**Exercise 1: Control Structures**

**Step 1: Create the tables**

CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

customer\_name VARCHAR2(50),

age NUMBER,

balance NUMBER,

isvip VARCHAR2(5)

);

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)

);

**Step 2: Insert sample data**

INSERT INTO customers VALUES (1, 'John', 65, 9500, 'FALSE');

INSERT INTO customers VALUES (2, 'Alice', 55, 11000, 'FALSE');

INSERT INTO customers VALUES (3, 'Mark', 70, 5000, 'FALSE');

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

INSERT INTO loans VALUES (102, 2, 12, SYSDATE + 40);

INSERT INTO loans VALUES (103, 3, 11, SYSDATE + 20);

COMMIT;

SELECT \* FROM CUSTOMERS;

CUSTOMER\_ID CUSTOMER\_NAME AGE BALANCE ISVIP

-------------------- -------------------------- -------- ---------------- ----------

1 John 65 9500 FALSE

2 Alice 55 11000 FALSE

3 Mark 70 5000 FALSE

SELECT \* FROM LOANS;

LOAN\_ID CUSTOMER\_ID INTEREST\_RATE DUE\_DATE

---------- ----------- ------------- ---------

101 1 10 08-JUL-25

102 2 12 07-AUG-25

103 3 11 18-JUL-25

**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.

SET SERVEROUTPUT ON;

DECLARE

CURSOR loan\_cursor IS

SELECT L.CUSTOMER\_ID, L.INTEREST\_RATE

FROM LOANS L

JOIN CUSTOMERS C ON L.CUSTOMER\_ID = C.CUSTOMER\_ID

WHERE C.AGE > 60;

BEGIN

FOR rec IN loan\_cursor LOOP

UPDATE LOANS

SET INTEREST\_RATE = INTEREST\_RATE - 1

WHERE CUSTOMER\_ID = rec.CUSTOMER\_ID;

END LOOP;

COMMIT;

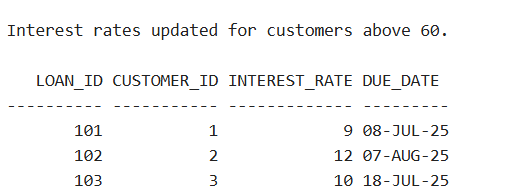
DBMS\_OUTPUT.PUT\_LINE('Interest rates updated for customers above 60.');

END;

/

* SELECT \* FROM LOANS;

**OUPUT:**



**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.

SET SERVEROUTPUT ON;

DECLARE

CURSOR customer\_cursor IS

SELECT CUSTOMER\_ID, BALANCE

FROM CUSTOMERS

WHERE BALANCE > 10000;

BEGIN

FOR rec IN customer\_cursor LOOP

-- Set ISVIP flag to 'TRUE'

UPDATE CUSTOMERS

SET ISVIP = 'TRUE'

WHERE CUSTOMER\_ID = rec.CUSTOMER\_ID;

END LOOP;

COMMIT;

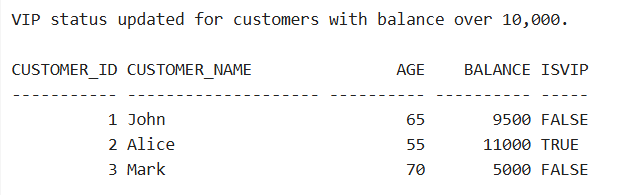
DBMS\_OUTPUT.PUT\_LINE('VIP status updated for customers with balance over 10,000.');

END;

/

* SELECT \* FROM CUSTOMERS;

**OUTPUT:**



**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.

SET SERVEROUTPUT ON;

DECLARE

CURSOR due\_loans\_cursor IS

SELECT CUSTOMER\_ID, DUE\_DATE

FROM LOANS

WHERE DUE\_DATE BETWEEN SYSDATE AND SYSDATE + 30;

BEGIN

FOR rec IN due\_loans\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Customer ' || rec.CUSTOMER\_ID ||

' has a loan due on ' || TO\_CHAR(rec.DUE\_DATE, 'DD-MON-YYYY'));

END LOOP;

END;

/

**OUPUT:**

Reminder: Customer 1 has a loan due on 08-JUL-2025

Reminder: Customer 3 has a loan due on 18-JUL-2025