



## ESCUELA POLITÉCNICA NACIONAL

FACULTAD: ESFOT

---

**PERÍODO ACADÉMICO:** 2024 A

**ASIGNATURA:** BASES DE DATOS

**DOCENTE:** MSc. Yadira Franco

**PARALELO:** GR1

---

**NOMBRE DEL ESTUDIANTE (S):** Andres Tufiño

**NOTA:**

**TEMA:** CONSULTAS BÁSICAS

**TIEMPO:**

---

### INSTRUCCIONES:

### PROBLEMA A ESTUDIAR Y RESOLVER

Estamos diseñando una base de datos para un banco “Los afortunados”. Este banco necesita una base de datos para gestionar a sus clientes, las cuentas bancarias de esos clientes y las transacciones que se realizan en esas cuentas.

1. Crear y relacionar tres tablas: Clientes, Cuentas y Transacciones.

### Requisitos TABLA:

#### Cliente

id\_cliente, nombre, email

#### Cuentas

id\_cuenta, id\_cliente, saldo

#### Transacciones

id\_transaccion, id\_cuenta, monto, fecha

### Relaciones:

1. Cada cliente puede tener múltiples cuentas, pero cada cuenta pertenece a un único cliente.
2. Cada cuenta puede tener múltiples transacciones, pero cada transacción está asociada a una sola cuenta.

```
1 • create database Banco;
2 • use Banco;
3 • create table Cliente (
4     id_cliente int auto_increment primary key,
5     nombre varchar(100) not null,
6     email varchar(100) not null
7 );
8 • create table Cuentas (
9     id_cuenta int auto_increment primary key,
10    id_cliente int,
11    saldo decimal(10, 2) not null,
12    foreign key (id_cliente) references Cliente(id_cliente)
13 );
14 • create table Transacciones (
15     id_transaccion int auto_increment primary key,
16     id_cuenta int,
17     monto decimal(10, 2) not null,
18     fecha date not null,
19     foreign key(id_cuenta) references Cuentas(id_cuenta)
20 );
21
22 • insert into Cliente (nombre, email) values
23     ('Juan Perez', 'juan.perez@example.com'),
24     ('Maria Lopez', 'maria.lopez@example.com'),
25     ('Carlos Gomez', 'carlos.gomez@example.com'),
26     ('Ana Torres', 'ana.torres@example.com'),
27     ('Luis Morales', 'luis.morales@example.com'),
28     ('Marta Sanchez', 'marta.sanchez@example.com'),
29     ('Pedro Ramirez', 'pedro.ramirez@example.com'),
```

```
22 • insert into Cliente (nombre, email) values
23 ('Juan Perez', 'juan.perez@example.com'),
24 ('Maria Lopez', 'maria.lopez@example.com'),
25 ('Carlos Gomez', 'carlos.gomez@example.com'),
26 ('Ana Torres', 'ana.torres@example.com'),
27 ('Luis Morales', 'luis.morales@example.com'),
28 ('Marta Sanchez', 'marta.sanchez@example.com'),
29 ('Pedro Ramirez', 'pedro.ramirez@example.com'),
30 ('Lucia Fernandez', 'lucia.fernandez@example.com'),
31 ('Miguel Diaz', 'miguel.diaz@example.com'),
32 ('Laura Gutierrez', 'laura.gutierrez@example.com');
33
34 • insert into Cuentas (id_cliente, saldo) values
35 (1, 5000.00),
36 (1, 1500.00),
37 (2, 3000.00),
38 (3, 7000.00),
39 (4, 1000.00),
40 (5, 2500.00),
41 (6, 4000.00),
42 (7, 3500.00),
43 (8, 2000.00),
44 (9, 4500.00),
45 (10, 6000.00);
--
47 • insert into Transacciones (id_cuenta, monto, fecha) values
48 (1, 200.00, '2024-01-15'),
49 (1, -100.00, '2024-02-10'),
50 (2, 150.00, '2024-03-05'),
51 (3, -200.00, '2024-04-20'),
52 (4, 300.00, '2024-05-25'),
53 (5, -150.00, '2024-06-30'),
54 (6, 400.00, '2024-07-15'),
55 (7, -250.00, '2024-08-10'),
56 (8, 350.00, '2024-09-05'),
57 (9, -300.00, '2024-10-20'),
58 (10, 450.00, '2024-11-25');
59
```

---

### 3. Insertar (20 registro)

## Ejercicios Propuestos

### Selección y Filtración Básica

CREAR EN GRUPO DE DOS ESTUDIANTES

#### Ejercicio 1: Selección y Filtración Básica

- Selecciona el `id_cuenta` y el `saldo` de todas las cuentas.

```
60 -- 1
61 • SELECT id_cuenta, saldo FROM Cuentas;
62
```

	id_cuenta	saldo
▶	1	5000.00
	2	1500.00
	3	3000.00
	4	7000.00
	5	1000.00
	6	2500.00
	7	4000.00
	8	3500.00
	9	2000.00
	10	4500.00
	11	6000.00
*	NULL	NULL

- Filtra las cuentas que tienen un saldo mayor a 4000.

```
63 -- b
64 • SELECT id_cuenta, saldo FROM Cuentas WHERE saldo > 4000;
65
```

	id_cuenta	saldo
▶	1	5000.00
	4	7000.00
	10	4500.00
	11	6000.00
*	NULL	NULL

- Seleccionar las cuentas con saldo menor a 2000

```

65      -- c
66      SELECT id_cuenta, saldo FROM Cuentas WHERE saldo < 2000;
67

```

Result Grid

	id_cuenta	saldo
▶	2	1500.00
	5	1000.00
*	NULL	NULL

- Mostrar los nombres de los clientes y la longitud de sus nombres

```

67      -- d
68      SELECT nombre, CHAR_LENGTH(nombre) AS longitud_nombre FROM Cliente;
69      -- e

```

Result Grid

	nombre	longitud_nombre
▶	Juan Perez	10
	Maria Lopez	11
	Carlos Gomez	12
	Ana Torres	10
	Luis Morales	12
	Marta Sanchez	13
	Pedro Ramirez	13
	Lucia Fernandez	15
	Miguel Diaz	11
	Laura Gutierrez	15

- Seleccionar las cuentas con saldo entre 1000 y 3000

```

69      -- e
70      SELECT id_cuenta, saldo FROM Cuentas WHERE saldo BETWEEN 1000 AND 3000;
71

```

Result Grid


	id_cuenta	saldo
▶	2	1500.00
	3	3000.00
	5	1000.00
	6	2500.00
	9	2000.00
*	NULL	NULL

- Obtener una lista de todos los clientes y sus cuentas, mostrando los valores NULL

```

/1      -- ↑
72 ●    SELECT Cliente.id_cliente, Cliente.nombre, Cuentas.id_cuenta
73      FROM Cliente
74      LEFT JOIN Cuentas ON Cliente.id_cliente = Cuentas.id_cliente;
75

```



Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Conte			
	id_cliente	nombre	id_cuenta
▶	1	Juan Perez	1
	1	Juan Perez	2
	2	Maria Lopez	3
	3	Carlos Gomez	4
	4	Ana Torres	5
	5	Luis Morales	6
	6	Marta Sanchez	7
	7	Pedro Ramirez	8
	8	Lucia Fernandez	9
	9	Miguel Diaz	10
	10	Laura Gutierrez	11

- Concatenar el nombre del cliente y su email con un separador

```

75      -- g
76 ●    SELECT CONCAT(nombre, ' - ', email) AS cliente_email FROM Cliente;
77

```

Result Grid	
Filter Rows: <input type="text"/>	
Export:  Wrap Cell Content: 	
	cliente_email
▶	Juan Perez - juan.perez@example.com
	Maria Lopez - maria.lopez@example.com
	Carlos Gomez - carlos.gomez@example.com
	Ana Torres - ana.torres@example.com
	Luis Morales - luis.morales@example.com
	Marta Sanchez - marta.sanchez@example.com
	Pedro Ramirez - pedro.ramirez@example.com
	Lucia Fernandez - lucia.fernandez@example.com
	Miguel Diaz - miguel.diaz@example.com
	Laura Gutierrez - laura.gutierrez@example.com

- Extraer los últimos 3 caracteres del email de cada cliente

```

78      -- h
79
80  ●   SELECT email, RIGHT(email, 3) AS ultimos_tres_caracteres FROM Cliente;
81

```

email	ultimos_tres_caracteres
juan.perez@example.com	com
maria.lopez@example.com	com
carlos.gomez@example.com	com
ana.torres@example.com	com
luis.morales@example.com	com
marta.sanchez@example.com	com
pedro.ramirez@example.com	com
lucia.fernandez@example.com	com
miguel.diaz@example.com	com
laura.gutierrez@example.com	com

- Convertir el nombre del cliente a minúsculas

```

82      -- i
83  ●   SELECT LOWER(nombre) AS nombre_minusculas FROM Cliente;
84

```

nombre_minusculas
juan perez
maria lopez
carlos gomez
ana torres
luis morales
marta sanchez
pedro ramirez
lucia fernandez
miguel diaz
laura gutierrez

- Convertir el nombre del cliente a mayúsculas

```

84      -- j
85      SELECT UPPER(nombre) AS nombre_mayusculas FROM Cliente;
86

```

Result Grid | Filter Rows: | Export: | Wrap C

	nombre_mayusculas
▶	JUAN PEREZ
	MARIA LOPEZ
	CARLOS GOMEZ
	ANA TORRES
	LUIS MORALES
	MARTA SANCHEZ
	PEDRO RAMIREZ
	LUCIA FERNANDEZ
	MIGUEL DIAZ
	LAURA GUTIERREZ

- Seleccionar las transacciones ordenadas por monto de mayor a menor, mostrando solo las primeras 10

```

86      -- k
87      SELECT * FROM Transacciones ORDER BY monto DESC LIMIT 10;
88

```

Result Grid | Filter Rows: | Edit: | Ex

	id_transaccion	id_cuenta	monto	fecha
▶	11	10	450.00	2024-11-25
	7	6	400.00	2024-07-15
	9	8	350.00	2024-09-05
	5	4	300.00	2024-05-25
	1	1	200.00	2024-01-15
	3	2	150.00	2024-03-05
	2	1	-100.00	2024-02-10
	6	5	-150.00	2024-06-30
	4	3	-200.00	2024-04-20
	8	7	-250.00	2024-08-10
*	NULL	NULL	NULL	NULL

- Calcular el saldo de cada cuenta después de aplicar un interés del 5%



```

89      -- 1
90
91 •    SELECT id_cuenta, saldo, saldo * 1.05 AS saldo_con_interes FROM Cuentas;
92

```

	id_cuenta	saldo	saldo_con_interes
▶	1	5000.00	5250.0000
	2	1500.00	1575.0000
	3	3000.00	3150.0000
	4	7000.00	7350.0000
	5	1000.00	1050.0000
	6	2500.00	2625.0000
	7	4000.00	4200.0000
	8	3500.00	3675.0000
	9	2000.00	2100.0000
	10	4500.00	4725.0000
	11	6000.00	6300.0000

- Reemplazar el dominio del email de los clientes: `email, '@example.com', '@newdomain.com'`

```

95 •    UPDATE Cliente
96      SET email = REPLACE(email, '@example.com', '@newdomain.com');
97 •    select*from Cliente;
98

```

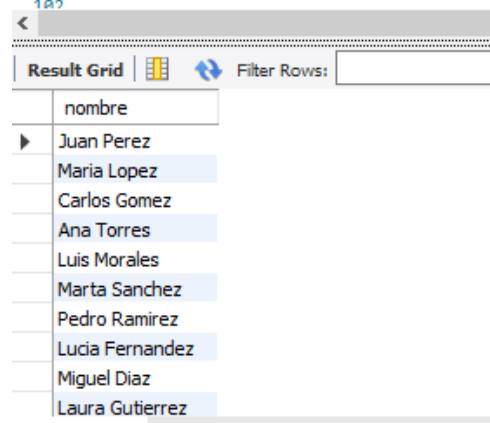
	id_cliente	nombre	email
▶	1	Juan Perez	juan.perez@newdomain.com
	2	Maria Lopez	maria.lopez@newdomain.com
	3	Carlos Gomez	carlos.gomez@newdomain.com
	4	Ana Torres	ana.torres@newdomain.com
	5	Luis Morales	luis.morales@newdomain.com
	6	Marta Sanchez	marta.sanchez@newdomain.com
	7	Pedro Ramirez	pedro.ramirez@newdomain.com
	8	Lucia Fernandez	lucia.fernandez@newdomain.com
	9	Miguel Diaz	miguel.diaz@newdomain.com

Cliente 13 x

## Ejercicio 2: Uso de Funciones de Cadena

- Selecciona el nombre de cada cliente.

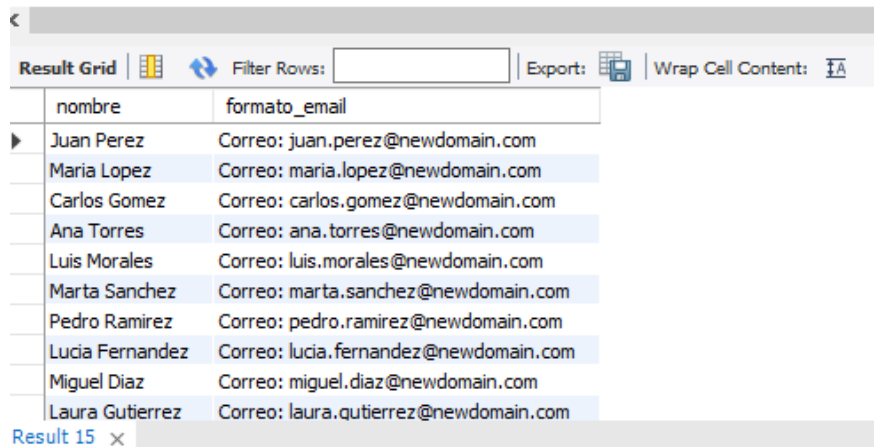
```
98
99      -- 2
100     -- a
101 •   SELECT nombre FROM Cliente;
102
```



The screenshot shows a SQL query result grid. The query is `SELECT nombre FROM Cliente;`. The result grid has a single column named 'nombre' and lists the following names: Juan Perez, Maria Lopez, Carlos Gomez, Ana Torres, Luis Morales, Marta Sanchez, Pedro Ramirez, Lucia Fernandez, Miguel Diaz, and Laura Gutierrez.

- Añade una columna que muestre el email del cliente en el formato Correo: [email].

```
102      -- b
103 •   SELECT nombre, CONCAT('Correo: ', email) AS formato_email FROM Cliente;
104
```



The screenshot shows a SQL query result grid. The query is `SELECT nombre, CONCAT('Correo: ', email) AS formato_email FROM Cliente;`. The result grid has two columns: 'nombre' and 'formato\_email'. The data is as follows:

nombre	formato_email
Juan Perez	Correo: juan.perez@newdomain.com
Maria Lopez	Correo: maria.lopez@newdomain.com
Carlos Gomez	Correo: carlos.gomez@newdomain.com
Ana Torres	Correo: ana.torres@newdomain.com
Luis Morales	Correo: luis.morales@newdomain.com
Marta Sanchez	Correo: marta.sanchez@newdomain.com
Pedro Ramirez	Correo: pedro.ramirez@newdomain.com
Lucia Fernandez	Correo: lucia.fernandez@newdomain.com
Miguel Diaz	Correo: miguel.diaz@newdomain.com
Laura Gutierrez	Correo: laura.gutierrez@newdomain.com

- Mostrar los emails de los clientes en minúsculas

```

104      -- c
105      SELECT LOWER(email) AS email_minusculas FROM Cliente;
106
107

```

email_minusculas
juan.perez@newdomain.com
maria.lopez@newdomain.com
carlos.gomez@newdomain.com
ana.torres@newdomain.com
luis.morales@newdomain.com
marta.sanchez@newdomain.com
pedro.ramirez@newdomain.com
lucia.fernandez@newdomain.com
miguel.diaz@newdomain.com
laura.gutierrez@newdomain.com

- Mostrar el nombre del cliente en mayúsculas

```

107      -- d
108      SELECT UPPER(nombre) AS nombre_mayusculas FROM Cliente;
109

```

nombre_mayusculas
JUAN PEREZ
MARIA LOPEZ
CARLOS GOMEZ
ANA TORRES
LUIS MORALES
MARTA SANCHEZ
PEDRO RAMIREZ
LUCIA FERNANDEZ
MIGUEL DIAZ
LAURA GUTIERREZ

### Ejercicio 3: Agregación de Datos

- Selecciona el id\_cuenta.

```

111      -- a
112 •    SELECT id_cuenta FROM Cuentas;
113

```

id_cuenta
1
2
3
4
5
6
7

- Calcula el total de las transacciones realizadas en cada cuenta.

```

114      -- b
115 •    SELECT id_cuenta, SUM(monto) AS total_transacciones FROM Transacciones GROUP BY id_cuenta;
116

```

id_cuenta	total_transacciones
1	100.00
2	150.00
3	-200.00
4	300.00
5	-150.00
6	400.00
7	-250.00
8	350.00
9	-300.00

- Calcular el saldo total de todas las cuentas

```

117      -- c
118 •    SELECT SUM(saldo) AS saldo_total FROM Cuentas;
119

```

saldo_total
40000.00

- Calcular el monto total de todas las transacciones

```

120      -- d
121      SELECT SUM(monto) AS monto_total FROM Transacciones;
122

```

Result Grid | Filter Rows: | Export: | Wra

	monto_total
▶	850.00

- Calcular el monto promedio de las transacciones

```

123      -- e
124
125      SELECT AVG(monto) AS monto_promedio FROM Transacciones;
126

```

Result Grid | Filter Rows: | Export: | Wrap Ce

	monto_promedio
▶	77.272727

#### Ejercicio 4: Funciones de Fecha:

- Selecciona el id\_transaccion y la fecha de todas las transacciones.

```

127      -- 4
128      -- a
129      SELECT id_transaccion, fecha FROM Transacciones;
130

```

Result Grid | Filter Rows: | Edit: |

	id_transaccion	fecha
▶	1	2024-01-15
	2	2024-02-10
	3	2024-03-05
	4	2024-04-20
	5	2024-05-25
	6	2024-06-30
	7	2024-07-15
	8	2024-08-10
	9	2024-09-05
	10	2024-10-20

- Extraer el año de la fecha de cada transacción

```

130 -- b
131 • SELECT id_transaccion, YEAR(fecha) AS anio FROM Transacciones;
132

```

id_transaccion	anio
1	2024
2	2024
3	2024
4	2024
5	2024
6	2024
7	2024

- Extraer el mes de la fecha de cada transacción

```

133 -- c
134 • SELECT id_transaccion, MONTH(fecha) AS mes FROM Transacciones;
135

```

id_transaccion	mes
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8

- Extraer el día de la semana de la fecha de cada transacción

```

136 -- d
137 • SELECT id_transaccion, DAYOFWEEK(fecha) AS dia_semana FROM Transacciones;
138

```

id_transaccion	dia_semana
1	2
2	7
3	3
4	7
5	7
6	1
7	2
8	7

- Filtra las transacciones que se realizaron en el año 2024.

```

139      -- e
140      SELECT * FROM Transacciones WHERE YEAR(fecha) = 2024;
141

```

	id_transaccion	id_cuenta	monto	fecha
▶	1	1	200.00	2024-01-15
	2	1	-100.00	2024-02-10
	3	2	150.00	2024-03-05
	4	3	-200.00	2024-04-20
	5	4	300.00	2024-05-25
	6	5	-150.00	2024-06-30
	7	6	400.00	2024-07-15

- Seleccionar las transacciones realizadas en 2023.

```

142      -- f
143      SELECT * FROM Transacciones WHERE YEAR(fecha) = 2023;
144

```

	id_transaccion	id_cuenta	monto	fecha
*	NULL	NULL	NULL	NULL

- Seleccionar las transacciones realizadas en el mes de junio de cualquier año

```

145      -- g
146      SELECT * FROM Transacciones WHERE MONTH(fecha) = 6;
147

```

	id_transaccion	id_cuenta	monto	fecha
▶	6	5	-150.00	2024-06-30
*	NULL	NULL	NULL	NULL

- Seleccionar las transacciones realizadas el 15 de cada mes

```

148      -- h
149      SELECT * FROM Transacciones WHERE DAY(fecha) = 15;
150

```

	id_transaccion	id_cuenta	monto	fecha
▶	1	1	200.00	2024-01-15
	7	6	400.00	2024-07-15
*	NULL	NULL	NULL	NULL

## Ejercicio 5: Agrupación y Contar Resultados

**Objetivo:** Contar el número de transacciones por cuenta.

- Selecciona el id\_cuenta.

```
111 -- a
112 • SELECT id_cuenta FROM Cuentas;
113
```

id_cuenta
1
2
3
4
5
6
7

- Cuenta el número de transacciones realizadas en cada cuenta.

```
155 -- b
156 • SELECT id_cuenta, COUNT(*) AS numero_transacciones FROM Transacciones GROUP BY id_cuenta;
157
```

id_cuenta	numero_transacciones
1	2
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

- Calcular el saldo promedio de todas las cuentas.



```
158      -- c
159      SELECT AVG(saldo) AS saldo_promedio FROM Cuentas;
160
```

Result Grid | Filter Rows: | Export: | W

saldo_promedio
3636.363636

## Ejercicio 6: Filtración con Condiciones

- Seleccionar las transacciones con un monto negativo

```
161      -- 6
162      -- a
163      SELECT * FROM Transacciones WHERE monto < 0;
164
```

Result Grid | Filter Rows: | Edit: |

	id_transaccion	id_cuenta	monto	fecha
▶	2	1	-100.00	2024-02-10
	4	3	-200.00	2024-04-20
	6	5	-150.00	2024-06-30
	8	7	-250.00	2024-08-10
	10	9	-300.00	2024-10-20
*	NULL	NULL	NULL	NULL

- Seleccionar las transacciones con un monto positivo

```
165      -- b
166      SELECT * FROM Transacciones WHERE monto > 0;
```

Result Grid | Filter Rows: | Edit: |

	id_transaccion	id_cuenta	monto	fecha
▶	1	1	200.00	2024-01-15
	3	2	150.00	2024-03-05
	5	4	300.00	2024-05-25
	7	6	400.00	2024-07-15
	9	8	350.00	2024-09-05
	11	10	450.00	2024-11-25
*	NULL	NULL	NULL	NULL

## Ejercicio 7: Ordenamiento

Seleccionar todas las cuentas ordenadas por saldo de mayor a menor:

```
168 -- 7
169 -- a
170 • SELECT * FROM Cuentas ORDER BY saldo DESC;
171
```

	id_cuenta	id_cliente	saldo
▶	4	3	7000.00
	11	10	6000.00
	1	1	5000.00
	10	9	4500.00
	7	6	4000.00
	8	7	3500.00
	3	2	3000.00
	6	5	2500.00
	9	8	2000.00
	2	1	1500.00

- Seleccionar todos los clientes ordenados alfabéticamente por nombre

```
171
172 • SELECT * FROM Cliente ORDER BY nombre;
173
```

	id_cliente	nombre	email
▶	4	Ana Torres	ana.torres@newdomain.com
	3	Carlos Gomez	carlos.gomez@newdomain.com
	1	Juan Perez	juan.perez@newdomain.com
	10	Laura Gutierrez	laura.gutierrez@newdomain.com
	8	Lucia Fernandez	lucia.fernandez@newdomain.com
	5	Luis Morales	luis.morales@newdomain.com
	2	Maria Lopez	maria.lopez@newdomain.com
	6	Marta Sanchez	marta.sanchez@newdomain.com
	9	Miguel Diaz	miguel.diaz@newdomain.com
	7	Pedro Ramirez	pedro.ramirez@newdomain.com

- Seleccionar las transacciones realizadas en 2023

173

174 • `SELECT * FROM Transacciones WHERE YEAR(fecha) = 2023;`

175

176

<

Result Grid | Filter Rows:  Edit:

	id_transaccion	id_cuenta	monto	fecha
*	NULL	NULL	NULL	NULL

- Seleccionar las cuentas ordenadas por saldo de menor a mayor, mostrando solo las primeras 5

176 • `SELECT * FROM Cuentas ORDER BY saldo ASC LIMIT 5;`

177

<

Result Grid | Filter Rows:  Edit:

	id_cuenta	id_cliente	saldo
▶	5	4	1000.00
	2	1	1500.00
	9	8	2000.00
	6	5	2500.00
	3	2	3000.00
*	NULL	NULL	NULL

▪

## Ejercicio 8: Contar Resultados

- Contar el número total de clientes

180 -- a

181 • `SELECT COUNT(*) AS total_clientes FROM Cliente;`

182

<

Result Grid | Filter Rows:  Export: W

	total_clientes
▶	10

- Contar el número total de cuentas

```

183      -- b
184      SELECT COUNT(*) AS total_cuentas FROM Cuentas;
185

```

Result Grid | Filter Rows: | Export: |

total_cuentas
11

- Contar el número total de transacciones

```

186      -- c
187      SELECT COUNT(*) AS total_transacciones FROM Transacciones;
188

```

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

total_transacciones
11

- Redondear el saldo de cada cuenta a dos decimales

```

189      -- d
190      SELECT id_cuenta, ROUND(saldo, 2) AS saldo_redondeado FROM Cuentas;
191

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	id_cuenta	saldo_redondeado
▶	1	5000.00
	2	1500.00
	3	3000.00
	4	7000.00
	5	1000.00
	6	2500.00
	7	4000.00
	8	3500.00

## Ejercicio 9: Máximos y Mínimos

- Encontrar la cuenta con el saldo más alto

```

194      -- a
195      SELECT * FROM Cuentas ORDER BY saldo DESC LIMIT 1;
196

```

id_cuenta	id_cliente	saldo
4	3	7000.00
NULL	NULL	NULL

- Encontrar la cuenta con el saldo más bajo

```

197      -- b
198      SELECT * FROM Cuentas ORDER BY saldo ASC LIMIT 1;
199

```

id_cuenta	id_cliente	saldo
5	4	1000.00
NULL	NULL	NULL

- Encontrar la transacción con el monto más alto

```

200      -- c
201      SELECT * FROM Transacciones ORDER BY monto DESC LIMIT 1;
202

```

id_transaccion	id_cuenta	monto	fecha
11	10	450.00	2024-11-25
NULL	NULL	NULL	NULL

- Encontrar la transacción con el monto más bajo

```

203      -- d
204      SELECT * FROM Transacciones ORDER BY monto ASC LIMIT 1;
205

```

id_transaccion	id_cuenta	monto	fecha
10	9	-300.00	2024-10-20
NULL	NULL	NULL	NULL

## PRESENTACIÓN

- PRESENTAR EN PDF
- SUBIR A GIT HUB

<https://github.com/Andrespipe1/bbd.git>

- Y SUBIR EL SQL-QUERY

PRACTICAR POR FAVOR PARA LA SIGUIENTE CLASE UN TEST DE CONOCIMIENTO, TEORICO Y PRACTICO,