**Exercise 1: Control Structures**

Scenario 1:

Code:

DECLARE

cursor cur is select customerid, trunc(months\_between(sysdate,dob)/12) as age from customers;

x Loans.customerid%type;

y number;

BEGIN

    open cur;

    LOOP

        FETCH cur into x,y;

        EXIT WHEN cur%NOTFOUND;

        if y>60 then

            update loans set interestrate=interestrate-1 where customerid=x;

            dbms\_output.put\_line('Intrest rate for '||x||' is decreased by 1% as age is '||y);

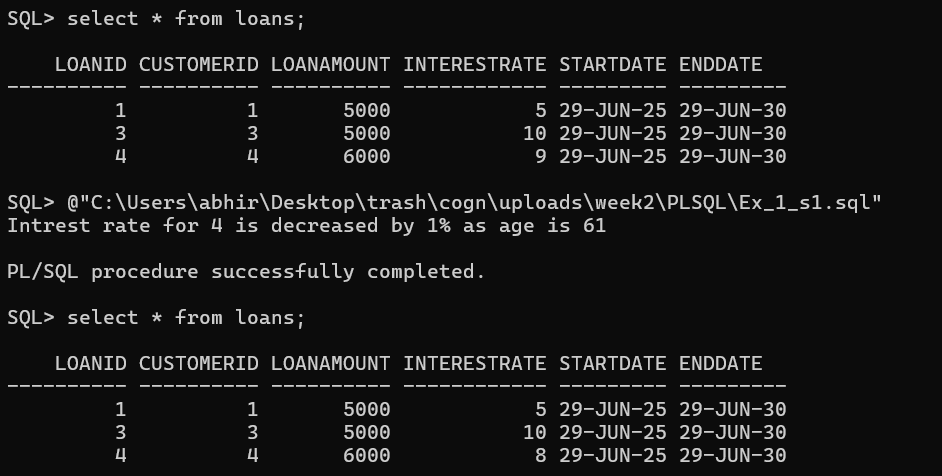
        end if;

    end loop;

end;

/

Output:



Scenario 2:

Code:

ALTER TABLE Customers ADD IsVIP CHAR(1) DEFAULT 'N';

DECLARE

    CURSOR cur is select customerid , balance from customers;

    x customers.CUSTOMERID%TYPE;

    y customers.BALANCE%TYPE;

BEGIN

    open cur;

    loop

        fetch cur into x,y;

        exit when cur%notfound;

        if y > 10000 THEN

        UPDATE CUSTOMERS set ISVIP='Y' where CUSTOMERID = x;

        dbms\_output.put\_line('VIP member '||x || ' balance '||y);

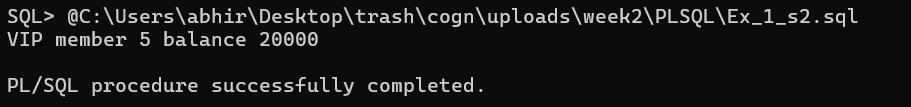
        end if;

    end loop;

end;

/

Output:



Scenario 3:

Code:

DECLARE

    CURSOR cur IS

    SELECT l.LoanID, l.customerid, l.EndDate, c.Name FROM Loans l JOIN Customers c ON l.customerid=c.customerid

    WHERE l.EndDate <= SYSDATE + 30;

    lid Loans.LoanID%TYPE;

    cid Loans.customerid%TYPE;

    edt Loans.EndDate%TYPE;

    nm Customers.Name%TYPE;

BEGIN

    FOR rec IN cur LOOP

    lid := rec.LoanID;

    cid := rec.customerid;

    edt := rec.EndDate;

    nm := rec.Name;

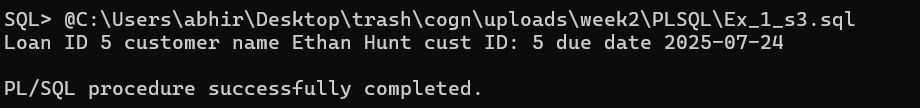
    dbms\_output.put\_line('Loan ID '||lid||' customer name '||nm||' cust ID: '||cid||' due date '||TO\_CHAR(edt,'YYYY-MM-DD'));

    END LOOP;

END;

/

Output:



**Exercise 3: Stored Procedures**

Scenario 1:

Code:

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

    UPDATE accounts SET balance = balance + (balance \* 0.01)

    WHERE accountType = 'Savings';

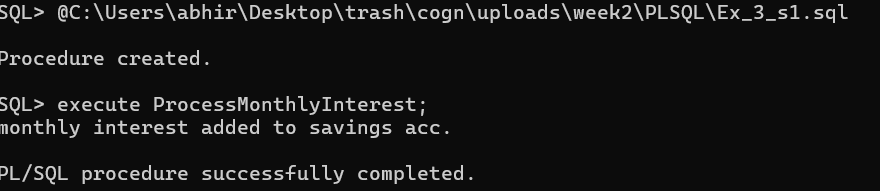
    COMMIT;

    dbms\_output.put\_line('monthly interest added to savings acc.');

END;

/

Output:



Scenario 2:

Code:

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (dept IN VARCHAR2,bonus IN NUMBER) IS

BEGIN

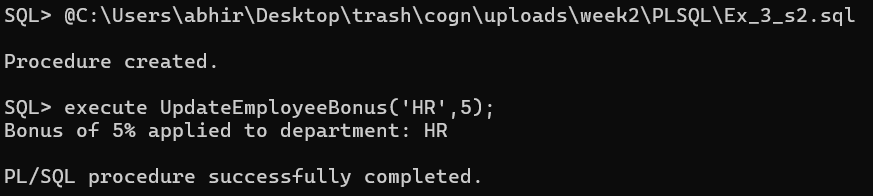
    UPDATE employees SET salary=salary+(salary\*bonus/100) WHERE department = dept;

    dbms\_output.put\_line('Bonus of '||bonus||'% applied to department: '||dept);

END;

/

Output:



Scenario 3:

Code:

CREATE OR REPLACE PROCEDURE TransferFunds(srcid IN NUMBER,destid IN NUMBER,amt IN NUMBER) IS

bal Accounts.Balance%TYPE;

srcCid Accounts.CustomerID%TYPE;

destCid Accounts.CustomerID%TYPE;

BEGIN

    IF amt <= 0 THEN

        RAISE\_APPLICATION\_ERROR(-20001, 'cannot transfer negative amount');

    END IF;

    SELECT balance, CustomerID INTO bal, srcCid FROM Accounts WHERE AccountID = srcid;

    SELECT CustomerID INTO destCid FROM Accounts WHERE AccountID = destid;

    IF bal < amt THEN

        RAISE\_APPLICATION\_ERROR(-20002, 'Insufficient funds');

    END IF;

    IF srcCid <> destCid THEN

        UPDATE Customers SET balance = balance - amt  WHERE CustomerID = srcCid;

        UPDATE Customers SET balance = balance + amt WHERE CustomerID = destCid;

    END IF;

    COMMIT;

    DBMS\_OUTPUT.PUT\_LINE('Transferred '||amt||' from '||srcid||' to '||destid);

EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

        RAISE\_APPLICATION\_ERROR(-20003, 'Invalid account provided.');

    WHEN OTHERS THEN

        ROLLBACK;

    RAISE;

END;

/

Output:

