

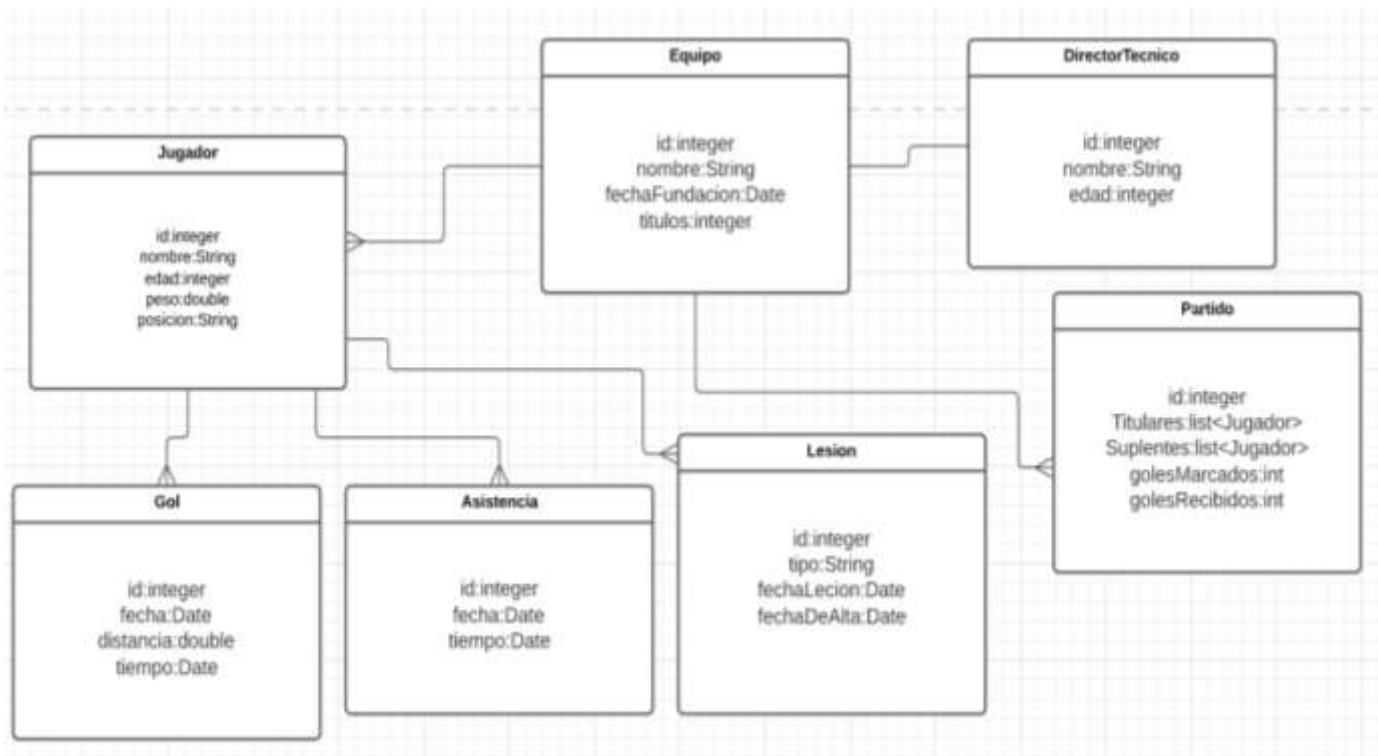
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


 <div style="float: right; text-align: right;">PRÁCTICA DE LABORATORIO</div>		
CARRERA: COMPUTACIÓN		ASIGNATURA: Programación Aplicada
NRO. PRÁCTICA:	6	TÍTULO PRÁCTICA: Base de Datos
OBJETIVO ALCANZADO: <ul style="list-style-type: none"> 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		
<p>1. Revisar la teoría y conceptos de Base de Datos</p> <p>Una Base de Datos es una serie de tablas que contienen información ordenada en alguna estructura que facilita el acceso a esas tablas, ordenarlas y seleccionar filas de las tablas según criterios específicos. Las bases de datos generalmente tienen índices asociados a alguna de sus columnas, de forma que el acceso sea lo más rápido posible.</p> <p>Las Bases de Datos son, sin lugar a duda, las estructuras más utilizadas en ordenadores; ya que son el corazón de sistemas tan complejos como el censo de una nación, la nómina de empleados de una empresa, el sistema de facturación de una multinacional, o el medio por el que nos expiden el billete para las próximas vacaciones.</p> <p>PostgreSQL:</p> <p>PostgreSQL es una de las opciones más interesantes en bases de datos relacionales open-source. Michael Stonebraker inició el proyecto bajo el nombre Post Ingres a mediados de los 80's con la idea de solucionar problemas existentes en las bases de datos en esa época. MySQL fue por mucho tiempo el motor más popular; pero hoy es propiedad de Oracle y esto limita su evolución.</p> <p>Es gratuito y libre, además de que hoy nos ofrece una gran cantidad de opciones avanzadas. De hecho, es considerado el motor de base de datos más avanzado en la actualidad por lo tanto en esta práctica se usará esta base de datos</p>		

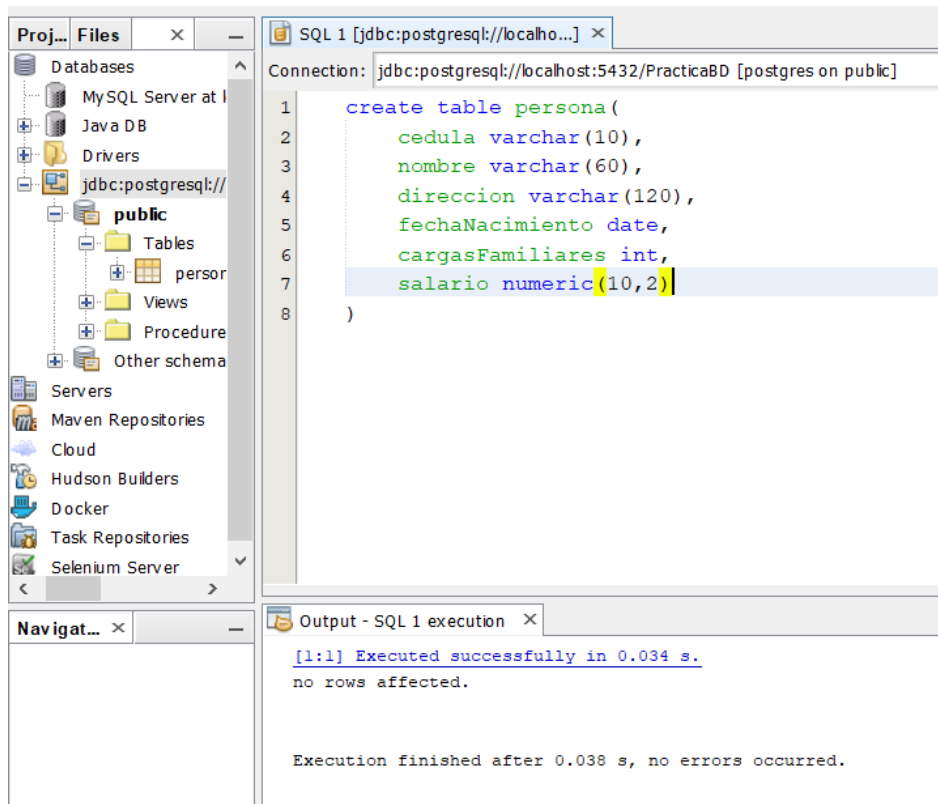
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

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



- 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

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



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

```

1  create table persona (
2      cedula varchar(10),
3      nombre varchar(60),
4      direccion varchar(120),
5      fechaNacimiento date,
6      cargasFamiliares int,
7      salario numeric(10,2)
8  )

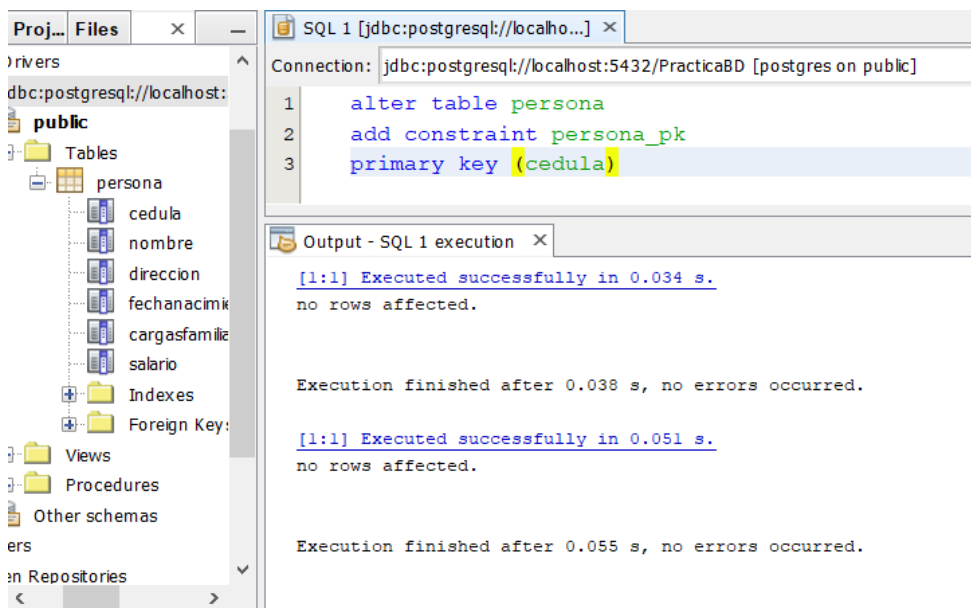
```

Output - SQL 1 execution

[1:1] Executed successfully in 0.034 s.
no rows affected.

Execution finished after 0.038 s, no errors occurred.

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



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

```

1  alter table persona
2  add constraint persona_pk
3  primary key (cedula)

```


Output - SQL 1 execution

[1:1] Executed successfully in 0.034 s.
no rows affected.

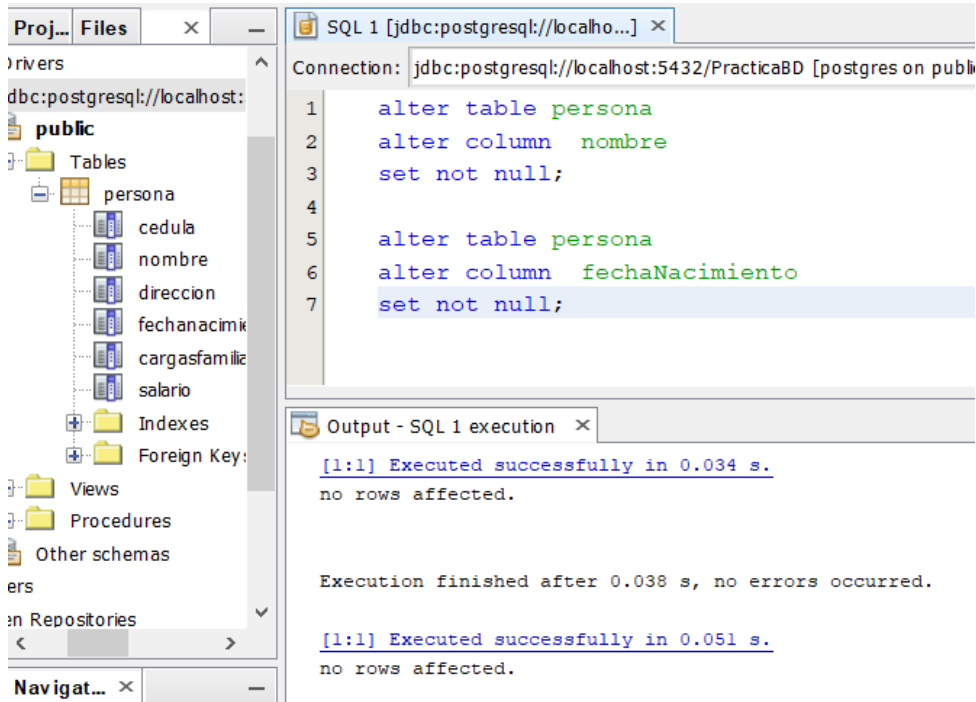
Execution finished after 0.038 s, no errors occurred.

[1:1] Executed successfully in 0.051 s.
no rows affected.

Execution finished after 0.055 s, no errors occurred.

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

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



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

```

1  alter table persona
2  alter column nombre
3  set not null;
4
5  alter table persona
6  alter column fechaNacimiento
7  set not null;

```

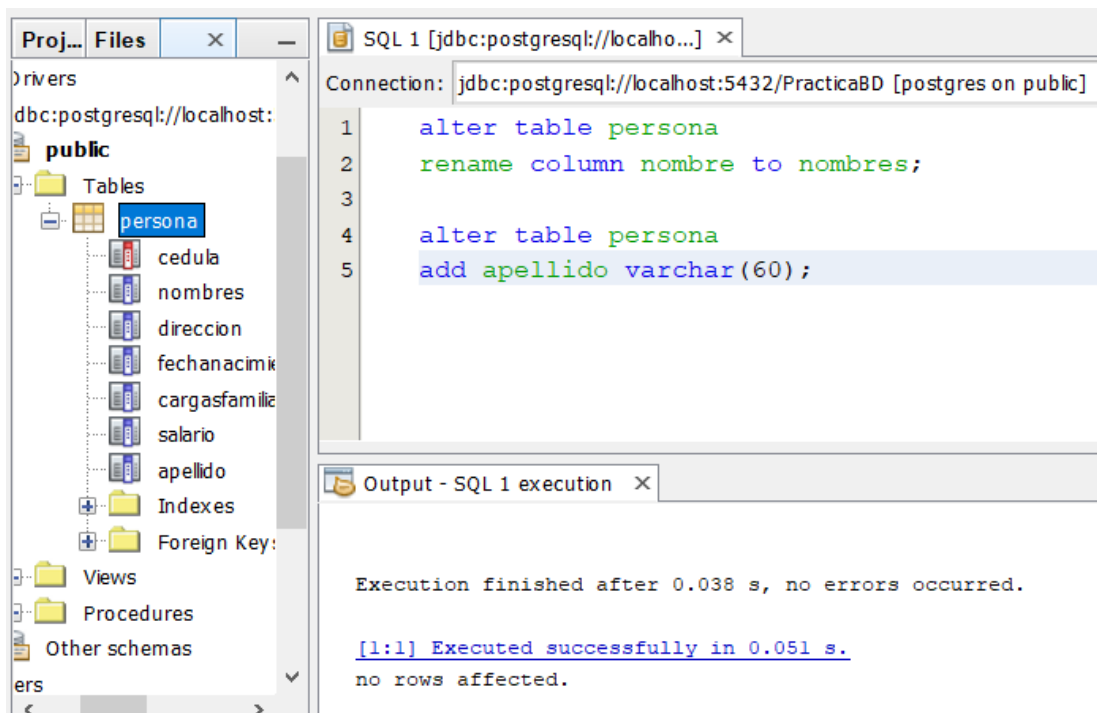
Output - SQL 1 execution

[1:1] Executed successfully in 0.034 s.
no rows affected.

Execution finished after 0.038 s, no errors occurred.

[1:1] Executed successfully in 0.051 s.
no rows affected.

- 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.



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

```


1  alter table persona
2  rename column nombre to nombres;
3
4  alter table persona
5  add apellido varchar(60);

```

Output - SQL 1 execution

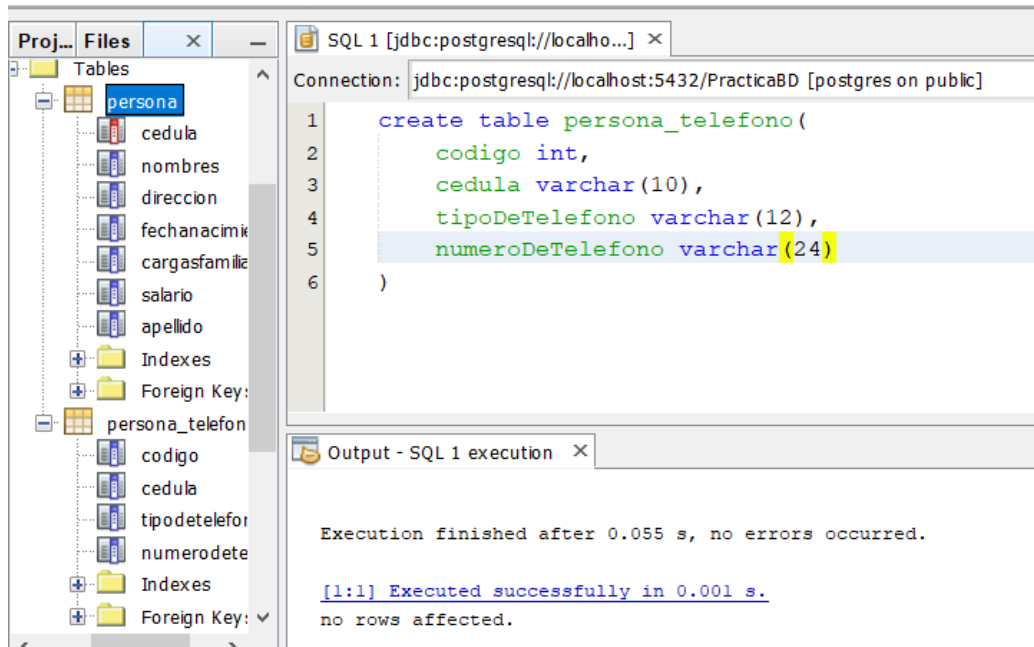
Execution finished after 0.038 s, no errors occurred.

[1:1] Executed successfully in 0.051 s.
no rows affected.

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

• **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 una dimensión de 12 caracteres, aquí se guardarán valores como celular o convencional
- Número de teléfono, con una dimensión de 24 caracteres



The screenshot shows a SQL IDE interface. On the left, a tree view displays the database structure, including a table named 'persona' and a newly created table 'persona_telefono'. The 'persona_telefono' table has columns: 'codigo', 'cedula', 'tipodetelefono', and 'numerodetelefono'. The main window shows the SQL command to create the table:

```

1 create table persona_telefono (
2     codigo int,
3     cedula varchar(10),
4     tipoDeTelefono varchar(12),
5     numeroDeTelefono varchar(24)
6 )

```

The output window shows the execution results:

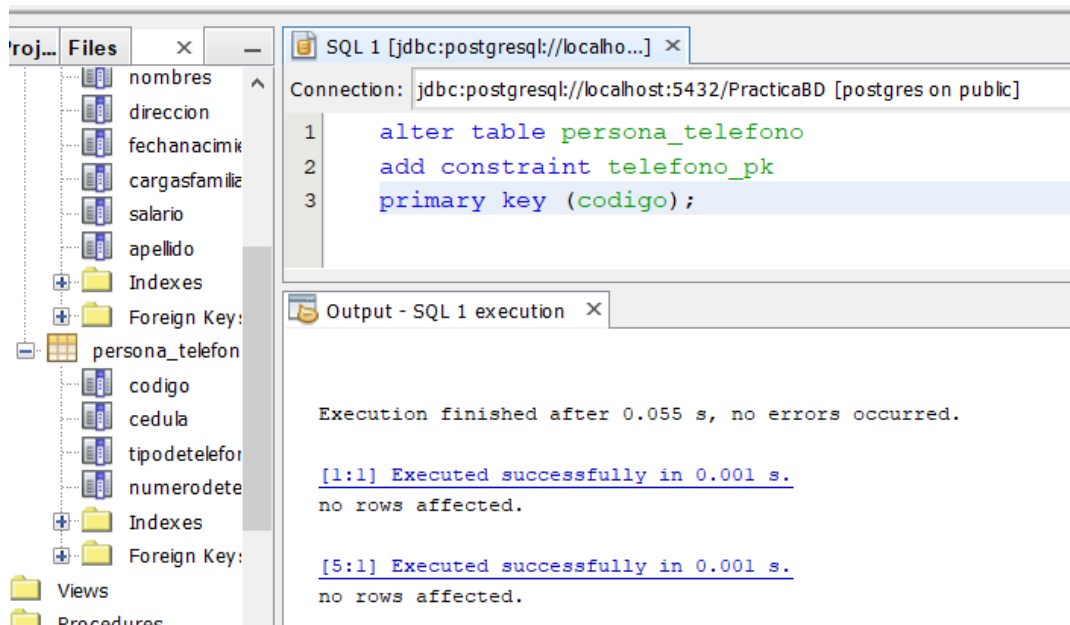
```

Execution finished after 0.055 s, no errors occurred.

[1:1] Executed successfully in 0.001 s.
no rows affected.

```

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



The screenshot shows the same SQL IDE interface. The main window now displays the SQL command to alter the table and add a primary key constraint:

```

1 alter table persona_telefono
2 add constraint telefono_pk
3 primary key (codigo);

```

The output window shows the execution results:

```

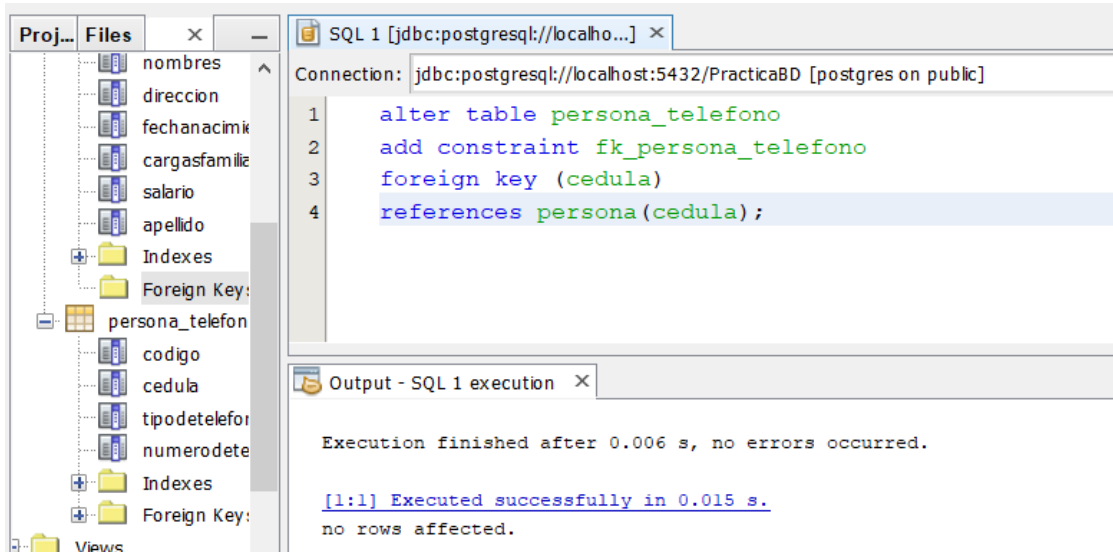
Execution finished after 0.055 s, no errors occurred.

[1:1] Executed successfully in 0.001 s.
no rows affected.

[5:1] Executed successfully in 0.001 s.
no rows affected.

```

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

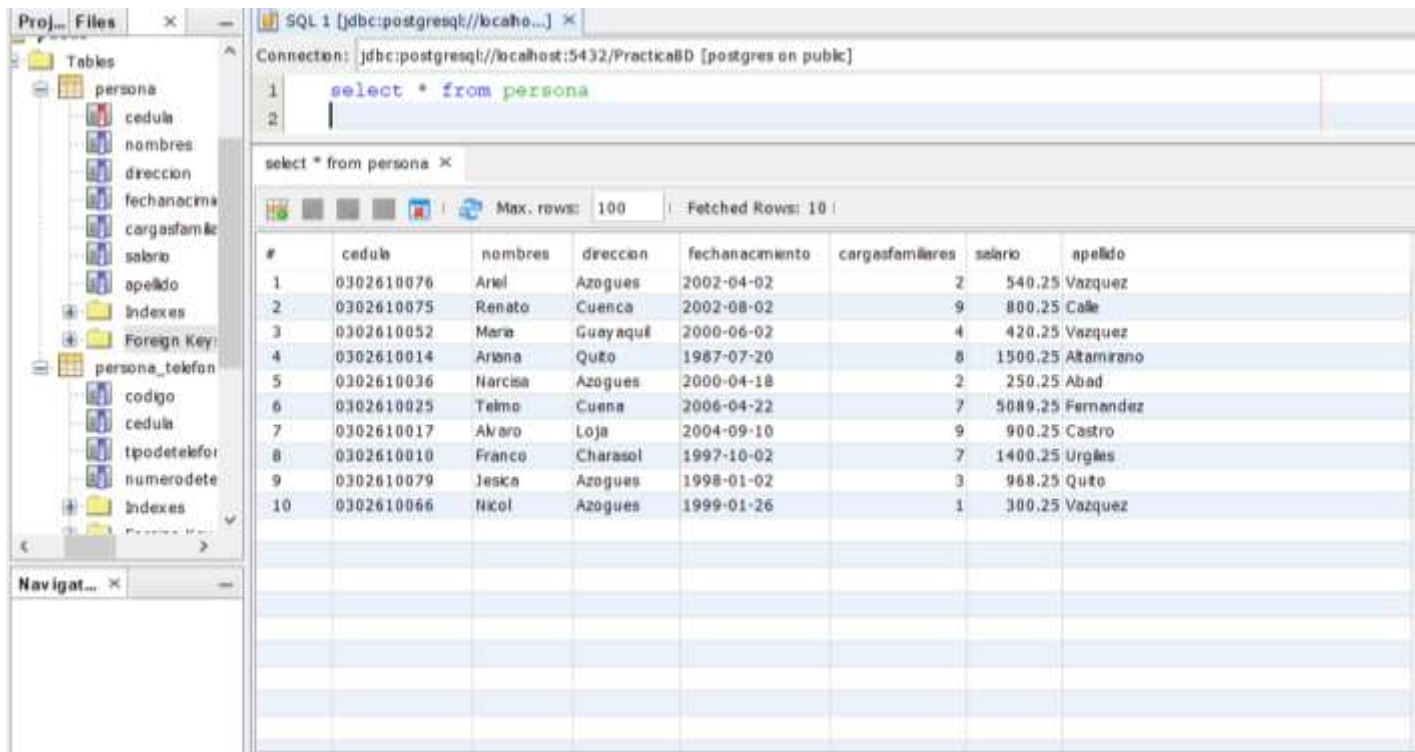


The screenshot shows a database management tool interface. On the left, a tree view displays the database structure, including tables like 'nombres', 'direccion', 'fechanacimiento', 'cargasfamiliares', 'salario', 'apellido', 'Indexes', 'Foreign Key:', 'persona_telefono', 'codigo', 'cedula', 'tipodetelefono', 'numerodetelefono', 'Indexes', 'Foreign Key:', and 'Views'. The main window displays the SQL command:

```
1 alter table persona_telefono
2 add constraint fk_persona_telefono
3 foreign key (cedula)
4 references persona (cedula);
```

Below the command, the output shows: "Execution finished after 0.006 s, no errors occurred." and "[1:1] Executed successfully in 0.015 s. no rows affected."

- **SELECT de todos los registros de personas**



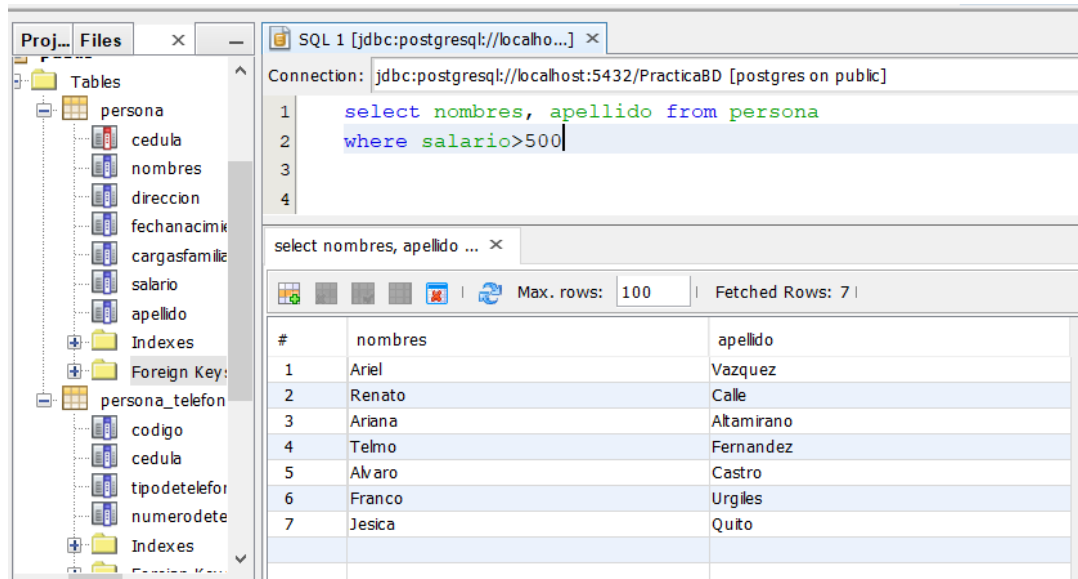
The screenshot shows the same database management tool interface. The main window displays the SQL command:

```
1 select * from persona
2
```

Below the command, the output shows a table with 10 rows of data. The table has columns: #, cedula, nombres, direccion, fechanacimiento, cargasfamiliares, salario, and apellido. The data is as follows:

#	cedula	nombres	direccion	fechanacimiento	cargasfamiliares	salario	apellido
1	0302610076	Ariel	Azogues	2002-04-02	2	540.25	Vazquez
2	0302610075	Renato	Cuenca	2002-08-02	9	800.25	Calle
3	0302610052	Maria	Guayaquil	2000-06-02	4	420.25	Vazquez
4	0302610014	Ariana	Quito	1987-07-20	8	1500.25	Altamirano
5	0302610036	Narcisa	Azogues	2000-04-18	2	250.25	Abad
6	0302610025	Telmo	Cuenca	2006-04-22	7	5089.25	Fernandez
7	0302610017	Alvaro	Loja	2004-09-10	9	900.25	Castro
8	0302610010	Franco	Charasol	1997-10-02	7	1400.25	Urgales
9	0302610079	Jesica	Azogues	1998-01-02	3	968.25	Quito
10	0302610066	Nicol	Azogues	1999-01-26	1	300.25	Vazquez

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



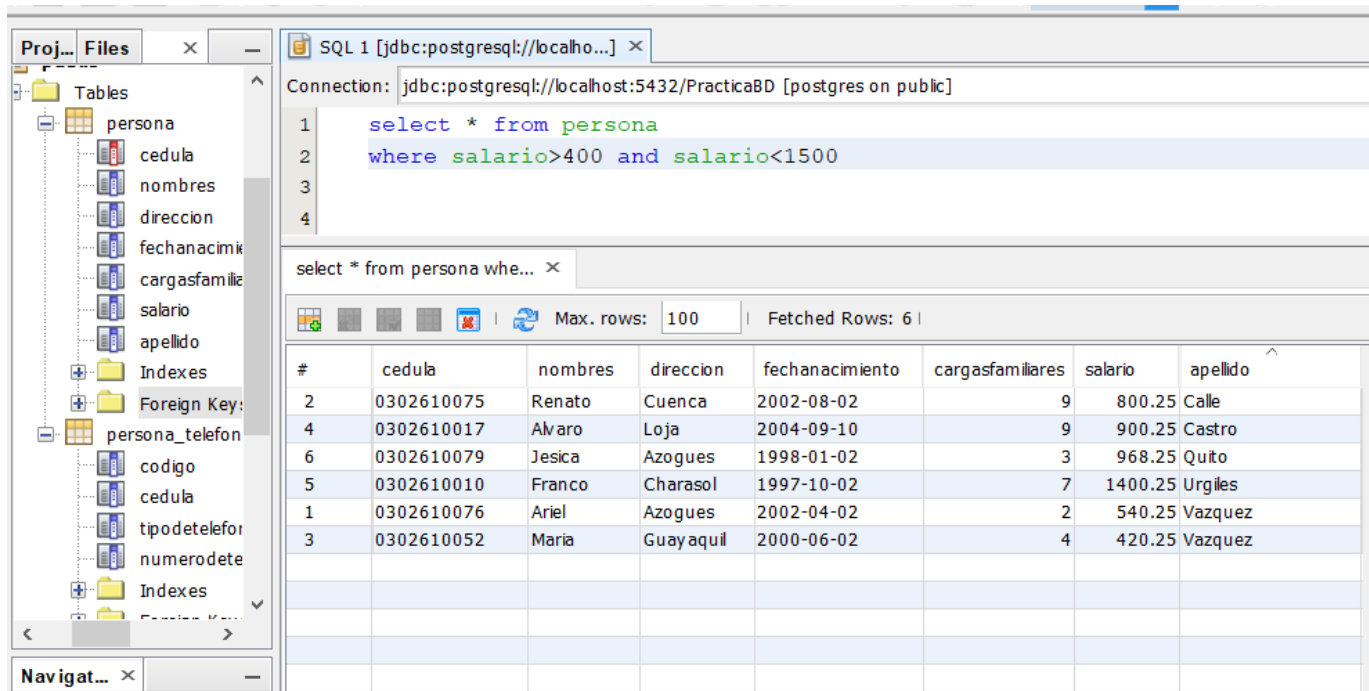
The screenshot shows a database management tool interface. On the left, a tree view displays the database structure, including tables like 'persona', 'cedula', 'nombres', 'direccion', 'fechanacimiento', 'cargasfamiliares', 'salario', 'apellido', 'Indexes', 'Foreign Key', 'persona_telefono', 'codigo', 'tipodetelefono', and 'numerodetelefono'. The main window displays a SQL query in a text editor:

```
1 select nombres, apellido from persona
2 where salario > 500
3
4
```

Below the query editor, the results are displayed in a table. The table has 7 rows and 3 columns: '#', 'nombres', and 'apellido'. The data is as follows:

#	nombres	apellido
1	Ariel	Vazquez
2	Renato	Calle
3	Ariana	Altamirano
4	Telmo	Fernandez
5	Alvaro	Castro
6	Franco	Urgiles
7	Jesica	Quito

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



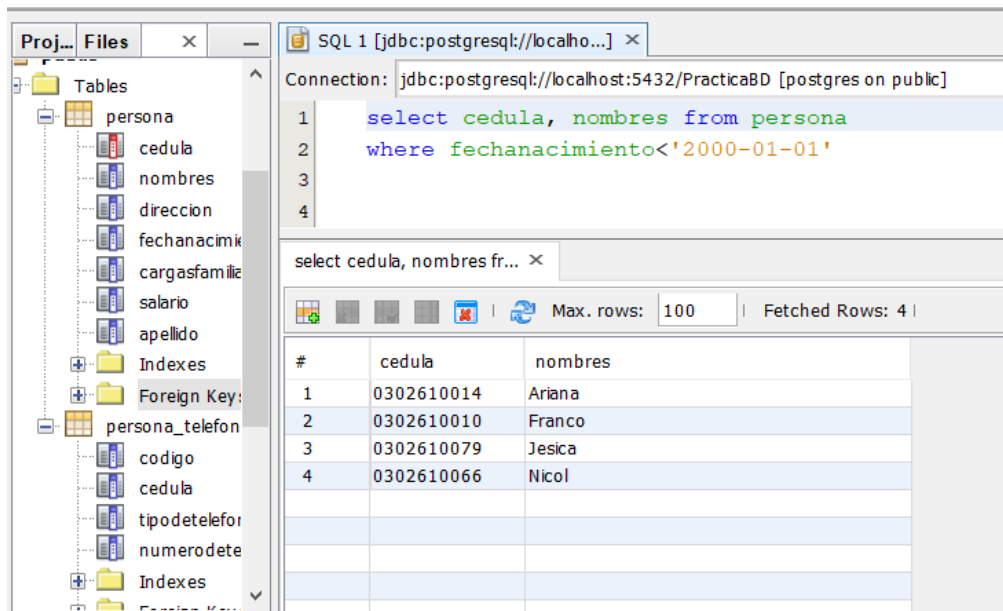
The screenshot shows a database management tool interface. On the left, a tree view displays the database structure, including tables like 'persona', 'cedula', 'nombres', 'direccion', 'fechanacimiento', 'cargasfamiliares', 'salario', 'apellido', 'Indexes', 'Foreign Key', 'persona_telefono', 'codigo', 'tipodetelefono', and 'numerodetelefono'. The main window displays a SQL query in a text editor:

```
1 select * from persona
2 where salario > 400 and salario < 1500
3
4
```

Below the query editor, the results are displayed in a table. The table has 8 columns: '#', 'cedula', 'nombres', 'direccion', 'fechanacimiento', 'cargasfamiliares', 'salario', and 'apellido'. The data is as follows:

#	cedula	nombres	direccion	fechanacimiento	cargasfamiliares	salario	apellido
2	0302610075	Renato	Cuenca	2002-08-02	9	800.25	Calle
4	0302610017	Alvaro	Loja	2004-09-10	9	900.25	Castro
6	0302610079	Jesica	Azogues	1998-01-02	3	968.25	Quito
5	0302610010	Franco	Charasol	1997-10-02	7	1400.25	Urgiles
1	0302610076	Ariel	Azogues	2002-04-02	2	540.25	Vazquez
3	0302610052	Maria	Guayaquil	2000-06-02	4	420.25	Vazquez

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



The screenshot shows a database management tool interface. On the left, a tree view displays the database structure, including tables like 'persona', 'cedula', 'nombres', 'direccion', 'fechanacimiento', 'cargasfamiliares', 'salario', 'apellido', 'persona_telefono', 'codigo', 'tipodetelefono', and 'numerosdetelefono'. The main window displays a SQL query in a text editor:

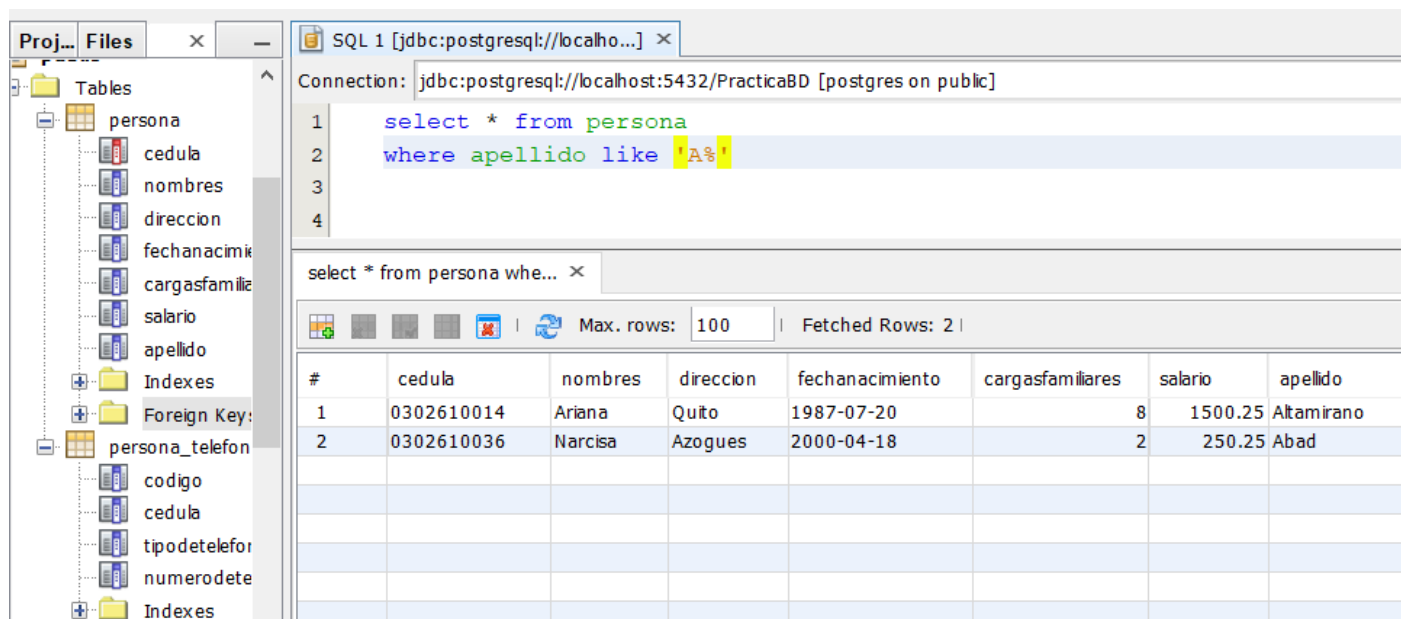
```
1 select cedula, nombres from persona
2 where fechanacimiento < '2000-01-01'
3
4
```

Below the query editor, the results of the query are displayed in a table. The table has two columns: 'cedula' and 'nombres'. The results are as follows:

#	cedula	nombres
1	0302610014	Ariana
2	0302610010	Franco
3	0302610079	Jesica
4	0302610066	Nicol

- Selecciones los datos de las personas

cuyo apellido empiece con la letra "a".



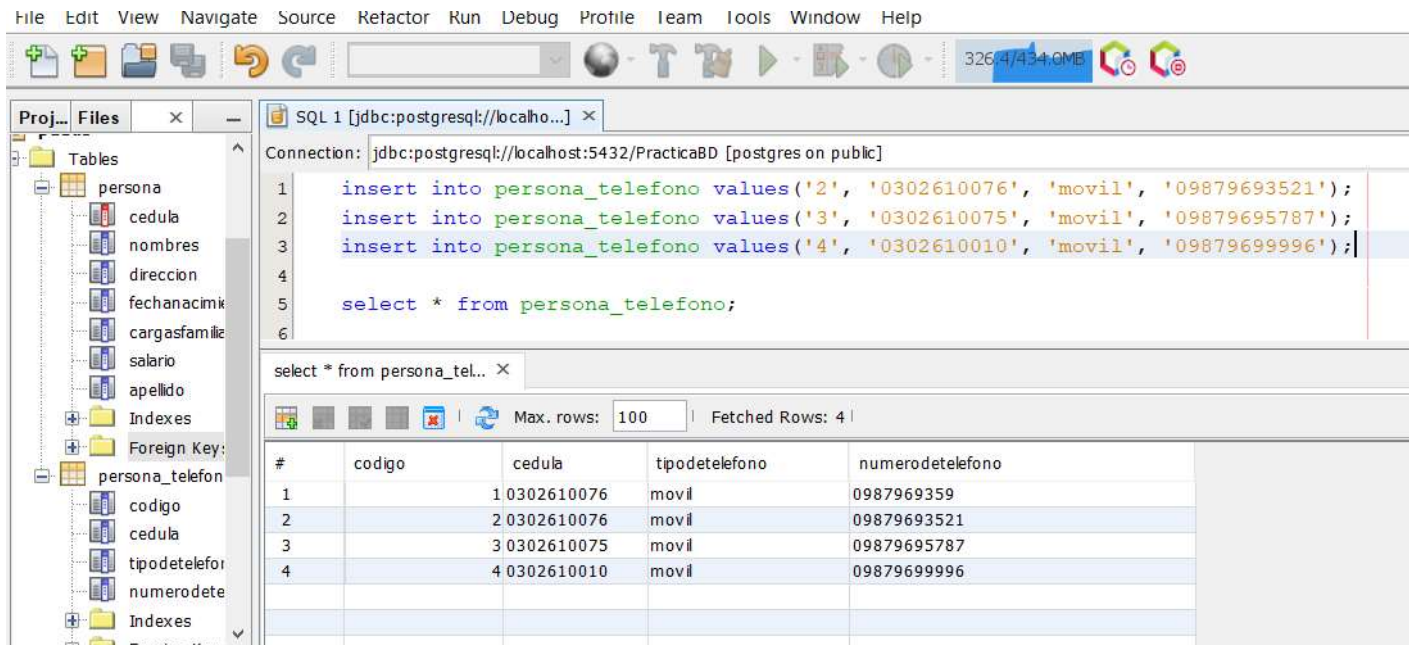
The screenshot shows the same database management tool interface. The SQL query in the text editor is now:

```
1 select * from persona
2 where apellido like 'A%'
3
4
```

Below the query editor, the results of the query are displayed in a table. The table has eight columns: '#', 'cedula', 'nombres', 'direccion', 'fechanacimiento', 'cargasfamiliares', 'salario', and 'apellido'. The results are as follows:

#	cedula	nombres	direccion	fechanacimiento	cargasfamiliares	salario	apellido
1	0302610014	Ariana	Quito	1987-07-20	8	1500.25	Altamirano
2	0302610036	Narcisa	Azogues	2000-04-18	2	250.25	Abad

- 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/PracticaBD [postgres on public]

```

1 insert into persona_telefono values ('2', '0302610076', 'movil', '09879693521');
2 insert into persona_telefono values ('3', '0302610075', 'movil', '09879695787');
3 insert into persona_telefono values ('4', '0302610010', 'movil', '09879699996');
4
5 select * from persona_telefono;
6

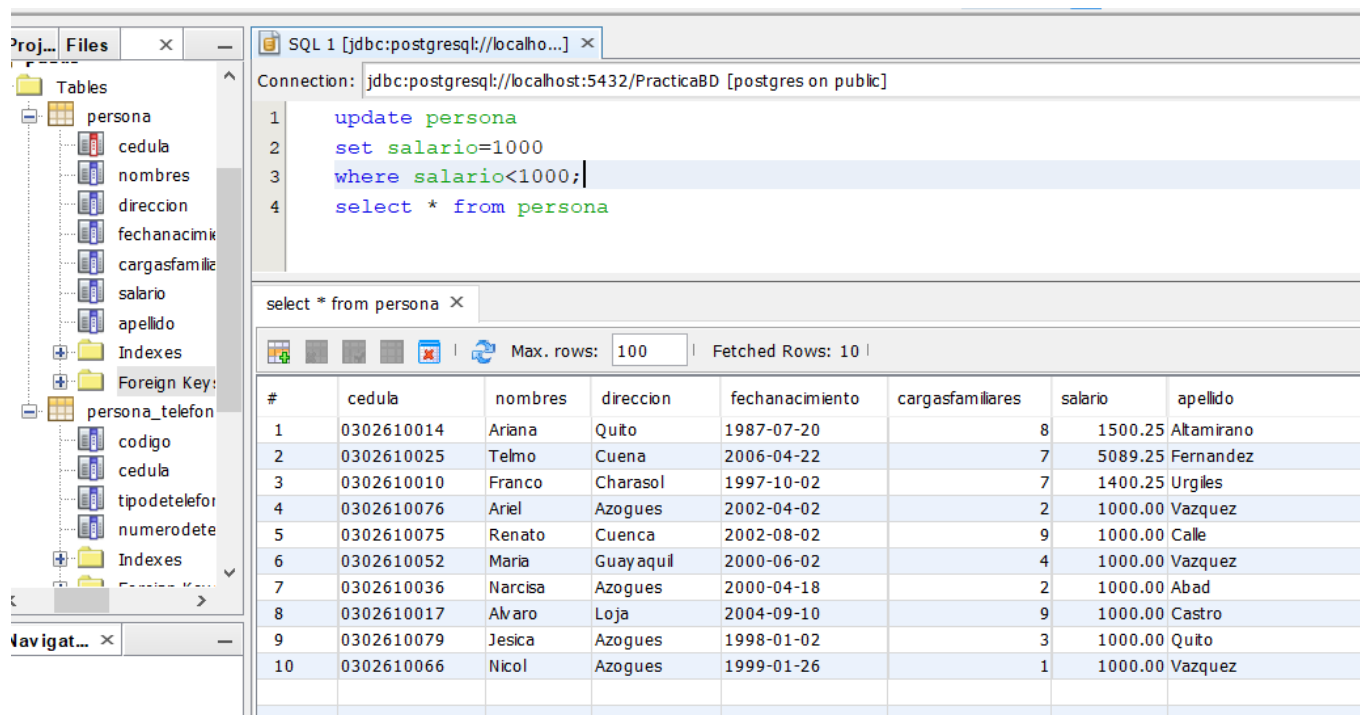
```

select * from persona_tel...

Max. rows: 100 | Fetched Rows: 4

#	codigo	cedula	tipodetelefono	numerodetelefono
1		1 0302610076	movil	0987969359
2		2 0302610076	movil	09879693521
3		3 0302610075	movil	09879695787
4		4 0302610010	movil	09879699996

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



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

```

1 update persona
2 set salario=1000
3 where salario<1000;
4 select * from persona

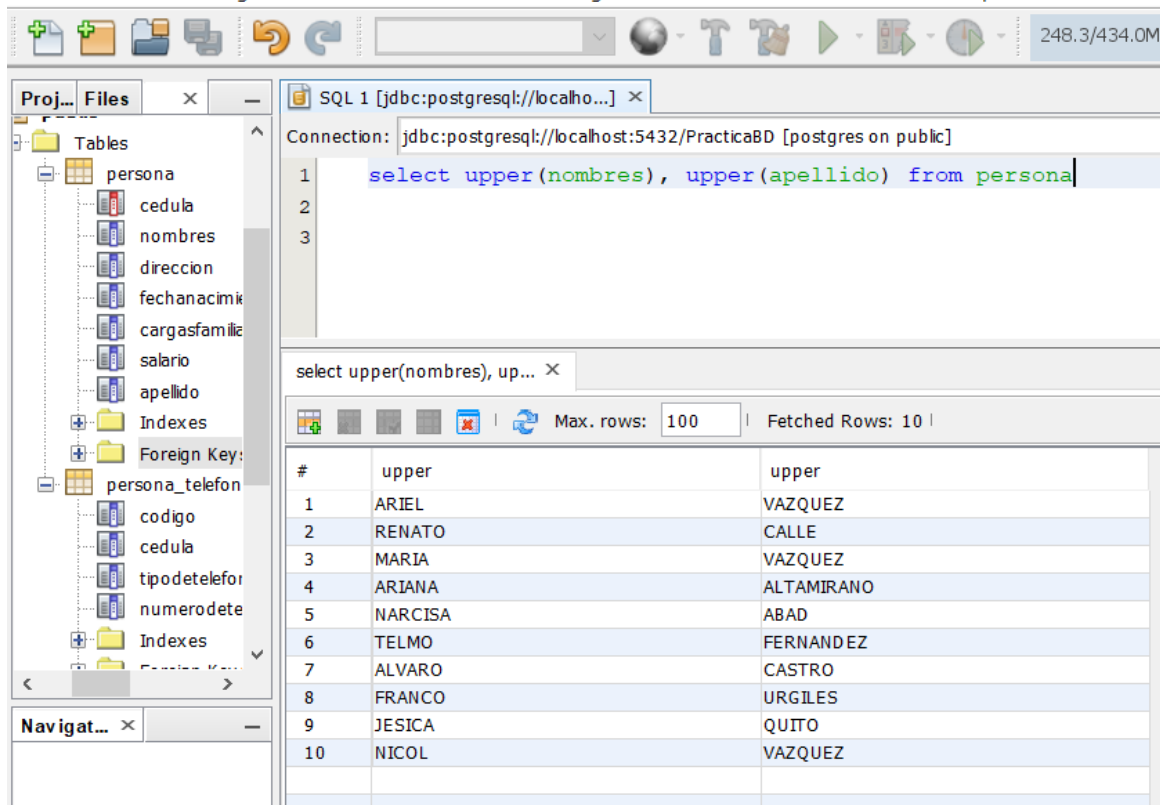
```

select * from persona

Max. rows: 100 | Fetched Rows: 10

#	cedula	nombres	direccion	fechanacimiento	cargasfamiliares	salario	apellido
1	0302610014	Ariana	Quito	1987-07-20	8	1500.25	Altamirano
2	0302610025	Telmo	Cuenca	2006-04-22	7	5089.25	Fernandez
3	0302610010	Franco	Charasol	1997-10-02	7	1400.25	Urgiles
4	0302610076	Ariel	Azogues	2002-04-02	2	1000.00	Vazquez
5	0302610075	Renato	Cuenca	2002-08-02	9	1000.00	Calle
6	0302610052	Maria	Guayaquil	2000-06-02	4	1000.00	Vazquez
7	0302610036	Narcisa	Azogues	2000-04-18	2	1000.00	Abad
8	0302610017	Alvaro	Loja	2004-09-10	9	1000.00	Castro
9	0302610079	Jesica	Azogues	1998-01-02	3	1000.00	Quito
10	0302610066	Nicol	Azogues	1999-01-26	1	1000.00	Vazquez

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



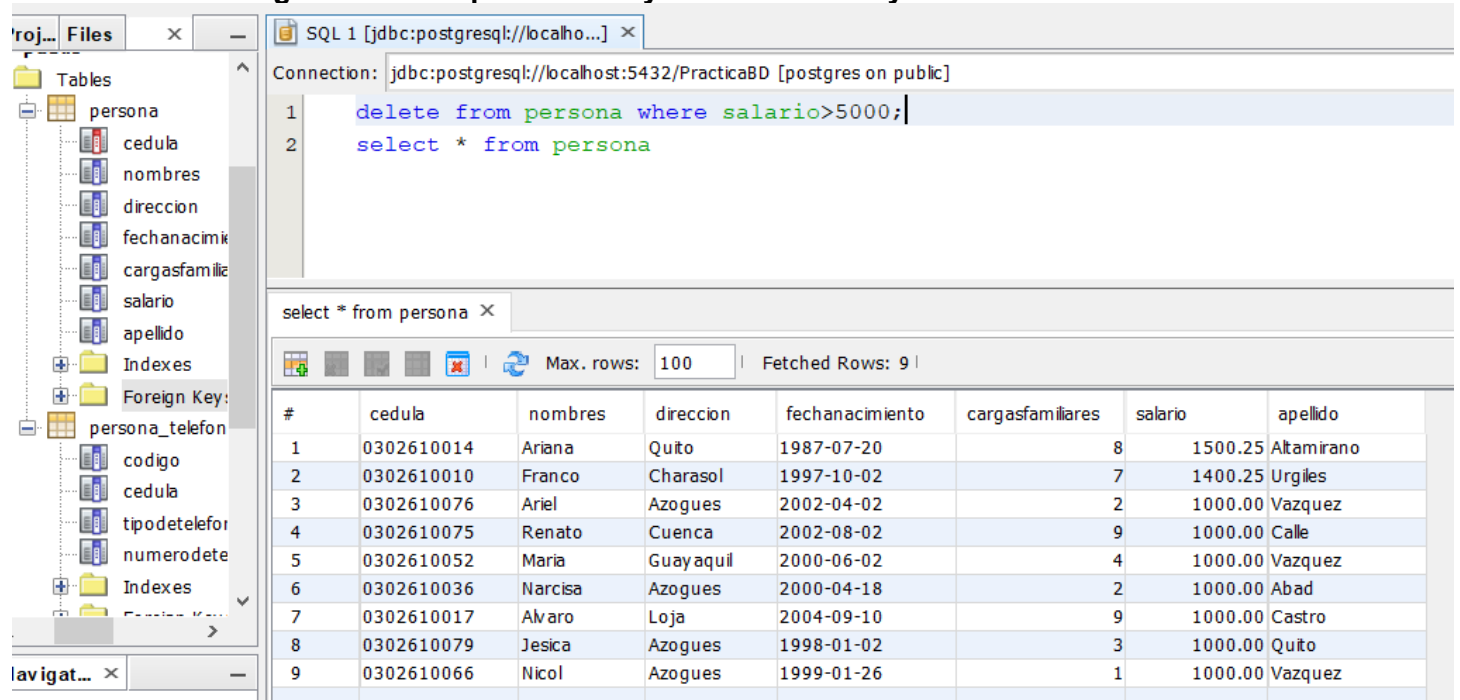
The screenshot shows a database management tool interface. On the left, a tree view displays the database structure, including tables like 'persona', 'cedula', 'nombres', 'direccion', 'fechanacimiento', 'cargasfamiliares', 'salario', 'apellido', 'persona_telefono', 'codigo', 'tipodetelefono', and 'numerodetelefono'. The main window shows a SQL query editor with the following query:

```
select upper(nombres), upper(apellido) from persona
```

Below the query editor, the results of the query are displayed in a table with 10 rows. The columns are labeled '#', 'upper', and 'upper'.

#	upper	upper
1	ARIEL	VAZQUEZ
2	RENATO	CALLE
3	MARIA	VAZQUEZ
4	ARIANA	ALTAMIRANO
5	NARCISA	ABAD
6	TELMO	FERNANDEZ
7	ALVARO	CASTRO
8	FRANCO	URGILES
9	JESICA	QUITO
10	NICOL	VAZQUEZ

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



The screenshot shows the same database management tool interface. The SQL query editor now contains the following query:

```
delete from persona where salario>5000;
select * from persona
```

Below the query editor, the results of the query are displayed in a table with 9 rows. The columns are labeled '#', 'cedula', 'nombres', 'direccion', 'fechanacimiento', 'cargasfamiliares', 'salario', and 'apellido'.

#	cedula	nombres	direccion	fechanacimiento	cargasfamiliares	salario	apellido
1	0302610014	Ariana	Quito	1987-07-20	8	1500.25	Altamirano
2	0302610010	Franco	Charasol	1997-10-02	7	1400.25	Urgiles
3	0302610076	Ariel	Azogues	2002-04-02	2	1000.00	Vazquez
4	0302610075	Renato	Cuenca	2002-08-02	9	1000.00	Calle
5	0302610052	Maria	Guayaquil	2000-06-02	4	1000.00	Vazquez
6	0302610036	Narcisa	Azogues	2000-04-18	2	1000.00	Abad
7	0302610017	Alvaro	Loja	2004-09-10	9	1000.00	Castro
8	0302610079	Jesica	Azogues	1998-01-02	3	1000.00	Quito
9	0302610066	Nicol	Azogues	1999-01-26	1	1000.00	Vazquez

FOREIGN KEY

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

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

312.9/440.0MB

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

Connection: jdbc:postgresql://localhost:5432/PracticaBD [postgres on public]

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

select p.nombres, p.apell...

Max. rows: 100 | Fetched Rows: 36 |

#	nombres	apellido	codigo	cedula	tipodetefono	numerodetefono
1	Ariana	Altamirano	1	0302610076	movil	0987969359
2	Franco	Urgiles	1	0302610076	movil	0987969359
3	Ariel	Vazquez	1	0302610076	movil	0987969359
4	Renato	Calle	1	0302610076	movil	0987969359
5	Maria	Vazquez	1	0302610076	movil	0987969359
6	Narcisa	Abad	1	0302610076	movil	0987969359
7	Alvaro	Castro	1	0302610076	movil	0987969359
8	Jesica	Quito	1	0302610076	movil	0987969359
9	Nicol	Vazquez	1	0302610076	movil	0987969359
10	Ariana	Altamirano	2	0302610076	movil	09879693521
11	Franco	Urgiles	2	0302610076	movil	09879693521
12	Ariel	Vazquez	2	0302610076	movil	09879693521
13	Renato	Calle	2	0302610076	movil	09879693521
14	Maria	Vazquez	2	0302610076	movil	09879693521
15	Narcisa	Abad	2	0302610076	movil	09879693521
16	Alvaro	Castro	2	0302610076	movil	09879693521
17	Jesica	Quito	2	0302610076	movil	09879693521
18	Nicol	Vazquez	2	0302610076	movil	09879693521
19	Ariana	Altamirano	3	0302610075	movil	09879695787
20	Franco	Urgiles	3	0302610075	movil	09879695787
21	Ariel	Vazquez	3	0302610075	movil	09879695787
22	Renato	Calle	3	0302610075	movil	09879695787
23	Maria	Vazquez	3	0302610075	movil	09879695787
24	Narcisa	Abad	3	0302610075	movil	09879695787
25	Alvaro	Castro	3	0302610075	movil	09879695787
26	Jesica	Quito	3	0302610075	movil	09879695787

<No View Available>

- 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/PracticaBD [postgres on public]

```

1 select p.*, pt.numerodetelefono
2 from persona as p, persona_telefono as pt
3 where p.cedula=pt.cedula and p.salario=1000

```

select p.*, pt.numerodete... ×

Max. rows: 100 | Fetched Rows: 3 |

#	cedula	nombres	direccion	fechanacimiento	cargasfamiliares	salario	apellido	numerodetelefono
1	0302610076	Ariel	Azogues	2002-04-02	2	1000.00	Vazquez	0987969
2	0302610076	Ariel	Azogues	2002-04-02	2	1000.00	Vazquez	0987969
3	0302610075	Renato	Cuenca	2002-08-02	9	1000.00	Calle	0987969

jdbc:postgres public
select p.*,
from person

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

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

```


1 select numeroDeTelefono from persona_telefono

```

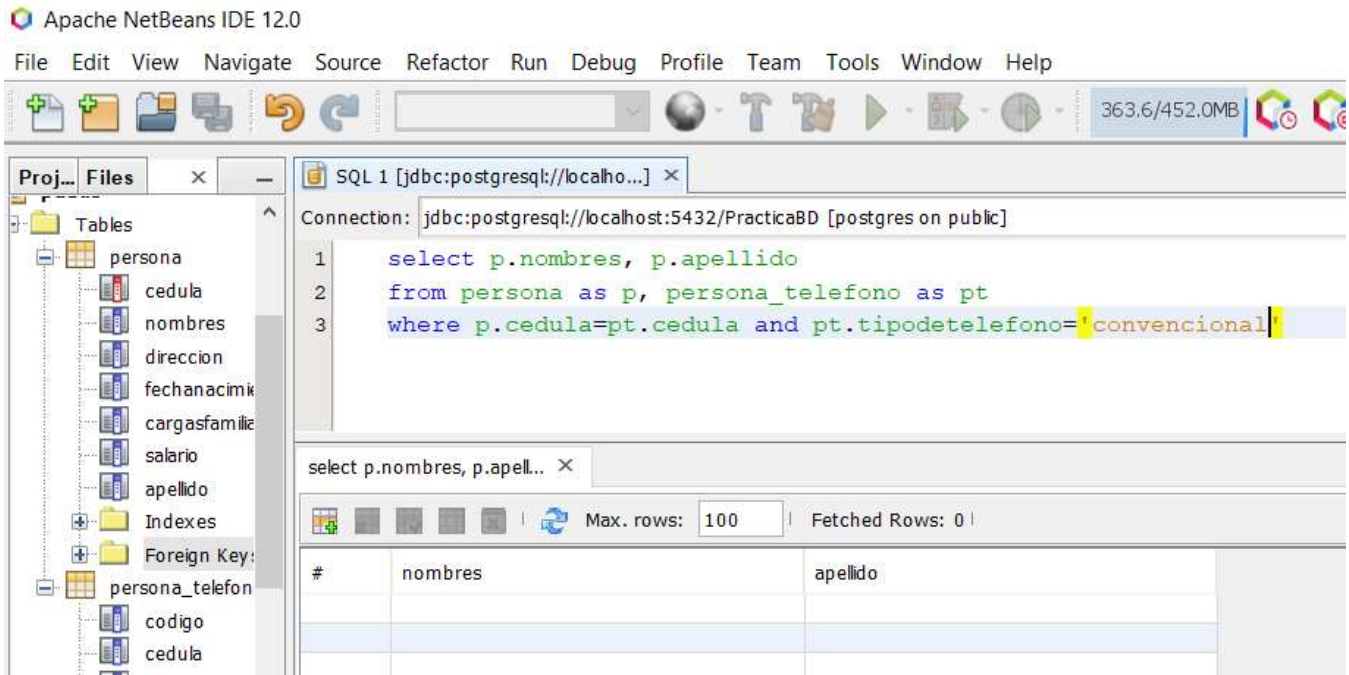
select numeroDeTelefono f... ×

Max. rows: 100 | Fetched Rows: 4 |

#	numerodetelefono
1	0987969359
2	09879693521
3	09879695787
4	09879699996

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

- 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

CONCLUSIONES:

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

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

RECOMENDACIONES:

- Realizar el trabajo dentro del tiempo establecido.

Nombre de estudiante: Ariel Vazquez

Firma de estudiante:

