

4.- Script de la base de datos: Un script es un archivo que contiene una serie de comandos de SQL, los cuales se ejecutan de forma secuencial y se utilizan para automatizar el trabajo dentro de una base de datos. Es importante señalar que un script puede contener cualquier comando de SQL valido y el archivo fuente tiene que ser de extensión **.SQL**.

Para ejecutar un script se aplica el comando **i (install)**

`=# \i Unidad:/...ruta/ ... /archivo_script.sql`

```
postgres=# \i c:/datos/script_datos.sql_
```

5.- Select --> Consultas

5.1.- SELECT * FROM <Tabla> ;

Muestra todos los campos y todos los registros de la tabla

Ejemplo: visualizar todos los campos y todos los registros de la TMEMPLEDOS

```
SELECT * FROM tmempleados ;
```

```
bdretardados=# SELECT * FROM tmempleados;
```

pkcedemple	nomemple	fecha	fkcodcar	fkcods
1000	ROBERTO JAIMES	1978-03-03	3	1
2000	ZILA CONTRERAS	1980-10-05	2	1
3000	MARTHA	1996-11-22	3	1
4000	CARLOS	2000-07-15	4	1
5000	MATIAS	2005-09-25	4	1
6000	NATASHA	2007-10-28	1	1
7000	PEGGY CARTER	1999-08-29	2	1
8000	YSA CAPACHO	1993-07-23	2	1
9000	MARIA MORALES	2003-07-07	4	1

(9 filas)

5.2.- **SELECT** Campo1 , Campo2 , , CampoN **FROM** <Tabla> **;**

Muestra de la tabla, todos los registros y el contenido de los campos mencionados.

Ejemplo: Mostrar de la tabla TMEMPLEDOS el contenido de los campos CEDEMPLE, FECHA Y FKCODCAR.

```
SELECT pkcedemple, fecha, fkcodcar FROM templeados;
```

```
bdretardados=# SELECT pkcedemple, fecha, fkcodcar FROM templeados;
pkcedemple | fecha      | fkcodcar
-----+-----+-----
1000       | 1978-03-03 | 3
2000       | 1980-10-05 | 2
3000       | 1996-11-22 | 3
4000       | 2000-07-15 | 4
5000       | 2005-09-25 | 4
6000       | 2007-10-28 | 1
7000       | 1999-08-29 | 2
8000       | 1993-07-23 | 2
9000       | 2003-07-07 | 4
(9 filas)
```

5.3.- SELECT * FROM Tabla1 , Tabla2 , , TablaN ;

Muestra en la consola el resultado de combinar dos o más tablas.

```
bdretardados=# select * from tmstatus;
pkcods | dstatus
-----+-----
      0 | ELIMINADO
      1 | ACTIVO
(2 filas)
```

```
bdretardados=# select * from tmcargos;
pkcodcar | dcargo | sueldo | fkcods
-----+-----+-----+-----
      1 | PRESIDENTE | 5000000.01 | 1
      2 | VICE PRESIDENTE | 4500000.02 | 0
      3 | GERENTE | 4000000.03 | 1
      4 | SUB-GERENTE | 3500000.04 | 0
      5 | COORDINADOR | 3000000.05 | 1
      6 | SUPERVISOR | 2500000.06 | 0
(6 filas)
```

Ejemplo mostrar todos los campos y todos los registros de la combinación de las tablas TMStatus y TMCargos

SELECT * FROM tmcargos, tmstatus

```
bdretardados=# select * from tmcargos, tmstatus;
pkcodcar | dcargo | sueldo | fkcods | pkcods | dstatus
-----+-----+-----+-----+-----+-----
      1 | PRESIDENTE | 5000000.01 | 1 | 0 | ELIMINADO
      2 | VICE PRESIDENTE | 4500000.02 | 0 | 0 | ELIMINADO
      3 | GERENTE | 4000000.03 | 1 | 0 | ELIMINADO
      4 | SUB-GERENTE | 3500000.04 | 0 | 0 | ELIMINADO
      5 | COORDINADOR | 3000000.05 | 1 | 0 | ELIMINADO
      6 | SUPERVISOR | 2500000.06 | 0 | 0 | ELIMINADO
      1 | PRESIDENTE | 5000000.01 | 1 | 1 | ACTIVO
      2 | VICE PRESIDENTE | 4500000.02 | 0 | 1 | ACTIVO
      3 | GERENTE | 4000000.03 | 1 | 1 | ACTIVO
      4 | SUB-GERENTE | 3500000.04 | 0 | 1 | ACTIVO
      5 | COORDINADOR | 3000000.05 | 1 | 1 | ACTIVO
      6 | SUPERVISOR | 2500000.06 | 0 | 1 | ACTIVO
(12 filas)
```

5.4.- SELECT Campo1, Campo2 ,, CampoN

FROM Tabla1, Tabla2, , TablaN ;

Se puede combinar el contenido de dos o más tablas, donde solamente se muestran los registros de los campos mencionados.

SELECT pkcedemple, nomemple, fecha, fkcodcar, pkcodcar, dcargo

FROM tmempleados, tmcargos;

```
bdretardados=# SELECT  pkcedemple, nomemple, fecha, fkcodcar, pkcodcar, dcargo
bdretardados=# FROM    tmempleados, tmcargos;
```

pkcedemple	nomemple	fecha	fkcodcar	pkcodcar	dcargo
1000	ROBERTO JAIMES	1978-03-03	3	1	PRESIDENTE
2000	ZILA CONTRERAS	1980-10-05	2	1	PRESIDENTE
3000	MARTHA	1996-11-22	3	1	PRESIDENTE
4000	CARLOS	2000-07-15	4	1	PRESIDENTE
5000	MATIAS	2005-09-25	4	1	PRESIDENTE
6000	NATASHA	2007-10-28	1	1	PRESIDENTE
7000	PEGGY CARTER	1999-08-29	2	1	PRESIDENTE
8000	YSA CAPACHO	1993-07-23	2	1	PRESIDENTE
9000	MARIA MORALES	2003-07-07	4	1	PRESIDENTE
1000	ROBERTO JAIMES	1978-03-03	3	2	VICE PRESIDENTE
2000	ZILA CONTRERAS	1980-10-05	2	2	VICE PRESIDENTE
3000	MARTHA	1996-11-22	3	2	VICE PRESIDENTE
4000	CARLOS	2000-07-15	4	2	VICE PRESIDENTE

*

*

4000	CARLOS	2000-07-15	4	5	COORDINADOR
5000	MATIAS	2005-09-25	4	5	COORDINADOR
6000	NATASHA	2007-10-28	1	5	COORDINADOR
7000	PEGGY CARTER	1999-08-29	2	5	COORDINADOR
8000	YSA CAPACHO	1993-07-23	2	5	COORDINADOR
9000	MARIA MORALES	2003-07-07	4	5	COORDINADOR
1000	ROBERTO JAIMES	1978-03-03	3	6	SUPERVISOR
2000	ZILA CONTRERAS	1980-10-05	2	6	SUPERVISOR
3000	MARTHA	1996-11-22	3	6	SUPERVISOR
4000	CARLOS	2000-07-15	4	6	SUPERVISOR
5000	MATIAS	2005-09-25	4	6	SUPERVISOR
6000	NATASHA	2007-10-28	1	6	SUPERVISOR
7000	PEGGY CARTER	1999-08-29	2	6	SUPERVISOR
8000	YSA CAPACHO	1993-07-23	2	6	SUPERVISOR
9000	MARIA MORALES	2003-07-07	4	6	SUPERVISOR

(54 filas)

5.5.- SELECT * FROM tabla WHERE Pregunta1 ;

Se pueden mostrar los campos y los registros que cumplan con una condición o pregunta dada.

Ejemplo: mostrar todos los campos y todos los registros de la TEMPLADOS cuya clave foránea del cargo es el 3.

SELECT * FROM templeados WHERE fkcodcar = 3 ;

```

bdretardados=# SELECT * FROM templeados WHERE fkcodcar = 3 ;
 pkcedemple | nomemple | fecha | fkcodcar | fkods
-----+-----+-----+-----+-----
 1000      | ROBERTO JAIMES | 1978-03-03 | 3 | 1
 3000      | MARTHA | 1996-11-22 | 3 | 1
(2 filas)

bdretardados=# SELECT * FROM templeados WHERE fkcodcar >= 3 ;
 pkcedemple | nomemple | fecha | fkcodcar | fkods
-----+-----+-----+-----+-----
 1000      | ROBERTO JAIMES | 1978-03-03 | 3 | 1
 3000      | MARTHA | 1996-11-22 | 3 | 1
 4000      | CARLOS | 2000-07-15 | 4 | 1
 5000      | MATIAS | 2005-09-25 | 4 | 1
 9000      | MARIA MORALES | 2003-07-07 | 4 | 1
(5 filas)

```

No	Condición	Ejemplos
1	=	Nomemple= ' SANDRA ' / FKCODCAR = 3 / FECHA= ' 2007-10-28 '
2	>	FKCODCAR > 3 / FECHA> ' 2007-10-28 '
3	>=	FKCODCAR >= 3 / FECHA>= ' 2007-10-28 '
4	<	FKCODCAR < 3 / FECHA< ' 2007-10-28 '
5	<=	FKCODCAR <= 3 / FECHA<= ' 2007-10-28 '
6	<> / !=	FKCODCAR <> 3 / FECHA<> ' 2007-10-28 '

5.6.- SELECT * FROM tabla

WHERE Pregunta1 **ConectivoLógico** Pregunta2.... **ConectivoLógico**

pregunta2..... **ConectivoLógico**. PreguntaN **;**

Se pueden mostrar todos los campos y todos los registros que cumplan con las condiciones o preguntas dadas.

Ejemplo 01: Mostrar todos los retardos del empleado cuya cédula es 7000 y el código de la excusa presentado sea mayor o igual a 3

SELECT * FROM tdretardos **WHERE** fkcedemple = '7000' **and** fkcodexcu >= 3 **;**

```
bdretardados=# SELECT * FROM tdretardos WHERE fkcedemple = '7000' and fkcodexcu = 3 ;
pknr | fkcedemple | fkcodexcu | fecha | hora | fkcods
-----+-----+-----+-----+-----+-----
 67 | 7000 | 3 | 2022-01-14 | 08:32:00 | 1
(1 fila)
```

```
bdretardados=# SELECT * FROM tdretardos WHERE fkcedemple = '7000' and fkcodexcu >= 3 ;
pknr | fkcedemple | fkcodexcu | fecha | hora | fkcods
-----+-----+-----+-----+-----+-----
 67 | 7000 | 3 | 2022-01-14 | 08:32:00 | 1
 97 | 7000 | 5 | 2022-11-27 | 08:20:00 | 1
101 | 7000 | 7 | 2022-03-05 | 09:11:00 | 1
108 | 7000 | 4 | 2022-10-27 | 08:25:00 | 1
116 | 7000 | 4 | 2022-11-10 | 08:27:00 | 1
(5 filas)
```

No	Conectivo Lógico	Explicación
1	AND (Y)	Si todas las preguntas se cumplen (es decir, dan verdadero) se muestran los registros
2	OR (O)	Si al menos una de las condiciones se cumple se muestran los registros
3	NOT (No)	Niega el resultado de la pregunta;

Ejemplo 02: Mostrar todos los retardos del empleado cuya cédula es 7000 o donde el código de la excusa presentado sea igual a 3

SELECT * FROM tdretardos **WHERE** fkcedemple = '7000' **or** fkcodexcu = 3;

```

bdretardados=# SELECT * FROM tdretardos WHERE fkcedemple = '7000' or fkcodexcu = 3;
 pknr | fkcedemple | fkcodexcu | fecha      | hora      | fkcods
-----+-----+-----+-----+-----+-----
  1   | 2000       | 3         | 2022-04-05 | 08:32:00 | 1
  2   | 5000       | 3         | 2022-04-05 | 08:50:00 | 1
  9   | 2000       | 3         | 2022-04-21 | 08:13:00 | 1
 20   | 3000       | 3         | 2022-01-07 | 10:50:00 | 1
 26   | 7000       | 1         | 2022-05-05 | 09:00:00 | 1
 27   | 6000       | 3         | 2022-05-07 | 08:20:00 | 1
 35   | 3000       | 3         | 2022-02-01 | 08:40:00 | 1
 38   | 6000       | 3         | 2022-02-07 | 08:20:00 | 1
 44   | 4000       | 3         | 2022-02-27 | 08:20:00 | 1
 49   | 3000       | 3         | 2022-03-06 | 09:13:00 | 1
 51   | 1000       | 3         | 2022-03-14 | 08:16:00 | 1
 59   | 2000       | 3         | 2022-03-25 | 08:32:00 | 1
 65   | 3000       | 3         | 2022-01-06 | 09:10:00 | 1
 67   | 7000       | 3         | 2022-01-14 | 08:32:00 | 1
 74   | 6000       | 3         | 2022-01-25 | 08:38:00 | 1
 78   | 7000       | 1         | 2022-06-05 | 09:00:00 | 1
 79   | 6000       | 3         | 2022-06-07 | 08:20:00 | 1
 90   | 7000       | 1         | 2022-07-05 | 09:00:00 | 1
 91   | 5000       | 3         | 2022-07-07 | 08:20:00 | 1
 97   | 7000       | 5         | 2022-11-27 | 08:20:00 | 1
100   | 4000       | 3         | 2022-02-03 | 08:35:00 | 1
101   | 7000       | 7         | 2022-03-05 | 09:11:00 | 1
108   | 7000       | 4         | 2022-10-27 | 08:25:00 | 1
111   | 4000       | 3         | 2022-11-03 | 08:35:00 | 1
116   | 7000       | 4         | 2022-11-10 | 08:27:00 | 1
119   | 6000       | 3         | 2022-11-12 | 08:40:00 | 1
(26 filas)

```

5.7.- SELECT *

FROM Tabla1, Tabla2,....., TablaN

WHERE Pregunta1 **ConectivoLógico** Pregunta2.... **ConectivoLógico**

Pregunta2... **ConectivoLógico** PreguntaN ;

Se pueden mostrar todos los campos y todos los registros de la combinación de dos o más tablas; y además, cumplan con las condiciones de las preguntas dadas. Ejemplo:

```
bdretardados=# select * from tmempleados, tmstatus;
pkcedemple | nomemple | fecha | fkcodcar | fkcods | pkcods | dstatus
-----+-----+-----+-----+-----+-----+-----
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1 | 0 | ELIMINADO
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1 | 0 | ELIMINADO
3000 | MARTHA | 1996-11-22 | 3 | 1 | 0 | ELIMINADO
4000 | CARLOS | 2000-07-15 | 4 | 1 | 0 | ELIMINADO
5000 | MATIAS | 2005-09-25 | 4 | 1 | 0 | ELIMINADO
6000 | NATASHA | 2007-10-28 | 1 | 1 | 0 | ELIMINADO
7000 | PEGGY CARTER | 1999-08-29 | 2 | 1 | 0 | ELIMINADO
8000 | YSA CAPACHO | 1993-07-23 | 2 | 1 | 0 | ELIMINADO
9000 | MARIA MORALES | 2003-07-07 | 4 | 1 | 0 | ELIMINADO
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1 | 1 | ACTIVO
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1 | 1 | ACTIVO
3000 | MARTHA | 1996-11-22 | 3 | 1 | 1 | ACTIVO
4000 | CARLOS | 2000-07-15 | 4 | 1 | 1 | ACTIVO
5000 | MATIAS | 2005-09-25 | 4 | 1 | 1 | ACTIVO
6000 | NATASHA | 2007-10-28 | 1 | 1 | 1 | ACTIVO
7000 | PEGGY CARTER | 1999-08-29 | 2 | 1 | 1 | ACTIVO
8000 | YSA CAPACHO | 1993-07-23 | 2 | 1 | 1 | ACTIVO
9000 | MARIA MORALES | 2003-07-07 | 4 | 1 | 1 | ACTIVO
(18 filas)
```

SELECT * **FROM** tmempleados, tmstatus **WHERE** fkcods = pkcods ;

```
bdretardados=# select * from tmempleados, tmstatus WHERE fkcods = cods;
cedemple | nomemple | fecha | fkcodcar | fkcods | cods | dstatus
-----+-----+-----+-----+-----+-----+-----
1000 | ROBERTO | 1978-03-03 | 1 | 1 | 1 | ACTIVO
2000 | MARTHA | 1980-10-05 | 2 | 1 | 1 | ACTIVO
3000 | SANDRA | 1996-11-22 | 3 | 1 | 1 | ACTIVO
4000 | WILMARY | 2000-07-15 | 4 | 1 | 1 | ACTIVO
5000 | CARLOS | 2005-09-25 | 4 | 1 | 1 | ACTIVO
6000 | NATASHA | 2007-10-28 | 4 | 1 | 1 | ACTIVO
7000 | MARIA | 1999-08-29 | 4 | 1 | 1 | ACTIVO
(7 filas)
```

5.8.- SELECT Campo1, Campo2 ,, CampoN
FROM Tabla1, Tabla2, , TablaN
WHERE Pregunta1 **ConectivoLógico**

Pregunta2... **ConectivoLógico**PreguntaN ;

Se pueden combinar el contenido de dos o más tablas, luego filtrar los datos según las preguntas o condiciones dadas; y además solamente mostrar el contenido de algunos campos.

Ejemplo 01: Mostrar la cédula, el nombre, la descripción del estatus de todos los empleados

SELECT pkcedemple, nomemple, fkcods, dstatus
FROM tmempleados, tmstatus
WHERE fkcods = pkcods ;

SELECT pkcedemple, nomemple, dstatus
FROM tmempleados, tmstatus
WHERE fkcods = pkcods ;

```
bdretardados=# SELECT pkcedemple, nomemple, fkcods, dstatus
bdretardados=# FROM tmempleados, tmstatus
bdretardados=# WHERE fkcods = pkcods ;
 pkcedemple | nomemple | fkcods | dstatus
-----+-----+-----+-----
 1000       | ROBERTO JAIMES | 1 | ACTIVO
 2000       | ZILA CONTRERAS | 1 | ACTIVO
 3000       | MARTHA         | 1 | ACTIVO
 4000       | CARLOS         | 1 | ACTIVO
 5000       | MATIAS         | 1 | ACTIVO
 6000       | NATASHA        | 1 | ACTIVO
 7000       | PEGGY CARTER  | 1 | ACTIVO
 8000       | YSA CAPACHO   | 1 | ACTIVO
 9000       | MARIA MORALES  | 1 | ACTIVO
(9 filas)
```

```
bdretardados=# SELECT pkcedemple, nomemple, dstatus
bdretardados=# FROM tmempleados, tmstatus
bdretardados=# WHERE fkcods = pkcods ;
 pkcedemple | nomemple | dstatus
-----+-----+-----
 1000       | ROBERTO JAIMES | ACTIVO
 2000       | ZILA CONTRERAS | ACTIVO
 3000       | MARTHA         | ACTIVO
 4000       | CARLOS         | ACTIVO
 5000       | MATIAS         | ACTIVO
 6000       | NATASHA        | ACTIVO
 7000       | PEGGY CARTER  | ACTIVO
 8000       | YSA CAPACHO   | ACTIVO
 9000       | MARIA MORALES  | ACTIVO
(9 filas)
```

Ejemplo 02: Mostrar el número del retardo, la cédula del empleado, el código de la excusa y la fecha de la excusa; del empleado con número de cédula '7000' y los retardos sean mayores o iguales a la fecha del 01/Mayo/2022.

```
SELECT pknr, fkcedemple, fkcodexcu, dexcusa, fecha
FROM tdretardos, tmexcusas
```

```
WHERE fkcedemple = '7000' and fkcodexcu = pkcodexcu and fecha >= '2022-05-01';
```

```
bdretardados=# SELECT pknr, fkcedemple, fkcodexcu, dexcusa, fecha
bdretardados=# FROM tdretardos, tmexcusas
bdretardados=# WHERE fkcedemple = '7000' and fkcodexcu = pkcodexcu and fecha >= '2022-05-01';
 pknr | fkcedemple | fkcodexcu | dexcusa | fecha
-----+-----+-----+-----+-----
 26 | 7000 | 1 | ME QUEDE DORMIDO | 2022-05-05
 78 | 7000 | 1 | ME QUEDE DORMIDO | 2022-06-05
 90 | 7000 | 1 | ME QUEDE DORMIDO | 2022-07-05
 97 | 7000 | 5 | MUCHA LLUVIA | 2022-11-27
108 | 7000 | 4 | PELIE CON EL AGUILA | 2022-10-27
116 | 7000 | 4 | PELIE CON EL AGUILA | 2022-11-10
(6 filas)
```

Ejemplo 03: Se necesita un reporte que muestre del cargo los siguientes datos: Código, descripción , sueldo, código del status y descripción del estatus.

```
SELECT * FROM tmcargos, tmstatus;
```

```
SELECT pkcodcar, dcargo, to_char(sueldo, '999G999G999D99'), fkcods, dstatus
FROM tmcargos, tmstatus
```

```
WHERE fkcods = pkcods ;
```

```
pkcodcar | dcargo | to_char | fkcods | dstatus
-----+-----+-----+-----+-----
 1 | PRESIDENTE | 5.000.000,01 | 1 | ACTIVO
 2 | VICE PRESIDENTE | 4.500.000,02 | 0 | ELIMINADO
 3 | GERENTE | 4.000.000,03 | 1 | ACTIVO
 4 | SUB-GERENTE | 3.500.000,04 | 0 | ELIMINADO
 5 | COORDINADOR | 3.000.000,05 | 1 | ACTIVO
 6 | SUPERVISOR | 2.500.000,06 | 0 | ELIMINADO
(6 filas)
```

5.9.- ORDENAR DATOS: Se puede usar la sentencia select para ordenar datos:

SELECT Campo1, Campo2, ..., CampoN

FROM Tabla1, Tabla2, ..., TablaN

WHERE Pregunta1 **ConectivoLógico** Pregunta2... **ConectivoLógico** PreguntaN

ORDER BY Campo1 ASC/DESC,, CampoN ASC/DESC ;

Esta fórmula indica que ud puede combinar dos o más tablas, mostrar algunos campos de la combinación de las tablas; además, se pueden hacer preguntas y finalmente **ordenar** los datos por uno o varios campos en forma ascendente (**A**) o descendente (**D**).

Es importante indicar que el SELECT ordena por defecto en forma ascendente (**A**), pero si se desea ordenar un campo en forma descendente se debe agregar obligatoriamente la cláusula **DESC**.

Ejemplo 01: Mostrar los datos de los empleados ordenados por los campos código del cargo (**A**- Ascendente) y fecha (**D** - Descendente)

SELECT * FROM templeados ;

SELECT * FROM templeados

ORDER BY fecha ;

SELECT * FROM templeados

ORDER BY fkcodcar, fecha **DESC** ;

bdretardados=#	pkcedemple	nomemple	fecha	fkcodcar	fkcds
1000		ROBERTO JAINES	1978-03-03	3	1
2000		ZILA CONTRERAS	1980-10-05	2	1
8000		YSA CAPACHO	1993-07-23	2	1
3000		MARTHA	1996-11-22	3	1
7000		PEGGY CARTER	1999-08-29	2	1
4000		CARLOS	2000-07-15	4	1
9000		MARIA MORALES	2003-07-07	4	1
5000		MATIAS	2005-09-25	4	1
6000		NATASHA	2007-10-28	1	1

(9 filas)

bdretardados=#	pkcedemple	nomemple	fecha	fkcodcar	fkcds	DESC
6000		NATASHA	2007-10-28	1	1	
7000		PEGGY CARTER	1999-08-29	2	1	
8000		YSA CAPACHO	1993-07-23	2	1	
2000		ZILA CONTRERAS	1980-10-05	2	1	
3000		MARTHA	1996-11-22	3	1	
1000		ROBERTO JAINES	1978-03-03	3	1	
5000		MATIAS	2005-09-25	4	1	
9000		MARIA MORALES	2003-07-07	4	1	
4000		CARLOS	2000-07-15	4	1	

(9 filas)

Ejemplo 02: Se desea mostrar el número del retardo, la cédula del empleado, el código de la excusa del retardo, descripción de la excusa del retardo y fecha del retardo; donde el retardo pertenece al empleado cuya cédula es '7000', y demás la fecha del retardo es mayor igual al 01/Febrero/2022. Finalmente, dicha consulta debe mostrar los datos ordenados por el campo fecha del retardo en forma Ascendente.

```
SELECT pknr, fkcedemple, fkcodexcu, fecha
FROM tdretardos, tmexcusas
```

```
WHERE fkcedemple = '7000' and fkcodexcu = pkcodexcu and fecha >= '2022-02-01';
```

```
SELECT pknr, fkcedemple, fkcodexcu, dexexcusa, fecha
FROM tdretardos, tmexcusas
```

```
WHERE fkcedemple = '7000' and fkcodexcu = pkcodexcu and fecha >= '2022-02-01'
```

```
ORDER BY fecha ;
```

```
bdretardados=# SELECT pknr, fkcedemple, fkcodexcu, fecha
bdretardados=# FROM tdretardos, tmexcusas
bdretardados=# WHERE fkcedemple = '7000' and fkcodexcu = pkcodexcu and fecha >= '2022-02-01';
 pknr | fkcedemple | fkcodexcu | fecha
-----+-----+-----+-----
 26 | 7000 | 1 | 2022-05-05
 78 | 7000 | 1 | 2022-06-05
 90 | 7000 | 1 | 2022-07-05
 97 | 7000 | 5 | 2022-11-27
101 | 7000 | 7 | 2022-03-05
108 | 7000 | 4 | 2022-10-27
116 | 7000 | 4 | 2022-11-10
(7 filas)

bdretardados=#
bdretardados=# SELECT pknr, fkcedemple, fkcodexcu, dexexcusa, fecha
bdretardados=# FROM tdretardos, tmexcusas
bdretardados=# WHERE fkcedemple = '7000' and fkcodexcu = pkcodexcu and fecha >= '2022-02-01'
bdretardados=# ORDER BY fecha ;
 pknr | fkcedemple | fkcodexcu | dexexcusa | fecha
-----+-----+-----+-----+-----
 101 | 7000 | 7 | EL CARRO NO ME QUIERE | 2022-03-05
 26 | 7000 | 1 | ME QUEDE DORMIDO | 2022-05-05
 78 | 7000 | 1 | ME QUEDE DORMIDO | 2022-06-05
 90 | 7000 | 1 | ME QUEDE DORMIDO | 2022-07-05
108 | 7000 | 4 | PELIE CON EL AGUILA | 2022-10-27
116 | 7000 | 4 | PELIE CON EL AGUILA | 2022-11-10
 97 | 7000 | 5 | MUCHA LLUVIA | 2022-11-27
(7 filas)

bdretardados=#
```

NOTA: **CAST()** / Valor **::** Tipo_Destino --→ Conversión temporal

La función CAST() en SQL convierte un tipo de dato en otro tipo compatible. Por ejemplo, puede transformar una cadena en un valor entero.

Usos:

- * Es una tarea común en el trabajo con bases de datos.
- * Es necesario para garantizar que los datos sean consistentes y del tipo de datos correcto para su uso previsto, ejemplo ordenar datos
- * CAST es SQL estándar ANSI; sin embargo, la sentencia CAST() se puede cambiar por el símbolo **::**

Ejemplo 03: Se desea insertar los siguientes registros en la tabla tmempleados

INSERT INTO tmempleados (pkcedemple, nomemple, fecha, fkcodcar) **VALUES**

```
('17',      'VALERIA',      '1981-04-07' , 3),
('13',      'ANGELY',      '1978-06-21' , 2),
('7',       'SERGIO CAPACHO', '1967-11-22' , 3),
('4020',    'YSAURA',      '2018-02-05' , 4),
('4017',    'MESSI',        '1996-06-09' , 4),
('4011',    'RONALDO',      '2011-12-25' , 1),
('9057',    'CARMENZA',     '1988-08-09' , 2),
('9026',    'FANNY',        '2005-07-10' , 2),

('10000',   'MERCY',        '2013-07-07' , 4) ;
```

Con lo cual al realizar una consulta tenemos lo siguiente:

```
SELECT * FROM templeados ;
```

```
bdretardados=# select * from templeados;
```

pkcedemple	nomemple	fecha	fkcodcar	fkcods
1000	ROBERTO JAIMES	1978-03-03	3	1
2000	ZILA CONTRERAS	1980-10-05	2	1
3000	MARTHA	1996-11-22	3	1
4000	CARLOS	2000-07-15	4	1
5000	MATIAS	2005-09-25	4	1
6000	NATASHA	2007-10-28	1	1
7000	PEGGY CARTER	1999-08-29	2	1
8000	YSA CAPACHO	1993-07-23	2	1
9000	MARIA MORALES	2003-07-07	4	1
17	VALERIA	1981-04-07	3	1
13	ANGELY	1978-06-21	2	1
7	SERGIO CAPACHO	1967-11-22	3	1
4020	YSAURA	2018-02-05	4	1
4017	MESSI	1996-06-09	4	1
4011	RONALDO	2011-12-25	1	1
9057	CARMENZA	1988-08-09	2	1
9026	FANNY	2005-07-10	2	1
10000	MERCY	2013-07-07	4	1

(18 filas)

Ahora bien, si realizamos una consulta que muestre todos los campos y todos los registros ordenados en forma ascendente (A) por el campo del número de cédula del empleado tenemos lo siguiente:

```
SELECT * FROM templeados
ORDER BY pkcedemple ;
```

```
bdretardados=# select * from templeados
bdretardados=# order by pkcedemple;
```

pkcedemple	nomemple	fecha	fkcodcar	fkcods
1000	ROBERTO JAIMES	1978-03-03	3	1
10000	MERCY	2013-07-07	4	1
13	ANGELY	1978-06-21	2	1
17	VALERIA	1981-04-07	3	1
2000	ZILA CONTRERAS	1980-10-05	2	1
3000	MARTHA	1996-11-22	3	1
4000	CARLOS	2000-07-15	4	1
4011	RONALDO	2011-12-25	1	1
4017	MESSI	1996-06-09	4	1
4020	YSAURA	2018-02-05	4	1
5000	MATIAS	2005-09-25	4	1
6000	NATASHA	2007-10-28	1	1
7	SERGIO CAPACHO	1967-11-22	3	1
7000	PEGGY CARTER	1999-08-29	2	1
8000	YSA CAPACHO	1993-07-23	2	1
9000	MARIA MORALES	2003-07-07	4	1
9026	FANNY	2005-07-10	2	1
9057	CARMENZA	1988-08-09	2	1

(18 filas)

Al revisar con detenimiento se observa que los números de cédula están en este orden: '1000', '10000', '13', '17', '2000', ..., '7', ...; Ud de una asumirá que le SQL está funcionando mal o algo escribió mal. Sin embargo, la respuesta es NO; ¿Entonces qué está pasando?. La cuestión es que el campo de la cédula es de tipo varchar(); es decir, el SQL no ve un número 1000, sino ve caracteres, SQL ve letras al estilo 'ABBB'. Por esta razón está ordenando de forma tan diferente.

Ahora bien, ¿Cómo obligo al SQL para que vea esos caracteres numéricos como un valor numérico y lograr ordenar en forma correcta?. Allí, es donde entra la función **CAST() o conversión temporal**. En nuestro caso esto funciona así para cada registro: se toma el contenido del campo varchar(), se convierte el contenido en forma temporal a un valor entero; y finalmente como son valores enteros, se ordena como valores numéricos tradicionales.

```
SELECT * FROM templeados ORDER BY CAST( pkcedemple AS Integer ) ;
```

```
SELECT * FROM templeados ORDER BY CAST( pkcedemple AS Int ) ;
```

```
SELECT * FROM templeados ORDER BY pkcedemple :: Int ;
```

```
SELECT * FROM templeados ORDER BY pkcedemple :: BIGINT ;
```

```
bdretardados=# SELECT * FROM templeados ORDER BY pkcedemple :: Int ;
```

pkcedemple	nomemple	fecha	fkcodcar	fkcods
7	SERGIO CAPACHO	1967-11-22	3	1
13	ANGELY	1978-06-21	2	1
17	VALERIA	1981-04-07	3	1
1000	ROBERTO JAIMES	1978-03-03	3	1
2000	ZILA CONTRERAS	1980-10-05	2	1
3000	MARTHA	1996-11-22	3	1
4000	CARLOS	2000-07-15	4	1
4011	RONALDO	2011-12-25	1	1
4017	MESSI	1996-06-09	4	1
4020	YSAURA	2018-02-05	4	1
5000	MATIAS	2005-09-25	4	1
6000	NATASHA	2007-10-28	1	1
7000	PEGGY CARTER	1999-08-29	2	1
8000	YSA CAPACHO	1993-07-23	2	1
9000	MARIA MORALES	2003-07-07	4	1
9026	FANNY	2005-07-10	2	1
9057	CARMENZA	1988-08-09	2	1
10000	MERCY	2013-07-07	4	1

(18 filas)

TABLA DE CONVERSIONES CAST()

Tipo de Dato Origen	Tipo de Dato Destino	Descripción de la Conversión	Ejemplo (SQL Standard)
Texto/Cadena (VARCHAR, NVARCHAR, CHAR, TEXT)	Números (INT, BIGINT, DECIMAL, NUMERIC, FLOAT)	Convierte una cadena de dígitos en un valor numérico. La cadena debe ser un número válido.	CAST('123' AS INT)
Texto/Cadena	Fecha y Hora (DATE, TIME, DATETIME, TIMESTAMP)	Convierte una cadena de texto en un formato de fecha u hora válido.	CAST('2025-11-21' AS DATE)
Texto/Cadena	Booleano (BOOLEAN/BIT)	Convierte 'true', 'false', '1', o '0' en un valor booleano.	CAST('true' AS BOOLEAN)
Números (INT, DECIMAL, etc.)	Texto/Cadena (VARCHAR, TEXT)	Convierte un valor numérico en su representación como cadena de texto.	CAST(12345 AS VARCHAR(10))
Números (INT, BIGINT)	Números con Decimales (DECIMAL, NUMERIC, FLOAT)	Convierte un entero a un número con precisión decimal.	CAST(10 AS DECIMAL(5, 2))
Fecha y Hora (DATE, DATETIME, TIMESTAMP)	Texto/Cadena (VARCHAR, TEXT)	Convierte un valor de fecha y hora a su representación textual.	CAST(GETDATE() AS VARCHAR(50))
DATETIME / TIMESTAMP	DATE	Extrae solo la parte de la fecha, eliminando la hora.	CAST(fecha_completa AS DATE)
DATETIME / TIMESTAMP	TIME	Extrae solo la parte de la hora, eliminando la fecha.	CAST(fecha_completa AS TIME)
Booleano (BOOLEAN/BIT)	Números (INT)	Convierte un booleano a 1 (true) o 0 (false).	CAST(col_booleana AS INT)

⚠ Consideraciones Importantes

- **Pérdida de Datos (Truncamiento):** Cuando conviertes un tipo de dato con mayor precisión o tamaño a uno menor (por ejemplo, de FLOAT a INT), puede ocurrir una pérdida de información (por ejemplo, se pierden los decimales).
- **Error de Conversión:** Si intentas convertir una cadena de texto que no representa un valor válido (por ejemplo, CAST('hola' AS INT)), la consulta fallará o devolverá un error. En algunos sistemas, puedes usar CONVERT() o funciones como TRY_CAST()/TRY_CONVERT().

Ejemplo 04: Convertir Texto a Número (Cálculos)

Este es el caso más común. Si tienes números almacenados como texto (VARCHAR o TEXT), no puedes realizar cálculos matemáticos directos. Debes convertirlos a un tipo numérico (INT, DECIMAL, FLOAT).

SELECT

producto,

CAST(precio_texto AS DECIMAL(10, 2)) * cantidad AS precio_total_calculado

FROM

Inventario ;

Explicación:

- Tienes una columna llamada **precio_texto** que almacena precios como cadenas de texto. Se quiere calcular el precio total multiplicando por la cantidad.
- **CAST(precio_texto AS DECIMAL(10, 2))** convierte la columna de texto **precio_texto** en un número decimal con **10** dígitos en total y **2** decimales.
- Esto permite que la operación de multiplicación (* cantidad) se realice correctamente.

Ejemplo 05: Convertir Fecha y Hora a solo Fecha (Comparaciones y Agrupaciones)

Cuando necesitas comparar o agrupar registros solo por la fecha, pero la columna original incluye la hora (DATETIME o TIMESTAMP), debes truncar la parte de la hora.

```
SELECT CAST( fecha_orden AS DATE ) AS solo_fecha, COUNT(*) AS total_ordenes
FROM Ordenes
GROUP BY CAST( fecha_orden AS DATE )
ORDER BY solo_fecha DESC ;
```

Explicación

- Quieres contar cuántas órdenes se realizaron en cada fecha, ignorando la hora exacta de la orden. Esto implica que el campo fecha_orden es de tipo TIMESTAMP.
- **CAST(fecha_orden AS DATE)** convierte el tipo DATETIME/TIMESTAMP a simplemente DATE, eliminando la información de la hora.
- Esto asegura que todas las órdenes del **mismo día** (p. ej., 2025-11-21 10:00:00 y 2025-11-21 15:30:00) se agrupen juntas en el GROUP BY.

Ejemplo 05: Convertir Número a Texto (Concatenación y Presentación)

A veces necesitas convertir un número a texto para poder concatenarlo con otras cadenas de texto o para asegurar un formato de salida específico.

```
SELECT 'El producto con ID: ' || CAST( id_producto AS VARCHAR(5) ) || ' está disponible.' AS descripcion_completa
```

```
FROM Productos
```

```
WHERE stock > 0 ;
```

Explicación

- Quieres generar una descripción de producto concatenando el ID del producto (que es un número) con un texto descriptivo.
- **Nota:** El operador de concatenación (||) puede variar. En **SQL Server** se usaría el operador +, y en **MySQL** se usaría la función CONCAT().
- **CAST(id_producto AS VARCHAR(5))** convierte el número entero **id_producto** a una cadena de texto de hasta 5 caracteres.
- Esto permite la **concatenación** con las otras cadenas de texto. Si no se hiciera el CAST(), la base de datos podría lanzar un error de tipo de dato.

Ejemplo 06:

```
SELECT SUM( CAST( monto AS INTEGER ) ) AS cantidad_pagos
```

```
FROM tdpagos
```

```
WHERE fkcc_alum = '1111';
```

5.10.- BETWEEN --> Permite filtrar todos los registros desde un valor inicial hasta un valor final. Los valores pueden ser números o fechas.

```
SELECT * FROM Tabla  
  
WHERE campo BETWEEN Valor_Inicial AND Valor_Final  
  
ORDER BY campo ASC/DESC ;
```

Ejemplo: Mostrar todos los registros de la tabla **tdretardos** que están entre el 01/11/2022 (Inclusive) y 30/11/2022 (Inclusive).

```
SELECT *  
FROM tdretardos  
WHERE fecha BETWEEN '2022/11/01' and '2022/11/30'  
  
ORDER BY fecha ;
```

```
bdretardados=# SELECT *  
bdretardados=# FROM tdretardos  
bdretardados=# WHERE fecha BETWEEN '2022/11/01' and '2022/11/30'  
bdretardados=# ORDER BY fecha ;
```

pknr	fkcedemple	fkcodexcu	fecha	hora	fkcods
110	1000	1	2022-11-01	08:20:00	1
123	6000	6	2022-11-03	09:05:00	1
111	4000	3	2022-11-03	08:35:00	1
112	6000	7	2022-11-05	09:11:00	1
113	5000	6	2022-11-07	08:21:00	1
116	7000	4	2022-11-10	08:27:00	1
119	6000	3	2022-11-12	08:40:00	1
114	3000	2	2022-11-13	08:17:00	1
115	2000	8	2022-11-15	08:12:00	1
117	6000	4	2022-11-16	09:15:00	1
124	6000	1	2022-11-19	09:05:00	1
118	1000	6	2022-11-19	09:19:00	1
120	6000	7	2022-11-26	08:37:00	1
121	6000	4	2022-11-27	08:25:00	1
97	7000	5	2022-11-27	08:20:00	1

(15 filas)

5.11.- IN --> Muestra solo los datos indicados en la lista. Permite ahorrar los conectivos lógicos OR.

SELECT * FROM Tabla

WHERE Pregunta1 ... AND/OR/NOT PreguntaN ... AND/OR/NOT

Campo **IN** (Valor1, Valor2, Valor3,, ValorN)

ORDER BY Campo1 ASC/DESC , ..., CampoN ASC/DESC ;

Ejemplo 01: Mostrar todos los campos y todos los registros de los retardos del empleado cuyo número de cédula es 6000; adicionalmente, solo se deben mostrar los registros donde las excusas presentadas correspondan al código de excusa: 2 o 4 o la 6. Y finalmente, los registros deben ser presentados ordenados en forma ascendente por el campo fecha.

SELECT *

FROM tdretardos

WHERE (fkcedemple = '6000') and fkcodexcu **IN** (2,4,6)

ORDER BY fecha ;

```
bdretardados=# SELECT *
bdretardados=# FROM tdretardos
bdretardados=# WHERE ( fkcedemple = '6000' ) and fkcodexcu IN (2,4,6)
bdretardados=# ORDER BY fecha ;
 pknr | fkcedemple | fkcodexcu | fecha | hora | fkcods
-----+-----+-----+-----+-----+-----
 62 | 6000 | 4 | 2022-01-05 | 08:32:00 | 1
 23 | 6000 | 6 | 2022-01-15 | 09:14:00 | 1
 69 | 6000 | 2 | 2022-01-21 | 08:24:00 | 1
 45 | 6000 | 6 | 2022-02-27 | 08:22:00 | 1
 34 | 6000 | 6 | 2022-05-27 | 08:22:00 | 1
 87 | 6000 | 6 | 2022-06-27 | 08:22:00 | 1
 86 | 6000 | 6 | 2022-06-27 | 08:22:00 | 1
 94 | 6000 | 6 | 2022-07-16 | 09:05:00 | 1
105 | 6000 | 4 | 2022-07-16 | 09:15:00 | 1
 96 | 6000 | 2 | 2022-07-26 | 08:17:00 | 1
123 | 6000 | 6 | 2022-11-03 | 09:05:00 | 1
117 | 6000 | 4 | 2022-11-16 | 09:15:00 | 1
121 | 6000 | 4 | 2022-11-27 | 08:25:00 | 1
(13 filas)
```

Ejemplo 02: El siguiente SQL selecciona todos los clientes que NO se encuentran en "Alemania", "Francia" o "Reino Unido":

```
SELECT * FROM tmclientes  
  
WHERE pais NOT IN ('Alemania', 'Francia', 'UK');
```

Ejemplo 03: El siguiente SQL selecciona todos los clientes que son de los mismos países que los proveedores:

```
SELECT * FROM tmclientes  
  
WHERE paises IN (SELECT países FROM tmproveedores);
```

Ejemplo 04:

```
SELECT *  
FROM tdretardos  
WHERE ( fecha BETWEEN '2022-11-01' and '2022-11-30' ) and fkcodexcu IN (2,4,6)  
ORDER BY fkcodexcu, fecha ;
```

```
bdretardados=# SELECT *  
bdretardados=# FROM tdretardos  
bdretardados=# WHERE ( fecha BETWEEN '2022-11-01' and '2022-11-30' ) and fkcodexcu IN (2,4,6)  
bdretardados=# ORDER BY fkcodexcu, fecha ;  
 pknr | fkcedemple | fkcodexcu | fecha | hora | fkods  
-----+-----+-----+-----+-----+-----  
 114 | 3000 | 2 | 2022-11-13 | 08:17:00 | 1  
 116 | 7000 | 4 | 2022-11-10 | 08:27:00 | 1  
 117 | 6000 | 4 | 2022-11-16 | 09:15:00 | 1  
 121 | 6000 | 4 | 2022-11-27 | 08:25:00 | 1  
 123 | 6000 | 6 | 2022-11-03 | 09:05:00 | 1  
 113 | 5000 | 6 | 2022-11-07 | 08:21:00 | 1  
 118 | 1000 | 6 | 2022-11-19 | 09:19:00 | 1  
(7 filas)
```

5.12.- LIKE: Se usa cuando se trabaja con patrones de cadena de texto

Nota: Hace diferencia entre mayúsculas y minúsculas.

```
SELECT Campo1, ....., CampoN
FROM Tabla1, ...., TablaN
```

```
WHERE Pregunta1 (And/Or/Not) ..... Campo LIKE 'Patron' ;
```

Ejemplo 1 (Texto completo): Se desean visualizar todos los campos y todos los registros donde el nombre del empleado sea NATASHA.

```
SELECT * FROM Templeados
```

```
WHERE nomemple LIKE 'NATASHA' ;
```

```
bdretardados=# select * from templeados;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----+-----+-----+-----+-----
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1
3000 | MARTHA | 1996-11-22 | 3 | 1
4000 | CARLOS | 2000-07-15 | 4 | 1
5000 | MATIAS | 2005-09-25 | 4 | 1
6000 | NATASHA | 2007-10-28 | 1 | 1
7000 | PEGGY CARTER | 1999-08-29 | 2 | 1
8000 | YSA CAPACHO | 1993-07-23 | 2 | 1
9000 | MARIA MORALES | 2003-07-07 | 4 | 1
(9 filas)

bdretardados=# SELECT * FROM Templeados
bdretardados=# WHERE nomemple LIKE 'NATASHA' ;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----+-----+-----+-----+-----
6000 | NATASHA | 2007-10-28 | 1 | 1
(1 fila)
```

Ejemplo 2 (% --> Cualquier cosa): Se desean visualizar todos los campos y todos los registros donde el nombre del empleado comienza con las letras MA y continúen con cualquier cosa.

```
SELECT * FROM Templeados
WHERE nomemple LIKE 'MA%' ;
```

```
bdretardados=# SELECT * FROM Templeados ;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----+-----+-----+-----+-----
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1
3000 | MARTHA | 1996-11-22 | 3 | 1
4000 | CARLOS | 2000-07-15 | 4 | 1
5000 | MATIAS | 2005-09-25 | 4 | 1
6000 | NATASHA | 2007-10-28 | 1 | 1
7000 | PEGGY CARTER | 1999-08-29 | 2 | 1
8000 | YSA CAPACHO | 1993-07-23 | 2 | 1
9000 | MARIA MORALES | 2003-07-07 | 4 | 1
(9 filas)

bdretardados=# SELECT * FROM Templeados
bdretardados=# WHERE nomemple LIKE 'MA%' ;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----+-----+-----+-----+-----
3000 | MARTHA | 1996-11-22 | 3 | 1
5000 | MATIAS | 2005-09-25 | 4 | 1
9000 | MARIA MORALES | 2003-07-07 | 4 | 1
(3 filas)
```

Ejemplo 3 (_ ---→ Un caracter): Se desean visualizar todos los campos y todos los registros donde el nombre del empleado tengan 6 caracteres como máximo.

```
SELECT * FROM Templeados
WHERE nomemple LIKE '_____';
```

```
bdretardados=# SELECT * FROM Templeados ;
```

pkcedemple	nomemple	fecha	fkcodcar	fkcods
1000	ROBERTO JAIMES	1978-03-03	3	1
2000	ZILA CONTRERAS	1980-10-05	2	1
3000	MARTHA	1996-11-22	3	1
4000	CARLOS	2000-07-15	4	1
5000	MATIAS	2005-09-25	4	1
6000	NATASHA	2007-10-28	1	1
7000	PEGGY CARTER	1999-08-29	2	1
8000	YSA CAPACHO	1993-07-23	2	1
9000	MARIA MORALES	2003-07-07	4	1

(9 filas)

```
bdretardados=# SELECT * FROM Templeados
WHERE nomemple LIKE '_____';
```

pkcedemple	nomemple	fecha	fkcodcar	fkcods
3000	MARTHA	1996-11-22	3	1
4000	CARLOS	2000-07-15	4	1
5000	MATIAS	2005-09-25	4	1

(3 filas)

Ejemplo 4: Se desean visualizar todos los campos y todos los registros donde el nombre del empleado comienza con cualquier letra, y además la segunda letra del nombre es **'A'**; y además, continua con cualquier otro carácter hasta el final.

```
SELECT * FROM Templeados
WHERE nomemple LIKE '_A%' ;
```

```
bdretardados=# SELECT * FROM Templeados ;
```

pkcedemple	nomemple	fecha	fkcodcar	fkcods
1000	ROBERTO JAIMES	1978-03-03	3	1
2000	ZILA CONTRERAS	1980-10-05	2	1
3000	MARTHA	1996-11-22	3	1
4000	CARLOS	2000-07-15	4	1
5000	MATIAS	2005-09-25	4	1
6000	NATASHA	2007-10-28	1	1
7000	PEGGY CARTER	1999-08-29	2	1
8000	YSA CAPACHO	1993-07-23	2	1
9000	MARIA MORALES	2003-07-07	4	1

(9 filas)

```
bdretardados=# SELECT * FROM Templeados
WHERE nomemple LIKE '_A%' ;
```

pkcedemple	nomemple	fecha	fkcodcar	fkcods
3000	MARTHA	1996-11-22	3	1
4000	CARLOS	2000-07-15	4	1
5000	MATIAS	2005-09-25	4	1
6000	NATASHA	2007-10-28	1	1
9000	MARIA MORALES	2003-07-07	4	1

(5 filas)

Ejemplo 5: Se desea visualizar todos los campos y todos los registros de aquellos empleados cuyo nombre tenga 6 caracteres; y además, el tercer carácter debe ser la letra 'R'..

```
SELECT * FROM Templeados
WHERE nomemple LIKE '__R__' ;
```

```
bdretardados=# SELECT * FROM Templeados ;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----+-----+-----+-----+-----
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1
3000 | MARTHA | 1996-11-22 | 3 | 1
4000 | CARLOS | 2000-07-15 | 4 | 1
5000 | MATIAS | 2005-09-25 | 4 | 1
6000 | NATASHA | 2007-10-28 | 1 | 1
7000 | PEGGY CARTER | 1999-08-29 | 2 | 1
8000 | YSA CAPACHO | 1993-07-23 | 2 | 1
9000 | MARIA MORALES | 2003-07-07 | 4 | 1
(9 filas)

bdretardados=# SELECT * FROM Templeados
WHERE nomemple LIKE '__R__' ;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----+-----+-----+-----+-----
3000 | MARTHA | 1996-11-22 | 3 | 1
4000 | CARLOS | 2000-07-15 | 4 | 1
(2 filas)
```

Ejemplo 6: Se desea visualizar todos los campos y todos los registros de aquellos empleados que dentro del nombre tengan la letra 'I', sin importar en que posición este la letra 'I'..

```
SELECT * FROM Templeados
WHERE nomemple LIKE '%I%' ;
```

```
bdretardados=# SELECT * FROM Templeados
WHERE nomemple LIKE '%I%' ;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----+-----+-----+-----+-----
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1
5000 | MATIAS | 2005-09-25 | 4 | 1
9000 | MARIA MORALES | 2003-07-07 | 4 | 1
(4 filas)
```

Probar iLIKE

5.13.- LIMIT: Limita la cantidad de filas que se pueden mostrar en una consulta.

Ejemplo 1: Se desean visualizar los primeros 10 registros de la TDRetardos.

```
SELECT * FROM tdretardos LIMIT 10 ;
```

```
bdretardados=# SELECT * FROM tdretardos LIMIT 10 ;
```

pknr	fkcedemple	fkcodexcu	fecha	hora	fkcods
1	2000	3	2022-04-05	08:32:00	1
2	5000	3	2022-04-05	08:50:00	1
3	4000	1	2022-04-06	09:09:00	1
4	3000	5	2022-04-06	09:13:00	1
5	6000	7	2022-04-11	08:21:00	1
6	1000	2	2022-04-14	08:16:00	1
7	2000	6	2022-04-17	10:00:00	1
8	3000	7	2022-04-19	08:11:00	1
9	2000	3	2022-04-21	08:13:00	1
10	1000	8	2022-04-21	08:15:00	1

(10 filas)

5.14.- OFFSET: Indica la cantidad de registros que se deben obviar al momento de ejecutar la consulta.

Ejemplo 01: Se desean visualizar de la tabla TMExcusas, todos los registros a partir del registro número 6 en adelante.

```
SELECT * FROM tmexcusas ;
```

```
SELECT * FROM tmexcusas OFFSET 5 ;
```

```
bdretardados=# SELECT * FROM tmexcusas ;
pkcodexcu |          dexcusa          | fkcods
-----+-----+-----
0 | ----- | 1
1 | ME QUEDE DORMIDO | 1
2 | SALI TARDE DE LA CASA | 1
3 | NO ME BUSCARON | 1
4 | PELIE CON EL AGUILA | 1
5 | MUCHA LLUVIA | 1
6 | MUCHO TRAFICO | 1
7 | EL CARRO NO ME QUIERE | 1
8 | ME SECUESTRARON | 1
(9 filas)

bdretardados=#
bdretardados=# SELECT * FROM tmexcusas OFFSET 5 ;
pkcodexcu |          dexcusa          | fkcods
-----+-----+-----
5 | MUCHA LLUVIA | 1
6 | MUCHO TRAFICO | 1
7 | EL CARRO NO ME QUIERE | 1
8 | ME SECUESTRARON | 1
(4 filas)
```

Ejemplo 02: Se desea visualizar 15 registro de la tdretrados, partiendo del registro número 8 en adelante.

```
SELECT * FROM tdretardos LIMIT 15 OFFSET 7;
```

```
bdretardados=# SELECT * FROM tdretardos LIMIT 15 OFFSET 7;
```

pknr	fkcedemple	fkcodexcu	fecha	hora	fkcods
8	3000	7	2022-04-19	08:11:00	1
9	2000	3	2022-04-21	08:13:00	1
10	1000	8	2022-04-21	08:15:00	1
11	5000	2	2022-04-21	08:20:00	1
12	2000	6	2022-04-23	08:27:00	1
13	3000	4	2022-04-23	03:30:00	1
14	4000	2	2022-04-23	08:45:00	1
15	6000	5	2022-04-25	08:05:00	1
16	2000	4	2022-04-25	08:32:00	1
17	4000	1	2022-04-27	09:00:00	1
18	6000	7	2022-04-28	09:00:00	1
19	1000	1	2022-01-07	10:43:00	1
20	3000	3	2022-01-07	10:50:00	1
21	4000	7	2022-01-10	09:00:00	1
22	5000	5	2022-01-12	08:20:00	1

(15 filas)

6.- Alias: Un alias **es un sinónimo** que se le asigna temporalmente al nombre del campo o al nombre de la tabla.

Ejemplo 01: En una consulta como la que tenemos a continuación, donde se muestran todos los campos y todos los registros de la TEMPLADOS. Si Ud observa con detenimiento el encabezado de las columnas, notara que son los nombres de los campos.

```
bdretardados=# SELECT * FROM templeados;
cedemple | nomemple | fecha | fkcodcar | fkcods
-----+-----+-----+-----+-----
1000 | ROBERTO | 1978-03-03 | 1 | 1
2000 | MARTHA | 1980-10-05 | 2 | 1
3000 | SANDRA | 1996-11-22 | 3 | 1
4000 | WILMARY | 2000-07-15 | 4 | 1
5000 | CARLOS | 2005-09-25 | 4 | 1
6000 | NATASHA | 2007-10-28 | 4 | 1
7000 | MARIA | 1999-08-29 | 4 | 1
(7 filas)
```

Ahora bien, si desea cambiar dichos títulos por un valor personalizado como por ejemplo:

- Cedemple ---→ Cédula
- Nomemple --→ Nombre
- Fkcodcar ----→ Cargo
- Fkcods-----→ Status

Entonces se usan alias.

SELECT pkcedemple **AS** "Cédula", nomemple **AS** "Nombre", fecha,
fkcodcar **AS** "Cargo", fkcods **AS** "Status"

FROM templeados ;

```
bdretardados=# SELECT pkcedemple AS "Cédula", nomemple AS "Nombre", fecha,
bdretardados=#          fkcodcar AS "Cargo", fkcods AS "Status"
bdretardados=# FROM templeados ;
Cédula | Nombre | fecha | Cargo | Status
-----+-----+-----+-----+-----
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1
3000 | MARTHA | 1996-11-22 | 3 | 1
4000 | CARLOS | 2000-07-15 | 4 | 1
5000 | MATIAS | 2005-09-25 | 4 | 1
6000 | NATASHA | 2007-10-28 | 1 | 1
7000 | PEGGY CARTER | 1999-08-29 | 2 | 1
8000 | YSA CAPACHO | 1993-07-23 | 2 | 1
9000 | MARIA MORALES | 2003-07-07 | 4 | 1
(9 filas)
```

Ejemplo 02: Se desea mostrar todos los campos y todos los registros de la combinación de las tablas TEMPLADOS y TMCARGOS

```
bdretardados=# SELECT * FROM templeados, tmcargos;
```

cedemple	nomemple	fecha	fkcodcar	fkcods	codcar	dcargo	sueldo	fkcods
1000	ROBERTO	1978-03-03	1	1	1	PRESIDENTE	5000000.00	1
2000	MARTHA	1980-10-05	2	1	1	PRESIDENTE	5000000.00	1
3000	SANDRA	1996-11-22	3	1	1	PRESIDENTE	5000000.00	1
4000	WILMARY	2000-07-15	4	1	1	PRESIDENTE	5000000.00	1
5000	CARLOS	2005-09-25	4	1	1	PRESIDENTE	5000000.00	1
6000	NATASHA	2007-10-28	4	1	1	PRESIDENTE	5000000.00	1
7000	MARIA	1999-08-29	4	1	1	PRESIDENTE	5000000.00	1
1000	ROBERTO	1978-03-03	1	1	2	VICE PRESIDENTE	4000000.00	1
2000	MARTHA	1980-10-05	2	1	2	VICE PRESIDENTE	4000000.00	1
3000	SANDRA	1996-11-22	3	1	2	VICE PRESIDENTE	4000000.00	1
4000	WILMARY	2000-07-15	4	1	2	VICE PRESIDENTE	4000000.00	1
5000	CARLOS	2005-09-25	4	1	2	VICE PRESIDENTE	4000000.00	1
6000	NATASHA	2007-10-28	4	1	2	VICE PRESIDENTE	4000000.00	1
7000	MARIA	1999-08-29	4	1	2	VICE PRESIDENTE	4000000.00	1
1000	ROBERTO	1978-03-03	1	1	3	GERENTE	3000000.00	1
2000	MARTHA	1980-10-05	2	1	3	GERENTE	3000000.00	1

Ahora bien; si se desea construir una consulta para mostrar los datos de los empleados cuyo código del estatus sea 1, se genera un error de un campo ambiguo. Esto ocurre porque la consulta tiene dos campos llamados **fkcods**, el primero pertenece a TEMPLADOS y el segundo pertenece a TMCARGOS, lo cual al momento de construir una pregunta con el campo **fkcods** se genera el siguiente conflicto en el servidor: ¿Con cuál **fkcods** voy a trabajar, con el campo **fkcods** de la tabla TEMPLADOS o con el campo **fkcods** de la TMCARGOS?. La solución es asignar alias a cada tabla así:

```
bdretardados=# SELECT cedemple, nomemple, fkcodcar, codcar, dcargo, M.fkcods FROM templeados AS M, tmcargos AS T
bdretardados=# WHERE M.fkcods=1;
```

cedemple	nomemple	fkcodcar	codcar	dcargo	fkcods
1000	ROBERTO	1	1	PRESIDENTE	1
2000	MARTHA	2	1	PRESIDENTE	1
3000	SANDRA	3	1	PRESIDENTE	1
4000	WILMARY	4	1	PRESIDENTE	1
5000	CARLOS	4	1	PRESIDENTE	1
6000	NATASHA	4	1	PRESIDENTE	1
7000	MARIA	4	1	PRESIDENTE	1
1000	ROBERTO	1	2	VICE PRESIDENTE	1
2000	MARTHA	2	2	VICE PRESIDENTE	1
3000	SANDRA	3	2	VICE PRESIDENTE	1
4000	WILMARY	4	2	VICE PRESIDENTE	1
5000	CARLOS	4	2	VICE PRESIDENTE	1
6000	NATASHA	4	2	VICE PRESIDENTE	1
7000	MARIA	4	2	VICE PRESIDENTE	1
1000	ROBERTO	1	3	GERENTE	1

Y finalmente para terminar de construir la consulta, realizamos lo siguiente:

```
SELECT pkcedemple, nomemple, dcargo, M.fkcods
FROM templeados AS M, tmcargos AS T
WHERE M.fkcods = 1 and M.fkcodcar = T.pkcodcar ;
```

```
bdretardados=# SELECT pkcedemple, nomemple, dcargo, M.fkcods
bdretardados=# FROM templeados AS M, tmcargos AS T
bdretardados=# WHERE M.fkcods = 1 and M.fkcodcar = T.pkcodcar
pkcedemple | nomemple | dcargo | fkcods
-----+-----+-----+-----
1000 | ROBERTO JAIMES | GERENTE | 1
2000 | ZILA CONTRERAS | VICE PRESIDENTE | 1
3000 | MARTHA | GERENTE | 1
4000 | CARLOS | SUB-GERENTE | 1
5000 | MATIAS | SUB-GERENTE | 1
6000 | NATASHA | PRESIDENTE | 1
7000 | PEGGY CARTER | VICE PRESIDENTE | 1
8000 | YSA CAPACHO | VICE PRESIDENTE | 1
9000 | MARIA MORALES | SUB-GERENTE | 1
(9 filas)
```

```
SELECT pkcedemple, nomemple, dcargo, dstatus
FROM templeados AS M, tmcargos AS T, tmstatus AS X
WHERE M.fkcods = X.pkcods and M.fkcodcar = T.pkcodcar ;
```

```
bdretardados=# SELECT pkcedemple, nomemple, dcargo, dstatus
bdretardados=# FROM templeados AS M, tmcargos AS T, tmstatus AS X
bdretardados=# WHERE M.fkcods = X.pkcods and M.fkcodcar = T.pkcodcar ;
pkcedemple | nomemple | dcargo | dstatus
-----+-----+-----+-----
1000 | ROBERTO JAIMES | GERENTE | ACTIVO
2000 | ZILA CONTRERAS | VICE PRESIDENTE | ACTIVO
3000 | MARTHA | GERENTE | ACTIVO
4000 | CARLOS | SUB-GERENTE | ACTIVO
5000 | MATIAS | SUB-GERENTE | ACTIVO
6000 | NATASHA | PRESIDENTE | ACTIVO
7000 | PEGGY CARTER | VICE PRESIDENTE | ACTIVO
8000 | YSA CAPACHO | VICE PRESIDENTE | ACTIVO
9000 | MARIA MORALES | SUB-GERENTE | ACTIVO
(9 filas)
```

Otra forma de crear un alias es usando el formato **NOMBRE_TABLA.NOMBRE_CAMPO**, como se observa en el siguiente ejemplo:

```
SELECT pkcedemple, nomemple, dcargo, TEMPLADOS.FKCODS
FROM templeados, tmcargos
WHERE ( TEMPLADOS.FKCODS = 1 )
```

```
and ( TEMPLADOS.FKCODCAR = TMCARGOS.PKCODCAR ) ;
```

```
bdretardados=# SELECT pkcedemple, nomemple, dcargo, TEMPLADOS.FKCODS
bdretardados=# FROM templeados, tmcargos
bdretardados=# WHERE ( TEMPLADOS.FKCODS = 1 )
bdretardados=# and ( TEMPLADOS.FKCODCAR = TMCARGOS.PKCODCAR ) ;
 pkcedemple | nomemple | dcargo | fkcods
-----|-----|-----|-----
 1000      | ROBERTO JAIMES | GERENTE | 1
 2000      | ZILA CONTRERAS | VICE PRESIDENTE | 1
 3000      | MARTHA | GERENTE | 1
 4000      | CARLOS | SUB-GERENTE | 1
 5000      | MATIAS | SUB-GERENTE | 1
 6000      | NATASHA | PRESIDENTE | 1
 7000      | PEGGY CARTER | VICE PRESIDENTE | 1
 8000      | YSA CAPACHO | VICE PRESIDENTE | 1
 9000      | MARIA MORALES | SUB-GERENTE | 1
(9 filas)
```

```
SELECT pkcedemple, nomemple, dcargo, dstatus
FROM templeados, tmcargos, tmstatus
WHERE ( TEMPLADOS.fkcods = TMSTATUS.pkcods )
```

```
and ( TEMPLADOS.fkcodcar = TMCARGOS.pkcodcar ) ;
```

```
bdretardados=# SELECT pkcedemple, nomemple, dcargo, dstatus
bdretardados=# FROM templeados, tmcargos, tmstatus
bdretardados=# WHERE ( TEMPLADOS.fkcods = TMSTATUS.pkcods )
bdretardados=# and ( TEMPLADOS.fkcodcar = TMCARGOS.pkcodcar ) ;
 pkcedemple | nomemple | dcargo | dstatus
-----|-----|-----|-----
 1000      | ROBERTO JAIMES | GERENTE | ACTIVO
 2000      | ZILA CONTRERAS | VICE PRESIDENTE | ACTIVO
 3000      | MARTHA | GERENTE | ACTIVO
 4000      | CARLOS | SUB-GERENTE | ACTIVO
 5000      | MATIAS | SUB-GERENTE | ACTIVO
 6000      | NATASHA | PRESIDENTE | ACTIVO
 7000      | PEGGY CARTER | VICE PRESIDENTE | ACTIVO
 8000      | YSA CAPACHO | VICE PRESIDENTE | ACTIVO
 9000      | MARIA MORALES | SUB-GERENTE | ACTIVO
(9 filas)
```

7.- Formato numérico --> To_Char(Número, '999G999G999D99')

Toma el contenido del campo numérico y lo convierte en una salida de formato numérico 999,999,999.99

```
SELECT Campo1, ..., to_char( Campo_Numérico, '999G999G999D99' )  
FROM tabla
```

```
WHERE Condición – criterio ORDER BY Campo1, CampoN... .;
```

Ejemplo 01: Se desea mostrar todos los registros de la tabla tmcargos, donde el sueldo se debe mostrar con el formato numérico 999.999.999,99

```
SELECT pkcodcar, dcargo, to_char( sueldo, '999G999G999D99' ) AS "Sueldo"  
FROM tmcargos ;
```

```
bdretardados=# select * from tmcargos;  
pkcodcar |      dcargo      |      sueldo      | fkcods  
-----+-----+-----+-----  
1 | PRESIDENTE      | 5000000.01      | 1  
2 | VICE PRESIDENTE | 4500000.02      | 0  
3 | GERENTE         | 4000000.03      | 1  
4 | SUB-GERENTE     | 3500000.04      | 0  
5 | COORDINADOR     | 3000000.05      | 1  
6 | SUPERVISOR      | 2500000.06      | 0  
(6 filas)
```

```
bdretardados=# SELECT pkcodcar, dcargo, to_char( sueldo, '999G999G999D99' ) AS "Sueldo"  
bdretardados=# FROM tmcargos ;  
pkcodcar |      dcargo      |      Sueldo      |  
-----+-----+-----+-----  
1 | PRESIDENTE      | 5.000.000,01    |  
2 | VICE PRESIDENTE | 4.500.000,02    |  
3 | GERENTE         | 4.000.000,03    |  
4 | SUB-GERENTE     | 3.500.000,04    |  
5 | COORDINADOR     | 3.000.000,05    |  
6 | SUPERVISOR      | 2.500.000,06    |  
(6 filas)
```

Nota (GD): Las unidades de miles, millones... con punto(.); el separador decimal (,)

Ejemplo 02: Se desea mostrar todos los registros de la tabla tmcargos, donde el sueldo se debe mostrar con el formato numérico de moneda así 999,999,999.99

```
bdretardados=# select * from tmcargos;
pkcodcar |      dcargo      |      sueldo      | fkcods
-----+-----+-----+-----
1 | PRESIDENTE      | 5000000.01      | 1
2 | VICE PRESIDENTE | 4500000.02      | 0
3 | GERENTE         | 4000000.03      | 1
4 | SUB-GERENTE     | 3500000.04      | 0
5 | COORDINADOR     | 3000000.05      | 1
6 | SUPERVISOR      | 2500000.06      | 0
(6 filas)
```

`SELECT pkcodcar, dcargo, to_char(sueldo, 'FM999,999,999.99') AS "Sueldo"`
`FROM tmcargos ;`

```
bdretardados=# SELECT pkcodcar, dcargo, to_char( sueldo, 'FM999,999,999.99' ) AS "Sueldo"
bdretardados=# FROM tmcargos ;
pkcodcar |      dcargo      |      Sueldo
-----+-----+-----
1 | PRESIDENTE      | 5,000,000.01
2 | VICE PRESIDENTE | 4,500,000.02
3 | GERENTE         | 4,000,000.03
4 | SUB-GERENTE     | 3,500,000.04
5 | COORDINADOR     | 3,000,000.05
6 | SUPERVISOR      | 2,500,000.06
(6 filas)
```

Nota (FM): Las unidades de miles, millardos... con coma(,); el separador decimal punto(.)

Ejemplo 03: Se desea mostrar la sumatoria de todos los pagos hecho por el estudiante cuya cédula es '1000'. Donde el sueldo se debe mostrar con el **formato numérico de moneda** así 999,999,999.99

`SELECT to_char(SUM(monto), 'FM999,999,999.99') AS "Monto Pagado:"`
`FROM tdpagos`

`WHERE fkcedalum = '1000'` ;  `Monto Pagado: 9,000,000.00`

```
bdretardados=# SELECT pkcodcar, dcargo, to_char( sueldo, 'FM999,999,999.99' ) AS
"Sueldo"
bdretardados=# FROM tmcargos ;
```

pkcodcar	dcargo	Sueldo
1	PRESIDENTE	5,000,000.01
2	VICE PRESIDENTE	4,500,000.02
3	GERENTE	4,000,000.03
4	SUB-GERENTE	3,500,000.04
5	COORDINADOR	3,000,000.05
6	SUPERVISOR	2,500,000.06

(6 filas)

```
bdretardados=# select * from tmcargos;
pkcodcar | dcargo | sueldo | fkcods
```

pkcodcar	dcargo	sueldo	fkcods
1	PRESIDENTE	5000000.01	1
2	VICE PRESIDENTE	4500000.02	0
3	GERENTE	4000000.03	1
4	SUB-GERENTE	3500000.04	0
5	COORDINADOR	3000000.05	1
6	SUPERVISOR	2500000.06	0

(6 filas)

Nota (FM): Las unidades de miles, millardos... con coma(,); el separador decimal punto(.)

Ejemplo 04:

`SELECT pkcodcar, dcargo, to_char(sueldo, 'L999G999G999D99') AS "Sueldo"`

`FROM tmcargos` ;

```
bdretardados=# SELECT pkcodcar, dcargo, to_char( sueldo, 'L999G999G999D99' ) AS "Sueldo"
bdretardados=# FROM tmcargos ;
```

pkcodcar	dcargo	Sueldo
1	PRESIDENTE	\$ 5.000.000,01
2	VICE PRESIDENTE	\$ 4.500.000,02
3	GERENTE	\$ 4.000.000,03
4	SUB-GERENTE	\$ 3.500.000,04
5	COORDINADOR	\$ 3.000.000,05
6	SUPERVISOR	\$ 2.500.000,06

(6 filas)

Nota (LG): L Símbolo de pesos \$; Las unidades de miles, millardos... con punto(.); el separador decimal coma (,)

8.- Formato fecha: To_Char(Campo_Fecha, 'DD/MM/YYYY')

```
SELECT Campo1, ... , to_char( Campo_Fecha, 'DD/MM/YYYY' )  
FROM tabla
```

```
WHERE Pregunta1, ..., PreguntaN ORDER BY Campo1, CampoN... ;
```

Ejemplo 04:

```
SELECT pkcedemple, nomemple, to_char( fecha, 'DD/MM/YYYY' ) As "Fecha"  
FROM templeados ;
```

```
bdretardados=# select pkcedemple, nomemple, fecha  
bdretardados=# from templeados;  
pkcedemple | nomemple | fecha  
-----+-----+-----  
1000 | ROBERTO JAIMES | 1978-03-03  
2000 | ZILA CONTRERAS | 1980-10-05  
3000 | MARTHA | 1996-11-22  
4000 | CARLOS | 2000-07-15  
5000 | MATIAS | 2005-09-25  
6000 | NATASHA | 2007-10-28  
7000 | PEGGY CARTER | 1999-08-29  
8000 | YSA CAPACHO | 1993-07-23  
9000 | MARIA MORALES | 2003-07-07  
17 | VALERIA | 1981-04-07  
13 | ANGELY | 1978-06-21  
7 | SERGIO CAPACHO | 1967-11-22  
4020 | YSAURA | 2018-02-05  
4017 | MESSI | 1996-06-09  
4011 | RONALDO | 2011-12-25  
9057 | CARMENZA | 1988-08-09  
9026 | FANNY | 2005-07-10  
10000 | MERCY | 2013-07-07  
(18 filas)
```

```
bdretardados=# SELECT pkcedemple, nomemple,  
bdretardados=# to_char( fecha, 'DD/MM/YYYY' ) As "Fecha"  
bdretardados=# FROM templeados;  
pkcedemple | nomemple | Fecha  
-----+-----+-----  
1000 | ROBERTO JAIMES | 03/03/1978  
2000 | ZILA CONTRERAS | 05/10/1980  
3000 | MARTHA | 22/11/1996  
4000 | CARLOS | 15/07/2000  
5000 | MATIAS | 25/09/2005  
6000 | NATASHA | 28/10/2007  
7000 | PEGGY CARTER | 29/08/1999  
8000 | YSA CAPACHO | 23/07/1993  
9000 | MARIA MORALES | 07/07/2003  
17 | VALERIA | 07/04/1981  
13 | ANGELY | 21/06/1978  
7 | SERGIO CAPACHO | 22/11/1967  
4020 | YSAURA | 05/02/2018  
4017 | MESSI | 09/06/1996  
4011 | RONALDO | 25/12/2011  
9057 | CARMENZA | 09/08/1988  
9026 | FANNY | 10/07/2005  
10000 | MERCY | 07/07/2013  
(18 filas)
```

9.- Sub – Cálculos:

```
SELECT Campo_Numerico, ....., ( Campo_Numerico + Valor ),  
      ( Campo_Numerico - Valor ), ( Campo_Numerico * Valor ) ,  
      ( Campo_Numerico / Valor )  
FROM Tabla1, .... , TablaN  
WHERE Pregunta1 (And/Or/Not) ... PreguntaN  
ORDER BY Campo1, ....., CampoN ;
```

Ejemplo 01: Se desean visualizar de la tmcargos los siguientes datos: código del cargo, descripción del cargo, y el sueldo mensual. Además, también se necesario visualizar una proyección de: un bono del 2% del sueldo base, una deducción del 5% del sueldo base y el sueldo neto.

Formato Numérico: **to_char**(Campo_Numérico, '999G999G999D99')

```
SELECT pkcodcar, dcargo,  
      to_char( sueldo, '999G999G999D99' ) As "Sueldo",  
      to_char( (sueldo*0.02), '999G999G999D99' ) AS "Bono",  
      to_char( (sueldo*0.05), '999G999G999D99' ) As "Deducción",  
      to_char( (sueldo + (sueldo * 0.02) - (sueldo * 0.05)), '999G999G999D99' ) As "SNeto",  
      fkcods  
FROM tmcargos ;
```

```
bdretardados=# SELECT pkcodcar, dcargo,  
bdretardados=#         to_char( sueldo, '999G999G999D99' ) As "Sueldo",  
bdretardados=#         to_char( (sueldo*0.02), '999G999G999D99' ) AS "Bono",  
bdretardados=#         to_char( (sueldo*0.05), '999G999G999D99' ) As "Deducción",  
bdretardados=#         to_char( (sueldo + (sueldo * 0.02) - (sueldo * 0.05)), '999G999G999D99' ) As "SNeto",  
bdretardados=#         fkcods  
bdretardados=# FROM tmcargos  
pkcodcar | dcargo | Sueldo | Bono | Deducción | SNeto | fkcods  
-----+-----+-----+-----+-----+-----+-----  
1 | PRESIDENTE | 5.000.000,01 | 100.000,00 | 250.000,00 | 4.850.000,01 | 1  
2 | VICE PRESIDENTE | 4.500.000,02 | 90.000,00 | 225.000,00 | 4.365.000,02 | 0  
3 | GERENTE | 4.000.000,03 | 80.000,00 | 200.000,00 | 3.880.000,03 | 1  
4 | SUB-GERENTE | 3.500.000,04 | 70.000,00 | 175.000,00 | 3.395.000,04 | 0  
5 | COORDINADOR | 3.000.000,05 | 60.000,00 | 150.000,00 | 2.910.000,05 | 1  
6 | SUPERVISOR | 2.500.000,06 | 50.000,00 | 125.000,00 | 2.425.000,06 | 0  
(6 filas)
```

10.- Extraer Año / Mes / Día

Ejemplo 01: Extraer el día, mes y año de nacimiento de todos los empleados

```
SELECT pkcedemple, nomemple, fecha,  
        EXTRACT( DAY FROM fecha ) As "Día",  
        EXTRACT( MONTH FROM fecha ) As "Mes",  
        EXTRACT( YEAR FROM fecha ) As "Año"  
  
FROM Templeados ;
```

```
bdretardados=# SELECT  pkcedemple, nomemple, fecha,  
bdretardados=#          EXTRACT( DAY FROM fecha)  As "Día",  
bdretardados=#          EXTRACT( MONTH FROM fecha) As "Mes",  
bdretardados=#          EXTRACT( YEAR FROM fecha)  As "Año"  
bdretardados=# FROM    Templeados ;
```

pkcedemple	nomemple	fecha	Día	Mes	Año
1000	ROBERTO JAIMES	1978-03-03	3	3	1978
2000	ZILA CONTRERAS	1980-10-05	5	10	1980
3000	MARTHA	1996-11-22	22	11	1996
4000	CARLOS	2000-07-15	15	7	2000
5000	MATIAS	2005-09-25	25	9	2005
6000	NATASHA	2007-10-28	28	10	2007
7000	PEGGY CARTER	1999-08-29	29	8	1999
8000	YSA CAPACHO	1993-07-23	23	7	1993
9000	MARIA MORALES	2003-07-07	7	7	2003
17	VALERIA	1981-04-07	7	4	1981
13	ANGELY	1978-06-21	21	6	1978
7	SERGIO CAPACHO	1967-11-22	22	11	1967
4020	YSAURA	2018-02-05	5	2	2018
4017	MESSI	1996-06-09	9	6	1996
4011	RONALDO	2011-12-25	25	12	2011
9057	CARMENZA	1988-08-09	9	8	1988
9026	FANNY	2005-07-10	10	7	2005
10000	MERCY	2013-07-07	7	7	2013

(18 filas)

Ejemplo 02: Calcular la edad actual de cada empleado

SELECT NOW() AS "Fecha Actual" ;

```
bdretardados=# SELECT NOW() AS "Fecha Actual";
Fecha Actual
-----
2025-11-21 15:05:43.634162-05
(1 fila)
```

SELECT pkcedemple, nomemple, fecha As "FeNac",
EXTRACT(YEAR FROM Now()) As "Año Actual",
(EXTRACT(YEAR FROM Now()) - EXTRACT(YEAR FROM fecha)) As

Edad

FROM Templeados

ORDER BY Edad **DESC**, CAST (pkcedemple AS **BIGINT**) ;

```
bdretardados=# SELECT pkcedemple, nomemple, fecha As "FeNac",
bdretardados=# EXTRACT( YEAR FROM Now() ) As "Año Actual",
bdretardados=# ( EXTRACT( YEAR FROM Now() ) - EXTRACT( YEAR FROM fecha ) ) As Edad
bdretardados=# FROM Templeados
bdretardados=# ORDER BY Edad DESC, CAST ( pkcedemple AS BIGINT) ;
```

pkcedemple	nomemple	FeNac	Año Actual	edad
7	SERGIO CAPACHO	1967-11-22	2025	58
13	ANGELY	1978-06-21	2025	47
1000	ROBERTO JAIMES	1978-03-03	2025	47
2000	ZILA CONTRERAS	1980-10-05	2025	45
17	VALERIA	1981-04-07	2025	44
9057	CARMENZA	1988-08-09	2025	37
8000	YSA CAPACHO	1993-07-23	2025	32
3000	MARTHA	1996-11-22	2025	29
4017	MESSI	1996-06-09	2025	29
7000	PEGGY CARTER	1999-08-29	2025	26
4000	CARLOS	2000-07-15	2025	25
9000	MARIA MORALES	2003-07-07	2025	22
5000	MATIAS	2005-09-25	2025	20
9026	FANNY	2005-07-10	2025	20
6000	NATASHA	2007-10-28	2025	18
4011	RONALDO	2011-12-25	2025	14
10000	MERCY	2013-07-07	2025	12
4020	YSAURA	2018-02-05	2025	7

(18 filas)

11.- Funciones de agregación

11.01.- Función Count --→ Retorna el número de registros que cumplen con la condición o el criterio dado.

```
SELECT COUNT(campo1), ..... , COUNT(campoN)  
FROM tabla
```

```
WHERE condición – criterio ;
```

Ejemplo 1: determinar cuántos registros existen con el código de la excusa es 3 en la `tdretardos`.

```
SELECT COUNT(fkcodexcu)  
FROM tdretardos
```

```
WHERE fkcodexcu = 3 ;
```

```
bdretardados=# \d tdretardos
          Tabla public.tdretardos
-----
```

Columna	Tipo	Ordenamiento	Nulable	Por omisión
<code>pknr</code>	<code>integer</code>		<code>not null</code>	<code>nextval('tdretardos_pknr_seq'::regclass)</code>
<code>fkcedemple</code>	<code>character varying(12)</code>		<code>not null</code>	
<code>fkcodexcu</code>	<code>integer</code>		<code>not null</code>	
<code>fecha</code>	<code>date</code>		<code>not null</code>	
<code>hora</code>	<code>time without time zone</code>		<code>not null</code>	
<code>fkcods</code>	<code>integer</code>		<code>not null</code>	<code>1</code>

```
-----
Índices:
"tdretardos_pkey" PRIMARY KEY, btree (pknr)
Restricciones de llave foránea:
"tdretardos_fkcedemple_fkey" FOREIGN KEY (fkcedemple) REFERENCES templeados(pkcedemple) ON UPDATE CASCADE ON DELETE RESTRICT
"tdretardos_fkcodexcu_fkey" FOREIGN KEY (fkcodexcu) REFERENCES texcusas(pkcodexcu) ON UPDATE CASCADE ON DELETE RESTRICT
"tdretardos_fkcods_fkey" FOREIGN KEY (fkcods) REFERENCES tstatus(pkcods) ON UPDATE CASCADE ON DELETE RESTRICT

bdretardados=# SELECT COUNT(fkcodexcu)
bdretardados=# FROM tdretardos
bdretardados=# WHERE fkcodexcu = 3 ;
count
-----
19
(1 fila)
```

11.02.- Función AVG --> Retorna el valor promedio los registros que cumplen con la condición o el criterio dado.

Nota: El campo debe ser numérico

```
SELECT AVG(Campo1) , ..... , AVG(Campo2)
FROM tabla
```

```
WHERE condición – criterio ;
```

Ejemplo 1: obtener el valor promedio de los sueldos cuyo código del cargo es 3 en la TMCARGOS.

Formato Numérico: **to_char**(Campo_Numérico, '999G999G999D99')

```
SELECT to_char( AVG(sueldo), '999G999G999D99' )
FROM tmcargos
```

```
WHERE pkcodcar = 3 ;
```

```
bdretardados=# select * from tmcargos;
 pkcodcar |      dcargo      |      sueldo      |  fkcods
-----+-----+-----+-----
      1 | PRESIDENTE      | 5000000.00      |      1
      2 | VICE PRESIDENTE | 4500000.00      |      0
      3 | GERENTE         | 4000000.00      |      1
      4 | SUB-GERENTE     | 3500000.00      |      0
      5 | COORDINADOR     | 3000000.00      |      1
      6 | SUPERVISOR      | 2500000.00      |      0
(6 filas)

bdretardados=# SELECT to_char( AVG(sueldo), '999G999G999D99' )
bdretardados=# FROM tmcargos
bdretardados=# WHERE pkcodcar = 3 ;
 to_char
-----
 4.000.000,00
(1 fila)
```

11.03.- Función SUM --> Retorna la sumatoria de todos los registros que cumplen con la condición o el criterio dado. **Nota:** El campo debe ser numérico

```
SELECT SUM(Campo1) , ..... , SUM(Campo2)
FROM tabla
```

```
WHERE condición – criterio ;
```

Ejemplo 1: obtener la sumatoria de todos los sueldos cuyo código del cargo es 2 en la TMCARGOS.

Formato Numérico: **to_char**(Campo_Numérico, '999G999G999D99')

```
SELECT to_char( SUM(sueldo) , '999G999G999D99' )
FROM tmcargos
```

```
WHERE pkcodcar = 2 ;
```

```
bdretardados=# select * from tmcargos;
pkcodcar |      dcargo      |      sueldo      | fkcods
-----+-----+-----+-----
      1 | PRESIDENTE      | 5000000.00      |      1
      2 | VICE PRESIDENTE | 4500000.00      |      0
      3 | GERENTE         | 4000000.00      |      1
      4 | SUB-GERENTE     | 3500000.00      |      0
      5 | COORDINADOR     | 3000000.00      |      1
      6 | SUPERVISOR      | 2500000.00      |      0
(6 filas)

bdretardados=# SELECT to_char( SUM(sueldo) , '999G999G999D99' )
bdretardados=# FROM tmcargos
bdretardados=# WHERE pkcodcar = 2 ;
to_char
-----
 4.500.000,00
(1 fila)
```

11.04.- Función MAX --> Retorna el valor máximo que existe en los registros que cumplen con la condición o el criterio dado. **Nota:** El campo debe ser numérico

```
SELECT MAX(Campo1), ..... , MAX(Campo2)
FROM tabla

WHERE condición – criterio ;
```

Ejemplo 1: obtener el valor máximo de todos los sueldos de la **TMCARGOS**.

Formato Numérico: **to_char**(Campo_Numérico, '999G999G999D99')

```
SELECT to_char( MAX(sueldo) , '999G999G999D99' )
FROM tmcargos ;
```

```
bdretardados=# SELECT * FROM TMCARGOS;
pkcodcar |      dcargo      |      sueldo      | fkcods
-----+-----+-----+-----
1 | PRESIDENTE      | 5000000.00      | 1
2 | VICE PRESIDENTE | 4500000.00      | 0
3 | GERENTE         | 4000000.00      | 1
4 | SUB-GERENTE     | 3500000.00      | 0
5 | COORDINADOR     | 3000000.00      | 1
6 | SUPERVISOR      | 2500000.00      | 0
(6 filas)

bdretardados=# SELECT to_char( MAX(sueldo) , '999G999G999D99' )
bdretardados=# FROM tmcargos ;
to_char
-----
5.000.000,00
(1 fila)
```

11.05.- Función MIN --→ Retorna el valor mínimo que existe en los registros que cumplen con la condición o el criterio. **Nota:** El campo debe ser numérico

```
SELECT MIN(Campo1) , ..... , MIN(Campo1),  
FROM tabla
```

```
WHERE condición – criterio ;
```

Ejemplo 1: obtener el valor mínimo de todos los sueldos de la **TMCARGOS**.

Formato Numérico: **to_char**(Campo_Numérico, '999G999G999D99')

```
SELECT to_char( MIN(sueldo) , '999G999G999D99' )
```

```
FROM tmcargos ;
```

```
bdretardados=# SELECT * FROM TMCARGOS;  
pkcodcar |      dcargo      |      sueldo      |      fkcods  
-----+-----+-----+-----  
1 | PRESIDENTE | 5000000.00 | 1  
2 | VICE PRESIDENTE | 4500000.00 | 0  
3 | GERENTE | 4000000.00 | 1  
4 | SUB-GERENTE | 3500000.00 | 0  
5 | COORDINADOR | 3000000.00 | 1  
6 | SUPERVISOR | 2500000.00 | 0  
(6 filas)  
  
bdretardados=# SELECT to_char( MIN(sueldo) , '999G999G999D99' )  
bdretardados=# FROM tmcargos ;  
to_char  
-----  
2.500.000,00  
(1 fila)
```

12.- Agrupación

12.01.- GROUP BY --> Permite realizar una consulta agrupando los datos por bloques de datos.

SELECT Campo_Grupo1, ..., Campo_GrupoN

FROM Tabla1, ..., TablaN

GROUP BY Campo_Grupo1, ..., Campo_GrupoN ;

Ejemplo 01: Dada la tabla TDRETARDOS

```
bdretardados=# \d tdretardos
          Tabla public.tdretardos
Columna | Tipo | Ordenamiento | Nulable | Por omisión
-----+-----+-----+-----+-----
pknr    | integer |                | not null | nextval('tdretardos_pknr_seq'::regclass)
fkcedemple | character varying(12) |                | not null |
fkcodexcu | integer |                | not null |
fecha   | date |                | not null |
hora    | time without time zone |                | not null |
fkcods  | integer |                | not null | 1
Índices:
"tdretardos_pkey" PRIMARY KEY, btree (pknr)
Restricciones de llave foránea:
"tdretardos_fkcedemple_fkey" FOREIGN KEY (fkcedemple) REFERENCES templeados(pkcedemple) ON UPDATE CASCADE ON DELETE RESTRICT
"tdretardos_fkcodexcu_fkey" FOREIGN KEY (fkcodexcu) REFERENCES tmexcusas(pkcodexcu) ON UPDATE CASCADE ON DELETE RESTRICT
"tdretardos_fkcods_fkey" FOREIGN KEY (fkcods) REFERENCES tmstatus(pkcods) ON UPDATE CASCADE ON DELETE RESTRICT
```

```
bdretardados=# select * from tdretardos;
pknr | fkcedemple | fkcodexcu | fecha | hora | fkcods
-----+-----+-----+-----+-----+-----
  1 | 2000 | 3 | 2022-04-05 | 08:32:00 | 1
  2 | 5000 | 3 | 2022-04-05 | 08:50:00 | 1
  3 | 4000 | 1 | 2022-04-06 | 09:09:00 | 1
  4 | 3000 | 5 | 2022-04-06 | 09:13:00 | 1
  5 | 6000 | 7 | 2022-04-11 | 08:21:00 | 1
  6 | 1000 | 2 | 2022-04-14 | 08:16:00 | 1
  7 | 2000 | 6 | 2022-04-17 | 10:00:00 | 1
  8 | 3000 | 7 | 2022-04-19 | 08:11:00 | 1
  9 | 2000 | 3 | 2022-04-21 | 08:13:00 | 1
 10 | 1000 | 8 | 2022-04-21 | 08:15:00 | 1
 11 | 5000 | 2 | 2022-04-21 | 08:20:00 | 1
 12 | 2000 | 6 | 2022-04-23 | 08:27:00 | 1
```

```
 118 | 1000 | 6 | 2022-11-19 | 09:19:00 | 1
 119 | 6000 | 3 | 2022-11-12 | 08:40:00 | 1
 120 | 6000 | 7 | 2022-11-26 | 08:37:00 | 1
 121 | 6000 | 4 | 2022-11-27 | 08:25:00 | 1
 122 | 3000 | 6 | 2021-11-22 | 08:41:00 | 1
 123 | 6000 | 6 | 2022-11-03 | 09:05:00 | 1
 124 | 6000 | 1 | 2022-11-19 | 09:05:00 | 1
(124 filas)
```

Nota: Es importante indicar que GROUP BY, establece la agrupación de los datos según el orden o la posición de declaración del campo, así se tiene: primero agrupa los datos por el Campo_Grupo1, luego agrupa los datos por el Campo_Grupo2, seguidamente agrupa los datos por el Campo_Grupo3, y así sucesivamente.

Además, todos los campos mencionados en la cláusula GROUP BY deben formar parte de los campos usados en el SELECT.

Se desea una consulta que muestre la cantidad total de retardos que tiene cada empleado:

```
SELECT fkcedemple, COUNT(fkcodexcu) AS Total_Retardos  
FROM tdretardos  
GROUP BY fkcedemple ;
```

```
bdretardados=# SELECT  fkcedemple, COUNT(fkcodexcu) AS Total_Retardos  
bdretardados=# FROM    tdretardos  GROUP BY fkcedemple ;  
fkcedemple | total_retardos  
-----+-----  
4000      |              21  
2000      |              17  
7000      |               8  
3000      |              23  
6000      |              28  
5000      |              10  
1000      |              17  
(7 filas)
```

Ejemplo 02: Dada la tabla llamada TDRETARDOS; se desea una consulta que muestre la cédula, el nombre y la cantidad total de retardo por empleado. Además, dicha consulta debe mostrar los datos ordenados por el campo cédula del empleado

```
SELECT fkcedemple, nomemple, COUNT(fkcodexcu) AS Total_Retardos  
FROM templeados, tdretardos  
WHERE templeados.pkcedemple = tdretardos.fkcedemple  
GROUP BY fkcedemple, nomemple  
ORDER BY fkcedemple;
```

```
bdretardados=# SELECT fkcedemple, nomemple, COUNT(fkcodexcu) AS Total_Retardos  
bdretardados=# FROM templeados, tdretardos  
bdretardados=# WHERE templeados.pkcedemple = tdretardos.fkcedemple  
bdretardados=# GROUP BY fkcedemple, nomemple  
bdretardados=# ORDER BY fkcedemple;  
fkcedemple | nomemple | total_retardos  
-----+-----+-----  
1000      | ROBERTO JAIMES | 17  
2000      | ZILA CONTRERAS | 17  
3000      | MARTHA         | 23  
4000      | CARLOS         | 21  
5000      | MATIAS         | 10  
6000      | NATASHA        | 28  
7000      | PEGGY CARTER   | 8  
(7 filas)
```

GROUP BY permite esencialmente dos cosas:

- 1.- Agrupar los datos por categorías,
- 2.- Usar en una consulta funciones de resumen o **funciones de agregación**, tales como: COUNT(), MAX(), MIN(), SUM(), AVG()...

Ejemplo 03: Se desea determinar el total de retardos agrupados año y mes respectivamente. Además, dicho reporte debe mostrar los datos debidamente ordenados por año (Ascendente) y mes (Ascendente).

```
SELECT EXTRACT(YEAR FROM fecha) AS "Año" ,  
        EXTRACT(MONTH FROM fecha) AS "Mes" ,  
        COUNT(fkcodexcu) AS Total  
FROM   tdretardos  
GROUP BY EXTRACT(YEAR FROM fecha),  
          EXTRACT(MONTH FROM fecha)  
ORDER BY EXTRACT(YEAR FROM fecha), EXTRACT(MONTH FROM fecha) ;
```

```
bdretardados=# SELECT  EXTRACT(YEAR FROM fecha) AS "Año" ,  
bdretardados=# EXTRACT(MONTH FROM fecha) AS "Mes" ,  
bdretardados=#          COUNT(fkcodexcu) AS Total  
bdretardados=# FROM    tdretardos  
bdretardados=# GROUP BY EXTRACT(YEAR FROM fecha),  
bdretardados=# EXTRACT(MONTH FROM fecha)  
bdretardados=# ORDER BY EXTRACT(YEAR FROM fecha), EXTRACT(MONTH FROM fecha) ;  
Año | Mes | total  
-----+-----+-----  
2021 | 11 | 2  
2022 | 1 | 20  
2022 | 2 | 12  
2022 | 3 | 17  
2022 | 4 | 19  
2022 | 5 | 12  
2022 | 6 | 13  
2022 | 7 | 11  
2022 | 8 | 1  
2022 | 9 | 1  
2022 | 10 | 1  
2022 | 11 | 15  
(12 filas)
```

Ejemplo 4: Se desea mostrar el total de retardos por cada año y mes desde el 01/Julio/2022 hasta el 30/Nov/2022. Dicho reporte debe mostrar los datos debidamente agrupados por año y mes; además, los datos se deben mostrar debidamente ordenados por año (Ascendente) y mes (Ascendente) .

```
SELECT EXTRACT(YEAR FROM fecha) AS "Año" ,  
        EXTRACT(MONTH FROM fecha) AS "Mes" ,  
        COUNT(fkcodexcu) AS Total  
FROM tdretardos  
WHERE fecha BETWEEN '2022-07-01' and '2022-11-30'  
GROUP BY EXTRACT(YEAR FROM fecha),  
           EXTRACT(MONTH FROM fecha)  
ORDER BY EXTRACT(YEAR FROM fecha), EXTRACT(MONTH FROM fecha) ;
```

```
bdretardados=# SELECT EXTRACT(YEAR FROM fecha) AS "Año" ,  
bdretardados=# EXTRACT(MONTH FROM fecha) AS "Mes" ,  
bdretardados=# COUNT(fkcodexcu) AS Total  
bdretardados=# FROM tdretardos  
bdretardados=# WHERE fecha BETWEEN '2022-07-01' and '2022-11-30'  
bdretardados=# GROUP BY EXTRACT(YEAR FROM fecha),  
bdretardados=# EXTRACT(MONTH FROM fecha)  
bdretardados=# ORDER BY EXTRACT(YEAR FROM fecha), EXTRACT(MONTH FROM fecha) ;  
Año | Mes | total  
-----+-----+-----  
2022 | 7 | 11  
2022 | 8 | 1  
2022 | 9 | 1  
2022 | 10 | 1  
2022 | 11 | 15  
(5 filas)
```

12.02.- HAVING --> Cuando dentro de la cláusula WHERE, se deben usar funciones de agregación, como por ejemplo: COUNT(), MAX(), MIN(), SUM(), AVG; se sustituye por la cláusula HAVING.

Ejemplo 01: Dada la tabla TDRETARDOS

```

bdretardados=# \d tdretardos
          Columna |          Tipo          | Tabla %public.tdretardos% | Ordenamiento | Nulable | Por omisión
-----|-----|-----|-----|-----|-----
pknr | integer | | | not null | nextval('tdretardos_pknr_seq'::regclass)
fkcedemple | character varying(12) | | | not null |
fkcodexcu | integer | | | not null |
fecha | date | | | not null |
hora | time without time zone | | | not null |
fkcods | integer | | | not null | 1
=ndices:
"tdretardos_pkey" PRIMARY KEY, btree (pknr)
Restricciones de llave foránea:
"tdretardos_fkcedemple_fkey" FOREIGN KEY (fkcedemple) REFERENCES tmempleados(pkcedemple) ON UPDATE CASCADE ON DELETE RESTRICT
"tdretardos_fkcodexcu_fkey" FOREIGN KEY (fkcodexcu) REFERENCES tmexcusas(pkcodexcu) ON UPDATE CASCADE ON DELETE RESTRICT
"tdretardos_fkcods_fkey" FOREIGN KEY (fkcods) REFERENCES tmstatus(pkcods) ON UPDATE CASCADE ON DELETE RESTRICT

```

```

bdretardados=# select * from tdretardos;
pknr | fkcedemple | fkcodexcu | fecha | hora | fkcods
-----|-----|-----|-----|-----|-----
1 | 2000 | 3 | 2022-04-05 | 08:32:00 | 1
2 | 5000 | 3 | 2022-04-05 | 08:50:00 | 1
3 | 4000 | 1 | 2022-04-06 | 09:09:00 | 1
4 | 3000 | 5 | 2022-04-06 | 09:13:00 | 1
5 | 6000 | 7 | 2022-04-11 | 08:21:00 | 1
6 | 1000 | 2 | 2022-04-14 | 08:16:00 | 1
7 | 2000 | 6 | 2022-04-17 | 10:00:00 | 1
8 | 3000 | 7 | 2022-04-19 | 08:11:00 | 1
9 | 2000 | 3 | 2022-04-21 | 08:13:00 | 1
10 | 1000 | 8 | 2022-04-21 | 08:15:00 | 1
11 | 5000 | 2 | 2022-04-21 | 08:20:00 | 1
12 | 2000 | 6 | 2022-04-23 | 08:27:00 | 1
118 | 1000 | 6 | 2022-11-19 | 09:19:00 | 1
119 | 6000 | 3 | 2022-11-12 | 08:40:00 | 1
120 | 6000 | 7 | 2022-11-26 | 08:37:00 | 1
121 | 6000 | 4 | 2022-11-27 | 08:25:00 | 1
122 | 3000 | 6 | 2021-11-22 | 08:41:00 | 1
123 | 6000 | 6 | 2022-11-03 | 09:05:00 | 1
124 | 6000 | 1 | 2022-11-19 | 09:05:00 | 1
(124 filas)

```

Se desea una consulta que muestre el número de cédula de los empleados **con 20 o más retardos registrados.**

```
SELECT fkcedemple, COUNT(fkcodexcu) AS  
"Total_Retardos"  
FROM tdretardos  
GROUP BY fkcedemple  
HAVING COUNT (fkcodexcu) >= 20 ;
```

```
bdretardados=# SELECT fkcedemple, COUNT(fkcodexcu) AS  
bdretardados-# "Total_Retardos"  
bdretardados-# FROM tdretardos  
bdretardados-# GROUP BY fkcedemple  
bdretardados-# HAVING COUNT (fkcodexcu) >= 20 ;  
fkcedemple | Total_Retardos  
-----+-----  
4000      |          21  
3000      |          23  
6000      |          28  
(3 filas)  
  
bdretardados=#
```

Ejemplo 02: Dada la tabla TDRETARDOS, se desea una consulta que muestre la cédula, el nombre y la cantidad total de retardos; solo de aquellos empleados **con 20 o más retardos acumulados**. Además, dicha consulta debe mostrar los datos ordenados por el campo cédula (Ascendente) del empleado.

```
SELECT fkcedemple, nomemple,  
        COUNT(fkcodexcu) AS "Total_Retardos"  
FROM templeados, tdretardos  
WHERE templeados.pkcedemple = tdretardos.fkcedemple  
GROUP BY fkcedemple, nomemple  
HAVING COUNT (fkcodexcu) >= 20  
  
ORDER BY fkcedemple ;
```

```
bdretardados=# SELECT  fkcedemple, nomemple,  
bdretardados=# COUNT(fkcodexcu) AS  "Total_Retardos"  
bdretardados=# FROM    templeados,  tdretardos  
bdretardados=# WHERE   templeados.pkcedemple = tdretardos.fkcedemple  
bdretardados=# GROUP BY fkcedemple,  nomemple  
bdretardados=# HAVING   COUNT (fkcodexcu) >= 20  
bdretardados=# ORDER BY    fkcedemple  ;  
fkcedemple | nomemple | Total_Retardos  
-----+-----+-----  
3000      | MARTHA  |          23  
4000      | CARLOS  |          21  
6000      | NATASHA |          28  
(3 filas)
```

Nota: La cláusula HAVING ejecuta el filtro sobre los datos ya agrupados.

13.- Modificar datos

13.01.- Update --> Se usa para actualizar el contenido de uno o varios campos en una tabla.

UPDATE Tabla

SET Campo1 = 'Valor_Texto/Fecha/Hora', Campo2 = Valor_Numérico,
Campo3= Valor, , CampoN = ValorN

WHERE Criterio o Condición ;

Ejemplo 1: Cambiar el nombre de Matias por Thalía

UPDATE templeados

SET nomemple = 'Thalía' WHERE pkCedemple = '5000' ;

```
bdretardados=# select * from templeados;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----|-----|-----|-----|-----
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1
3000 | MARTHA | 1996-11-22 | 3 | 1
4000 | CARLOS | 2000-07-15 | 4 | 1
5000 | MATIAS | 2005-09-25 | 4 | 1
6000 | NATASHA | 2007-10-28 | 1 | 1
7000 | PEGGY CARTER | 1999-08-29 | 2 | 1
8000 | YSA CAPACHO | 1993-07-23 | 2 | 1
9000 | MARIA MORALES | 2003-07-07 | 4 | 1
(9 filas)

bdretardados=# UPDATE templeados
bdretardados=# SET nomemple = 'Thalía' WHERE pkCedemple = '5000' ;
UPDATE 1
bdretardados=# select * from templeados;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----|-----|-----|-----|-----
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1
3000 | MARTHA | 1996-11-22 | 3 | 1
4000 | CARLOS | 2000-07-15 | 4 | 1
6000 | NATASHA | 2007-10-28 | 1 | 1
7000 | PEGGY CARTER | 1999-08-29 | 2 | 1
8000 | YSA CAPACHO | 1993-07-23 | 2 | 1
9000 | MARIA MORALES | 2003-07-07 | 4 | 1
5000 | Thalía | 2005-09-25 | 4 | 1
(9 filas)
```

Nota: si se desea actualizar todos los registros de una tabla, no se debe incluir la cláusula WHERE.

UPDATE templeados

SET fkcods = 0 ;

```
bdretardados=# select * from templeados;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----+-----+-----+-----+-----
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1
3000 | MARTHA | 1996-11-22 | 3 | 1
4000 | CARLOS | 2000-07-15 | 4 | 1
6000 | NATASHA | 2007-10-28 | 1 | 1
7000 | PEGGY CARTER | 1999-08-29 | 2 | 1
8000 | YSA CAPACHO | 1993-07-23 | 2 | 1
9000 | MARIA MORALES | 2003-07-07 | 4 | 1
5000 | Thalía | 2005-09-25 | 4 | 1
(9 filas)

bdretardados=# UPDATE templeados
bdretardados=# SET fkcods = 0 ;
UPDATE 9
bdretardados=# select * from templeados;
pkcedemple | nomemple | fecha | fkcodcar | fkcods
-----+-----+-----+-----+-----
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 0
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 0
3000 | MARTHA | 1996-11-22 | 3 | 0
4000 | CARLOS | 2000-07-15 | 4 | 0
6000 | NATASHA | 2007-10-28 | 1 | 0
7000 | PEGGY CARTER | 1999-08-29 | 2 | 0
8000 | YSA CAPACHO | 1993-07-23 | 2 | 0
9000 | MARIA MORALES | 2003-07-07 | 4 | 0
5000 | Thalía | 2005-09-25 | 4 | 0
(9 filas)
```

13.02.- Actualización en cascada --> Por lo general al momento de crear las tablas se incluye como condición “**on update cascade**”, que significa: en caso de que se realice un cambio en el campo padre, automáticamente todos los campos hijos se actualizan.

```
CREATE TABLE tmcargos(
  pkcodcar SERIAL primary key not null,
  dcargo varchar(30) not null UNIQUE,
  sueldo decimal(12,2) not null,
  fkcods INTEGER not null DEFAULT 1,
  foreign key(fkcods) references tmstatus(pkcods)
  on update cascade on delete restrict );
```

Ejemplo 1: Cambiar en la tabla tmstatus, el contenido del campo pkcods del código 1 (Activo) al código 8.

Antes

```
bdretardados=# select * from tmstatus;
pkcods | dstatus
-----+-----
0 | ELIMINADO
1 | ACTIVO
(2 filas)
```

```
bdretardados=# select * from tmcargos;
pkcodcar | dcargo | sueldo | fkcods
-----+-----+-----+-----
1 | PRESIDENTE | 5000000.01 | 1
2 | VICE PRESIDENTE | 4500000.02 | 0
3 | GERENTE | 4000000.03 | 1
4 | SUB-GERENTE | 3500000.04 | 0
5 | COORDINADOR | 3000000.05 | 1
6 | SUPERVISOR | 2500000.06 | 0
(6 filas)
```

UPDATE tmstatus SET pkcods = 8 WHERE pkcods = 1 ;

Después

```
bdretardados=# SELECT * FROM tmstatus;
pkcods | dstatus
-----+-----
0 | ELIMINADO
8 | ACTIVO
(2 filas)
```

```
bdretardados=# SELECT * FROM tmcargos;
pkcodcar | dcargo | sueldo | fkcods
-----+-----+-----+-----
2 | VICE PRESIDENTE | 4500000.02 | 0
4 | SUB-GERENTE | 3500000.04 | 0
6 | SUPERVISOR | 2500000.06 | 0
1 | PRESIDENTE | 5000000.01 | 8
3 | GERENTE | 4000000.03 | 8
5 | COORDINADOR | 3000000.05 | 8
(6 filas)
```

13.02.- Delete --> Permite borrar físicamente uno o varios registros en una tabla; siempre y cuando se cumpla la integridad referencial.

DELETE FROM Tabla **WHERE** Criterio-Condición ;

Ejemplo 1: borrar todos los registros del empleado con la cédula número 7000, y que están en la **tdretardos**.

SELECT * FROM Tdretardos **WHERE** fkcedemple = '7000' ;

DELETE FROM Tdretardos **WHERE** fkcedemple = '7000' ;

```
bdretardados=# SELECT * FROM Tdretardos WHERE fkcedemple = '7000' ;
pknr | fkcedemple | fkcodexcu | fecha | hora | fkcods
-----+-----+-----+-----+-----+-----
 26 | 7000 | 1 | 2022-05-05 | 09:00:00 | 1
 67 | 7000 | 3 | 2022-01-14 | 08:32:00 | 1
 78 | 7000 | 1 | 2022-06-05 | 09:00:00 | 1
 90 | 7000 | 1 | 2022-07-05 | 09:00:00 | 1
 97 | 7000 | 5 | 2022-11-27 | 08:20:00 | 1
101 | 7000 | 7 | 2022-03-05 | 09:11:00 | 1
108 | 7000 | 4 | 2022-10-27 | 08:25:00 | 1
116 | 7000 | 4 | 2022-11-10 | 08:27:00 | 1
(8 filas)

bdretardados=#
bdretardados=# DELETE FROM Tdretardos WHERE fkcedemple = '7000' ;
DELETE 8
bdretardados=# SELECT * FROM Tdretardos WHERE fkcedemple = '7000' ;
pknr | fkcedemple | fkcodexcu | fecha | hora | fkcods
-----+-----+-----+-----+-----+-----
(0 filas)
```

Nota: si no se coloca la cláusula **WHERE** se eliminan todos los registros de la tabla.

DELETE FROM Tabla ;

14.- Modificar la estructura de la tabla (ALTER TABLE)

14.01.- Agregar un campo

ALTER TABLE tabla

ADD COLUMN Campo_Nuevo Tipo_Dato Condición_Campo ;

Ejemplo: Agregar los campos fecha de tipo date y el campo edad de tipo entero con el valor por defecto 23.

ALTER TABLE tusuarios

ADD COLUMN fecha date,

ADD COLUMN edad integer default 23 ;

```
bdretardados=# \d tusuarios
          Tabla public.tusuarios
-----
Columna | Tipo | Ordenamiento | Nulable | Por omisión
-----|-----|-----|-----|-----
pknu    | integer |                | not null | nextval('tusuarios_pknu_seq'::regclass)
nom_emple | character varying(40) |                | not null |
usuario | character varying(40) |                | not null |
clave   | character varying(40) |                | not null |
foto_perfil | character varying(40) |                | not null |
nivel   | integer |                | not null |
fkcods  | integer |                | not null | 1
Indices:
"tusuarios_pkey" PRIMARY KEY, btree (pknu)
Restricciones de llave foránea:
"tusuarios_fkcods_fkey" FOREIGN KEY (fkcods) REFERENCES tmstatus(pkcods) ON UPDATE CASCADE ON DELETE RESTRICT

bdretardados=# ALTER TABLE tusuarios
bdretardados=# ADD COLUMN fecha date,
bdretardados=# ADD COLUMN edad integer default 23 ;
ALTER TABLE
bdretardados=# select * from tusuarios;
 pknu | nom_emple | usuario | clave | foto_perfil | nivel | fkcods | fecha | edad
-----|-----|-----|-----|-----|-----|-----|-----|-----
 1 | WILFREDO | CERO    | 1234  | foto00.jpg | 0     | 1     |      | 23
 2 | YSA      | UNO     | 1234  | foto00.jpg | 1     | 1     |      | 23
 3 | GREISY   | DOS     | 1234  | foto00.jpg | 2     | 1     |      | 23
 4 | ZILA     | TRES    | 1234  | foto00.jpg | 3     | 1     |      | 23
 5 | SERGIO   | CUATRO  | 1234  | foto00.jpg | 4     | 1     |      | 23
(5 filas)
```

14.02.- Modificar el campo

```
ALTER TABLE Nombre_Tabla
```

```
ALTER COLUMN Campo TYPE Tipo_Nuevo Condición_Nueva ;
```

Ejemplo 01:

```
ALTER TABLE xcargos
```

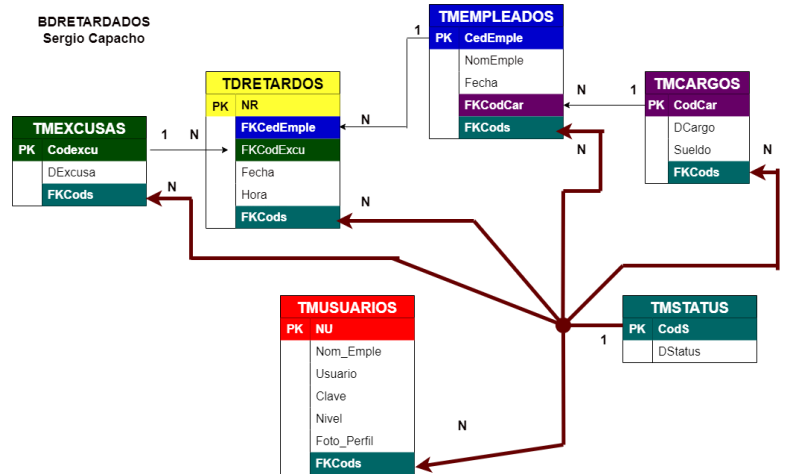
```
ALTER COLUMN sueldo TYPE integer ;
```

```
bdretardados=# select * from xcargos;
 pkcodcar | dcargo | sueldo | fkcods
-----+-----+-----+-----
 1 | PRESIDENTE | 5000000.00 | 1
 2 | VICE PRESIDENTE | 4500000.00 | 1
 3 | GERENTE | 4000000.00 | 1
 4 | SUB-GERENTE | 3500000.00 | 1
 5 | COORDINADOR | 3000000.00 | 1
 6 | SUPERVISOR | 2500000.00 | 1
(6 filas)

bdretardados=# ALTER TABLE xcargos
ALTER COLUMN sueldo TYPE integer ;
bdretardados=# select * from xcargos;
 pkcodcar | dcargo | sueldo | fkcods
-----+-----+-----+-----
 1 | PRESIDENTE | 5000000 | 1
 2 | VICE PRESIDENTE | 4500000 | 1
 3 | GERENTE | 4000000 | 1
 4 | SUB-GERENTE | 3500000 | 1
 5 | COORDINADOR | 3000000 | 1
 6 | SUPERVISOR | 2500000 | 1
(6 filas)

bdretardados=# \d xcargos
          Tabla "public.xcargos"
Columna | Tipo | Ordenamiento | Nulable | Por omisi\u00f3n
-----+-----+-----+-----+-----
 pkcodcar | integer | | not null | nextval('xcargos_pkcodcar_seq'::regclass)
 dcargo | character varying(30) | | not null |
 sueldo | integer | | not null |
 fkcods | integer | | not null | 1
```

Ejemplo 02: Después de analizar la tabla **TMUsuarios** se observa que debe existir una relación con la tabla **TMCargos**; es decir, se debe crear una asociación entre **TMCargos (PKCodcar)** y **TMUsuario (FKCodcar)**.



```
SELECT * FROM tmsuarios ;
```

```
\d tmsuarios ;
```

```
ALTER TABLE tmsuarios ADD COLUMN fkcodcar integer DEFAULT 1 ;
```

```
ALTER TABLE tmsuarios ADD CONSTRAINT fkcodcar
```

```
FOREIGN KEY (fkcodcar) REFERENCES tmcargos(pkcodcar)
```

```
ON DELETE RESTRICT ON UPDATE CASCADE ;
```

```
SELECT * FROM tmsuarios ;
```

```
\d tmsuarios ;
```

Ejemplo 03:

```
ALTER TABLE productos
```

```
ALTER COLUMN precio TYPE numérico(10,2) ;
```

14.03.- Renombrar un campo

```
ALTER TABLE tabla
```

```
RENAME COLUMN Campo_Viejo TO Campo_Nuevo Condición ;
```

Ejemplo:

```
ALTER TABLE tmsuarios
```

```
RENAME COLUMN fecha TO fecha_nueva ;
```

```
bdcapacho02=# select * from tmsuarios;
 nu | ape_u | nom_u | clave_u | nivel_ | fecha
-----+-----+-----+-----+-----+-----
  1 | YSA   | UNO   | 1234    |         | 1967-11-22
  2 | GREISY | DOS   | 1234    |         | 1967-11-22
  3 | ZILA  | TRES  | 1234    |         | 1967-11-22
  4 | SERGIO | CUATRO | 1234    |         | 1967-11-22
  5 | WILFREDO | CINCO | 1234    |         | 1967-11-22
(5 filas)

bdcapacho02=# ALTER TABLE tmsuarios RENAME COLUMN fecha TO fecha_nueva;
ALTER TABLE
bdcapacho02=# select * from tmsuarios;
 nu | ape_u | nom_u | clave_u | nivel_ | fecha_nueva
-----+-----+-----+-----+-----+-----
  1 | YSA   | UNO   | 1234    |         | 1967-11-22
  2 | GREISY | DOS   | 1234    |         | 1967-11-22
  3 | ZILA  | TRES  | 1234    |         | 1967-11-22
  4 | SERGIO | CUATRO | 1234    |         | 1967-11-22
  5 | WILFREDO | CINCO | 1234    |         | 1967-11-22
(5 filas)

bdcapacho02=# _
```

14.04.- Eliminar un campo

```
ALTER TABLE tabla
```

```
DROP COLUMN Campo_Eliminar ;
```

Ejemplo:

```
ALTER TABLE tmusuarios
```

```
DROP COLUMN edad ;
```

```
bdcapacho02=# select * from tmusuarios;
```


nu	ape_u	nom_u	clave_u	nivel_u	fecha	edad
1	YSA	UNO	1234	1	1967-11-22	54
2	GREISY	DOS	1234	2	1967-11-22	54
3	ZILA	TRES	1234	3	1967-11-22	54
4	SERGIO	CUATRO	1234	4	1967-11-22	54
5	WILFREDO	CINCO	1234	0	1967-11-22	54

(5 filas)

```
bdcapacho02=# ALTER TABLE tmusuarios DROP COLUMN edad;
ALTER TABLE
bdcapacho02=# select * from tmusuarios;
```

nu	ape_u	nom_u	clave_u	nivel_u	fecha
1	YSA	UNO	1234	1	1967-11-22
2	GREISY	DOS	1234	2	1967-11-22
3	ZILA	TRES	1234	3	1967-11-22
4	SERGIO	CUATRO	1234	4	1967-11-22
5	WILFREDO	CINCO	1234	0	1967-11-22

(5 filas)



Nota: al momento de eliminar un campo el PostgreSQL verifica la integridad referencial

14.05.- Cambiar el nombre de una tabla (RENAME)

`ALTER TABLE nombre_actual_de_la_tabla`

`RENAME TO nuevo_nombre_de_la_tabla ;`

Ejemplo:

`ALTER TABLE tmexcusas`

`RENAME TO excusas ;`

```
bdcapacho02=# \d
Listado de relaciones
Esquema | Nombre | Tipo | Due
-----|-----|-----|-----
public  | tdretardos | tabla | postgres
public  | tdretardos_nr_seq | secuencia | postgres
public  | tmcargos | tabla | postgres
public  | tmcargos_codcar_seq | secuencia | postgres
public  | templeados | tabla | postgres
public  | tmexcusas | tabla | postgres
public  | tmexcusas_codexcu_seq | secuencia | postgres
public  | tmstatus | tabla | postgres
public  | tmusuarios | tabla | postgres
public  | tmusuarios_nu_seq | secuencia | postgres
(10 filas)

bdcapacho02=# ALTER TABLE tmexcusas RENAME TO excusas;
ALTER TABLE
bdcapacho02=# \d
Listado de relaciones
Esquema | Nombre | Tipo | Due
-----|-----|-----|-----
public  | excusas | tabla | postgres
public  | tdretardos | tabla | postgres
public  | tdretardos_nr_seq | secuencia | postgres
public  | tmcargos | tabla | postgres
public  | tmcargos_codcar_seq | secuencia | postgres
public  | templeados | tabla | postgres
public  | tmexcusas_codexcu_seq | secuencia | postgres
public  | tmstatus | tabla | postgres
public  | tmusuarios | tabla | postgres
public  | tmusuarios_nu_seq | secuencia | postgres
(10 filas)

bdcapacho02=#
```

15.- JOIN

¿Qué son las consultas JOIN y por qué son importantes?

Nosotros podemos consultar el contenido de una tabla muy puntualmente así:

```
SELECT *..FROM Tabla ;
```

Sin embargo, generalmente para obtener información se deben combinar los datos de dos o más tablas, pues es el único camino viable para obtenerla(Información) es esta vía. Esta acción, le permite al programador obtener datos confiable, los cuales serán luego utilizados en la toma de decisiones.

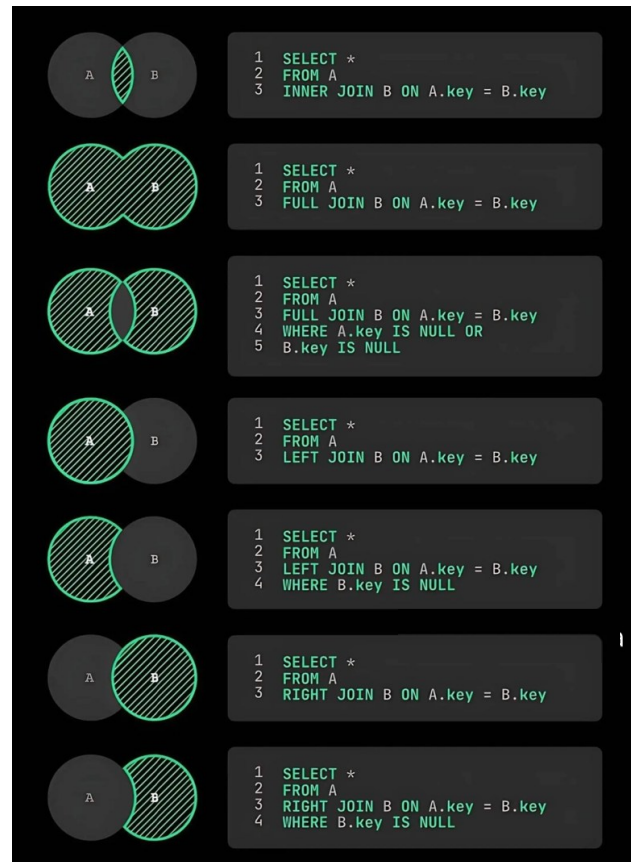
Cuando trabajamos con bases de datos relacionales:

- Una tabla fuerte representa una entidad independiente (**Tabla Maestra**), y una tabla débil (**Tabla Detalle**) representa una entidad que depende de otra.
- Las relaciones de dependencia entre tablas se representan por las asociaciones entre claves primarias y claves foráneas.

Entonces, la cláusula **JOIN nos permite asociar 2 o más tablas, en base a una columna que tengan en común.**

Ejemplo: XEmpleados y XCargos

A continuación, vamos a analizar la cláusula **JOIN** y sus variantes.



Para ello, empecemos definiendo 2 tablas, que serán la base para nuestros ejemplos.

Por un lado, vamos a tener una tabla llamada **xempleados** (que almacenará una lista de empleados y con una clave foránea código del cargo); por otro lado, una tabla llamada **xcargos** (con la lista de cargos que existen en la empresa).

Script de la tabla XCargos (Contiene todos los cargos de la empresa)

```
CREATE TABLE xcargos(  
    pkcodcar SERIAL primary key not null,  
    dcargo   varchar(30)         not null,  
    sueldo   decimal(12,2)       not null,  
    fkcods   INTEGER             not null DEFAULT 1,  
    foreign  key(fkcods) references tmstatus(pkcods)  
  
                on update cascade on delete restrict ) ;
```

```
INSERT INTO xcargos (dcargo, sueldo) VALUES  
('PRESIDENTE',      5000000.00) ,  
('VICE PRESIDENTE', 4500000.00) ,  
('GERENTE',         4000000.00),  
('SUB-GERENTE',     3500000.00) ,  
('COORDINADOR',    3000000.00),  
  
('SUPERVISOR',      2500000.00) ;
```

```
SELECT pkcodcar, dcargo,  
        to_char( sueldo, '999G999G999D99' ) As "Sueldo"  
  
FROM xcargos ;
```

```
bdretardados=# SELECT pkcodcar,  
bdretardados=# dcargo,  
bdretardados=# to_char( sueldo, '999G999G999D99' ) As "Sueldo"  
bdretardados=# FROM xcargos ;  
pkcodcar | dcargo | Sueldo  
-----  
1 | PRESIDENTE | 5.000.000,00  
2 | VICE PRESIDENTE | 4.500.000,00  
3 | GERENTE | 4.000.000,00  
4 | SUB-GERENTE | 3.500.000,00  
5 | COORDINADOR | 3.000.000,00  
6 | SUPERVISOR | 2.500.000,00  
(6 filas)
```

Script de la tabla xempleados (Tiene le listado de tofos los datos)

```
CREATE TABLE xempleados (  
    pkcedemple          varchar(12) not null PRIMARY KEY,  
    Nomemple           varchar(40) not null,  
    Fecha              date not null,  
    Fkcodcar           integer ,  
    fkcods             integer not null DEFAULT 1,  
    foreign key(fkcodcar) references xcargos(pkcodcar)  
                                on update cascade on delete restrict,  
    foreign key(fkcods) references tmstatus(pkcods)  
                                on update cascade on delete restrict) ;
```

```
INSERT INTO xempleados (pkcedemple, nomemple, fecha, fkcodcar) VALUES  
( '1000' , 'ROBERTO JAIMES' , '1978-03-03' , 3),  
( '2000' , 'ZILA CONTRERAS' , '1980-10-05' , 2),  
( '3000' , 'MARTHA' , '1996-11-22' , 3),  
( '4000' , 'CARLOS' , '2000-07-15' , 4),  
( '5000' , 'MATIAS' , '2005-09-25' , 4),  
( '6000' , 'NATASHA' , '2007-10-28' , 1),  
( '7000' , 'PEGGY CARTER' , '1999-08-29' , 2),  
( '8000' , 'YSA CAPACHO' , '1993-07-23' , 2),  
( '9000' , 'MARIA' , '2003-07-07' , 4);
```

```
INSERT INTO xempleados (pkcedemple, nomemple, fecha) VALUES  
( '3001' , 'RITA' , '1978-04-03'),  
( '3002' , 'JOHANNA' , '1980-07-20'),  
( '3003' , 'VLADIMIR' , '1998-05-22'),  
( '3004' , 'ARMANDO' , '2007-07-21'),  
( '7001' , 'MARIA GIL' , '2002-04-17'),  
( '7002' , 'JAMES' , '1997-10-28'),  
( '7003' , 'FANNY' , '1999-08-27'),  
( '7004' , 'NATHALY' , '1993-07-23'),  
  
( '7005' , 'THALIA' , '2003-07-09') ;
```

```
SELECT * FROM xempleados ORDER BY pkcedemple ;
```

```
bdretardados=# SELECT * FROM xempleados ORDER BY pkcedemple ;
```

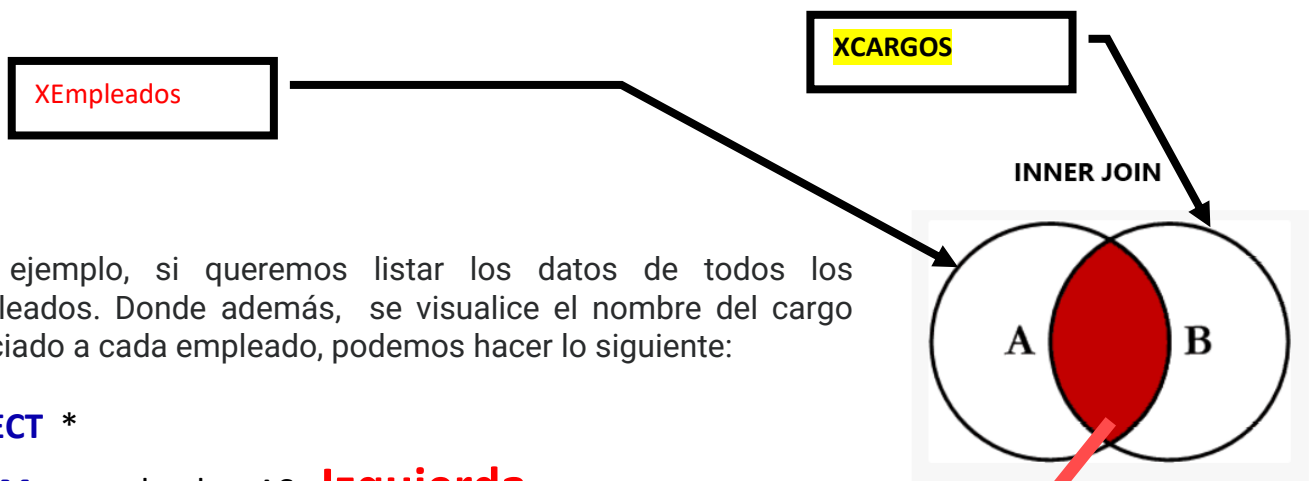
pkcedemple	nomemple	fecha	fkcodcar	fkcods
1000	ROBERTO JAIMES	1978-03-03	3	1
2000	ZILA CONTRERAS	1980-10-05	2	1
3000	MARTHA	1996-11-22	3	1
3001	RITA	1978-04-03		1
3002	JOHANNA	1980-07-20		1
3003	VLADIMIR	1998-05-22		1
3004	ARMANDO	2007-07-21		1
4000	CARLOS	2000-07-15	4	1
5000	MATIAS	2005-09-25	4	1
6000	NATASHA	2007-10-28	1	1
7000	PEGGY CARTER	1999-08-29	2	1
7001	MARIA GIL	2002-04-17		1
7002	JAMES	1997-10-28		1
7003	FANNY	1999-08-27		1
7004	NATHALY	1993-07-23		1
7005	THALIA	2003-07-09		1
8000	YSA CAPACHO	1993-07-23	2	1
9000	MARIA	2003-07-07	4	1

(18 filas)

16.01.- Cláusula INNER JOIN

Lo más usual, lo primero que se suele aprender, es el uso de **INNER JOIN**, generalmente abreviado como **JOIN**.

Esta cláusula busca coincidencias entre 2 tablas, en función a **una columna que tienen una relación o asociación**. De tal modo que sólo se mostraran los datos o registros que se intercepten.



Por ejemplo, si queremos listar los datos de todos los empleados. Donde además, se visualice el nombre del cargo asociado a cada empleado, podemos hacer lo siguiente:

```
SELECT *  
FROM xempleados AS Izquierda  
JOIN xcargos AS Derecha  
ON Izquierda.fkcodcar = Derecha.pkcodcar  
ORDER BY Izquierda.pkcedemple ;
```

```
bdretardados=# SELECT *  
bdretardados=# FROM xempleados AS Izquierda  
bdretardados=# JOIN xcargos AS Derecha  
bdretardados=# ON Izquierda.fkcodcar = Derecha.pkcodcar  
bdretardados=# ORDER BY Izquierda.pkcedemple ;  
pkcedemple | nomemple | fecha | fkcodcar | fkods | pkcodcar | dcargo | sueldo | fkods  
-----  
1000 | ROBERTO JAIMES | 1978-03-03 | 3 | 1 | 3 | GERENTE | 4000000.00 | 1  
2000 | ZILA CONTRERAS | 1980-10-05 | 2 | 1 | 2 | VICE PRESIDENTE | 4500000.00 | 1  
3000 | MARTHA | 1996-11-22 | 3 | 1 | 3 | GERENTE | 4000000.00 | 1  
4000 | CARLOS | 2000-07-15 | 4 | 1 | 4 | SUB-GERENTE | 3500000.00 | 1  
5000 | MATIAS | 2005-09-25 | 4 | 1 | 4 | SUB-GERENTE | 3500000.00 | 1  
6000 | NATASHA | 2007-10-28 | 1 | 1 | 1 | PRESIDENTE | 5000000.00 | 1  
7000 | PEGGY CARTER | 1999-08-29 | 2 | 1 | 2 | VICE PRESIDENTE | 4500000.00 | 1  
8000 | YSA CAPACHO | 1993-07-23 | 2 | 1 | 2 | VICE PRESIDENTE | 4500000.00 | 1  
9000 | MARIA | 2003-07-07 | 4 | 1 | 4 | SUB-GERENTE | 3500000.00 | 1  
(9 filas)
```

Aquí podemos notar: en primer lugar lo siguiente: Los cargos "Coordinador" y "Supervisor" no aparece en los resultados, ya que no pertenece a ningún cargo asignado a los empleados; en segundo lugar los empleados: 'RITA', 'JOHANNA', 'VLADIMIR', 'ARMANDO', 'MARIA GIL', 'JAMES', 'FANNY', 'NATHALY' y 'THALIA'; tan poco aparecen, pues no tiene asignado ningún código de cargo para construir la relación.

¿Por qué ocurre esto? Porque **JOIN** muestra como resultado los registros que tiene una intersección entre ambas tablas, en este caso se establece a través del campo código del cargo.

También hay que tener en cuenta que en los resultados vemos todos los campos que corresponden a la tabla **xempleados** y a la tabla **xcargos**; esto ocurre porque estamos seleccionando todas las columnas con un *****.

Sin embargo, es posible seleccionar algunos campos (además de asignarles un alias):

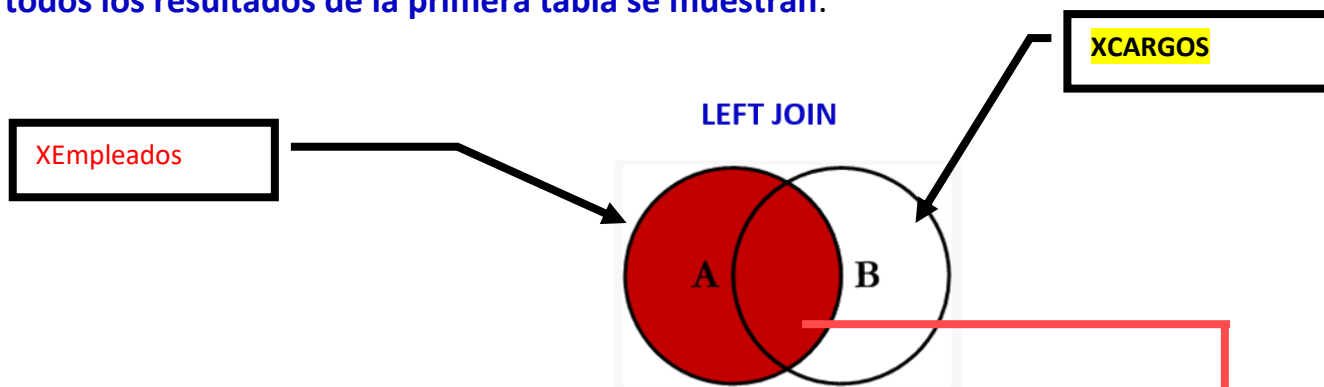
```
SELECT Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
FROM xempleados AS Izq
JOIN xcargos AS Der ON Izq.fkcodcar = Der.pkcodcar
ORDER BY Izq.pkcedemple ;
```

```
bdretardados=# SELECT Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
bdretardados=# FROM xempleados AS Izq
bdretardados=# JOIN xcargos AS Der
bdretardados=# ON Izq.fkcodcar = Der.pkcodcar
bdretardados=# ORDER BY Izq.pkcedemple ;
 pkcedemple | nomemple | fecha | fkcodcar | dcargo
-----+-----+-----+-----+-----
 1000 | ROBERTO JAIMES | 1978-03-03 | 3 | GERENTE
 2000 | ZILA CONTRERAS | 1980-10-05 | 2 | VICE PRESIDENTE
 3000 | MARTHA | 1996-11-22 | 3 | GERENTE
 4000 | CARLOS | 2000-07-15 | 4 | SUB-GERENTE
 5000 | MATIAS | 2005-09-25 | 4 | SUB-GERENTE
 6000 | NATASHA | 2007-10-28 | 1 | PRESIDENTE
 7000 | PEGGY CARTER | 1999-08-29 | 2 | VICE PRESIDENTE
 8000 | YSA CAPACHO | 1993-07-23 | 2 | VICE PRESIDENTE
 9000 | MARIA | 2003-07-07 | 4 | SUB-GERENTE
(9 filas)
```

16.02.- Cláusula LEFT JOIN

A diferencia de un **INNER JOIN**, donde se busca extraer los datos de la intersección de las tablas; con **LEFT JOIN** damos prioridad a la tabla de la izquierda, y buscamos en la tabla de la derecha los registros que tengan alguna asociación directa.

Si no existe ninguna coincidencia para alguna de las filas de la tabla de la izquierda, de igual forma **todos los resultados de la primera tabla se muestran**.



```
SELECT  Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
FROM    xempleados AS Izq
LEFT JOIN xcargos AS Der ON Izq.fkcodcar = Der.pkcodcar

ORDER BY Izq.pkcedemple ;
```

```
bdretardados=# SELECT      Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
bdretardados=# FROM        xempleados AS Izq
bdretardados=# LEFT JOIN   xcargos AS Der
bdretardados=# ON         Izq.fkcodcar = Der.pkcodcar
bdretardados=# ORDER BY  Izq.pkcedemple ;
 pkcedemple | nomemple | fecha | fkcodcar | dcargo
-----|-----|-----|-----|-----
 1000 | ROBERTO JAIMES | 1978-03-03 | 3 | GERENTE
 2000 | ZILA CONTRERAS | 1980-10-05 | 2 | VICE PRESIDENTE
 3000 | MARTHA | 1996-11-22 | 3 | GERENTE
 3001 | RITA | 1978-04-03 | 3 | GERENTE
 3002 | JOHANNA | 1980-07-20 | 3 | GERENTE
 3003 | VLADIMIR | 1998-05-22 | 3 | GERENTE
 3004 | ARMANDO | 2007-07-21 | 3 | GERENTE
 4000 | CARLOS | 2000-07-15 | 4 | SUB-GERENTE
 5000 | MATIAS | 2005-09-25 | 4 | SUB-GERENTE
 6000 | NATASHA | 2007-10-28 | 1 | PRESIDENTE
 7000 | PEGGY CARTER | 1999-08-29 | 2 | VICE PRESIDENTE
 7001 | MARIA GIL | 2002-04-17 | 2 | VICE PRESIDENTE
 7002 | JAMES | 1997-10-28 | 2 | VICE PRESIDENTE
 7003 | FANNY | 1999-08-27 | 2 | VICE PRESIDENTE
 7004 | NATHALY | 1993-07-23 | 2 | VICE PRESIDENTE
 7005 | THALIA | 2003-07-09 | 2 | VICE PRESIDENTE
 8000 | YSA CAPACHO | 1993-07-23 | 2 | VICE PRESIDENTE
 9000 | MARIA | 2003-07-07 | 4 | SUB-GERENTE
(18 filas)
```

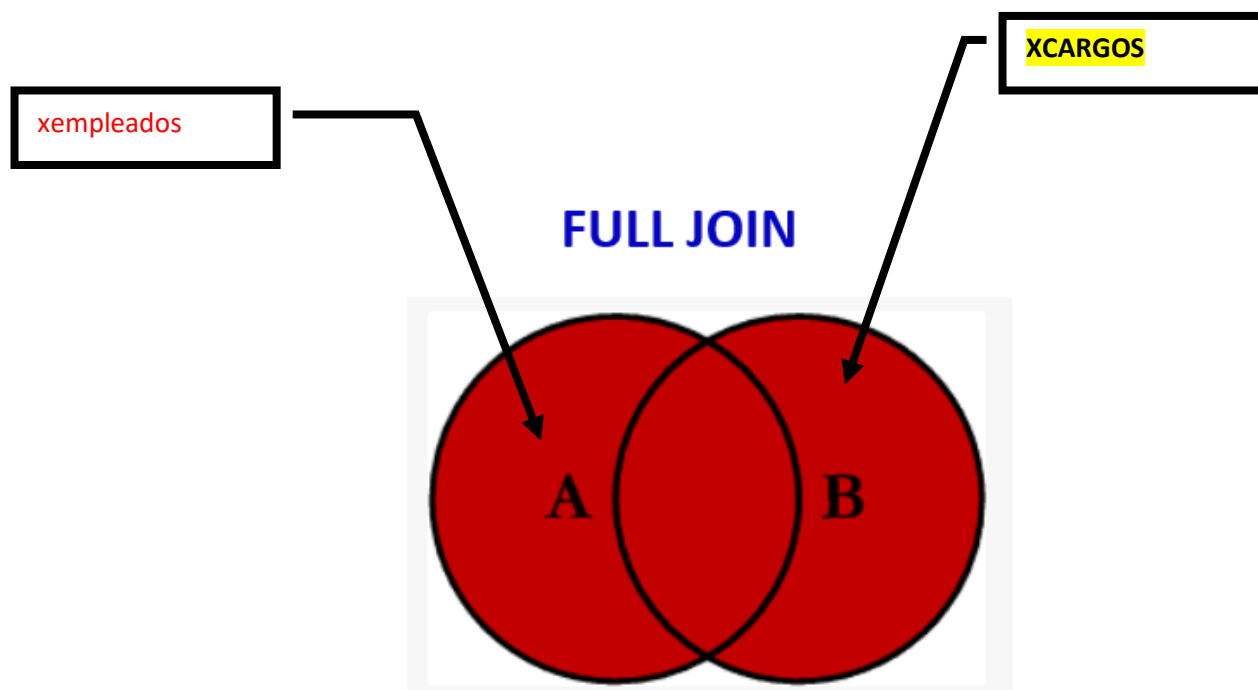

En este caso, la tabla de la izquierda es **xempleados**, mientras que **xcargos** es la tabla de la derecha.

La tabla asociada al **FROM** será siempre la tabla **LEFT(Izquierda)**, y la tabla que viene después del **JOIN** será la tabla **RIGHT (Derecha)**.

Como se observa en la consulta, se muestran todos los cargos al menos una vez; y en los casos donde no existen empleados asociado con un cargo, se mostrará NULL. Pero el cargo aparecerá.

16.04.- Cláusula FULL JOIN

Mientras que **LEFT JOIN** muestra todas las filas de la tabla izquierda, y **RIGHT JOIN** muestra todas las correspondientes a la tabla derecha, **FULL OUTER JOIN** (o simplemente **FULL JOIN**) se encarga de mostrar todas las filas de ambas tablas, sin importar que no existan coincidencias (usará **NULL** como un valor por defecto para dichos casos).



Para nuestro ejemplo, ocurre lo siguiente:

```
SELECT Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
FROM xempleados AS Izq
FULL JOIN xcargos AS Der ON Izq.fkcodcar = Der.pkcodcar

ORDER BY Izq.pkcedemple ;
```

Nótese, que se muestran los cargos “Coordinador” y “Supervisor” a pesar de que no está asignado a ningún empleado:

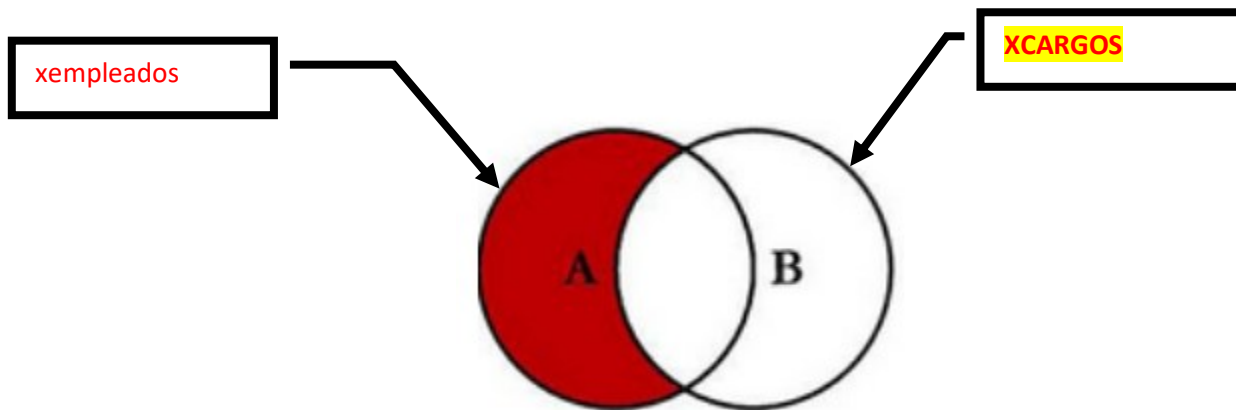
```
bdretardados=# SELECT Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
bdretardados=# FROM xempleados AS Izq
bdretardados=# FULL JOIN xcargos AS Der
bdretardados=# ON Izq.fkcodcar = Der.pkcodcar
bdretardados=# ORDER BY Izq.pkcedemple
```

pkcedemple	nomemple	fecha	fkcodcar	dcargo
1000	ROBERTO JAIMES	1978-03-03	3	GERENTE
2000	ZILA CONTRERAS	1980-10-05	2	VICE PRESIDENTE
3000	MARTHA	1996-11-22	3	GERENTE
3001	RITA	1978-04-03		
3002	JOHANNA	1980-07-20		
3003	VLADIMIR	1998-05-22		
3004	ARMANDO	2007-07-21		
4000	CARLOS	2000-07-15	4	SUB-GERENTE
5000	MATIAS	2005-09-25	4	SUB-GERENTE
6000	NATASHA	2007-10-28	1	PRESIDENTE
7000	PEGGY CARTER	1999-08-29	2	VICE PRESIDENTE
7001	MARIA GIL	2002-04-17		
7002	JAMES	1997-10-28		
7003	FANNY	1999-08-27		
7004	NATHALY	1993-07-23		
7005	THALIA	2003-07-09		
8000	YSA CAPACHO	1993-07-23	2	VICE PRESIDENTE
9000	MARIA	2003-07-07	4	SUB-GERENTE
				COORDINADOR
				SUPERVISOR

(20 filas)

16.05.- Cláusula LEFT JOIN (IS NULL)

Muestra los registros de la tabla izquierda menos los registros coincidentes con la tabla derecha



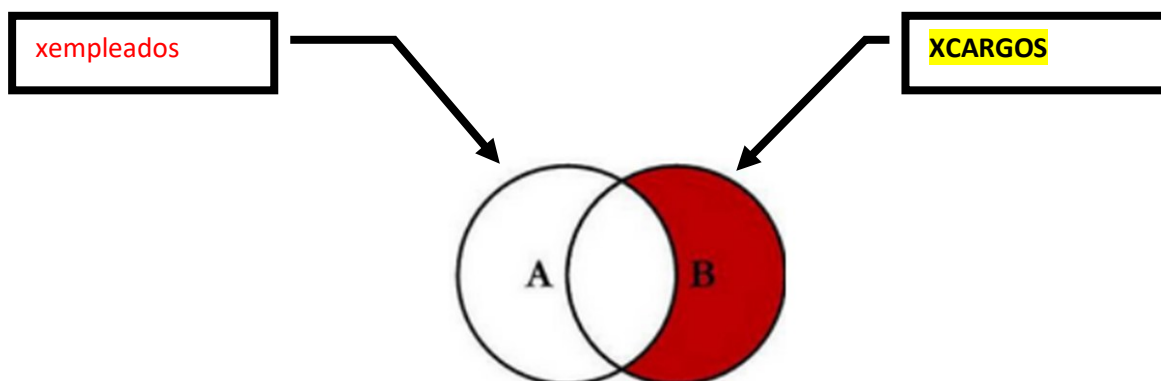
```
SELECT    Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
FROM      empleados AS Izq
LEFT JOIN xcargos AS Der ON Izq.fkcodcar = Der.pkcodcar
WHERE     Der.pkcodcar IS NULL

ORDER BY Izq.pkcedemple ;
```

```
bdretardados=# SELECT      Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
bdretardados=# FROM      empleados AS Izq
bdretardados=# LEFT JOIN xcargos AS Der
bdretardados=# ON        Izq.fkcodcar = Der.pkcodcar
bdretardados=# WHERE     Der.pkcodcar IS NULL
bdretardados=# ORDER BY Izq.pkcedemple ;
 pkcedemple | nomemple | fecha | fkcodcar | dcargo
-----|-----|-----|-----|-----
3001 | RITA | 1978-04-03 | |
3002 | JOHANNA | 1980-07-20 | |
3003 | VLADIMIR | 1998-05-22 | |
3004 | ARMANDO | 2007-07-21 | |
7001 | MARIA GIL | 2002-04-17 | |
7002 | JAMES | 1997-10-28 | |
7003 | FANNY | 1999-08-27 | |
7004 | NATHALY | 1993-07-23 | |
7005 | THALIA | 2003-07-09 | |
(9 filas)
```

16.06.- Cláusula RIGHT JOIN (IS NULL)

Muestra los registros de la tabla derecha menos los registros coincidentes con la tabla izquierda



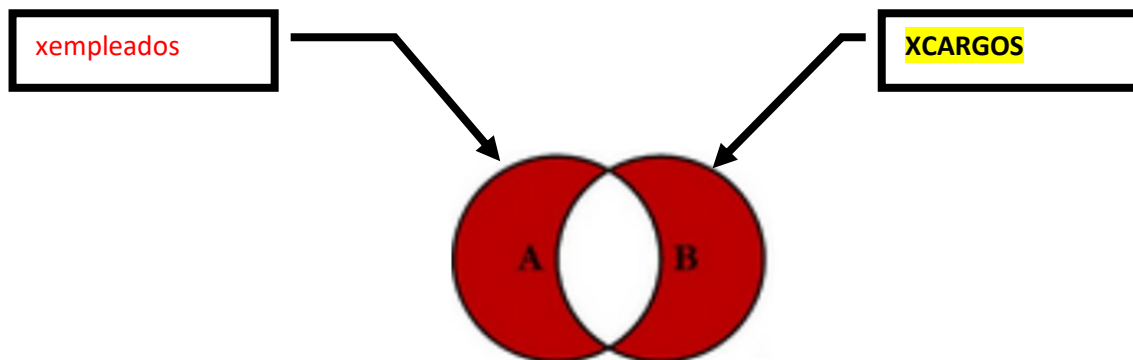
```
SELECT    Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
FROM      xempleados AS Izq
RIGHT JOIN xcargos AS Der ON      Izq.fkcodcar = Der.pkcodcar
WHERE     Izq.fkcodcar IS NULL

ORDER BY  Izq.pkcedemple ;
```

```
bdretardados=# SELECT      Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
bdretardados=# FROM        xempleados AS Izq
bdretardados=# RIGHT JOIN  xcargos AS Der
bdretardados=# ON          Izq.fkcodcar = Der.pkcodcar
bdretardados=# WHERE       Izq.fkcodcar IS NULL
bdretardados=# ORDER BY   Izq.pkcedemple ;
 pkcedemple | nomemple | fecha | fkcodcar | dcargo
-----|-----|-----|-----|-----
          |          |       |          | COORDINADOR
          |          |       |          | SUPERVISOR
(2 filas)
```

16.07.- Cláusula FULL JOIN (IS NULL)

Muestra los registros de la tabla izquierda y la tabla derecha menos los registros coincidentes entre ambas



```
SELECT    Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
FROM      xempleados AS Izq
FULL JOIN xcargos AS Der ON      Izq.fkcodcar = Der.pkcodcar
WHERE     Izq.fkcodcar IS NULL OR Der.pkcodcar IS NULL

ORDER BY  Izq.pkcedemple ;
```

```
bdretardados=# SELECT      Izq.pkcedemple, Izq.nomemple, Izq.fecha, Izq.fkcodcar, Der.dcargo
bdretardados=# FROM      xempleados AS Izq
bdretardados=# FULL JOIN xcargos AS Der
bdretardados=# ON      Izq.fkcodcar = Der.pkcodcar
bdretardados=# WHERE     Izq.fkcodcar IS NULL OR Der.pkcodcar IS NULL
bdretardados=# ORDER BY  Izq.pkcedemple ;
pkcedemple | nomemple | fecha | fkcodcar | dcargo
-----+-----+-----+-----+-----
3001       | RITA    | 1978-04-03 |          |
3002       | JOHANNA | 1980-07-20 |          |
3003       | VLADIMIR | 1998-05-22 |          |
3004       | ARMANDO | 2007-07-21 |          |
7001       | MARIA GIL | 2002-04-17 |          |
7002       | JAMES   | 1997-10-28 |          |
7003       | FANNY   | 1999-08-27 |          |
7004       | NATHALY | 1993-07-23 |          |
7005       | THALIA  | 2003-07-09 |          |
          |          |          |          | COORDINADOR
          |          |          |          | SUPERVISOR
(11 filas)
```

16.08.- Cláusula JOIN con varias tablas

Ejemplo 01:

```
SELECT pkcedemple AS "Cédula",
       Nomemple AS "Nombre",
       fkcodcar AS "Código Cargo",
       xcargos.dcargo,
       xempleados.fkcods AS "Código Status",
       tmstatus.dstatus
FROM xempleados
JOIN xcargos ON xempleados.fkcodcar = xcargos.pkcodcar
JOIN tmstatus ON xempleados.fkcods = tmstatus.pkcods ;
```

```
bdretardados=# SELECT pkcedemple AS "Cédula",
bdretardados=# Nomemple AS "Nombre",
bdretardados=# fkcodcar AS "Código Cargo",
bdretardados=# xcargos.dcargo,
bdretardados=# xempleados.fkcods AS "Código Status",
bdretardados=# tmstatus.dstatus
bdretardados=# FROM xempleados
bdretardados=# JOIN xcargos ON xempleados.fkcodcar = xcargos.pkcodcar
bdretardados=# JOIN tmstatus ON xempleados.fkcods = tmstatus.pkcods ;
 Cédula | Nombre | Código Cargo | dcargo | Código Status | dstatus
-----+-----+-----+-----+-----+-----
 1000 | ROBERTO JAIMES | 3 | GERENTE | 1 | ACTIVO
 2000 | ZILA CONTRERAS | 2 | VICE PRESIDENTE | 1 | ACTIVO
 3000 | MARTHA | 3 | GERENTE | 1 | ACTIVO
 4000 | CARLOS | 4 | SUB-GERENTE | 1 | ACTIVO
 5000 | MATIAS | 4 | SUB-GERENTE | 1 | ACTIVO
 6000 | NATASHA | 1 | PRESIDENTE | 1 | ACTIVO
 7000 | PEGGY CARTER | 2 | VICE PRESIDENTE | 1 | ACTIVO
 8000 | YSA CAPACHO | 2 | VICE PRESIDENTE | 1 | ACTIVO
 9000 | MARIA | 4 | SUB-GERENTE | 1 | ACTIVO
(9 filas)
```

Ejemplo 02:

```
SELECT Izq.pknr,  
       Izq.fkcedemple AS "Cédula",  
       Der1.nomemple AS "Nombre",  
       Izq.fkcodexcu AS "Código",  
       Der2.dexcusa,  
       Izq.fecha,  
       Izq.hora  
FROM   tdretardos AS Izq  
JOIN   tmempleados AS Der1 ON Izq.fkcedemple = Der1.pkcedemple  
JOIN   tmexcusas AS Der2 ON Izq.fkcodexcu = Der2.pkcodexcu ;
```

```
bdretardados=# SELECT Izq.pknr,  
bdretardados=#       Izq.fkcedemple AS "Cédula",  
bdretardados=#       Der1.nomemple AS "Nombre",  
bdretardados=#       Izq.fkcodexcu AS "Código",  
bdretardados=#       Der2.dexcusa,  
bdretardados=#       Izq.fecha,  
bdretardados=#       Izq.hora  
bdretardados=# FROM   tdretardos AS Izq  
bdretardados=# JOIN   tmempleados AS Der1 ON Izq.fkcedemple = Der1.pkcedemple  
bdretardados=# JOIN   tmexcusas AS Der2 ON Izq.fkcodexcu = Der2.pkcodexcu ;
```

pknr	Cédula	Nombre	Código	dexcusa	fecha	hora
1	2000	ZILA CONTRERAS	3	NO ME BUSCARON	2022-04-05	08:32:00
2	5000	Thalía	3	NO ME BUSCARON	2022-04-05	08:50:00
3	4000	CARLOS	1	ME QUEDE DORMIDO	2022-04-06	09:09:00
4	3000	MARTHA	5	MUCHA LLUVIA	2022-04-06	09:13:00
5	6000	NATASHA	7	EL CARRO NO ME QUIERE	2022-04-11	08:21:00
6	1000	ROBERTO JAIMES	2	SALI TARDE DE LA CASA	2022-04-14	08:16:00
7	2000	ZILA CONTRERAS	6	MUCHO TRAFICO	2022-04-17	10:00:00
8	3000	MARTHA	7	EL CARRO NO ME QUIERE	2022-04-19	08:11:00
9	2000	ZILA CONTRERAS	3	NO ME BUSCARON	2022-04-21	08:13:00
10	1000	ROBERTO JAIMES	8	ME SECUESTRARON	2022-04-21	08:15:00
11	5000	Thalía	2	SALI TARDE DE LA CASA	2022-04-21	08:20:00
12	2000	ZILA CONTRERAS	6	MUCHO TRAFICO	2022-04-23	08:27:00
13	3000	MARTHA	4	PELIE CON EL AGUILA	2022-04-23	03:30:00
14	4000	CARLOS	2	SALI TARDE DE LA CASA	2022-04-23	08:45:00
15	6000	NATASHA	5	MUCHA LLUVIA	2022-04-25	08:05:00
16	2000	ZILA CONTRERAS	4	PELIE CON EL AGUILA	2022-04-25	08:32:00
17	4000	CARLOS	1	ME QUEDE DORMIDO	2022-04-27	09:00:00

Ejemplo 03:

```

SELECT pknr, fkcodexcu AS "Código", dexcusa AS "Excusa",
       fkcedemple AS "Cédula", nomemple AS "Nombre",
       tdretardos.fecha AS "Fecha", tdretardos.hora AS "Hora"
FROM   tdretardos
FULL JOIN tmempleados ON fkcedemple = pkcedemple
FULL JOIN tmexcusas   ON fkcodexcu  = pkcodexcu

WHERE   fkcodexcu IN (2, 4, 6)

ORDER BY tdretardos.fkcodexcu, tdretardos.fkcedemple, tdretardos.fecha ;

```

pknr	Código	Excusa	Cédula	Nombre	Fecha	Hora
6	2	SALI TARDE DE LA CASA	1000	ROBERTO JAIMES	2022-04-14	08:16:00
88	2	SALI TARDE DE LA CASA	1000	ROBERTO JAIMES	2022-07-01	08:40:00
93	2	SALI TARDE DE LA CASA	2000	ZILA CONTRERAS	2022-07-15	08:23:00
24	2	SALI TARDE DE LA CASA	3000	MARTHA	2022-05-01	08:40:00
103	2	SALI TARDE DE LA CASA	3000	MARTHA	2022-05-13	08:17:00
76	2	SALI TARDE DE LA CASA	3000	MARTHA	2022-06-01	08:40:00
114	2	SALI TARDE DE LA CASA	3000	MARTHA	2022-11-13	08:17:00
40	2	SALI TARDE DE LA CASA	4000	CARLOS	2022-02-15	08:23:00
14	2	SALI TARDE DE LA CASA	4000	CARLOS	2022-04-23	08:45:00
29	2	SALI TARDE DE LA CASA	4000	CARLOS	2022-05-15	08:23:00
32	2	SALI TARDE DE LA CASA	4000	CARLOS	2022-05-26	08:17:00
81	2	SALI TARDE DE LA CASA	4000	CARLOS	2022-06-15	08:23:00
84	2	SALI TARDE DE LA CASA	4000	CARLOS	2022-06-26	08:17:00
54	2	SALI TARDE DE LA CASA	5000	MATIAS	2022-03-21	08:20:00
11	2	SALI TARDE DE LA CASA	5000	MATIAS	2022-04-21	08:20:00
69	2	SALI TARDE DE LA CASA	6000	NATASHA	2022-01-21	08:24:00
96	2	SALI TARDE DE LA CASA	6000	NATASHA	2022-07-26	08:17:00
46	4	PELIE CON EL AGUILA	2000	ZILA CONTRERAS	2022-03-05	08:32:00
16	4	PELIE CON EL AGUILA	2000	ZILA CONTRERAS	2022-04-25	08:32:00
13	4	PELIE CON EL AGUILA	3000	MARTHA	2022-04-23	03:30:00
62	4	PELIE CON EL AGUILA	6000	NATASHA	2022-01-05	08:32:00
105	4	PELIE CON EL AGUILA	6000	NATASHA	2022-07-16	09:15:00
117	4	PELIE CON EL AGUILA	6000	NATASHA	2022-11-16	09:15:00
121	4	PELIE CON EL AGUILA	6000	NATASHA	2022-11-27	08:25:00
108	4	PELIE CON EL AGUILA	7000	PEGGY CARTER	2022-10-27	08:25:00
116	4	PELIE CON EL AGUILA	7000	PEGGY CARTER	2022-11-10	08:27:00
30	6	MUCHO TRAFICO	1000	ROBERTO JAIMES	2022-05-16	09:05:00
82	6	MUCHO TRAFICO	1000	ROBERTO JAIMES	2022-06-16	09:05:00
106	6	MUCHO TRAFICO	1000	ROBERTO JAIMES	2022-08-19	09:19:00
118	6	MUCHO TRAFICO	1000	ROBERTO JAIMES	2022-11-19	09:19:00
52	6	MUCHO TRAFICO	2000	ZILA CONTRERAS	2022-03-17	10:00:00
55	6	MUCHO TRAFICO	2000	ZILA CONTRERAS	2022-03-23	08:27:00
7	6	MUCHO TRAFICO	2000	ZILA CONTRERAS	2022-04-17	10:00:00
12	6	MUCHO TRAFICO	2000	ZILA CONTRERAS	2022-04-23	08:27:00
122	6	MUCHO TRAFICO	3000	MARTHA	2021-11-22	08:41:00
109	6	MUCHO TRAFICO	3000	MARTHA	2021-11-22	08:41:00

Ejemplo 04:

```
SELECT pknr,  
       fkcedemple AS "Cédula",  
       nomemple AS "Nombre",  
       fkcodexcu AS "Código",  
       dexcusa,  
       tdretardos.fecha,  
       tdretardos.hora  
FROM   tdretardos  
INNER JOIN tmempleados ON fkcedemple = pkcedemple  
INNER JOIN tmexcusas   ON fkcodexcu = pkcodexcu  
WHERE  fkcedemple = '5000'  
  
ORDER BY tdretardos.fkcodexcu, tdretardos.fecha ;
```

```
bdretardados=# SELECT pknr,  
bdretardados=#         fkcedemple AS "Cédula",  
bdretardados=#         nomemple AS "Nombre",  
bdretardados=#         fkcodexcu AS "Código",  
bdretardados=#         dexcusa,  
bdretardados=#         tdretardos.fecha,  
bdretardados=#         tdretardos.hora  
bdretardados=# FROM   tdretardos  
bdretardados=# INNER JOIN tmempleados ON fkcedemple = pkcedemple  
bdretardados=# INNER JOIN tmexcusas   ON fkcodexcu = pkcodexcu  
bdretardados=# WHERE  fkcedemple = '5000'  
bdretardados=# ORDER BY tdretardos.fkcodexcu, tdretardos.fecha ;  
 pknr | Cédula | Nombre | Código | dexcusa | fecha | hora  
-----+-----+-----+-----+-----+-----+-----  
 54 | 5000 | Thalía | 2 | SALI TARDE DE LA CASA | 2022-03-21 | 08:20:00  
 11 | 5000 | Thalía | 2 | SALI TARDE DE LA CASA | 2022-04-21 | 08:20:00  
 2 | 5000 | Thalía | 3 | NO ME BUSCARON | 2022-04-05 | 08:50:00  
 91 | 5000 | Thalía | 3 | NO ME BUSCARON | 2022-07-07 | 08:20:00  
 22 | 5000 | Thalía | 5 | MUCHA LLUVIA | 2022-01-12 | 08:20:00  
 37 | 5000 | Thalía | 6 | MUCHO TRAFICO | 2022-02-01 | 09:00:00  
 47 | 5000 | Thalía | 6 | MUCHO TRAFICO | 2022-03-05 | 08:50:00  
102 | 5000 | Thalía | 6 | MUCHO TRAFICO | 2022-04-07 | 08:21:00  
113 | 5000 | Thalía | 6 | MUCHO TRAFICO | 2022-11-07 | 08:21:00  
 66 | 5000 | Thalía | 7 | EL CARRO NO ME QUIERE | 2022-01-11 | 08:17:00  
(10 filas)
```

Ejemplo 05:

```
SELECT pknr,  
        fkcedemple AS "Cédula",  
        nomemple AS "Nombre",  
        fkcodexcu AS "Código",  
        dexcusa,  
        tdretardos.fecha,  
        tdretardos.hora  
FROM   tdretardos , templeados , tmexcusas  
WHERE  (fkcedemple = pkcedemple) and  
        (fkcodexcu = pkcodexcu) and  
        (fkcedemple = '5000' )  
  
ORDER BY tdretardos.fkcodexcu, tdretardos.fecha ;
```

```
SELECT   pknr,  
          fkcedemple AS "Cédula",  
          nomemple AS "Nombre",  
          fkcodexcu AS "Código",  
          dexcusa AS "Excusa",  
          tdretardos.fecha AS "Fecha",  
          tdretardos.hora AS "Hora"  
FROM   tdretardos  
INNER JOIN templeados ON fkcedemple = pkcedemple  
INNER JOIN tmexcusas ON fkcodexcu = pkcodexcu  
WHERE   fkcedemple = '5000'  
  
ORDER BY tdretardos.fkcodexcu, tdretardos.fecha ;
```

Ejemplo 06: Determinara la cantidad Total de los pagos hechos a un bioanalista con el número de cédula '114567839'

```
SELECT    pkcc_bio as cedula, nombre_bio as nombre, COUNT(*) as total_pagos
FROM      tdpagos
JOIN      tmbioanalistas ON fkcc_bio = pkcc_bio
WHERE     pkcc_bio = '114567839'

GROUP BY pkcc_bio, nombre_bio ;
```

RESPUESTA

cedula	nombre	total_pagos
114567839	Oscar Valencia	4

(1 fila)

```
SELECT    fkcc_bio AS bioanalista, COUNT(*) AS cantidad_pagos
FROM      tdpagos
WHERE     fkcc_bio = '114567839'

GROUP BY fkcc_bio ;
```

Ejemplo 07: Generar un listado donde se muestren la cedula, nombre, la descripción del cargo, la descripción del estado civil, el código del sexo y edad actual de todos los bioanalistas registrados en la base de datos debidamente ordenados en forma ascendente por el campo cedula.

```
SELECT    pkcc_bio      AS "Cedula",
           nombre_bio  AS "Nombre",
           dcargo      AS "Cargo",
           destado_civil AS "Est. Cívil",
           dsexo       AS "Sexo",
           date_part('year', age(current_date, fecha_nac_bio) ) AS "Edad Actual"
FROM      tmbioanalistas
JOIN      tmcargos      ON   fkcod_car = pkcod_car
JOIN      tmestados_civiles ON  fkcod_ec = pkcod_ec
JOIN      tmsexo        ON   fkcod_sx = pkcod_sx

ORDER BY pkcc_bio ;
```

Ejemplo 08:

```
SELECT    pknr, fkcedemple AS "Cédula", nomemple AS "Nombre",
           fkcodexcu AS "Código", dexcusa AS "Excusa", tdretardos.fecha AS "Fecha",
           tdretardos.hora AS "Hora"
FROM      tdretardos
INNER JOIN tmempleados ON   fkcedemple = pkcedemple
INNER JOIN tmexcusas   ON   fkcodexcu   = pkcodexcu
WHERE     fkcedemple = '5000'

ORDER BY tdretardos.fkcodexcu, tdretardos.fecha ;
```

JOIN - CAST()

SELECT

```
p.npago          AS "Numero de Pago",
a.cedalum       AS "Cedula Estudiante",
a.apealum       AS "Apellido Estudiante",
d.ceddocente    AS "Cedula Docente",
d.apdocente     AS "Apellido Docente",
b.cod_ban       AS "Codigo Banco",
b.nom_ban       AS "Nombre Banco",
p.ndeposi       AS "Numero Depósito",
p.monto         AS "Monto",
p.fechareg      AS "Fecha Registro"
```

FROM tdpagos p

JOIN tmalumnos a **ON** p.fkcc_alum = a.cedalum

JOIN tmdocentes d **ON** p.fkcc_doc = d.ceddocente

JOIN tmbancos b **ON** p.fkcodban = b.cod_ban

ORDER BY **CAST(** d.ceddocente **AS INTEGER)** ASC, b.cod_ban DESC ;

16.- Borrar tabla / base de datos

16.1.- Drop Table --> Permite borrar físicamente una tabla, siempre y cuando se mantenga la integridad referencial.

DROP TABLE Tabla ;

Ejemplo:

DROP TABLE Tusuarios ;

```
bdcapacho02=# \d
Listado de relaciones
Esquema | Nombre | Tipo | Dueño
-----|-----|-----|-----
public  | tdretardos | tabla | postgres
public  | tdretardos_nr_seq | secuencia | postgres
public  | tmcargos | tabla | postgres
public  | tmcargos_codcar_seq | secuencia | postgres
public  | templeados | tabla | postgres
public  | tmexcusas | tabla | postgres
public  | tmexcusas_codexcu_seq | secuencia | postgres
public  | tmstatus | tabla | postgres
public  | tusuarios | tabla | postgres
public  | tusuarios_cu_seq | secuencia | postgres
(10 filas)

bdcapacho02=# DROP TABLE tusuarios;
DROP TABLE
bdcapacho02=#
bdcapacho02=# \d
Listado de relaciones
Esquema | Nombre | Tipo | Dueño
-----|-----|-----|-----
public  | tdretardos | tabla | postgres
public  | tdretardos_nr_seq | secuencia | postgres
public  | tmcargos | tabla | postgres
public  | tmcargos_codcar_seq | secuencia | postgres
public  | templeados | tabla | postgres
public  | tmexcusas | tabla | postgres
public  | tmexcusas_codexcu_seq | secuencia | postgres
public  | tmstatus | tabla | postgres
(8 filas)

bdcapacho02=# _
```

16.2.- Drop DataBase --> Permite borrar físicamente todo el contenido de una base de datos.

Nota: Es importante indicar para poder eliminar una base de datos, se debe estar fuera de ella.

\c OtraBaseDatos

OtraBaseDatos=# DROP DATABASE BaseDatos ;

Ejemplo:

DROP DATABASE BDCapacho02 ;

```
bdcapacho02=# \l
      Nombre | Dueño | Codificación | Listado
-----|-----|-----|-----
 bdcapacho02 | postgres | UTF8 | Spanish_C
 postgres | postgres | UTF8 | Spanish_C
 template0 | postgres | UTF8 | Spanish_C
 template1 | postgres | UTF8 | Spanish_C
(4 filas)

bdcapacho02=# \c postgres
Ahora está conectado a la base de datos «postgres»
postgres=# DROP DATABASE bdcapacho02;
DROP DATABASE
postgres=# \l
      Nombre | Dueño | Codificación | Listado
-----|-----|-----|-----
 postgres | postgres | UTF8 | Spanish_Col
 template0 | postgres | UTF8 | Spanish_Col
 template1 | postgres | UTF8 | Spanish_Col
(3 filas)

postgres=#
```

17.- Orden para escribir un comando SQL:

- SELECT
- FROM
- JOIN ON
- WHERE
- GROUP BY
- HAVING
- ORDER BY
- LIMIT ... OFFSET



Fuente

- 1.- <https://www.postgresql.org/docs/current/>
- 2.- <https://neon.tech/postgresql/tutorial>
- 3.- <https://www.w3schools.com/postgresql/>
- 5.- <https://www.todopostgresql.com/comandos-postgresql-alter/>
- 6.- <https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-alter-table/>
- 7.- <https://www.postgresql.org/docs/current/ddl-alter.html>
- 8.- <https://www.postgresql.org/docs/current/sql-altertable.html>
- 9.- https://www.w3schools.com/sql/sql_ref_group_by.asp
- 10.-
<https://www.tutorialesprogramacionya.com/postgresqlya/temarios/descripcion.php?inicio=0&cod=181&pagina=23>
- 11.- <https://www.w3schools.com/sql/>
- 12.- Fuente: <https://programacionymas.com/blog/como-funciona-inner-left-right-full-join>
- 13.- LISTADO DE FUNCIONES DE LIBRERIA SQL DE POSTGRESQL
<https://learn.microsoft.com/es-es/sql/t-sql/functions/functions?view=sql-server-ver16>
- 14.- <https://www.pgadmin.org/docs/pgadmin4/development/genindex.html>