

Ejercicios propuestos 1

Jesus Iñiguez García

March 30, 2017

1 Crear base de datos

```
CREATE DATABASE banco;  
USE banco;
```

2 Creación de tablas

```
CREATE TABLE Banco (  
    Numero_cuenta char(20) NOT NULL,  
    Nombre varchar(50) ,  
    Apellidos varchar(50) ,  
    Saldo float NOT NULL DEFAULT '0',  
    PRIMARY KEY (Numero_cuenta)  
);  
  
CREATE TABLE Operacion(  
    id INT AUTO_INCREMENT,  
    asunto VARCHAR(100),  
    fecha DATETIME NOT NULL DEFAULT NOW(),  
    cantidad FLOAT NOT NULL,  
    cuenta char(20) NOT NULL,  
    PRIMARY KEY(id),  
    CONSTRAINT cuenta FOREIGN KEY (cuenta) REFERENCES Banco(Numero_cuenta)  
);
```

Al ser la relacion 1 a muchos, simplemente añado la cuenta a operacion como no nula.

3 Creación de disparadores

```
CREATE TRIGGER nueva_operacion  
AFTER INSERT ON Operacion FOR EACH ROW
```

```
UPDATE Banco SET Saldo = Saldo + NEW.cantidad
where Numero_cuenta = NEW.cuenta;
```

```
DELIMITER $$
CREATE TRIGGER nueva_cuenta
BEFORE INSERT ON Banco FOR EACH ROW
BEGIN
    IF NEW.Saldo < 0 THEN
        SET NEW.Saldo = 0;
    END IF;
END$$
DELIMITER ;
```

Para el segundo trigger hay otra manera de hacerlo, algo más compleja pero también válida.

```
DELIMITER $$
CREATE TRIGGER nueva_cuenta
AFTER INSERT ON Banco FOR EACH ROW
BEGIN
    IF NEW.Saldo < 0 THEN
        UPDATE Banco SET Saldo = 0 where Numero_cuenta = NEW.Numero_cuenta;
    END IF;
END$$
DELIMITER ;
```

La diferencia en este segundo es que realizo la modificación en el registro de Banco ya creado, puesto que la columna NEW no la puedo alterar en un "AFTER INSERT"

4 Inserción de datos

```
INSERT INTO Banco (Numero_cuenta, Nombre, Apellidos)
VALUES ("12345678901234567890", "juan", "muñoz");

INSERT INTO Banco (Numero_cuenta, Nombre, Apellidos, Saldo)
VALUES ("12345678901234567891", "pepe", "perez", -200);

INSERT INTO Operacion(asunto, cantidad, cuenta)
VALUES ("incremento", 100, "12345678901234567890");
```

Con estos casos compruebo qué ocurre al dejar campos vacíos (evidentemente se rellenan aquellos que estan como default, sólo esos) y comprobando los disparadores, que reaccionan como se esperaba.

5 Creación de procedimientos

```
DELIMITER $$
CREATE PROCEDURE transaccion(
    IN cuenta1 CHAR(20),
    IN cuenta2 CHAR(20),
    IN cantidad FLOAT)
BEGIN
    UPDATE Banco SET Saldo = Saldo - cantidad where Numero_cuenta = cuenta1;
    UPDATE Banco SET Saldo = Saldo + cantidad where Numero_cuenta = cuenta2;
END$$
DELIMITER ;
```

Esta era la respuesta inicial, pero realmente no es muy correcta.

```
DELIMITER $$
CREATE PROCEDURE transaccion(
    IN cuenta1 CHAR(20),
    IN cuenta2 CHAR(20),
    IN cant FLOAT,
    IN concepto VARCHAR(100))
BEGIN
    INSERT INTO Operacion(asunto, cantidad, cuenta)
    VALUES (concepto, -cant, cuenta1);
    INSERT INTO Operacion(asunto, cantidad, cuenta)
    VALUES (concepto, cant, cuenta2);
END$$
DELIMITER ;
```

Es la opción más correcta porque así la operación queda guardada con un concepto en la tabla operacion.

- En el caso de poner una cuenta inexistente, el procedimiento fallará en esa operación concreta.
- Si el procedimiento falla en alguna operación, no realizará nada más a partir de ahí, pero lo previo queda hecho, esto es, si la segunda cuenta es la incorrecta, la primera cuenta verá reflejada la operación pero no la segunda.

6 Manejo de errores y transacciones

```
DELIMITER $$

CREATE PROCEDURE transaccion(
    IN cuenta1 CHAR(20),
    IN cuenta2 CHAR(20),
```

```

        IN cant FLOAT,
        IN concepto VARCHAR(100))

BEGIN
    DECLARE EXIT HANDLER FOR 1452 ROLLBACK;
    START TRANSACTION;

    IF cant < 0 THEN
        SIGNAL SQLSTATE 'ERROR' SET MESSAGE_TEXT = 'Cantidad negativa';
    END IF;

    INSERT INTO Operacion(asunto, cantidad, cuenta)
        VALUES (concepto, -cant, cuenta1);
    INSERT INTO Operacion(asunto, cantidad, cuenta)
        VALUES (concepto, cant, cuenta2);

    COMMIT;
END$$
DELIMITER ;

```

He utilizado el signal que bloquea la transaccion y no se continua la ejecución, al final pongo un commit para controlar que la transaccion acabe cuando yo lo deseo, y el rollback en caso de rror 1452 (El error relacionado con las tablas)