



PostgreSQL

Uso de PostgreSQL Select

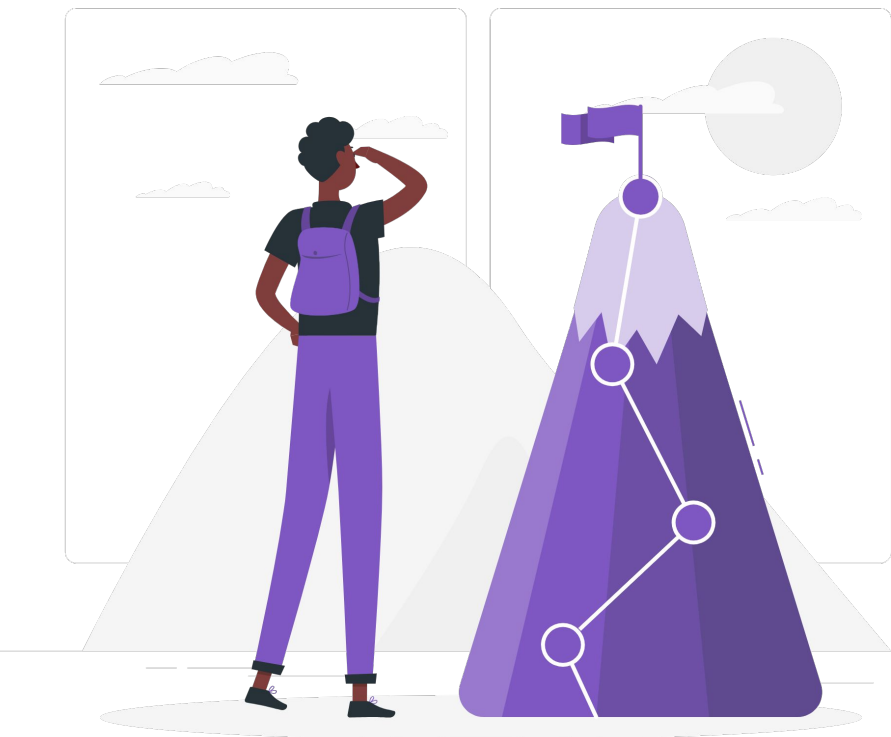
DEV.F.
DESARROLLAMOS(PERSONAS);

Elaborado por: César Guerra

www.cesarquerra.mx



Buy me a coffee



Objetivos de la Sesión

- Aprender a cargar una base de datos con pgAdmin (dvd rentals)
- Ejecutar consultas SELECT para traer todos los elementos
- Ejecutar consultas SELECT para traer ciertos elementos (condicionales)
- Ejecutar consultas SELECT para traer valores ordenados de cierta forma.
- Ejecutar consultas SELECT que realicen operaciones.

Setup de Base de Datos DVD Rentals

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Descargar Base de Datos DVD Rental

Descargar de:

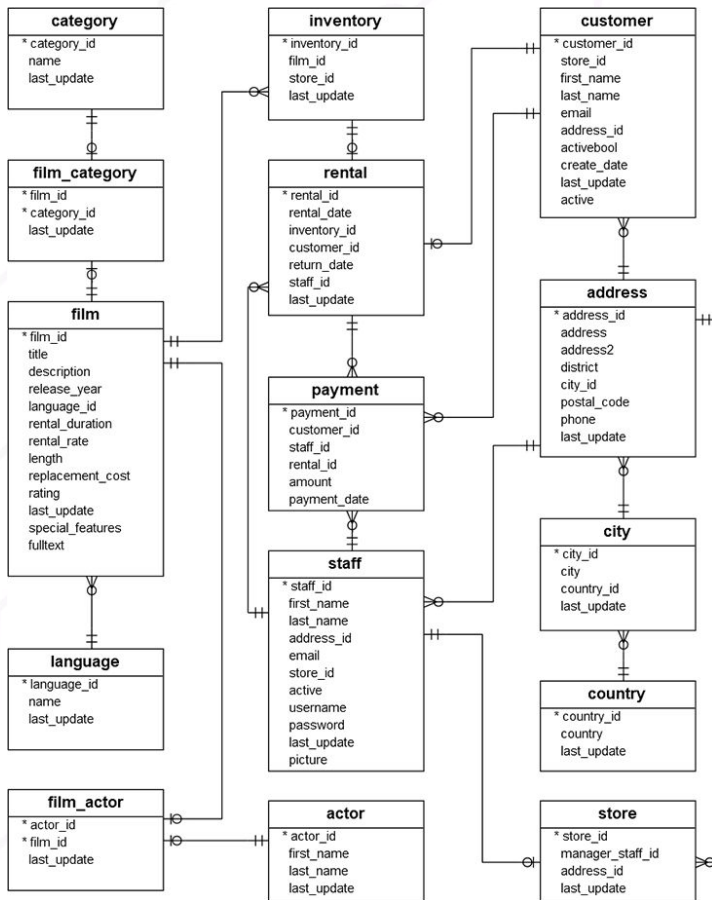
<https://www.postgresqltutorial.com/postgresql-sample-database/>

Download PostgreSQL sample database

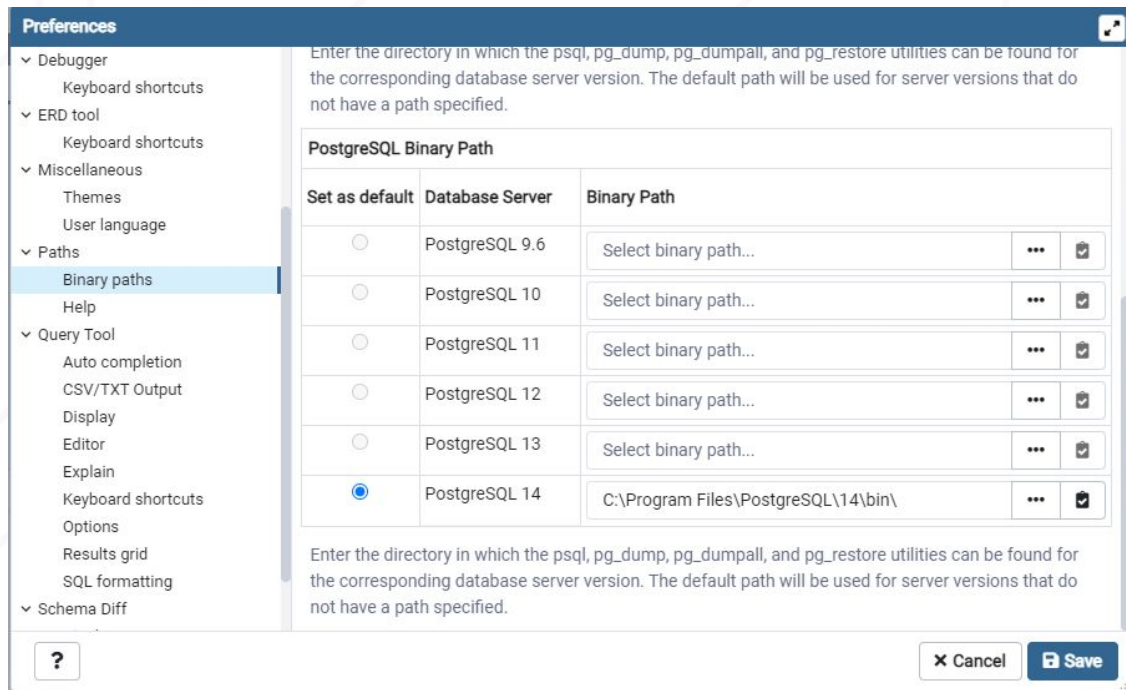
You can download the PostgreSQL DVD Rental sample database via the following link:

Download DVD Rental Sample Database

The database file is in `zip` format (`dvdrental.zip`) so you need to extract it to `dvdrental.tar` before loading the sample database into the PostgreSQL database server.



Establecer Binary Path



Ir a las preferencias de pgAdmin y establecer la ruta del binario donde se encuentra instalado PostgreSQL.

C:\Program Files\PostgreSQL\14\bin\

Restaurar base de datos

Restore (Database: dvdrental)

General | Restore options

Format: Custom or tar

Filename: D:\Downloads\dvdrental.tar

Number of jobs:

Role name: Select an item...

i ? X Cancel Restore

Como restaurar(cargar) una base de datos:

<https://www.postgresqltutorial.com/load-postgresql-sample-database/>

Consultas a Base de Datos

SELECT

DEV.FX
DESARROLLAMOS(PERSONAS);

dev



SELECT

Una base de datos nos permite almacenar información, pero para qué esta sea útil también nos debe permitir recuperar la información almacenada.

Una tarea común al trabajar con bases de datos es ejecutar consultas (queries) con la finalidad de recuperar la información contenida en las tablas por medio de SELECT.

Los resultados de la consulta van a ser datos individuales, tuplas(filas) o tablas generados a partir de consultas en las que se establecen una serie de condiciones.

SELECT es de las instrucciones más complejas en PostgreSQL debido a todas las cláusulas que se pueden usar para darle flexibilidad a las búsquedas.

SINTAXIS BÁSICA SELECT

Sintáxis mínima

SELECT campos
FROM tabla

Notas:

Si se necesita más de un campo, se colocan los demás separados por coma.

Si se necesitan todos los campos, se puede abreviar con un *

Ejemplos en DVD Rentals

Seleccionar todos los campos de actores (tabla actor)

```
SELECT * FROM actor;
```

Seleccionar el nombre y el id de los actores (tabla actor)

```
SELECT first_name, actor_id FROM actor;
```

Seleccionar todos los datos de la tabla clientes (customer)

RESPUESTA

SINTAXIS SELECT CON LIMITACIÓN DE REGISTROS

Sintaxis

SELECT campos
FROM tabla
LIMIT #_de_registros

Notas:

LIMIT nos ayuda a solo devolver una cantidad limitada de registros de la base de datos en vez de todos los registros.

Ejemplos en DVD Rentals

Seleccionar todos los campos de clientes (customers) pero solo traer 5 resultados

```
SELECT * FROM customer LIMIT 5;
```

Seleccionar todos los campos de películas (film) pero solo traer 10 resultados

```
SELECT * FROM film LIMIT 5;
```

SINTAXIS SELECT CON LIMITACIÓN DE REGISTROS

Sintaxis

SELECT campos
FROM tabla
LIMIT #_de_registros

Notas:

LIMIT nos ayuda a solo devolver una cantidad limitada de registros de la base de datos en vez de todos los registros.

Ejemplos en DVD Rentals

Seleccionar todos los campos de clientes (customers) pero solo traer 5 resultados

```
SELECT * FROM customer LIMIT 5;
```

Seleccionar todos los campos de películas (film) pero solo traer 10 resultados

```
SELECT * FROM film LIMIT 5;
```

SINTAXIS SELECT CON SKIP DE REGISTROS

Sintaxis

SELECT campos

FROM tabla

LIMIT #_de_registros

OFFSET #_de_registros

Notas:

OFFSET nos ayuda a brincaros cierta cantidad de registros.
Se suele usar en conjunto con LIMIT.

Ejemplos en DVD Rentals

Seleccionar todos los campos de clientes (customers), trayendo los 5 siguientes registros después de los primeros 100

```
SELECT * FROM customer LIMIT 5 OFFSET 100;
```

SINTAXIS SELECT CONDICIONAL

SELECT campos
FROM tabla
WHERE condición

WHERE es una de las instrucciones más usada, ya que nos permite “filtrar” solo ciertos datos que cumplan con la condición especificada.

Es posible usar **negación(!=)**, **and** y **or** en dichas condicionales.

Ejemplos en DVD Rentals

Seleccionar todos los campos de clientes (customers) qué se llamen John.

SELECT * FROM customer **WHERE** first_name='John';

Seleccionar todos los campos de customers pero solo si sus ids están entre 4 y 100.

SELECT * FROM customer **WHERE** customer_id>=4 **AND** customer_id<=100;

Seleccionar todos los campos de customers que se llamen John o Barbara.

SELECT * FROM customer **WHERE** first_name='John' **OR** first_name='Barbara';

Seleccionar todos los campos de customers, menos los qué llamen John y Barbara.

SELECT * FROM customer **WHERE** first_name!='John' **AND** first_name!='Barbara';

Seleccionar todos los campos de películas que tengan un rating de PG o G.

RESPUESTA

BUSCAR VALORES ENTRE RANGO (BETWEEN - AND)

Sintaxis

SELECT campos
FROM tabla
WHERE condición
BETWEEN #
AND #

Notas:

WHERE con **BETWEEN** y **AND** nos ayudan a establecer una condición donde regresará solo los datos dentro del rango numérico especificado.

Ejemplos en DVD Rentals

Seleccionar todos los campos de películas (film) donde el id este entre 8 y 4.

```
SELECT * FROM film WHERE film_id BETWEEN 8 AND 14;
```

BUSCAR VALORES ESPECÍFICOS (WHERE IN)

Sintaxis

SELECT campos
FROM tabla
WHERE condición
IN (valor1, valor2, ...)

Notas:

WHERE IN nos ayudan a establecer una condición donde regresará solo los datos que se encuentren dentro de la lista especificada en **IN**.

Ejemplos en DVD Rentals

Seleccionar todos los campos de películas (film) donde los id sean 9,12,99,22, y 44.

```
SELECT * FROM film WHERE film_id IN (9,12,99,22,44)
```

BUSCAR VALORES NULOS (NULL) Y VACÍOS

Sintaxis

SELECT campos
FROM tabla
WHERE condición
IS NULL

Notas:

Los valores **nulos(null)** son diferentes de los valores **vacíos**.

address character varying (50)	address2 character varying (50)	district character varying (20)	city_id smallint	postal_code character varying (10)
47 MySakila Drive	[null]	Alberta	300	
28 MySQL Boulevard	[null]	QLD	576	

Cuando al insertar un registro no se especifica un valor para ese campo, se le asigna **null**. Si consideramos ese dato, pero colocamos "", es vacío.

Ejemplos en DVD Rentals

Seleccionar todos los campos de dirección (address) donde el address2 sea NULO.

SELECT * FROM address WHERE address2 IS NULL;

Seleccionar todos los campos de dirección (address) donde el address2 sea VACÍO.

SELECT * FROM address WHERE address2 = ""; → (lleva 2 comillas simples)

ORDENAR REGISTROS (ORDER BY)

Sintaxis

SELECT campos
FROM tabla
WHERE cond.(opcional)
ORDER BY campo(s)
DESC (opcional)

Notas:

ORDER BY nos permite ordenar los resultados en orden ASCENDENTE (por defecto) o descendente (indicando **DESC** al final) de acuerdo a un campo especificado de ordenamiento.

Ejemplos en DVD Rentals

Seleccionar todos los campos de películas (film) y ordenarlos por su id.

SELECT * FROM film ORDER BY film_id; → *Ordenado Ascendente*

SELECT * FROM film ORDER BY film_id DESC; → *Ordenado Descendente*

Seleccionar customer_id y rental_id de rental y ordenarlos por customer_id y luego por rental_id

SELECT customer_id, rental_id FROM rental ORDER BY customer_id, rental_id;

SEL. LA CANTIDAD MENOR O MAYOR (MIN & MAX)

```
SELECT MIN(campo)  
FROM tabla  
WHERE condición
```

```
SELECT MAX(campo)  
FROM tabla  
WHERE condición
```

Ejemplos en DVD Rentals

Seleccionar la cantidad más pequeña de payments

```
SELECT MIN(amount) FROM payment;
```

Seleccionar la cantidad más grande de payments

```
SELECT MAX(amount) FROM payment;
```

Seleccionar la cantidad más grande de payments, pero con un tope máximo (< 7)

```
SELECT MAX(amount) FROM payment WHERE amount < 7;
```

Notas:

Al especificar **MIN** o **MAX** después del **SELECT**, buscará el valor mínimo o máximo de acuerdo a la consulta ejecutada. También puede delimitarse aún más el resultado complementando con un **WHERE** (condición).

CONTAR TODOS LOS ELEMENTOS (COUNT)

Sintaxis

```
SELECT  
COUNT(*)  
FROM tabla  
WHERE cond.(opcional)
```

Notas:

COUNT nos devuelve el valor número de la cantidad de resultados de una consulta (query). Es muy útil cuando queremos saber CUÁNTOS elementos cumplen cierto criterio o cuántos tenemos. Si queremos que los valores que se repitan solo sean contabilizados una sola vez, usamos **COUNT(DISTINCT campo)**, para en base a ese campo, evitar repeticiones.

Ejemplos en DVD Rentals

¿Cuántos clientes (customer) tenemos?

```
SELECT COUNT(*) FROM customer;
```

¿Cuántos códigos postales diferentes tenemos en nuestra tabla de direcciones (address)?

```
SELECT COUNT (DISTINCT postal_code) FROM address;
```

AGRUPAR CONSULTAS (GROUP BY)

Sintaxis

```
SELECT  
COUNT(* ó campo)  
FROM tabla  
WHERE cond.(opcional)  
GROUP BY campo
```

Ejemplos en DVD Rentals

¿Cuántas películas clasificación PG, PG-13 y NC-17 tenemos?

```
SELECT rating, COUNT(rating)  
FROM film  
WHERE rating='PG' OR rating = 'PG-13' OR rating='NC-17'  
GROUP BY rating;
```

Notas:

GROUP BY es bastante usado para agrupar resultados similares que cumplan la condición, y es ampliamente usado con operaciones que tengan que ver con cálculos (contar, promedios, etc.)

	Data Output	Explain	Message
	rating mpaa_rating	count bigint	
1	PG	194	
2	PG-13	223	
3	NC-17	210	

AGRUPAR CONSULTAS POR PROMEDIO (AVG GROUP BY)

Sintáxis

```
SELECT  
AVG(campo)  
FROM tabla  
WHERE cond.(opcional)  
GROUP BY campo
```

Ejemplos en DVD Rentals

¿Cuál es el gasto promedio de cada cliente?

```
SELECT customer_id,  
AVG (amount)  
FROM payment  
GROUP BY customer_id;
```


Notas:

AVG (average) nos permite calcular el promedio de un campo sobre una determinada consulta. Se suele usar con **GROUP BY** para ver los resultados de forma agrupada (compacta).

	customer_id smallint	avg numeric
1	184	0000000000000
2	87	714285714286
3	477	380952380952
4	273	714285714286
5	550	258064516129
6	51	333333333333

Aplicando lo aprendido y complicando las cosas...

¿Calcular el gasto promedio de los clientes, ordenado por cantidad de mayor a menor? (Mi top de clientes)

 dvdrental/master@devf

Query Editor

Query History




1 **SELECT** customer_id, **avg**(amount) **FROM** payment **GROUP BY** customer_id **ORDER BY** avg(amount) **DESC**

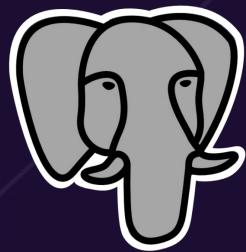
Data Output

Explain

Messages

Notifications

	 customer_id smallint	 avg numeric	
1	288	2.7794736842105263	
2	557	2.8990909090909091	
3	248	2.9130769230769231	
4	252	2.9400000000000000	
5	586	2.9900000000000000	
6	283	2.9900000000000000	



PostgreSQL

Ejercicios #3

Consultas SELECT

DEV.F.
DESARROLLAMOS(PERSONAS);

Elabora las siguientes Querys en dvd rentals

1. Selecciona las columnas **film_id** y title de la tabla **film**.
2. Seleccionar 5 filas de la tabla **film**, obteniendo todas las columnas.
3. Seleccionar las filas de la tabla **film** donde el **film_id** es menor que 4.
4. Seleccionar las filas de la tabla **actor** donde su nombre sean: Angela, Angelina, o Audrey usando **IN**.
5. Obtenga una lista de actores con el nombre de Julia.
6. Obtenga una lista de actores con el nombre Chris, Cameron o Cuba.
7. Seleccione la fila del cliente para el cliente llamado Jamie Rice.
8. Seleccione el monto y la fecha de pago del pago fue inferior a \$ 1.
9. ¿Cuáles son los ID de los últimos 3 clientes para devolver un alquiler?
10. ¿Cuántas películas tienen la clasificación NC-17, PG y PG-13?
11. ¿Qué film (id) tiene más actores?
12. ¿Qué actor (id) aparece en más películas?
13. ¿Cuáles son los diferentes periodos de renta que permite la tienda?
14. ¿Cuál es el costo de reemplazo promedio de las películas por rating?



¡GRACIAS!



César Guerra



www.cesarguerra.mx