**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

**Solution**

First, we create two tables, load some sample data

CREATE TABLE CUSTOMER (

CUSTOMER\_ID NUMBER PRIMARY KEY,

NAME VARCHAR2(50),

AGE NUMBER,

LOAN\_INTEREST\_RATE NUMBER(5,2),

BALANCE NUMBER(10,2),

ISVIP VARCHAR2(1) DEFAULT 'N'

);

-- LOAN table holds due dates

CREATE TABLE LOAN (

LOAN\_ID NUMBER PRIMARY KEY,

CUSTOMER\_ID NUMBER,

DUE\_DATE DATE,

FOREIGN KEY (CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID)

);

-- Sample data

INSERT INTO CUSTOMER VALUES (1, 'Alice', 65, 7.50, 5000, 'N');

INSERT INTO CUSTOMER VALUES (2, 'Bob', 58, 8.00, 15000, 'N');

INSERT INTO CUSTOMER VALUES (3, 'Carol', 72, 7.75, 20000, 'N');

INSERT INTO CUSTOMER VALUES (4, 'David', 45, 6.90, 8000, 'N');

INSERT INTO CUSTOMER VALUES (5, 'Eve', 61, 8.25, 12000, 'N');

INSERT INTO LOAN VALUES (101, 1, SYSDATE + 10); -- due in 10 days

INSERT INTO LOAN VALUES (102, 2, SYSDATE + 40); -- due in 40 days

INSERT INTO LOAN VALUES (103, 3, SYSDATE + 25); -- due in 25 days

INSERT INTO LOAN VALUES (104, 4, SYSDATE + 5); -- due in 5 days

INSERT INTO LOAN VALUES (105, 5, SYSDATE - 2); -- already due

COMMIT;

**Scenario 1: 1% discount on loan interest for customers > 60**

BEGIN

FOR rec IN (SELECT CUSTOMER\_ID, NAME, AGE, LOAN\_INTEREST\_RATE

FROM CUSTOMER)

LOOP

IF rec.AGE > 60 THEN

UPDATE CUSTOMER

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

WHERE CUSTOMER\_ID = rec.CUSTOMER\_ID;

DBMS\_OUTPUT.PUT\_LINE(

'Applied 1% discount to ' || rec.NAME

|| ' (ID=' || rec.CUSTOMER\_ID

|| '), new rate: '

|| TO\_CHAR(rec.LOAN\_INTEREST\_RATE - 1, '90.00') || '%'

);

END IF;

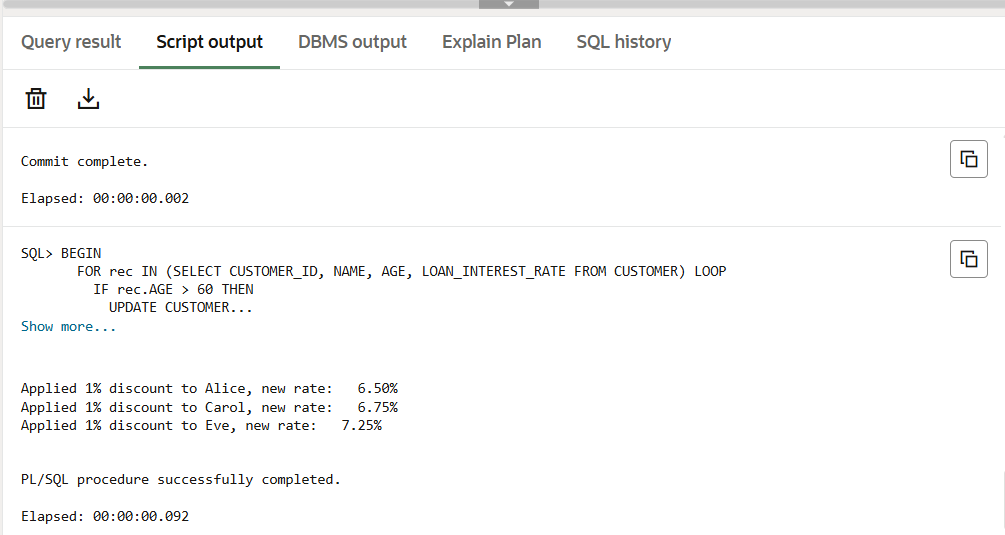
END LOOP;

COMMIT;

END;

/

**OUTPUT**

****

**Scenario 2: Promote to VIP if balance > 10,000**

BEGIN

FOR rec IN (SELECT CUSTOMER\_ID, NAME, BALANCE

FROM CUSTOMER)

LOOP

IF rec.BALANCE > 10000 THEN

UPDATE CUSTOMER

SET ISVIP = 'Y'

WHERE CUSTOMER\_ID = rec.CUSTOMER\_ID;

DBMS\_OUTPUT.PUT\_LINE(

'Set VIP for ' || rec.NAME

|| ' (ID=' || rec.CUSTOMER\_ID

|| '), balance: $' || TO\_CHAR(rec.BALANCE, '999999.99')

);

END IF;

END LOOP;

COMMIT;

END;

/

**OUTPUT**

A screenshot of a computer

AI-generated content may be incorrect.

**Scenario 3: Reminders for loans due in next 30 days**

BEGIN

FOR loan\_rec IN (

SELECT L.LOAN\_ID, L.CUSTOMER\_ID, L.DUE\_DATE, C.NAME

FROM LOAN L

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

WHERE L.DUE\_DATE BETWEEN SYSDATE AND SYSDATE + 30

ORDER BY L.DUE\_DATE

)

LOOP

DBMS\_OUTPUT.PUT\_LINE(

'REMINDER: Loan ' || loan\_rec.LOAN\_ID

|| ' for ' || loan\_rec.NAME

|| ' (CustID=' || loan\_rec.CUSTOMER\_ID

|| ') is due on '

|| TO\_CHAR(loan\_rec.DUE\_DATE, 'DD‑Mon‑YYYY')

);

END LOOP;

END;

/

**OUTPUT**

**A screenshot of a computer

AI-generated content may be incorrect.**