WEEK: 2

**Exercise 1: Control Structures**

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

Table Creation:

CREATE TABLE CUSTOMERS (

CUSTOMER\_ID NUMBER PRIMARY KEY,

AGE NUMBER,

NTEREST\_RATE NUMBER(5,2),

BALANCE NUMBER,

STATUS VARCHAR2(20)

);

INSERT INTO CUSTOMERS VALUES (1, 65, 8.50, 500000, 'REGULAR');

INSERT INTO CUSTOMERS VALUES (2, 45, 9.00, 2000000, 'REGULAR');

INSERT INTO CUSTOMERS VALUES (3, 70, 7.75, 300000, 'REGULAR');

INSERT INTO CUSTOMERS VALUES (4, 62, 10.00, 1500000, 'REGULAR');

INSERT INTO CUSTOMERS VALUES (5, 30, 11.00, 100000, 'REGULAR');

COMMIT;

SET SERVEROUTPUT ON;

DECLARE

CURSOR cust\_cur IS

SELECT CUSTOMER\_ID, AGE, INTEREST\_RATE

FROM CUSTOMERS

WHERE AGE > 60;

BEGIN

FOR cust\_rec IN cust\_cur LOOP

UPDATE CUSTOMERS

SET LOAN\_INTEREST\_RATE = INTEREST\_RATE \* 0.99

WHERE CUSTOMER\_ID = cust\_rec.CUSTOMER\_ID;

DBMS\_OUTPUT.PUT\_LINE('Applied 1% discount for Customer ID: ' || cust\_rec.CUSTOMER\_ID);

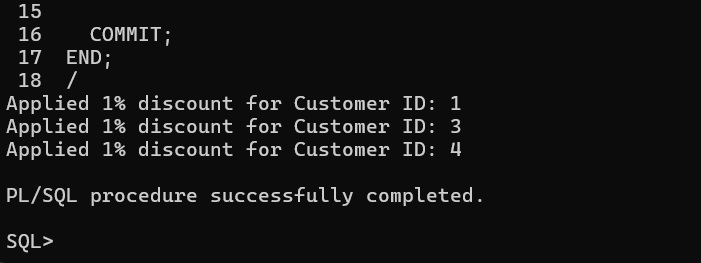
END LOOP;

COMMIT;

END;

/

Output:



**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.

CREATE TABLE CUSTOMERS (

CUSTOMER\_ID NUMBER PRIMARY KEY,

NAME VARCHAR2(50),

AGE NUMBER,

BALANCE NUMBER(12,2),

ISVIP CHAR(1) DEFAULT 'N'

)

INSERT INTO CUSTOMERS (CUSTOMER\_ID, NAME, AGE, BALANCE) VALUES (1, 'Harshita, 65, 5000);

INSERT INTO CUSTOMERS (CUSTOMER\_ID, NAME, AGE, BALANCE) VALUES (2, 'Yagnya', 45, 25000);

INSERT INTO CUSTOMERS (CUSTOMER\_ID, NAME, AGE, BALANCE) VALUES (3, 'Uday', 70, 15000);

INSERT INTO CUSTOMERS (CUSTOMER\_ID, NAME, AGE, BALANCE) VALUES (4, 'Rahul', 30, 8000);

INSERT INTO CUSTOMERS (CUSTOMER\_ID, NAME, AGE, BALANCE) VALUES (5, 'Phani', 55, 12000);

COMMIT;

SET SERVEROUTPUT ON;

DECLARE

CURSOR vip\_cur IS

SELECT CUSTOMER\_ID, BALANCE

FROM CUSTOMERS;

BEGIN

FOR rec IN vip\_cur LOOP

IF rec.BALANCE > 10000 THEN

UPDATE CUSTOMERS

SET ISVIP = 'Y'

WHERE CUSTOMER\_ID = rec.CUSTOMER\_ID;

DBMS\_OUTPUT.PUT\_LINE('Promoted Customer ID: ' || rec.CUSTOMER\_ID || ' to VIP');

END IF;

END LOOP;

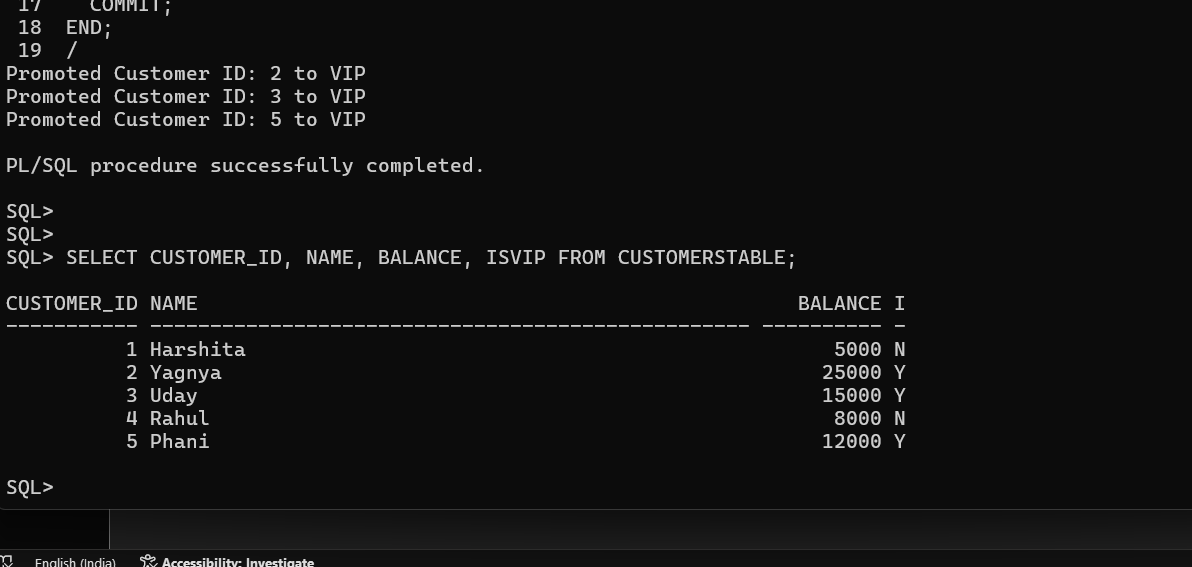
COMMIT;

END;

/

SELECT CUSTOMER\_ID, NAME, BALANCE, ISVIP 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.

CREATE TABLE LOANS (

LOAN\_ID NUMBER PRIMARY KEY,

CUSTOMER\_ID NUMBER,

DUE\_DATE DATE

);

INSERT INTO LOANS (LOAN\_ID, CUSTOMER\_ID, DUE\_DATE) VALUES (1, 101, SYSDATE + 10);

INSERT INTO LOANS (LOAN\_ID, CUSTOMER\_ID, DUE\_DATE) VALUES (2, 102, SYSDATE + 25);

INSERT INTO LOANS (LOAN\_ID, CUSTOMER\_ID, DUE\_DATE) VALUES (3, 103, SYSDATE + 40);

INSERT INTO LOANS (LOAN\_ID, CUSTOMER\_ID, DUE\_DATE) VALUES (4, 104, SYSDATE - 5);

INSERT INTO LOANS (LOAN\_ID, CUSTOMER\_ID, DUE\_DATE) VALUES (5, 105, SYSDATE + 30);

COMMIT;

SET SERVEROUTPUT ON;

DECLARE

CURSOR due\_loans\_cur IS

SELECT loan\_id, customer\_id, due\_date

FROM loans

WHERE due\_date BETWEEN SYSDATE AND SYSDATE + 30;

v\_loan\_id LOANS.LOAN\_ID%TYPE;

v\_customer\_id LOANS.CUSTOMER\_ID%TYPE;

v\_due\_date LOANS.DUE\_DATE%TYPE;

BEGIN

OPEN due\_loans\_cur;

LOOP

FETCH due\_loans\_cur INTO v\_loan\_id, v\_customer\_id, v\_due\_date;

EXIT WHEN due\_loans\_cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(

'Reminder: Customer ' || v\_customer\_id ||

', your loan ' || v\_loan\_id ||

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

);

END LOOP;

CLOSE due\_loans\_cur;

END;

/

DROP TABLE LOANS;

Output:

