**Exercise 1: Control Structures**

**DB Structure and Dummy Data :**

SET SERVEROUTPUT ON SIZE 1000000;

CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

customer\_name VARCHAR2(100) NOT NULL,

date\_of\_birth DATE NOT NULL,

balance NUMBER(15,2) DEFAULT 0,

is\_vip CHAR(1) DEFAULT 'N' CHECK (is\_vip IN ('Y', 'N')),

email VARCHAR2(100),

phone VARCHAR2(15),

created\_date DATE DEFAULT SYSDATE

);

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER NOT NULL,

loan\_amount NUMBER(15,2) NOT NULL,

interest\_rate NUMBER(5,2) NOT NULL,

loan\_start\_date DATE NOT NULL,

loan\_due\_date DATE NOT NULL,

loan\_status VARCHAR2(20) DEFAULT 'ACTIVE',

CONSTRAINT fk\_customer\_loan FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

CREATE SEQUENCE customer\_seq START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE loan\_seq START WITH 1 INCREMENT BY 1;

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'John Smith', DATE '1950-05-15', 15000.00, 'john.smith@email.com', '555-0101');

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Maria Garcia', DATE '1960-08-22', 8500.00, 'maria.garcia@email.com', '555-0102');

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Robert Johnson', DATE '1955-12-10', 25000.00, 'robert.johnson@email.com', '555-0103');

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Emily Davis', DATE '1985-03-18', 12000.00, 'emily.davis@email.com', '555-0104');

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Michael Wilson', DATE '1945-07-30', 5000.00, 'michael.wilson@email.com', '555-0105');

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Sarah Brown', DATE '1992-11-05', 7500.00, 'sarah.brown@email.com', '555-0106');

INSERT INTO loans (loan\_id, customer\_id, loan\_amount, interest\_rate, loan\_start\_date, loan\_due\_date) VALUES

(loan\_seq.NEXTVAL, 1, 50000.00, 5.5, DATE '2024-01-15', SYSDATE + 20);

INSERT INTO loans (loan\_id, customer\_id, loan\_amount, interest\_rate, loan\_start\_date, loan\_due\_date) VALUES

(loan\_seq.NEXTVAL, 2, 30000.00, 6.0, DATE '2024-03-10', SYSDATE + 15);

INSERT INTO loans (loan\_id, customer\_id, loan\_amount, interest\_rate, loan\_start\_date, loan\_due\_date) VALUES

(loan\_seq.NEXTVAL, 3, 75000.00, 4.8, DATE '2024-02-20', SYSDATE + 45);

INSERT INTO loans (loan\_id, customer\_id, loan\_amount, interest\_rate, loan\_start\_date, loan\_due\_date) VALUES

(loan\_seq.NEXTVAL, 4, 40000.00, 5.2, DATE '2024-05-05', SYSDATE + 25);

INSERT INTO loans (loan\_id, customer\_id, loan\_amount, interest\_rate, loan\_start\_date, loan\_due\_date) VALUES

(loan\_seq.NEXTVAL, 5, 20000.00, 6.5, DATE '2024-04-12', SYSDATE + 10);

COMMIT;

**Scenario 1:**

DECLARE

v\_total\_processed NUMBER := 0;

v\_total\_discounted NUMBER := 0;

v\_current\_age NUMBER;

v\_rows\_affected NUMBER;

CURSOR senior\_cursor IS

SELECT c.customer\_id, c.customer\_name, c.date\_of\_birth,

TRUNC(MONTHS\_BETWEEN(SYSDATE, c.date\_of\_birth) / 12) AS age

FROM customers c

WHERE EXISTS (

SELECT 1 FROM loans l

WHERE l.customer\_id = c.customer\_id

)

ORDER BY c.customer\_name;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('--- APPLYING INTEREST DISCOUNT FOR SENIOR CUSTOMERS ---');

FOR cust IN senior\_cursor LOOP

v\_total\_processed := v\_total\_processed + 1;

IF cust.age >= 61 THEN

DBMS\_OUTPUT.PUT\_LINE('Eligible: ' || cust.customer\_name || ' | Age: ' || cust.age);

UPDATE loans

SET interest\_rate = CASE

WHEN interest\_rate > 1 THEN interest\_rate - 1

ELSE interest\_rate

END

WHERE customer\_id = cust.customer\_id AND loan\_status = 'ACTIVE';

v\_rows\_affected := SQL%ROWCOUNT;

v\_total\_discounted := v\_total\_discounted + v\_rows\_affected;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Not Eligible: ' || cust.customer\_name || ' | Age: ' || cust.age);

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Processed Customers: ' || v\_total\_processed);

DBMS\_OUTPUT.PUT\_LINE('Loans Discounted: ' || v\_total\_discounted);

IF v\_total\_discounted > 0 THEN

COMMIT;

END IF;

EXCEPTION

WHEN OTHERS THEN

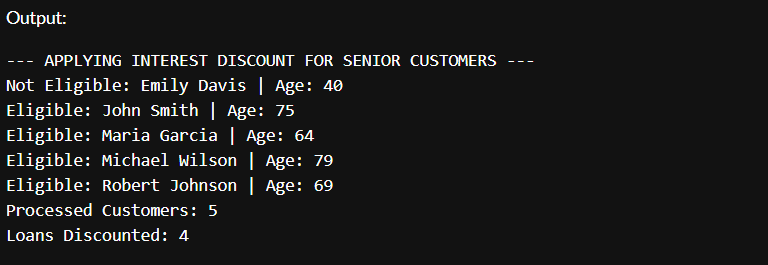
ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

**Output :**



**Scenario 2:**

DECLARE

v\_customers\_checked NUMBER := 0;

v\_promoted\_count NUMBER := 0;

v\_vip\_existing NUMBER := 0;

v\_ineligible\_customers NUMBER := 0;

CURSOR vip\_cursor IS

SELECT customer\_id, customer\_name, balance, is\_vip

FROM customers

ORDER BY balance DESC;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('--- EVALUATING CUSTOMERS FOR VIP PROMOTION ---');

FOR cust IN vip\_cursor LOOP

v\_customers\_checked := v\_customers\_checked + 1;

IF cust.balance > 10000 THEN

IF cust.is\_vip = 'Y' THEN

v\_vip\_existing := v\_vip\_existing + 1;

ELSE

UPDATE customers SET is\_vip = 'Y' WHERE customer\_id = cust.customer\_id;

v\_promoted\_count := v\_promoted\_count + 1;

DBMS\_OUTPUT.PUT\_LINE('Promoted to VIP: ' || cust.customer\_name);

END IF;

ELSE

v\_ineligible\_customers := v\_ineligible\_customers + 1;

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Total Evaluated: ' || v\_customers\_checked);

DBMS\_OUTPUT.PUT\_LINE('New VIPs: ' || v\_promoted\_count);

DBMS\_OUTPUT.PUT\_LINE('Already VIP: ' || v\_vip\_existing);

DBMS\_OUTPUT.PUT\_LINE('Not Eligible: ' || v\_ineligible\_customers);

IF v\_promoted\_count > 0 THEN

COMMIT;

END IF;

EXCEPTION

WHEN OTHERS THEN

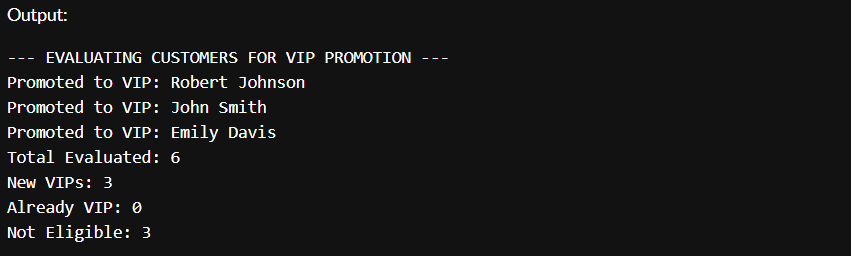
ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

**Output :**



**Scenario 3:**

BEGIN

DBMS\_OUTPUT.PUT\_LINE('--- REMINDER: Upcoming Loan Dues in Next 30 Days ---');

FOR rec IN (

SELECT l.loan\_id, l.customer\_id, l.loan\_due\_date, l.loan\_amount,

c.customer\_name, c.email, c.phone

FROM loans l

JOIN customers c ON l.customer\_id = c.customer\_id

WHERE l.loan\_status = 'ACTIVE'

AND l.loan\_due\_date BETWEEN SYSDATE AND (SYSDATE + 30)

ORDER BY l.loan\_due\_date

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder for ' || rec.customer\_name);

DBMS\_OUTPUT.PUT\_LINE('-> Loan ID: ' || rec.loan\_id ||

', Due Date: ' || TO\_CHAR(rec.loan\_due\_date, 'DD-Mon-YYYY') ||

', Amount: ₹' || TO\_CHAR(rec.loan\_amount, '999,999.99'));

DBMS\_OUTPUT.PUT\_LINE('-> Contact: ' || rec.email || ' | ' || rec.phone);

DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------');

END LOOP;

EXCEPTION

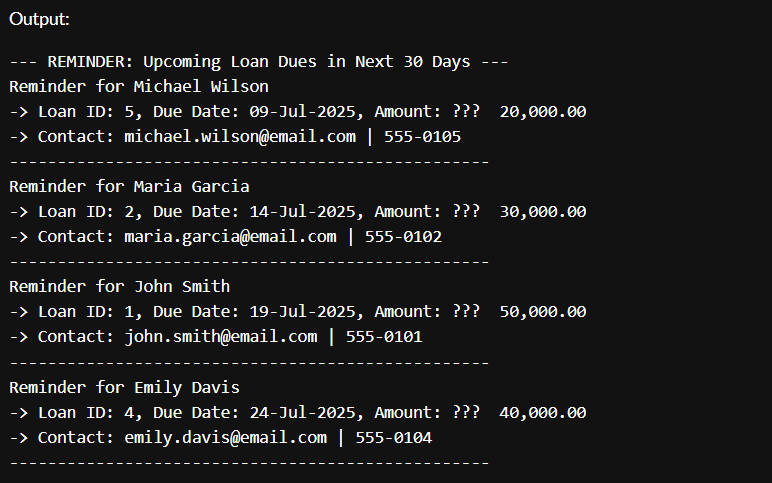
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error occurred while fetching reminders: ' || SQLERRM);

END;

/

**Output :**



**Exercise 3 : Stored Procedure**

SET SERVEROUTPUT ON SIZE 1000000;

CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

customer\_name VARCHAR2(100) NOT NULL,

date\_of\_birth DATE NOT NULL,

balance NUMBER(15,2) DEFAULT 0,

is\_vip CHAR(1) DEFAULT 'N' CHECK (is\_vip IN ('Y', 'N')),

email VARCHAR2(100),

phone VARCHAR2(15),

created\_date DATE DEFAULT SYSDATE

);

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER NOT NULL,

loan\_amount NUMBER(15,2) NOT NULL,

interest\_rate NUMBER(5,2) NOT NULL,

loan\_start\_date DATE NOT NULL,

loan\_due\_date DATE NOT NULL,

loan\_status VARCHAR2(20) DEFAULT 'ACTIVE',

CONSTRAINT fk\_customer\_loan FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

CREATE SEQUENCE customer\_seq START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE loan\_seq START WITH 1 INCREMENT BY 1;

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Vikram Shah', DATE '1951-04-20', 16400.00, 'vikram.shah@email.com', '999-1211');

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Rina Dutta', DATE '1961-07-14', 8100.00, 'rina.dutta@email.com', '999-1212');

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Karthik Rao', DATE '1954-01-25', 23400.00, 'karthik.rao@email.com', '999-1213');

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Neha Pillai', DATE '1988-02-28', 13800.00, 'neha.pillai@email.com', '999-1214');

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Manoj Kapoor', DATE '1946-09-03', 5100.00, 'manoj.kapoor@email.com', '999-1215');

INSERT INTO customers (customer\_id, customer\_name, date\_of\_birth, balance, email, phone) VALUES

(customer\_seq.NEXTVAL, 'Swathi Balan', DATE '1991-12-12', 7600.00, 'swathi.balan@email.com', '999-1216');

INSERT INTO loans (loan\_id, customer\_id, loan\_amount, interest\_rate, loan\_start\_date, loan\_due\_date) VALUES

(loan\_seq.NEXTVAL, 1, 41000.00, 5.4, DATE '2024-01-20', SYSDATE + 17);

INSERT INTO loans (loan\_id, customer\_id, loan\_amount, interest\_rate, loan\_start\_date, loan\_due\_date) VALUES

(loan\_seq.NEXTVAL, 2, 28000.00, 5.9, DATE '2024-03-15', SYSDATE + 14);

INSERT INTO loans (loan\_id, customer\_id, loan\_amount, interest\_rate, loan\_start\_date, loan\_due\_date) VALUES

(loan\_seq.NEXTVAL, 3, 72000.00, 4.9, DATE '2024-02-18', SYSDATE + 41);

INSERT INTO loans (loan\_id, customer\_id, loan\_amount, interest\_rate, loan\_start\_date, loan\_due\_date) VALUES

(loan\_seq.NEXTVAL, 4, 37500.00, 5.3, DATE '2024-05-02', SYSDATE + 21);

INSERT INTO loans (loan\_id, customer\_id, loan\_amount, interest\_rate, loan\_start\_date, loan\_due\_date) VALUES

(loan\_seq.NEXTVAL, 5, 19500.00, 6.2, DATE '2024-04-10', SYSDATE + 11);

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

employee\_name VARCHAR2(100),

department\_id NUMBER,

salary NUMBER(10,2)

);

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

balance NUMBER(15,2)

);

INSERT INTO employees VALUES (21, 'Aakash Mehta', 15, 48000);

INSERT INTO employees VALUES (22, 'Bhavna Reddy', 15, 45500);

INSERT INTO employees VALUES (23, 'Harsh Soni', 25, 59500);

INSERT INTO accounts VALUES (3001, 1, 3200);

INSERT INTO accounts VALUES (3002, 2, 1500);

COMMIT;

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

UPDATE customers

SET balance = balance + (balance \* 0.01)

WHERE balance > 0;

DBMS\_OUTPUT.PUT\_LINE('Monthly interest of 1% applied to all savings accounts.');

COMMIT;

END;

/

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_dept\_id IN NUMBER,

p\_bonus\_percent IN NUMBER

) AS

BEGIN

UPDATE employees

SET salary = salary + (salary \* p\_bonus\_percent / 100)

WHERE department\_id = p\_dept\_id;

DBMS\_OUTPUT.PUT\_LINE('Bonus of ' || p\_bonus\_percent || '% applied to department ' || p\_dept\_id);

COMMIT;

END;

/

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

SELECT balance INTO v\_balance FROM accounts WHERE account\_id = p\_from\_account FOR UPDATE;

IF v\_balance < p\_amount THEN

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

END IF;

UPDATE accounts SET balance = balance - p\_amount WHERE account\_id = p\_from\_account;

UPDATE accounts SET balance = balance + p\_amount WHERE account\_id = p\_to\_account;

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || p\_amount || ' from account ' || p\_from\_account || ' to account ' || p\_to\_account);

COMMIT;

END;

/