

# **PROYECTO FINAL – SISTEMA DE GESTIÓN BANCARIA**

**Curso: Bases de Datos**

Generación 1 — Mayo 2025

---

## **Caso de Estudio: Banco**

Diseño, implementación y documentación de una base de datos relacional.

### **Autores:**

**Garcia Monjaraz Jessica Stephanie  
Jiménez Sánchez Eduardo Yael**

*Este documento contiene el modelo del caso, los scripts SQL y evidencias gráficas del correcto funcionamiento.*

Fecha de entrega: **18 de mayo de 2025**

# Contenidos

<b>Introducción</b>	<b>3</b>
Ejecución del Script DDL . . . . .	3
Capturas por script que muestran la ejecución. . . . .	3
Extra: . . . . .	11
Ejecución del Script DML . . . . .	11
Capturas por script que muestran la ejecución. . . . .	12
Ejecución del Script DQL . . . . .	23
Capturas por script que muestran la ejecución. . . . .	24
<b>Conclusiones:</b>	<b>31</b>
Jessica Stephanie Garcia Monjaraz: . . . . .	31
Eduardo Yael Jiménez Sanchez: . . . . .	32

# Evidencias de Ejecución de Scripts SQL del proyecto de gestión bancaria.

👤 Stephanie Monjaraz & 🤤 Eduardo Jiménez

## Introducción

---

El presente documento contiene las evidencias de ejecución de los tres scripts requeridos en el proyecto final de la materia "Bases de Datos":

- **DDL\_GMJS\_EYJS.sql** – Define la estructura de la base de datos mediante la creación de tablas, claves primarias, foráneas y restricciones.
- **DML\_GMJS\_EYJS.sql** – Inserta registros de ejemplo y realiza operaciones de modificación, eliminación y actualización.
- **DQL\_GMJS\_EYJS.sql** – Contiene consultas que permiten verificar la correcta organización y recuperación de los datos en un contexto realista.

Cada sección de este documento presenta las sentencias ejecutadas y las capturas de pantalla que validan su correcta implementación en el gestor de bases de datos.

## Ejecución del Script DDL

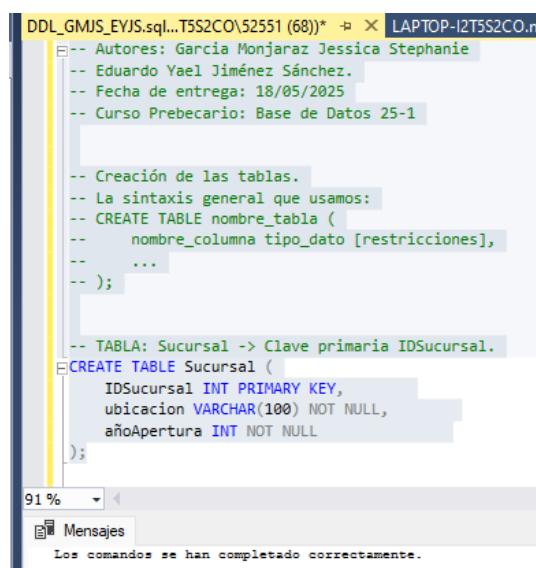
---

En esta sección presentamos las evidencias correspondientes a la ejecución correcta del archivo **DDL\_GMJS\_EYJS.sql**, en el cual, definimos la estructura de la base de datos del proyecto final "Sistema de Gestión Bancaria".

Este script incluye la creación de todas las tablas derivadas del análisis del caso de estudio, con sus respectivos atributos, claves primarias, claves foráneas, restricciones y relaciones. Las capturas de pantalla muestran que cada sentencia fue ejecutada exitosamente en el sistema gestor de bases de datos.

### Capturas por script que muestran la ejecución.

---



```

-- Autores: García Monjaraz Jessica Stephanie
-- Eduardo Yael Jiménez Sánchez.
-- Fecha de entrega: 18/05/2025
-- Curso Prebacario: Base de Datos 25-1

-- Creación de las tablas.
-- La sintaxis general que usamos:
-- CREATE TABLE nombre_tabla (
--   nombre_columna tipo_dato [restricciones],
--   ...
-- );

-- TABLA: Sucursal -> Clave primaria IDSucursal.
CREATE TABLE Sucursal (
  IDSucursal INT PRIMARY KEY,
  ubicacion VARCHAR(100) NOT NULL,
  añoApertura INT NOT NULL
);

Los comandos se han completado correctamente.
  
```

Figura 1. Creación de la tabla Sucursal. Incluye comentarios con la sintaxis general y definición explícita de tipos de datos, restricciones NOT NULL y clave primaria.

Nombre de columna	Tipo de datos	Permitir valores NULL
IDSucursal	int	<input type="checkbox"/>
ubicacion	varchar(100)	<input type="checkbox"/>
añoApertura	int	<input type="checkbox"/>
		<input type="checkbox"/>

Figura 2. Estructura de la tabla Sucursal creada en SQL Server Management Studio. Se muestra la definición de atributos IDSucursal, ubicacion y añoApertura, así como la restricción de clave primaria sobre IDSucursal.

```
-- TABLA: Empleado -> Clave primaria: IDEmppleado. Llaves foráneas: IDJefe e IDSucursal.
-- Sintaxis para llaves foráneas:
-- FOREIGN KEY (columna_relacionada) REFERENCES TablaRelacionada (llave_primaria)
CREATE TABLE Empleado (
    IDEmppleado INT PRIMARY KEY,
    nombres VARCHAR(100) NOT NULL,
    apellidoPaterno VARCHAR(50) NOT NULL,
    apellidoMaterno VARCHAR(50),
    fechaContratacion DATE NOT NULL,
    IDJefe INT,
    IDSucursal INT NOT NULL,
    FOREIGN KEY (IDJefe) REFERENCES Empleado(IDEmpleado),
    FOREIGN KEY (IDSucursal) REFERENCES Sucursal(IDSucursal)
);
```

91 % ▾

Mensajes

Los comandos se han completado correctamente.

Figura 3. Creación de la tabla Empleado. Incluye comentarios con la sintaxis general y definición explícita de tipos de datos, restricciones NOT NULL y clave primaria.

Nombre de columna	Tipo de datos	Permitir valores NULL
IDEmppleado	int	<input type="checkbox"/>
nombres	varchar(100)	<input type="checkbox"/>
apellidoPaterno	varchar(50)	<input type="checkbox"/>
apellidoMaterno	varchar(50)	<input checked="" type="checkbox"/>
fechaContratacion	date	<input type="checkbox"/>
IDJefe	int	<input checked="" type="checkbox"/>
IDSucursal	int	<input type="checkbox"/>
		<input type="checkbox"/>

Figura 4. Estructura de la tabla Empleado creada en SQL Server Management Studio. Se muestra la definición de los atributos así como las claves foráneas sobre IDSucursal e IDJefe.

```
-- TABLA: Cliente -> Clave primaria: IDCliente. Llaves foráneas: IDSucursal.
CREATE TABLE Cliente (
    IDCliente INT PRIMARY KEY,
    nombres VARCHAR(100) NOT NULL,
    apellidoPaterno VARCHAR(50) NOT NULL,
    apellidoMaterno VARCHAR(50),
    direccion VARCHAR(150) NOT NULL,
    IDSucursal INT NOT NULL,
    FOREIGN KEY (IDSucursal) REFERENCES Sucursal(IDSucursal)
);

91 % ▾
Mensajes
Los comandos se han completado correctamente.
```

Figura 5. Creación de la tabla Cliente. Llave primaria IDCliente y llave foránea IDSucursal.

Nombre de columna	Tipo de datos	Permitir valores NULL
IDCliente	int	<input type="checkbox"/>
nombres	varchar(100)	<input type="checkbox"/>
apellidoPaterno	varchar(50)	<input type="checkbox"/>
apellidoMaterno	varchar(50)	<input checked="" type="checkbox"/>
direccion	varchar(150)	<input type="checkbox"/>
IDSucursal	int	<input type="checkbox"/>
		<input type="checkbox"/>

Figura 6. Estructura de la tabla Cliente creada en SQL Server Management Studio. Se muestra la definición de los atributos así como las claves foráneas sobre IDSucursal e IDJefe.

```
-- TABLA: Cuenta -> Clave primaria: NumeroCuenta. Llaves foráneas: IDCliente.
CREATE TABLE Cuenta (
    NumeroCuenta INT PRIMARY KEY,
    -- Con CHECK restringimos el tipo de cuenta a los dos valores válidos.
    EsUna VARCHAR(20) CHECK (EsUna IN ('Ahorro', 'Corriente')),
    saldoInicial DECIMAL(10,2) NOT NULL,
    IDCliente INT NOT NULL,
    FOREIGN KEY (IDCliente) REFERENCES Cliente(IDCliente) -- Vinculamso a la cuenta con su propietario.
);

% ▾
Mensajes
Los comandos se han completado correctamente.
```

Figura 7. Creación de la tabla Cuenta. Incluye la restricción para solo permitir los dos tipos de cuentas. Llave primaria NumeroCuenta y llave foránea IDCliente.

```
-- TABLA: CuentaAhorro -> Clave primaria: NumeroCuenta. Llaves foráneas: NumeroCuenta.
CREATE TABLE CuentaAhorro (
    NumeroCuenta INT PRIMARY KEY, -- También es su clave foránea y se hereda de Cuenta,
    interes DECIMAL(5,2) NOT NULL,
    duracionPlazo INT NOT NULL, -- Duración del plazo de ahorro, en meses o años.
    FOREIGN KEY (NumeroCuenta) REFERENCES Cuenta(NumeroCuenta)
);

L % <
Mensajes
Los comandos se han completado correctamente.
```

Figura 8. Creación de la tabla CuentaAhorro. Llave primaria NumeroCuenta.

	Nombre de columna	Tipo de datos	Permitir valores NULL
PK	NumeroCuenta	int	<input type="checkbox"/>
	interes	decimal(5, 2)	<input type="checkbox"/>
	duracionPlazo	int	<input type="checkbox"/>

Figura 9. Estructura de la tabla CuentaAhorro.

```
-- TABLA: CuentaCorriente -> Clave primaria: NumeroCuenta. Llaves foráneas: NumeroCuenta e IDAsesor
CREATE TABLE CuentaCorriente (
    NumeroCuenta INT PRIMARY KEY,
    numeroChequera VARCHAR(50) NOT NULL,
    saldoMin DECIMAL(10,2) NOT NULL,
    IDAsesor INT NOT NULL,
    -- Clave foránea para conocer que empleado asesora la cuenta.
    FOREIGN KEY (NumeroCuenta) REFERENCES Cuenta(NumeroCuenta),
    FOREIGN KEY (IDAsesor) REFERENCES Empleado(IDEmpleado) *
);

1 %
Mensajes
Los comandos se han completado correctamente.
```

Figura 10. Creación de la tabla CuentaCorriente. Llave primaria NumeroCuenta y llaves foráneas NumeroCuenta e IDAsesor.

	Nombre de columna	Tipo de datos	Permitir valores NULL
PK	NumeroCuenta	int	<input type="checkbox"/>
	numeroChequera	varchar(50)	<input type="checkbox"/>
	saldoMin	decimal(10, 2)	<input type="checkbox"/>
	IDAsesor	int	<input type="checkbox"/>

Figura 11. Estructura de la tabla CuentaCorriente.

```
-- TABLA: Prestamo -> Clave primaria: IDPrestamo. Llave foránea: IDCiente.
CREATE TABLE Prestamo (
    IDPrestamo INT PRIMARY KEY,
    fechaSolicitud DATE NOT NULL,
    siPrimerPrestamo BIT NOT NULL, -- booleano 1 = sí, 0 = no.
    montoTotal DECIMAL(10,2) NOT NULL,
    -- Con CHECK restringimos el plazo del prestamos a los tres valores válidos.
    plazoMeses INT CHECK (plazoMeses IN (6, 12, 24)),-- Plazo limitado.
    IDCiente INT NOT NULL,
    FOREIGN KEY (IDCiente) REFERENCES Cliente(IDCiente)
);
```

L % Mensajes  
Los comandos se han completado correctamente.

Figura 12. Creación de la tabla Prestamo. Llave primaria IDCliente y restricción para solo aceptar plazos de 6,12 y 24 meses.

Nombre de columna	Tipo de datos	Permitir valores NULL
IDPrestamo	int	<input type="checkbox"/>
FechaDeSolicitud	date	<input type="checkbox"/>
PlazoMeses	int	<input type="checkbox"/>
MontoTotal	decimal(15, 2)	<input type="checkbox"/>
SiPrimerPrestamo	bit	<input type="checkbox"/>
idCliente	int	<input type="checkbox"/>
		<input type="checkbox"/>

Figura 13. Estructura de la tabla Prestamo.

```
-- TABLA: Pago -> Clave primaria: IDPago. Llave foránea: IDPrestamo.
CREATE TABLE Pago (
    IDPago INT PRIMARY KEY,
    numeroPago INT NOT NULL,
    montoPago DECIMAL(10,2) NOT NULL,
    IDPrestamo INT NOT NULL,
    FOREIGN KEY (IDPrestamo) REFERENCES Prestamo(IDPrestamo)
);
```

1 % Mensajes  
Los comandos se han completado correctamente.

Figura 14. Creación de la tabla Pago. Llave primaria IDPago y llave foránea IDPrestamos.

	Nombre de columna	Tipo de datos	Permitir valores NULL
PK	IDPago	int	<input type="checkbox"/>
	numeroPago	int	<input type="checkbox"/>
	montoPago	decimal(10, 2)	<input type="checkbox"/>
	IDPrestamo	int	<input type="checkbox"/>
			<input type="checkbox"/>

Figura 15. Estructura de la tabla Pago.

```
-- TABLA: Inversion -> Clave primaria: CódigoInversion. Llave foránea: IDCliente.
CREATE TABLE Inversion (
    CódigoInversion INT PRIMARY KEY,
    monto DECIMAL(12,2) NOT NULL,
    plazo INT NOT NULL,
    GAT DECIMAL(5,2) NOT NULL,
    -- Con CHECK restringuimos los 3 posibles tipos de inversiones.
    EsUna VARCHAR(30) CHECK (EsUna IN ('Deuda Pública', 'Bolsa', 'Activos Banco')),
    IDCliente INT NOT NULL,
    FOREIGN KEY (IDCliente) REFERENCES Cliente(IDCliente)
);

% ▾
Mensajes
Los comandos se han completado correctamente.
```

Figura 16. Creación de la tabla Inversion. Llave primaria CódigoInversion y llave foránea IDCliente. Se incluye la restricción para solo permitir los 3 tipos de inversión válidos.

	Nombre de columna	Tipo de datos	Permitir valores NULL
PK	CódigoInversion	int	<input type="checkbox"/>
	monto	decimal(12, 2)	<input type="checkbox"/>
	plazo	int	<input type="checkbox"/>
	GAT	decimal(5, 2)	<input type="checkbox"/>
	EsUna	varchar(30)	<input checked="" type="checkbox"/>
	IDCliente	int	<input type="checkbox"/>
			<input type="checkbox"/>

Figura 17. Estructura de la tabla Inversion.

```
-- TABLA: InversionDeuda -> Clave primaria: CódigoInversion. Llave foránea: CódigoInversion.
CREATE TABLE InversionDeuda (
    CódigoInversion INT PRIMARY KEY,
    país VARCHAR(50) NOT NULL,
    moneda VARCHAR(20) NOT NULL,
    FOREIGN KEY (CódigoInversion) REFERENCES Inversion(CódigoInversion)
);

1 % ▾
Mensajes
Los comandos se han completado correctamente.
```

Figura 18. Creación de la tabla InversionDeuda. Llave primaria y foránea CódigoInversion.

	Nombre de columna	Tipo de datos	Permitir valores NULL
▶	Codigoinversion	int	<input type="checkbox"/>
	país	varchar(50)	<input type="checkbox"/>
	moneda	varchar(20)	<input type="checkbox"/>
			<input type="checkbox"/>

Figura 19. Estructura de la tabla InversionDeuda.

```
-- TABLA: InversionBolsa -> Clave primaria: CódigoInversion. Llave foránea: CódigoInversion.
CREATE TABLE InversionBolsa (
    CódigoInversion INT PRIMARY KEY,
    índiceBursatil VARCHAR(50) NOT NULL,
    nombreFondo VARCHAR(100) NOT NULL,
    FOREIGN KEY (CódigoInversion) REFERENCES Inversión(CódigoInversion)
);
1 %
Mensajes
Los comandos se han completado correctamente.
```

Figura 20. Creación de la tabla InversionBolsa. Llave primaria y foránea Codigoinversion.

	Nombre de columna	Tipo de datos	Permitir valores NULL
▶	Codigoinversion	int	<input type="checkbox"/>
	índiceBursatil	varchar(50)	<input type="checkbox"/>
	nombreFondo	varchar(100)	<input type="checkbox"/>
			<input type="checkbox"/>

Figura 21. Estructura de la tabla InversionBolsa.

```
-- TABLA: InversionActivosBanco -> Clave primaria: CódigoInversion. Llave foránea: CódigoInversion.
CREATE TABLE InversionActivosBanco (
    CódigoInversion INT PRIMARY KEY,
    FOREIGN KEY (CódigoInversion) REFERENCES Inversión(CódigoInversion)
);
1 %
Mensajes
Los comandos se han completado correctamente.
```

Figura 22. Creación de la tabla InversionActivosBanco. Llave primaria y foránea Codigoinversion.

```
-- TABLA: Transaccion -> Clave primaria: IDTransaccion. Llave foránea: CódigoInversión.  
CREATE TABLE Transaccion (  
    IDTransaccion INT PRIMARY KEY,  
    tipo VARCHAR(10) CHECK (tipo IN ('Compra', 'Venta')),  
    nombreBanco VARCHAR(100) NOT NULL,  
    fecha DATE NOT NULL,  
    hora TIME NOT NULL,  
    CódigoInversión INT NOT NULL,  
    FOREIGN KEY (CódigoInversión) REFERENCES Inversión(CódigoInversión)  
);
```

1 % ▾

Mensajes

Los comandos se han completado correctamente.

Figura 23. Creación de la tabla Tramsaccion. Llave primaria IDTransaccion y llave foránea CódigoInversión.

```
-- TABLA: Auditor -> Clave primaria: IDAuditor.  
CREATE TABLE Auditor (  
    IDAuditor INT PRIMARY KEY,  
    cedulaProfesional VARCHAR(20) UNIQUE NOT NULL, -- UNIQUE para que podamos evitar duplicados en esta columna.  
    nombres VARCHAR(100) NOT NULL,  
    apellidoPaterno VARCHAR(50) NOT NULL,  
    apellidoMaterno VARCHAR(50)  
);
```

1 % ▾

Mensajes

Los comandos se han completado correctamente.

Figura 24. Creación de la tabla Auditor. Llave primaria IDAuditor.

	Nombre de columna	Tipo de datos	Permitir valores NULL
PK	IDAuditor	int	<input type="checkbox"/>
	cedulaProfesional	varchar(20)	<input type="checkbox"/>
	nombres	varchar(100)	<input type="checkbox"/>
	apellidoPaterno	varchar(50)	<input type="checkbox"/>
	apellidoMaterno	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 25. Estructura de la tabla Auditor.

## Extra:

The screenshot shows the SQL Server Management Studio interface. At the top, there is a code editor window containing a SQL script. The script creates a view named 'VistaClientesCuentas' and inserts sample data into tables 'Cliente', 'Sucursal', and 'Cuenta'. Below the code editor is a results pane titled 'Resultados' which displays a single row of data from the query.

```
CREATE VIEW VistaClientesCuentas AS
SELECT
    c.IDCliente,
    CONCAT(c.nombres, ' ', c.apellidoPaterno, ' ', ISNULL(c.apellidoMaterno, '')) AS nombreCompleto,
    cu.NumeroCuenta,
    cu.EsUna AS tipoCuenta,
    cu.saldoInicial
FROM Cliente c
JOIN Cuenta cu ON c.IDCliente = cu.IDCliente;

-- EJEMPLO DE PRUEBA para la vista
-- Insertamos una sucursal para poder satisfacer la restricción de clave foránea en Cliente
INSERT INTO Sucursal (IDSucursal, ubicacion, añoApertura)
VALUES (1, 'CDMX', 2020);

-- Insermos un cliente de prueba
INSERT INTO Cliente (IDCliente, nombres, apellidoPaterno, apellidoMaterno, direccion, IDSucursal)
VALUES (1, 'Ana', 'Hernández', 'Lopez', 'Calle Falsa 123', 1);

-- Insertamos para ese cliente, una cuenta de ahorro
INSERT INTO Cuenta (NumeroCuenta, EsUna, saldoInicial, IDCiente)
VALUES (100, 'Ahorro', 5000.00, 1);

-- Consulta para probar la vista
SELECT * FROM VistaClientesCuentas;
```

IDCliente	nombreCompleto	NumeroCuenta	tipoCuenta	saldoinicial
1	Ana Hernández Lopez	100	Ahorro	5000.00

Extra. Vista creada para mostrar clientes con sus cuentas y tipos asociados, como parte del análisis extendido.

La vista VistaClientesCuentas la creamos como parte de los puntos extra del proyecto. Esta vista muestra el identificador del cliente, su nombre completo (concatenando nombre y apellidos), el número de cuenta asociada, el tipo de cuenta y el saldo inicial.

Esta vista facilitaría la consulta consolidada de datos relacionados sin tener que realizar un JOIN manual cada vez. Esto resulta útil para reportes o visualización rápida de relaciones clave en la base de datos.

Para probar esta vista insertamos datos ficticios de una sucursal, un cliente y una cuenta asociada, que nos permitió verificar que se ejecutaría correctamente la consulta.

## Ejecución del Script DML

En esta sección presentamos las evidencias de ejecución del archivo DML\_<Iniciales>.sql, el cual tiene como propósito insertar, actualizar y eliminar datos en las tablas previamente creadas en el sistema de gestión bancaria.

El script incluye:

- Inserciones (INSERT INTO) para poblar las tablas con datos de ejemplo coherentes con las relaciones y restricciones definidas en el modelo.
- Actualizaciones (UPDATE) para modificar registros específicos.
- Eliminaciones (DELETE) de datos controladas para verificar la integridad referencial y las restricciones impuestas por las claves foráneas.

## Capturas por script que muestran la ejecución.

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)          DDL_GMJS_EYJS.sql...UARDO\eduji (83)          DML_GMJS_EYJS.sql...UARDO\eduji (82)*  ✎ X
1  -- Autores: García Monjaraz Jessica Stephanie
2  -- Eduardo Yael Jiménez Sánchez.
3  -- Fecha de entrega: 18/05/2025
4  -- Curso Prebecario: Base de Datos 25-1
5
6
7  -- Insertar sucursales
8  INSERT INTO Sucursal VALUES
9  (1, 'Ciudad de México', 2010), (2, 'Guadalajara', 2012), (3, 'Monterrey', 2015),
10 (4, 'Puebla', 2013), (5, 'Querétaro', 2018), (6, 'León', 2017), (7, 'Toluca', 2011),
11 (8, 'Mérida', 2016), (9, 'Cancún', 2020), (10, 'Tijuana', 2014);
12
13 select * from Sucursal;
14

```

Results

IDSucursal	ubicacion	añoApertura
1	Ciudad de México	2010
2	Guadalajara	2012
3	Monterrey	2015
4	Puebla	2013
5	Querétaro	2018
6	León	2017
7	Toluca	2011
8	Mérida	2016
9	Cancún	2020
10	Tijuana	2014

Query executed successfully.

Inserción 1. Insertar sucursales con sus respectivas ubicaciones y años de apertura

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)          DDL_GMJS_EYJS.sql...UARDO\eduji (83)          DML_GMJS_EYJS.sql...UARDO\eduji (82)*  ✎ X
15
16  -- Insertar empleados
17  INSERT INTO Empleado VALUES
18  (101, 'Luis', 'Ramírez', 'López', '2020-01-10', NULL, 1),
19  (102, 'Ana', 'Gómez', 'Sánchez', '2019-03-15', 101, 2),
20  (103, 'Carlos', 'Hernández', 'Pérez', '2018-06-20', 101, 3),
21  (104, 'María', 'Martínez', 'Ruiz', '2021-08-01', 102, 4),
22  (105, 'José', 'Díaz', 'García', '2017-09-12', NULL, 5),
23  (106, 'Sofía', 'Torres', 'Mendoza', '2022-02-22', 101, 6),
24  (107, 'Pedro', 'Morales', 'Flores', '2016-07-03', 105, 7),
25  (108, 'Lucía', 'Rojas', 'Castillo', '2015-11-17', 103, 8),
26  (109, 'Miguel', 'Ortíz', 'Navarro', '2020-04-09', 104, 9),
27  (110, 'Elena', 'Vargas', 'Silva', '2019-12-30', NULL, 10);
28

```

Results

IDEmpleado	nombres	apellidoPaterno	apellidoMaterno	fechaContratacion	IDJefe	IDSucursal
101	Luis	Ramírez	López	2020-01-10	NULL	1
102	Ana	Gómez	Sánchez	2019-03-15	101	2
103	Carlos	Hernández	Pérez	2018-06-20	101	3
104	María	Martínez	Ruiz	2021-08-01	102	4
105	José	Díaz	García	2017-09-12	NULL	5
106	Sofía	Torres	Mendoza	2022-02-22	101	6
107	Pedro	Morales	Flores	2016-07-03	105	7
108	Lucía	Rojas	Castillo	2015-11-17	103	8
109	Miguel	Ortíz	Navarro	2020-04-09	104	9
110	Elena	Vargas	Silva	2019-12-30	NULL	10

Query executed successfully.

Inserción 2. Insertar empleados con datos personales, fecha de contratación, jefe y sucursal asignada

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)          DQL_GMJS_EYJS.sql...UARDO\eduji (83)          DML_GMJS_EYJS.sql...UARDO\eduji (82)*  ✎ X
20 select * from Cliente;
21
22
23
24
25
26
27
28
29
30 -- Insertar clientes
31 INSERT INTO Cliente VALUES
32 (201, 'Raúl', 'Fernández', 'Romero', 'Calle 1', 1),
33 (202, 'Daniela', 'Cruz', 'Salinas', 'Calle 2', 2),
34 (203, 'Alejandro', 'Ramos', 'Ibarra', 'Calle 3', 3),
35 (204, 'Brenda', 'Aguilar', 'Nava', 'Calle 4', 4),
36 (205, 'Fernando', 'Reyes', 'Domínguez', 'Calle 5', 5),
37 (206, 'Verónica', 'Molina', 'Zapata', 'Calle 6', 6),
38 (207, 'Jorge', 'Paredes', 'Mejía', 'Calle 7', 7),
39 (208, 'Patricia', 'Luna', 'Camacho', 'Calle 8', 8),
40 (209, 'Ricardo', 'Carriollo', 'Solís', 'Calle 9', 9),
41 (210, 'Claudia', 'Peña', 'Delgado', 'Calle 10', 10);
42
43 select * from Cliente;

100 % ↴
Results  ↴ Messages
IDCliente nombre apellidoPaterno apellidoMaterno direccion IDSucursal
1 201 Raúl Fernández Romero Calle 1 1
2 202 Daniela Cruz Salinas Calle 2 2
3 203 Alejandro Ramos Ibarra Calle 3 3
4 204 Brenda Aguilar Nava Calle 4 4
5 205 Fernando Reyes Domínguez Calle 5 5
6 206 Verónica Molina Zapata Calle 6 6
7 207 Jorge Paredes Mejía Calle 7 7
8 208 Patricia Luna Camacho Calle 8 8
9 209 Ricardo Carriollo Solís Calle 9 9
10 210 Claudia Peña Delgado Calle 10 10

Query executed successfully. LAP-DE-EDUARDO\SEYSQL2022 ... LAP-DE-EDUARDO\eduji (82) ProyectoBanco 00:00:00 | 10 rows

```

*Inserción 3. Insertar clientes con datos personales, dirección y sucursal asignada*

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)          DQL_GMJS_EYJS.sql...UARDO\eduji (83)          DML_GMJS_EYJS.sql...UARDO\eduji (82)*  ✎ X
45 select * from Cliente;
46
47
48
49
50
51
52
53
54
55
56
57
58 -- Insertar cuentas
59 INSERT INTO Cuenta VALUES
60 (3001, 'Ahorro', 5000.00, 201),
61 (3002, 'Corriente', 8000.00, 202),
62 (3003, 'Ahorro', 7000.00, 203),
63 (3004, 'Corriente', 6500.00, 204),
64 (3005, 'Ahorro', 10000.00, 205),
65 (3006, 'Corriente', 9000.00, 206),
66 (3007, 'Ahorro', 4000.00, 207),
67 (3008, 'Corriente', 11000.00, 208),
68 (3009, 'Ahorro', 3000.00, 209),
69 (3010, 'Corriente', 9500.00, 210);
70
71 select * from Cuenta;

100 % ↴
Results  ↴ Messages
NúmeroCuenta Estatus saldoInicial IDCliente
1 3001 Ahorro 5000.00 201
2 3002 Corriente 8000.00 202
3 3003 Ahorro 7000.00 203
4 3004 Corriente 6500.00 204
5 3005 Ahorro 10000.00 205
6 3006 Corriente 9000.00 206
7 3007 Ahorro 4000.00 207
8 3008 Corriente 11000.00 208
9 3009 Ahorro 3000.00 209
10 3010 Corriente 9500.00 210

Query executed successfully. LAP-DE-EDUARDO\SEYSQL2022 ... LAP-DE-EDUARDO\eduji (82) ProyectoBanco 00:00:00 | 10 rows

```

*Inserción 4. Insertar cuentas de ahorro y corriente asociadas a clientes con su saldo inicial*

```

52   (3000, 'Corriente', 5000.00, 200),
53   (3008, 'Ahorro', 4000.00, 207),
54   (3008, 'Corriente', 11000.00, 208),
55   (3009, 'Ahorro', 3000.00, 209),
56   (3010, 'Corriente', 9500.00, 210);
57
58 select * from Cuenta;
59
60 -- Insertar cuentas ahorro
61 INSERT INTO CuentaAhorro VALUES
62 (3001, 2.5, 12), (3003, 3.0, 24), (3005, 2.0, 6),
63 (3007, 1.8, 12), (3009, 2.2, 18);
64
65 select * from CuentaAhorro;
66
67 -- Insertar cuentas corriente

```

Results

	NumeroCuenta	interes	duracionPlazo
1	3001	2.50	12
2	3003	3.00	24
3	3005	2.00	6
4	3007	1.80	12
5	3009	2.20	18

Query executed successfully.

Inserción 5. Insertar cuentas de ahorro con su interés y duración del plazo asociado a cada cuenta

```

64
65 select * from CuentaAhorro;
66
67 -- Insertar cuentas corriente
68 INSERT INTO CuentaCorriente VALUES
69 (3002, 'CHQ-001', 1500.00, 102),
70 (3004, 'CHQ-002', 2000.00, 104),
71 (3006, 'CHQ-003', 1000.00, 105),
72 (3008, 'CHQ-004', 2500.00, 106),
73 (3010, 'CHQ-005', 3000.00, 103);
74
75 select * from CuentaCorriente;
76
77 -- Insertar préstamos
78 INSERT INTO Prestamo VALUES
79 (DATE '2023-01-10', 1, 10000, 10, 12, 201)

```

Results

	NumeroCuenta	numeroChequera	saldoMin	IDAsesor
1	3002	CHQ-001	1500.00	102
2	3004	CHQ-002	2000.00	104
3	3006	CHQ-003	1000.00	105
4	3008	CHQ-004	2500.00	106
5	3010	CHQ-005	3000.00	103

Query executed successfully.

Inserción 6. Insertar cuentas corrientes con número de chequera, saldo mínimo y asesor asignado

```

DDL_GMIS_EYJS_(0)_.UARDO\eduj (65)* DDL_GMIS_EYJS.sql..UARDO\eduj (83) DML_GMIS_EYJS.sql..UARDO\eduj (82)*
76
77    -- Insertar préstamos
78    INSERT INTO Prestamo VALUES
79    ('4001', '2023-01-10', 1, 10000.00, 12, 201),
80    ('4002', '2023-02-15', 0, 15000.00, 24, 202),
81    ('4003', '2023-03-20', 1, 8000.00, 6, 203),
82    ('4004', '2023-04-25', 0, 12000.00, 12, 204),
83    ('4005', '2023-05-30', 1, 9500.00, 6, 205),
84    ('4006', '2023-06-05', 0, 7000.00, 12, 206),
85    ('4007', '2023-07-10', 1, 6000.00, 6, 207),
86    ('4008', '2023-08-15', 0, 11000.00, 24, 208),
87    ('4009', '2023-09-20', 1, 13000.00, 12, 209),
88    ('4010', '2023-10-25', 0, 10500.00, 6, 210);
89
90    select * from Prestamo;

```

Results

IDPrestamo	fechaSolicitud	sIPrimerPrestamo	montoTotal	plazoMeses	IDCliente	
1	4001	2023-01-10	1	10000.00	12	201
2	4002	2023-02-15	0	15000.00	24	202
3	4003	2023-03-20	1	8000.00	6	203
4	4004	2023-04-25	0	12000.00	12	204
5	4005	2023-05-30	1	9500.00	6	205
6	4006	2023-06-05	0	7000.00	12	206
7	4007	2023-07-10	1	6000.00	6	207
8	4008	2023-08-15	0	11000.00	24	208
9	4009	2023-09-20	1	13000.00	12	209
10	4010	2023-10-25	0	10500.00	6	210

Message

Query executed successfully.

Inserción 7. Insertar préstamos con su fecha de solicitud, monto, plazo, si es primer préstamo y cliente asociado

```

DDL_GMIS_EYJS_(0)_.UARDO\eduj (65)* DDL_GMIS_EYJS.sql..UARDO\eduj (83) DML_GMIS_EYJS.sql..UARDO\eduj (82)*
87    ('4010', '2023-10-25', 0, 10500.00, 6, 210);
88
89    select * from Prestamo;
90
91    -- Insertar pagos
92    INSERT INTO Pago VALUES
93    ('501', 1, 1000.00, 4001), ('502', 2, 1000.00, 4001), ('503', 1, 1250.00, 4002),
94    ('504', 1, 800.00, 4003), ('505', 1, 1200.00, 4004), ('506', 2, 1200.00, 4004),
95    ('507', 1, 950.00, 4005), ('508', 1, 700.00, 4006), ('509', 1, 600.00, 4007),
96    ('510', 1, 1100.00, 4008);
97
98    select * from pago;
99
100   -- Insertar inversiones
101   INSERT INTO Inversion VALUES
102

```

Results

IDPago	numeroPago	montoPago	IDPrestamo
501	1	1000.00	4001
502	2	1000.00	4001
503	1	1250.00	4002
504	1	800.00	4003
505	1	1200.00	4004
506	2	1200.00	4004
507	1	950.00	4005
508	1	700.00	4006
509	1	600.00	4007
510	1	1100.00	4008

Message

Query executed successfully.

Inserción 8. Insertar pagos de préstamos con número de pago, monto y préstamo asociado

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DQL_GMJS_EYJS.sql...UARDO\eduji (83) DML_GMJS_EYJS.sql...UARDO\eduji (82)*

100
101 -- Insertar inversiones
102 INSERT INTO Inversion VALUES
103 (6001, 50000.00, 12, 5.5, 'Deuda Pública', 201),
104 (6002, 40000.00, 6, 4.0, 'Bolsa', 202),
105 (6003, 70000.00, 24, 6.2, 'Activos Banco', 203),
106 (6004, 30000.00, 12, 5.1, 'Deuda Pública', 204),
107 (6005, 60000.00, 6, 4.5, 'Bolsa', 205),
108 (6006, 45000.00, 18, 5.9, 'Activos Banco', 206),
109 (6007, 52000.00, 12, 6.0, 'Bolsa', 207),
110 (6008, 48000.00, 6, 4.7, 'Deuda Pública', 208),
111 (6009, 55000.00, 12, 5.3, 'Activos Banco', 209),
112 (6010, 39000.00, 6, 4.8, 'Bolsa', 210);
113
114 select * from Inversion;

```

Results Messages

Codigoinversion	monto	plazo	GAT	EstUna	IDCliente
6001	50000.00	12	5.5	Deuda Pública	201
6002	40000.00	6	4.0	Bolsa	202
6003	70000.00	24	6.2	Activos Banco	203
6004	30000.00	12	5.1	Deuda Pública	204
6005	60000.00	6	4.5	Bolsa	205
6006	45000.00	18	5.9	Activos Banco	206
6007	52000.00	12	6.0	Bolsa	207
6008	48000.00	6	4.7	Deuda Pública	208
6009	55000.00	12	5.3	Activos Banco	209
6010	39000.00	6	4.8	Bolsa	210

Query executed successfully. LAP-DE-EDUARDO\JSEYSQL2022 ... LAP-DE-EDUARDO\eduji (82) ProyectoBanco 00:00:00 | 10 rows

Inserción 9. Insertar inversiones con monto, plazo, GAT, tipo de inversión y cliente asociado

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DQL_GMJS_EYJS.sql...UARDO\eduji (83) DML_GMJS_EYJS.sql...UARDO\eduji (82)*

110 (6003, 55000.00, 12, 5.5, 'Activos Banco', 209),
111 (6010, 39000.00, 6, 4.8, 'Bolsa', 210);
112
113
114 select * from Inversion;
115
116 -- Inversiones específicas
117 INSERT INTO InversionDeuda VALUES
118 (6001, 'México', 'MXN'), (6004, 'EE.UU.', 'USD'), (6008, 'Brasil', 'BRL');
119
120 select * from InversionDeuda;
121
122 INSERT INTO InversionBolsa VALUES
123 (6002, 'S&P 500', 'Fondo Alfa'), (6005, 'NASDAQ', 'Fondo Beta'), (6007, 'Bovespa', 'Fondo Gamma'), (6010, 'IPC', 'Fondo Delta');
124
125 INSERT INTO InversionActivosBanco VALUES
126 (6003), (6006), (6009);

```

Results Messages

Codigoinversion	país	moneda
6003	México	MXN
6004	EE.UU.	USD
6008	Brasil	BRL

Query executed successfully. LAP-DE-EDUARDO\JSEYSQL2022 ... LAP-DE-EDUARDO\eduji (82) ProyectoBanco 00:00:00 | 3 rows

Inserción 10. Insertar inversiones de deuda con país emisor y moneda correspondiente

```

DDL_GMIS_EViS_0..UARDO(edui (65))          DQL_GMIS_EViS.sql..UARDO(edui (83))      DML_GMIS_EViS.sql..UARDO(edui (82)) * ×
118  INSERT INTO InversionDeuda VALUES
119  ((6001, 'México', 'MXN'), (6004, 'EE.UU.', 'USD'), (6008, 'Brasil', 'BRL'));
120  select * from InversionDeuda;
121
122  INSERT INTO InversionBolsa VALUES
123  ((6002, 'S&P 500', 'Fondo Alpha'), (6005, 'NASDAQ', 'Fondo Beta'), (6007, 'Bovespa', 'Fondo Gamma'), (6010, 'IPC', 'Fondo Delta'));
124  select * from InversionBolsa;
125
126  INSERT INTO InversionActivosBanco VALUES
127  ((6003), (6006), (6009));
128
129  select * from InversionActivosBanco;
130
131  -- Transacciones
132

```

Results Messages

Codigoinversion
6003
6006
6009

Query executed successfully. LAP-DE-EDUARDO\SEYSQ2022... LAP-DE-EDUARDO\edui (82) ProyectoBanco 00:00:00 3 rows

### Inserción 11. Insertar claves de inversiones correspondientes a activos del banco

```

DDL_GMIS_EViS_0..UARDO(edui (65)) * × DQL_GMIS_EViS.sql..UARDO(edui (83))      DML_GMIS_EViS.sql..UARDO(edui (82)) * ×
131
132  -- Transacciones
133  INSERT INTO Transaccion VALUES
134  ((7001, 'Compra', 'Bancomer', '2024-01-01', '10:00:00', 6001),
135  (7002, 'Venta', 'Banorte', '2024-01-02', '11:00:00', 6002),
136  (7003, 'Compra', 'Santander', '2024-01-03', '12:00:00', 6003),
137  (7004, 'Venta', 'HSBC', '2024-01-04', '13:00:00', 6004),
138  (7005, 'Compra', 'Scotiabank', '2024-01-05', '14:00:00', 6005),
139  (7006, 'Venta', 'Banamex', '2024-01-06', '15:00:00', 6006),
140  (7007, 'Compra', 'BBVA', '2024-01-07', '16:00:00', 6007),
141  (7008, 'Venta', 'Banregio', '2024-01-08', '17:00:00', 6008),
142  (7009, 'Compra', 'Banco Azteca', '2024-01-09', '18:00:00', 6009),
143  (7010, 'Venta', 'Inbursa', '2024-01-10', '19:00:00', 6010);
144
145  select * from Transaccion;

```

Results Messages

IDTransaccion	Tipo	NombreBanco	fecha	hora	Codigoinversion	
1	7001	Compra	Bancomer	2024-01-01	10:00:00	6001
2	7002	Venta	Banorte	2024-01-02	11:00:00	6002
3	7003	Compra	Santander	2024-01-03	12:00:00	6003
4	7004	Venta	HSBC	2024-01-04	13:00:00	6004
5	7005	Compra	Scotiabank	2024-01-05	14:00:00	6005
6	7006	Venta	Banamex	2024-01-06	15:00:00	6006
7	7007	Compra	BBVA	2024-01-07	16:00:00	6007
8	7008	Venta	Banregio	2024-01-08	17:00:00	6008
9	7009	Compra	Banco Azteca	2024-01-09	18:00:00	6009
10	7010	Venta	Inbursa	2024-01-10	19:00:00	6010

Query executed successfully. LAP-DE-EDUARDO\SEYSQ2022... LAP-DE-EDUARDO\edui (82) ProyectoBanco 00:00:00 10 rows

### Inserción 12. Insertar transacciones realizadas sobre cada inversión (compra o venta con banco asociado)

```

DDL_GMS_EVS_0..UARDO(ediji (65)) DDL_GMS_EVS.sql..UARDO(ediji (83)) DML_GMS_EVS.sql..UARDO(ediji (82)) + ×
145
146
147 -- Auditores
148 INSERT INTO Auditor VALUES
149 ,(801, '123456', 'Gabriel', 'Santos', 'López'),
150 ,(802, '789012', 'Norma', 'Vega', 'Moreno'),
151 ,(803, '345678', 'Andrés', 'León', 'Torres'),
152 ,(804, '901234', 'Rocío', 'Palma', 'Díaz'),
153 ,(805, '567890', 'Héctor', 'Mejía', 'Fuentes'),
154 ,(806, '112233', 'Laura', 'Silva', 'Campos'),
155 ,(807, '445566', 'Esteban', 'Cano', 'Ortiz'),
156 ,(808, '778899', 'Paola', 'Méndez', 'Lara'),
157 ,(809, '990011', 'Iván', 'Guzmán', 'Estrada'),
158 ,(810, '223344', 'Carmen', 'Salas', 'Luna');
159
160 select * from Auditor;
161
162
163
164
165
166
167
168
169
170
171
172

```

Results Messages

IDAuditor	cedulaProfesional	nombres	apellidoPaterno	apellidoMaterno
1	123456	Gabriel	Santos	López
2	789012	Norma	Vega	Moreno
3	345678	Andrés	León	Torres
4	901234	Rocío	Palma	Díaz
5	567890	Héctor	Mejía	Fuentes
6	112233	Laura	Silva	Campos
7	445566	Esteban	Cano	Ortiz
8	778899	Paola	Méndez	Lara
9	990011	Iván	Guzmán	Estrada
10	223344	Carmen	Salas	Luna

Query executed successfully. LAP-DE-EDUARDO\edujeysql2022... LAP-DE-EDUARDO\ediji (82) ProyectoBanco 00:00:00 10 rows

Inserción 13. Insertar los datos de los auditores con su respectiva cédula profesional y nombre completo

```

DDL_GMS_EVS_0..UARDO(ediji (65)) + × DDL_GMS_EVS.sql..UARDO(ediji (83)) DML_GMS_EVS.sql..UARDO(ediji (82)) + ×
153
154
155
156
157
158 ,(810, '223344', 'Carmen', 'Salas', 'Luna');
159
160 select * from Auditor;
161
162 -- Auditorias
163 INSERT INTO Auditoria VALUES
164 ,(901, 6001, 801), (902, 6002, 802), (903, 6003, 803), (904, 6004, 804), (905, 6005, 805),
165 ,(906, 6006, 806), (907, 6007, 807), (908, 6008, 808), (909, 6009, 809), (910, 6010, 810);
166
167 select * from Auditoria;
168
169 -- 1. UPDATE: Cambiar la ciudad de la sucursal 9
170 UPDATE Sucursal SET ciudad = 'Playa del Carmen' WHERE idSucursal = 9;
171
172 -- 2. UPDATE: Aumentar el saldo de la cuenta 3005 en 2000

```

Results Messages

IDAuditorias	Codigoinversion	IDAuditor
1	6001	801
2	6002	802
3	6003	803
4	6004	804
5	6005	805
6	6006	806
7	6007	807
8	6008	808
9	6009	809
10	6010	810

Query executed successfully. LAP-DE-EDUARDO\edujeysql2022... LAP-DE-EDUARDO\ediji (82) ProyectoBanco 00:00:00 10 rows

Selección. Seleccionamos desde auditoria.

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DQL_GMJS_EYJS.sql...UARDO\eduji (83) DML_GMJS_EYJS.sql...UARDO\eduji (82)* ×
166
167 select * from Auditoria;
168
169 -- 1. UPDATE: Cambiar la ciudad de la sucursal 9
170 UPDATE Sucursal SET ubicacion = 'Playa del Carmen' WHERE idSucursal = 9;
171
172 select * from Sucursal;
173

```

Results

IDSucursal	ubicacion	añoApertura
1	Ciudad de México	2010
2	Guadalajara	2012
3	Monterrey	2015
4	Puebla	2013
5	Querétaro	2018
6	León	2017
7	Toluca	2011
8	Mérida	2016
9	Playa del Carmen	2020
10	Tijuana	2014

Query executed successfully.

Actualización 1. Se modificó la ciudad de la sucursal con ID 9, cambiando su ubicación de \*Cancún\* a \*Playa del Carmen\*.

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* × DQL_GMJS_EYJS.sql...UARDO\eduji (83) DML_GMJS_EYJS.sql...UARDO\eduji (82)* ×
172
173 C:\Users\eduji\Downloads\38833FF26BA1D.UnigramPreview_g9c9v27vpyspw!App\DDL_GMJS_EYJS_(0).sql - LAP-DE-EDUARDO\SEYSQL2022.ProyectoBanco (LAP-DE-EDUARD
174
175 -- 2. UPDATE: Aumentar el saldo de la cuenta 3005 en 2000
176 UPDATE Cuenta SET saldo = saldo + 2000 WHERE idCuenta = 3005;
177
178
179 -- 3. UPDATE: Cambiar el nombre del cliente 207

```

Results

NumerоСuenta	EslUna	saldoInicial	IDCliente
1	Ahorro	5000.00	201
2	Corriente	8000.00	202
3	Ahorro	7000.00	203
4	Corriente	6500.00	204
5	Ahorro	10000.00	205
6	Corriente	9000.00	206
7	Ahorro	4000.00	207
8	Corriente	11000.00	208
9	Ahorro	3000.00	209
10	Corriente	9500.00	210

Query executed successfully.

Actualización 2. Se incrementó el saldo de la cuenta número \*\*3005\*\* en \*\*2000 unidades monetarias\*\*, pasando de \*\*10,000.00\*\* a \*\*12,000.00\*\*.

DDL\_GMIS\_EYJS\_(0)...UARDO\eduji (65)\* DQL\_GMIS\_EYJS.sql...UARDO\eduji (83) DML\_GMIS\_EYJS.sql...UARDO\eduji (82)\*

```

179 -- 3. UPDATE: Cambiar el nombre del cliente 207
180 UPDATE Cliente SET nombres = 'Joaquín' WHERE idCliente = 207;
181
182 select * from cliente;
183
184 -- 4. DELETE: Eliminar el préstamo con ID 4007
185 DELETE FROM Prestamo WHERE idPrestamo = 4007;
186
187
188
189
190
191
192
193
194
195

```

Results Messages

IDCliente	nombres	apellidoPaterno	apellidoMaterno	direccion	IDSucursal	
1	201	Raúl	Fernández	Romero	Calle 1	1
2	202	Daniel	Cruz	Salinas	Calle 2	2
3	203	Alejandro	Ramos	Ibarra	Calle 3	3
4	204	Brenda	Aguilar	Nava	Calle 4	4
5	205	Fernando	Reyes	Dominguez	Calle 5	5
6	206	Veronica	Molina	Zapata	Calle 6	6
7	207	Joaquín	Paredes	Mejía	Calle 7	7
8	208	Patricia	Luna	Camacho	Calle 8	8
9	209	Ricardo	Carillo	Solis	Calle 9	9
10	210	Claudia	Peña	Delgado	Calle 10	10

Query executed successfully. LAP-DE-EDUARDO\SEYSQ2022 ... | LAP-DE-EDUARDO\eduji (82) | ProyectoBanco | 00:00:00 | 10 rows

Actualización 3. Se modificó exitosamente el nombre del cliente con ID \*\*207\*\*, cambiando de \*\*Jorge\*\* a \*\*Joaquín\*\*, como se refleja en la tabla 'Cliente'.

DDL\_GMIS\_EYJS\_(0)...UARDO\eduji (65)\* DQL\_GMIS\_EYJS.sql...UARDO\eduji (83) DML\_GMIS\_EYJS.sql...UARDO\eduji (82)\*

```

183
184 -- 4. DELETE: Eliminar el préstamo con ID 4007
185 -- Verifica si hay pagos relacionados al préstamo
186 DELETE FROM pago WHERE idPrestamo = 4007;
187
188 -- Luego elimina el préstamo
189 DELETE FROM Prestamo WHERE idPrestamo = 4007;
190
191 -- Confirmar que se eliminó
192 SELECT * FROM Prestamo;
193
194 -- 5. DELETE: Eliminar la inversión 6008
195 DELETE FROM Inversion WHERE CodigoInversion = 6008;
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210

```

Results Messages

IDPrestamo	fechaSolicitud	sPrimerPrestamo	montoTotal	pizosMeses	IDCliente
1	2023-01-10	1	10000.00	12	201
2	2023-02-15	0	15000.00	24	202
3	2023-03-20	1	8000.00	6	203
4	2023-04-25	0	12000.00	12	204
5	2023-05-30	1	9500.00	6	205
6	2023-06-05	0	7000.00	12	206
7	2023-08-15	0	11000.00	24	208
8	2023-09-20	1	13000.00	12	209
9	2023-10-25	0	10500.00	6	210

Query executed successfully. LAP-DE-EDUARDO\SEYSQ2022 ... | LAP-DE-EDUARDO\eduji (82) | ProyectoBanco | 00:00:00 | 9 rows

Actualización 4. Se eliminó correctamente el préstamo con ID \*\*4007\*\*.

DDL\_GMIS\_EYJS\_(0)...UARDO\eduji (65)\* DQL\_GMIS\_EYJS.sql...UARDO\eduji (83) DML\_GMIS\_EYJS.sql...UARDO\eduji (82)\*

```

193
194      -- 5. DELETE: Eliminar la inversión 6008
195
196      --   Eliminar registros de Transacción ligados a la inversión
197      DELETE FROM Transaccion WHERE CodigoInversion = 6008;
198
199      --   Eliminar registros de InversionDeuda ligados a la inversión
200      DELETE FROM InversionDeuda WHERE CodigoInversion = 6008;
201
202      --   Eliminar registros de Auditoría ligados a la inversión
203      DELETE FROM Auditoria WHERE CodigoInversion = 6008;
204
205      --   Eliminar registros de Inversión (ya sin dependencias)
206      DELETE FROM Inversion WHERE CodigoInversion = 6008;
207
208      -- Verificación opcional
209      SELECT * FROM Inversion;
210
211

```

Results Messages

CodigoInversion	monto	plazo	QAT	EstInva	IDCliente
6001	50000.00	12	5.50	Deuda Pública	201
6002	40000.00	6	4.00	Bolse	202
6003	70000.00	24	6.20	Activos Banco	203
6004	30000.00	12	5.10	Deuda Pública	204
6005	60000.00	6	4.50	Bolse	205
6006	45000.00	18	5.90	Activos Banco	206
6007	52000.00	12	6.00	Bolse	207
6008	55000.00	12	5.30	Activos Banco	209
6010	39000.00	6	4.80	Bolse	210

Query executed successfully. LAP-DE-EDUARDO\SEYSQ2022 ... | LAP-DE-EDUARDO\eduji (82) | ProyectoBanco | 00:00:00 | 9 rows

Actualización 5. Se eliminó correctamente la inversión con código 6008, incluyendo sus registros relacionados en las tablas Transacción, InversionDeuda y Auditoría, como se muestra en la tabla Inversión.

DDL\_GMIS\_EYJS\_(0)...UARDO\eduji (65)\* DQL\_GMIS\_EYJS.sql...UARDO\eduji (83) DML\_GMIS\_EYJS.sql...UARDO\eduji (82)\*

```

208      -- Verificación opcional
209      SELECT * FROM Inversion;
210
211
212      -- 6. DELETE: Eliminar al auditor con ID 810
213      --   Eliminar registros de Auditoría ligados al auditor
214      DELETE FROM Auditoria WHERE idAuditor = 810;
215
216      --   Eliminar registro del auditor
217      DELETE FROM Auditor WHERE idAuditor = 810;
218
219      -- Verificación opcional
220      SELECT * FROM Auditor ;
221
222      -- 7. RENAME: Renombrar la tabla Cliente a ClienteBanco
223      ALTER TABLE Cliente RENAME TO ClienteBanco;
224
225      -- 8. RENAME: Renombrar la columna 'saldo' de Cuenta a 'monto'
226      ALTER TABLE Cuenta RENAME COLUMN saldo TO monto;

```

Results Messages

IDAuditor	cedulaProfesional	nombres	apellidoPaterno	apellidoMaterno
801	123456	Gabriel	Santos	López
802	789012	Norma	Vega	Moreno
803	345678	Andrés	León	Torres
804	901234	Rocío	Palma	Díaz
805	567890	Héctor	Mejía	Fuentes
806	112233	Laura	Silva	Campos
807	445566	Esteban	Cano	Ortiz
808	778899	Paola	Méndez	Lara
809	990011	Iván	Guzmán	Estrada

Query executed successfully. LAP-DE-EDUARDO\SEYSQ2022 ... | LAP-DE-EDUARDO\eduji (82) | ProyectoBanco | 00:00:00 | 9 rows

Actualización 6. Se eliminó correctamente el auditor con ID 810, junto con sus registros asociados en la tabla Auditoría, como se observa en la tabla Auditor.

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)*          DQL_GMJS_EYJS.sql...UARDO\eduji (83)          DML_GMJS_EYJS.sql...UARDO\eduji (82)*  ↵ X
220 SELECT * FROM Auditor ;
221
222 -- 7. RENAME: Renombrar la tabla Cliente a ClienteBanco
223 EXEC sp_rename 'Cliente', 'ClienteBanco';
224
225 select * from clientebanco;
226
227 -- 8. RENAME: Renombrar la columna 'saldo' de Cuenta a 'monto'
228 EXEC sp_rename 'Cuenta.saldo', 'monto', 'COLUMN';
229
230 select * from Cuenta;
231
100 %

```

Results

IDCliente	nombre	apellidoPaterno	apellidoMaterno	direccion	IDSucursal
201	Raúl	Fernández	Romeo	Calle 1	1
202	Daniel	Cruz	Salinas	Calle 2	2
203	Alejandro	Ramos	Ibarra	Calle 3	3
204	Brenda	Aguilar	Nava	Calle 4	4
205	Fernando	Reyes	Dominguez	Calle 5	5
206	Verónica	Molina	Zapata	Calle 6	6
207	Joaquín	Paredes	Mejía	Calle 7	7
208	Patricia	Luna	Camacho	Calle 8	8
209	Ricardo	Carillo	Solis	Calle 9	9
210	Claudia	Peña	Delgado	Calle 10	10

Query executed successfully.

Actualización 7. Se renombró exitosamente la tabla 'Cliente' como 'ClienteBanco', conservando todos los registros intactos.

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)*          DQL_GMJS_EYJS.sql...UARDO\eduji (83)          DML_GMJS_EYJS.sql...UARDO\eduji (82)*  ↵ X
223 EXEC sp_rename 'Cliente', 'ClienteBanco';
224
225 select * from clientebanco;
226
227 -- 8. RENAME: Renombrar la columna 'saldo' de Cuenta a 'monto'
228 EXEC sp_rename 'Cuenta.saldoinicial', 'monto', 'COLUMN';
229
230 select * from Cuenta;
231
100 %

```

Results

NumeroCuenta	Estatus	monto	IDCliente
3001	Ahorro	5000.00	201
3002	Corriente	8000.00	202
3003	Ahorro	7000.00	203
3004	Corriente	6500.00	204
3005	Ahorro	10000.00	205
3006	Corriente	9000.00	206
3007	Ahorro	4000.00	207
3008	Corriente	11000.00	208
3009	Ahorro	3000.00	209
3010	Corriente	9500.00	210

Query executed successfully.

Actualización 8. Se renombró exitosamente la columna 'saldoinicial' de la tabla 'Cuenta' a 'monto', manteniendo intactos los valores registrados.

```

229
230 select * from Cuenta;
231
232 -- 9. DROP: Eliminar la tabla InversionActivosBanco
233 DROP TABLE InversionActivosBanco;
234
235 select * from InversionActivosBanco;
236
237 -- 10. DROP: Eliminar la tabla Auditoria
238
239
240
241
242

```

100 %

Messages

Msg 208, Level 16, State 1, Line 238  
Invalid object name 'InversionActivosBanco'.

Completion time: 2026-06-18T17:52:10.8880447-06:00

100 %

Query completed with errors.

LAP-DE-EDUARDO\SEYSQ2022 ... LAP-DE-EDUARDO\eduji (82) ProyectoBanco 00:00:00 | 0 rows

*Actualización 9. Se eliminó exitosamente la tabla 'InversionActivosBanco', quedando confirmada su ausencia al intentar consultarla posteriormente.*

```

226
227 -- 8. RENAME: Renombrar la columna 'saldo' de Cuenta a 'monto'
228 EXEC sp_rename 'Cuenta.saldoInicial', 'monto', 'COLUMN';
229
230 select * from Cuenta;
231
232 -- 9. DROP: Eliminar la tabla InversionActivosBanco
233 DROP TABLE InversionActivosBanco;
234
235 select * from InversionActivosBanco;
236
237 -- 10. DROP: Eliminar la tabla auditor
238 DROP TABLE Auditor;
239
240 select * from auditor;
241
242

```

100 %

Messages

Msg 3701, Level 11, State 5, Line 238  
Cannot drop the table 'Auditor', because it does not exist or you do not have permission.  
Msg 208, Level 16, State 1, Line 240  
Invalid object name 'auditor'.

Completion time: 2026-06-18T17:56:27.6493047-06:00

100 %

Query completed with errors.

LAP-DE-EDUARDO\SEYSQ2022 ... LAP-DE-EDUARDO\eduji (82) ProyectoBanco 00:00:00 | 0 rows

*Actualización 10. Se eliminó correctamente la tabla 'Auditor', como lo confirma el error de consulta indicando que el objeto ya no existe.*

## Ejecución del Script DQL

En este sección presentamos las evidencias de ejecución del archivo DQL\_GMJS\_EYJS.sql, el cual contiene el conjunto de consultas SQL diseñadas para que podamos validar el modelo de datos del sistema de gestión bancaria propuesto.

Cada consulta responde a una situación realista que hemos planteado en el contexto del proyecto, aplicando cláusulas como SELECT, WHERE, ORDER BY, funciones agregadas, operadores lógicos y JOINs entre múltiples tablas. Las capturas

adjuntas demuestran que cada consulta fue ejecutada correctamente y devuelve información coherente con los datos insertados.

### Capturas por script que muestran la ejecución.

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DQL_GMJS_EYJS.sql...UARDO\eduji (83)* DML_GMJS_EYJS.sql...UARDO\eduji (82)*
1 -- Autores: García Monjaraz Jessica Stephanie
2 -- Eduardo Yael Jiménez Sánchez.
3 -- Fecha de entrega: 18/05/2025
4 -- Curso Prebecario: Base de Datos 25-1
5
6
7 -- 1. Empleados con su jefe (usando subconsulta para el nombre del jefe)**
8 SELECT e.IDEmpleado, e.nombres, e.apellidoPaterno, e.apellidoMaterno,
9      (SELECT CONCAT(j.nombres, ' ', j.apellidoPaterno)
10     FROM Empleado j WHERE j.IDEmpleado = e.IDJefe) AS nombreJefe
11    FROM Empleado e;
12
13 -- 2. Clientes con la sucursal (subconsulta para sucursal)
14 SELECT c.IDCliente, c.nombres, c.apellidoPaterno, c.apellidoMaterno,
15      (SELECT s.ubicacion FROM Sucursal s WHERE s.IDSucursal = c.IDSucursal) AS ubicacionSucursal
16    FROM clientebanco;
17
18 -- 3. Cuentas con cliente (subconsulta para nombre cliente)
19
20
21
100 %
  
```

IDEmpleado	nombres	apellidoPaterno	apellidoMaterno	nombreJefe
101	Luis	Ramirez	López	NULL
102	Ana	Gómez	Sánchez	Luis Ramirez
103	Carlos	Hernández	Pérez	Luis Ramirez
104	Maria	Martínez	Ruiz	Ana Gómez
105	José	Díaz	García	NULL
106	Sofia	Torres	Mendoza	Luis Ramirez
107	Pedro	Morales	Flores	José Díaz
108	Lucía	Rojas	Castillo	Carlos Hernández
109	Miguel	Ortíz	Navarro	Maria Martínez
110	Elena	Vargas	Silva	NULL

Query executed successfully. LAP-DE-EDUARDO\JSEYSQL2022 ... | LAP-DE-EDUARDO\eduji (83) | ProyectoBanco | 00:00:01 | 10 rows

Consulta 1. Empleados con su jefe (usando subconsulta para obtener el nombre del jefe)

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DQL_GMJS_EYJS.sql...UARDO\eduji (83)* DML_GMJS_EYJS.sql...UARDO\eduji (82)*
4 -- Curso Prebecario: Base de Datos 25-1
5
6
7 -- 1. Empleados con su jefe (usando subconsulta para el nombre del jefe)**
8 SELECT e.IDEmpleado, e.nombres, e.apellidoPaterno, e.apellidoMaterno,
9      (SELECT CONCAT(j.nombres, ' ', j.apellidoPaterno)
10     FROM Empleado j WHERE j.IDEmpleado = e.IDJefe) AS nombreJefe
11    FROM Empleado e;
12
13 -- 2. Clientes con la sucursal (subconsulta para sucursal)
14 SELECT c.IDCliente, c.nombres, c.apellidoPaterno, c.apellidoMaterno,
15      (SELECT s.ubicacion FROM Sucursal s WHERE s.IDSucursal = c.IDSucursal) AS ubicacionSucursal
16    FROM cliente c;
17
18 -- 3. Cuentas con cliente (subconsulta para nombre cliente)
19 SELECT cu.NumeroCuenta, cu.EsUna, cu.saldoInicial,
20      (SELECT cl.nombres FROM Cliente cl WHERE cl.IDCliente = cu.IDCliente) AS nombreCliente
21    FROM Cuenta cu;
100 %
  
```

IDCliente	nombres	apellidoPaterno	apellidoMaterno	ubicacionSucursal
201	Raúl	Fernández	Romero	Ciudad de México
202	Daniela	Cruz	Salinas	Guadalajara
203	Alejandro	Ramos	Ibarra	Monterrey
204	Brenda	Aguilar	Nava	Puebla
205	Fernando	Reyes	Dominguez	Querétaro
206	Verónica	Molina	Zapata	León
207	Jorge	Paredes	Mejía	Toluca
208	Patricia	Luna	Carmacho	Mérida
209	Ricardo	Carillo	Solis	Cancún
210	Claudia	Peña	Delgado	Tijuana

Query executed successfully. LAP-DE-EDUARDO\JSEYSQL2022 ... | LAP-DE-EDUARDO\eduji (83) | xd | 00:00:00 | 10 rows

Consulta 2. Clientes con la sucursal (usando subconsulta para obtener ubicación)

```
DML_GMJS_EVJS_(0)...UARDO\eduji (65)* DQL_GMJS_EVJS.sql...UARDO\eduji (83)* x DML_GMJS_EVJS.sql...UARDO\eduji (82)*

18     FROM Empleado j WHERE j.IDEmpleado = e.IDJefe) AS nombreJefe
19
20 FROM Empleado e;
21
22
23 -- 2. Clientes con la sucursal (subconsulta para sucursal)
24
25 SELECT c.IDCliente, c.nombres, c.apellidoPaterno, c.apellidoMaterno,
26       (SELECT s.ubicacion FROM Sucursal s WHERE s.IDSucursal = c.IDSucursal) AS ubicacionSucursal
27
28 FROM cliente c;
29
30
31 -- 3. Cuentas con cliente (subconsulta para nombre cliente)
32
33 SELECT cu.NumeroCuenta, cu.EsUna, cu.saldoInicial,
34       (SELECT cl.nombres FROM Cliente cl WHERE cl.IDCliente = cu.IDCliente) AS nombreCliente
35
36 FROM Cuenta cu;
37
38
39 -- 4. Préstamos > 100000 con cliente (subconsulta)
40
41 SELECT p.IDprestamo, p.fechaSolicitud, p.montoTotal, p.plazoMeses,
42       (SELECT cl.nombres FROM Cliente cl WHERE cl.IDCliente = p.IDCliente) AS nombreCliente
43
44 FROM Prestamo p
45
46 WHERE p.montoTotal > 100000;
```

Consulta 3. Cuentas con cliente (usando subconsulta para obtener el nombre del cliente)

```
DDL_GMJS_EVJS_(0)...UARDO\eduji (65)* DQL_GMJS_EVJS.sql...UARDO\eduji (83)* DML_GMJS_EVJS.sql...UARDO\eduji (82)*

16 FROM cliente c;
17
18 -- 3. Cuentas con cliente (subconsulta para nombre cliente)
19 SELECT cu.NumeroCuenta, cu.EsUna, cu.saldoInicial,
20       (SELECT cl.nombres FROM Cliente cl WHERE cl.IDCliente = cu.IDCliente) AS nombreCliente
21 FROM Cuenta cu;
22
23 -- 4. Préstamos > 100000 con cliente (subconsulta)
24 SELECT p.IDPrestamo, p.fechaSolicitud, p.montoTotal, p.plazoMeses,
25       (SELECT cl.nombres FROM Cliente cl WHERE cl.IDCliente = p.IDCliente) AS nombreCliente
26 FROM Prestamo p
27 WHERE p.montoTotal > 10000;
28
29 -- 5. Pagos de préstamo IDPrestamo = 4002
30 SELECT IDPago, numeroPago, montoPago
31 FROM Pago
32 WHERE IDPrestamo = 4002;
33

100 %
```

Result Messages

IDPrestamo	fechaSolicitud	montoTotal	plazoMeses	nombreCliente
1 4002	2023-02-15	15000.00	24	Daniela
2 4004	2023-04-25	12000.00	12	Brenda
3 4008	2023-08-15	11000.00	24	Patricia
4 4009	2023-09-20	13000.00	12	Ricardo
5 4010	2023-10-25	10500.00	6	Claudia

Query executed successfully.

Consulta 4. Préstamos mayores a 10000 con nombre del cliente (usando subconsulta)

DDL\_GMJS\_EYJS\_(0)...UARDO\eduji (65)\* DQL\_GMJS\_EYJS.sql...UARDO\eduji (83)\* DML\_GMJS\_EYJS.sql...UARDO\eduji (82)\*

```

19 SELECT cu.NumeroCuenta, cu.EsUna, cu.saldoInicial,
20       (SELECT cl.nombres FROM Cliente cl WHERE cl.IDCliente = cu.IDCliente) AS nombreCliente
21   FROM Cuenta cu;
22
23 -- 4. Préstamos > 100000 con cliente (subconsulta)
24 SELECT p.IDPrestamo, p.fechaSolicitud, p.montoTotal, p.plazoMeses,
25       (SELECT cl.nombres FROM Cliente cl WHERE cl.IDCliente = p.IDCliente) AS nombreCliente
26   FROM Prestamo p
27 WHERE p.montoTotal > 10000;
28
29 -- 5. Pagos de préstamo IDPrestamo = 4002
30 SELECT IDPago, numeroPago, montoPago
31   FROM Pago
32 WHERE IDPrestamo = 4002;
33
34
35 -- 6. Inversiones Bolsa con cliente (subconsulta para cliente)
36 SELECT i.CodigoInversion, i.monto, i.plazo, i.GAT,
37       (SELECT c.nombres FROM Cliente c WHERE c.IDCliente = i.IDCliente) AS nombreCliente
38   FROM Inversion i
39 WHERE i.EsUna = 'Bolsa';
40
41
42
43
44
45

```

100 %

Results Messages

IDPago	numeroPago	montoPago
503	1	1250.00

Query executed successfully.

Consulta 5. Pagos correspondientes al préstamo con ID = 4002

DDL\_GMJS\_EYJS\_(0)...UARDO\eduji (65)\* DQL\_GMJS\_EYJS.sql...UARDO\eduji (83)\* DML\_GMJS\_EYJS.sql...UARDO\eduji (82)\*

```

28
29 -- 5. Pagos de préstamo IDPrestamo = 4002
30 SELECT IDPago, numeroPago, montoPago
31   FROM Pago
32 WHERE IDPrestamo = 4002;
33
34
35 -- 6. Inversiones Bolsa con cliente (subconsulta para cliente)
36 SELECT i.CodigoInversion, i.monto, i.plazo, i.GAT,
37       (SELECT b indiceBursatil FROM InversionBolsa b WHERE b.CodigoInversion = i.CodigoInversion) AS indiceBursatil,
38       (SELECT b.nombreFondo FROM InversionBolsa b WHERE b.CodigoInversion = i.CodigoInversion) AS nombreFondo,
39       (SELECT c.nombres FROM Cliente c WHERE c.IDCliente = i.IDCliente) AS nombreCliente
40   FROM Inversion i
41 WHERE i.EsUna = 'Bolsa';
42
43
44
45

```

100 %

Results Messages

CodigoInversion	monto	plazo	GAT	indiceBursatil	nombreFondo	nombreCliente
6002	40000.00	6	4.00	S&P 500	Fondo Alta	Daniela
6005	60000.00	6	4.50	NASDAQ	Fondo Beta	Fernando
6007	52000.00	12	6.00	Bovespa	Fondo Gamma	Jorge
6010	39000.00	6	4.80	IPC	Fondo Delta	Claudia

Query executed successfully.

Consulta 6. Inversiones en bolsa con cliente (usando subconsulta para índice y fondo)

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DQL_GMJS_EYJS.sql...UARDO\eduji (83)* DML_GMJS_EYJS.sql...UARDO\eduji (82)*
34
35 -- 6. Inversiones Bolsa con cliente (subconsulta para cliente)
36 SELECT i.CodigoInversion, i.monto, i.plazo, i.GAT,
37   (SELECT b.IndiceBursatil FROM InversionBolsa b WHERE b.CodigoInversion = i.CodigoInversion) AS indiceBursatil,
38   (SELECT b.nombreFondo FROM InversionBolsa b WHERE b.CodigoInversion = i.CodigoInversion) AS nombreFondo,
39   (SELECT c.nombres FROM Cliente c WHERE c.IDCliente = i.IDCliente) AS nombreCliente
40
41 FROM Inversion i
42 WHERE i.EsUna = 'Bolsa';
43
44 -- 7. Contar empleados por sucursal (agrupando en empleados)
45 SELECT IDSucursal, COUNT(*) AS totalEmpleados
46 FROM Empleado
47 GROUP BY IDSucursal;
48
49 -- 8. Cuentas corrientes con asesor (subconsulta para asesor)
50 SELECT cc.NumeroCuenta, cc.numeroChequera, cc.saldoMin,
51   (SELECT e.nombres FROM Empleado e WHERE e.IDEmpleado = cc.IDAsesor) AS nombreAsesor
52
53 FROM CuentaCorriente cc;
54
55 -- 9. Transacciones con inversión (subconsulta para monto inversión)
56 SELECT t.IDTransaccion, t.tipo, t.nombreBanco, t.fecha, t.hora,
57   (SELECT i.monto FROM Inversion i WHERE i.CodigoInversion = t.CodigoInversion) AS montoInversion,
58   (SELECT i.EsUna FROM Inversion i WHERE i.CodigoInversion = t.CodigoInversion) AS tipoInversion
59
60 FROM Transaccion t;
61
62 -- 10. Auditorías con auditor (subconsulta)
63 SELECT a.IDAuditoria, a.CodigoInversion,

```

Results

IDSucursal	totalEmpleados
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

LAP-DE-EDUARDO\SEYSQ2022 ... | LAP-DE-EDUARDO\eduji (83) | xd | 00:00:00 | 10 rows

Consulta 7. Contar empleados por sucursal (agrupando por ID de sucursal)

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DQL_GMJS_EYJS.sql...UARDO\eduji (83)* DML_GMJS_EYJS.sql...UARDO\eduji (82)*
43 -- 7. Contar empleados por sucursal (agrupando en empleados)
44 SELECT IDSucursal, COUNT(*) AS totalEmpleados
45 FROM Empleado
46 GROUP BY IDSucursal;
47
48 -- 8. Cuentas corrientes con asesor (subconsulta para asesor)
49 SELECT cc.NumeroCuenta, cc.numeroChequera, cc.saldoMin,
50   (SELECT e.nombres FROM Empleado e WHERE e.IDEmpleado = cc.IDAsesor) AS nombreAsesor
51
52 FROM CuentaCorriente cc;
53
54 -- 9. Transacciones con inversión (subconsulta para monto inversión)
55 SELECT t.IDTransaccion, t.tipo, t.nombreBanco, t.fecha, t.hora,
56   (SELECT i.monto FROM Inversion i WHERE i.CodigoInversion = t.CodigoInversion) AS montoInversion,
57   (SELECT i.EsUna FROM Inversion i WHERE i.CodigoInversion = t.CodigoInversion) AS tipoInversion
58
59 -- 10. Auditorías con auditor (subconsulta)
60 SELECT a.IDAuditoria, a.CodigoInversion,

```

Results

NúmeroCuenta	numeroChequera	saldoMin	nombreAsesor
3002	CHO-001	1500.00	Ana
3004	CHO-002	2000.00	Maria
3006	CHO-003	1000.00	José
3008	CHO-004	2500.00	Sofía
3010	CHO-005	3000.00	Carlos

LAP-DE-EDUARDO\SEYSQ2022 ... | LAP-DE-EDUARDO\eduji (83) | xd | 00:00:00 | 5 rows

Consulta 8. Cuentas corrientes con asesor (usando subconsulta para obtener nombre del asesor)

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DQL_GMJS_EYJS.sql...UARDO\eduji (83)* DML_GMJS_EYJS.sql...UARDO\eduji (82)*
43 -- 7. Contar empleados por sucursal (agrupando en empleados)
44 SELECT IDSucursal, COUNT(*) AS totalEmpleados
45 FROM Empleado
46 GROUP BY IDSucursal;
47
48 -- 8. Cuentas corrientes con asesor (subconsulta para asesor)
49 SELECT cc.NumeroCuenta, cc.numeroChequera, cc.saldoMin,
50       (SELECT e.nombre FROM Empleado e WHERE e.IDEmpleado = cc.IDAsesor) AS nombreAsesor
51 FROM CuentaCorriente cc;
52
53 -- 9. Transacciones con inversión (subconsulta para monto inversión)
54 SELECT t.IDTransaccion, t.tipo, t.nombreBanco, t.fecha, t.hora,
55       (SELECT i.monto FROM Inversion i WHERE i.CodigoInversion = t.CodigoInversion) AS montoInversion,
56       (SELECT i.EsUna FROM Inversion i WHERE i.CodigoInversion = t.CodigoInversion) AS tipoInversion
57 FROM Transaccion t;
58
59 -- 10. Auditorías con auditor (subconsulta)
60 SELECT a.IDAuditoria, a.CodigoInversion,

```

100 %

Results Messages

IDTransaccion	tipo	nombreBanco	fecha	hora	montoInversion	tipoInversion
1 7001	Compra	Bancomer	2024-01-01	10:00:00.000000	50000.00	Deuda Pública
2 7002	Venta	Banorte	2024-01-02	11:00:00.000000	40000.00	Bolsa
3 7003	Compra	Santander	2024-01-03	12:00:00.000000	70000.00	Activos Banco
4 7004	Venta	HSBC	2024-01-04	13:00:00.000000	30000.00	Deuda Pública
5 7005	Compra	Scotiabank	2024-01-05	14:00:00.000000	60000.00	Bolsa
6 7006	Venta	Banamex	2024-01-06	15:00:00.000000	45000.00	Activos Banco
7 7007	Compra	BBVA	2024-01-07	16:00:00.000000	52000.00	Bolsa
8 7008	Venta	Banregio	2024-01-08	17:00:00.000000	48000.00	Deuda Pública
9 7009	Compra	Banco Azteca	2024-01-09	18:00:00.000000	55000.00	Activos Banco
10 7010	Venta	Inbursa	2024-01-10	19:00:00.000000	39000.00	Bolsa

Query executed successfully.

Consulta 9. Transacciones con inversión (usando subconsulta para monto y tipo de inversión)

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DQL_GMJS_EYJS.sql...UARDO\eduji (83)* DML_GMJS_EYJS.sql...UARDO\eduji (82)*
52
53 -- 9. Transacciones con inversión (subconsulta para monto inversión)
54 SELECT t.IDTransaccion, t.tipo, t.nombreBanco, t.fecha, t.hora,
55       (SELECT i.monto FROM Inversion i WHERE i.CodigoInversion = t.CodigoInversion) AS montoInversion,
56       (SELECT i.EsUna FROM Inversion i WHERE i.CodigoInversion = t.CodigoInversion) AS tipoInversion
57 FROM Transaccion t;
58
59 -- 10. saldo inicial promedio de las cuentas
60 SELECT AVG(saldoInicial) AS PromedioSaldoInicial
61 FROM cuenta;
62
63
64 -- 11. Clientes con más de una cuenta (solo en Cuenta, luego subconsulta para cliente)
65 SELECT IDCliente, COUNT(*) AS totalCuentas
66 FROM Cuenta
67 GROUP BY IDCliente
68 HAVING COUNT(*) > 1;
69

```

100 %

Results Messages

PromedioSaldoInicial
7300.000000

Query executed successfully.

Consulta 10. Saldo inicial promedio de las cuentas

```

DDL_GMIS_EViS_(0)...UARDO\eduij (65)* → X DQL_GMIS_EViS.sql...UARDO\eduij (83)* → X DML_GMIS_EViS.sql...UARDO\eduij (82)*

52 -- 9. Transacciones con inversión (subconsulta para monto inversión)
53 SELECT t.IDtransaccion, t.tipo, t.nombreBanco, t.fecha, t.hora,
54       (SELECT i.monto FROM Inversion i WHERE i.CodigoInversion = t.CodigoInversion) AS montoInversion,
55       (SELECT i.EsUna FROM Inversion i WHERE i.CodigoInversion = t.CodigoInversion) AS tipoInversion
56   FROM Transaccion t;
57
58
59 -- 10. saldo inicial promedio de las cuentas
60 SELECT AVG(saldoInicial) AS PromedioSaldoInicial
61   FROM cuenta;
62
63
64 -- 11. Clientes con más de una cuenta (solo en Cuenta, luego subconsulta para cliente)
65 SELECT IDCliente, COUNT(*) AS totalCuentas
66   FROM Cuenta
67  GROUP BY IDCliente
68  HAVING COUNT(*) > 1;
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Results Messages

PromedioSaldoInicial
7300.00000

Query executed successfully. LAP-DE-EDUARDO\SEYSQ2022 .. LAP-DE-EDUARDO\eduij (83) xd 00:00:00 1 rows

Consulta 11. Clientes que viven en la calle 1

```

DDL_GMIS_EViS_(0)...UARDO\eduij (65)* → X DQL_GMIS_EViS.sql...UARDO\eduij (83)* → X DML_GMIS_EViS.sql...UARDO\eduij (82)*

64 -- 11. Clientes que viven en la calle 1
65 select nombres, apellidoPaterno from Cliente
66 where direccion = 'calle 1';
67
68
69 -- 12. Préstamos que son primer préstamo (directo)
70 SELECT p.IDprestamo, p.fechaSolicitud, p.montoTotal, p.IDCliente, p.siPrimerPrestamo
71   FROM Prestamo p
72  WHERE p.siPrimerPrestamo = 1;
73
74 -- 13. Inversiones por cliente y tipo (subconsulta para nombre cliente)
75 SELECT i.CodigoInversion, i.EsUna, i.monto, i.plazo,
76       (SELECT c.nombres FROM Cliente c WHERE c.IDCliente = i.IDCliente) AS nombreCliente
77   FROM Inversion i
78  ORDER BY nombreCliente;
79
80 -- 14. Empleados sin jefe asignado
81 SELECT IDEmpleado, nombres, apellidoPaterno
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Results Messages

IDPrestamo	fechaSolicitud	montoTotal	IDCliente	siPrimerPrestamo
1 40001	2023-01-10	10000.00	201	1
2 4003	2023-03-20	8000.00	203	1
3 4005	2023-05-30	9500.00	205	1
4 4007	2023-07-10	6000.00	207	1
5 4009	2023-09-20	13000.00	209	1

Query executed successfully. LAP-DE-EDUARDO\SEYSQ2022 .. LAP-DE-EDUARDO\eduij (83) xd 00:00:00 5 rows

Consulta 12. Préstamos que son primer préstamo (consulta directa por atributo booleano)

DDL\_GMIS\_EYIS\_0\_UARDO(ediji (65)) DDL\_GMIS\_EYIS.sql\_UARDO(ediji (83)) X DML\_GMIS\_EYIS.sql\_UARDO(ediji (82))

```

67
68
69 -- 12. Préstamos que son primer préstamo (directo)
70 SELECT p.IDPrestamo, p.fechaSolicitud, p.montoTotal, p.IDCliente, p.siPrimerPrestamo
71 FROM Prestamo p
72 WHERE p.siPrimerPrestamo = 1;
73
74 -- 13. Inversiones por cliente y tipo (subconsulta para nombre cliente)
75 SELECT i.CodigoInversion, i.EsIva, i.monto, i.plazo,
76       (SELECT c.nombres FROM Cliente c WHERE c.IDCliente = i.IDCliente) AS nombreCliente
77 FROM Inversion i
78 ORDER BY nombreCliente;
79
80 -- 14. Empleados sin jefe asignado
81 SELECT E.IDempleado, E.nombres, E.apellidoPaterno
82 FROM Empleado E
83 WHERE E.IDjefe IS NULL;
84

```

Results Messages

	CodigoInversion	EstIva	monto	plazo	nombreCliente
1	6003	Activo Banco	70000.00	24	Alejandro
2	6004	Deuda Pública	30000.00	12	Brenda
3	6010	Bolsa	39000.00	6	Claudia
4	6002	Bolsa	40000.00	6	Daniela
5	6005	Bolsa	60000.00	6	Fernando
6	6007	Bolsa	52000.00	12	Jorge
7	6006	Deuda Pública	48000.00	6	Patricia
8	6001	Deuda Pública	50000.00	12	Raul
9	6009	Activo Banco	55000.00	12	Ricardo
10	6008	Activo Banco	45000.00	18	Verónica

Query executed successfully.

Consulta 13. Inversiones por cliente y tipo (usando subconsulta para obtener nombre del cliente)

```

-- TABLA: Pago -> Clave primaria: IDPago. Llave foránea: IDPrestamo.
CREATE TABLE Pago (
    IDPago INT PRIMARY KEY,
    numeroPago INT NOT NULL,
    montoPago DECIMAL(10,2) NOT NULL,
    IDPrestamo INT NOT NULL,
    FOREIGN KEY (IDPrestamo) REFERENCES Prestamo(IDPrestamo)
);

```

Mensajes

Los comandos se han completado correctamente.

Figura 14. Creación de la tabla Pago. Llave primaria IDPago y llave foránea IDPrestamos.

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DDL_GMJS_EYJS.sql...UARDO\eduji (83)* DML_GMJS_EYJS.sql...UARDO\eduji (82)*
73
74 -- 13. Inversiones por cliente y tipo (subconsulta para nombre cliente)
75 SELECT i.CodigoInversion, i.EsUna, i.monto, i.plazo,
76       (SELECT c.nombres FROM Cliente c WHERE c.IDCliente = i.IDCliente) AS nombreCliente
77   FROM Inversion i
78 ORDER BY nombreCliente;
79
80 -- 14. Empleados sin jefe asignado
81 SELECT IDEmppleado, nombres, apellidoPaterno
82   FROM Empleado
83  WHERE IDJefe IS NULL;
84
85 -- 15. Detalles de cuentas de ahorro con cliente (subconsulta para cliente)
86 SELECT ca.NumeroCuenta, ca.interes, ca.duracionPlazo,
87       (SELECT cl.nombres FROM Cliente cl
88        JOIN Cuenta cu ON cl.IDCliente = cu.IDCliente
89        WHERE cu.NumeroCuenta = ca.NumeroCuenta) AS nombreCliente
90   FROM CuentaAhorro ca;
100 %

```

Results Messages

IDEmpleado	nombres	apellidoPaterno
101	Luis	Ramirez
105	José	Diaz
110	Elena	Vargas

Query executed successfully. LAP-DE-EDUARDO\SEYSQ2022 ... LAP-DE-EDUARDO\eduji (83) | xd | 00:00:00 | 3 rows

Consulta 14. Empleados sin jefe asignado (usando condición nula sobre IDJefe)

```

DDL_GMJS_EYJS_(0)...UARDO\eduji (65)* DDL_GMJS_EYJS.sql...UARDO\eduji (83)* DML_GMJS_EYJS.sql...UARDO\eduji (82)*
76       (SELECT c.nombres FROM Cliente c WHERE c.IDCliente = i.IDCliente) AS nombreCliente
77   FROM Inversion i
78 ORDER BY nombreCliente;
79
80 -- 14. Empleados sin jefe asignado
81 SELECT IDEmppleado, nombres, apellidoPaterno
82   FROM Empleado
83  WHERE IDJefe IS NULL;
84
85 -- 15. Detalles de cuentas de ahorro con cliente (subconsulta para cliente)
86 SELECT ca.NumeroCuenta, ca.interes, ca.duracionPlazo,
87       (SELECT cl.nombres FROM Cliente cl
88        JOIN Cuenta cu ON cl.IDCliente = cu.IDCliente
89        WHERE cu.NumeroCuenta = ca.NumeroCuenta) AS nombreCliente
90   FROM CuentaAhorro ca;
91
100 %

```

Results Messages

NumeroCuenta	interes	duracionPlazo	nombreCliente
3001	250	12	Raul
3003	300	24	Alejandro
3005	200	6	Fernando
3007	180	12	Jorge
3009	220	18	Ricardo

Query executed successfully. LAP-DE-EDUARDO\SEYSQ2022 ... LAP-DE-EDUARDO\eduji (83) | xd | 00:00:00 | 5 rows

Consulta 15. Detalles de cuentas de ahorro con cliente (usando subconsulta y join para obtener nombre)

## Conclusiones:

**Jessica Stephanie Garcia Monjaraz:**

Considero que personalmente el análisis realizado fue sumamente necesario, ya que me permitió comprender mucho mejor como es que se integra por completo la estructura del caso de la gestión bancaria. A través del estudio detallado de las entidades y sus relaciones, pude identificar cómo es que se conectan entre sí los diferentes elementos del sistema, lo cual me ayudó muchísimo para el diseño lógico y físico de la base de datos.

El Modelo Entidad-Relación (MER) extendido fue especialmente útil como herramienta visual, ya que me facilitó poder identificar las llaves primarias (PK), llaves foráneas (FK) y la forma en que estas se relacionan entre las distintas entidades. Personalmente, mi forma de comprender las cosas tiene un enfoque mucho más visual, así que esta representación gráfica me ayudó a prevenir errores de integridad y a diseñar una base de datos más robusta y coherente.

Asimismo, la implementación de las sentencias SQL DML como 'INSERT', 'UPDATE', 'DELETE', 'RENAME' y 'DROP' me permitió validar el funcionamiento de la base de datos mediante la inserción de registros, la actualización de información, la eliminación de datos específicos y renombrar objetos. Estas acciones me permitieron verificar que la estructura del modelo fuera la correcta, además de me permitir tener una práctica real sobre cómo mantener y modificar una base de datos según las necesidades del sistema.

En conjunto, el trabajo con el MER extendido y el lenguaje SQL me ayudó a mejorar mis conocimientos técnicos de bases de datos, así como desarrollar una comprensión más profunda de la lógica que rige las bases de datos relacionales y su aplicación en contextos reales.

---

### **Eduardo Yael Jiménez Sanchez:**

Este proyecto me permitió aplicar de forma práctica los conocimientos adquiridos durante el curso de bases de datos, desde el análisis del caso de estudio hasta la implementación completa de una base de datos relacional en un escenario realista y funcional.

El caso del banco representó un reto interesante, ya que implicó una estructura compleja con múltiples relaciones entre entidades, restricciones y reglas de negocio. Enfrentar y resolver estos desafíos fortaleció mi pensamiento lógico, mi capacidad de análisis y mi habilidad para estructurar información de forma eficiente y coherente.

Además, el hecho de documentar cada etapa del proyecto, registrar evidencias y estructurar adecuadamente los archivos, me ayudó a desarrollar una mejor organización en el trabajo técnico. En resumen, este proyecto fue una experiencia muy enriquecedora que no solo reforzó mis conocimientos teóricos y prácticos, sino que también me preparó mejor para futuros retos relacionados con el diseño y gestión de bases de datos en entornos reales.