**Exercise 1: Control Structures**

**Creation of Schemas and Insertion of Values:**

CREATE TABLE Customers (

CustomerID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

DOB DATE,

Balance NUMBER,

LastModified DATE

);

CREATE TABLE Loans (

LoanID NUMBER PRIMARY KEY,

CustomerID NUMBER,

LoanAmount NUMBER,

InterestRate NUMBER,

StartDate DATE,

EndDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

INSERT INTO CUSTOMERS VALUES(1, 'Manoj', TO\_DATE('1950-06-01','YYYY-MM-DD'),15000,SYSDATE);

INSERT INTO CUSTOMERS VALUES(2, 'Ravi', TO\_DATE('1980-05-01','YYYY-MM-DD'),9500,SYSDATE);

INSERT INTO CUSTOMERS VALUES(3, 'Geetha', TO\_DATE('1972-03-01','YYYY-MM-DD'),11000,SYSDATE);

INSERT INTO CUSTOMERS VALUES(4, 'Arjun', TO\_DATE('1995-01-01','YYYY-MM-DD'),3000,SYSDATE);

INSERT INTO LOANS VALUES (101, 1, 100000, 8.5, SYSDATE-200, SYSDATE+20);

INSERT INTO LOANS VALUES (102, 2, 50000, 9.0, SYSDATE-100, SYSDATE+45);

INSERT INTO LOANS VALUES (103, 3, 75000, 7.5, SYSDATE-300, SYSDATE+15);

COMMIT;

**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.
  + **Solution :**

BEGIN

FOR rec IN (

SELECT c.CustomerID, c.DOB, l.LoanID, l.InterestRate

FROM CUSTOMERS c

JOIN LOANS l ON c.CUSTOMERID = l.CUSTOMERID

FOR UPDATE

) LOOP

IF MONTHS\_BETWEEN(SYSDATE, rec.DOB)/12>60 THEN

UPDATE LOANS

SET InterestRate = rec.InterestRate-1

WHERE LoanID = rec.LoanID;

END IF;

END LOOP;

COMMIT;

END;

/

A screenshot of a computer program

AI-generated content may be incorrect.

SELECT c.Name, l.InterestRate FROM CUSTOMERS c JOIN LOANS l ON c.CUSTOMERID = l.CUSTOMERID;

A screenshot of a computer

AI-generated content may be incorrect.

**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.
  + **Solution:**

BEGIN

FOR rec IN (SELECT CustomerID, Balance FROM CUSTOMERS FOR UPDATE) LOOP

IF rec.Balance > 10000 THEN

UPDATE CUSTOMERS

SET LastModified = SYSDATE

WHERE CustomerID = rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('Customer '||rec.CustomerID||' marked as VIP.');

END IF;

END LOOP;

COMMIT;

END;

/

A white background with black text

AI-generated content may be incorrect.

SELECT \* FROM Customers;

A screenshot of a data

AI-generated content may be incorrect.

**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:**

BEGIN

FOR rec IN(

SELECT c.Name, l.LoanID, l.EndDate

FROM LOANS l

JOIN CUSTOMERS c on c.CUSTOMERID = l.CUSTOMERID

WHERE l.ENDDATE BETWEEN SYSDATE AND SYSDATE+30

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Remainder: Dear '|| rec.Name || ', your loan ID '|| rec.LoanID || 'is due on '|| TO\_CHAR(rec.EndDaTE,'DD-MON-YYYY')|| '.');

END LOOP;

END;

/

A computer screen shot of a computer error

AI-generated content may be incorrect.