



ESCUELA POLITÉCNICA NACIONAL

FACULTAD: ESFOT

PERÍODO ACADÉMICO: 2024 A

ASIGNATURA: BASES DE DATOS

DOCENTE: MSc. Yadira Franco

PARALELO: GR1

NOMBRE DEL ESTUDIANTE (S): Richard Soria

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

```

1  -- Crear base de datos banco
2  • create database banco;
3
4  • use banco;
5
6  -- Crear tabla cliente
7  • create table cliente(
8      id_cliente int auto_increment primary key,
9      nombre_cliente varchar(25) not null,
10     email_cliente varchar(50) not null);
11
12  -- Crear tabla cuenta
13  • create table cuenta(
14      id_cuenta int auto_increment primary key,
15      saldo_cuenta decimal(10, 2) not null,
16      id_cliente int not null,
17      foreign key (id_cliente) references cliente(id_cliente));
18
19  -- Crear tabla transaccion
20  • create table transaccion(
21      id_transaccion int auto_increment primary key,
22      monto_transaccion decimal(10, 2) not null,
23      fecha_transaccion date not null,
24      id_cuenta int not null,
25      foreign key (id_cuenta) references cuenta(id_cuenta));
--

```

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.

3. Insertar (20 registro)

```
27 -- Insertar 20 registro en las tablas
28 • insert into cliente(nombre_cliente, email_cliente) values
29 ('Juan Perez', 'juan.perez@example.com'),
30 ('Maria Lopez', 'maria.lopez@example.com'),
31 ('Carlos Sanchez', 'carlos.sanchez@example.com'),
32 ('Ana Fernandez', 'ana.fernandez@example.com'),
33 ('Luis Gomez', 'luis.gomez@example.com'),
34 ('Sofia Ramirez', 'sofia.ramirez@example.com'),
35 ('Jorge Diaz', 'jorge.diaz@example.com'),
36 ('Marta Torres', 'marta.torres@example.com'),
37 ('Pedro Ruiz', 'pedro.ruiz@example.com'),
38 ('Lucia Flores', 'lucia.flores@example.com'),
39 ('Miguel Vega', 'miguel.vega@example.com'),
40 ('Laura Castillo', 'laura.castillo@example.com'),
41 ('Ricardo Jimenez', 'ricardo.jimenez@example.com'),
42 ('Carmen Morales', 'carmen.morales@example.com'),
43 ('Alberto Rios', 'alberto.rios@example.com'),
44 ('Gloria Castro', 'gloria.castro@example.com'),
45 ('Enrique Herrera', 'enrique.herrera@example.com'),
46 ('Patricia Guzman', 'patricia.guzman@example.com'),
47 ('Francisco Ortiz', 'francisco.ortiz@example.com'),
48 ('Veronica Silva', 'veronica.silva@example.com');
--

50 • insert into cuenta(saldo_cuenta, id_cliente) values
51 (1000.50, 1),
52 (2000.75, 2),
53 (1500.30, 3),
54 (2500.10, 4),
55 (3000.60, 5),
56 (3500.80, 6),
57 (4000.40, 7),
58 (4500.70, 8),
59 (5000.20, 9),
60 (5500.90, 10),
61 (6000.00, 11),
62 (6500.25, 12),
63 (7000.35, 13),
64 (7500.50, 14),
65 (8000.65, 15),
66 (8500.75, 16),
67 (9000.85, 17),
68 (9500.95, 18),
69 (10000.00, 19),
70 (10500.10, 20);
```

```

72 • insert into transaccion(monto_transaccion, fecha_transaccion, id_cuenta) values
73 (100.50, '2024-07-01', 1),
74 (200.75, '2024-07-03', 2),
75 (150.30, '2024-07-05', 3),
76 (250.10, '2024-07-07', 4),
77 (300.60, '2024-07-09', 5),
78 (350.80, '2024-07-11', 6),
79 (400.40, '2024-07-13', 7),
80 (450.70, '2024-07-15', 8),
81 (500.20, '2024-07-17', 9),
82 (550.90, '2024-07-19', 10),
83 (600.00, '2024-07-21', 11),
84 (650.25, '2024-07-23', 12),
85 (700.35, '2024-07-25', 13),
86 (750.50, '2024-07-27', 14),
87 (800.65, '2024-07-29', 15),
88 (850.75, '2024-07-31', 16),
89 (900.85, '2024-08-02', 17),
90 (950.95, '2024-08-04', 18),
91 (1000.00, '2024-08-06', 19),
92 (1050.10, '2024-08-08', 20);

```

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.

```

95 -- ■ Selecciona el id_cuenta y el saldo de todas las cuentas.
96 • select id_cuenta, saldo_cuenta from cuenta;

```

Result Grid		Filter Rows:	Edit:	Export/Import:
	id_cuenta	saldo_cuenta		
▶	1	1000.50		
	2	2000.75		
	3	1500.30		
	4	2500.10		
	5	3000.60		
	6	3500.80		
	7	4000.40		
	8	4500.70		
	9	5000.20		
	10	5500.90		
	11	6000.00		
	12	6500.25		
	13	7000.35		
	14	7500.50		
	15	8000.65		
	16	8500.75		
	17	9000.85		
	18	9500.95		
	19	10000.00		
	20	10500.10		
•	NULL	NULL		

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

```

98      --  Filtra las cuentas que tienen un saldo mayor a 4000.
99 •    select id_cuenta, saldo_cuenta from cuenta where saldo_cuenta > 4000;

```

	id_cuenta	saldo_cuenta
▶	7	4000.40
	8	4500.70
	9	5000.20
	10	5500.90
	11	6000.00
	12	6500.25
	13	7000.35
	14	7500.50
	15	8000.65
	16	8500.75
	17	9000.85
	18	9500.95
	19	10000.00
	20	10500.10
•	NULL	NULL

- Seleccionar las cuentas con saldo menor a 2000

```

101      --  Seleccionar las cuentas con saldo menor a 2000
102 •    select id_cuenta, saldo_cuenta from cuenta where saldo_cuenta < 2000;

```

	id_cuenta	saldo_cuenta
▶	1	1000.50
	3	1500.30

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

```

104 -- @ Mostrar los nombres de los clientes y la longitud de sus nombres
105 • select nombre_cliente, length(nombre_cliente) as longitud_nombre from cliente;

```

nombre_cliente	longitud_nombre
Juan Perez	10
Maria Lopez	11
Carlos Sanchez	14
Ana Fernandez	13
Luis Gomez	10
Sofia Ramirez	13
Jorge Diaz	10
Marta Torres	12
Pedro Ruiz	10
Lucia Flores	12
Miguel Vega	11
Laura Castillo	14
Ricardo Jimenez	15
Carmen Morales	14
Alberto Rios	12
Gloria Castro	13
Enrique Herrera	15
Patricia Guzman	15
Francisco Ortiz	15
Veronica Silva	14

- Seleccionar las cuentas con saldo entre 1000 y 3000

```

107 -- @ Seleccionar las cuentas con saldo entre 1000 y 3000
108 • select id_cuenta, saldo_cuenta from cuenta where saldo_cuenta between 1000 and 3000;

```

id_cuenta	saldo_cuenta
1	1000.50
2	2000.75
3	1500.30
4	2500.10
5	3000.60
6	3500.80
7	4000.40
8	4500.70
9	5000.20
10	5500.90
11	6000.00
12	6500.25
13	7000.35
14	7500.50
15	8000.65
16	8500.75
17	9000.85
18	9500.95
19	10000.00
20	10500.10

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

```

110 -- @ Obtener una lista de todos los clientes y sus cuentas, mostrando los valores NULL
111 • select cliente.id_cliente, cliente.nombre_cliente, cuenta.saldo_cuenta from cliente left join cuenta on cliente.id_cliente = cuenta.id_cliente;

```

id_cliente	nombre_cliente	saldo_cuenta
1	Juan Perez	1000.50
2	Maria Lopez	2000.75
3	Carlos Sanchez	1500.30
4	Ana Fernandez	2500.10
5	Luis Gomez	3000.60
6	Sofia Ramirez	3500.80
7	Jorge Diaz	4000.40
8	Marta Torres	4500.70
9	Pedro Ruiz	5000.20
10	Lucia Flores	5500.90
11	Miguel Vega	6000.00
12	Laura Castillo	6500.25
13	Ricardo Jimenez	7000.35
14	Carmen Morales	7500.50
15	Alberto Rios	8000.65
16	Gloria Castro	8500.75
17	Enrique Herrera	9000.85
18	Patricia Guzman	9500.95
19	Francisco Ortiz	10000.00
20	Veronica Silva	10500.10

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

```
113 -- @ Concatenar el nombre del cliente y su email con un separador
114 • select concat(nombre_cliente, " ",email_cliente) as nombre_correo_cliente from cliente;
```

nombre_correo_cliente
Juan Perez juan.perez@example.com
Maria Lopez maria.lopez@example.com
Carlos Sanchez carlos.sanchez@example.com
Ana Fernandez ana.fernandez@example.com
Luis Gomez luis.gomez@example.com
Sofia Ramirez sofia.ramirez@example.com
Jorge Diaz jorge.diaz@example.com
Marta Torres marta.torres@example.com
Pedro Ruiz pedro.ruiz@example.com
Lucia Flores lucia.flores@example.com
Miguel Vega miguel.vega@example.com
Laura Castillo laura.castillo@example.com
Ricardo Jimenez ricardo.jimenez@example.com
Carmen Morales carmen.morales@example.com
Alberto Rios alberto.rios@example.com
Gloria Castro gloria.castro@example.com
Enrique Herrera enrique.herrera@example.com
Patricia Guzman patricia.guzman@example.com
Francisco Ortiz francisco.ortiz@example.com
Veronica Silva veronica.silva@example.com

- Extraer los últimos 3 caracteres del email de cada cliente

```
116 -- @ Extraer los últimos 3 caracteres del email de cada cliente
117 • select nombre_cliente, email_cliente, substring(email_cliente, -3) as ultimos_3_caracteres from cliente;
```

nombre_cliente	email_cliente	ultimos_3_caracteres
Juan Perez	juan.perez@example.com	com
Maria Lopez	maria.lopez@example.com	com
Carlos Sanchez	carlos.sanchez@example.com	com
Ana Fernandez	ana.fernandez@example.com	com
Luis Gomez	luis.gomez@example.com	com
Sofia Ramirez	sofia.ramirez@example.com	com
Jorge Diaz	jorge.diaz@example.com	com
Marta Torres	marta.torres@example.com	com
Pedro Ruiz	pedro.ruiz@example.com	com
Lucia Flores	lucia.flores@example.com	com
Miguel Vega	miguel.vega@example.com	com
Laura Castillo	laura.castillo@example.com	com
Ricardo Jimenez	ricardo.jimenez@example.com	com
Carmen Morales	carmen.morales@example.com	com
Alberto Rios	alberto.rios@example.com	com
Gloria Castro	gloria.castro@example.com	com
Enrique Herrera	enrique.herrera@example.com	com
Patricia Guzman	patricia.guzman@example.com	com
Francisco Ortiz	francisco.ortiz@example.com	com
Veronica Silva	veronica.silva@example.com	com

- Convertir el nombre del cliente a minúsculas

```
119 -- Convertir el nombre del cliente a minúsculas
120 • select nombre_cliente, lower(nombre_cliente) as nombre_minusculas from cliente;
```

nombre_cliente	nombre_minusculas
Juan Perez	juan perez
Maria Lopez	maria lopez
Carlos Sanchez	carlos sanchez
Ana Fernandez	ana fernandez
Luis Gomez	luis gomez
Sofia Ramirez	sofia ramirez
Jorge Diaz	jorge diaz
Marta Torres	marta torres
Pedro Ruiz	pedro ruiz
Lucia Flores	lucia flores
Miguel Vega	miguel vega
Laura Castillo	laura castillo
Ricardo Jimenez	ricardo jimenez
Carmen Morales	carmen morales
Alberto Rios	alberto rios
Gloria Castro	gloria castro
Enrique Herrera	enrique herrera
Patricia Guzman	patricia guzman
Francisco Ortiz	francisco ortiz
Veronica Silva	veronica silva

- Convertir el nombre del cliente a mayúsculas

```
122 -- Convertir el nombre del cliente a mayúsculas
123 • select nombre_cliente, upper(nombre_cliente) as nombre_mayusculas from cliente;
```

nombre_cliente	nombre_mayusculas
Juan Perez	JUAN PEREZ
Maria Lopez	MARIA LOPEZ
Carlos Sanchez	CARLOS SANCHEZ
Ana Fernandez	ANA FERNANDEZ
Luis Gomez	LUIS GOMEZ
Sofia Ramirez	SOFIA RAMIREZ
Jorge Diaz	JORGE DIAZ
Marta Torres	MARTA TORRES
Pedro Ruiz	PEDRO RUIZ
Lucia Flores	LUCIA FLORES
Miguel Vega	MIGUEL VEGA
Laura Castillo	LAURA CASTILLO
Ricardo Jimenez	RICARDO JIMENEZ
Carmen Morales	CARMEN MORALES
Alberto Rios	ALBERTO RIOS
Gloria Castro	GLORIA CASTRO
Enrique Herrera	ENRIQUE HERRERA
Patricia Guzman	PATRICIA GUZMAN
Francisco Ortiz	FRANCISCO ORTIZ
Veronica Silva	VERONICA SILVA

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

```
125 -- @ Seleccionar las transacciones ordenadas por monto de mayor a menor, mostrando solo las primeras 10
126 • select id_transaccion, monto_transaccion, fecha_transaccion from transaccion order by monto_transaccion desc limit 10;
```

Result Grid			
Filter Rows:			
Edit: Export/Import: Wrap Cell Content: Fetch rows:			
	id_transaccion	monto_transaccion	fecha_transaccion
▶	20	1050.10	2024-08-08
	19	1000.00	2024-08-06
	18	950.95	2024-08-04
	17	900.85	2024-08-02
	16	850.75	2024-07-31
	15	800.65	2024-07-29
	14	750.50	2024-07-27
	13	700.35	2024-07-25
	12	650.25	2024-07-23
	11	600.00	2024-07-21
•	NULL	NULL	NULL

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

```
128 -- @ Calcular el saldo de cada cuenta después de aplicar un interés del 5%
129 • select id_cuenta, saldo_cuenta, round(saldo_cuenta + (saldo_cuenta*0.05), 2) as interes_5 from cuenta;
```

Result Grid			
Filter Rows:			
Export: Wrap Cell Content:			
	id_cuenta	saldo_cuenta	interes_5
▶	1	1000.50	1050.53
	2	2000.75	2100.79
	3	1500.30	1575.32
	4	2500.10	2625.11
	5	3000.60	3150.63
	6	3500.80	3675.84
	7	4000.40	4200.42
	8	4500.70	4725.74
	9	5000.20	5250.21
	10	5500.90	5775.95
	11	6000.00	6300.00
	12	6500.25	6825.26
	13	7000.35	7350.37
	14	7500.50	7875.53
	15	8000.65	8400.68
	16	8500.75	8925.79
	17	9000.85	9450.89
	18	9500.95	9976.00
	19	10000.00	10500.00
	20	10500.10	11025.11

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

```
131 -- @ Reemplazar el dominio del email de los clientes: email, '@example.com', '@newdomain.com'
132 • update cliente set email_cliente = replace(email_cliente, '@example.com', '@newdomain.com');
133 • select * from cliente;
```

Result Grid			
Filter Rows: <input type="text"/>			
Edit:			
Export/Import:			
Wrap Cell Content:			
	id_cliente	nombre_cliente	email_cliente
▶	1	Juan Perez	juan.perez@newdomain.com
	2	Maria Lopez	maria.lopez@newdomain.com
	3	Carlos Sanchez	carlos.sanchez@newdomain.com
	4	Ana Fernandez	ana.fernandez@newdomain.com
	5	Luis Gomez	luis.gomez@newdomain.com
	6	Sofia Ramirez	sofia.ramirez@newdomain.com
	7	Jorge Diaz	jorge.diaz@newdomain.com
	8	Marta Torres	marta.torres@newdomain.com
	9	Pedro Ruiz	pedro.ruiz@newdomain.com
	10	Lucia Flores	lucia.flores@newdomain.com
	11	Miguel Vega	miguel.vega@newdomain.com
	12	Laura Castillo	laura.castillo@newdomain.com
	13	Ricardo Jimenez	ricardo.jimenez@newdomain.com
	14	Carmen Morales	carmen.morales@newdomain.com
	15	Alberto Rios	alberto.rios@newdomain.com
	16	Gloria Castro	gloria.castro@newdomain.com
	17	Enrique Herrera	enrique.herrera@newdomain.com
	18	Patricia Guzman	patricia.guzman@newdomain.com
	19	Francisco Ortiz	francisco.ortiz@newdomain.com
	20	Veronica Silva	veronica.silva@newdomain.com
*	NULL	NULL	NULL

Ejercicio 2: Uso de Funciones de Cadena

- Selecciona el nombre de cada cliente.

```
135 -- @ Selecciona el nombre de cada cliente.
136 • select nombre_cliente from cliente;
```

Result Grid	
Filter Rows: <input type="text"/>	
Export:	
Wrap Cell Content:	
	nombre_cliente
▶	Juan Perez
	Maria Lopez
	Carlos Sanchez
	Ana Fernandez
	Luis Gomez
	Sofia Ramirez
	Jorge Diaz
	Marta Torres
	Pedro Ruiz
	Lucia Flores
	Miguel Vega
	Laura Castillo
	Ricardo Jimenez
	Carmen Morales
	Alberto Rios
	Gloria Castro
	Enrique Herrera
	Patricia Guzman
	Francisco Ortiz
	Veronica Silva

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

```
138 -- @ Añade una columna que muestre el email del cliente en el formato Correo: [email].
139 • select nombre_cliente, concat("Correo: ", email_cliente) as correo_cliente from cliente;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
nombre_cliente	correo_cliente			
Juan Perez	Correo: juan.perez@newdomain.com			
Maria Lopez	Correo: maria.lopez@newdomain.com			
Carlos Sanchez	Correo: carlos.sanchez@newdomain.com			
Ana Fernandez	Correo: ana.fernandez@newdomain.com			
Luis Gomez	Correo: luis.gomez@newdomain.com			
Sofia Ramirez	Correo: sofia.ramirez@newdomain.com			
Jorge Diaz	Correo: jorge.diaz@newdomain.com			
Marta Torres	Correo: marta.torres@newdomain.com			
Pedro Ruiz	Correo: pedro.ruiz@newdomain.com			
Lucia Flores	Correo: lucia.flores@newdomain.com			
Miguel Vega	Correo: miguel.vega@newdomain.com			
Laura Castillo	Correo: laura.castillo@newdomain.com			
Ricardo Jimenez	Correo: ricardo.jimenez@newdomain.com			
Carmen Morales	Correo: carmen.morales@newdomain.com			
Alberto Rios	Correo: alberto.rios@newdomain.com			
Gloria Castro	Correo: gloria.castro@newdomain.com			
Enrique Herrera	Correo: enrique.herrera@newdomain.com			
Patricia Guzman	Correo: patricia.guzman@newdomain.com			
Francisco Ortiz	Correo: francisco.ortiz@newdomain.com			
Veronica Silva	Correo: veronica.silva@newdomain.com			

- Mostrar los emails de los clientes en minúsculas


```
141 -- @ Mostrar los emails de los clientes en minúsculas
142 • select nombre_cliente, lower(email_cliente) as email_minusculas from cliente;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
nombre_cliente	email_minusculas			
Juan Perez	juan.perez@newdomain.com			
Maria Lopez	maria.lopez@newdomain.com			
Carlos Sanchez	carlos.sanchez@newdomain.com			
Ana Fernandez	ana.fernandez@newdomain.com			
Luis Gomez	luis.gomez@newdomain.com			
Sofia Ramirez	sofia.ramirez@newdomain.com			
Jorge Diaz	jorge.diaz@newdomain.com			
Marta Torres	marta.torres@newdomain.com			
Pedro Ruiz	pedro.ruiz@newdomain.com			
Lucia Flores	lucia.flores@newdomain.com			
Miguel Vega	miguel.vega@newdomain.com			
Laura Castillo	laura.castillo@newdomain.com			
Ricardo Jimenez	ricardo.jimenez@newdomain.com			
Carmen Morales	carmen.morales@newdomain.com			
Alberto Rios	alberto.rios@newdomain.com			
Gloria Castro	gloria.castro@newdomain.com			
Enrique Herrera	enrique.herrera@newdomain.com			
Patricia Guzman	patricia.guzman@newdomain.com			
Francisco Ortiz	francisco.ortiz@newdomain.com			
Veronica Silva	veronica.silva@newdomain.com			

- Mostrar el nombre del cliente en mayúsculas


```
141 -- @ Mostrar los emails de los clientes en minúsculas
142 • select nombre_cliente, lower(email_cliente) as email_minusculas from cliente;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:




	nombre_cliente	nombre_mayusculas
▶	Juan Perez	JUAN PEREZ
	Maria Lopez	MARIA LOPEZ
	Carlos Sanchez	CARLOS SANCHEZ
	Ana Fernandez	ANA FERNANDEZ
	Luis Gomez	LUIS GOMEZ
	Sofia Ramirez	SOFIA RAMIREZ
	Jorge Diaz	JORGE DIAZ
	Marta Torres	MARTA TORRES
	Pedro Ruiz	PEDRO RUIZ
	Lucia Flores	LUCIA FLORES
	Miguel Vega	MIGUEL VEGA
	Laura Castillo	LAURA CASTILLO
	Ricardo Jimenez	RICARDO JIMENEZ
	Carmen Morales	CARMEN MORALES
	Alberto Rios	ALBERTO RIOS
	Gloria Castro	GLORIA CASTRO
	Enrique Herrera	ENRIQUE HERRERA
	Patricia Guzman	PATRICIA GUZMAN
	Francisco Ortiz	FRANCISCO ORTIZ
	Veronica Silva	VERONICA SILVA

Ejercicio 3: Agregación de Datos

- Selecciona el id_cuenta.

```
148 -- Ejercicio 3: Agregación de Datos
149 -- @ Selecciona el id_cuenta.
150 • select id_cuenta from cuenta;
```

Result Grid			Filter Rows: <input type="text"/>	Edit: <input type="text"/>
	id_cuenta			
▶	1			
	2			
	3			
	4			
	5			
	6			
	7			
	8			
	9			
	10			
	11			
	12			
	13			
	14			
	15			
	16			
	17			
	18			
	19			
	20			
✱	NULL			

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

```

152 --  Calcula el total de las transacciones realizadas en cada cuenta.
153 • select cuenta.id_cuenta, count(transaccion.id_transaccion) as cantidad_transacciones from cuenta left join transaccion on cuenta.id_cuenta = transaccion.id_transaccion group by cuenta.id_cuenta;

```

id_cuenta	cantidad_transacciones
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1

- Calcular el saldo total de todas las cuentas

```

155 --  Calcular el saldo total de todas las cuentas
156 • select sum(saldo_cuenta) as saldo_total_cuentas from cuenta;

```

saldo_total_cuentas
115009.65

- Calcular el monto total de todas las transacciones

```

158 --  Calcular el monto total de todas las transacciones
159 • select sum(monto_transaccion) as monto_total_transaccion from transaccion;

```

monto_total_transaccion
11509.65

- Calcular el monto promedio de las transacciones

sdñ

```

161 --  Calcular el monto promedio de las transacciones
162 • select round(avg(monto_transaccion), 2) as promedio_monto_transaccion from transaccion;

```

promedio_monto_transaccion
575.48

Ejercicio 4: Funciones de Fecha:

- Selecciona el id_transaccion y la fecha de todas las transacciones.

```
165 -- @ Selecciona el id_transaccion y la fecha de todas las transacciones.
166 • select id_transaccion, fecha_transaccion from transaccion;
```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell
	id_transaccion	fecha_transaccion			
▶	1	2024-07-01			
	2	2024-07-03			
	3	2024-07-05			
	4	2024-07-07			
	5	2024-07-09			
	6	2024-07-11			
	7	2024-07-13			
	8	2024-07-15			
	9	2024-07-17			
	10	2024-07-19			
	11	2024-07-21			
	12	2024-07-23			
	13	2024-07-25			
	14	2024-07-27			
	15	2024-07-29			
	16	2024-07-31			
	17	2024-08-02			
	18	2024-08-04			
	19	2024-08-06			
	20	2024-08-08			
•	NULL	NULL			

- Extraer el año de la fecha de cada transacción

```
168 -- @ Extraer el año de la fecha de cada transacción
169 • select fecha_transaccion, year(fecha_transaccion) as anio_transaccion from transaccion;
```

Result Grid		Filter Rows:	Exports:	Wrap Cell Content:
	fecha_transaccion	anio_transaccion		
▶	2024-07-01	2024		
	2024-07-03	2024		
	2024-07-05	2024		
	2024-07-07	2024		
	2024-07-09	2024		
	2024-07-11	2024		
	2024-07-13	2024		
	2024-07-15	2024		
	2024-07-17	2024		
	2024-07-19	2024		
	2024-07-21	2024		
	2024-07-23	2024		
	2024-07-25	2024		
	2024-07-27	2024		
	2024-07-29	2024		
	2024-07-31	2024		
	2024-08-02	2024		
	2024-08-04	2024		
	2024-08-06	2024		
	2024-08-08	2024		

- Extraer el mes de la fecha de cada transacción

```
171 -- @ Extraer el mes de la fecha de cada transacción
172 • select fecha_transaccion, monthname(fecha_transaccion) as mes_transaccion from transaccion;
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:		
	fecha_transaccion	mes_transaccion
▶	2024-07-01	July
	2024-07-03	July
	2024-07-05	July
	2024-07-07	July
	2024-07-09	July
	2024-07-11	July
	2024-07-13	July
	2024-07-15	July
	2024-07-17	July
	2024-07-19	July
	2024-07-21	July
	2024-07-23	July
	2024-07-25	July
	2024-07-27	July
	2024-07-29	July
	2024-07-31	July
	2024-08-02	August
	2024-08-04	August
	2024-08-06	August
	2024-08-08	August

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

```
174 -- @ Extraer el día de la semana de la fecha de cada transacción
175 • select fecha_transaccion, dayname(fecha_transaccion) as dia_semana_transaccion from transaccion;
```

Result Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:		
	fecha_transaccion	dia_semana_transaccion
▶	2024-07-01	Monday
	2024-07-03	Wednesday
	2024-07-05	Friday
	2024-07-07	Sunday
	2024-07-09	Tuesday
	2024-07-11	Thursday
	2024-07-13	Saturday
	2024-07-15	Monday
	2024-07-17	Wednesday
	2024-07-19	Friday
	2024-07-21	Sunday
	2024-07-23	Tuesday
	2024-07-25	Thursday
	2024-07-27	Saturday
	2024-07-29	Monday
	2024-07-31	Wednesday
	2024-08-02	Friday
	2024-08-04	Sunday
	2024-08-06	Tuesday
	2024-08-08	Thursday

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

```
177 -- Filtra las transacciones que se realizaron en el año 2024.
178 • select id_transaccion, fecha_transaccion from transaccion where year(fecha_transaccion) = 2024;
```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id_transaccion	fecha_transaccion			
▶	1	2024-07-01			
	2	2024-07-03			
	3	2024-07-05			
	4	2024-07-07			
	5	2024-07-09			
	6	2024-07-11			
	7	2024-07-13			
	8	2024-07-15			
	9	2024-07-17			
	10	2024-07-19			
	11	2024-07-21			
	12	2024-07-23			
	13	2024-07-25			
	14	2024-07-27			
	15	2024-07-29			
	16	2024-07-31			
	17	2024-08-02			
	18	2024-08-04			
	19	2024-08-06			
	20	2024-08-08			
*	NULL	NULL			

- Seleccionar las transacciones realizadas en 2023.

```
180 -- Seleccionar las transacciones realizadas en 2023.
181 • select id_transaccion, fecha_transaccion from transaccion where year(fecha_transaccion) = 2023;
```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id_transaccion	fecha_transaccion			
*	NULL	NULL			

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

```
183 -- Seleccionar las transacciones realizadas en el mes de junio de cualquier año
184 • select id_transaccion, fecha_transaccion from transaccion where monthname(fecha_transaccion) = "June";
```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id_transaccion	fecha_transaccion			
*	NULL	NULL			

- Seleccionar las transacciones realizadas el 15 de cada mes

```
186 -- Seleccionar las transacciones realizadas el 15 de cada mes
187 • select fecha_transaccion, dayofmonth(fecha_transaccion) as dia_semana_transaccion from transaccion where dayofmonth(fecha_transaccion) = 15;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	fecha_transaccion	dia_semana_transaccion		
▶	2024-07-15	15		

Ejercicio 5: Agrupación y Contar Resultados

Objetivo: Contar el número de transacciones por cuenta.

- Selecciona el id_cuenta.

```
189 -- Ejercicio 5: Agrupación y Contar Resultados
190 -- @ Selecciona el id_cuenta.
191 • select id_cuenta from cuenta;
```

id_cuenta
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
NULL

- Cuenta el número de transacciones realizadas en cada cuenta.

```
193 -- @ Cuenta el número de transacciones realizadas en cada cuenta.
194 • select cuenta.id_cuenta, count(transaccion.id_transaccion) as cantidad_transacciones from cuenta left join transaccion on cuenta.id_cuenta = transaccion.id_transaccion group by cuenta.id_cuenta;
```

id_cuenta	cantidad_transacciones
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1

- Calcular el saldo promedio de todas las cuentas.

```
196 -- @ Calcular el saldo promedio de todas las cuentas.
197 • select round(avg(saldo_cuenta), 2) as promedio_saldo_cuenta from cuenta;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	promedio_saldo_cuenta			
▶	5750.48			

Ejercicio 6: Filtración con Condiciones

- Seleccionar las transacciones con un monto negativo

```
199 -- Ejercicio 6: Filtración con Condiciones
200 -- @ Seleccionar las transacciones con un monto negativo
201 • select id_transaccion, monto_transaccion from transaccion where monto_transaccion < 0;
```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id_transaccion	monto_transaccion			
*	NULL	NULL			

- Seleccionar las transacciones con un monto positivo

```
203 -- @ Seleccionar las transacciones con un monto positivo
204 • select id_transaccion, monto_transaccion from transaccion where monto_transaccion > 0;
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	id_transaccion	monto_transaccion
▶	1	100.50
	2	200.75
	3	150.30
	4	250.10
	5	300.60
	6	350.80
	7	400.40
	8	450.70
	9	500.20
	10	550.90
	11	600.00
	12	650.25
	13	700.35
	14	750.50
	15	800.65
	16	850.75
	17	900.85
	18	950.95
	19	1000.00
	20	1050.10

Ejercicio 7: Ordenamiento

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

- Seleccionar todos los clientes ordenados alfabéticamente por nombre

```
206 -- Ejercicio 7: Ordenamiento
207 -- @ Seleccionar todos los clientes ordenados alfabéticamente por nombre
208 • select id_cliente, nombre_cliente from cliente order by nombre_cliente asc;
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Co
	id_cliente	nombre_cliente		
▶	15	Alberto Rios		
	4	Ana Fernandez		
	3	Carlos Sanchez		
	14	Carmen Morales		
	17	Enrique Herrera		
	19	Francisco Ortiz		
	16	Gloria Castro		
	7	Jorge Diaz		
	1	Juan Perez		
	12	Laura Castillo		
	10	Lucia Flores		
	5	Luis Gomez		
	2	Maria Lopez		
	8	Marta Torres		
	11	Miguel Vega		
	18	Patricia Guzman		
	9	Pedro Ruiz		
	13	Ricardo Jimenez		
	6	Sofia Ramirez		
	20	Veronica Silva		
*	NULL	NULL		

- Seleccionar las transacciones realizadas en 2023

```
210 -- @ Seleccionar las transacciones realizadas en 2023
211 • select id_transaccion, fecha_transaccion from transaccion where year(fecha_transaccion) = 2023;
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id_transaccion	fecha_transaccion		
*	NULL	NULL		

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

```
213 -- @ Seleccionar las cuentas ordenadas por saldo de menor a mayor, mostrando solo las primeras 5
214 • select saldo_cuenta from cuenta order by saldo_cuenta desc limit 5;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	saldo_cuenta			
▶	10500.10			
	10000.00			
	9500.95			
	9000.85			
	8500.75			

Ejercicio 8: Contar Resultados

- Contar el número total de clientes

```
216      -- Ejercicio 8: Contar Resultados
217      -- ② Contar el número total de clientes
218 •    select count(id_cliente) as total_clientes from cliente;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_clientes			
▶	20			

- Contar el número total de cuentas

```
220      -- ② Contar el número total de cuentas
221 •    select count(id_cuenta) as total_cuentas from cuenta;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_cuentas			
▶	20			

- Contar el número total de transacciones

```
223      -- ② Contar el número total de transacciones
224 •    select count(id_transaccion) as total_transacciones from transaccion;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_transacciones			
▶	20			

- Redondear el saldo de cada cuenta a dos decimales

```
226 -- Redondear el saldo de cada cuenta a dos decimales
227 • select id_cuenta, round(saldo_cuenta, 2) as saldo_cuenta_redondeado from cuenta;
```

	id_cuenta	saldo_cuenta_redondeado
▶	1	1000.50
	2	2000.75
	3	1500.30
	4	2500.10
	5	3000.60
	6	3500.80
	7	4000.40
	8	4500.70
	9	5000.20
	10	5500.90
	11	6000.00
	12	6500.25
	13	7000.35
	14	7500.50
	15	8000.65
	16	8500.75
	17	9000.85
	18	9500.95
	19	10000.00
	20	10500.10

Ejercicio 9: Máximos y Mínimos

- Encontrar la cuenta con el saldo más alto

```
229 -- Ejercicio 9: Máximos y Mínimos
230 -- Encontrar la cuenta con el saldo más alto
231 • select id_cuenta, saldo_cuenta as cuenta_saldo_alto from cuenta where saldo_cuenta = (select max(saldo_cuenta) from cuenta);
```

	id_cuenta	cuenta_saldo_alto
▶	20	10500.10
	NULL	NULL
*		

- Encontrar la cuenta con el saldo más bajo

```
233 -- Encontrar la cuenta con el saldo más bajo
234 • select id_cuenta, saldo_cuenta as cuenta_saldo_bajo from cuenta where saldo_cuenta = (select min(saldo_cuenta) from cuenta);
```

	id_cuenta	cuenta_saldo_bajo
▶	1	1000.50
	NULL	NULL
*		

- Encontrar la transacción con el monto más alto

```
236 -- # Encontrar la transacción con el monto más alto
237 • select id_transaccion, monto_transaccion as monto_transaccion_alto from transaccion where monto_transaccion = (select max(monto_transaccion) from transaccion);
```

Result Grid

	id_transaccion	monto_transaccion_alto
▶	20	1050.10
•	NULL	NULL

- Encontrar la transacción con el monto más bajo

```
239 -- # Encontrar la transacción con el monto más bajo
240 • select id_transaccion, monto_transaccion as monto_transaccion_bajo from transaccion where monto_transaccion = (select min(monto_transaccion) from transaccion);
```

Result Grid

	id_transaccion	monto_transaccion_bajo
▶	1	100.50
•	NULL	NULL

PRESENTACIÓN

- PRESENTAR EN PDF
- SUBIR A GIT HUB
- Y SUBIR EL SQL-QUERY

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