**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.

**Query**

1. **Create the Table**

CREATE TABLE CUSTOMERS (

CUSTOMER\_ID NUMBER PRIMARY KEY,

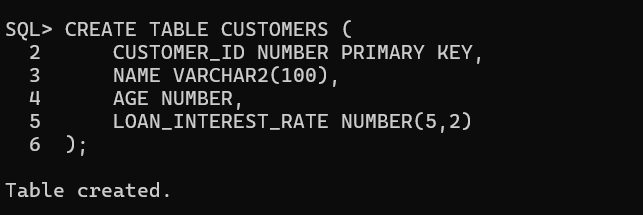
NAME VARCHAR2(100),

AGE NUMBER,

LOAN\_INTEREST\_RATE NUMBER(5,2)

);

**Output**

****

**Query**

1. **Insert Sample Data**

INSERT INTO CUSTOMERS VALUES (1, 'Alice', 45, 10.5);

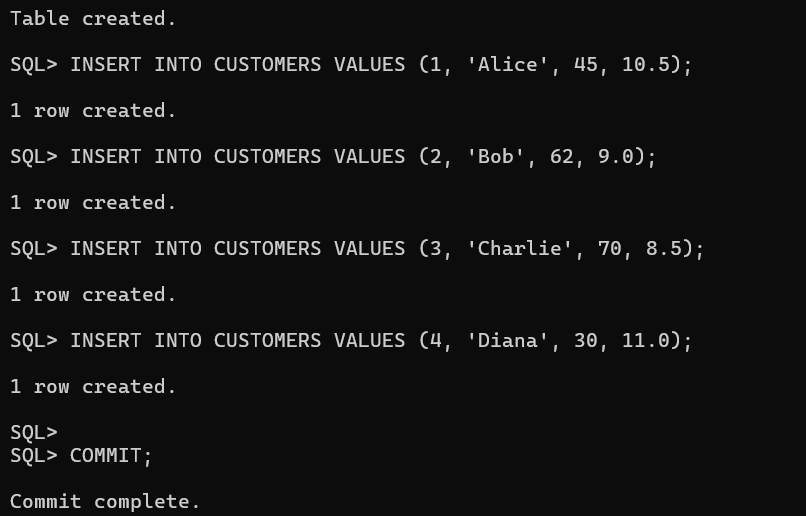
INSERT INTO CUSTOMERS VALUES (2, 'Bob', 62, 9.0);

INSERT INTO CUSTOMERS VALUES (3, 'Charlie', 70, 8.5);

INSERT INTO CUSTOMERS VALUES (4, 'Diana', 30, 11.0);

COMMIT;

**Output**



**Query**

1. **PL/SQL Block to Apply 1% Discount**

DECLARE

CURSOR cust\_cursor IS

SELECT CUSTOMER\_ID, AGE, LOAN\_INTEREST\_RATE

FROM CUSTOMERS

FOR UPDATE;

BEGIN

FOR cust\_rec IN cust\_cursor LOOP

IF cust\_rec.AGE > 60 THEN

UPDATE CUSTOMERS

SET LOAN\_INTEREST\_RATE = LOAN\_INTEREST\_RATE - 1

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

END IF;

END LOOP;

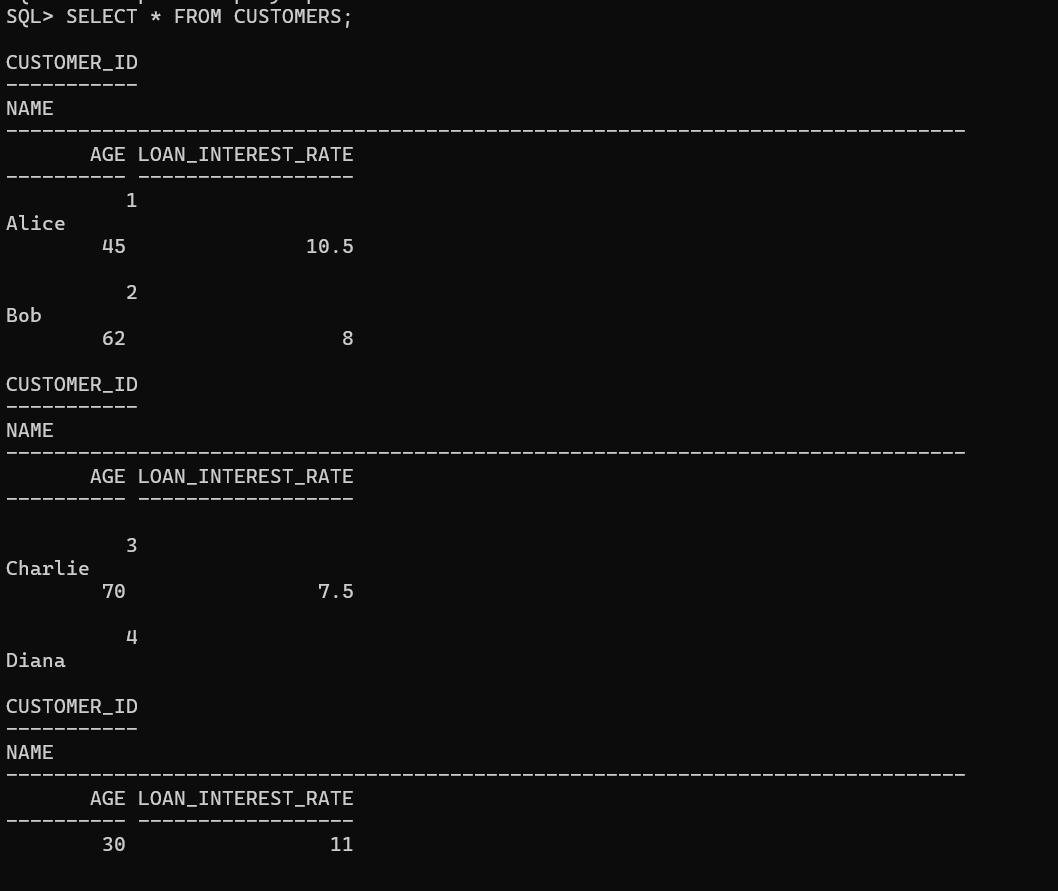
COMMIT;

END;

/

SQL> SELECT \* FROM CUSTOMERS;

**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.

**Query**

1. **Create Table**

CREATE TABLE CUSTOMERS (

CUSTOMER\_ID NUMBER PRIMARY KEY,

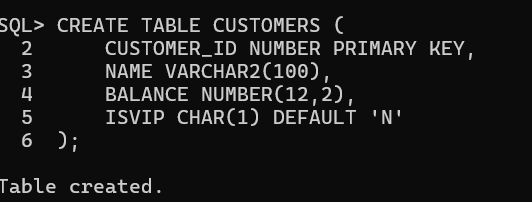
NAME VARCHAR2(100),

BALANCE NUMBER(12,2),

ISVIP CHAR(1) DEFAULT 'N'

);

**Output**



**Query**

1. **Insert Sample Data**

INSERT INTO CUSTOMERS VALUES (1, 'Alice', 9500.00, 'N');

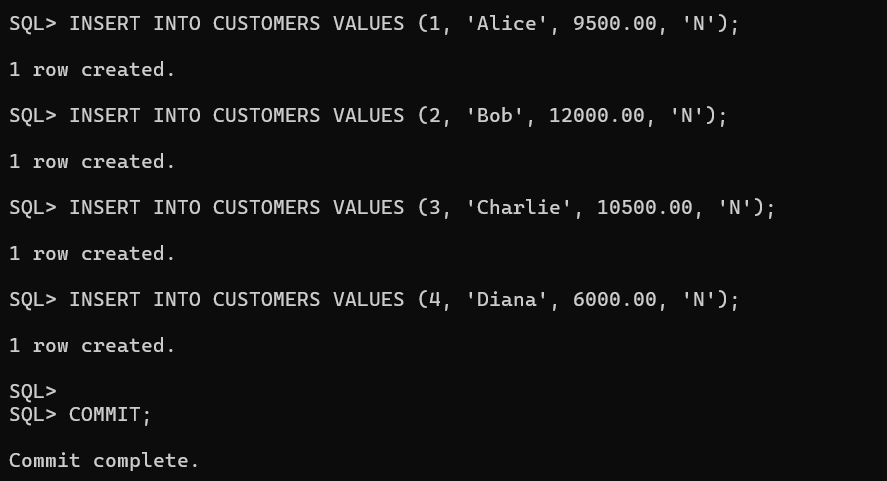
INSERT INTO CUSTOMERS VALUES (2, 'Bob', 12000.00, 'N');

INSERT INTO CUSTOMERS VALUES (3, 'Charlie', 10500.00, 'N');

INSERT INTO CUSTOMERS VALUES (4, 'Diana', 6000.00, 'N');

COMMIT;

**Output**



**Query**

**🔹 3. PL/SQL Block to Update VIP Status**

BEGIN

FOR cust\_rec IN (

SELECT CUSTOMER\_ID, BALANCE

FROM CUSTOMERS

WHERE BALANCE > 10000

FOR UPDATE

) LOOP

UPDATE CUSTOMERS

SET ISVIP = 'Y'

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

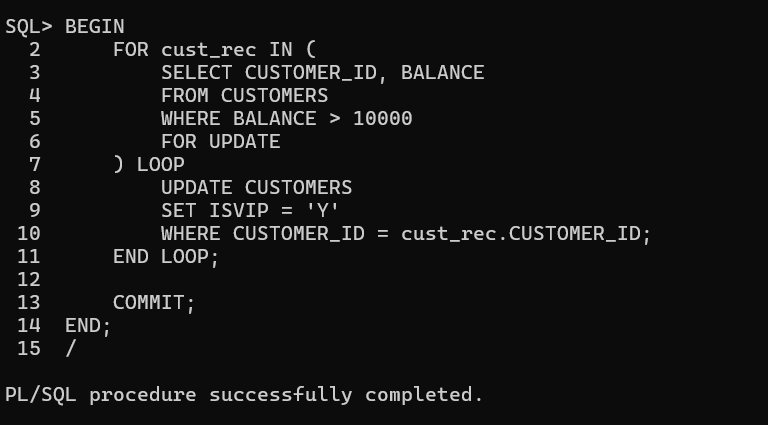
END LOOP;

COMMIT;

END;

/

**Output**

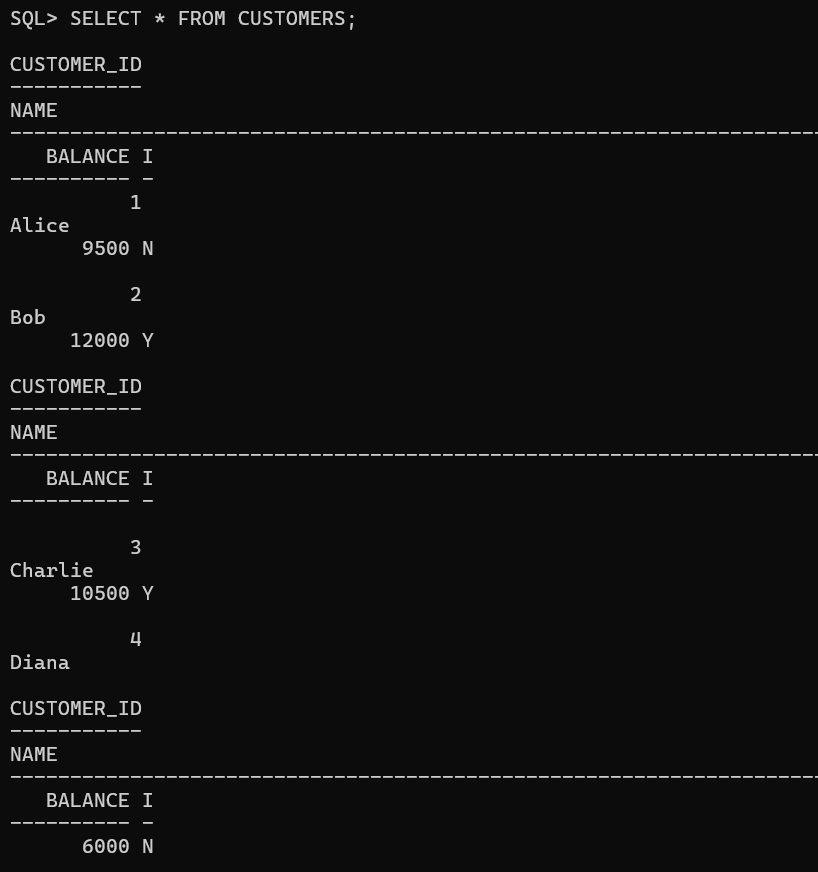


**Query**

**4.View Updated Table**

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.

**Query**

1. **Create Table**

CREATE TABLE LOANS (

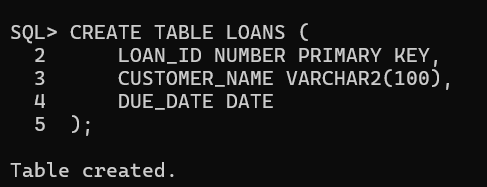
LOAN\_ID NUMBER PRIMARY KEY,

CUSTOMER\_NAME VARCHAR2(100),

DUE\_DATE DATE

);

**Output**



**Query**

1. **Insert Sample Data**

INSERT INTO LOANS VALUES (1, 'Alice', SYSDATE + 5);

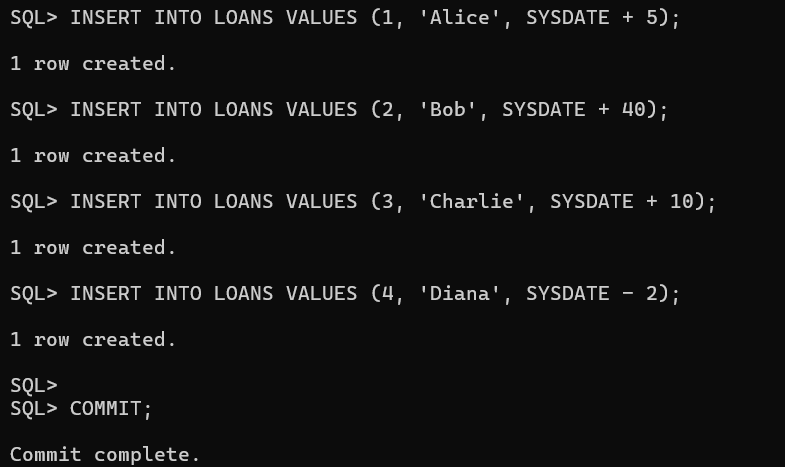
INSERT INTO LOANS VALUES (2, 'Bob', SYSDATE + 40);

INSERT INTO LOANS VALUES (3, 'Charlie', SYSDATE + 10);

INSERT INTO LOANS VALUES (4, 'Diana', SYSDATE - 2);

COMMIT;

**Output**



**Query**

1. **PL/SQL Block to Print Reminders**

BEGIN

FOR loan\_rec IN (

SELECT CUSTOMER\_NAME, DUE\_DATE

FROM LOANS

WHERE DUE\_DATE BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE(

'Reminder: Loan for ' || loan\_rec.CUSTOMER\_NAME ||

' is due on ' || TO\_CHAR(loan\_rec.DUE\_DATE, 'DD-MON-YYYY')

);

END LOOP;

END;

/

SET SERVEROUTPUT ON;

**Output**

