**NAME: ABINAYA T S**

**SUPERSET\_ID: 6412256**

**WEEK 2: PL/SQL EXERCISE**

**Exercise 1: Control Structures**

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

* + **Question:** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

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

* + **Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

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

* + **Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

SAMPLE CODE:

-- Drop tables if they already exist (for re-runnable script)

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE loans';

EXECUTE IMMEDIATE 'DROP TABLE customers';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

-- Create customers table

CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER,

isvip VARCHAR2(5)

);

-- Create loans table

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

interest\_rate NUMBER,

due\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

-- Insert sample data

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 loans VALUES (101, 1, 5.5, SYSDATE + 20); -- Alice

INSERT INTO loans VALUES (102, 2, 6.2, SYSDATE + 45); -- Bob

INSERT INTO loans VALUES (103, 3, 7.0, SYSDATE + 10); -- Charlie

COMMIT;

-- Main PL/SQL Block

BEGIN

-- Scenario 1: Apply 1% Interest Discount for Customers Over 60

FOR rec1 IN (

SELECT l.loan\_id

FROM loans l

JOIN customers c ON l.customer\_id = c.customer\_id

WHERE c.age > 60

)

LOOP

UPDATE loans

SET interest\_rate = interest\_rate - 1

WHERE loan\_id = rec1.loan\_id;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('✔ Interest rate discount applied for customers over 60.');

-- Scenario 2: Promote to VIP Based on Balance

FOR rec2 IN (

SELECT customer\_id

FROM customers

WHERE balance > 10000

)

LOOP

UPDATE customers

SET isvip = 'TRUE'

WHERE customer\_id = rec2.customer\_id;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('✔ VIP status updated for customers with balance > $10,000.');

-- Scenario 3: Reminders for Loans Due in Next 30 Days

FOR rec3 IN (

SELECT c.name AS customer\_name, l.loan\_id, l.due\_date

FROM loans l

JOIN customers c ON l.customer\_id = c.customer\_id

WHERE l.due\_date <= SYSDATE + 30

)

LOOP

DBMS\_OUTPUT.PUT\_LINE(

'📅 Reminder: Dear ' || rec3.customer\_name ||

', your loan (ID: ' || rec3.loan\_id ||

') is due on ' || TO\_CHAR(rec3.due\_date, 'DD-MON-YYYY')

);

END LOOP;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('✅ All updates and reminders processed successfully.');

END;

/  
  
  
  
OUTPUT:  
  
