UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

PRÁCTICA DE LABORATORIO

CARRERA: COMPUTACION		**ASIGNATURA**: PROGRAMACION APLICADA
NRO. PRÁCTICA:	5	**TÍTULO PRÁCTICA**: Bases 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

1. Revisar la teoría y conceptos de Base de Datos

Un lenguaje de definición de datos (Data Definition Language, DDL por sus siglas en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de esta llevar a cabo las tareas de definición de las estructuras que almacenarán los datos, así como de los procedimientos o funciones que permitan consultarlos.

La definición de la estructura de la base de datos incluye tanto la creación inicial de los diferentes objetos que formarán la base de datos, como el mantenimiento de esa estructura. Las sentencias del DDL utilizan unos verbos que se repiten para los distintos objetos. Por ejemplo, para crear un objeto nuevo el verbo será CREATE y a continuación el tipo de objeto a crear. CREATE DATABASE es la sentencia para crear una base de datos, CREATE TABLE nos permite crear una nueva tabla, CREATE INDEX crear un nuevo índice… Para eliminar un objeto utilizaremos el verbo DROP (DROP TABLE, DROP INDEX…) y para modificar algo de la definición de un objeto ya creado utilizamos el verbo ALTER (ALTER TABLE, ALTER INDEX…).

Las principales funcionalidades de SQL como lenguaje de definición (DDL) son la creación, modificación y borrado de las tablas que componen la base de datos, así como de los índices, vistas, sinónimos, permisos, etc. que pudieran definirse sobre las mismas. Este documento introduce los comandos para el trabajo básico con tablas.

Lenguaje de Manipulación de Datos (Data Manipulation Language, DML) es un idioma proporcionado por los sistemas gestores de bases de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o modificación de los datos contenidos en las Bases de Datos del Sistema Gestor de Bases de Datos. El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional. Otros ejemplos de DML son los usados por bases de datos IMS/DL1, CODASYL u otras.

Elementos del lenguaje de manipulación de datos:

Select, Insert, Delete y Update

Clasificación de los DML.

Se clasifican en dos grandes grupos:

- Lenguajes de consulta procedimentales

Lenguajes procedimentales. En este tipo de lenguaje el usuario da instrucciones al sistema para que realice una serie de procedimientos u operaciones en la base de datos para calcular un resultado final.

- Lenguajes de consulta no procedimentales

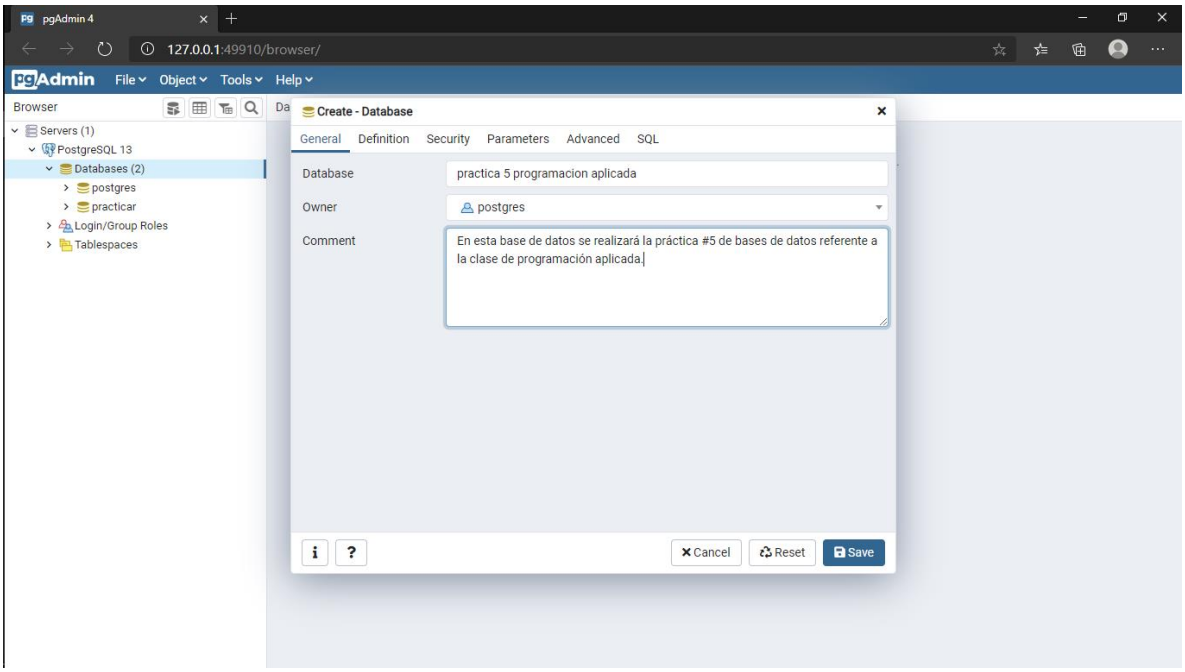
En los lenguajes no procedimentales el usuario describe la información deseada sin un procedimiento específico para obtener esa información.

(Vargas, 2005)

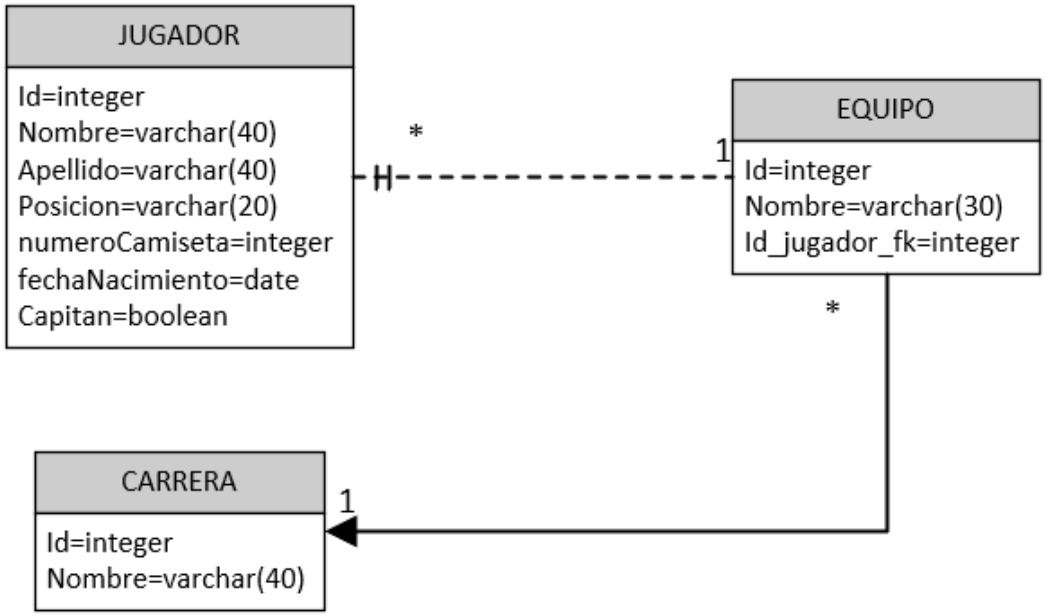
Bibliography

Vargas, G. d. (2005, Noviembre 29). *Diseño de bases de datos*. Retrieved from monografias: https://www.monografias.com/trabajos30/base-datos/base-datos.shtml

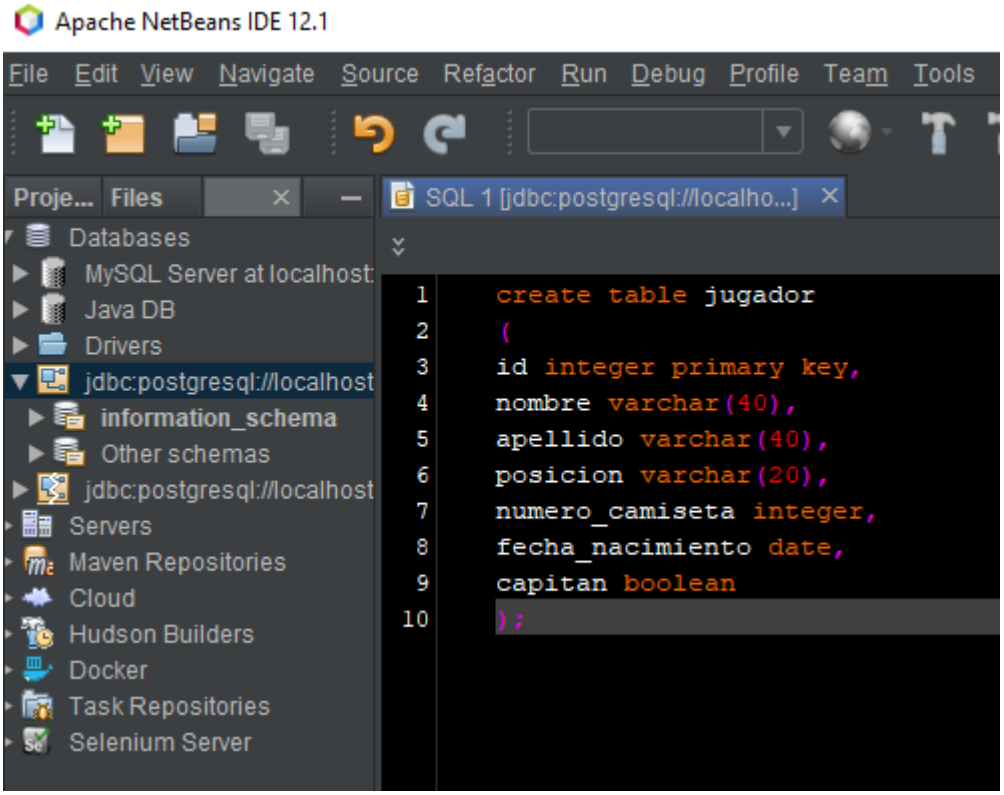
2. Diseñar e implementar las siguientes prácticas de Base de datos solamente utilizando código SQL (No se permite el uso del generador o ayuda de ningún IDE):



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



Creación de la tabla jugador:



Creación de la tabla carrera:

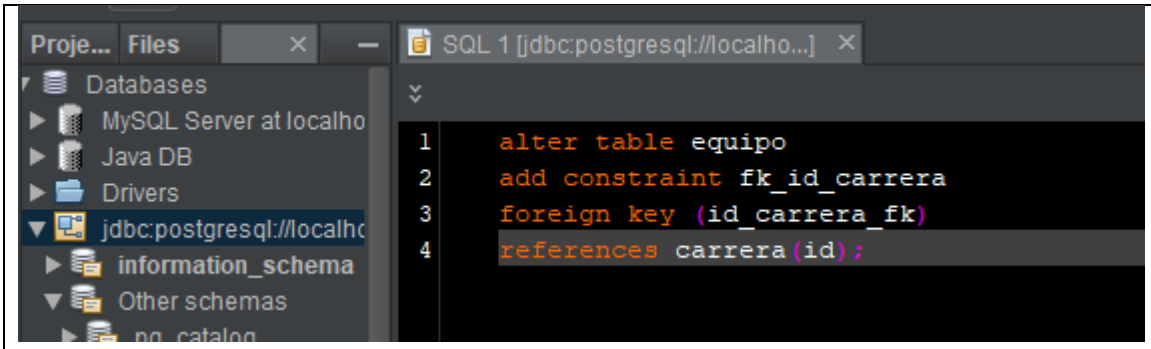
```
create table carrera
(
id integer primary key,
nombre varchar(40)
);
```

Creación de la tabla equipo:

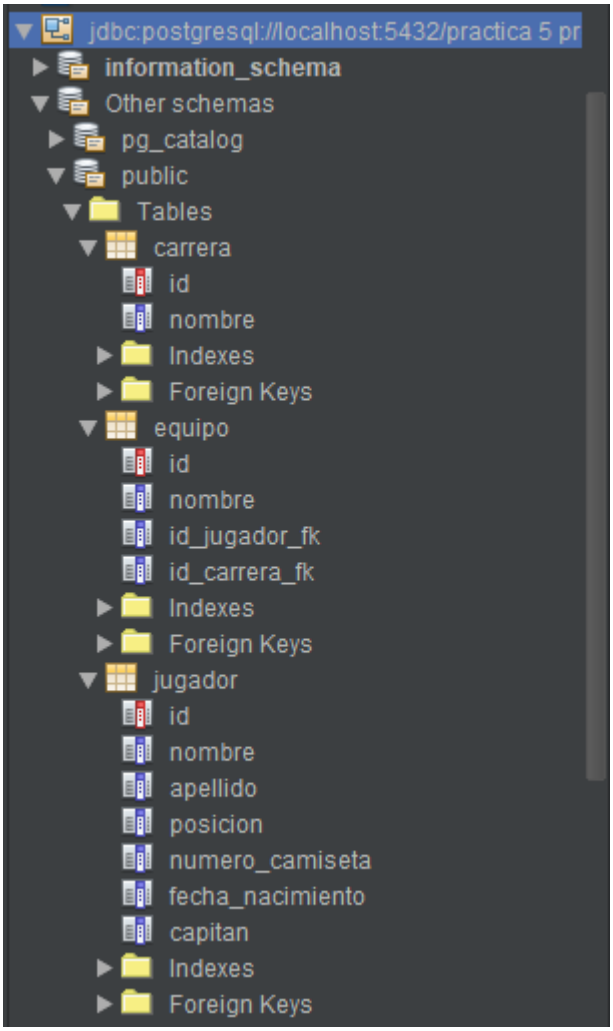
```
create table equipo
(
id integer primary key,
nombre varchar(30),
id_jugador_fk integer,
id_carrera_fk integer
);
```

Seteo de las foreign keys:

```
alter table equipo
add constraint fk_id_jugador
foreign key (id_jugador_fk)
references jugador(id);
```



Demostración de las tablas creadas:



Ahora vamos a crear un jugador:

```
insert into jugador
(id, nombre, apellido, posicion, numero_camiseta, fecha_nacimiento, capitan)
values
(1, 'Cristiano', 'Ronaldo', 'Extremo derecho', 7, '1985-02-05', TRUE)
```

Ahora creamos una carrera:

```
insert into carrera
(id, nombre)
values
(1, 'Ingenieria en Computacion')
```

Ahora creamos un equipo:

```
insert into equipo
(id, nombre, id_jugador_fk, id_carrera_fk)
values
(1, 'Juventus', 1, 1)
```

Por último, mostramos la tabla con los datos:

```
select e.id, e.nombre,
j.id, j.nombre || ' ' || j.apellido as nombre_jugador, j.posicion, j.numero_camiseta,
c.id, c.nombre
from equipo as e, jugador as j, carrera as c
where e.id_jugador_fk = j.id and e.id_carrera_fk = c.id
```

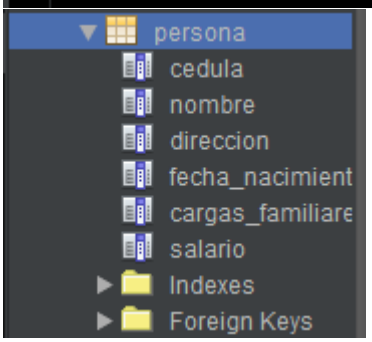
select e.id, e.nombre, j.... ×

Max. rows: 100 Fetched Rows: 1 Matching Ro

#	id	nombre	id	nombre_jugador	posicion	numero_camiseta	id	nombre
1	1	Juventus	1	Cristiano Ronaldo	Extremo derecho	7	1	Ingenieria en Computacion

- 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

```
create table persona
(
cedula varchar(10),
nombre varchar(60),
direccion varchar(120),
fecha_nacimiento date,
cargas_familiares integer,
salario numeric(10,2)
)
```



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

```
alter table persona
add constraint cedula_pk
primary key (cedula)
```

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

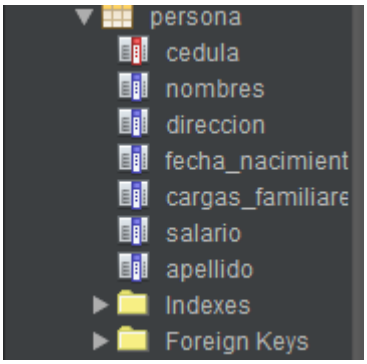
```
alter table persona
alter column nombre
set not null
```

```
alter table persona
alter column fecha_nacimiento
set not null
```

4) 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.

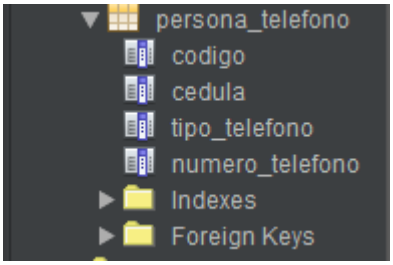
```
alter table persona
rename column nombre
to nombres
```

```
alter table persona
add apellido varchar(60)
```



- 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
(
codigo integer,
cedula varchar(10),
tipo_telefono varchar(12),
numero_telefono varchar(24)
)
```

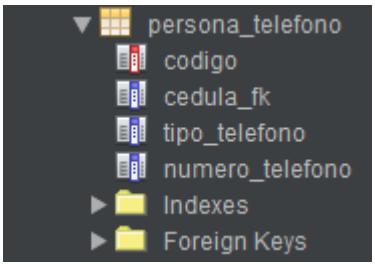


6) Modificar la tabla PERSONA_TELEFONO de tal manera que el campo código sea clave primaria.

```
1  alter table persona_telefono
2  add constraint codigo_pk
3  primary key (codigo)
```

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

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



2.3 SELECT de todos los registros de personas.

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

select * from persona ×

Max. rows: 100 Fetched Rows: 10 Matching

#	cedula	nombres	direccion	fecha_nacimi	cargas_familiares	salario	apellido
1	0106836976	Adolfo Sebastian	Remigio Tamariz	2000-07-08	0	1000.00	Jara
2	0106836987	Juan Sebastian	El valle	1993-08-11	1	2000.00	Rivera
3	0105536987	Martin	Yunguilla	1977-12-11	4	1500.63	Martinez
4	0105534422	Israel	Ordoñez Lasso	1982-12-31	2	2500355.00	Montoya
5	0106253422	Juan Jesus	10 de Agosto	2003-05-31	0	200.00	Loya
6	0106801222	Gabriela	Americas	2003-03-21	0	352.21	Maldonado
7	0106803698	Paola	Solano	1995-07-21	0	499.21	Hidalgo
8	0106815488	Agustina	Puertas del sol	2005-01-16	0	123.21	Vazquez
9	0106897410	Alejandra	Vista linda	1998-06-20	1	899.36	Cordero
10	0106898520	Mateo	Av. Loja	1996-09-26	2	1254.36	Salgado

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

```
select nombres, apellido
from persona
where salario>500
```

select nombres, apellido ... ×

Max. rows: 100 Fetched Rows: 6

#	nombres	apellido
1	Adolfo Sebastian	Jara
2	Juan Sebastian	Rivera
3	Martin	Martinez
4	Israel	Montoya
5	Alejandra	Cordero
6	Mateo	Salgado

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

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

select * from persona whe... ×

Max. rows: 100 Fetched Rows: 4

#	cedula	nombres	direccion	fecha_nacimiento	cargas_familiar	salario	apellido
1	0106836976	Adolfo Sebastian	Remigio Tamariz	2000-07-08	0	1000.00	Jara
2	0106803698	Paola	Solano	1995-07-21	0	499.21	Hidalgo
3	0106897410	Alejandra	Vista linda	1998-06-20	1	899.36	Cordero
4	0106898520	Mateo	Av. Loja	1996-09-26	2	1254.36	Salgado

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

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

select cedula, nombres fr... ×

Max. rows: 100 Fetche

#	cedula	nombres
1	0106836987	Juan Sebastian
2	0105536987	Martin
3	0105534422	Israel
4	0106803698	Paola
5	0106897410	Alejandra
6	0106898520	Mateo

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

```
select cedula, nombres
from persona
where apellido like 'A%'
```

select cedula, nombres fr... ×

Max. rows: 100 Fetched Row

#	cedula	nombres

- 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.

```
insert into persona_telefono
(codigo, cedula_fk, tipo_telefono, numero_telefono)
values
(1, '0106836976', 'Celular','0963204010')
```

```
insert into persona_telefono
(codigo, cedula_fk, tipo_telefono, numero_telefono)
values
(2, '0106836987', 'Convenciona','2455629')
```

```
insert into persona_telefono
(codigo, cedula_fk, tipo_telefono, numero_telefono)
values
(3, '0105536987', 'Convencional','2863512')
```

#	codigo	cedula_fk	tipo_telefono	numero_telefono
1	1	0106836976	Celular	0963204010
2	3	0105536987	Convencional	2863512
3	2	0106836987	Convencional	2455629

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

```
update persona
set salario=1000
where salario<1000
```

#	cedula	nombres	direccion	fecha_nacimiento	cargas_f	salario	apellido
1	0106836976	Adolfo Sebastian	Remigio Tamariz	2000-07-08	0	1000.00	Jara
2	0106836987	Juan Sebastian	El valle	1993-08-11	1	2000.00	Rivera
3	0105536987	Martin	Yunguilla	1977-12-11	4	1500.63	Martinez
4	0105534422	Israel	Ordoñez Lasso	1982-12-31	2	2500355.00	Montoya
5	0106898520	Mateo	Av. Loja	1996-09-26	2	1254.36	Salgado
6	0106253422	Juan Jesus	10 de Agosto	2003-05-31	0	1000.00	Loya
7	0106801222	Gabriela	Americas	2003-03-21	0	1000.00	Maldonado
8	0106803698	Paola	Solano	1995-07-21	0	1000.00	Hidalgo
9	0106815488	Agustina	Puertas del sol	2005-01-16	0	1000.00	Vazquez
10	0106897410	Alejandra	Vista linda	1998-06-20	1	1000.00	Cordero

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

```
update persona
set nombres=upper(nombres), apellido=upper(apellido)
```

#	nombres	apellido
1	ADOLFO SEBASTIAN	JARA
2	JUAN SEBASTIAN	RIVERA
3	MARTIN	MARTINEZ
4	ISRAEL	MONTOYA
5	MATEO	SALGADO
6	JUAN JESUS	LOYA
7	GABRIELA	MALDONADO
8	PAOLA	HIDALGO
9	AGUSTINA	VAZQUEZ
10	ALEJANDRA	CORDERO

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

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

#	cedula	nombres	direccion	fecha_nacimiento	cargas_familiares	salario	apellido
1	0106836976	ADOLFO SEBASTIAN	Remigio Tamariz	2000-07-08	0	1000.00	JARA
2	0106836987	JUAN SEBASTIAN	El valle	1993-08-11	1	2000.00	RIVERA
3	0105536987	MARTIN	Yunguilla	1977-12-11	4	1500.63	MARTINEZ
4	0106898520	MATEO	Av. Loja	1996-09-26	2	1254.36	SALGADO
5	0106253422	JUAN JESUS	10 de Agosto	2003-05-31	0	1000.00	LOYA
6	0106801222	GABRIELA	Americas	2003-03-21	0	1000.00	MALDONADO
7	0106803698	PAOLA	Solano	1995-07-21	0	1000.00	HIDALGO
8	0106815488	AGUSTINA	Puertas del sol	2005-01-16	0	1000.00	VAZQUEZ
9	0106897410	ALEJANDRA	Vista linda	1998-06-20	1	1000.00	CORDERO

2.4 FOREIGN KEY

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

```
select p.nombres, p.apellido, t.*
from persona as p, persona_telefono as t
```

1	ADOLFO SEBASTIAN	JARA	1	0106836976	Celular	0963204010
2	JUAN SEBASTIAN	RIVERA	1	0106836976	Celular	0963204010
3	MARTIN	MARTINEZ	1	0106836976	Celular	0963204010
4	MATEO	SALGADO	1	0106836976	Celular	0963204010
5	JUAN JESUS	LOYA	1	0106836976	Celular	0963204010
6	GABRIELA	MALDONADO	1	0106836976	Celular	0963204010
7	PAOLA	HIDALGO	1	0106836976	Celular	0963204010
8	AGUSTINA	VAZQUEZ	1	0106836976	Celular	0963204010
9	ALEJANDRA	CORDERO	1	0106836976	Celular	0963204010
10	ADOLFO SEBASTIAN	JARA	3	0105536987	Convencional	2863512
11	JUAN SEBASTIAN	RIVERA	3	0105536987	Convencional	2863512
12	MARTIN	MARTINEZ	3	0105536987	Convencional	2863512
13	MATEO	SALGADO	3	0105536987	Convencional	2863512
14	JUAN JESUS	LOYA	3	0105536987	Convencional	2863512
15	GABRIELA	MALDONADO	3	0105536987	Convencional	2863512
16	PAOLA	HIDALGO	3	0105536987	Convencional	2863512
17	AGUSTINA	VAZQUEZ	3	0105536987	Convencional	2863512
18	ALEJANDRA	CORDERO	3	0105536987	Convencional	2863512
19	ADOLFO SEBASTIAN	JARA	2	0106836987	Convencional	2455629
20	JUAN SEBASTIAN	RIVERA	2	0106836987	Convencional	2455629
21	MARTIN	MARTINEZ	2	0106836987	Convencional	2455629
22	MATEO	SALGADO	2	0106836987	Convencional	2455629
23	JUAN JESUS	LOYA	2	0106836987	Convencional	2455629
24	GABRIELA	MALDONADO	2	0106836987	Convencional	2455629
25	PAOLA	HIDALGO	2	0106836987	Convencional	2455629
26	AGUSTINA	VAZQUEZ	2	0106836987	Convencional	2455629
27	ALEJANDRA	CORDERO	2	0106836987	Convencional	2455629

```
1  select p.nombres, p.apellido, t.*
2  from persona as p, persona_telefono as t
3  where p.cedula = t.cedula_fk
```

select p.nombres, p.apell... ×

Max. rows: 100 Fetched Rows: 3

#	nombres	apellido	codigo	cedula_fk	tipo_telefono	numero_telefono
1	ADOLFO SEBASTIAN	JARA	1	0106836976	Celular	0963204010
2	MARTIN	MARTINEZ	3	0105536987	Convencional	2863512
3	JUAN SEBASTIAN	RIVERA	2	0106836987	Convencional	2455629

Se crearon más teléfonos:

```
1  select p.nombres, p.apellido, t.*
2  from persona as p, persona_telefono as t
3  where p.cedula=t.cedula_fk
4
```

select p.nombres, p.apell... ×

Max. rows: 100 Fetched Rows: 9

#	nombres	apellido	codigo	cedula_fk	tipo_telefono	numero_telefono
1	ADOLFO SEBASTIAN	JARA	1	0106836976	Celular	0963204010
2	MARTIN	MARTINEZ	3	0105536987	Convencional	2863512
3	JUAN SEBASTIAN	RIVERA	2	0106836987	Convencional	2455629
4	MATEO	SALGADO	4	0106898520	Celular	0965413248
5	JUAN JESUS	LOYA	5	0106253422	Celular	09654135888
6	PAOLA	HIDALGO	7	0106803698	Convencional	2520147
7	GABRIELA	MALDONADO	6	0106801222	Convencional	2896321
8	AGUSTINA	VAZQUEZ	8	0106815488	Celular	0998841165
9	ALEJANDRA	CORDERO	9	0106897410	Celular	09652548741

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

```
select p.*, t.numero_telefono
from persona as p, persona_telefono as t
where p.cedula=t.cedula_fk and p.salario=1000
```

select p.*, t.numero_tele... ×

Max. rows: 100 Fetched Rows: 6

#	cedula	nombres	direccion	fecha_nacimiento	cargas_fa	salario	apellido	numero_telefono
1	0106836976	ADOLFO SEBASTIAN	Remigio Tamariz	2000-07-08	0	1000.00	JARA	0963204010
2	0106253422	JUAN JESUS	10 de Agosto	2003-05-31	0	1000.00	LOYA	09654135888
3	0106803698	PAOLA	Solano	1995-07-21	0	1000.00	HIDALGO	2520147
4	0106801222	GABRIELA	Americas	2003-03-21	0	1000.00	MALDONADO	2896321
5	0106815488	AGUSTINA	Puertas del sol	2005-01-16	0	1000.00	VAZQUEZ	0998841165
6	0106897410	ALEJANDRA	Vista linda	1998-06-20	1	1000.00	CORDERO	09652548741

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

```
select numero_telefono from persona_telefono
```

select numero_telefono fr... ×

Max. rows: 100 Fetched Rows:

#	numero_telefono
1	0963204010
2	2863512
3	2455629
4	0965413248
5	09654135888
6	2520147
7	2896321
8	0998841165
9	09652548741

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

```
select p.nombres, p.apellido
from persona as p, persona_telefono as t
where p.cedula=t.cedula_fk and t.tipo_telefono='Convencional'
```

select p.nombres, p.apell... ×

Max. rows: 100 Fetched Rows: 4

#	nombres	apellido
1	JUAN SEBASTIAN	RIVERA
2	MARTIN	MARTINEZ
3	GABRIELA	MALDONADO
4	PAOLA	HIDALGO

3. Probar y modificar datos utilizando el IDE.

RESULTADO(S) OBTENIDO(S):

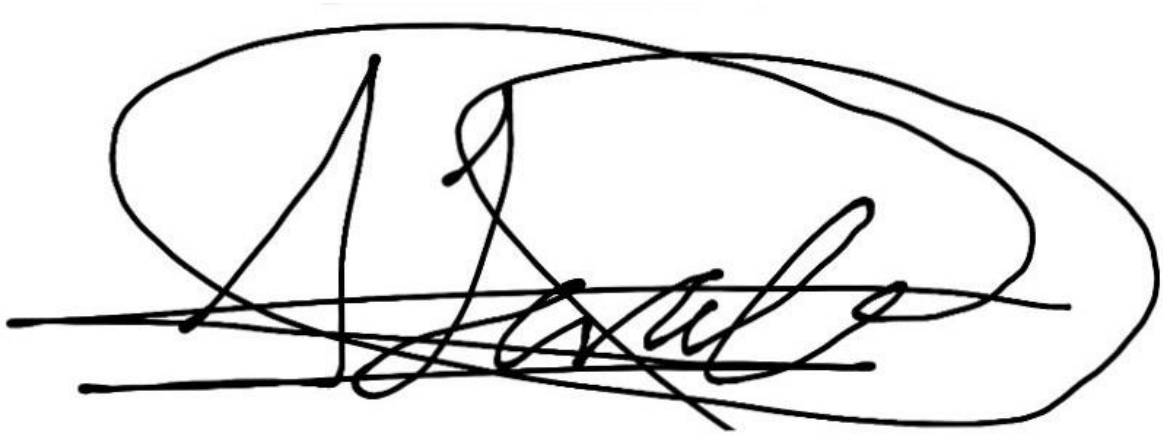
- Familiarizarse con los comandos de DDL y DML.
- Aprender a manipular bases de datos desde la línea de comandos.
- Investigar sobre algunos comandos.
- Leer sobre bases de datos.

CONCLUSIONES:

En conclusión, está practica ha sido muy útil para lograr practicar claramente sobre las bases de datos y así poder aprender los distintos comandos que nos ofrece el DDL y el DML. En está practica aprendí mucho sobre la manipulación de base de datos las cuales nos sirven mucho en un futuro como programadores.

RECOMENDACIONES:

No existe ninguna recomendación.

***Nombre de estudiante*:** Adolfo Sebastián Jara Gavilanes.

***Firma de estudiante*:**