

Universidad Tecnológica de Panamá

Centro Regional De Veraguas



Facultad de Ingeniería de Sistemas
Computacionales

Curso: Base de Datos II

Profesor: Abdiel Kapell

PARCIAL 2

Estudiantes

Elbin Puga, Arland Barrera

2024

Contenido

1	Enunciado	4
2	Desarrollo	5
2.1	Diagramas	5
2.2	Creación de Base de Datos y Tablas	6
2.3	Creación de Triggers	9
2.4	Inserción de Registros	12
2.5	Probar Triggers	13
2.6	Ejercicios	15

Lista de figuras

2.1	Diagrama Entidad-Relación	5
2.2	Diagrama Relacional	6
2.3	Creación Base de Datos DB-Banco	6
2.4	Creación de la tabla TB-Clientes	7
2.5	Creación de la tabla TB-Prestamos	7
2.6	Creación de la tabla TB-Cuentas	7
2.7	Creación de la tabla TB-Transacciones	8
2.8	Creación de la tabla TB-Ahorro	8
2.9	Creación de la tabla TB-PlazoFijo	8
2.10	Creación de la tabla TB-ListaNegra	9
2.11	Creación Trigger TR-ValidarEstadoPrestamo	9
2.12	Creación Trigger TR-ValidarPeriodoPrestamo	10
2.13	Creación Trigger TR-ValidarTipoTransaccion	10
2.14	Creación Trigger TR-ValidarTipoCuenta	10
2.15	Creación Trigger TR-ValidarTransaccionRetiro	11
2.16	Creación Trigger TR-ValidarTransaccionDeposito	11
2.17	Creación Trigger TR-MoverAListaNegra	11
2.18	Inserción registros Clientes	12
2.19	Inserción registros Prestamos	12
2.20	Inserción registros Cuentas	12
2.21	Inserción registros Ahorro	13
2.22	Inserción registros PlazoFijo	13
2.23	Restricción Transacción PlazoFijo	13
2.24	Restricción Transacción límite en Retiro	14
2.25	Activar Trigger aumento de Depósito	14
2.26	Activar Trigger enviar a Lista Negra	14
2.27	Ejercicio Fondos Totales	15

Enunciado

La Base da Datos consiste en la gestión un banco, que opera con reserva fraccionaria, y sus clientes.

Los clientes pueden solicitar préstamos y pueden tener cuentas de dos tipos: ahorro y plazo fijo.

En las cuentas de ahorro se pueden realizar transacciones, más no en las cuentas a plazo fijo.

Las transacciones pueden ser de retiro o depósito. Las cuentas de ahorro tienen una cantidad máxima de retiro.

Si los clientes no pagan sus préstamos a tiempo, son enviados a la lista negra.

Las cuentas realizan un importe a los clientes en base a sus fondos en depósito, los cuales operan con un interés simple.

El interés de los préstamos es compuesto. Más te vale liquidar tus deudas o las deudas de liquidarán!.

Desarrollo

2.1 Diagramas

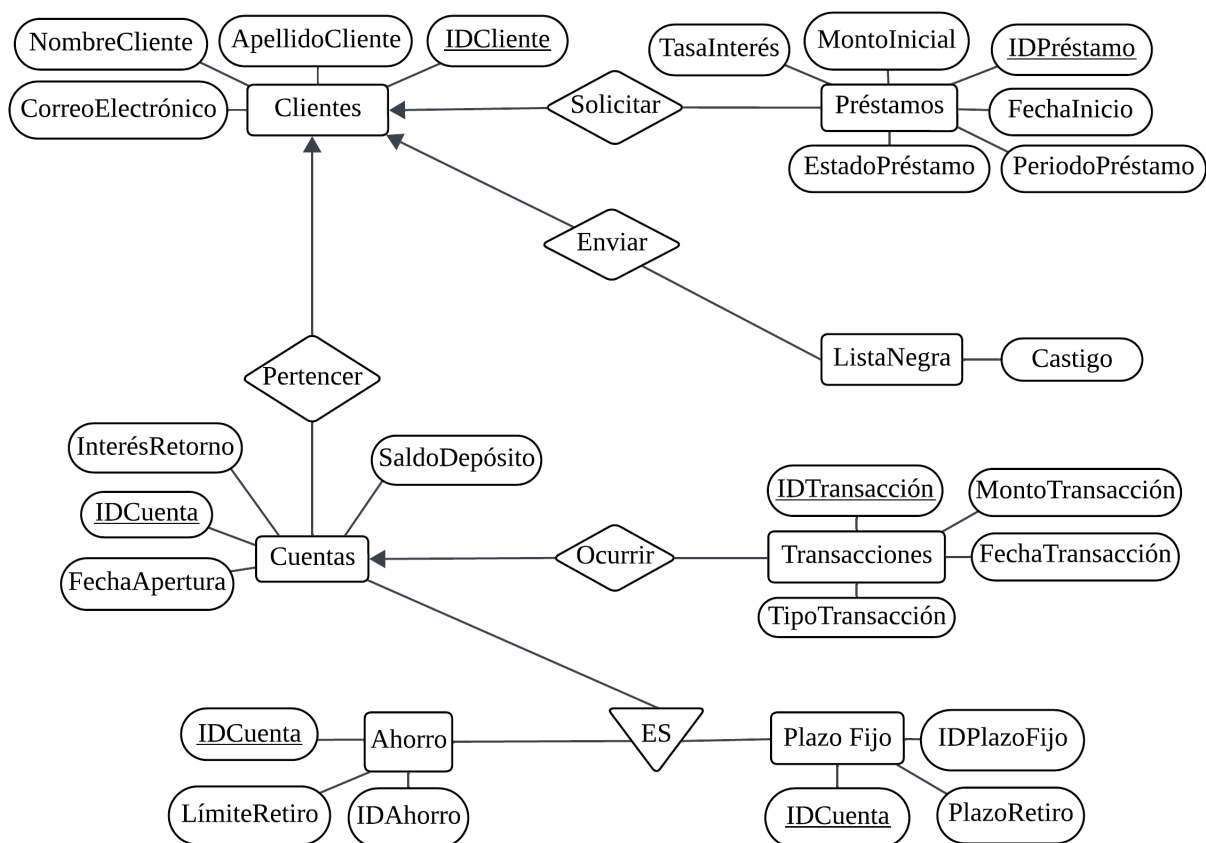


Figura 2.1: Diagrama Entidad-Relación

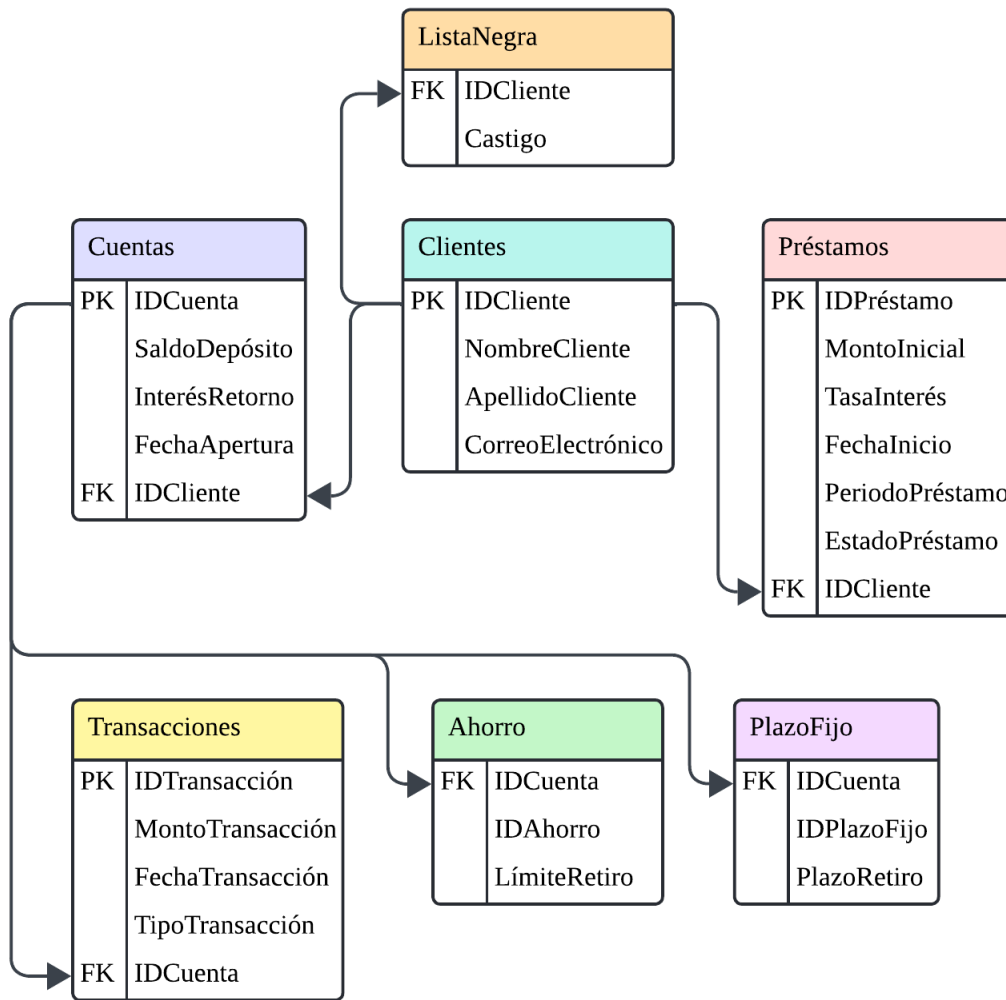


Figura 2.2: Diagrama Relacional

2.2 Creación de Base de Datos y Tablas

```

CREATE DATABASE DB_Banco
GO

```

Messages

Commands completed successfully.

Completion time: 2024-10-09T19:26:32.2643369-05:00

Figura 2.3: Creación Base de Datos DB-Banco

```

CREATE TABLE TB_Clientes(
    IDCliente INT NOT NULL,
    NombreCliente VARCHAR(30) NOT NULL,
    ApellidoCliente VARCHAR(30) NOT NULL,
    CorreoElectronico VARCHAR(60) NOT NULL,
    CONSTRAINT PK_TB_Clientes_IDCliente PRIMARY KEY (IDCliente),
);

```

Messages

Commands completed successfully.

Completion time: 2024-10-09T19:26:32.2643369-05:00

Figura 2.4: Creación de la tabla TB-Clientes

```

CREATE TABLE TB_Prestamos(
    IDPrestamo INT NOT NULL,
    IDCliente INT NOT NULL,
    MontoInicial DECIMAL(10, 2) NOT NULL,
    TasaInteres DECIMAL(10, 2) NOT NULL,
    FechaInicio DATE NOT NULL,
    PeriodoPrestamo VARCHAR(7) NOT NULL, --Diario, Mensual, Anual
    EstadoPrestamo VARCHAR(6) NOT NULL, --Activo, Pagado
    CONSTRAINT PK_TB_Prestamos_IDPrestamo PRIMARY KEY (IDPrestamo),
    CONSTRAINT FK_TB_Prestamos_TB_Clientes_IDCliente
    FOREIGN KEY (IDCliente) REFERENCES TB_Clientes(IDCliente),
);

```

Messages

Commands completed successfully.

Completion time: 2024-10-09T19:26:32.2643369-05:00

Figura 2.5: Creación de la tabla TB-Prestamos

```

CREATE TABLE TB_Cuentas(
    IDCuenta INT NOT NULL,
    IDCliente INT NOT NULL,
    SaldoDeposito DECIMAL(10, 2) NOT NULL,
    InteresRetorno DECIMAL(10, 2) NOT NULL,
    FechaApertura DATE NOT NULL,
    CONSTRAINT PK_TB_Cuentas_IDCuenta PRIMARY KEY (IDCuenta),
    CONSTRAINT FK_TB_Cuentas_TB_Clientes_IDCliente
    FOREIGN KEY (IDCliente) REFERENCES TB_Clientes(IDCliente),
);

```

Messages

Commands completed successfully.

Completion time: 2024-10-09T19:26:32.2643369-05:00

Figura 2.6: Creación de la tabla TB-Cuentas

```
CREATE TABLE TB_Transacciones(  
    IDTransaccion INT NOT NULL,  
    IDCuenta INT NOT NULL,  
    MontoTransaccion DECIMAL(10, 2) NOT NULL,  
    TipoTransaccion VARCHAR(8) NOT NULL, --Retiro, Deposito  
    FechaTransaccion DATE NOT NULL,  
    CONSTRAINT PK_TB_Transacciones_IDTransaccion PRIMARY KEY (IDTransaccion),  
    CONSTRAINT FK_TB_Transacciones_TB_Cuentas_IDCuenta  
    FOREIGN KEY (IDCuenta) REFERENCES TB_Cuentas(IDCuenta),  
);
```

Messages
Commands completed successfully.

Completion time: 2024-10-09T19:26:32.2643369-05:00

Figura 2.7: Creación de la tabla TB-Transacciones

```
CREATE TABLE TB_Ahorro(  
    IDCuenta INT NOT NULL,  
    IDAhorro INT NOT NULL,  
    LimiteRetiro DECIMAL(10, 2) NOT NULL,  
    CONSTRAINT FK_TB_Ahorro_TB_Cuentas_IDCuenta  
    FOREIGN KEY (IDCuenta) REFERENCES TB_Cuentas(IDCuenta),  
);
```

Messages
Commands completed successfully.

Completion time: 2024-10-09T19:26:32.2643369-05:00

Figura 2.8: Creación de la tabla TB-Ahorro

```
CREATE TABLE TB_PlazoFijo(  
    IDCuenta INT NOT NULL,  
    IDPlazoFijo INT NOT NULL,  
    PlazoRetiro INT NOT NULL, --Anual  
    CONSTRAINT FK_TB_PlazoFijo_TB_Cuentas_IDCuenta  
    FOREIGN KEY (IDCuenta) REFERENCES TB_Cuentas(IDCuenta),  
);
```

Messages
Commands completed successfully.

Completion time: 2024-10-09T19:26:32.2643369-05:00

Figura 2.9: Creación de la tabla TB-PlazoFijo


```
--Crear Tabla ListaNegra
CREATE TABLE TB_ListaNegra(
    IDCliente INT NOT NULL,
    Castigo VARCHAR(50) NOT NULL DEFAULT 'Expropiacion de Vivienda',
    CONSTRAINT FK_TB_ListaNegra_TB_Clientes_IDCliente
    FOREIGN KEY (IDCliente) REFERENCES TB_Clientes(IDCliente),
);
GO
```

Messages

Commands completed successfully.

Completion time: 2024-10-10T16:58:17.6894276-05:00

Figura 2.10: Creación de la tabla TB-ListaNegra

2.3 Creación de Triggers

```
CREATE TRIGGER TR_ValidarEstadoPrestamo
ON TB_Prestamos
AFTER INSERT, UPDATE
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE EstadoPrestamo NOT IN ('Activo', 'Pagado')
    )
    BEGIN
        ROLLBACK TRANSACTION;
        RAISERROR('El estado del préstamo solo puede ser 'Activo' o 'Pagado'.', 16, 1);
    END
END;
GO
```

Messages

Commands completed successfully.

Completion time: 2024-10-11T19:28:45.8930626-05:00

Figura 2.11: Creación Trigger TR-ValidarEstadoPrestamo

```

CREATE TRIGGER TR_ValidarPeriodoPrestamo
ON TB_Prestamos
AFTER INSERT, UPDATE
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE PeriodoPrestamo NOT IN ('Diario', 'Semanal', 'Mensual', 'Anual')
    )
    BEGIN
        ROLLBACK TRANSACTION;
        RAISERROR('El periodo del préstamo solo puede ser ''Diario'', ''Semanal'', ''Mensual'' o ''Anual''.', 16, 1);
    END
END;
GO

```

Messages

Commands completed successfully.

Completion time: 2024-10-11T19:28:45.8930626-05:00

Figura 2.12: Creación Trigger TR-ValidarPeriodoPrestamo

```

CREATE TRIGGER TR_ValidarTipoTransaccion
ON TB_Transacciones
AFTER INSERT, UPDATE
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE TipoTransaccion NOT IN ('Deposito', 'Retiro')
    )
    BEGIN
        ROLLBACK TRANSACTION;
        RAISERROR('El tipo de transacción solo puede ser ''Deposito'' o ''Retiro''.', 16, 1);
    END
END;
GO

```

Messages

Commands completed successfully.

Completion time: 2024-10-11T19:28:45.8930626-05:00

Figura 2.13: Creación Trigger TR-ValidarTipoTransaccion

```

CREATE TRIGGER TR_ValidarTipoCuenta
ON TB_Transacciones
AFTER INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM TB_PlazoFijo PF
        JOIN inserted I ON PF.IDCuenta = I.IDCuenta
    )
    BEGIN
        ROLLBACK TRANSACTION;
        RAISERROR('No se pueden realizar transacciones en cuentas de Plazo Fijo.', 16, 1);
    END
END;
GO

```

Messages

Commands completed successfully.

Completion time: 2024-10-11T19:28:45.8930626-05:00

Figura 2.14: Creación Trigger TR-ValidarTipoCuenta

```

CREATE TRIGGER TR_ValidarTransaccionRetiro
ON TB_Transacciones
AFTER INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE TipoTransaccion = 'Retiro'
    )
    BEGIN
        DECLARE @IDCuenta INT, @MontoRetiro DECIMAL(10, 2), @SaldoDeposito DECIMAL(10, 2), @LimiteRetiro DECIMAL(10, 2);
        --Seleccionar IDCuenta y MontoTransaccion (Retiro)
        SELECT @IDCuenta = IDCuenta, @MontoRetiro = MontoTransaccion
        FROM inserted;
        --Seleccionar SaldoDeposito
        SELECT @SaldoDeposito = SaldoDeposito
        FROM TB_Cuentas
    END
END

```

Messages
Commands completed successfully.
Completion time: 2024-10-11T19:28:45.8930626-05:00

Figura 2.15: Creación Trigger TR-ValidarTransaccionRetiro

```

CREATE TRIGGER TR_ValidarTransaccionDeposito
ON TB_Transacciones
AFTER INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE TipoTransaccion = 'Deposito'
    )
    BEGIN
        DECLARE @IDCuenta INT, @MontoDeposito DECIMAL(10, 2);
        --Seleccionar IDCuenta y MontoTransaccion (Deposito)
        SELECT @IDCuenta = IDCuenta, @MontoDeposito = MontoTransaccion
        FROM inserted;
        --Aumentar SaldoCuenta
        UPDATE TB_Cuentas
        SET SaldoDeposito = SaldoDeposito + @MontoDeposito
    END
END

```

Messages
Commands completed successfully.
Completion time: 2024-10-11T19:28:45.8930626-05:00

Figura 2.16: Creación Trigger TR-ValidarTransaccionDeposito

```

CREATE TRIGGER TR_MoverAListaNegra
ON TB_Prestamos
AFTER UPDATE
AS
BEGIN
    DECLARE @IDCliente INT, @IDPrestamo INT, @FechaInicio DATE, @PeriodoPrestamo VARCHAR(7);
    -- Seleccionar datos del prestamo
    SELECT @IDCliente = i.IDCliente, @IDPrestamo = i.IDPrestamo, @FechaInicio = i.FechaInicio, @PeriodoPrestamo = i.PeriodoPrestamo
    FROM inserted i
    WHERE i.EstadoPrestamo = 'Activo';
    -- Validar si el préstamo es diario y han pasado más de 120 días
    IF @PeriodoPrestamo = 'Diario' AND DATEDIFF(DAY, @FechaInicio, GETDATE()) > 120
    BEGIN
        GOTO EnviarListaNegra;
    END;
    EnviarListaNegra:
    INSERT INTO TB_ListaNegra (IDCliente)
    VALUES (@IDCliente);
    GO
END;

```

Messages
Commands completed successfully.
Completion time: 2024-10-13T17:52:51.3286764-05:00

Figura 2.17: Creación Trigger TR-MoverAListaNegra

2.4 Inserción de Registros

```
DECLARE @NombreCliente VARCHAR(30), @ApellidoCliente VARCHAR(30), @CorreoElectronico VARCHAR(60)
SET @NombreCliente = 'Gendor';
SET @ApellidoCliente = 'Skraivok';
SET @CorreoElectronico = 'painted.count@nostramo.com';

INSERT INTO TB_Clientes
(NombreCliente, ApellidoCliente, CorreoElectronico)
VALUES
(@NombreCliente, @ApellidoCliente, @CorreoElectronico),
('Zahariel', 'El Zurias', 'caliban.order@lion.com'),
('Arabella', 'Stern', 'sororita.canoness@ministorum.com')
GO
```

%

Messages

Completion time: 2024-10-11T19:38:43.5485599-05:00

Figura 2.18: Inserción registros Clientes

```
INSERT INTO TB_Prestamos
(IDCliente, MontoInicial, FechaInicio, PeriodoPrestamo)
VALUES
(1, 800, '2024-10-10', 'Mensual'),
(3, 1500, '2023-05-23', 'Diario')
GO
```

%

Messages

Completion time: 2024-10-11T19:38:43.5485599-05:00

Figura 2.19: Inserción registros Prestamos

```
INSERT INTO TB_Cuentas
(IDCliente, SaldoDeposito, InteresRetorno, FechaApertura)
VALUES
(2, 30000, 1.1, '2021-02-20'),
(3, 46000, 4.6, '2019-06-17')
GO
```

%

Messages

Completion time: 2024-10-11T19:38:43.5485599-05:00

Figura 2.20: Inserción registros Cuentas

```

INSERT INTO TB_Ahorro
(IDCuenta)
VALUES
(1)
GO

```

Completion time: 2024-10-11T19:38:43.5485599-05:00

Figura 2.21: Inserción registros Ahorro

```

INSERT INTO TB_PlazoFijo
(IDCuenta)
VALUES
(2)
GO

```

Completion time: 2024-10-11T19:38:43.5485599-05:00

Figura 2.22: Inserción registros PlazoFijo

2.5 Probar Triggers

```

INSERT INTO TB_Transacciones
(IDCuenta, MontoTransaccion, TipoTransaccion, FechaTransaccion)
VALUES
(2, 150, 'Retiro', GETDATE())
GO

```

Completion time: 2024-10-11T19:44:07.9115668-05:00

Msg 50000, Level 16, State 1, Procedure TR_ValidarTipoCuenta, Line 15 [I
No se pueden realizar transacciones en cuentas de Plazo Fijo.
Msg 3609, Level 16, State 1, Line 6
The transaction ended in the trigger. The batch has been aborted.

Figura 2.23: Restricción Transacción PlazoFijo

```

INSERT INTO TB_Transacciones
(IDCuenta, MontoTransaccion, TipoTransaccion, FechaTransaccion)
VALUES
(1, 5001, 'Retiro', GETDATE())
GO

```

Messages

Msg 50000, Level 16, State 1, Procedure TR_ValidarTransaccionRetiro, Lin
El monto de retiro excede el límite de retiro permitido para esta cuenta
Msg 3609, Level 16, State 1, Line 13
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2024-10-11T19:47:22.9329601-05:00

Figura 2.24: Restricción Transacción límite en Retiro

```

SELECT * FROM TB_Cuentas
INSERT INTO TB_Transacciones
(IDCuenta, MontoTransaccion, TipoTransaccion, FechaTransaccion)
VALUES
(1, 12000, 'Deposito', GETDATE())
GO
SELECT * FROM TB_Transacciones
SELECT * FROM TB_Cuentas

```

Results Messages

IDCuenta	IDCliente	SaldoDeposito	InteresRetorno	FechaApertura
1	2	30000.00	1.10	2021-02-20
2	3	46000.00	4.60	2019-06-17

IDTransaccion	IDCuenta	MontoTransaccion	TipoTransaccion	FechaTransaccion
3	1	12000.00	Deposito	2024-10-11

IDCuenta	IDCliente	SaldoDeposito	InteresRetorno	FechaApertura
1	2	42000.00	1.10	2021-02-20
2	3	46000.00	4.60	2019-06-17

Figura 2.25: Activar Trigger aumento de Depósito

```

SELECT * FROM TB_ListaNegra
UPDATE TB_Prestamos
SET TasaInteres = 4.5 --4.6
WHERE IDPrestamo = 2
SELECT * FROM TB_ListaNegra
GO

```

Results Messages

IDCliente	Castigo
3	Expropiacion de Vivienda

Figura 2.26: Activar Trigger enviar a Lista Negra

2.6 Ejercicios

```
--Declarar variables
DECLARE @IDCliente INT, @SaldoTotal DECIMAL(20, 2), @InteresPrestamo DECIMAL(20, 2),
@MontoPrestamo DECIMAL(20, 2), @InteresPlazoFijo DECIMAL(20, 2), @FondosTotales DECIMAL(20, 2);

--Establecer el ID del cliente
SET @IDCliente = (SELECT IDCliente FROM TB_Clientes WHERE IDCliente = 3);

--Obtener el saldo total de las cuentas del cliente (solo Plazo Fijo)
SELECT @SaldoTotal = SUM(SaldoDeposito)
FROM TB_Cuentas
WHERE IDCliente = @IDCliente AND IDCuenta IN (SELECT IDCuenta FROM TB_PlazoFijo);

--Obtener el interés generado en la cuenta a Plazo Fijo con interés simple
--Interés Simple = MontoInicial * TasaInteres * CantidadPlazos
SELECT @InteresPlazoFijo = SUM(SaldoDeposito * InteresRetorno / 100 * PlazoRetiro)
FROM TB_Cuentas c
JOIN TB_PlazoFijo p ON c.IDCuenta = p.IDCuenta
```

Results Messages

SaldoTotal	InteresPlazoFijo	MontoPrestamo	DiasAcumulados	InteresPrestamo	FondosTotales
46000.00	6348.00	1500.00	507	82997667505951.67	-82997667455103.67

Figura 2.27: Ejercicio Fondos Totales