**Exercise 3: Stored Procedures**

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

**o Question:** Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

Code

-- Initializing database

CREATE TABLE CustomerLoans (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(50),

Age NUMBER,

Balance NUMBER(10,2),

IsVIP CHAR(1) DEFAULT 'N', -- 'Y' or 'N'

InterestRate NUMBER(5,2),

DueDate DATE

);

-- Inserting Sample Records

INSERT INTO CustomerLoans VALUES (1, 'Alice', 65, 15000.00, 'N', 7.5, SYSDATE + 20);

INSERT INTO CustomerLoans VALUES (2, 'Bob', 45, 8000.00, 'N', 8.0, SYSDATE + 40);

INSERT INTO CustomerLoans VALUES (3, 'Charlie', 70, 12000.00, 'N', 7.0, SYSDATE + 15);

INSERT INTO CustomerLoans VALUES (4, 'Diana', 30, 5000.00, 'N', 9.0, SYSDATE + 5);

INSERT INTO CustomerLoans VALUES (5, 'Edward', 61, 11000.00, 'N', 8.5, SYSDATE + 50);

COMMIT;

--PL/SQL Program

DECLARE

i NUMBER := 1;

name VARCHAR2(50);

age NUMBER;

ir NUMBER(5,2);

BEGIN

LOOP

SELECT Name,Age,InterestRate INTO name,age,ir FROM CustomerLoans WHERE CustomerID=i;

if(age > 60) then

UPDATE CustomerLoans SET InterestRate = InterestRate - 1 WHERE CustomerID=i;

ir := ir-1;

dbms\_output.put\_line('CUSTOMER DETAILS:');

dbms\_output.put\_line('NAME:' || name);

dbms\_output.put\_line('AGE:' || age);

dbms\_output.put\_line('INTEREST RATE:' || ir);

END IF;

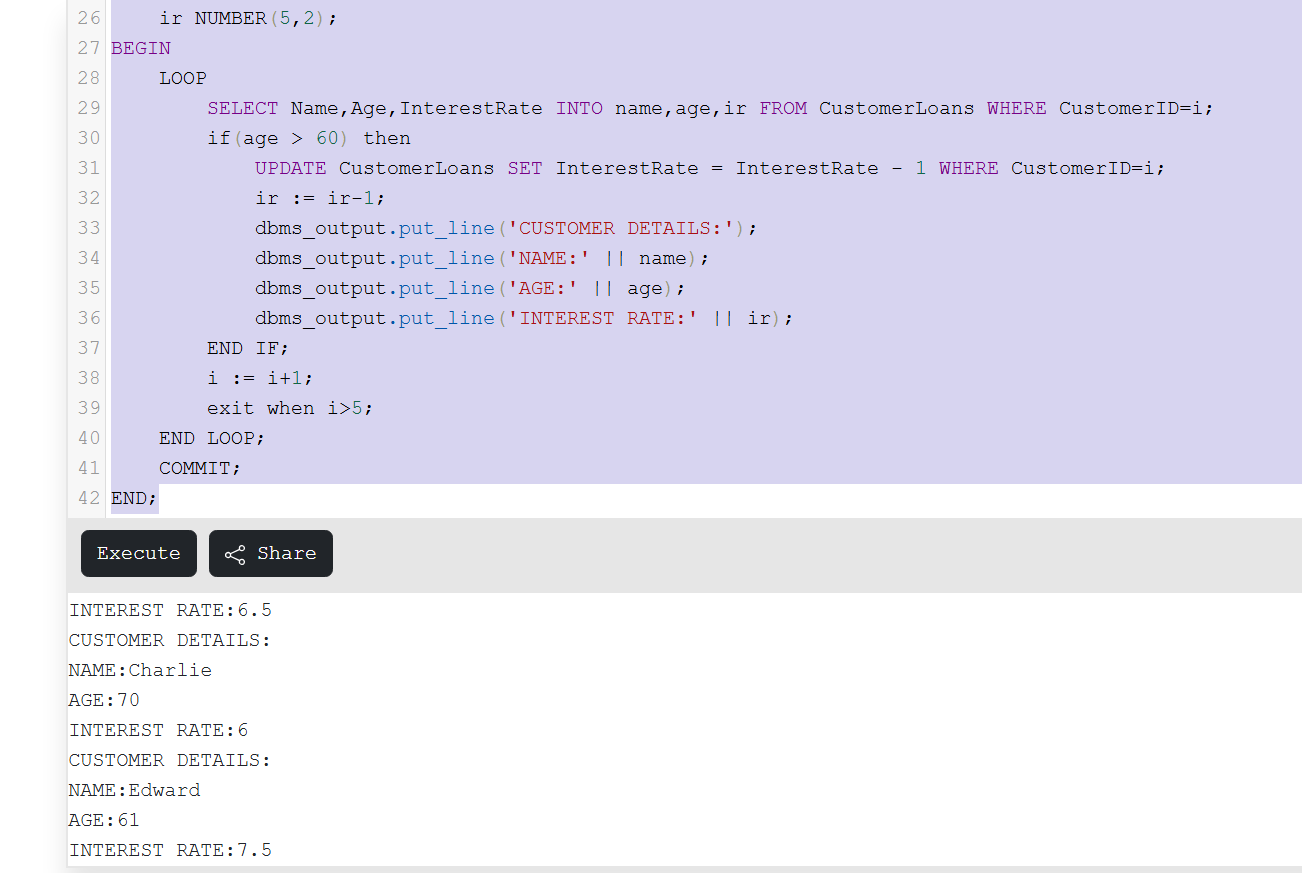
i := i+1;

exit when i>5;

END LOOP;

COMMIT;

END;

Output  
  


**Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

**o Question:** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

CODE

-- Initializing database

CREATE TABLE CustomerLoans (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(50),

Age NUMBER,

Balance NUMBER(10,2),

IsVIP CHAR(1) DEFAULT 'F', -- 'T' or 'F'

InterestRate NUMBER(5,2),

DueDate DATE

);

-- Inserting Sample Records

INSERT INTO CustomerLoans VALUES (1, 'Alice', 65, 15000.00, 'N', 7.5, SYSDATE + 20);

INSERT INTO CustomerLoans VALUES (2, 'Bob', 45, 8000.00, 'N', 8.0, SYSDATE + 40);

INSERT INTO CustomerLoans VALUES (3, 'Charlie', 70, 12000.00, 'N', 7.0, SYSDATE + 15);

INSERT INTO CustomerLoans VALUES (4, 'Diana', 30, 5000.00, 'N', 9.0, SYSDATE + 5);

INSERT INTO CustomerLoans VALUES (5, 'Edward', 61, 11000.00, 'N', 8.5, SYSDATE + 50);

COMMIT;

--PL/SQL Program

DECLARE

i NUMBER := 1;

name VARCHAR2(50);

bal NUMBER(10,2);

flag CHAR(1);

BEGIN

LOOP

SELECT Name,Balance,IsVIP INTO name,bal,flag FROM CustomerLoans WHERE CustomerID=i;

if(bal > 10000) then

UPDATE CustomerLoans SET IsVIP = 'T' WHERE CustomerID = i;

flag := 'T';

dbms\_output.put\_line('CUSTOMER DETAILS:');

dbms\_output.put\_line('NAME:' || name);

dbms\_output.put\_line('BALANCE:' || bal);

dbms\_output.put\_line('IS VIP (T or F):' || flag);

END IF;

i := i+1;

exit when i>5;

END LOOP;

COMMIT;

END;

OUTPUT



**Scenario 3:** Customers should be able to transfer funds between their accounts.

**o Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

Code

-- Initializing database

CREATE TABLE CustomerLoans (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(50),

Age NUMBER,

Balance NUMBER(10,2),

IsVIP CHAR(1) DEFAULT 'F', -- 'T' or 'F'

InterestRate NUMBER(5,2),

DueDate DATE

);

-- Inserting Sample Records

INSERT INTO CustomerLoans VALUES (1, 'Alice', 65, 15000.00, 'N', 7.5, SYSDATE + 20);

INSERT INTO CustomerLoans VALUES (2, 'Bob', 45, 8000.00, 'N', 8.0, SYSDATE + 40);

INSERT INTO CustomerLoans VALUES (3, 'Charlie', 70, 12000.00, 'N', 7.0, SYSDATE + 15);

INSERT INTO CustomerLoans VALUES (4, 'Diana', 30, 5000.00, 'N', 9.0, SYSDATE + 5);

INSERT INTO CustomerLoans VALUES (5, 'Edward', 61, 11000.00, 'N', 8.5, SYSDATE + 50);

COMMIT;

--PL/SQL Program

DECLARE

i NUMBER := 1;

name VARCHAR2(50);

due DATE;

today DATE;

BEGIN

today := SYSDATE;

LOOP

SELECT Name,DueDate INTO name,due FROM CustomerLoans WHERE CustomerID=i;

IF(due - today < 30) then

dbms\_output.put\_line('CUSTOMER DETAILS:');

dbms\_output.put\_line('NAME:' || name);

dbms\_output.put\_line('Dear ' || name || ', Your due date is on ' || due || '. This is a reminder message sent to remind you that your due date is less than 30 days.');

END IF;

i := i+1;

exit when i>5;

END LOOP;

COMMIT;

END;

Output  
  
