

## Funciones de agregación

### Funciones de agregación

Las funciones de agregación en SQL nos permiten efectuar operaciones sobre un conjunto de resultados, pero devolviendo un único valor agregado para todos ellos. Es decir, nos permiten obtener medias, máximos, etc... sobre un conjunto de valores.

Las funciones de agregación básicas que soportan todos los gestores de datos son las siguientes:

1. COUNT: Supongamos que tenemos una tabla llamada "Productos" con una columna llamada "Nombre". Para contar el número total de productos en la tabla, puedes usar la función COUNT de la siguiente manera:

`SELECT COUNT(*) FROM Productos;`  
Esto devolverá el número total de filas (productos) en la tabla.

2. MIN: Supongamos que tenemos una tabla llamada "Ventas" con una columna llamada "Precio". Para encontrar el valor mínimo de los precios de venta, puedes usar la función MIN de la siguiente manera:

`SELECT MIN(Precio) FROM Ventas;`  
Esto devolverá el valor mínimo de la columna "Precio".

3. MAX: Supongamos que tenemos una tabla llamada "Ventas" con una columna llamada "Precio". Para encontrar el valor máximo de los precios de venta, puedes usar la función MAX de la siguiente manera:

`SELECT MAX(Precio) FROM Ventas;`  
Esto devolverá el valor máximo de la columna "Precio".

4. SUM: Supongamos que tenemos una tabla llamada "Ventas" con una columna llamada "Cantidad". Para sumar todos los valores de la columna "Cantidad", puedes usar la función SUM de la siguiente manera:

`SELECT SUM(Cantidad) FROM Ventas;`  
Esto devolverá la suma de todos los valores de la columna "Cantidad".

5. AVG: Supongamos que tenemos una tabla llamada "Ventas" con una columna llamada "Cantidad". Para calcular el promedio de los valores de la columna "Cantidad", puedes usar la función AVG de la siguiente manera:

`SELECT AVG(Cantidad) FROM Ventas;`

## Relación cuentas con usuario

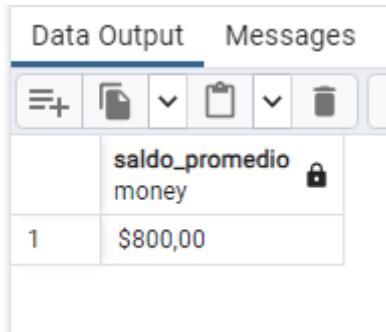
**Función de agregación 1:** Obtener el saldo promedio de todas las cuentas de un usuario específico.

### Requisitos:

- Conocer la cédula del usuario específico.

### Instrucciones:

1. Escribir la sentencia SQL utilizando la función de agregación AVG para calcular el promedio del saldo de las cuentas.
2. Utilizar la cláusula WHERE para filtrar las cuentas del usuario específico utilizando su cédula.
3. Ejecutar la consulta en la base de datos.

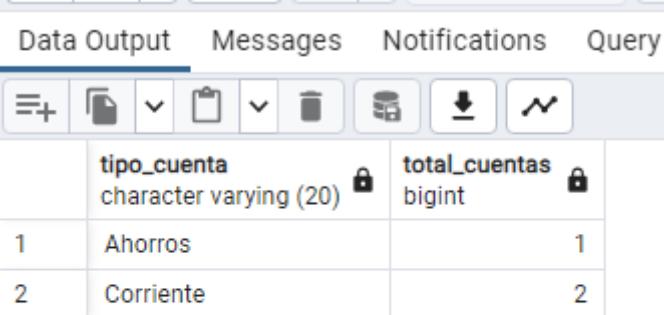


	saldo_promedio	money
1	\$800,00	

**Función de agregación 2:** Obtener el número total de cuentas de cada tipo de cuenta.

**Instrucciones:**

1. Escribir la sentencia SQL utilizando la función de agregación COUNT para contar el número de cuentas.
2. Utilizar la cláusula GROUP BY para agrupar las cuentas por tipo de cuenta.
3. Ejecutar la consulta en la base de datos.



The screenshot shows a database interface with a toolbar at the top and a table below it. The toolbar has buttons for Data Output, Messages, Notifications, and Query, with Data Output being the active tab. Below the toolbar is a set of icons for file operations like new, open, save, and print. The table has two columns: 'tipo\_cuenta' and 'total\_cuentas'. The first row contains 'Ahorros' and '1'. The second row contains 'Corriente' and '2'.

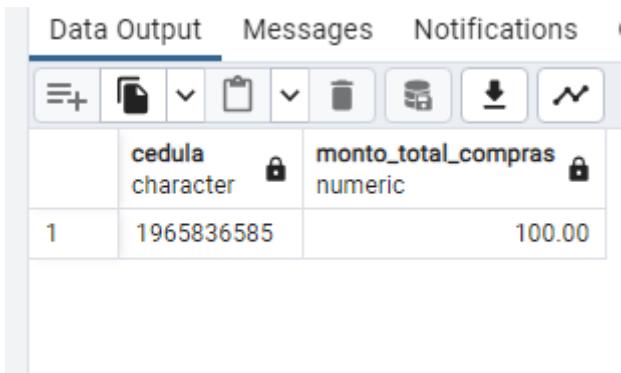
	tipo_cuenta character varying (20)	total_cuentas bigint
1	Ahorros	1
2	Corriente	2

## Relación de clientes con compras

**Función de agregación 1:** Obtener el monto total de compras realizadas por cada cliente.

### Instrucciones:

1. Escribir la sentencia SQL utilizando la función de agregación SUM para sumar los montos de las compras.
2. Utilizar la cláusula GROUP BY para agrupar las compras por cliente.
3. Ejecutar la consulta en la base de datos.



The screenshot shows a software interface for managing database queries. At the top, there are tabs for 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with several icons. The main area displays a table with two columns: 'cedula' (character) and 'monto\_total\_compras' (numeric). A single row of data is shown, with the value '1965836585' in the first column and '100.00' in the second column. The table has a header row and a data row.

	cedula character	monto_total_compras numeric
1	1965836585	100.00

**Función de agregación 2:** Obtener la cantidad total de compras realizadas en una fecha específica.

### Requisitos:

- Conocer la fecha específica de las compras.

### Instrucciones:

1. Escribir la sentencia SQL utilizando la función de agregación COUNT para contar el número de compras.
2. Utilizar la cláusula WHERE para filtrar las compras por la fecha específica.
3. Ejecutar la consulta en la base de datos.

The screenshot shows a database interface with a toolbar at the top labeled "Data Output", "Messages", and "Notifications". Below the toolbar is a row of icons for file operations like new, open, save, and delete. The main area displays a table with two columns: "fecha\_compra" (date) and "total\_compras" (bigint). There is one row of data with values "2023-10-11" and "1".

	fecha_compra	total_compras
1	2023-10-11	1

### Relación entre estudiantes y profesores

**Función de agregación 1:** Obtener la cantidad total de estudiantes asignados a cada profesor.

#### Instrucciones:

1. Escribir la sentencia SQL utilizando la función de agregación COUNT para contar el número de estudiantes.
2. Utilizar la cláusula GROUP BY para agrupar los estudiantes por profesor.
3. Ejecutar la consulta en la base de datos.

The screenshot shows a database interface with a toolbar at the top labeled "Data Output", "Messages", and "Notifications". Below the toolbar is a row of icons for file operations like new, open, save, and delete. The main area displays a table with two columns: "codigo\_profesor" (integer) and "total\_estudiantes" (bigint). There are three rows of data with values (1, 3), (2, 6), and (3, 7).

	codigo_profesor	total_estudiantes
1	3	3
2	2	6
3	1	7

**Función de agregación 2:** Obtener la edad promedio de los estudiantes.

**Instrucciones:**

1. Escribir la sentencia SQL utilizando la función ROUND para mostrar la edad\_promedio como entero, dentro de ROUND la función de agregación AVG (EXTRACT(YEAR FROM CURRENT\_DATE) Extract para obtener la diferencia en años entre la fecha actual y la fecha de nacimiento de cada estudiante seguido de un guion hacer otro EXTRACT que lleva dentro EL AÑO (YEAR) de la fecha de nacimiento de los estudiantes, con ello obtendremos el promedio de las edades de los estudiantes.
2. Ejecutar la consulta en la base de datos.

Data Output		Messages				
edad_promedio numeric						

1	19
---	----

**Relación entre persona y préstamo**

**Función de agregación 1:** Obtener la suma total de los montos de préstamos para cada persona.

**Instrucciones:**

1. Escribir la sentencia SQL utilizando la función de agregación SUM para sumar los montos de los préstamos.
2. Utilizar la cláusula GROUP BY para agrupar los préstamos por persona.
3. Ejecutar la consulta en la base de datos.

Data Output    Messages    Notifications    Query

	cedula [PK] character						
1	monto_total_prestamos money						
\$500,00							

**Función de agregación 2:** Obtener la cantidad total de personas que tienen más de un hijo.

**Instrucciones:**

1. Escribir la sentencia SQL utilizando la función de agregación COUNT para contar el número de personas.
2. Utilizar la cláusula WHERE para filtrar las personas que tienen más de un hijo.
3. Ejecutar la consulta en la base de datos.

Data Output    Messages    Nc

	total_personas bigint				
1	5				

**Relación entre Productos y Ventas**

**Función de agregación 1:** Obtener el máximo precio de todos los productos.

**Instrucciones:**

1. Escribir la sentencia SQL utilizando la función de agregación MAX para obtener el máximo precio de los productos.
2. Ejecutar la consulta en la base de datos.

	precio_maximo
1	\$5,50

**Función de agregación 2:** Obtener la suma total de la cantidad de productos vendidos.

#### Instrucciones:

1. Escribir la sentencia SQL utilizando la función de agregación SUM para sumar la cantidad de productos vendidos.
2. Ejecutar la consulta en la base de datos.

	cantidad_total_vendida
1	55

#### Relación entre Transacciones y Banco

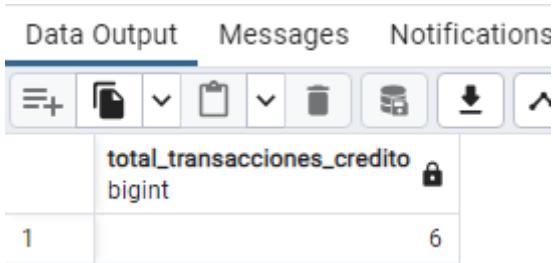
**Función de agregación 1:** Obtener la cantidad total de transacciones de tipo 'C' (crédito).

**Instrucciones:**

1. Escribir la sentencia SQL utilizando la función de agregación COUNT para contar el número de transacciones.
2. Utilizar la cláusula WHERE para filtrar las transacciones de tipo 'C' (crédito).
3. Ejecutar la consulta en la base de datos.

Data Output    Messages    Notifications

---



The screenshot shows a software interface for viewing database query results. At the top, there are three tabs: "Data Output" (underlined), "Messages", and "Notifications". Below the tabs is a toolbar with several icons: a grid, a file, a dropdown, a clipboard, another dropdown, a trash can, a save icon, a download icon, and an up arrow. The main area displays a single row of data in a table format. The first column is labeled "total\_transacciones\_credito" and has a type of "bigint". The value in the cell is "6". There is also a small lock icon next to the column header.

	total_transacciones_credito
1	6

**Función de agregación 2:** Obtener el promedio de montos de transacciones para cada número de cuenta.

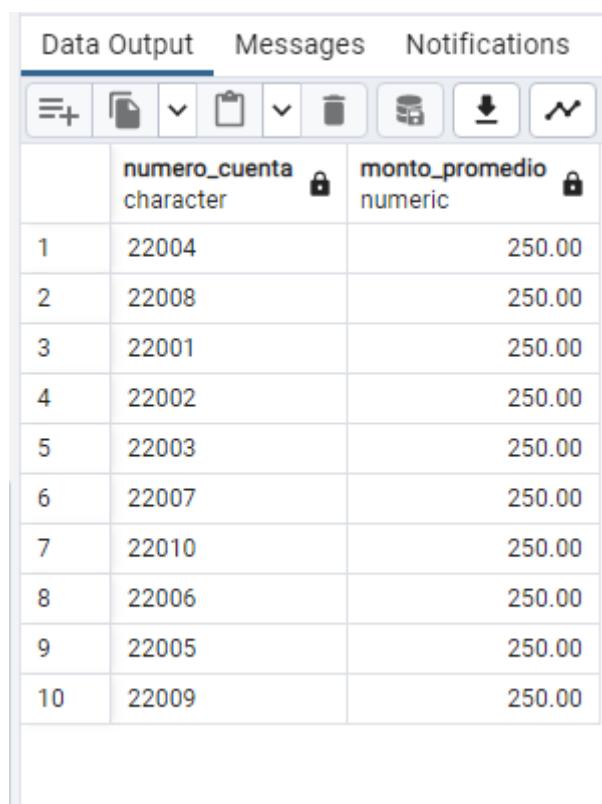
**Instrucciones:**

1. Escribir la sentencia SQL selecciona el número de cuenta seguido de una coma utilizar la función ROUND dentro utiliza la función de agregación AVG para calcular el promedio de los montos de las transacciones dentro de la función AVG usa la función CAST para el monto castéalo como decimal y fuera del paréntesis especificar el numero 2 para que nos resulten dos decimales y con ello obtendremos el monto\_promedio a través de las transacciones.

**Ejemplo:**

```
SELECT ***** ROUND(AVG(CAST(*****AS decimal)),2) AS *****
FROM ***** GROUP BY *****;
```

2. Utilizar la cláusula GROUP BY para agrupar las transacciones por número de cuenta.
3. Ejecutar la consulta en la base de datos.



	numero_cuenta character	monto_promedio numeric
1	22004	250.00
2	22008	250.00
3	22001	250.00
4	22002	250.00
5	22003	250.00
6	22007	250.00
7	22010	250.00
8	22006	250.00
9	22005	250.00
10	22009	250.00

## Relación entre Videojuegos y Plataformas

**Función de agregación 1:** Obtener la cantidad total de plataformas disponibles para cada videojuego.

### Instrucciones:

1. Escribir la sentencia SQL utilizando la función de agregación COUNT para contar el número de plataformas.
2. Utilizar la cláusula GROUP BY para agrupar las plataformas por videojuego usando el código del videojuego.
3. Ejecutar la consulta en la base de datos.

	codigo_videojuego integer	total_plataformas bigint
1	3	1
2	2	1
3	1	3

**Función de agregación 2:** Obtener la valoración promedio de todos los videojuegos.

### Instrucciones:

1. Escribir la sentencia SQL utilizando la función de agregación AVG para calcular el promedio de las valoraciones de los videojuegos, Utiliza la función ROUND antes de AVG para redondear a 2 decimales el resultado .

### Ejemplo:

```
SELECT ROUND(AVG(*****),2) AS *****  
FROM *****;
```

Data Output	Messages	Notifications
		
	valoracion_promedio	locked
1	8.63	

### Relación entre registros\_entrada y empleado

**Función de agregación 1:** Obtener la cantidad total de registros de entrada realizados por cada empleado.

#### Instrucciones:

1. Escribir la sentencia SQL utilizando la función de agregación COUNT para contar el número de registros de entrada.
2. Utilizar la cláusula GROUP BY para agrupar los registros de entrada por empleado.
3. Ejecutar la consulta en la base de datos.

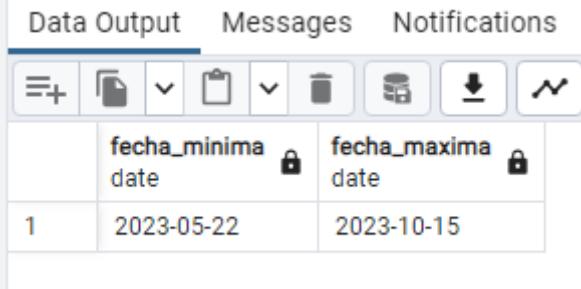
Data Output	Messages	Notifications	Query	Qu
				
	cedula_empleado	locked	total_registros_entrada	locked
	character varying (50)		bigint	
1	2209		1	
2	1578963587		6	
3	1789635021		1	
4	2206		1	
5	2210		1	
6	2204		1	
7	1236574896		1	
8	2202		1	
9	2205		1	
10	1576923224		1	
11	2207		1	
12	2203		1	
13	2201		1	
14	2208		1	

**Función de agregación 2:** Obtener la fecha mínima y máxima de los registros de entrada.

**Instrucciones:**

Escribir la sentencia SQL utilizando la función de agregación MIN para obtener la fecha mínima y la función de agregación MAX para obtener la fecha máxima de los registros de entrada.

Ejecutar la consulta en la base de datos.



	fecha_minima	fecha_maxima
1	2023-05-22	2023-10-15