**Exercise 1: Control Structures**

CREATE TABLE CUSTOMERS (

CUSTOMER\_ID NUMBER PRIMARY KEY,

NAME VARCHAR2(100),

AGE NUMBER,

BALANCE NUMBER(10, 2),

ISVIP VARCHAR2(5)

);

CREATE TABLE LOANS (

LOAN\_ID NUMBER PRIMARY KEY,

CUSTOMER\_ID NUMBER REFERENCES CUSTOMERS(CUSTOMER\_ID),

INTEREST\_RATE NUMBER(5, 2),

DUE\_DATE DATE

);

INSERT INTO CUSTOMERS VALUES (1, 'Alice Johnson', 65, 15000, 'FALSE');

INSERT INTO CUSTOMERS VALUES (2, 'Bob Smith', 45, 8000, 'FALSE');

INSERT INTO CUSTOMERS VALUES (3, 'Charlie Lee', 70, 12000, 'FALSE');

INSERT INTO CUSTOMERS VALUES (4, 'Diana Moore', 30, 9500, 'FALSE');

INSERT INTO LOANS VALUES (101, 1, 5.5, SYSDATE + 10);

INSERT INTO LOANS VALUES (102, 2, 6.0, SYSDATE + 40);

INSERT INTO LOANS VALUES (103, 3, 4.5, SYSDATE + 20);

INSERT INTO LOANS VALUES (104, 4, 7.0, SYSDATE + 5);

COMMIT;

**Scenario-1:**

BEGIN

FOR cust\_rec IN (SELECT CUSTOMER\_ID FROM CUSTOMERS WHERE AGE > 60) LOOP

UPDATE LOANS

SET INTEREST\_RATE = INTEREST\_RATE - 1

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

DBMS\_OUTPUT.PUT\_LINE('Updated loans for customer ID: ' || cust\_rec.CUSTOMER\_ID);

END LOOP;

COMMIT;

END;

/

**Scenario-2:**

BEGIN

FOR cust\_rec IN (SELECT CUSTOMER\_ID FROM CUSTOMERS WHERE BALANCE > 10000) LOOP

UPDATE CUSTOMERS

SET ISVIP = 'TRUE'

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

DBMS\_OUTPUT.PUT\_LINE('Promoted customer ID ' || cust\_rec.CUSTOMER\_ID || ' to VIP');

END LOOP;

COMMIT;

END;

/

**Scenario-3:**

SET SERVEROUTPUT ON;

BEGIN

FOR loan\_rec IN (

SELECT l.LOAN\_ID, l.CUSTOMER\_ID, c.NAME, l.DUE\_DATE

FROM LOANS l

JOIN CUSTOMERS c ON l.CUSTOMER\_ID = c.CUSTOMER\_ID

WHERE l.DUE\_DATE <= SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || loan\_rec.LOAN\_ID ||

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

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

END LOOP;

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 3: Stored Procedures**

CREATE TABLE SavingsAccount (

AccountID NUMBER PRIMARY KEY,

CustomerName VARCHAR2(100),

Balance NUMBER

);

INSERT INTO SavingsAccount VALUES (1, 'Ajay', 1000);

INSERT INTO SavingsAccount VALUES (2, 'Bella', 1500);

INSERT INTO SavingsAccount VALUES (3, 'Dev', 2000);

CREATE TABLE Employee (

EmpID NUMBER PRIMARY KEY,

EmpName VARCHAR2(100),

Department VARCHAR2(50),

Salary NUMBER

);

INSERT INTO Employee VALUES (101, 'Ravi', 'Sales', 5000);

INSERT INTO Employee VALUES (102, 'Hari', 'Sales', 5500);

INSERT INTO Employee VALUES (103, 'Siri', 'IT', 6000);

CREATE TABLE CustomerAccount (

AccountID NUMBER PRIMARY KEY,

CustomerName VARCHAR2(100),

Balance NUMBER

);

INSERT INTO CustomerAccount VALUES (10, 'Sita', 3000);

INSERT INTO CustomerAccount VALUES (11, 'Kishore', 2500);

**Scenario-1:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

UPDATE SavingsAccount

SET Balance = Balance + (Balance \* 0.01);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Monthly interest processed for all savings accounts.');

END;

/

BEGIN

ProcessMonthlyInterest;

END;

/

**Scenario-2:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_Department IN VARCHAR2,

p\_BonusPercent IN NUMBER

) AS

BEGIN

UPDATE Employee

SET Salary = Salary + (Salary \* (p\_BonusPercent / 100))

WHERE Department = p\_Department;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Employee bonuses updated in department: ' || p\_Department);

END;

/

BEGIN

UpdateEmployeeBonus('Sales', 10);

END;

/

**Scenario-3:**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_FromAccountID IN NUMBER,

p\_ToAccountID IN NUMBER,

p\_Amount IN NUMBER

) AS

v\_FromBalance NUMBER;

BEGIN

SELECT Balance INTO v\_FromBalance

FROM CustomerAccount

WHERE AccountID = p\_FromAccountID

FOR UPDATE;

IF v\_FromBalance < p\_Amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account.');

END IF;

UPDATE CustomerAccount

SET Balance = Balance - p\_Amount

WHERE AccountID = p\_FromAccountID;

UPDATE CustomerAccount

SET Balance = Balance + p\_Amount

WHERE AccountID = p\_ToAccountID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || p\_Amount || ' from Account ' || p\_FromAccountID || ' to Account ' || p\_ToAccountID);

END;

/

BEGIN

TransferFunds(10, 11, 500);

END;

/

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.