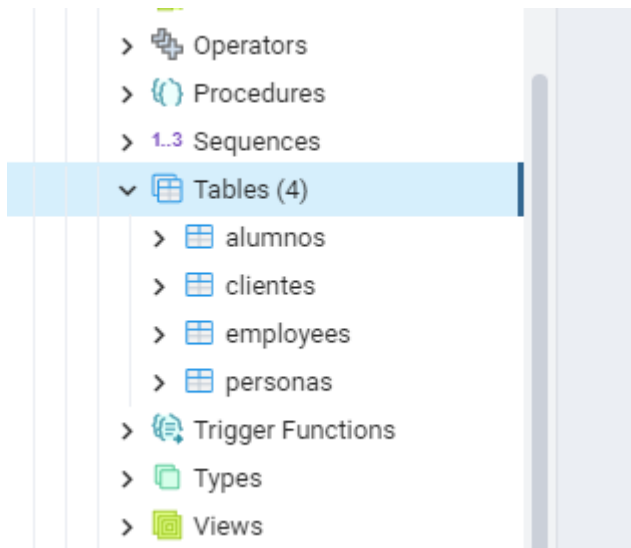


Ejercicios con Postresql , PGAdmin y Base de datos



- Creacion de una base de datos llamada Curso

Creacion de una tabla llamada personas

```
CREATE DATABASE curso
```

```
CREATE TABLE personas(  
  dni VARCHAR(8),  
  nombre VARCHAR(45),  
  apellidopaterno VARCHAR(45),  
  apellidomaterno VARCHAR(45),  
  fechanacimiento DATE,  
  municipio VARCHAR(45),  
  estado VARCHAR(45),  
  telefono VARCHAR(12),  
  sexo CHARACTER(1)  
);
```

Luego, ejecución para creación de la tabla mediante la instrucción “CREATE TABLE” (ejecutando play) , una vez ejecutada, analizar el return del código.

Una vez creada ya vamos a poder analizar y retornar algún dato, por ejemplo

Con `SELECT*FROM personas (nombre de la tabla)`

El código (explicarlo MUY detalladamente línea por línea , cada detalle, es el siguiente)

```
CREATE TABLE personas(  
dni VARCHAR(8),  
nombre VARCHAR(45),  
apellidopaterno VARCHAR(45),  
apellidomaterno VARCHAR(45),  
fechanacimiento DATE,  
municipio VARCHAR(45),  
estado VARCHAR(45),  
telefono VARCHAR(12),  
sexo CHARACTER(1)  
);
```

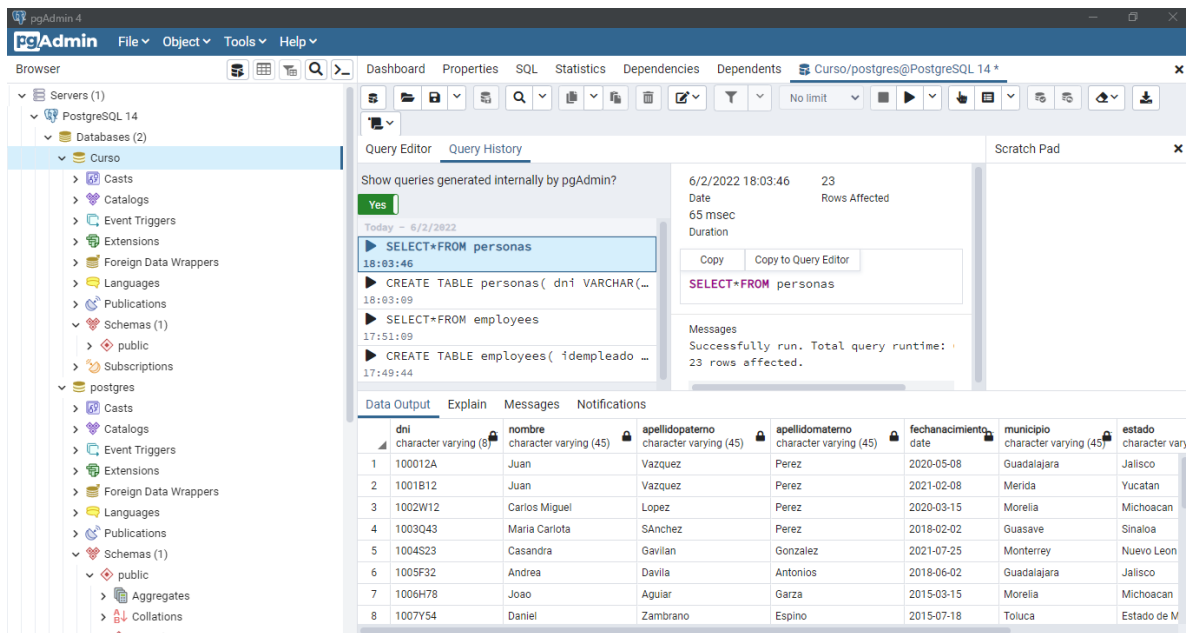
`INSERT INTO`

`personas(dni,nombre,apellidopaterno,apellidomaterno,fechanacimiento,municipio,estado,telefono,sexo)`

`VALUES('100012A','Juan','Vazquez','Perez','2020-05-08','Guadalajara','Jalisco','1234564343','M'),`
`('1001B12','Juan','Vazquez','Perez','2021-02-08','Merida','Yucatan','1234564348','M'),`

('1002W12','Carlos Miguel','Lopez','Perez','2020-03-15','Morelia','Michoacan','1234564349','M'),
('1003Q43','Maria Carlota','SAnchez','Perez','2018-02-02','Guasave','Sinaloa','1234564322','F'),
('1004S23','Casandra','Gavilan','Gonzalez','2021-07-25','Monterrey','Nuevo Leon','1234564335','F'),
('1005F32','Andrea','Davila','Antonios','2018-06-02','Guadalajara','Jalisco','1234564326','F'),
('1006H78','Joao','Aguiar','Garza','2015-03-15','Morelia','Michoacan','1234564327','M'),
('1007Y54','Daniel','Zambrano','Espino','2015-07-18','Toluca','Estado de Mexico','1234564328','M'),
('1008W43','Flor','Velazquez','Espinoza','2017-02-08','Guadalajara','Jalisco','1234564345','F'),
('1009W23','Celeste','Vazquez','De la O','2015-08-22','Tijuana','Baja California','1234564385','F'),
('1010W12','Abigail','Andrade','Beltran','2020-05-12','Guadalajara','Jalisco','1234564373','F'),
('1011Q25','Juan Carlos','Espinoza','Campos','2020-05-15','Tijuana','Baja California','1234564399','M'),
('1012W25','Dionicio','Espino','Espinoza','2021-06-15','Guadalajara','Jalisco','1234564398','F'),
('1013Q45','Jose Carlos','Flores','Garcia','2021-07-17','Guasave','Sinaloa','1234564390','M'),
('1014Y59','Jose Pedro','Valle','Perez','2021-08-01','Mazatlan','Sinaloa','1234564312','M'),
('1015P45','Miguel Luis','Flores','Sanchez','2021-01-15','Monterrey','Nuevo Leon','1234564315','M'),
('1016H89','JoseMarcelo','Gonzalez','Miranda','2018-05-12','Guadalajara','Jalisco','1234564222','M'),
('1017278','Flor Estela','Huerta','Espinosa','2018-07-09','Guadalajara','Jalisco','1234564555','F'),
('1018Q34','Cristian Jesus','Kilberth','Perez','2018-01-08','Ensenada','Baja California','1234564532','F'),
('1019W47','Maria Cecilia','Lopez','Lopez','2045-07-28','Guadalajara','Jalisco','1234564145','F'),
('1020P30','Juan Alberto','Martinez','Vazquez','2016-07-22','Guadalajara','Jalisco','1234564142','M'),
('1021Q25','Franchesco Daniel','Nunez','Perez','2017-07-15','Puerto Vallarta','Jalisco','1234564248','M'),
('1022F78','Laura','Quinonez','Garcia','2020-02-08','Puebla','Puebla','1234564788','F');

Ahora si ejecutamos un `SELECT*FROM personas`, deberíamos recibir todo lo cargado con `INSERT INTO personas`



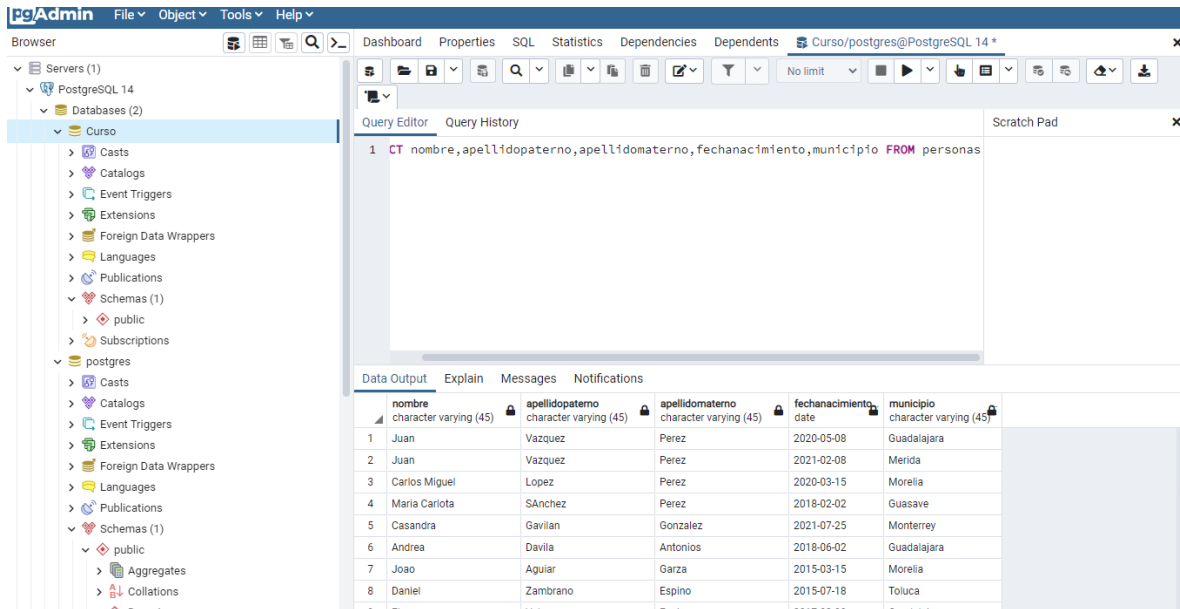
FILTREMOS AHORA POR QUERY

Si queremos ver solo el nombre completo y el municipio

- = significa ver todo

`SELECT nombre,apellidopaterno,apellidomaterno,fechanacimiento,municipio FROM personas`

Deberíamos poder visualizar solo el nombre completo y municipio y nacimiento



APLICAR VARIACIONES AQUÍ PARA DEMOSTRAR EL PUNTO

TIPOS DE DATOS EN POSTRGRESQL

Introducción

PostgreSQL tiene un amplio conjunto de tipos de datos nativos disponibles para los usuarios. Los usuarios pueden agregar nuevos tipos a PostgreSQL usando el comando CREAM TIPO.

<https://www.postgresql.org/docs/9.6/static/datatype.html>

Tipos numericos

| Nombre | Tamaño de almacenamiento | Descripción | Distancia |
|------------------|--------------------------|---|---|
| smallint | 2 bytes | entero de rango pequeño | -32768 a +32767 |
| integer | 4 bytes | elección típica para entero | -2147483648 a +2147483647 |
| bigint | 8 bytes | entero de gran rango | -9223372036854775808 a +9223372036854775807 |
| decimal | variable | precisión especificada por el usuario, exacta | hasta 131072 dígitos antes del punto decimal; hasta 16383 dígitos después del punto decimal |
| numeric | variable | precisión especificada por el usuario, exacta | hasta 131072 dígitos antes del punto decimal; hasta 16383 dígitos después del punto decimal |
| real | 4 bytes | precisión variable, inexacta | Precisión de 6 dígitos decimales |
| double precision | 8 bytes | precisión variable, inexacta | Precisión de 15 dígitos decimales |
| smallserial | 2 bytes | pequeño entero autoincremento | 1 a 32767 |
| serial | 4 bytes | autoincremento entero | 1 a 2147483647 |
| bigserial | 8 bytes | entero autoincremento grande | 1 a 9223372036854775807 |
| int4range | | Rango de enteros | |
| int8range | | Gama de bigint | |
| numrange | | Rango de numérico | |

Tipos de fecha / hora

| Nombre | Tamaño de almacenamiento | Descripción | Bajo valor | Alto valor | Resolución |
|------------------------------|--------------------------|---------------------------------|------------|------------|------------------------------|
| timestamp (sin zona horaria) | 8 bytes | fecha y hora (sin zona horaria) | 4713 aC | 294276 dC | 1 microsegundo 14 dígitos |
| timestamp (con zona horaria) | 8 bytes | fecha y hora, con zona horaria | 4713 aC | 294276 dC | 1 microsegundo 14 dígitos |
| date | 4 bytes | fecha (sin hora del día) | 4713 aC | 5874897 dC | 1 día |

| Nombre | Tamaño de almacenamiento | Descripción | Bajo valor | Alto valor | Resolución |
|-------------------------|--------------------------|---|-------------------|-----------------|------------------------------|
| time (sin zona horaria) | 8 bytes | hora del día (sin fecha) | 00:00:00 | 24:00:00 | 1 microsegundo 14 dígitos |
| time (con zona horaria) | 12 bytes | solo horas del día, con zona horaria | 00: 00: 00 + 1459 | 24: 00: 00-1459 | 1 microsegundo 14 dígitos |
| interval | 16 bytes | intervalo de tiempo | -178000000 años | 178000000 años | 1 microsegundo 14 dígitos |
| tsrange | | rango de marca de tiempo sin zona horaria | | | |
| tstzrange | | rango de marca de tiempo con zona horaria | | | |
| daterange | | rango de fecha | | | |

Tipos geometricos

| Nombre | Tamaño de almacenamiento | Descripción | Representación |
|---------|--------------------------|--------------------------------------|------------------------------------|
| point | 16 bytes | Punto en un avion | (x, y) |
| line | 32 bytes | Linea infinita | {A B C} |
| lseg | 32 bytes | Segmento de línea finita | ((x1, y1), (x2, y2)) |
| box | 32 bytes | Caja rectangular | ((x1, y1), (x2, y2)) |
| path | 16 + 16n bytes | Ruta cerrada (similar a polígono) | ((x1, y1), ...) |
| path | 16 + 16n bytes | Camino abierto | [(x1, y1), ...] |
| polygon | 40 + 16n bytes | Polígono (similar a la ruta cerrada) | ((x1, y1), ...) |
| circle | 24 bytes | Circulo | <(x, y), r> (punto central, radio) |

Tipos de direcciones de red

| Nombre | Tamaño de almacenamiento | Descripción |
|--------|--------------------------|---------------------------|
| cidr | 7 o 19 bytes | Redes IPv4 e IPv6 |
| inet | 7 o 19 bytes | IPv4 e IPv6 hosts y redes |

| Nombre | Tamaño de almacenamiento | Descripción |
|---------|--------------------------|-----------------|
| macaddr | 6 bytes | Direcciones MAC |

Tipos de personajes

| Nombre | Descripción |
|-----------------------------------|------------------------------------|
| character varying(n) , varchar(n) | longitud variable con límite |
| character(n) , char(n) | longitud fija, acolchado en blanco |
| text | longitud ilimitada variable |

Arrays

En PostgreSQL puede crear matrices de cualquier tipo integrado, definido por el usuario o enumeración. De forma predeterminada, no hay límite para una matriz, pero *puede* especificarla.

Declarando una matriz

```
SELECT integer[];
SELECT integer[3];
SELECT integer[][];
SELECT integer[3][3];
SELECT integer ARRAY;
SELECT integer ARRAY[3];
```

Creando un Array

```
SELECT '{0,1,2}';
SELECT '{{0,1},{1,2}}';
SELECT ARRAY[0,1,2];
SELECT ARRAY[ARRAY[0,1],ARRAY[1,2]];
```

-----*

Probemos ahora con otra tabla, la tabla alumnos, la cual podemos crear con

```
CREATE TABLE alumnos(
```



```

numcontrol BIGINT,
nombre VARCHAR(45),
apellidopaterno VARCHAR(45),
apellidomaterno VARCHAR(45),
fechanacimiento DATE,
fechaingreso DATE,
telefono VARCHAR(12),
carrera VARCHAR(45),
promedio DECIMAL(6,2)
);

```

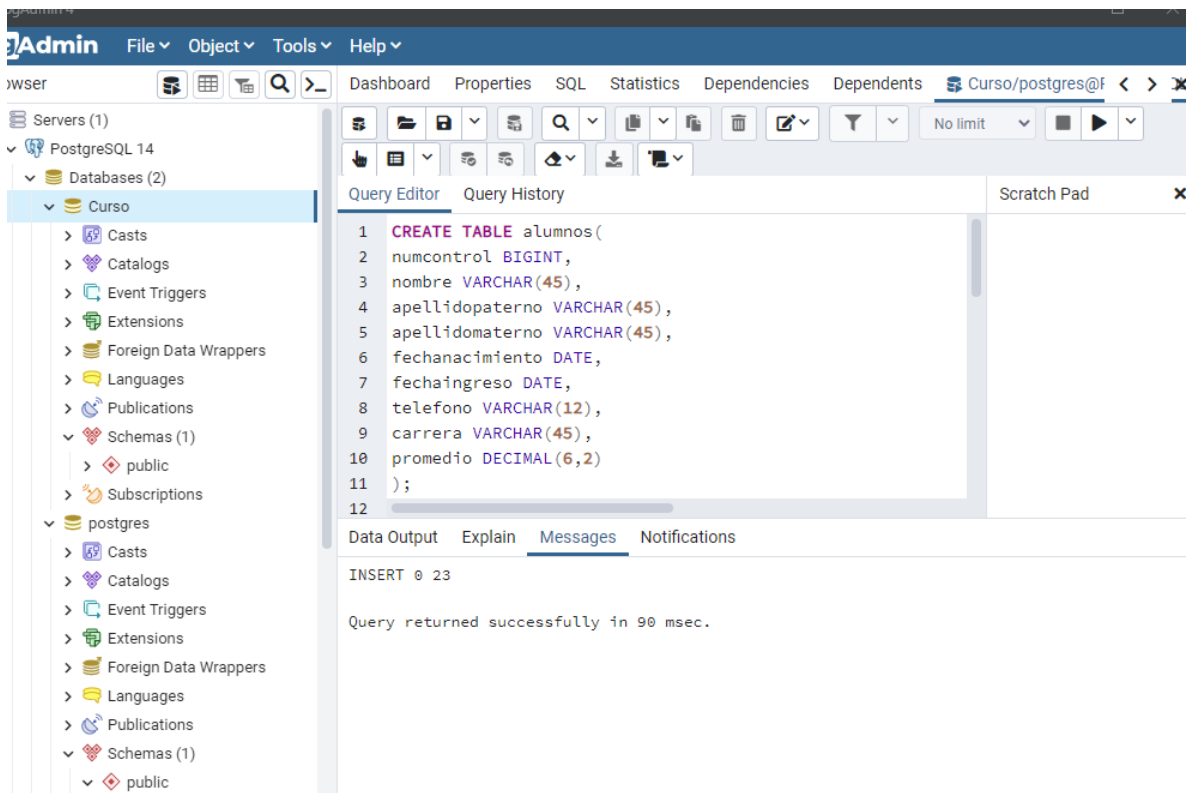
```

INSERT INTO
alumnos(numcontrol,nombre,apellidopaterno,apellidomaterno,fechanacimiento,fechaingreso,telefono,carrera,promedio)
VALUES(1000,'Juan','Vazquez','Perez','2016-05-08','1992-02-18',
'1234564343','Mercadotecnia',95.92),(1001,'Juan','Vazquez','Perez','2018-02-08','1991-05-12',
'1234564348','Ingenieria en Sistemas',85.2),(1002,'Carlos
Miguel','Lopez','Perez','2017-03-15','1989-07-15',
'1234564349','Fisico Matematico',90),(1003,'Maria
Carlota','SAnchez','Perez','2018-02-02','1995-08-17',
'1234564322','Psicologia',95.8),(1004,'Casandra','Gavilan','Gonzalez','2018-07-25','1995-06-18',
'1234564335','Matematicas
puras',89.2),(1005,'Andrea','Davila','Antonios','2018-06-02','1996-06-21',
'1234564326','Biologia',99),(1006,'Joao','Aguiar','Garza','2015-03-15','1993-03-28',
'1234564327','Ingenieria
Mecatronica',92.4),(1007,'Daniel','Zambrano','Espino','2015-07-18','1996-02-05',
'1234564328','Robotica',100),(1008,'Flor','Velazquez','Espinoza','2017-02-08','1995-05-07',
'1234564345','Contabilidad',100),(1009,'Celeste','Vazquez','De la O','2015-08-22','1995-01-31',
'1234564385','Bionica',99.9),(1010,'Abigail','Andrade','Beltran','2020-05-12','1999-01-17',

```

'1234564373','Fisico Matematico',85.2),(1011,'Juan Carlos','Espinoza','Campos','2020-05-15','1995-01-15',
'1234564399','Fisico Matematico',100),(1012,'Dionicio','Espino','Espinoza','2021-06-15','2003-05-12',
'1234564398','Ingenieria en Sistemas',85),(1013,'Jose Carlos','Flores','Garcia','2021-07-17','2003-08-12',
'1234564390','Contabilidad',82.9),(1014,'Jose Pedro','Valle','Perez','2021-08-01','2002-06-25',
'1234564312','Ingenieria Quimica',84.2),(1015,'Miguel Luis','Flores','Sanchez','2021-01-15','2002-05-28',
'1234564315','Fisico Matematico',92.5),(1016,'Jose Marcelo','Gonzalez','Miranda','2018-05-12','1998-06-25',
'1234564222','Mercadotecnia',75.9),(1017,'Flor Estela','Huerta','Espinosa','2018-07-09','1998-06-25',
'1234564555','Fisica',91.1),(1018,'Cristian Jesus','Kilberth','Perez','2018-01-08','1998-02-01',
'1234564532','Artes',94.2),(1019,'Maria Cecilia','Lopez','Lopez','2020-07-28','2003-05-25',
'1234564145','Psicologia',85.6),(1020,'Juan Alberto','Martinez','Vazquez','2016-07-22','1998-02-25',
'1234564142','Medicina',80.2),(1021,'Franchesco Daniel','Nunez','Perez','2017-07-15','1995-09-02',
'1234564248','Medicina',99.2),(1022,'Laura','Quinonez','Garcia','2020-02-08','2003-05-04',
'1234564788','Medicina',95.2);

Creemos la tabla



Ahora, algunos comandos

CLAUSULA WHERE

SELECT * FROM alumnos WHERE promedio >= 90 (solo mostrara a los alumnos de la tabla alumnos donde el promedio sea mayor a 90)

pgAdmin 4

gAdmin File Object Tools Help

Browser Dashboard Properties SQL Statistics Dependencies Dependents Curso/postgres@f

Servers (1)

- PostgreSQL 14
 - Databases (2)
 - Curso
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Subscriptions
 - postgres
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates

Query Editor Query History Scratch Pad

1 SELECT * FROM alumnos WHERE promedio >= 90

Data Output Explain Messages Notifications

| apellidomaterno | fechanacimiento | fechaingreso | telefono | carrera | promedio |
|------------------------|-----------------|--------------|------------------------|------------------------|---------------|
| character varying (45) | date | date | character varying (10) | character varying (45) | numeric (6,2) |
| Perez | 2016-05-08 | 1992-02-18 | 1234564343 | Mercadotecnia | 95.92 |
| Perez | 2017-03-15 | 1989-07-15 | 1234564349 | Fisico Matematico | 90.00 |
| Perez | 2018-02-02 | 1995-08-17 | 1234564322 | Psicologia | 95.80 |
| Antonios | 2018-06-02 | 1996-06-21 | 1234564326 | Biologia | 99.00 |
| Garza | 2015-03-15 | 1993-03-28 | 1234564327 | Ingenieria Mecatronica | 92.40 |
| Espino | 2015-07-18 | 1996-02-05 | 1234564328 | Robotica | 100.00 |
| Espinoza | 2017-02-08 | 1995-05-07 | 1234564345 | Contabilidad | 100.00 |
| De la O | 2015-08-22 | 1995-01-21 | 1234564385 | Quimica | 99.90 |

También

SELECT nombre,apellidopaterno,apellidomaterno,carrera,promedio FROM alumnos

```
1  
2 SELECT nombre,apellidopaterno,apellidomaterno,carrera,promedio FROM alumnos
```

Scratch Pad

Data Output Explain Messages Notifications

| | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | carrera character varying (45) | promedio numeric (6,2) | |
|----|----------------------------------|---|---|-----------------------------------|---------------------------|--|
| 1 | Juan | Vazquez | Perez | Mercadotecnia | 95.92 | |
| 2 | Juan | Vazquez | Perez | Ingenieria en Sistemas | 85.20 | |
| 3 | Carlos Miguel | Lopez | Perez | Fisico Matematico | 90.00 | |
| 4 | Maria Carlota | SAnchez | Perez | Psicologia | 95.80 | |
| 5 | Casandra | Gavilan | Gonzalez | Matematicas puras | 89.20 | |
| 6 | Andrea | Davila | Antonios | Biologia | 99.00 | |
| 7 | Joao | Aguiar | Garza | Ingenieria Mecatronica | 92.40 | |
| 8 | Daniel | Zambrano | Espino | Robotica | 100.00 | |
| 9 | Flor | Velazquez | Espinoza | Contabilidad | 100.00 | |
| 10 | Celeste | Vazquez | De la O | Bionica | 99.90 | |
| 11 | Abigail | Andrade | Beltran | Fisico Matematico | 85.20 | |

Y si le agregamos una clausula WHERE

```
SELECT nombre,apellidopaterno,apellidomaterno,carrera,promedio FROM alumnos WHERE  
promedio>=90
```

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Servers' tree is expanded to show a PostgreSQL 14 server with a database named 'Curso'. The 'Query Editor' tab is active, displaying the following SQL query:

```
SELECT nombre, apellidopaterno, apellidomaterno, carrera, promedio FROM alumnos WHERE
```


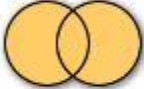
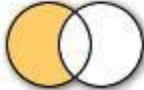
Below the query editor, the 'Data Output' tab shows the results of the query as a table with 5 columns: nombre, apellidopaterno, apellidomaterno, carrera, and promedio. The table contains 12 rows of data.

| nombre | apellidopaterno | apellidomaterno | carrera | promedio |
|---------------|-----------------|-----------------|------------------------|----------|
| Juan | Vazquez | Perez | Mercadotecnia | 95.92 |
| Carlos Miguel | Lopez | Perez | Fisico Matematico | 90.00 |
| Maria Carlota | Sanchez | Perez | Psicologia | 95.80 |
| Andrea | Davila | Antonios | Biologia | 99.00 |
| Joao | Aguilar | Garza | Ingenieria Mecatronica | 92.40 |
| Daniel | Zambrano | Espino | Robotica | 100.00 |
| Flor | Velazquez | Espinoza | Contabilidad | 100.00 |
| Celeste | Vazquez | De la O | Bionica | 99.90 |
| Juan Carlos | Espinoza | Campos | Fisico Matematico | 100.00 |
| Miguel Luis | Flores | Sanchez | Fisico Matematico | 92.50 |

PROBEMOS BOOLEANOS EN POSTGRES

La lógica Booleana toma su nombre del matemático británico George Boole (1815-1864) quien escribió acerca de un sistema de lógica diseñado para producir mejores resultados de búsqueda al formular demandas precisas. Lo llamó el “cálculo del pensamiento”

Los operadores booleanos forman la base de los conjuntos matemáticos y la lógica para la búsqueda en las bases de datos. Los operadores conectan sus palabras de búsqueda para estrechar o ampliar los resultados. Los tres operadores booleanos básicos son: AND, OR y NOT.

| Boolean Operator | Examples | Retrieves |
|------------------|---|---|
| AND | children and television rodgers AND hammerstein children AND poverty |  Retrieves records containing both terms |
| OR | television or television viewing sixties OR 60s OR 1960s labor OR labour |  Retrieves records containing either one or both terms |
| NOT | television not movies caribbean NOT cuba s1 NOT s2 |  Excludes records containing the second term |

Ahora, utilizaremos un poco de algebra booleana

```
SELECT nombre,apellidopaterno,apellidomaterno,carrera,promedio FROM alumnos WHERE
promedio >=90 AND carrera='Psicologia'
```

Dashboard Properties SQL Statistics Dependencies Dependents Curso/postgres@PostgreSQL 14 *

Query Editor Query History

```
2  
3  
4 SELECT nombre,apellidopaterno,apellidomaterno,carrera,promedio FROM alumnos WHERE promedio >=90 AND carrera='Psicologia'  
5
```







Scratch Pad

Data Output Explain Messages Notifications

| | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | carrera character varying (45) | promedio numeric (6,2) |
|---|----------------------------------|---|---|-----------------------------------|---------------------------|
| 1 | Maria Carlota | SAnchez | Perez | Psicologia | 95.80 |

Que podemos ver?

Un SOLO RESULTADO

| Data Output | | Explain | Messages | Notifications | |
|---|---|--|--|--|--|
|  | nombre character varying (45)  | apellidopaterno character varying (45)  | apellidomaterno character varying (45)  | carrera character varying (45)  | promedio numeric (6,2)  |
| 1 | Maria Carlota | Sánchez | Perez | Psicologia | 95.80 |

Y esto es por el OPERADOR AND, que obliga a que ambas condiciones (WHERE y AND) sean verdaderas, para mostrar el dato, por lo tanto solo un alumno cumple con ambas Verdades!

Probemos ahora Invirtiendo el símbolo de MAX y MIN

```
SELECT nombre,apellidopaterno,apellidomaterno,carrera,promedio FROM alumnos WHERE  
promedio < 90 AND carrera='Medicina'
```


| Data Output | | | | | | Explain | Messages | Notifications |
|-------------|----------------------------------|---|---|-----------------------------------|---------------------------|---------|----------|---------------|
| | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | carrera character varying (45) | promedio numeric (6,2) | | | |
| 1 | Juan Alberto | Martinez | Vazquez | Medicina | 80.20 | | | |

Ahora, probemos el operador OR

EJECUTEMOS el siguiente query

SELECT*FROM alumnos WHERE carrera= 'Ingenieria en Sistemas' OR carrera= 'Psicologia'

The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree is expanded to show the 'Curso' database. The 'Query Editor' is active, displaying the following SQL query:

```
SELECT*FROM alumnos WHERE carrera= 'Ingenieria en Sistemas' OR carrera= 'Psicologia'
```

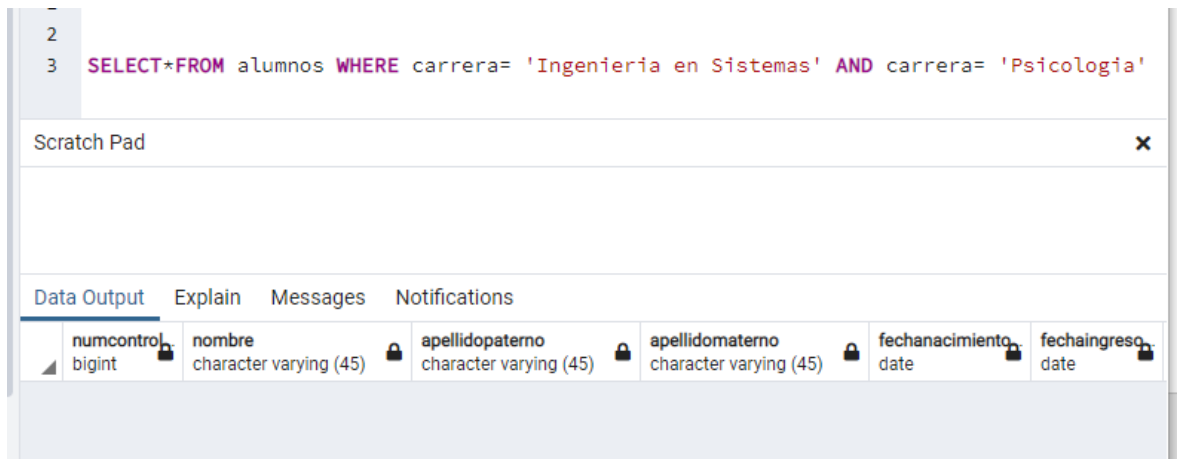
Below the query editor, the 'Data Output' tab is selected, showing the results of the query. The results are displayed in a table with the following columns: numcontrol, nombre, apellidopaterno, apellidomaterno, fechanacimiento, and fechaingreso.

| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date | fechaingreso date |
|---|----------------------|----------------------------------|---|---|-------------------------|----------------------|
| 1 | 1001 | Juan | Vazquez | Perez | 2018-02-08 | 1991-05-12 |
| 2 | 1003 | Maria Carlota | SAncchez | Perez | 2018-02-02 | 1995-08-17 |
| 3 | 1012 | Dionicio | Espino | Espinoza | 2021-06-15 | 2003-05-12 |
| 4 | 1019 | Maria Cecilia | Lopez | Lopez | 2020-07-28 | 2003-05-25 |

Veamos que sucede si intentamos romper la lógica

Ejecutemos la siguiente query y analizemos el error

SELECT*FROM alumnos WHERE carrera= 'Ingenieria en Sistemas' AND carrera= 'Psicologia'



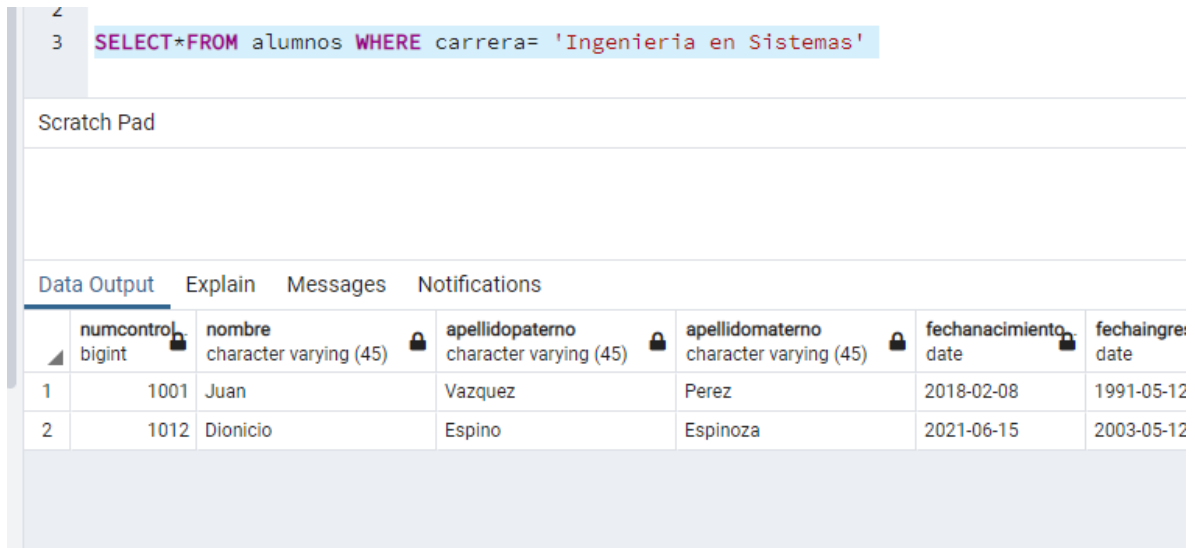
Que sucedió? Es simple, un alumno no puede cursar 2 Carreras, por lo tanto el operador AND no es el indicado aquí, ya que obliga que sean verdaderas dos condiciones excluyentes entre si, por lo tanto nunca podrá mostrar nada,

EL OPERADOR NOT

(where not)

Ejecutemos

SELECT*FROM alumnos WHERE carrera= 'Ingenieria en Sistemas'



The screenshot shows a SQL IDE interface. At the top, a query is entered in a text area: `SELECT*FROM alumnos WHERE carrera= 'Ingenieria en Sistemas'`. Below the text area is a 'Scratch Pad' section. At the bottom, there is a 'Data Output' tab which is active, displaying the results of the query in a table. The table has 7 columns: `numcontrol` (bigint), `nombre` (character varying (45)), `apellidopaterno` (character varying (45)), `apellidomaterno` (character varying (45)), `fechanacimiento` (date), and `fechaingre` (date). The results show two rows of data.

| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date | fechaingre date |
|---|----------------------|----------------------------------|---|---|-------------------------|--------------------|
| 1 | 1001 | Juan | Vazquez | Perez | 2018-02-08 | 1991-05-12 |
| 2 | 1012 | Dionicio | Espino | Espinoza | 2021-06-15 | 2003-05-12 |

Luego, intentemos negar la misma query, para que muestre aquellos resultados de aquellos que NO forman parte de la carrera de Ingenieria

Por ejemplo

SELECT*FROM alumnos WHERE NOT carrera= 'Ingenieria en Sistemas'

Query Editor

Query History

1

2

3

SELECT*FROM alumnos WHERE NOT carrera= 'Ingenieria en Sistemas'

Scratch Pad

Data Output

Explain

Messages

Notifications

| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date | fechaingre date |
|----|----------------------|----------------------------------|---|---|-------------------------|--------------------|
| 1 | 1000 | Juan | Vazquez | Perez | 2016-05-08 | 1992-02-18 |
| 2 | 1002 | Carlos Miguel | Lopez | Perez | 2017-03-15 | 1989-07-18 |
| 3 | 1003 | Maria Carlota | SA Sanchez | Perez | 2018-02-02 | 1995-08-17 |
| 4 | 1004 | Casandra | Gavilan | Gonzalez | 2018-07-25 | 1995-06-18 |
| 5 | 1005 | Andrea | Davila | Antonios | 2018-06-02 | 1996-06-21 |
| 6 | 1006 | Joao | Aguiar | Garza | 2015-03-15 | 1993-03-28 |
| 7 | 1007 | Daniel | Zambrano | Espino | 2015-07-18 | 1996-02-08 |
| 8 | 1008 | Flor | Velazquez | Espinoza | 2017-02-08 | 1995-05-07 |
| 9 | 1009 | Celeste | Vazquez | De la O | 2015-08-22 | 1995-01-31 |
| 10 | 1010 | Abigail | Andrade | Beltran | 2020-05-12 | 1999-01-17 |

Podemos ver como nos muestra aquellos resultados que NIEGAN la condiciones, es decir, nos muestra todo aquello que NO ES ingeniería en sistemas

AHORA TRABAJEMOS CON UNA NUEVA TABLA

LA TABLA CLIENTES

```
CREATE TABLE clientes(  
    idcliente INTEGER,  
    nombre VARCHAR(45),  
    apellidopaterno VARCHAR(45),  
    apellidomaterno VARCHAR(45),  
    telefono VARCHAR(12),  
    ciudad VARCHAR(45),  
    estado VARCHAR(45),  
    totaldeuda DECIMAL(8,2),  
    tipocliente CHARACTER(1)  
    );
```

```
INSERT INTO clientes  
VALUES(1000,'Alberto','Esparza','Suarez','2222222222','Mochis','Si  
naloa',8000,'A'),  
  
(1001,'Alberto','Gomez','Perez','2222222222','Mochis','Sinaloa',80  
00,'A'),  
  
(1002,'Juan  
Jose','Sanchez','Guerra','2222222222','Mazatlan','Sinaloa',15000,'  
A'),
```

(1003,'Maria
Antonia','Lopez','Gamez','2222222222','Guasave','Sinaloa',2500,'B'
)

(1004,'Luis
Roman','Espino','Garza','2222222222','Guadalajara','Jalisco',1000,'
A'),

(1005,'Ana
Estela','Juarez','Lima','2222222222','Mochis','Sinaloa',250.4,'C'),

(1006,'Jose','Esparza','Suarez','2222222222','Guadalajara','Jalisco',
800,'A'),

(1007,'Alberto','Flores','Gonzalez','2222222222','Tijuana','Baja
California',1500,'B'),

(1008,'Luis Pedro','Higo','Suarez','2222222222','Tijuana','Baja
California',1000,'A'),

(1009,'Jose
Carlos','Mendoza','Lopez','2222222222','Ensenada','Baja
California',4500,'A'),

(1010,'Maria
Laura','Puga','Gomez','2222222222','Mochis','Sinaloa',2000,'A'),

(1011,'Sasha','Lopez','Quintana','2222222222','Puerto
Vallarta','Jalisco',1500,'B'),

(1012,'Flor
Celeste','Ramos','Espino','2222222222','Veracruz','Veracruz',8000,'
A'),

(1013,'Jose
 Alberto','Sosa','Flores','2222222222','Acapulco','Guerrero',8000,'B'
),
 (1014,'Luis
 Miguel','Beltran','Gamez','2222222222','Mochis','Sinaloa',4000.2,'
 A'),
 (1014,'Daniel','Esparza','Suarez','2222222222','Mochis','Sinaloa',12
 000,'C');

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays the database structure: Servers (1) > PostgreSQL 14 > Databases (2) > Curso. The 'Query Editor' pane contains the following SQL code:

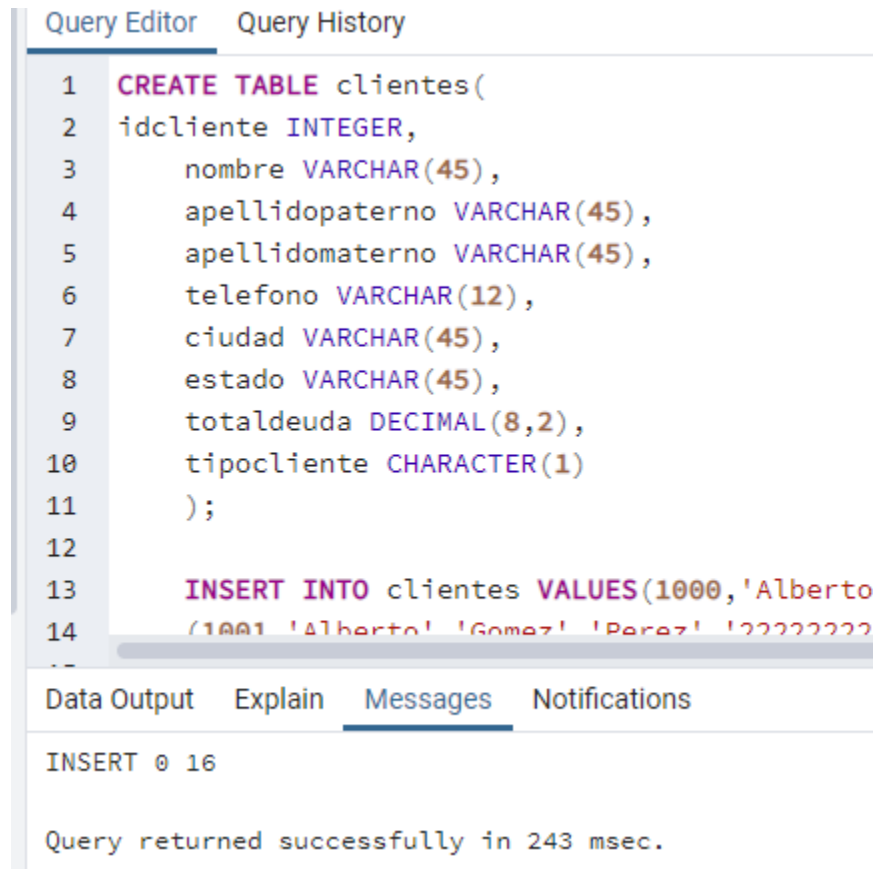
```

1 CREATE TABLE clientes(
2   idcliente INTEGER,
3   nombre VARCHAR(45),
4   apellidopaterno VARCHAR(45),
5   apellidomaterno VARCHAR(45),
6   telefono VARCHAR(12),
7   ciudad VARCHAR(45),
8   estado VARCHAR(45),
9   totaldeuda DECIMAL(8,2),
10  tipocliente CHARACTER(1)
11 );
12
13 INSERT INTO clientes VALUES(1000,'Alberto','Esparza','Suarez','2222222222','Mochi
14 /1001','Alberto','Gamez','Perez','2222222222','Mochis','Sinaloa',8000,'A')
  
```

Below the query editor, the 'Data Output' tab shows the results of the query. The table has 7 columns: numcontrol, nombre, apellidopaterno, apellidomaterno, fechanacimiento, and fechaingreso. The results are as follows:

| numcontrol | nombre | apellidopaterno | apellidomaterno | fechanacimiento | fechaingreso |
|------------|-------------------|-----------------|-----------------|-----------------|--------------|
| 16 | Flor Estela | Huerta | Espinosa | 2018-07-09 | 1998-06-25 |
| 17 | Cristian Jesus | Kilberth | Perez | 2018-01-08 | 1998-02-01 |
| 18 | Maria Cecilia | Lopez | Lopez | 2020-07-28 | 2003-05-25 |
| 19 | Juan Alberto | Martinez | Vazquez | 2016-07-22 | 1998-02-25 |
| 20 | Franchesco Daniel | Nunez | Perez | 2017-07-15 | 1995-09-02 |

TABLA CREADA



The screenshot shows a database query editor with two tabs: 'Query Editor' and 'Query History'. The 'Query Editor' tab is active and displays the following SQL code:

```
1 CREATE TABLE clientes(  
2   idcliente INTEGER,  
3   nombre VARCHAR(45),  
4   apellidopaterno VARCHAR(45),  
5   apellidomaterno VARCHAR(45),  
6   telefono VARCHAR(12),  
7   ciudad VARCHAR(45),  
8   estado VARCHAR(45),  
9   totaldeuda DECIMAL(8,2),  
10  tipocliente CHARACTER(1)  
11 );  
12  
13 INSERT INTO clientes VALUES(1000,'Alberto  
14 (1001 'Alberto' 'Gomez' 'Perez' '22222222
```

Below the query editor, there are four tabs: 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Messages' tab is active and displays the following output:

```
INSERT 0 16  
  
Query returned successfully in 243 msec.
```

Ahora, dentro de esta nueva tabla ejecutemos una query para conocer quienes tienen una deuda mayor a 10000 pesos

**SELECT * FROM clientes WHERE estado = 'Sinaloa' AND
totaldeuda >= 10000**


```
1 SELECT*FROM clientes WHERE estado = 'Sinaloa' AND totaldeuda>=10000
```

Data Output

Explain

Messages

Notifications

| | idcliente integer | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | telefono character vary | ciudad character varying (45) |
|---|----------------------|----------------------------------|---|---|----------------------------|----------------------------------|
| 1 | 1002 | Juan Jose | Sanchez | Guerra | 222222222 | Mazatlan |
| 2 | 1014 | Daniel | Esparza | Suarez | 222222222 | Mochis |

Y ahora si queremos podemos concatenar AND

```
SELECT*FROM clientes WHERE tipocliente='A' AND estado =  
'Sinaloa' AND totaldeuda>=10000
```

```
1 SELECT*FROM clientes WHERE tipocliente='A' AND estado = 'Sinaloa' AND totaldeuda>=10000
```

| Data Output | | Explain | Messages | Notifications | | |
|-------------|----------------------|----------------------------------|---|---|----------------------------|----------------------------------|
| | idcliente integer | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | telefono character vary | ciudad character varying (45) |
| 1 | 1002 | Juan Jose | Sanchez | Guerra | 2222222222 | Mazatlan |

Y si queremos filtrar por tipo de cliente?

Tipo de cliente A

```
SELECT*FROM clientes WHERE tipocliente='A' AND estado =  
'Sinaloa' AND totaldeuda>=10000
```

[Query Editor](#) [Query History](#)

```
1  SELECT*FROM clientes WHERE tipocliente='A' AND estado = 'Sinaloa' AND totaldeuda>=10000
```

| Data Output | Explain | Messages | Notifications |
|--|--|---|---|
| apellidomaterno character varying (45) | apellidomaterno character varying (45) | telefono character varying (45) | ciudad character varying (45) |
| | Guerra | 222222222 | Mazatlan |
| | | | Sinaloa |
| | | | 15000.00 |
| | | | A |

Tipo de Cliente B

```
SELECT*FROM clientes WHERE tipocliente='B' AND estado =  
        'Sinaloa' AND totaldeuda>=10000
```

Query Editor

Query History

1 SELECT*FROM clientes WHERE tipocliente='B' AND estado = 'Sinaloa'

Data Output

Explain

Messages

Notifications

| | idcliente integer | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | telefono character vary | ciudad character varying (45) |
|---|----------------------|----------------------------------|---|---|----------------------------|----------------------------------|
| 1 | 1003 | Maria Antonia | Lopez | Gamez | 222222222 | Guasave |

Tipo de Cliente C

SELECT*FROM clientes WHERE tipocliente='C' AND estado = 'Sinaloa' AND totaldeuda>=10000

Query Editor

Query History

1

SELECT*FROM clientes WHERE tipocliente='C' AND estado = 'Sinaloa'

Data Output

Explain

Messages

Notifications

| | idcliente integer | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | telefono character vary | ciudad character varying (45) |
|---|----------------------|----------------------------------|---|---|----------------------------|----------------------------------|
| 1 | 1005 | Ana Estela | Juarez | Lima | 2222222222 | Mochis |
| 2 | 1014 | Daniel | Esparza | Suarez | 2222222222 | Mochis |

OPERACIONES EN LA TABLA

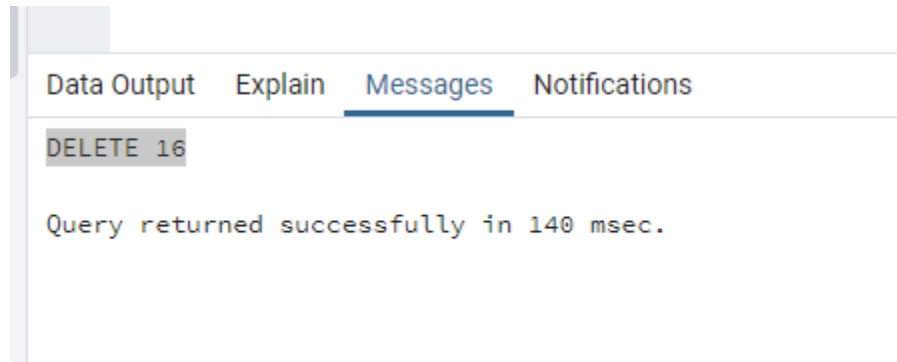
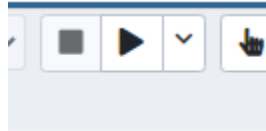
Eliminar registros de la tabla

```
1 DELETE FROM clientes
```

DELETE FROM clientes

Eliminaría toda la tabla

ejecutemoslo



**Y si ahora deseamos obtener algún dato
de la tabla, no existe**

Query Editor

Query History

1

`SELECT*FROM clientes`

Data Output

Explain

Messages

Notifications

| idcliente | nombre | apellidopaterno | apellidomaterno | telefono | ciudad |
|-----------|------------------------|------------------------|------------------------|----------------|------------------------|
| integer | character varying (45) | character varying (45) | character varying (45) | character vary | character varying (45) |

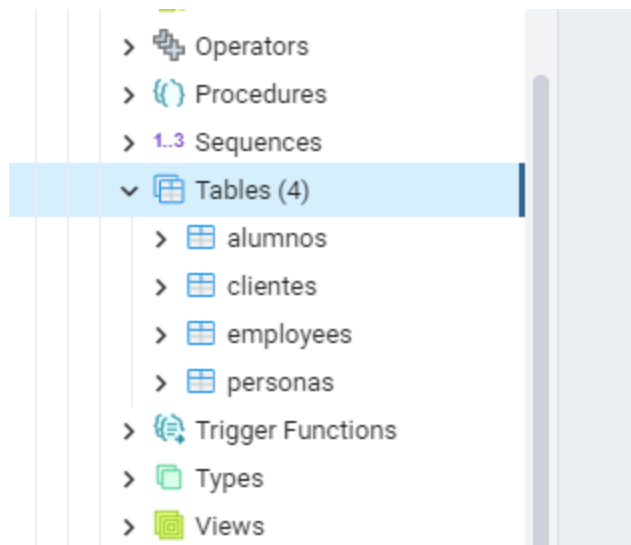
Successfully run. Total query runtime: 150 msec.
0 rows affected.

Ninguna fila afectada (0 row affected)

BORRADO SELECTIVO

Y SI ELIMNAMOS DE MANERA SELECTIVA NUESTROS DATOS?

Volvamos a la tabla alumnos



SELECT*FROM alumnos

| | | | | | | | |
|--------------|----------------------|----------------------------------|---|---|-------------------------|---------------------|--|
| Query Editor | | Query History | | | | | |
| 1 | | SELECT*FROM alumnos | | | | | |
| Data Output | | Explain | | Messages | | Notifications | |
| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date | fechaingres date | |
| 1 | 1000 | Juan | Vazquez | Perez | 2016-05-08 | 1992-02-18 | |
| 2 | 1001 | Juan | Vazquez | Perez | 2018-02-08 | 1991-05-12 | |
| 3 | 1002 | Carlos Miguel | Lopez | Perez | 2017-03-15 | 1989-07-15 | |
| 4 | 1003 | María Carlota | Sánchez | Perez | 2018-02-02 | 1995-08-17 | |
| 5 | 1004 | Casandra | Gavilan | Gonzalez | 2018-07-25 | 1995-06-18 | |
| 6 | 1005 | Andrea | Davila | Antonios | 2018-06-02 | 1996-06-21 | |
| 7 | 1006 | Joao | Aguiar | Garza | 2015-03-15 | 1993-03-28 | |
| 8 | 1007 | Daniel | Zambrano | Espino | 2015-07-18 | 1996-02-05 | |
| 9 | 1008 | Flor | Velazquez | Espinoza | 2017-02-08 | 1995-05-07 | |
| 10 | 1009 | Celeste | Vazquez | De la O | 2015-08-22 | 1995-01-31 | |
| 11 | 1010 | Abigail | Andrade | Beltran | 2020-05-12 | 1999-01-17 | |
| 12 | 1011 | Juan Carlos | Espinoza | Campos | 2020-05-15 | 1995-01-15 | |
| 13 | 1012 | Dionicio | Espino | Espinoza | 2021-06-15 | 2003-05-12 | |
| 14 | 1013 | Jose Carlos | Flores | Garcia | 2021-07-17 | 2003-08-12 | |
| 15 | 1014 | Jose Pedro | Valle | Perez | 2021-08-01 | 2002-06-25 | |
| 16 | 1015 | Miguel Luis | Flores | Sanchez | 2021-01-15 | 2002-05-28 | |

Y si deseamos eliminar solo un campo del registro de nuestra TABLA

Veamos que parece que se cargo 2 veces el mismo dato, y deberíamos eliminar 1

| | | | | | | |
|---|------|---------------|---------|-------|------------|------------|
| 1 | 1000 | Juan | Vazquez | Perez | 2016-05-08 | 1992-02-18 |
| 2 | 1001 | Juan | Vazquez | Perez | 2018-02-08 | 1991-05-12 |
| 3 | 1002 | Carlos Miguel | Lopez | Perez | 2017-03-15 | 1989-07-15 |

Debemos eliminar el 2do registro de Juan, que tiene el id 1000

Para hacerlo realizamos

DELETE FROM alumnos WHERE numcontrol=1000

```
1 DELETE FROM alumnos WHERE numcontrol=1000
```

Lo ejecutamos y

```
Data Explorer | Enquiry | Messages | Notifications
DELETE 1
Query returned successfully in 76 msec.
```

Luego volvemos a consultar

SELECT*FROM alumnos

```
1 SELECT * FROM alumnos
```

Data Output Explain Messages Notifications

| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date | fechaingre date |
|----|----------------------|----------------------------------|---|---|-------------------------|--------------------|
| 1 | 1001 | Juan | Vazquez | Perez | 2018-02-08 | 1991-05-11 |
| 2 | 1002 | Carlos Miguel | Lopez | Perez | 2017-03-15 | 1989-07-11 |
| 3 | 1003 | Maria Carlota | SAnchez | Perez | 2018-02-02 | 1995-08-11 |
| 4 | 1004 | Cassandra | Gavilan | Gonzalez | 2018-07-25 | 1995-06-11 |
| 5 | 1005 | Andrea | Davila | Antonios | 2018-06-02 | 1996-06-21 |
| 6 | 1006 | Joao | Aguiar | Garza | 2015-03-15 | 1993-03-21 |
| 7 | 1007 | Daniel | Zambrano | Espino | 2015-07-18 | 1996-02-01 |
| 8 | 1008 | Flor | Velazquez | Espinoza | 2017-02-08 | 1995-05-01 |
| 9 | 1009 | Celeste | Vazquez | De la O | 2015-08-22 | 1995-01-31 |
| 10 | 1010 | Abigail | Andrade | Beltran | 2020-05-12 | 1999-01-11 |
| 11 | 1011 | Juan Carlos | Espinoza | Campos | 2020-05-15 | 1995-01-11 |

Bien, nuestra tabla es un éxito, pudimos eliminar el registro con ID 1000.

Y si queremos eliminarlo manualmente, ejecutamos

```
DELETE FROM alumnos WHERE nombre='Juan' AND  
apellidopaterno='Vazquez' AND apellidomaterno='Perez'
```

Query Editor Query History

```
1 DELETE FROM alumnos WHERE nombre='Juan' AND apellidopaterno='Vazquez' AND apellidomaterno='Perez'
```

Data Output Explain Messages Notifications

DELETE 1

Query returned successfully in 210 msec.

Y veamos que ya no existen los registros

Query Editor Query History

```
1 SELECT * FROM alumnos
```

Data Output Explain Messages Notifications

| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date | fechaingreso date | telefono character vary | car ch: |
|----|----------------------|----------------------------------|---|---|-------------------------|----------------------|----------------------------|------------|
| 1 | 1002 | Carlos Miguel | Lopez | Perez | 2017-03-15 | 1989-07-15 | 1234564349 | Fis |
| 2 | 1003 | Maria Carlota | Sanchez | Perez | 2018-02-02 | 1995-08-17 | 1234564322 | Psi |
| 3 | 1004 | Cassandra | Gavilan | Gonzalez | 2018-07-25 | 1995-06-18 | 1234564335 | Ma |
| 4 | 1005 | Andrea | Davila | Antonios | 2018-06-02 | 1996-06-21 | 1234564326 | Bio |
| 5 | 1006 | Joao | Aguiar | Garza | 2015-03-15 | 1993-03-28 | 1234564327 | Ing |
| 6 | 1007 | Daniel | Zambrano | Espino | 2015-07-18 | 1996-02-05 | 1234564328 | Rot |
| 7 | 1008 | Flor | Velazquez | Espinoza | 2017-02-08 | 1995-05-07 | 1234564345 | Coi |
| 8 | 1009 | Celeste | Vazquez | De la O | 2015-08-22 | 1995-01-31 | 1234564385 | Bio |
| 9 | 1010 | Abigail | Andrade | Beltran | 2020-05-12 | 1999-01-17 | 1234564373 | Fis |
| 10 | 1011 | Juan Carlos | Espinoza | Campos | 2020-05-15 | 1995-01-15 | 1234564399 | Fis |
| 11 | 1012 | Dionicio | Espino | Espinoza | 2021-06-15 | 2003-05-12 | 1234564398 | Ing |
| 12 | 1013 | Jose Carlos | Flores | Garcia | 2021-07-17 | 2003-08-12 | 1234564390 | Coi |
| 13 | 1014 | Jose Pedro | Valle | Perez | 2021-08-01 | 2002-06-25 | 1234564312 | Ing |
| 14 | 1015 | Miguel Luis | Flores | Sanchez | 2021-01-15 | 2002-05-28 | 1234564315 | Fis |
| 15 | 1016 | JoseMarcelo | Gonzalez | Miranda | 2018-05-12 | 1998-06-25 | 1234564222 | Me |

Ya no empieza mas con “Juan Perez”

Ahora, eliminemos en base a condiciones

Primero , visualizemos todo con

SELECT*FROM alumnos WHERE promedio<90

Aquí veremos a los alumnos con promedio mayor a 90

Query Editor

Query History

1

SELECT*FROM alumnos WHERE promedio<90

2

DELETE FROM alumnos WHERE promedio<90

Data Output

Explain

Messages

Notifications

| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date | fechaingreso date | telefono character vary | carrera charact |
|---|----------------------|----------------------------------|---|---|-------------------------|----------------------|----------------------------|--------------------|
| 1 | 1004 | Casandra | Gavilan | Gonzalez | 2018-07-25 | 1995-06-18 | 1234564335 | Matema |
| 2 | 1010 | Abigail | Andrade | Beltran | 2020-05-12 | 1999-01-17 | 1234564373 | Fisico IV |
| 3 | 1012 | Dionicio | Espino | Espinoza | 2021-06-15 | 2003-05-12 | 1234564398 | Ingenier |
| 4 | 1013 | Jose Carlos | Flores | Garcia | 2021-07-17 | 2003-08-12 | 1234564390 | Contabi |
| 5 | 1014 | Jose Pedro | Valle | Perez | 2021-08-01 | 2002-06-25 | 1234564312 | Ingenier |
| 6 | 1016 | JoseMarcelo | Gonzalez | Miranda | 2018-05-12 | 1998-06-25 | 1234564222 | Mercado |
| 7 | 1019 | Maria Cecilia | Lopez | Lopez | 2020-07-28 | 2003-05-25 | 1234564145 | Psicolog |
| 8 | 1020 | Juan Alberto | Martinez | Vazquez | 2016-07-22 | 1998-02-25 | 1234564142 | Medicin |

Ahora, eliminemos esos registros

```
1
2 DELETE FROM alumnos WHERE promedio<90
```

Data Output Explain Messages Notifications

DELETE 8

Query returned successfully in 65 msec.

Vemos que se eliminaron 8 registros, intentemos ver

Query Editor Query History

```
1 SELECT*FROM alumnos WHERE promedio<90
```

Data Output Explain Messages Notifications

| numcontrol | nombre | apellidopaterno | apellidomaterno | fechanacimiento | fechaingreso | telefono |
|------------|------------------------|------------------------|------------------------|-----------------|--------------|----------|
| bigint | character varying (45) | character varying (45) | character varying (45) | date | date | chara |

Nada.

SENTENCIA DROP TABLE

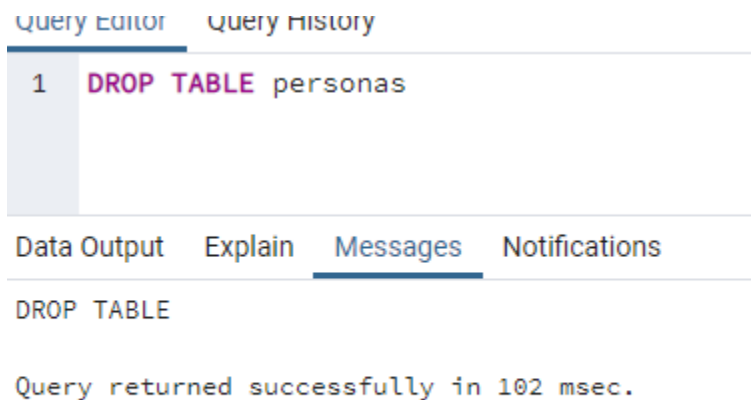
Usando

DROP TABLE,tabla1,tabla2,..

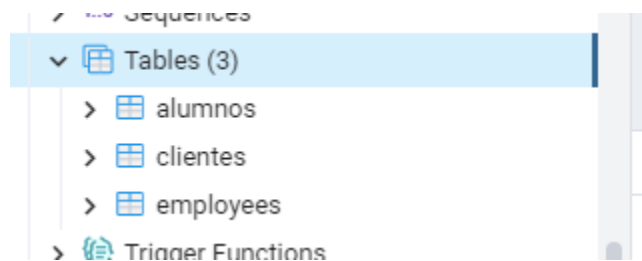
PODEMOS ELIMINAR TABLAS.

Comenzemos.

DROP TABLE personas



Ahora ya no podremos ver la tabla

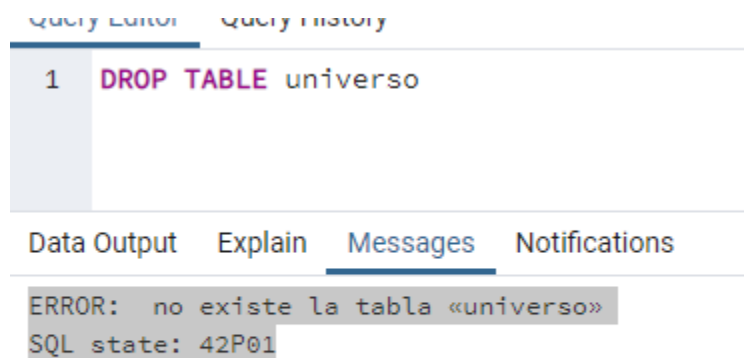


Se elimino la tabla “personas”

Probemos ahora una variación

IF EXISTS

Que sucede si queremos eliminar una tabla que
no existe?



Podemos utilizar una instrucción que solo eliminara la tabla si existe.

DROP TABLE IF EXISTS

```
1 DROP TABLE IF EXISTS universo
```

Data Output Explain Messages Notifications

```
NOTICE: la tabla «universo» no existe, omitiendo  
DROP TABLE
```

```
Query returned successfully in 121 msec.
```

En el caso que no exista. La omite

ACTUALIZAR UNA TABLA

**Ahora aprenderemos a actualizar una tabla, para comenzar,
listemos una tabla**

SELECT*FROM alumnos

1

SELECT*FROM alumnos

Data Output

Explain

Messages

Notifications

| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date | f |
|----|----------------------|----------------------------------|---|---|-------------------------|---|
| 1 | 1002 | Carlos Miguel | Lopez | Perez | 2017-03-15 | 1 |
| 2 | 1003 | Maria Carlota | SAncchez | Perez | 2018-02-02 | 1 |
| 3 | 1005 | Andrea | Davila | Antonios | 2018-06-02 | 1 |
| 4 | 1006 | Joao | Aguiar | Garza | 2015-03-15 | 1 |
| 5 | 1007 | Daniel | Zambrano | Espino | 2015-07-18 | 1 |
| 6 | 1008 | Flor | Velazquez | Espinoza | 2017-02-08 | 1 |
| 7 | 1009 | Celeste | Vazquez | De la O | 2015-08-22 | 1 |
| 8 | 1011 | Juan Carlos | Espinoza | Campos | 2020-05-15 | 1 |
| 9 | 1015 | Miguel Luis | Flores | Sanchez | 2021-01-15 | 2 |
| 10 | 1017 | Flor Estela | Huerta | Espinosa | 2018-07-09 | 1 |
| 11 | 1018 | Cristian Jesus | Velazquez | Perez | 2018-01-08 | 1 |

SENTENCIA UPDATE SET

Con la instrucción UPDATE apuntando a una tabla podemos actualizar sus registros, como por ejemplo

UPDATE empleados SET nombre= 'Hola Mundo'

ATENCION: este comando tiene un problema, y es que, actualizara el campo nombre en todas sus filas, es decir, todos los campos nombre tendrán ahora el valor “ 'Hola Mundo' ”

Veamos

Antes :

| Data Output | | Explain | Messages | Notifications | | |
|-------------|----------------------|----------------------------------|---|---|-------------------------|--------|
| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date | f c |
| 1 | 1002 | Carlos Miguel | Lopez | Perez | 2017-03-15 | 1 |
| 2 | 1003 | Maria Carlota | SAncchez | Perez | 2018-02-02 | 1 |
| 3 | 1005 | Andrea | Davila | Antonios | 2018-06-02 | 1 |
| 4 | 1006 | Joao | Aguiar | Garza | 2015-03-15 | 1 |
| 5 | 1007 | Daniel | Zambrano | Espino | 2015-07-18 | 1 |
| 6 | 1008 | Flor | Velazquez | Espinoza | 2017-02-08 | 1 |
| 7 | 1009 | Celeste | Vazquez | De la O | 2015-08-22 | 1 |
| 8 | 1011 | Juan Carlos | Espinoza | Campos | 2020-05-15 | 1 |
| 9 | 1015 | Miguel Luis | Flores | Sanchez | 2021-01-15 | 2 |
| 10 | 1017 | Flor Estela | Huerta | Espinosa | 2018-07-09 | 1 |
| 11 | 1018 | Cristian Jesus | Kilberth | Perez | 2018-01-08 | 1 |
| 12 | 1021 | Franchesco Daniel | Nunez | Perez | 2017-07-15 | 1 |
| 13 | 1022 | Laura | Quinonez | Garcia | 2020-02-08 | 2 |

Despues :

```
1 UPDATE alumnos SET nombre='Hola Mundo'
```

```
SET/TEXT SET/TEXT
```

```
1 UPDATE alumnos SET nombre='Hola Mundo'
```

Data Output Explain Messages Notifications

UPDATE 13

Query returned successfully in 86 msec.

SELECT*FROM alumnos

Query Editor Query History

1 **SELECT*****FROM** alumnos

Data Output Explain Messages Notifications

| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date |
|----|----------------------|----------------------------------|---|---|-------------------------|
| 1 | 1002 | Hola Mundo | Lopez | Perez | 2017-03-15 |
| 2 | 1003 | Hola Mundo | SAnchez | Perez | 2018-02-02 |
| 3 | 1005 | Hola Mundo | Davila | Antonios | 2018-06-02 |
| 4 | 1006 | Hola Mundo | Aguiar | Garza | 2015-03-15 |
| 5 | 1007 | Hola Mundo | Zambrano | Espino | 2015-07-18 |
| 6 | 1008 | Hola Mundo | Velazquez | Espinoza | 2017-02-08 |
| 7 | 1009 | Hola Mundo | Vazquez | De la O | 2015-08-22 |
| 8 | 1011 | Hola Mundo | Espinoza | Campos | 2020-05-15 |
| 9 | 1015 | Hola Mundo | Flores | Sanchez | 2021-01-15 |
| 10 | 1017 | Hola Mundo | Huerta | Espinosa | 2018-07-09 |
| 11 | 1018 | Hola Mundo | Kilberth | Perez | 2018-01-08 |
| 12 | 1021 | Hola Mundo | Nunez | Perez | 2017-07-15 |
| 13 | 1022 | Hola Mundo | Quinonez | Garcia | 2020-02-08 |

| | |
|------------------------|--|
| nombre | |
| character varying (45) | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |
| Hola Mundo | |

Eliminemos entonces esta tabla que se encuentra errónea.

DROP TABLE alumnos

Query Editor

Query History

1

DROP TABLE alumnos

Data Output

Explain

Messages

Notifications

DROP TABLE

Query returned successfully in 66 msec.

Ahora, se eliminara

```
1 DROP TABLE alumnos
```

Data Output

Explain

Messages

Notifications

| | numcontrol bigint | nombre character varying (45) | apellidopaterno character varying (45) | apellidomaterno character varying (45) | fechanacimiento date | |
|----|----------------------|----------------------------------|---|---|-------------------------|---|
| 1 | 1002 | Hola Mundo | Lopez | Perez | 2017-03-15 | 1 |
| 2 | 1003 | Hola Mundo | SA Sanchez | Perez | 2018-02-02 | 1 |
| 3 | 1005 | Hola Mundo | Davila | Antonios | 2018-06-02 | 1 |
| 4 | 1006 | Hola Mundo | Aguar | Garza | 2015-03-15 | 1 |
| 5 | 1007 | Hola Mundo | Zambrano | Espino | 2015-07-18 | 1 |
| 6 | 1008 | Hola Mundo | Velazquez | Espinoza | 2017-02-08 | 1 |
| 7 | 1009 | Hola Mundo | Vazquez | De la O | 2015-08-22 | 1 |
| 8 | 1011 | Hola Mundo | Espinoza | Campos | 2020-05-15 | 1 |
| 9 | 1015 | Hola Mundo | Flores | Sanchez | 2021-01-15 | 2 |
| 10 | 1017 | Hola Mundo | Huerta | Espinosa | 2018-07-09 | 1 |
| 11 | 1018 | Hola Mundo | Kilberth | Perez | 2018-01-08 | 1 |
| 12 | 1021 | Hola Mundo | Nunez | Perez | 2017-07-15 | 1 |
| 13 | 1022 | Hola Mundo | Quinonez | Garcia | 2020-02-08 | 2 |

Sequences

Tables (2)

> clientes

> employees

> Trigger Functions

Data

DROP