**Exercise 1: Control Structures**

**Scenario 1:**

**CODE:**

CREATE TABLE Customers (

CustomerID INTEGER PRIMARY KEY,

Name TEXT,

DOB TEXT, -- YYYY-MM-DD format

Balance REAL,

LastModified TEXT

);

CREATE TABLE Loans (

LoanID INTEGER PRIMARY KEY,

CustomerID INTEGER,

LoanAmount REAL,

InterestRate REAL,

StartDate TEXT,

EndDate TEXT,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified) VALUES

(1, 'John Doe', '1950-05-15', 12000, DATE('now')),

(2, 'Jane Smith', '1990-07-20', 1500, DATE('now'));

INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate) VALUES

(1, 1, 5000, 5.00, DATE('now'), DATE('now', '+12 months')),

(2, 2, 3000, 4.50, DATE('now'), DATE('now', '+12 months'));

UPDATE Loans

SET InterestRate = InterestRate - (InterestRate \* 0.01)

WHERE CustomerID IN (

SELECT CustomerID

FROM Customers

WHERE (CAST(strftime('%Y', 'now') AS INTEGER) - CAST(strftime('%Y', DOB) AS INTEGER)) > 60

);

SELECT

l.LoanID,

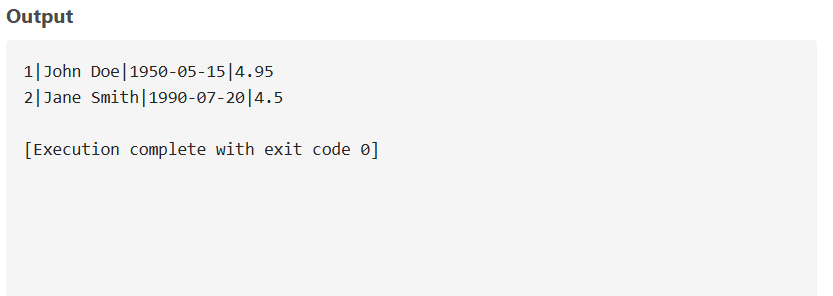
c.Name AS CustomerName,

c.DOB,

l.InterestRate AS UpdatedInterestRate

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID;

**OUTPUT:**

**Scenario 2:**

**CODE:**

DROP TABLE IF EXISTS Customers;

CREATE TABLE Customers (

CustomerID INTEGER PRIMARY KEY,

Name TEXT,

DOB DATE,

Balance REAL,

LastModified DATE,

IsVIP TEXT DEFAULT 'FALSE'

);

INSERT INTO Customers VALUES

(1, 'John Doe', '1985-05-15', 8000, DATE('now'), 'FALSE'),

(2, 'Jane Smith', '1990-07-20', 15000, DATE('now'), 'FALSE'),

(3, 'Michael Ray', '1970-02-12', 20000, DATE('now'), 'FALSE');

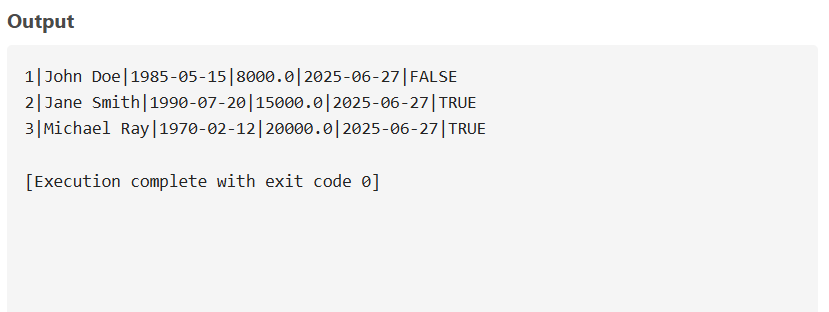
UPDATE Customers

SET IsVIP = 'TRUE'

WHERE Balance > 10000;

SELECT \* FROM Customers;

**OUTPUT:**



**Scenario 3:**

**CODE:**

DROP TABLE IF EXISTS Customers;

DROP TABLE IF EXISTS Loans;

CREATE TABLE Customers (

CustomerID INTEGER PRIMARY KEY,

Name TEXT,

DOB DATE

);

CREATE TABLE Loans (

LoanID INTEGER PRIMARY KEY,

CustomerID INTEGER,

LoanAmount REAL,

InterestRate REAL,

StartDate DATE,

EndDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

INSERT INTO Customers VALUES

(1, 'John Doe', '1985-05-15'),

(2, 'Jane Smith', '1990-07-20'),

(3, 'Michael Ray', '1970-02-12');

INSERT INTO Loans VALUES

(1, 1, 5000, 5, '2022-01-01', DATE('now', '+10 days')), -- Due soon

(2, 2, 10000, 4.5, '2023-01-01', DATE('now', '+40 days')), -- Not due yet

(3, 3, 15000, 6, '2021-01-01', DATE('now', '+5 days')); -- Due very soon

SELECT

c.Name AS CustomerName,

l.LoanID,

l.LoanAmount,

l.EndDate,

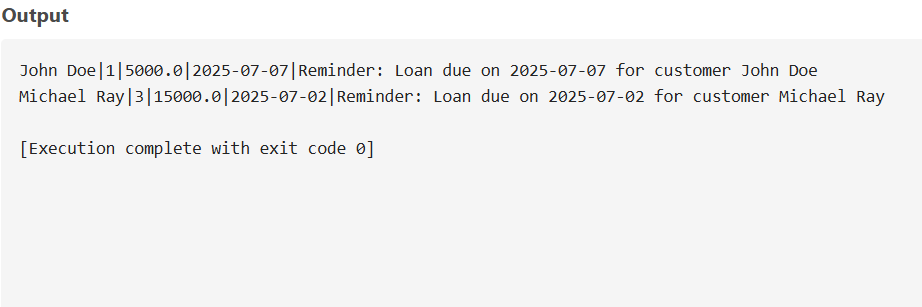
'Reminder: Loan due on ' || l.EndDate || ' for customer ' || c.Name AS ReminderMessage

FROM Loans l

JOIN Customers c ON l.CustomerID = c.CustomerID

WHERE l.EndDate BETWEEN DATE('now') AND DATE('now', '+30 days');

**OUTPUT:**



**Exercise 2: Stored Procedures**

**Scenario 1:**

**CODE:**

DROP TABLE IF EXISTS Accounts

CREATE TABLE Accounts (

AccountID INTEGER PRIMARY KEY,

CustomerID INTEGER,

AccountType TEXT,

Balance REAL,

LastModified DATE

);

INSERT INTO Accounts VALUES

(1, 101, 'Savings', 1000.00, DATE('now')),

(2, 102, 'Savings', 2000.00, DATE('now')),

(3, 103, 'Checking', 1500.00, DATE('now'));

SELECT 'Before interest' AS Note, AccountID, AccountType, Balance FROM Accounts;

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'Savings';

SELECT 'After interest' AS Note, AccountID, AccountType, Balance FROM Accounts;

**OUTPUT:**



**Scenario 2:**

**CODE:**

DROP TABLE IF EXISTS Employees;

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

Name TEXT,

Position TEXT,

Salary REAL,

Department TEXT,

HireDate DATE

);

INSERT INTO Employees VALUES

(1, 'Alice Johnson', 'Manager', 70000, 'HR', '2015-06-15'),

(2, 'Bob Brown', 'Developer', 60000, 'IT', '2017-03-20'),

(3, 'Charlie Lee', 'Developer', 62000, 'IT', '2018-08-10');

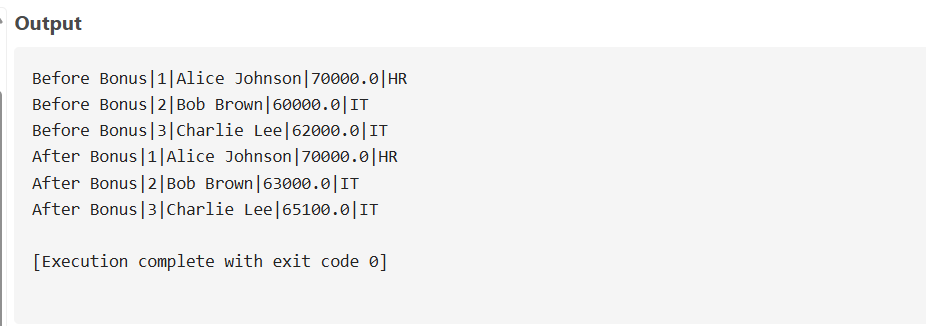
SELECT 'Before Bonus' AS Note, EmployeeID, Name, Salary, Department FROM Employees;

UPDATE Employees

SET Salary = Salary + (Salary \* 5.0 / 100)

WHERE Department = 'IT';

SELECT 'After Bonus' AS Note, EmployeeID, Name, Salary, Department FROM Employees;

**OUTPUT:**

**Scenario 3:**

**CODE:**

DROP TABLE IF EXISTS Accounts;

CREATE TABLE Accounts (

AccountID INTEGER PRIMARY KEY,

CustomerID INTEGER,

AccountType TEXT,

Balance REAL,

LastModified DATE

);

INSERT INTO Accounts VALUES

(1, 101, 'Savings', 1000.00, DATE('now')),

(2, 101, 'Checking', 500.00, DATE('now'));

SELECT 'Before Transfer' AS Note, AccountID, Balance FROM Accounts;

BEGIN TRANSACTION;

SELECT Balance FROM Accounts WHERE AccountID = 1;

UPDATE Accounts

SET Balance = Balance - 300

WHERE AccountID = 1 AND Balance >= 300;

UPDATE Accounts

SET Balance = Balance + 300

WHERE AccountID = 2;

COMMIT;

SELECT 'After Transfer' AS Note, AccountID, Balance FROM Accounts;

**OUTPUT:**

