**1.Control Structures:**

CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

age NUMBER,

balance NUMBER,

isvip VARCHAR2(5)

);

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

interest\_rate NUMBER,

due\_date DATE,

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

);

INSERT INTO customers VALUES (1, 65, 12000, 'FALSE');

INSERT INTO customers VALUES (2, 45, 8000, 'FALSE');

INSERT INTO customers VALUES (3, 70, 15000, 'FALSE');

INSERT INTO customers VALUES (4, 30, 9500, 'FALSE');

INSERT INTO loans VALUES (101, 1, 7.5, SYSDATE + 10);

INSERT INTO loans VALUES (102, 2, 8.0, SYSDATE + 40);

INSERT INTO loans VALUES (103, 3, 9.0, SYSDATE + 20);

INSERT INTO loans VALUES (104, 4, 7.2, SYSDATE + 5);

COMMIT;

BEGIN

FOR cust\_rec IN (SELECT customer\_id FROM customers WHERE age > 60) LOOP

UPDATE loans

SET interest\_rate = interest\_rate - (interest\_rate \* 0.01)

WHERE customer\_id = cust\_rec.customer\_id;

END LOOP;

COMMIT;

END;

/

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;

END LOOP;

COMMIT;

END;

/

BEGIN

FOR loan\_rec IN (

SELECT loan\_id, customer\_id, due\_date

FROM loans

WHERE due\_date BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE(

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

' for Customer ID ' || loan\_rec.customer\_id ||

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

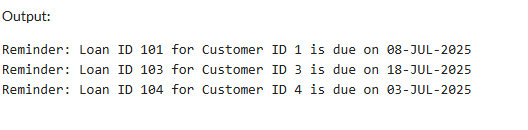
);

END LOOP;

END;

/

OUTPUT;



**3.Stored Procedures:**

SET SERVEROUTPUT ON;

CREATE TABLE savings\_accounts (

account\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

balance NUMBER

);

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

department\_id NUMBER,

salary NUMBER

);

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

balance NUMBER

);

INSERT INTO savings\_accounts VALUES (101, 1, 5000);

INSERT INTO savings\_accounts VALUES (102, 2, 7000);

INSERT INTO savings\_accounts VALUES (103, 3, 10000);

INSERT INTO employees VALUES (201, 10, 30000);

INSERT INTO employees VALUES (202, 10, 35000);

INSERT INTO employees VALUES (203, 20, 40000);

INSERT INTO accounts VALUES (301, 1, 8000);

INSERT INTO accounts VALUES (302, 1, 3000);

INSERT INTO accounts VALUES (303, 2, 10000);

COMMIT;

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (SELECT account\_id, balance FROM savings\_accounts) LOOP

UPDATE savings\_accounts

SET balance = balance + (balance \* 0.01)

WHERE account\_id = acc.account\_id;

DBMS\_OUTPUT.PUT\_LINE('Interest applied to Account ID ' || acc.account\_id);

END LOOP;

COMMIT;

END;

/

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_department\_id IN NUMBER,

p\_bonus\_percent IN NUMBER

) IS

v\_count NUMBER := 0;

BEGIN

UPDATE employees

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

WHERE department\_id = p\_department\_id;

v\_count := SQL%ROWCOUNT;

DBMS\_OUTPUT.PUT\_LINE(v\_count || ' employee(s) in department ' || p\_department\_id || ' received a ' || p\_bonus\_percent || '% bonus.');

COMMIT;

END;

/

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

) IS

v\_balance NUMBER;

BEGIN

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

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;

/

BEGIN

ProcessMonthlyInterest;

END;

/

BEGIN

UpdateEmployeeBonus(10, 10);

END;

/

BEGIN

TransferFunds(301, 302, 2000);

END;

/

OUTPUT:

