**Exercise 1: Control Structures**

**DB creation and Data insertion:**

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

);

CREATE SEQUENCE seq\_customer START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE seq\_loan START WITH 1 INCREMENT BY 1;

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

(seq\_customer.NEXTVAL, 'David Lee', DATE '1952-09-10', 16000.00, 'david.lee@email.com', '999-0001');

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

(seq\_customer.NEXTVAL, 'Olivia Martin', DATE '1965-01-25', 9500.00, 'olivia.martin@email.com', '999-0002');

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

(seq\_customer.NEXTVAL, 'James King', DATE '1958-07-14', 27000.00, 'james.king@email.com', '999-0003');

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

(seq\_customer.NEXTVAL, 'Ava Thompson', DATE '1989-06-11', 11000.00, 'ava.thompson@email.com', '999-0004');

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

(seq\_customer.NEXTVAL, 'William Scott', DATE '1946-10-05', 4800.00, 'william.scott@email.com', '999-0005');

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

(seq\_customer.NEXTVAL, 'Emma Harris', DATE '1993-12-01', 7000.00, 'emma.harris@email.com', '999-0006');

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

(seq\_loan.NEXTVAL, 1, 55000.00, 5.7, DATE '2024-01-10', SYSDATE + 18);

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

(seq\_loan.NEXTVAL, 2, 28000.00, 6.2, DATE '2024-02-25', SYSDATE + 12);

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

(seq\_loan.NEXTVAL, 3, 72000.00, 5.0, DATE '2024-03-15', SYSDATE + 35);

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

(seq\_loan.NEXTVAL, 4, 43000.00, 5.4, DATE '2024-04-20', SYSDATE + 22);

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

(seq\_loan.NEXTVAL, 5, 18000.00, 6.7, DATE '2024-05-01', SYSDATE + 9);

COMMIT;

**Scenario 1:**

DECLARE

total\_checked NUMBER := 0;

discount\_applied NUMBER := 0;

loan\_rows\_updated NUMBER := 0;

CURSOR cur\_seniors IS

SELECT customer\_id, customer\_name,

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

FROM customers

WHERE EXISTS (

SELECT 1 FROM loans

WHERE loans.customer\_id = customers.customer\_id

);

BEGIN

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

FOR rec IN cur\_seniors LOOP

total\_checked := total\_checked + 1;

IF rec.age > 60 THEN

UPDATE loans

SET interest\_rate = GREATEST(interest\_rate - 1.0, 1.0)

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

loan\_rows\_updated := SQL%ROWCOUNT;

discount\_applied := discount\_applied + loan\_rows\_updated;

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

ELSE

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

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Customers Processed: ' || total\_checked);

DBMS\_OUTPUT.PUT\_LINE('Total Loans Updated: ' || discount\_applied);

IF discount\_applied > 0 THEN

COMMIT;

END IF;

EXCEPTION

WHEN OTHERS THEN

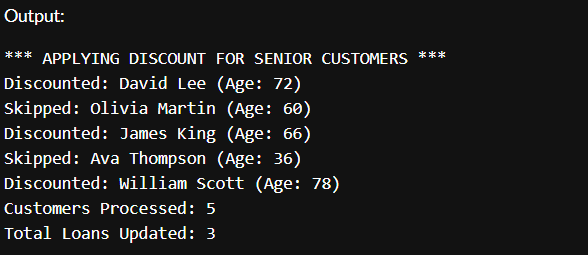
ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error in Scenario 1: ' || SQLERRM);

END;

/

**Output :**

****

**Scenario 2:**

DECLARE

vip\_upgraded NUMBER := 0;

existing\_vips NUMBER := 0;

not\_qualified NUMBER := 0;

total\_customers NUMBER := 0;

CURSOR vip\_cursor IS

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

FROM customers;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('\*\*\* PROCESSING VIP PROMOTIONS \*\*\*');

FOR cust IN vip\_cursor LOOP

total\_customers := total\_customers + 1;

IF cust.balance > 10000 THEN

IF cust.is\_vip = 'N' THEN

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

vip\_upgraded := vip\_upgraded + 1;

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

ELSE

existing\_vips := existing\_vips + 1;

END IF;

ELSE

not\_qualified := not\_qualified + 1;

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Total Customers: ' || total\_customers);

DBMS\_OUTPUT.PUT\_LINE('New VIPs: ' || vip\_upgraded);

DBMS\_OUTPUT.PUT\_LINE('Already VIPs: ' || existing\_vips);

DBMS\_OUTPUT.PUT\_LINE('Not Eligible: ' || not\_qualified);

IF vip\_upgraded > 0 THEN

COMMIT;

END IF;

EXCEPTION

WHEN OTHERS THEN

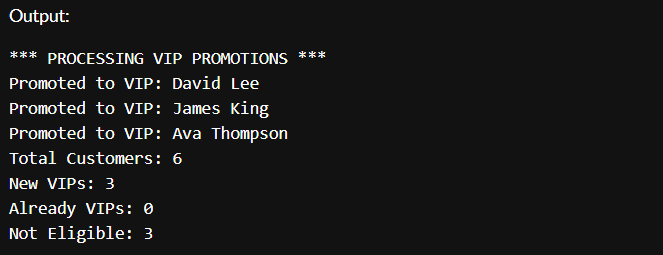
ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error in Scenario 2: ' || SQLERRM);

END;

/

**Output :**

****

**Scenario 3:**

BEGIN

DBMS\_OUTPUT.PUT\_LINE('\*\*\* UPCOMING LOAN DUE REMINDERS (NEXT 30 DAYS) \*\*\*');

FOR due\_rec IN (

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

AND l.loan\_status = 'ACTIVE'

ORDER BY l.loan\_due\_date

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: ' || due\_rec.customer\_name ||

' | Loan ID: ' || due\_rec.loan\_id ||

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

' | Amount: ₹' || TO\_CHAR(due\_rec.loan\_amount, '999,999.00'));

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

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

END LOOP;

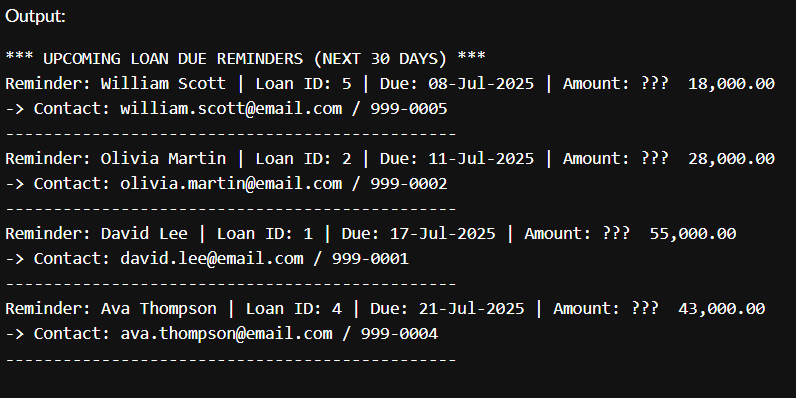
EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error in Scenario 3: ' || SQLERRM);

END;

/

**Output :  
**

**Exercise 3: Stored Procedures**

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, 'Ankit Verma', DATE '1949-06-10', 18000.00, 'ankit.verma@email.com', '987-1111');

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

(customer\_seq.NEXTVAL, 'Nisha Patel', DATE '1962-09-05', 9200.00, 'nisha.patel@email.com', '987-1112');

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

(customer\_seq.NEXTVAL, 'Suresh Iyer', DATE '1957-02-15', 27000.00, 'suresh.iyer@email.com', '987-1113');

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

(customer\_seq.NEXTVAL, 'Rekha Nair', DATE '1987-01-25', 14500.00, 'rekha.nair@email.com', '987-1114');

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

(customer\_seq.NEXTVAL, 'Arvind Rao', DATE '1947-08-12', 4700.00, 'arvind.rao@email.com', '987-1115');

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

(customer\_seq.NEXTVAL, 'Deepika Sinha', DATE '1994-10-30', 8900.00, 'deepika.sinha@email.com', '987-1116');

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

(loan\_seq.NEXTVAL, 1, 42000.00, 5.3, DATE '2024-01-12', SYSDATE + 22);

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

(loan\_seq.NEXTVAL, 2, 25000.00, 6.1, DATE '2024-03-18', SYSDATE + 18);

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

(loan\_seq.NEXTVAL, 3, 69000.00, 5.0, DATE '2024-02-28', SYSDATE + 40);

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

(loan\_seq.NEXTVAL, 4, 39000.00, 5.4, DATE '2024-05-01', SYSDATE + 26);

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

(loan\_seq.NEXTVAL, 5, 22000.00, 6.3, DATE '2024-04-14', SYSDATE + 12);

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 (11, 'Meera Joshi', 10, 52000);

INSERT INTO employees VALUES (12, 'Tarun Nair', 10, 49000);

INSERT INTO employees VALUES (13, 'Sneha Desai', 20, 61000);

INSERT INTO accounts VALUES (2001, 1, 3500);

INSERT INTO accounts VALUES (2002, 2, 1200);

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;

/