Problem Statement 2:

Implementing SQL Triggers for Data Validation Scenario:

Your team is developing a financial application, and you need to implement SQL triggers to ensure data accuracy and validation. Create triggers to automatically enforce constraints and validate incoming data.

WHAT IS A SQL TRIGGER?

A trigger is a special type of stored procedure in SQL that automatically executes when a specific event (INSERT, DELETE, UPDATE) occurs in a table. Triggers help in enforcing business rules, data validation and ensuring data integrity.

KEY FEATURES:

Automatic execution -- triggers run automatically when the specified event occurs.

Data Validation -- Prevents incorrect or unwanted changes to the database.

Enforcing business rules -- Ensures data integrity without requiring additional application logic.

TYPES OF TRIGGERS

BEFORE TRIGGER:	Executes before the event (used for validation).
AFTER TRIGGER:	Executes after the event (used for logging or auditing).
INSTEAD OF TRIGGER:	Replaces the execution of event (common of views).

WHAT IS DELIMITER IN SQL?

DELIMITER is a command in MYSQL to change the default statement terminator (;).

It is mainly used in triggers, stored procedures, or functions to avoid conflicts with semicolons inside the block.

Example:

```
DELIMITER //

{…

….}

//

DELIMITER;
```

Explanation:

1. **DELIMITER //** – Changes the delimiter from **;** to **//.**
2. **END //** – Ends the trigger using **//** instead of **;** .
3. **DELIMITER ;** – Resets back to **;** for normal SQL execution.

```
-- Ensure transaction amounts are positive

DELIMITER //



CREATE TRIGGER before_insert_transaction

BEFORE INSERT ON transactions

FOR EACH ROW

BEGIN

   IF NEW.amount <= 0 THEN

      SIGNAL SQLSTATE '45000'

      SET MESSAGE_TEXT = 'Transaction amount must be positive';

   END IF;

END;

//



-- Prevent negative account balance

CREATE TRIGGER before_withdrawal

BEFORE INSERT ON transactions

FOR EACH ROW

BEGIN

   DECLARE account_balance DECIMAL(10,2);

   -- Get current balance

   SELECT balance INTO account_balance FROM accounts WHERE account_id = NEW.account_id;

   -- Check if balance is sufficient

   IF NEW.transaction_type = 'withdrawal' AND (account_balance - NEW.amount < 0) THEN

      SIGNAL SQLSTATE '45000'

      SET MESSAGE_TEXT = 'Insufficient funds';

   END IF;

END;

//-- Prevent self-transfers
```

```
CREATE TRIGGER before_self_transfer
BEFORE INSERT ON transactions
FOR EACH ROW
BEGIN
    IF NEW.sender_id = NEW.receiver_id THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Cannot transfer money to the same account';
    END IF;
END;
//

DELIMITER ;
```