```sql
CREATE DATABASE IF NOT EXISTS new_code;

USE new_code;

-- Customers Table

CREATE TABLE IF NOT EXISTS customers (

  CustomerID INT PRIMARY KEY,

  Name VARCHAR(100),

  Age INT,

  Balance DECIMAL(10, 2),

  IsVIP BOOLEAN DEFAULT FALSE

);

-- Loans Table

CREATE TABLE IF NOT EXISTS loans (

  LoanID INT PRIMARY KEY,

  CustomerID INT,

  InterestRate DECIMAL(5,2),

  DueDate DATE,

  FOREIGN KEY (CustomerID) REFERENCES customers(CustomerID)

);

-- Sample Customers

INSERT INTO customers (CustomerID, Name, Age, Balance) VALUES

(1, 'John Doe', 65, 12000.00),

(2, 'Jane Smith', 45, 8000.00),

(3, 'Alice Brown', 70, 15000.00);

-- Sample Loans

INSERT INTO loans (LoanID, CustomerID, InterestRate, DueDate) VALUES

(101, 1, 9.5, CURDATE() + INTERVAL 10 DAY),

(102, 2, 8.0, CURDATE() + INTERVAL 40 DAY),

(103, 3, 10.0, CURDATE() + INTERVAL 20 DAY);
```

-- Exercise 1: Control Structures

-- Scenario 1: The bank wants to apply a discount to loan interest rates for customers above 60 years old.

```
--  SCENARIO 1: Apply 1% discount for customers above 60
DELIMITER $$
CREATE PROCEDURE ApplySeniorDiscount()
BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE cust_id INT;
  DECLARE cur_age INT;
  DECLARE loan_cursor CURSOR FOR
    SELECT c.CustomerID, c.Age
    FROM customers c
    JOIN loans l ON c.CustomerID = l.CustomerID;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
  OPEN loan_cursor;
  read_loop: LOOP
    FETCH loan_cursor INTO cust_id, cur_age;
    IF done THEN
      LEAVE read_loop;
    END IF;
    IF cur_age > 60 THEN
      UPDATE loans
      SET InterestRate = InterestRate - 1
      WHERE CustomerID = cust_id;
    END IF;
  END LOOP;
  CLOSE loan_cursor;
END$$
DELIMITER ;
```

-- Scenario 2: A customer can be promoted to VIP status based on their balance.

-- SCENARIO 2: Promote customers to VIP if balance > $10,000

```
DROP PROCEDURE IF EXISTS ApplySeniorDiscount;
DELIMITER $$
CREATE PROCEDURE ApplySeniorDiscount()
BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE cust_id INT;
  DECLARE cur_age INT;

  DECLARE loan_cursor CURSOR FOR
    SELECT c.CustomerID, c.Age
    FROM customers c
    JOIN loans l ON c.CustomerID = l.CustomerID;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
  OPEN loan_cursor;
  read_loop: LOOP
    FETCH loan_cursor INTO cust_id, cur_age;
    IF done THEN
      LEAVE read_loop;
    END IF;
    IF cur_age > 60 THEN
      UPDATE loans
      SET InterestRate = InterestRate - 1
      WHERE CustomerID = cust_id;
    END IF;
  END LOOP;
  CLOSE loan_cursor;
END$$
DELIMITER ;
```

-- Scenario 3: The bank wants to send reminders to customers whose loans are due within the next 30 days.

-- SCENARIO 3: Send reminders for loans due in 30 days

```
DELIMITER $$
CREATE PROCEDURE PromoteToVIP()
BEGIN
  UPDATE customers
  SET IsVIP = TRUE
  WHERE Balance > 10000;
END$$
DELIMITER ;
DELIMITER $$
CREATE PROCEDURE SendLoanReminders()
BEGIN
  DECLARE done INT DEFAULT FALSE;
  DECLARE cust_name VARCHAR(100);
  DECLARE due_date DATE;
  DECLARE loan_cursor CURSOR FOR
    SELECT c.Name, l.DueDate
    FROM customers c
    JOIN loans l ON c.CustomerID = l.CustomerID
    WHERE l.DueDate BETWEEN CURDATE() AND CURDATE() + INTERVAL 30 DAY;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
  OPEN loan_cursor;
  read_loop: LOOP
    FETCH loan_cursor INTO cust_name, due_date;
    IF done THEN
      LEAVE read_loop;
    END IF;
    SELECT CONCAT('Reminder: Dear ', cust_name, ', your loan is due on ', due_date) AS Reminder;
  END LOOP;
  CLOSE loan_cursor;
END$$
DELIMITER ;
```

```
-- Exercise 3: Stored Procedures

-- Scenario 1: The bank needs to process monthly interest for all savings accounts.

-- Scenario 1: Process Monthly Interest (1%)

-- Savings Accounts Table

CREATE TABLE IF NOT EXISTS savings_accounts (

  AccountID INT PRIMARY KEY,

  CustomerID INT,

  Balance DECIMAL(10,2)

);

-- Employees Table

CREATE TABLE IF NOT EXISTS employees (

  EmployeeID INT PRIMARY KEY,

  Name VARCHAR(100),

  Department VARCHAR(100),

  Salary DECIMAL(10,2)

);

-- Accounts Table for Fund Transfers

CREATE TABLE IF NOT EXISTS accounts (

  AccountID INT PRIMARY KEY,

  CustomerID INT,

  Balance DECIMAL(10,2)

);

DELIMITER $$

CREATE PROCEDURE ProcessMonthlyInterest()

BEGIN

  UPDATE savings_accounts

  SET Balance = Balance + (Balance * 0.01);

END$$

DELIMITER ;
```

-- Scenario 2: A customer can be promoted to VIP status based on their balance

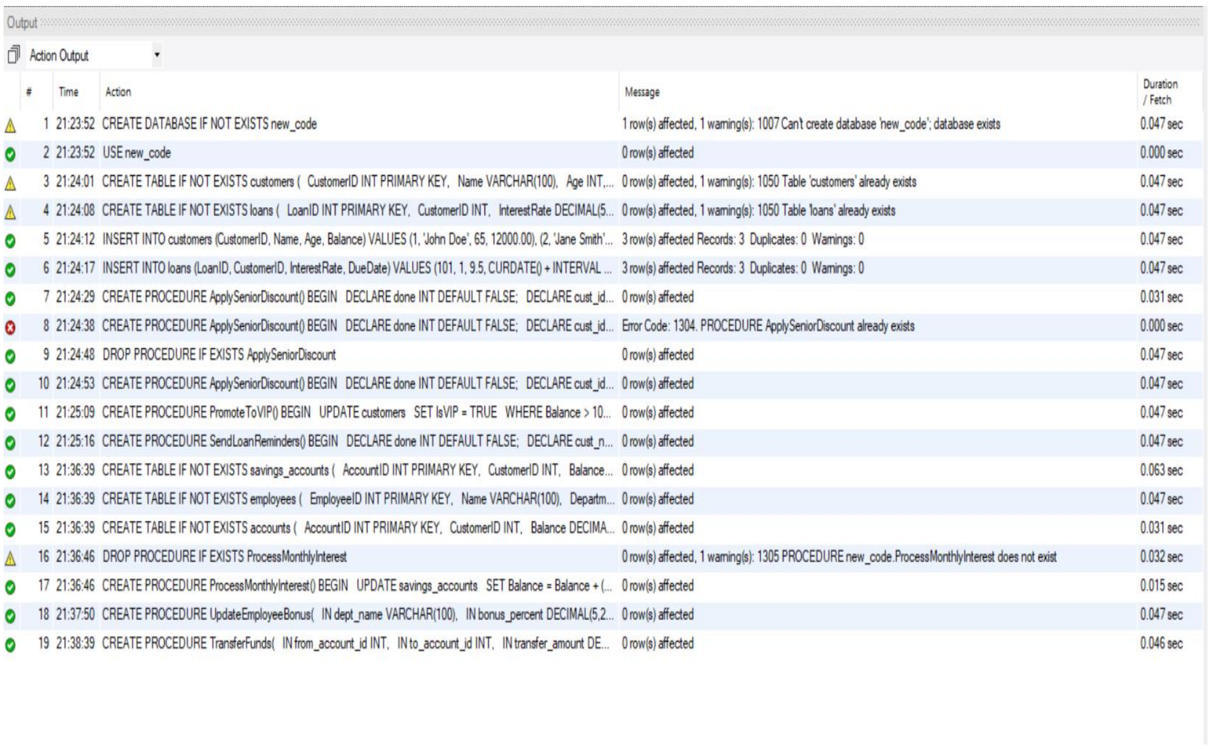
-- Scenario 2: Update Employee Bonus (Parameterized by department and bonus %)

```
DROP PROCEDURE IF EXISTS UpdateEmployeeBonus;


DELIMITER $$


CREATE PROCEDURE UpdateEmployeeBonus(
  IN dept_name VARCHAR(100),
  IN bonus_percent DECIMAL(5,2)
)
BEGIN
  UPDATE employees
  SET Salary = Salary + (Salary * bonus_percent / 100)
  WHERE Department = dept_name;
END$$


DELIMITER ;
```

-- Scenario 3: The bank wants to send reminders to customers whose loans are due within the next 30 days.

-- Scenario 3: Transfer Funds Between Accounts

```
DROP PROCEDURE IF EXISTS TransferFunds;


DELIMITER $$
CREATE PROCEDURE TransferFunds(
  IN from_account_id INT,
  IN to_account_id INT,
  IN transfer_amount DECIMAL(10,2)
)
BEGIN
  DECLARE from_balance DECIMAL(10,2);
 -- Get current balance of source account
  SELECT Balance INTO from_balance
  FROM accounts
  WHERE AccountID = from_account_id;
  -- Check if balance is sufficient
  IF from_balance >= transfer_amount THEN
    -- Deduct from source account
    UPDATE accounts
    SET Balance = Balance - transfer_amount
    WHERE AccountID = from_account_id;
    -- Add to destination account
    UPDATE accounts
    SET Balance = Balance + transfer_amount
    WHERE AccountID = to_account_id;
  ELSE
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Insufficient balance in source account';
  END IF;
END$$
DELIMITER ;
```



```
Output
Action Output ▾

   #  Time      Action                                                                                               Message                                                                                      Duration / Fetch
⚠  1  21:23:52  CREATE DATABASE IF NOT EXISTS new_code                                                               1 row(s) affected, 1 warning(s): 1007 Can't create database 'new_code'; database exists       0.047 sec
✓  2  21:23:52  USE new_code                                                                                         0 row(s) affected                                                                            0.000 sec
⚠  3  21:24:01  CREATE TABLE IF NOT EXISTS customers (   CustomerID INT PRIMARY KEY,   Name VARCHAR(100),   Age INT,...  0 row(s) affected, 1 warning(s): 1050 Table 'customers' already exists                        0.047 sec
⚠  4  21:24:08  CREATE TABLE IF NOT EXISTS loans (   LoanID INT PRIMARY KEY,   CustomerID INT,   InterestRate DECIMAL(5...  0 row(s) affected, 1 warning(s): 1050 Table 'loans' already exists                            0.047 sec
✓  5  21:24:12  INSERT INTO customers (CustomerID, Name, Age, Balance) VALUES (1, 'John Doe', 65, 12000.00), (2, 'Jane Smith'...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0                          0.047 sec
✓  6  21:24:17  INSERT INTO loans (LoanID, CustomerID, InterestRate, DueDate) VALUES (101, 1, 9.5, CURDATE() + INTERVAL ...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0                          0.047 sec
✓  7  21:24:29  CREATE PROCEDURE ApplySeniorDiscount() BEGIN   DECLARE done INT DEFAULT FALSE;   DECLARE cust_id...     0 row(s) affected                                                                            0.031 sec
✗  8  21:24:38  CREATE PROCEDURE ApplySeniorDiscount() BEGIN   DECLARE done INT DEFAULT FALSE;   DECLARE cust_id...     Error Code: 1304. PROCEDURE ApplySeniorDiscount already exists                               0.000 sec
✓  9  21:24:48  DROP PROCEDURE IF EXISTS ApplySeniorDiscount                                                         0 row(s) affected                                                                            0.047 sec
✓ 10  21:24:53  CREATE PROCEDURE ApplySeniorDiscount() BEGIN   DECLARE done INT DEFAULT FALSE;   DECLARE cust_id...     0 row(s) affected                                                                            0.047 sec
✓ 11  21:25:09  CREATE PROCEDURE PromoteToVIP() BEGIN   UPDATE customers   SET IsVIP = TRUE   WHERE Balance > 10...     0 row(s) affected                                                                            0.047 sec
✓ 12  21:25:16  CREATE PROCEDURE SendLoanReminders() BEGIN   DECLARE done INT DEFAULT FALSE;   DECLARE cust_n...        0 row(s) affected                                                                            0.047 sec
✓ 13  21:36:39  CREATE TABLE IF NOT EXISTS savings_accounts (   AccountID INT PRIMARY KEY,   CustomerID INT,   Balance...  0 row(s) affected                                                                         0.063 sec
✓ 14  21:36:39  CREATE TABLE IF NOT EXISTS employees (   EmployeeID INT PRIMARY KEY,   Name VARCHAR(100),   Departm...  0 row(s) affected                                                                          0.047 sec
✓ 15  21:36:39  CREATE TABLE IF NOT EXISTS accounts (   AccountID INT PRIMARY KEY,   CustomerID INT,   Balance DECIMA...  0 row(s) affected                                                                           0.031 sec
⚠ 16  21:36:46  DROP PROCEDURE IF EXISTS ProcessMonthlyInterest                                                      0 row(s) affected, 1 warning(s): 1305 PROCEDURE new_code.ProcessMonthlyInterest does not exist  0.032 sec
✓ 17  21:36:46  CREATE PROCEDURE ProcessMonthlyInterest() BEGIN   UPDATE savings_accounts   SET Balance = Balance + (...  0 row(s) affected                                                                          0.015 sec
✓ 18  21:37:50  CREATE PROCEDURE UpdateEmployeeBonus(   IN dept_name VARCHAR(100),   IN bonus_percent DECIMAL(5,2...    0 row(s) affected                                                                            0.047 sec
✓ 19  21:38:39  CREATE PROCEDURE TransferFunds(   IN from_account_id INT,   IN to_account_id INT,   IN transfer_amount DE...  0 row(s) affected                                                                   0.046 sec
```