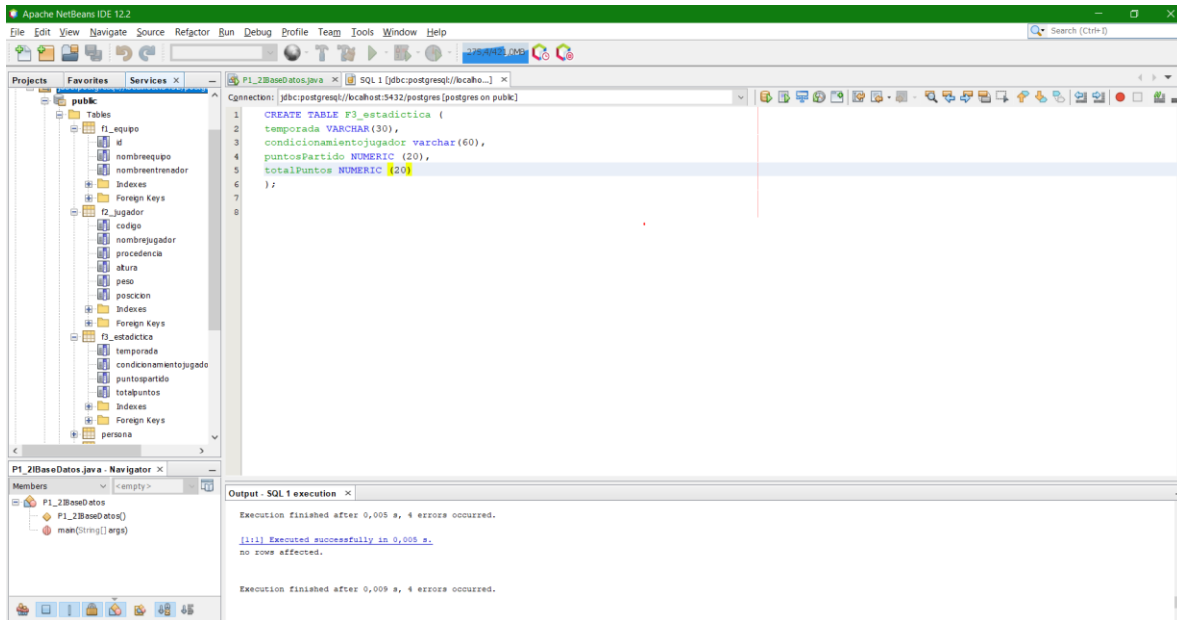
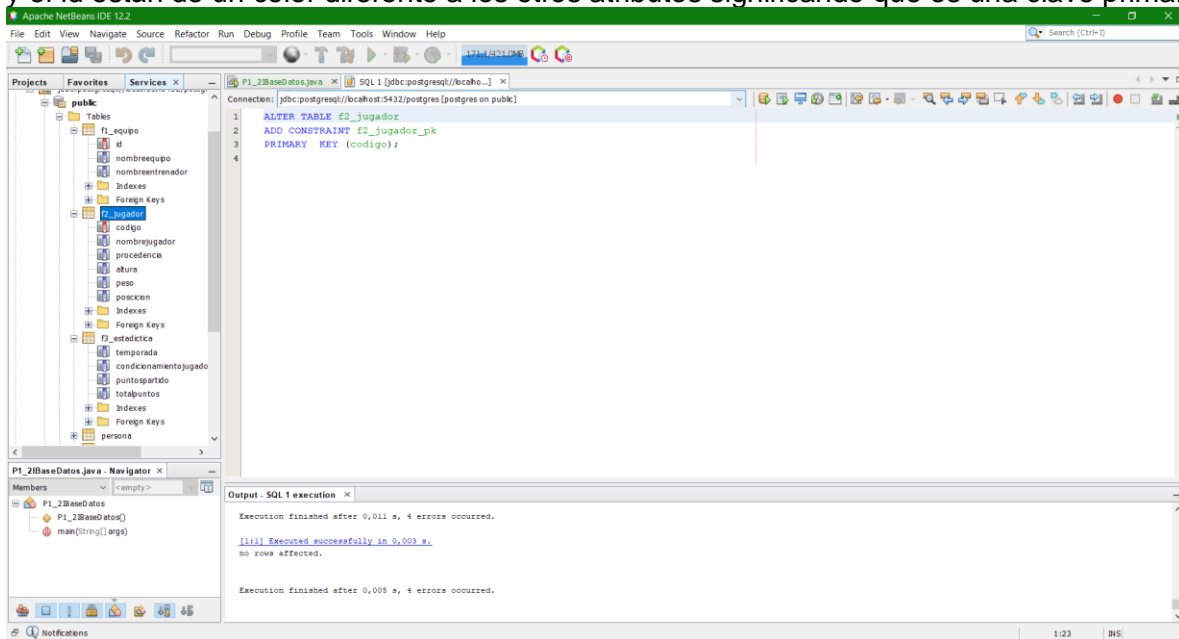
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

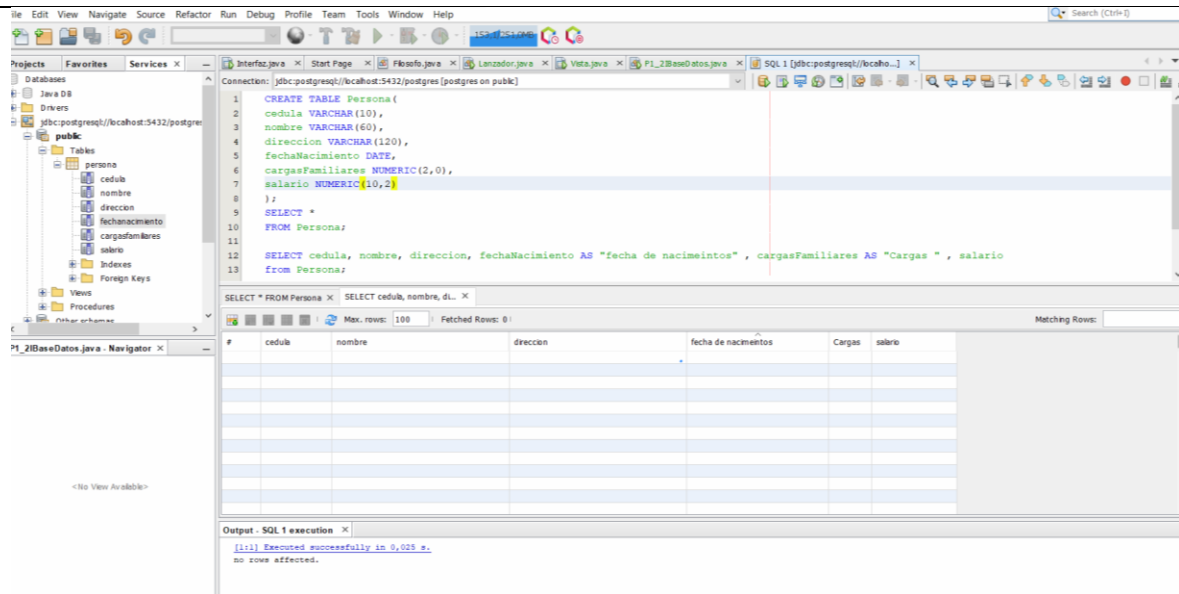
		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: COMPUTACIÓN		ASIGNATURA: Programación Aplicada	
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Base de Datos	
OBJETIVO: Identificar las sentencias SQL Diseñar e Implementar codigos DDL, DML Entender cada una de las características del uso de una Base de Datos			
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):	1. Revisar los conceptos fundamentales de SQL		
	2. Establecer como implementar DDL y DML		
	3. Implementar y diseñar nuevas estructuras de base de datos y datos.		
	4. Realizar el informe respectivo según los datos solicitados.		
ACTIVIDADES POR DESARROLLAR (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1. Revisar la teoría y conceptos de Base de Datos			
2. Diseñar e implementar las siguientes prácticas de Base de datos solamente utilizando código SQL (No se permite el uso del generador o ayuda de ningún IDE): <ul style="list-style-type: none"> 2.1 Actividad: Realizar una base de datos para el equipo de futbol del curso. Desarrollo del diagrama uml para posteriormente poder crear las tablas en el editor de SQL 			
 <pre> classDiagram class F1_Equip { -id: Integer -nombreEquipo: varchar -nombreEntrador: varchar } class F2_Jugador { -código: numeric -nombreJugador: varchar -procedencia: varchar -altura: numeric -peso: numeric -poscicion: varchar } class F3_Estadistica { -temporada: varchar -condicionamientojugador: varchar -puntosPartido: numeric -totalPuntos: numeric } F1_Equip "1" -- "*" F2_Jugador F1_Equip "1" -- "*" F3_Estadistica F2_Jugador "1" -- "*" F3_Estadistica </pre>			
Codificación SQL. En la siguiente imagen se procedió a la codificación de algoritmos que crean tablas de acuerdo al diagrama uml planteado posteriormente.			



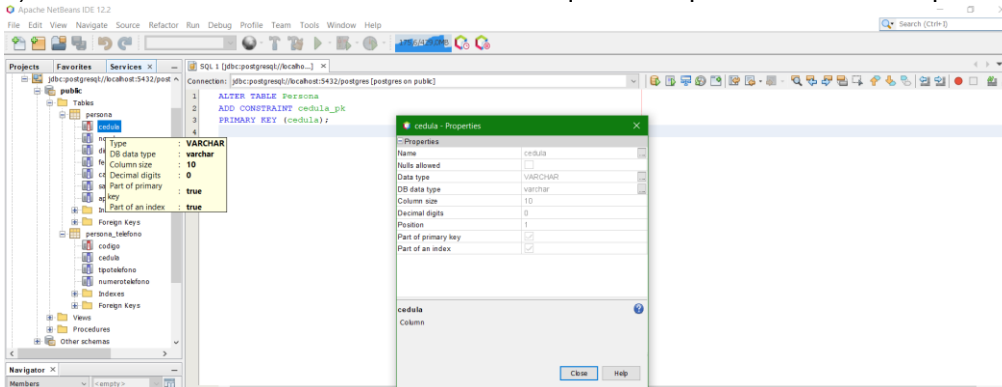
En este caso se procede a evidenciar las tablas creadas junto con el Código para tener llaves primarias en algunas de las tablas antes descritas y eso se corrobora observando que el Código y el id están de un color diferente a los otros atributos significando que es una clave primaria.



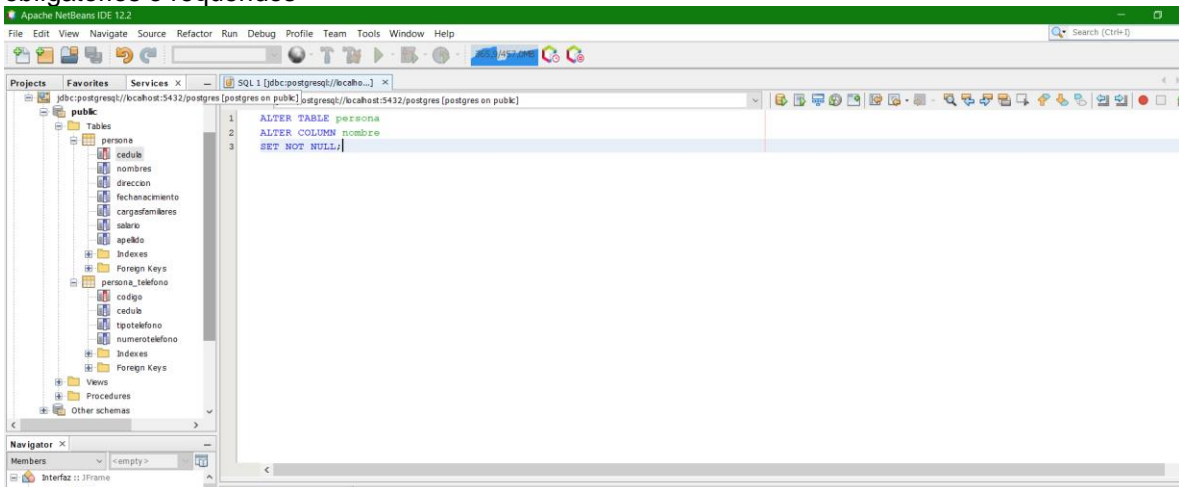
- 2.2 Crear las tabla PERSONA, con los siguientes datos:
 - Cedula con una dimensión de 10 caracteres
 - Nombre con una dimensión de 60 caracteres
 - Dirección con una dimensión de 120 caracteres
 - Fecha de nacimiento de tipo fecha
 - Número de cargas familiares
 - Salario de tipo real con 10 dígitos enteros y dos decimales

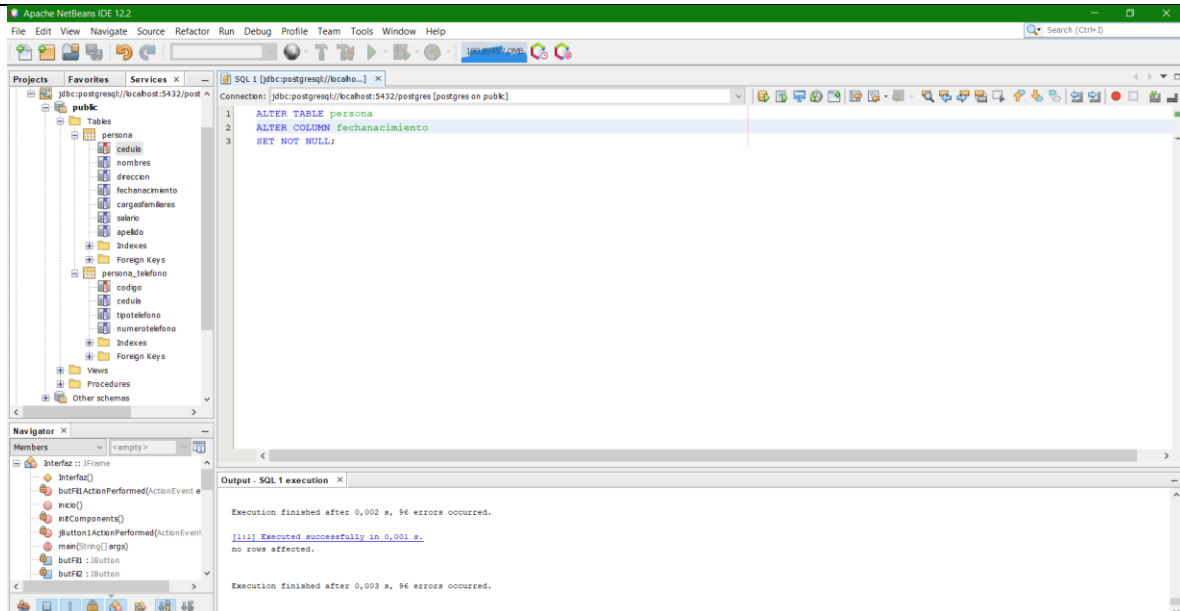


2) Modificar la tabla PERSONA de tal manera que el campo cedula sea clave primaria.



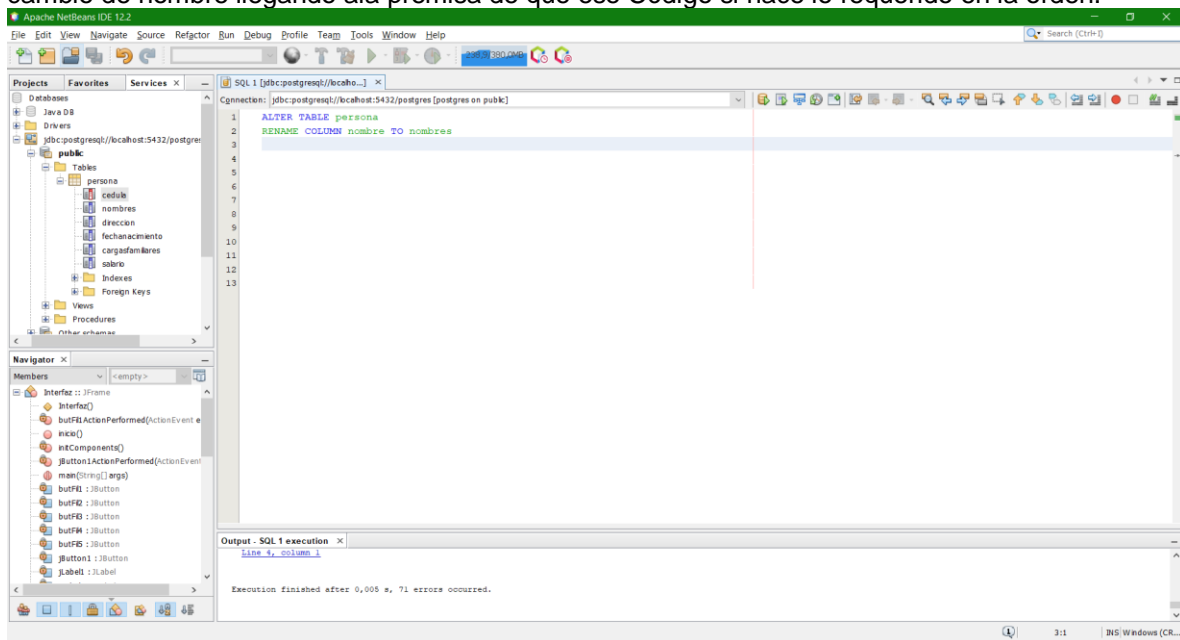
3) Modificar la tabla PERSONA de tal manera que los campos nombre, fecha de nacimiento sean obligatorios o requeridos



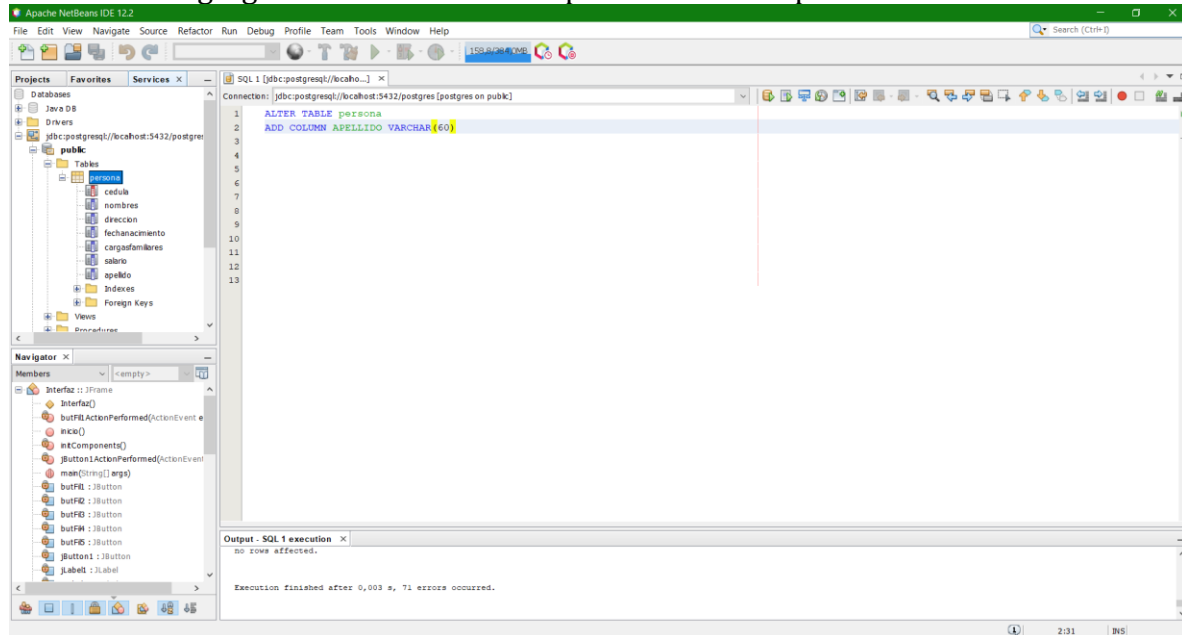


4) Modificar la tabla PERSONA de tal manera la columna nombre cambie a nombres, y se añade una columna para el apellido con un dimensión de 60 caracteres.

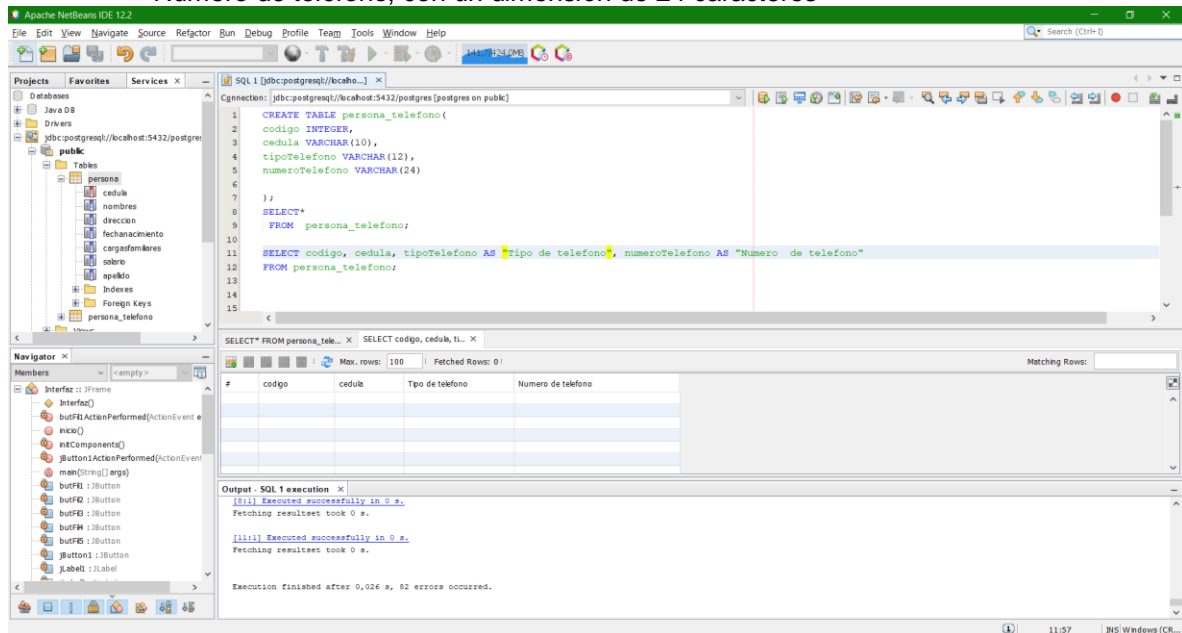
Como se podrá observar en la tabla hay que regresarle luego de codificar y ejecutar los códigos para el cambio de nombre llegando ala premisa de que ese Código si hace lo requerido en la orden.



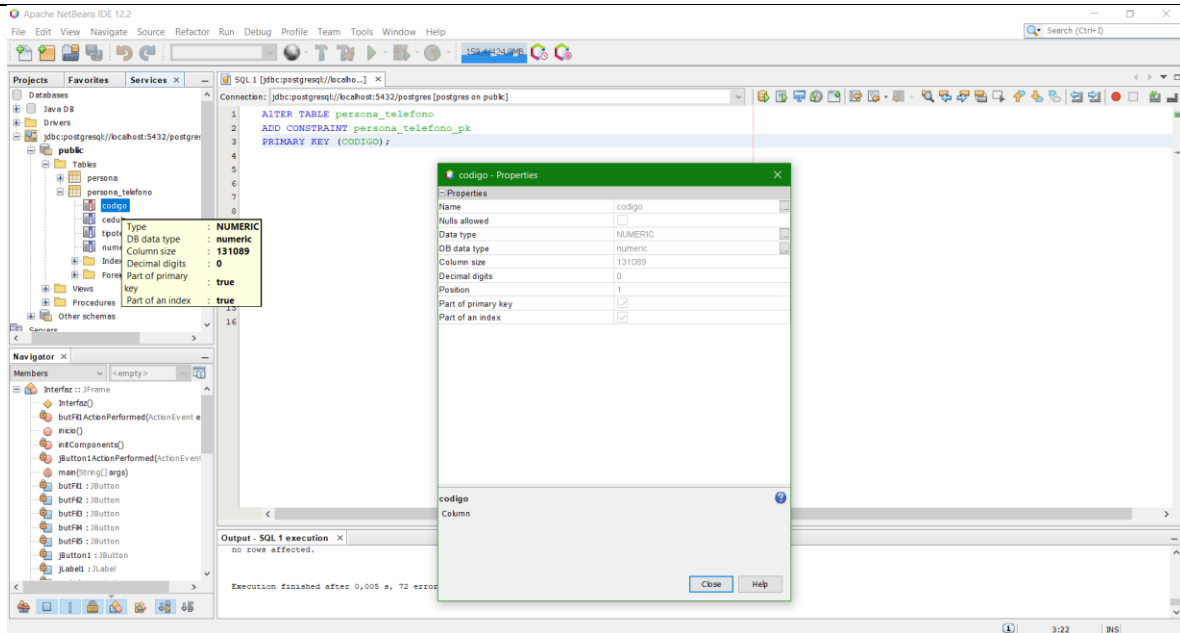
En este caso agregaremos una columna apellido a la tabla persona



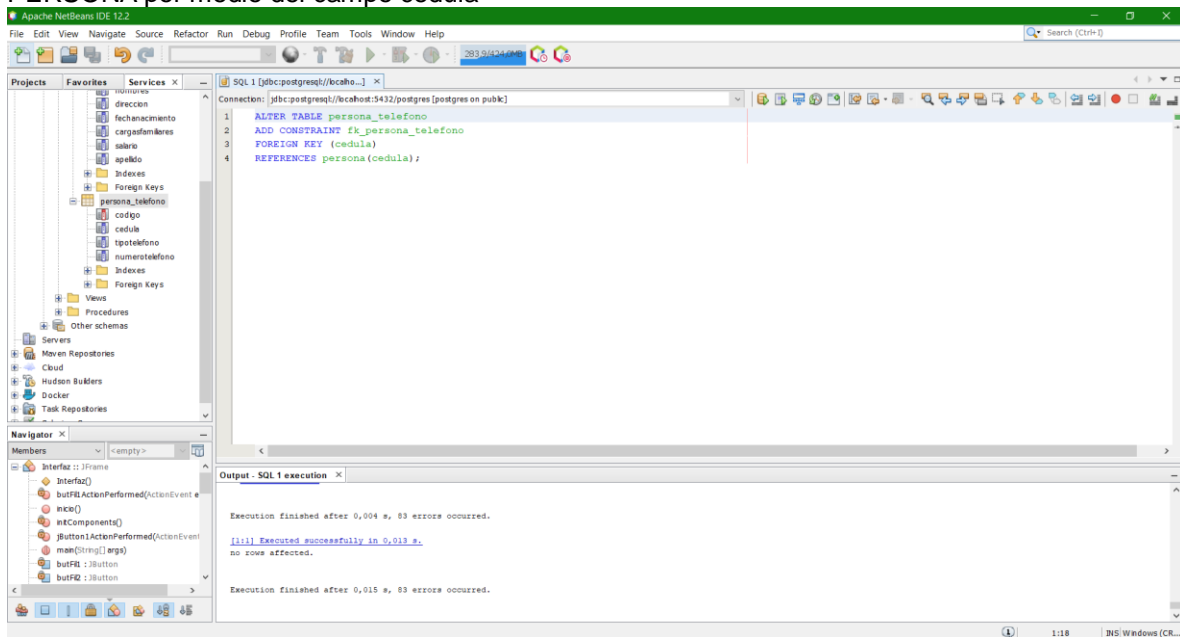
- Crear las tablas PERSONA_TELEFONO, con los siguientes datos:
 - Código, de tipo entero
 - Cedula, con una dimensión de 10 caracteres (es la referencia de la tabla persona)
 - Tipo de teléfono, con un dimensión de 12 caracteres, aquí se guardarán valores como celular o convencional
 - Número de teléfono, con un dimensión de 24 caracteres



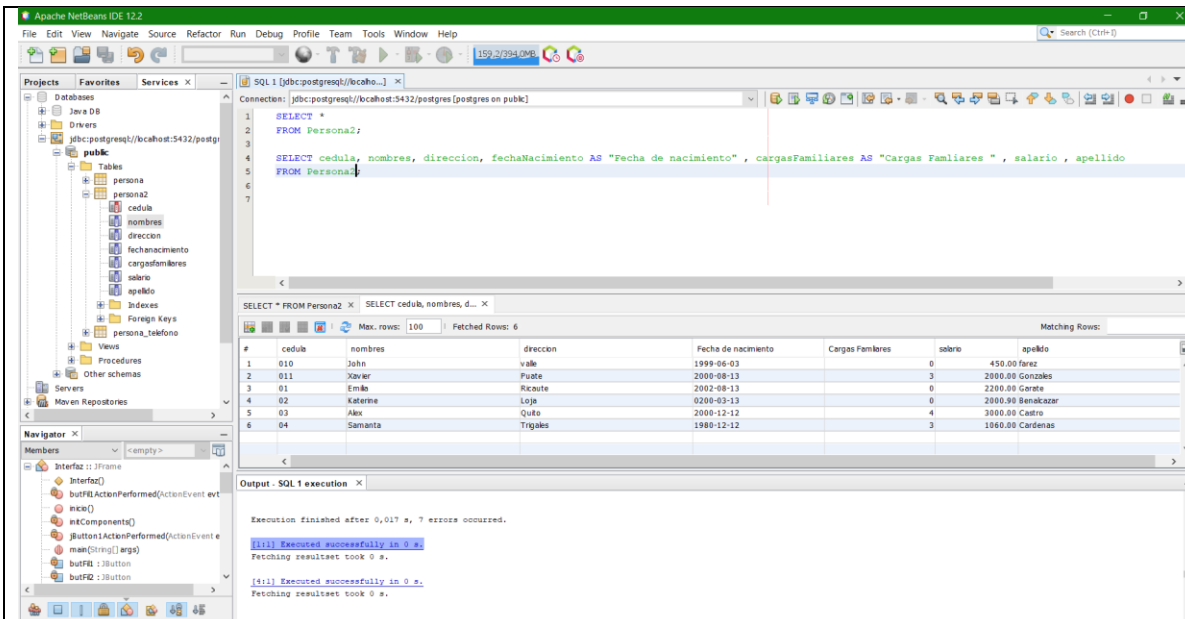
6) Modificar la tabla PERSONA_TELEFONO de tal manera que el campo código sea clave primaria.



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



2.3 SELECT de todos los registros de personas Persona creadas.



SQL 1 [jdbc:postgresql://localhost:5432/postgres [postgres on public]]

```

1 SELECT *
2 FROM persona2;
3
4 SELECT cedula, nombres, direccion, fechaNacimiento AS "Fecha de nacimiento", cargasFamiliares AS "Cargas Familiares", salario, apellido
5 FROM persona2;
6
7

```

#	cedula	nombres	direccion	Fecha de nacimiento	Cargas Familiares	salario	apellido
1	010	John	vale	1999-06-03	0	430.00	faraz
2	011	Xavier	Paste	2000-08-13	3	2000.00	Gonzales
3	01	Emile	Ricarte	2002-08-13	0	2200.00	Garate
4	02	Katerine	Loja	0200-03-13	0	2000.00	Benckazar
5	03	Alex	Quito	2000-12-12	4	3000.00	Castro
6	04	Samanta	Trigales	1980-12-12	3	1000.00	Cardenas

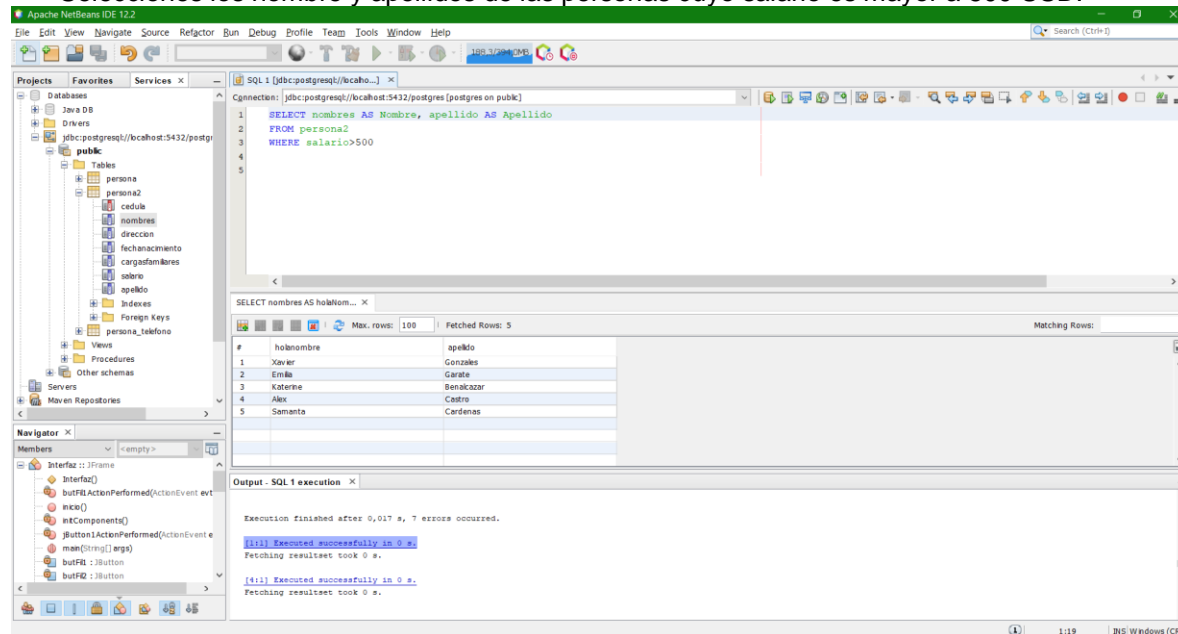
Output - SQL 1 execution

Execution finished after 0,017 s., 7 errors occurred.

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

[4:1] Executed successfully in 0 s.
Fetching resultset took 0 s.

- Selecciones los nombre y apellidos de las personas cuyo salario es mayor a 500 USD.



SQL 1 [jdbc:postgresql://localhost:5432/postgres [postgres on public]]

```

1 SELECT nombres AS Nombre, apellido AS Apellido
2 FROM persona2
3 WHERE salario>500
4
5

```

#	nombre	apellido
1	Xavier	Gonzales
2	Emile	Garate
3	Katerine	Benckazar
4	Alex	Castro
5	Samanta	Cardenas

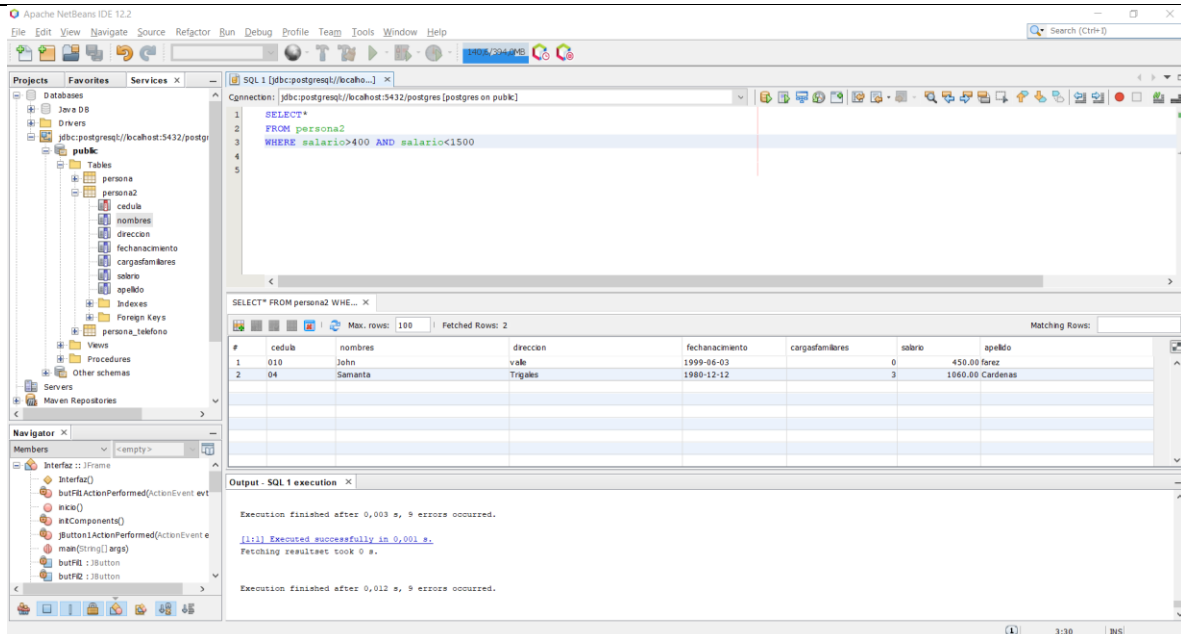
Output - SQL 1 execution

Execution finished after 0,017 s., 7 errors occurred.

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

[4:1] Executed successfully in 0 s.
Fetching resultset took 0 s.

- Selecciones los todos los datos de las personas cuyo salario es mayor a 400 USD y menos a 1500.



SQL 1 [jdbc:postgresql://localhost:5432/postgres [postgres on public]]

```

1 SELECT *
2 FROM persona2
3 WHERE salario > 400 AND salario < 1500
4
5

```

SELECT* FROM persona2 WHERE...

#	cedula	nombres	direccion	fechanacimiento	cargafamilares	salario	apellido
1	010	John	vale	1999-06-03	0	450.00	farez
2	04	Samanta	Trigales	1980-12-12	3	1060.00	Cardenas

Output - SQL 1 execution

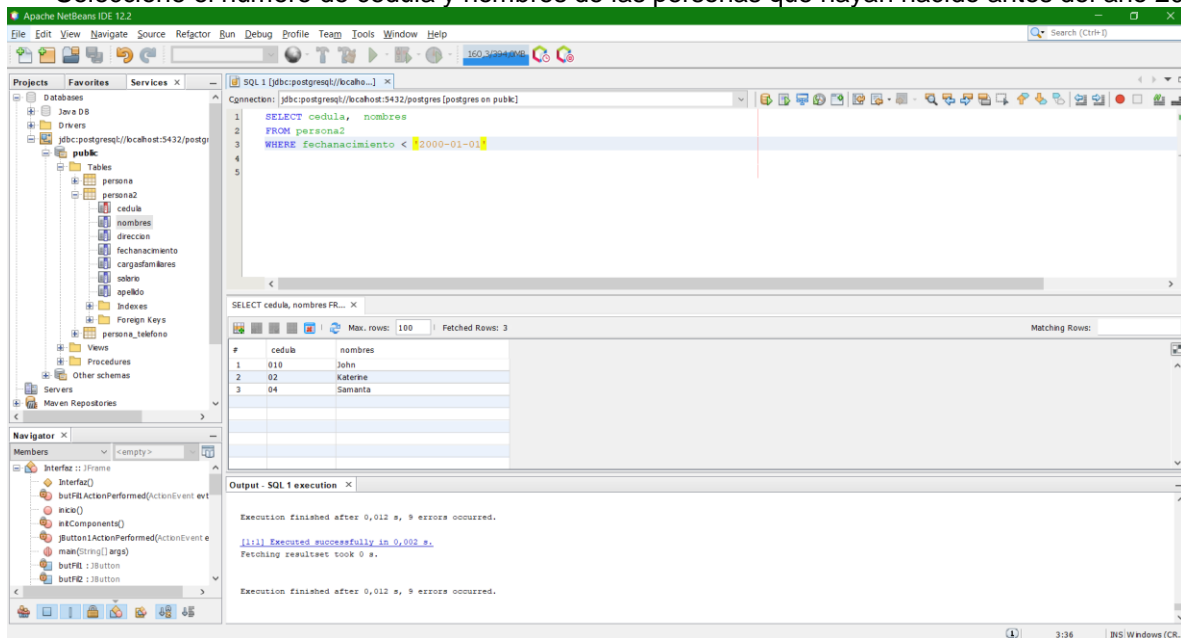
Execution finished after 0,003 s, 9 errors occurred.

[...] Executed successfully in 0,003 s.

Fetching resultset took 0 s.

Execution finished after 0,012 s, 9 errors occurred.

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



SQL 1 [jdbc:postgresql://localhost:5432/postgres [postgres on public]]

```

1 SELECT cedula, nombres
2 FROM persona2
3 WHERE fechanacimiento < '2000-01-01'
4
5

```

SELECT cedula, nombres FROM...

#	cedula	nombres
1	010	John
2	02	Katherine
3	04	Samanta

Output - SQL 1 execution

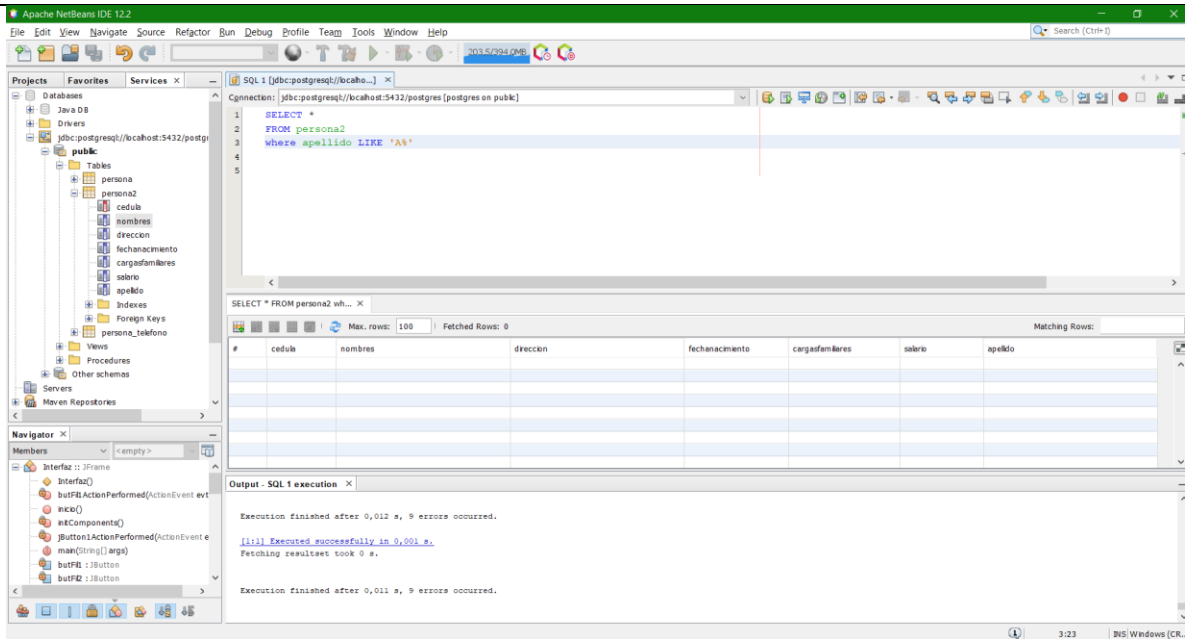
Execution finished after 0,012 s, 9 errors occurred.

[...] Executed successfully in 0,002 s.

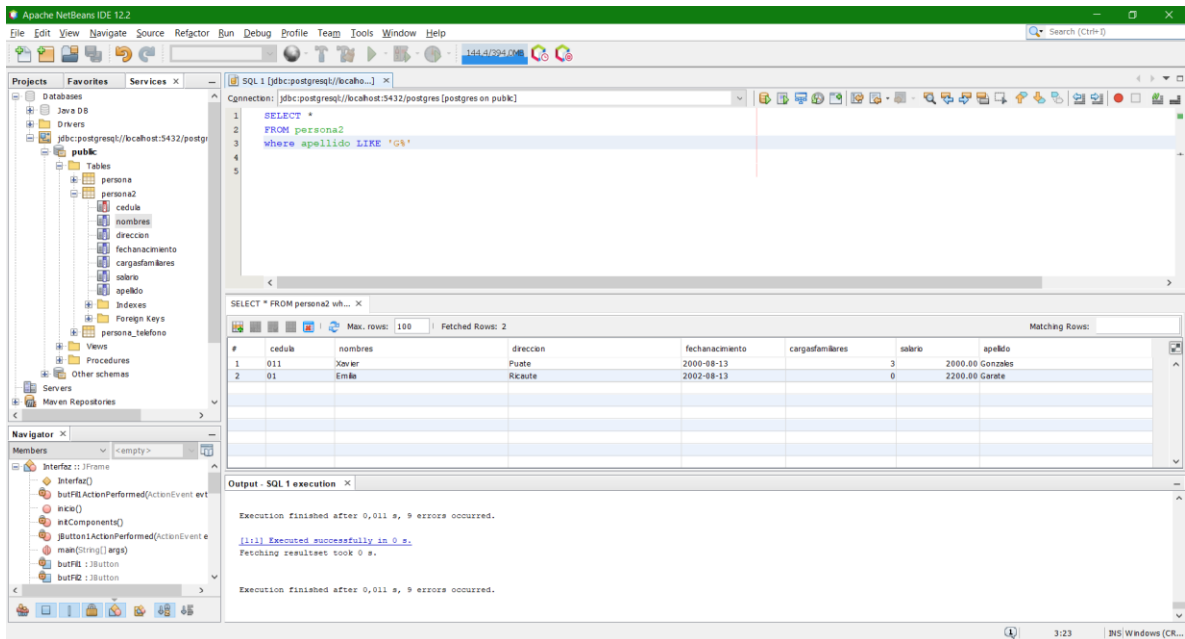
Fetching resultset took 0 s.

Execution finished after 0,012 s, 9 errors occurred.

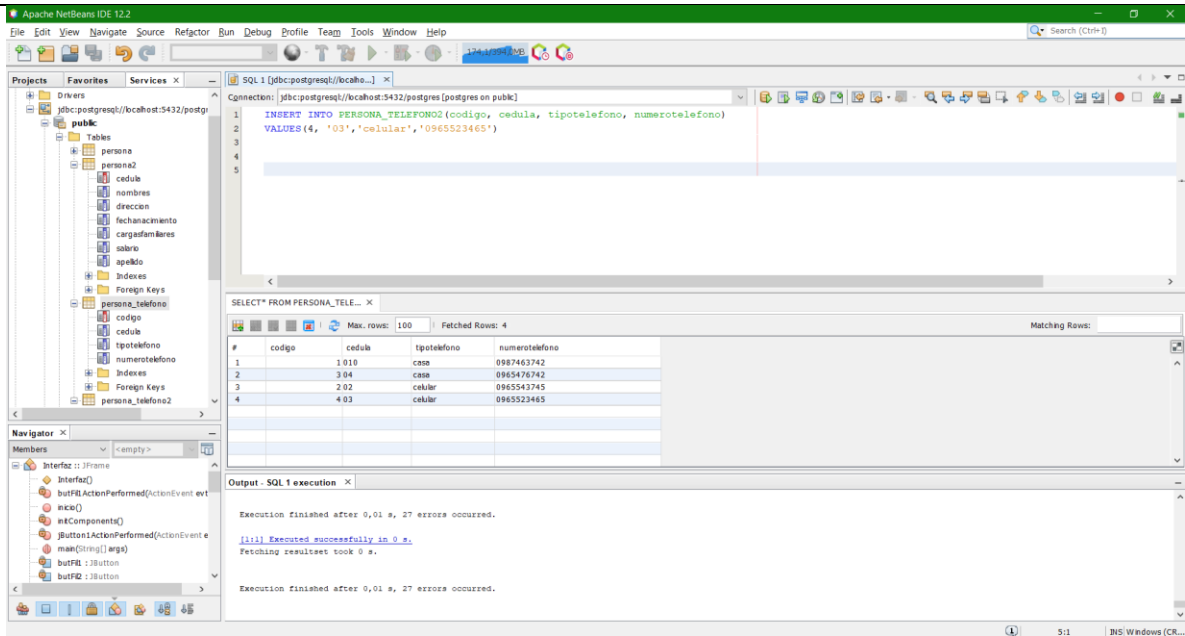
- Selecciones los datos de las personas cuyo apellido empiece con la letra "a".



En este caso se procedió a cambiar la orden pero con la evidencia de que no existe ningún apellido que empiece con la letra a por esa razón se hizo un cambio de la letra "A" a "G" obteniendo lo siguiente



- 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.



SQL 1 [jdbc:postgresql://localhost:5432/postgres [postgres on public]]

```
1 INSERT INTO PERSONA_TELEFONO2 (codigo, cedula, tipotelefono, numerotelefono)
2 VALUES (4, '03','celular','0965523465')
```

SELECT* FROM PERSONA_TELEFONO2

#	codigo	cedula	tipotelefono	numerotelefono
1	1010	caso	0987463742	
2	304	caso	0965476742	
3	202	celular	0965543745	
4	403	celular	0965523465	

Output - SQL 1 execution

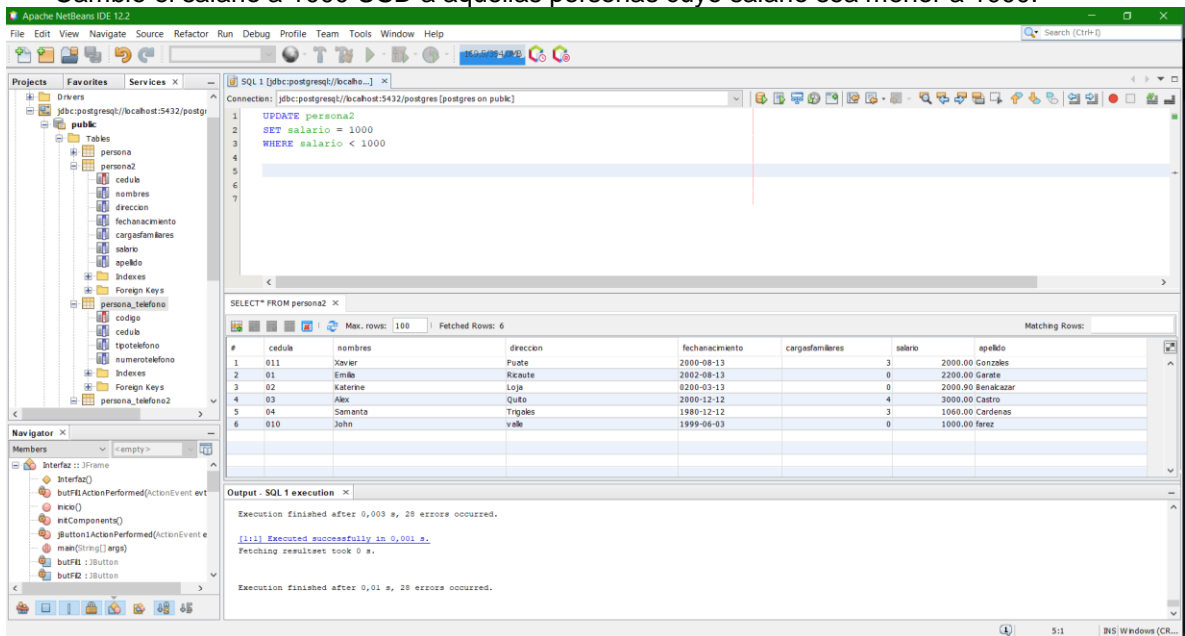
Execution finished after 0,01 s, 27 errors occurred.

[i:] Executed successfully in 0 s.

Fetching resultset took 0 s.

Execution finished after 0,01 s, 27 errors occurred.

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



SQL 1 [jdbc:postgresql://localhost:5432/postgres [postgres on public]]

```
1 UPDATE persona2
2 SET salario = 1000
3 WHERE salario < 1000
```

SELECT* FROM persona2

#	cedula	nombres	direccion	fechanacimiento	cargasfamiliares	salario	apellido
1	011	Xavier	Puente	2000-08-13	3	2000.00	Gonzalez
2	01	Emilia	Ricacha	2002-08-13	0	2200.00	Garate
3	02	Katherine	Loja	0200-03-13	0	2000.90	Benacazar
4	03	Alex	Quito	2000-12-12	4	3000.00	Castro
5	04	Samanta	Trigales	1980-12-12	3	1000.00	Cardenas
6	010	Jahn	vale	1999-06-03	0	1000.00	farez

Output - SQL 1 execution

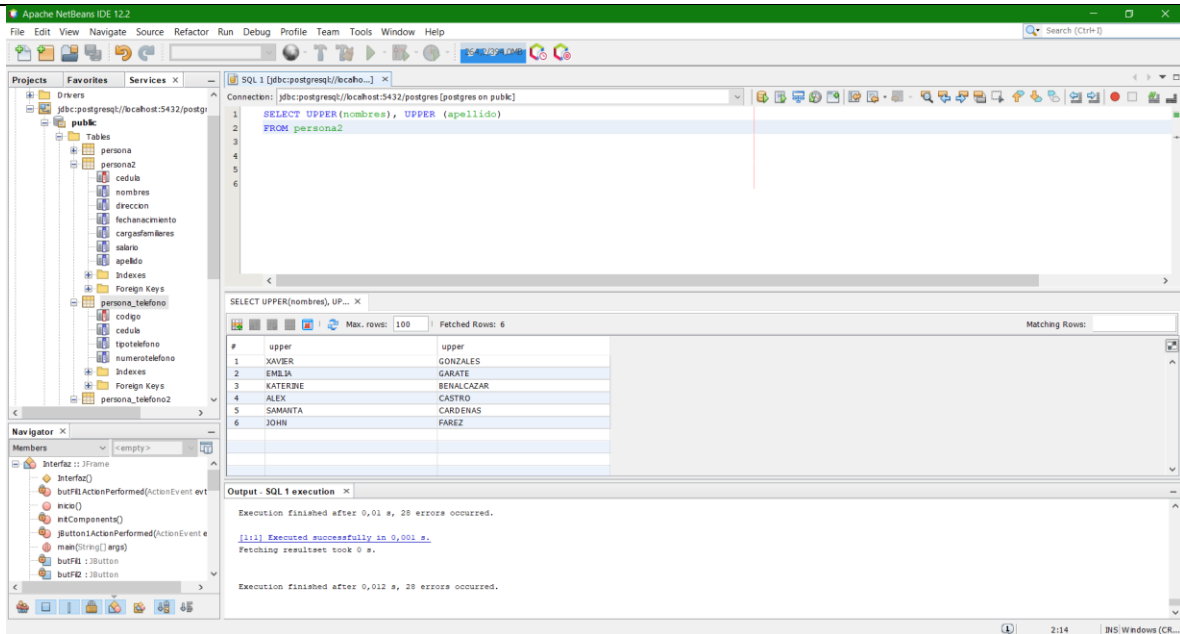
Execution finished after 0,003 s, 28 errors occurred.

[i:] Executed successfully in 0,001 s.

Fetching resultset took 0 s.

Execution finished after 0,01 s, 28 errors occurred.

- Cambie los nombres y apellidos de todas las personas a mayúsculas



SQL 1 [jdbc:postgresql://localhost:5432/postgres/postgres on public]

```
1 SELECT UPPER(nombres), UPPER(apellido)
2 FROM persona2
3
4
5
6
```

SELECT UPPER(nombres), UPPER(apellido) X

Max. rows: 100 | Fetched Rows: 6

#	upper	upper
1	XAVIER	GONZALES
2	EMILIA	GARATE
3	KATERINE	BENALCAZAR
4	ALEX	CASTRO
5	SAMANTA	CARDENAS
6	JOHN	FAREZ

Output - SQL 1 execution X

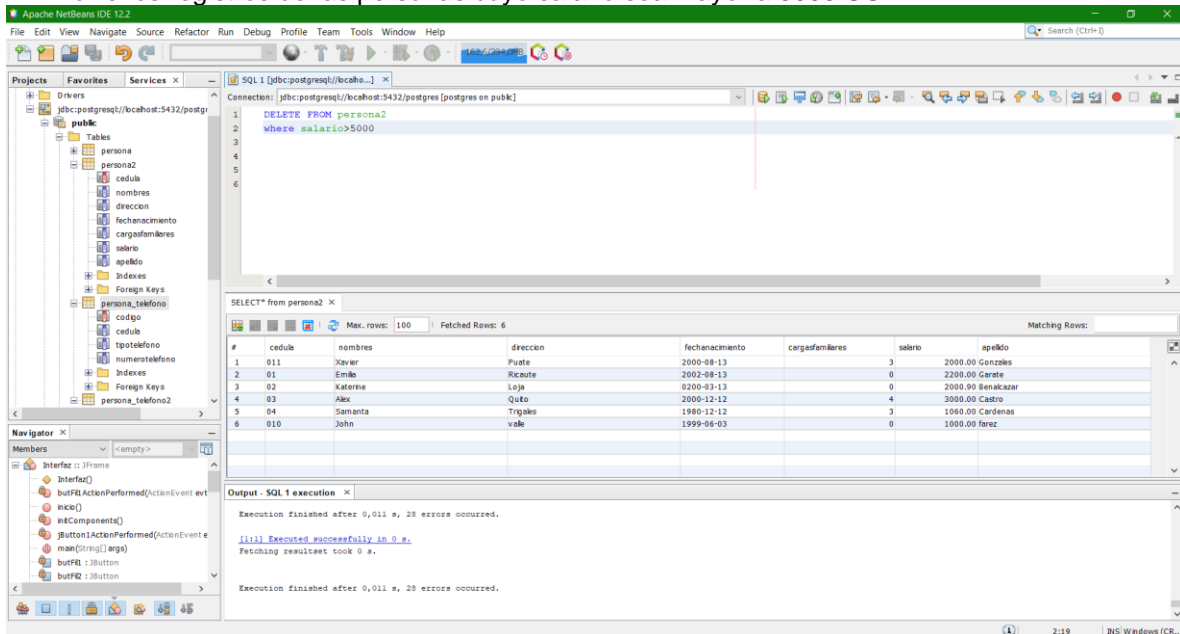
Execution finished after 0,01 s, 28 errors occurred.

[1] Executed successfully in 0,001 s.

Fetching resultset took 0 s.

Execution finished after 0,012 s, 28 errors occurred.

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



SQL 1 [jdbc:postgresql://localhost:5432/postgres/postgres on public]

```
1 DELETE FROM persona2
2 where salario > 5000
3
4
5
6
```

SELECT* from persona2 X

Max. rows: 100 | Fetched Rows: 6

#	cedula	nombres	direccion	fechanacimiento	cargosfamilares	salario	apellido
1	011	Xavier	Puate	2000-08-13	3	2000.00	Gonzales
2	011	Emilia	Rcaute	2002-08-13	0	2200.00	Garate
3	02	Katerine	Leja	0200-03-13	0	2000.90	Benalcazar
4	03	Alex	Queto	2000-12-12	4	3000.00	Castro
5	04	Samanta	Trigales	1980-12-12	3	1000.00	Cardenas
6	010	John	vale	1999-06-03	0	1000.00	farez

Output - SQL 1 execution X

Execution finished after 0,01 s, 28 errors occurred.

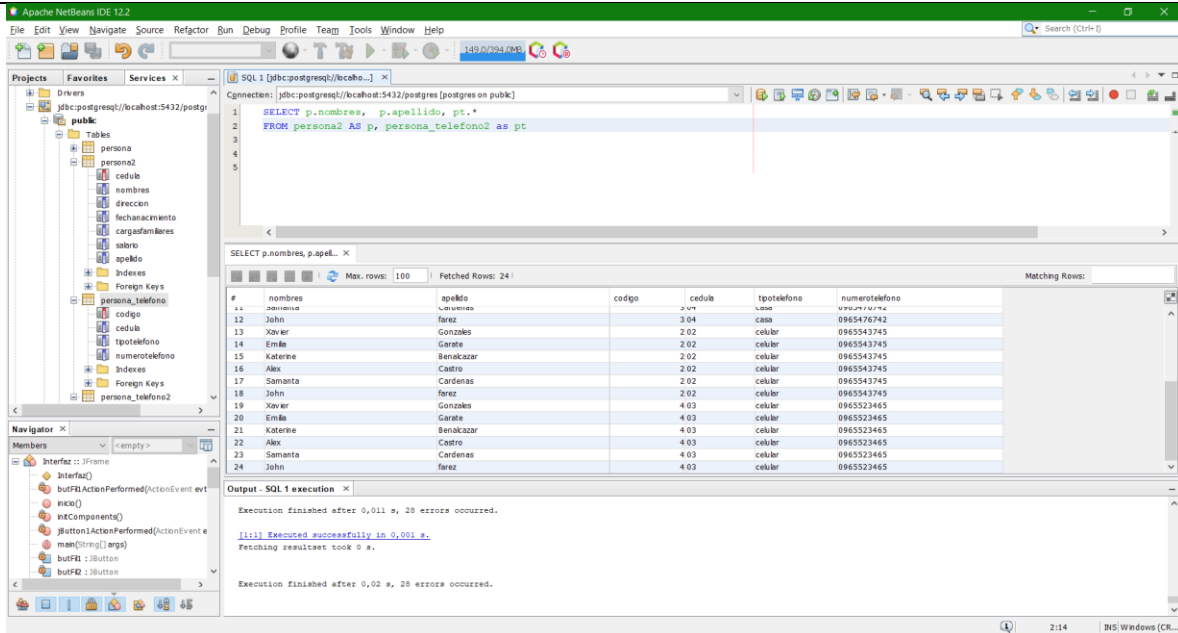
[1] Executed successfully in 0 s.

Fetching resultset took 0 s.

Execution finished after 0,011 s, 28 errors occurred.

2.4 FOREIGN KEY

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



SQL 1 [jdbc:postgresql://localhost:5432/postgres [postgres on public]

```
1 SELECT p.nombres, p.apellido, pt.*
2 FROM persona2 AS p, persona_telefono2 as pt
3
4
5
```

Max. rows: 100 | Fetched Rows: 24

#	nombres	apellido	codigo	cedula	tptelefono	numerotelefono
11	John	farez	2 04	caso	0965476742	
12	Xaver	Gonzalez	2 02	celular	0965543745	
13	Emilia	Garate	2 02	celular	0965543745	
14	Katerine	Benakazar	2 02	celular	0965543745	
15	Alex	Castro	2 02	celular	0965543745	
16	Samanta	Cardenas	2 02	celular	0965543745	
17	John	farez	2 02	celular	0965543745	
18	Xaver	Gonzalez	4 03	celular	0965523465	
19	Emilia	Garate	4 03	celular	0965523465	
20	Katerine	Benakazar	4 03	celular	0965523465	
21	Alex	Castro	4 03	celular	0965523465	
22	Samanta	Cardenas	4 03	celular	0965523465	
23	John	farez	4 03	celular	0965523465	
24						

Output - SQL 1 execution

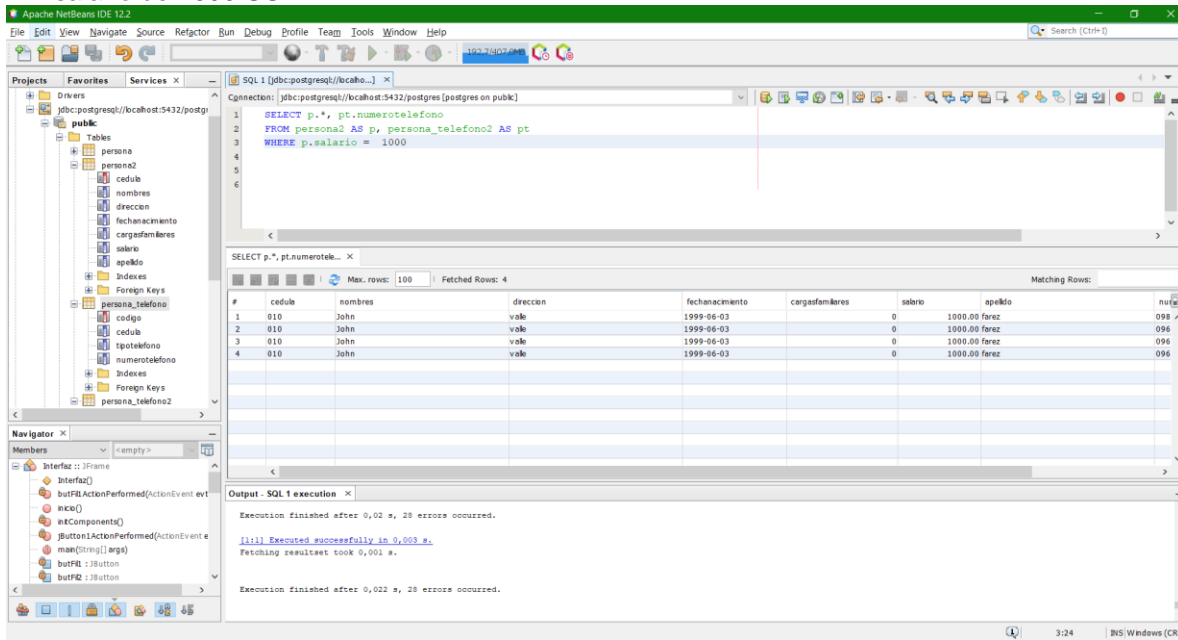
Execution finished after 0,011 s, 28 errors occurred.

||||| Executed successfully in 0,001 s.

Fetching resultset took 0 s.

Execution finished after 0,02 s, 28 errors occurred.

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



SQL 1 [jdbc:postgresql://localhost:5432/postgres [postgres on public]

```
1 SELECT p.*, pt.numerotelefono
2 FROM persona2 AS p, persona_telefono2 AS pt
3 WHERE p.salario = 1000
4
5
6
```

Max. rows: 100 | Fetched Rows: 4

#	cedula	nombres	direccion	fechacamiento	cargafamiliares	salario	apellido	numerotelefono
1	010	John	vale	1999-06-03	0	1000.00	farez	096
2	010	John	vale	1999-06-03	0	1000.00	farez	096
3	010	John	vale	1999-06-03	0	1000.00	farez	096
4	010	John	vale	1999-06-03	0	1000.00	farez	096

Output - SQL 1 execution

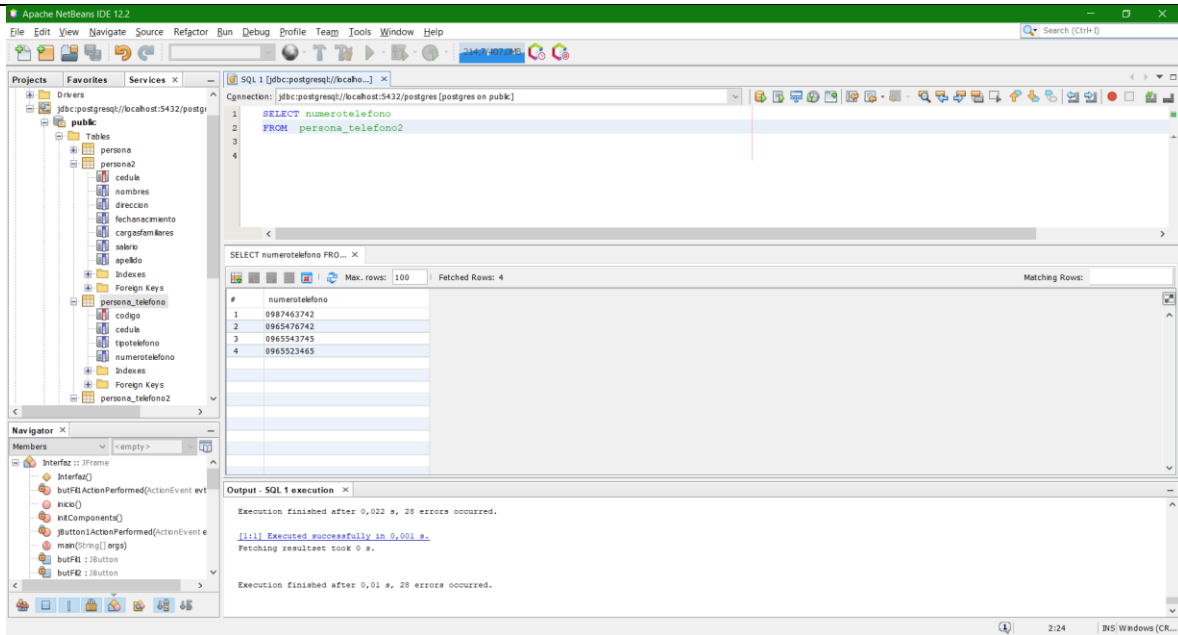
Execution finished after 0,02 s, 28 errors occurred.

||||| Executed successfully in 0,003 s.

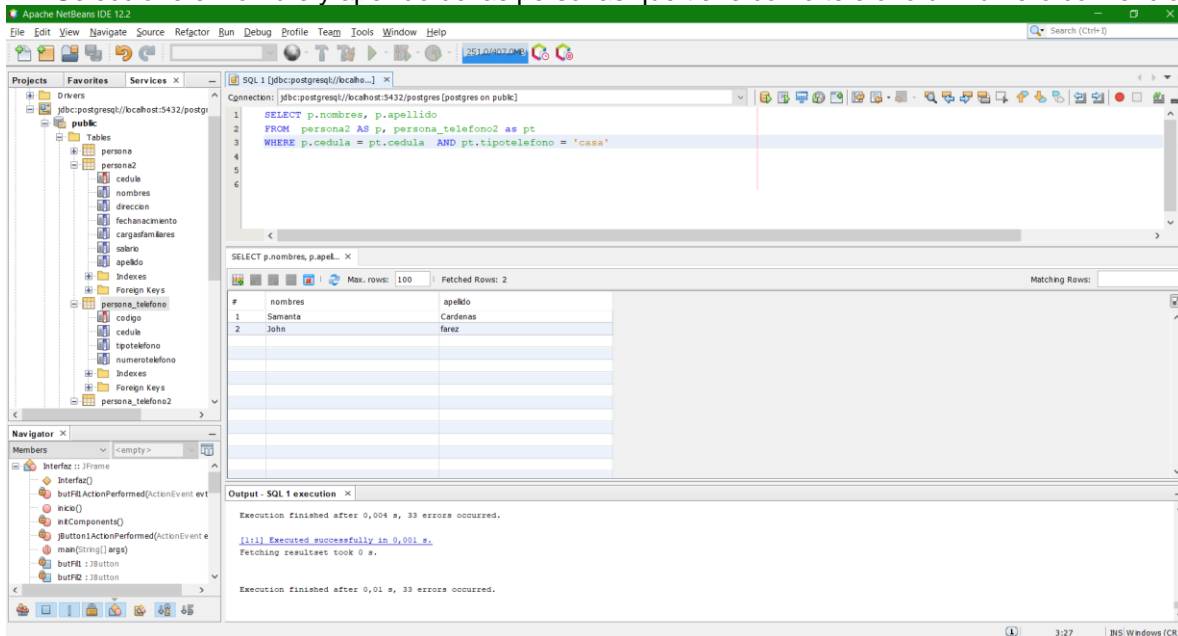
Fetching resultset took 0,001 s.

Execution finished after 0,022 s, 28 errors occurred.

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




- Seleccione el nombre y apellido de las personas que tiene como teléfono un numero convencional.



RESULTADO(S) OBTENIDO(S):

Realizar procesos de Bases de Datos

Entender las aplicaciones de codificación de las nuevas características de base de datos

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

CONCLUSIONES:

Aprenden a trabajar en grupo dentro de plazos de tiempo establecidos, manejando el lenguaje de programación de Java y SQL.

RECOMENDACIONES:

Realizar el trabajo dentro del tiempo establecido.

Nombre de estudiante: John Farez.

Firma de estudiante:

