

# USUARIO Y CUENTAS

Query

Query History

127

where (cu.saldo between '100' and '1000')

128

--SUBCONSULTA: DATOS DE CUENTAS Y USUARIOS DONDE LA FECHA DE CREACIÓN DE LA CUE

129

select cu.numero\_cuenta, cu.fecha\_creacion, cu.saldo, u.cedula from cuentas c

130

where (u.cedula = any(select cu.cedula\_propietario from cuentas where cu.fecl

131

132

--EJERCICIO DE FUNCIONES DE AGREGACIÓN

133

--PROMEDIO DE SALDO DE UN USUARIO EN ESPECÍFICO

134

select cast(AVG(cast(saldo as numeric))as money)saldo\_promedio from cuentas

135

where cedula\_propietario='1750056789'

136

137

--TOTAL DE CUENTAS SEGÚN SU TIPO

138

select tipo\_cuenta, count(tipo\_cuenta)total\_cuentas from cuentas, usuario

139

WHERE cedula\_propietario = cedula

140

group by tipo\_cuenta

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑

🗑

📥

⬇

📈

SQL

	saldo_promedio	
	money	🔒
1	592,23 €	

Query

Query History

127

where (cu.saldo between '100' and '1000')

128

--SUBCONSULTA: DATOS DE CUENTAS Y USUARIOS DONDE LA FECHA DE CREACIÓN DE LA CUE

129

select cu.numero\_cuenta, cu.fecha\_creacion, cu.saldo, u.cedula from cuentas cu,

130

where (u.cedula = any(select cu.cedula\_propietario from cuentas where cu.fecha

131

132

--EJERCICIO DE FUNCIONES DE AGREGACIÓN

133

--PROMEDIO DE SALDO DE UN USUARIO EN ESPECÍFICO

134

select cast(AVG(cast(saldo as numeric))as money)saldo\_promedio from cuentas

135

where cedula\_propietario='1750056789'

136

137

--TOTAL DE CUENTAS SEGÚN SU TIPO

138

select tipo\_cuenta, count(tipo\_cuenta)total\_cuentas from cuentas, usuario

139

WHERE cedula\_propietario = cedula

140

group by tipo\_cuenta

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑

🗑

📥

⬇

📈

SQL

	tipo_cuenta	total_cuentas
	character varying (20)	bigint
1	Ahorros	4
2	Corriente	7

## CLIENTES Y COMPRAS

```
Query Query History
56 select * from clientes
57 where (nombre='Mónica')
58
59 --EJERCICIO DE FUNCIONES DE AGREGACIÓN
60 --SUMA DE COMPRAS POR CLIENTE
61 select cedula, sum(monto) monto_total_compras from compras
62 group by cedula
63
64 --CANTIDAD TOTAL DE COMPRAS REALIZADOS EN UNA FECHA ESPECÍFICA
65 select fecha_compra, count(monto) total_compras from compras
66 where fecha_compra='2024-09-24'
67 group by fecha_compra
```

Data Output Messages Notifications

	cedula character (10) 🔒	monto_total_compras numeric 🔒
1	5566778899	186.15
2	7788990011	451.05
3	6677889900	285.40
4	8899001122	135.00
5	4455667788	180.50

```
Query  Query History
56 select * from clientes
57 where (nombre='Mónica')
58
59 --EJERCICIO DE FUNCIONES DE AGREGACIÓN
60 --SUMA DE COMPRAS POR CLIENTE
61 select cedula, sum(monto) monto_total_compras from compras
62 group by cedula
63
64 --CANTIDAD TOTAL DE COMPRAS REALIZADOS EN UNA FECHA ESPECÍFICA
65 select fecha_compra, count(monto) total_compras from compras
66 where fecha_compra='2024-09-24'
67 group by fecha_compra
```

Data Output Messages Notifications

	fecha_compra date	total_compras bigint
1	2024-09-24	1

# ESTUDIANTES Y PROFESORES

Query

Query History

145

```
select profe.codigo, es.nombre, es.apellido from profesores profe, estudiantes
146 where (es.apellido like '%ñ%')
147 --SUBCONSULTA: ESTUDIANTES CUYO CÓDIGO DE PROFESOR CORRESPONDE AL NOMBRE DE "F
148 select * from estudiantes es
149 where (es.codigo_profesor = any(select profe.codigo from profesores profe wher
150
151 --EJERCICIO DE FUNCIONES DE AGREGACIÓN
152 --CANTIDAD TOTAL DE ESTUDIANTES ASIGNADOS A CADA PROFESOR
153 select codigo_profesor, count(estudiantes)total_estudiantes from estudiantes
154 group by codigo_profesor
155 ---EDAD PROMEDIO DE LOS ESTUDIANTES
156 select cast(AVG(EXTRACT(YEAR FROM CURRENT_DATE)-EXTRACT(YEAR FROM fecha_de_nac
```

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑

🗑

📥

📥

📈

SQL

	codigo_profesor integer	totalEstudiantes bigint
1	1	2
2	3	2
3	5	2
4	4	2
5	2	2

Query

Query History

145

```
codigo, es.nombre, es.apellido from profesores profe, estudiantes es
146 llido like '%ñ%')
147 : ESTUDIANTES CUYO CÓDIGO DE PROFESOR CORRESPONDE AL NOMBRE DE "FRANCISCO"
148 estudiantes es
149 igo_profesor = any(select profe.codigo from profesores profe where profe.nombre='Francisco'))
150
151 E FUNCIONES DE AGREGACIÓN
152 TAL DE ESTUDIANTES ASIGNADOS A CADA PROFESOR
153 o_profesor, count(estudiantes)total_estudiantes from estudiantes
154 go_profesor
155 DIO DE LOS ESTUDIANTES
156 VG(EXTRACT(YEAR FROM CURRENT_DATE)-EXTRACT(YEAR FROM fecha_de_nacimiento))as int)edad_promedio from estudiantes
```

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑

🗑

📥

📥

📈



SQL

	edad_promedio integer
1	24

## PERSONAS Y PRÉSTAMO

```
Query Query History
54 where (pres.monto between '100' and '1000')
55 --SUBCONSULTA: DATOS DE LA PERSONA CUYA CEDULA SEA IGUAL A LA CEDULA DE NOMBRE "SEAN"
56 select * from persona
57 where nombre='Sean'
58
59 --EJERCICIO DE FUNCIONES DE AGREGACIÓN
60 --SUMA TOTAL DE LOS MONTOS DE PRÉSTAMOS PARA CADA PERSONA
61 select per.cedula, sum(monto) from persona per, prestamo pre
62 where per.cedula = pre.cedula
63 group by per.cedula
64 --CANTIDAD TOTAL DE PERSONAS QUE TIENEN MÁS DE UN HIJO
65 select count(*)total_personas from persona
66 where numero_hijos>1
```

Data Output Messages Notifications

	cedula [PK] character (10) 	sum money 
1	5678901234	150,00 €
2	1234567890	15.000,00 €
3	4567890123	5.000,00 €
4	3456789012	12.000,00 €
5	2345678901	8.000,00 €

```
Query | Query History
54 where (pres.monto between '100' and '1000')
55 --SUBCONSULTA: DATOS DE LA PERSONA CUYA CEDULA SEA IGUAL A LA CEDULA DE NOMBRE "SEAN"
56 select * from persona
57 where nombre='Sean'
58
59 --EJERCICIO DE FUNCIONES DE AGREGACIÓN
60 --SUMA TOTAL DE LOS MONTO DE PRÉSTAMOS PARA CADA PERSONA
61 select per.cedula, sum(monto) from persona per, prestamo pre
62 where per.cedula = pre.cedula
63 group by per.cedula
64 --CANTIDAD TOTAL DE PERSONAS QUE TIENEN MÁS DE UN HIJO
65 select count(*)total_personas from persona
66 where numero_hijos>1
```

Data Output Messages Notifications

	total_personas bigint
1	3

# PRODUCTOS Y VENTAS

Query

Query History

128

```
select pro.nombre, pro.stock, ven.cantidad from productos pro, ventas ven
129 where (pro.nombre like '%m%' and pro.descripcion is null)
130 --SUBCONSULTA: OBTENER EL NOMBRE Y STOCK DE LOS PRODUCTOS DONDE EL CÓDIGO DE LO
131 select pro.nombre, pro.stock from productos pro
132 where (pro.codigo = any(select codigo_producto from ventas where cantidad=5))
133
134 --EJERCICIO DE FUNCIONES DE AGREGACIÓN
135 --MÁXIMO PRECIO DE TODOS LOS PRODUCTOS
136 select max(precio)precio_maximo from productos
137 --SUMA TOTAL DE LA CANTIDAD DE PRODUCTOS VENDIDOS
138 select sum(cantidad) from ventas
```

Data Output

Messages

Notifications

≡+

	precio_maximo money	
1	1.200,00 €	

Query

Query History

128

```
select pro.nombre, pro.stock, ven.cantidad from productos pro
129 where (pro.nombre like '%m%' and pro.descripcion is null)
130 --SUBCONSULTA: OBTENER EL NOMBRE Y STOCK DE LOS PRODUCTOS DON
131 select pro.nombre, pro.stock from productos pro
132 where (pro.codigo = any(select codigo_producto from ventas wh
133
134 --EJERCICIO DE FUNCIONES DE AGREGACIÓN
135 --MÁXIMO PRECIO DE TODOS LOS PRODUCTOS
136 select max(precio)precio_maximo from productos
137 --SUMA TOTAL DE LA CANTIDAD DE PRODUCTOS VENDIDOS
138 select sum(cantidad) from ventas
```

Data Output

Messages

Notifications

≡+

	sum bigint	
1	24	

# TRANSACCIONES Y BANCO

Query

Query History

122

--SUBCONSULTA: OBTENER TODOS LOS TRANSACCIONES CUYO CÓDIGO CORRESPONDE

123

select \* from transacciones

124

where codigo = any(select codigo\_banco from banco where codigo\_banco=1

125

126

--EJERCICIO DE FUNCIONES DE AGREGACIÓN

127

--CANTIDAD TOTAL DE TRANSACCIONES DE TIPO 'C' (CRÉDITO)

128

select count(tipo)total\_transacciones\_credito from transacciones

129

where tipo='C'

130

--PROMEDIO DE MONTOS DE TRANSACCIONES PARA CADA NÚMERO DE CUENTA

131

select numero\_cuenta, cast(AVG(cast(monto as numeric))as money)monto\_p

132

group by numero\_cuenta

Data Output

Messages

Notifications

≡

+

📄

▼

📋

▼

🗑

🗑

📦

⬇

📈

SQL

	total_transacciones_credito	
	bigint	
1	10	

Query

Query History

125

126

--EJERCICIO DE FUNCIONES DE AGREGACIÓN

127

--CANTIDAD TOTAL DE TRANSACCIONES DE TIPO 'C' (CRÉDITO)

128

select count(tipo)total\_transacciones\_credito from transacciones

129

where tipo='C'

130

--PROMEDIO DE MONTOS DE TRANSACCIONES PARA CADA NÚMERO DE CUENTA

131

select numero\_cuenta, cast(AVG(cast(monto as numeric))as money)monto\_promedio from transacciones

132

group by numero\_cuenta

Data Output

Messages

Notifications

≡

+

📄

▼

📋

▼

🗑

🗑

📦

⬇

📈

SQL

	numero_cuenta	monto_promedio
	character (5)	money
1	85213	2.000,00 €
2	10005	500,00 €
3	32102	500,65 €
4	22004	150,00 €
5	75369	200,00 €
6	10007	220,10 €
7	02147	300,00 €
8	10009	450,00 €
9	74125	900,00 €
10	10004	120,75 €
11	85214	600,00 €
12	22002	1.600,00 €

Total rows: 21 of 21    Query complete 00:00:00.094    Ln 132, Col 23

## VIDEOJUEGOS Y PLATAFORMAS

Query	Query History
93	<code>or ((nombre like 'C%') and (valoracion&gt;8))</code>
94	<code>or (nombre like 'D%')</code>
95	<code>--SUBCONSULTA: TODOS LOS DATOS DE PLATAFORMAS DONDE PERTENEZCA EL CÓDIGO</code>
96	<code>select * from plataformas</code>
97	<code>where codigo_videojuego = any(select vj.codigo from videojuegos vj where</code>
98	
99	<code>--EJERCICIO DE FUNCIONES DE AGREGACIÓN</code>
100	<code>--CANTIDAD TOTAL DE PLATAFORMAS DISPONIBLES PARA CADA VIDEOJUEGO</code>
101	<code>select codigo_videojuego, count(*)total_plataformas from plataformas</code>
102	<code>group by codigo_videojuego</code>
103	<code>--PROMEDIO DE LA VALORACIÓN DE TODOS LOS VIDEOJUEGOS</code>
104	<code>select ROUND(AVG(valoracion),2) from videojuegos</code>

Data Output	Messages	Notifications
	<b>codigo_videojuego</b> integer	<b>total_plataformas</b> bigint
1	3	1
2	5	1
3	4	1
4	2	1
5	1	1

Query	Query History
93	<code>or ((nombre like 'C%') and (valoracion&gt;8))</code>
94	<code>or (nombre like 'D%')</code>
95	<code>--SUBCONSULTA: TODOS LOS DATOS DE PLATAFORMAS DONDE PERTENEZCA EL CÓDIGO</code>
96	<code>select * from plataformas</code>
97	<code>where codigo_videojuego = any(select vj.codigo from videojuegos vj where</code>
98	
99	<code>--EJERCICIO DE FUNCIONES DE AGREGACIÓN</code>
100	<code>--CANTIDAD TOTAL DE PLATAFORMAS DISPONIBLES PARA CADA VIDEOJUEGO</code>
101	<code>select codigo_videojuego, count(*)total_plataformas from plataformas</code>
102	<code>group by codigo_videojuego</code>
103	<code>--PROMEDIO DE LA VALORACIÓN DE TODOS LOS VIDEOJUEGOS</code>
104	<code>select ROUND(AVG(valoracion),2) from videojuegos</code>

Data Output	Messages	Notifications
	<b>round</b> numeric	
1	8.20	



## REGISTRO DE ENTRADAS Y EMPLEADOS

```
Query History
142 --DE ENTRADA DE CÉDULA '2201'
143 select e.codigo_empleado, e.nombre from empleado e
144 where (e.codigo_empleado = any(select re.codigo_registro from registros_entrada re
145
146 --EJERCICIO DE FUNCIONES DE AGREGACIÓN
147 --CANTIDAD TOTAL DE REGISTROS DE ENTRADA REALIZADOS POR CADA EMPLEADO
148 select cedula_empleado, count(*) from registros_entrada
149 group by cedula_empleado
150 --OBTENER LA FECHA MÍNIMA Y MÁXIMA DE REGISTROS DE ENTRADA
151 select min(fecha)fecha_minima, max(fecha)fecha_maxima from registros_entrada
```

Data Output Messages Notifications

	cedula_empleado character (10) 🔒	count bigint 🔒
1	5678901234	4
2	3456789012	2
3	1234567890	2
4	4567890123	2
5	2345678901	2

```
Query History
142 --DE ENTRADA DE CÉDULA '2201'
143 select e.codigo_empleado, e.nombre from empleado e
144 where (e.codigo_empleado = any(select re.codigo_registro from registros_entrada re
145
146 --EJERCICIO DE FUNCIONES DE AGREGACIÓN
147 --CANTIDAD TOTAL DE REGISTROS DE ENTRADA REALIZADOS POR CADA EMPLEADO
148 select cedula_empleado, count(*) from registros_entrada
149 group by cedula_empleado
150 --OBTENER LA FECHA MÍNIMA Y MÁXIMA DE REGISTROS DE ENTRADA
151 select min(fecha)fecha_minima, max(fecha)fecha_maxima from registros_entrada
```

Data Output Messages Notifications

	fecha_minima date	fecha_maxima date
1	2024-09-01	2024-09-10