**Exercise 1: Control Structures**

**1.1 Code:**

2

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE Loans';

EXECUTE IMMEDIATE 'DROP TABLE Customers';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Age NUMBER,

Balance NUMBER,

IsVIP VARCHAR2(5)

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

InterestRate NUMBER,

DueDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

BEGIN

INSERT INTO Customers VALUES (1, 'Alice', 65, 12000, 'FALSE');

INSERT INTO Customers VALUES (2, 'Bob', 45, 8000, 'FALSE');

INSERT INTO Customers VALUES (3, 'Charlie', 70, 15000, 'FALSE');

INSERT INTO Customers VALUES (4, 'David', 30, 5000, 'FALSE');

INSERT INTO Loans VALUES (101, 1, 7.5, SYSDATE + 10);

INSERT INTO Loans VALUES (102, 2, 8.0, SYSDATE + 40);

INSERT INTO Loans VALUES (103, 3, 6.5, SYSDATE + 5);

INSERT INTO Loans VALUES (104, 4, 9.0, SYSDATE + 25);

END;

/

**-- Scenario 1: Apply 1% discount to interest rate if age > 60**

BEGIN

FOR rec IN (

SELECT L.LoanID, C.Age

FROM Loans L

JOIN Customers C ON L.CustomerID = C.CustomerID

)

LOOP

IF rec.Age > 60 THEN

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE LoanID = rec.LoanID;

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Interest rates updated for customers above 60.');

END;

/

**-- Scenario 2: Set IsVIP = 'TRUE' for balance > 10000**

BEGIN

FOR rec IN (SELECT CustomerID, Balance FROM Customers)

LOOP

IF rec.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = rec.CustomerID;

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('VIP status updated for high balance customers.');

END;

/

**-- Scenario 3: Send reminders for loans due in next 30 days**

BEGIN

FOR rec IN (

SELECT L.LoanID, L.DueDate, C.Name

FROM Loans L

JOIN Customers C ON L.CustomerID = C.CustomerID

WHERE L.DueDate <= SYSDATE + 30

)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || rec.LoanID ||

' for customer ' || rec.Name ||

' is due on ' || TO\_CHAR(rec.DueDate, 'DD-Mon-YYYY'));

END LOOP;

END;

/

COLUMN CustomerID FORMAT 9999 HEADING 'CUSTOMERID';

COLUMN Name FORMAT A15 HEADING 'NAME';

COLUMN Age FORMAT 999 HEADING 'AGE';

COLUMN Balance FORMAT 999999 HEADING 'BALANCE';

COLUMN IsVIP FORMAT A5 HEADING 'ISVIP';

COLUMN LoanID FORMAT 9999 HEADING 'LOANID';

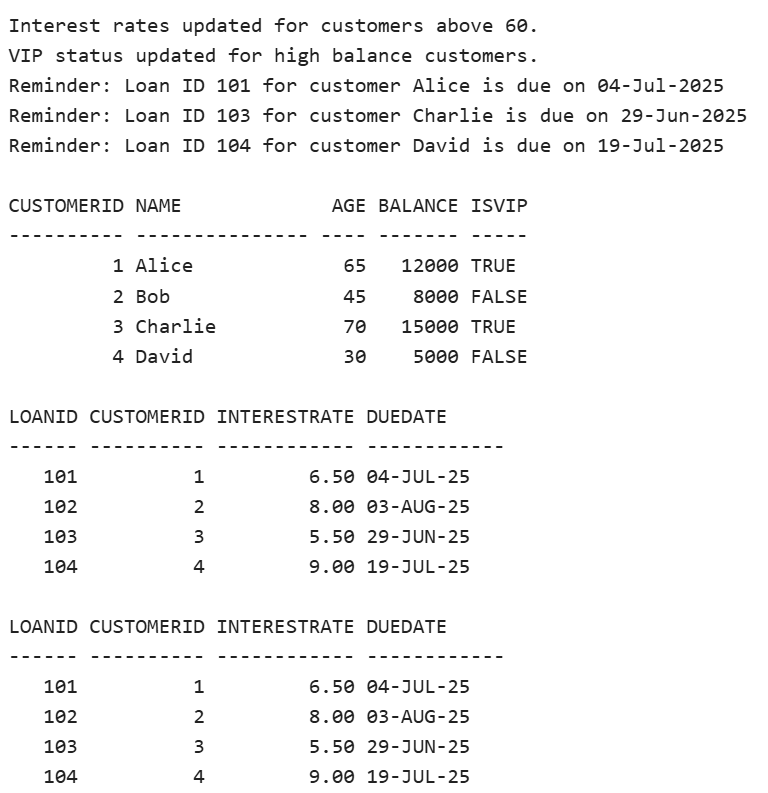
COLUMN InterestRate FORMAT 99.99 HEADING 'INTERESTRATE';

COLUMN DueDate FORMAT A12 HEADING 'DUEDATE';

SELECT \* FROM Customers;

SELECT \* FROM Loans;

**1.2 Output:**



**Exercise 3: Stored Procedures**

**3.1 Code:**

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE accounts';

EXECUTE IMMEDIATE 'DROP TABLE employees';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

customer\_name VARCHAR2(50),

account\_type VARCHAR2(20),

balance NUMBER

);

INSERT INTO accounts VALUES (1, 'John Doe', 'Savings', 10000);

INSERT INTO accounts VALUES (2, 'Jane Smith', 'Savings', 5000);

INSERT INTO accounts VALUES (3, 'John Doe', 'Current', 2000);

INSERT INTO accounts VALUES (4, 'Jane Smith', 'Current', 1000);

CREATE TABLE employees (

emp\_id NUMBER PRIMARY KEY,

emp\_name VARCHAR2(50),

department VARCHAR2(30),

salary NUMBER

);

INSERT INTO employees VALUES (101, 'Alice', 'HR', 50000);

INSERT INTO employees VALUES (102, 'Bob', 'HR', 60000);

INSERT INTO employees VALUES (103, 'Charlie', 'IT', 70000);

COMMIT;

**-- Scenario 1: ProcessMonthlyInterest**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE accounts

SET balance = balance \* 1.01

WHERE account\_type = 'Savings';

DBMS\_OUTPUT.PUT\_LINE('Monthly interest applied to all savings accounts.');

END;

/

BEGIN

ProcessMonthlyInterest;

END;

/

**-- Scenario 2: UpdateEmployeeBonus**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

dept\_name IN VARCHAR2,

bonus\_percent IN NUMBER

) IS

BEGIN

UPDATE employees

SET salary = salary + (salary \* bonus\_percent / 100)

WHERE department = dept\_name;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to department: ' || dept\_name);

END;

/

BEGIN

UpdateEmployeeBonus('HR', 10);

END;

/

**-- Scenario 3: TransferFunds**

CREATE OR REPLACE PROCEDURE TransferFunds (

from\_acc IN NUMBER,

to\_acc IN NUMBER,

amount IN NUMBER

) IS

from\_balance NUMBER;

BEGIN

SELECT balance INTO from\_balance FROM accounts WHERE account\_id = from\_acc;

IF from\_balance < amount THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Insufficient funds in source account.');

ELSE

UPDATE accounts SET balance = balance - amount WHERE account\_id = from\_acc;

UPDATE accounts SET balance = balance + amount WHERE account\_id = to\_acc;

DBMS\_OUTPUT.PUT\_LINE('Transfer successful from Account ' || from\_acc || ' to Account ' || to\_acc);

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Error: One or both account IDs not found.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Unexpected error: ' || SQLERRM);

END;

/

BEGIN

TransferFunds(1, 3, 1000);

END;

/

DECLARE

CURSOR c\_accounts IS

SELECT account\_id, customer\_name, account\_type, balance FROM accounts;

r\_accounts c\_accounts%ROWTYPE;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(CHR(10) || 'ACCOUNTS SUMMARY:');

DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------');

DBMS\_OUTPUT.PUT\_LINE('ID | CUSTOMER NAME | ACCOUNT TYPE | BALANCE');

DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------');

FOR r\_accounts IN c\_accounts LOOP

DBMS\_OUTPUT.PUT\_LINE(

RPAD(r\_accounts.account\_id, 4) || '|' ||

RPAD(r\_accounts.customer\_name, 20) || '|' ||

RPAD(r\_accounts.account\_type, 14) || '|' ||

TO\_CHAR(r\_accounts.balance, '99999.99')

);

END LOOP;

END;

/

DECLARE

CURSOR c\_employees IS

SELECT emp\_id, emp\_name, department, salary FROM employees;

r\_employees c\_employees%ROWTYPE;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(CHR(10) || 'EMPLOYEES SUMMARY:');

DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------');

DBMS\_OUTPUT.PUT\_LINE('ID | EMPLOYEE NAME | DEPARTMENT | SALARY');

DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------');

FOR r\_employees IN c\_employees LOOP

DBMS\_OUTPUT.PUT\_LINE(

RPAD(r\_employees.emp\_id, 4) || '|' ||

RPAD(r\_employees.emp\_name, 20) || '|' ||

RPAD(r\_employees.department, 15) || '|' ||

TO\_CHAR(r\_employees.salary, '99999.99')

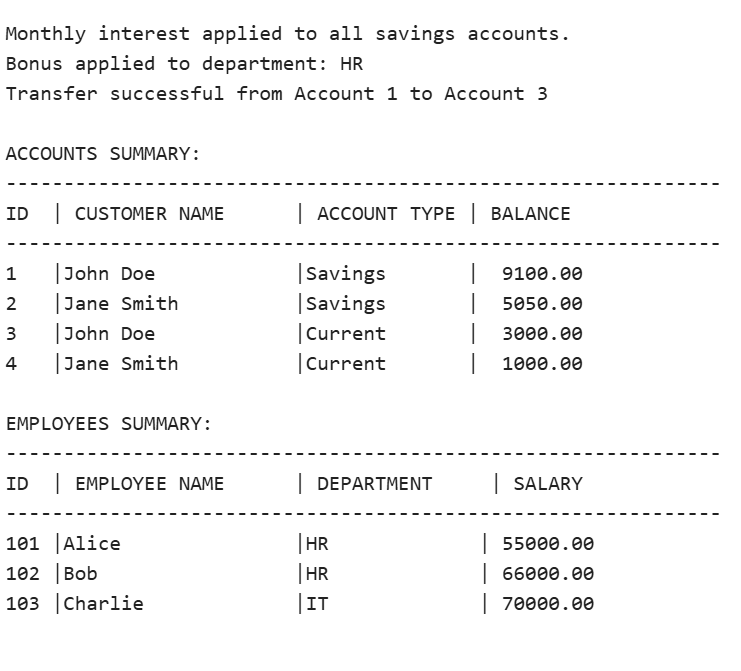
);

END LOOP;

END;

/

**3.2 Output:**

****