**Exercise 6: Cursors**

**Step 1: Generate Monthly Statements**

DECLARE

CURSOR cur\_transactions IS

SELECT t.customer\_id, c.customer\_name, t.transaction\_date, t.transaction\_type, t.amount

FROM transactions t

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

WHERE EXTRACT(MONTH FROM t.transaction\_date) = EXTRACT(MONTH FROM SYSDATE)

AND EXTRACT(YEAR FROM t.transaction\_date) = EXTRACT(YEAR FROM SYSDATE);

v\_customer\_id transactions.customer\_id%TYPE;

v\_customer\_name customers.customer\_name%TYPE;

v\_transaction\_date transactions.transaction\_date%TYPE;

v\_transaction\_type transactions.transaction\_type%TYPE;

v\_amount transactions.amount%TYPE;

BEGIN

OPEN cur\_transactions;

LOOP

FETCH cur\_transactions INTO v\_customer\_id, v\_customer\_name, v\_transaction\_date, v\_transaction\_type, v\_amount;

EXIT WHEN cur\_transactions%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || v\_customer\_id);

DBMS\_OUTPUT.PUT\_LINE('Customer Name: ' || v\_customer\_name);

DBMS\_OUTPUT.PUT\_LINE('Transaction Date: ' || v\_transaction\_date);

DBMS\_OUTPUT.PUT\_LINE('Transaction Type: ' || v\_transaction\_type);

DBMS\_OUTPUT.PUT\_LINE('Amount: ' || v\_amount);

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

END LOOP;

CLOSE cur\_transactions;

END;

/

**Step 2: Apply Annual Fee**

DECLARE

CURSOR cur\_accounts IS

SELECT account\_id, balance

FROM accounts;

v\_account\_id accounts.account\_id%TYPE;

v\_balance accounts.balance%TYPE;

v\_annual\_fee NUMBER := 50; -- Example annual fee amount

BEGIN

OPEN cur\_accounts;

LOOP

FETCH cur\_accounts INTO v\_account\_id, v\_balance;

EXIT WHEN cur\_accounts%NOTFOUND;

-- Deduct the annual fee from the balance

UPDATE accounts

SET balance = balance - v\_annual\_fee

WHERE account\_id = v\_account\_id;

DBMS\_OUTPUT.PUT\_LINE('Applied annual fee to account ID: ' || v\_account\_id);

END LOOP;

CLOSE cur\_accounts;

COMMIT;

END;

/

**Step 3: Update Loan Interest Rates**

DECLARE

CURSOR cur\_loans IS

SELECT loan\_id, loan\_interest\_rate

FROM loans;

v\_loan\_id loans.loan\_id%TYPE;

v\_loan\_interest\_rate loans.loan\_interest\_rate%TYPE;

v\_new\_interest\_rate NUMBER;

BEGIN

OPEN cur\_loans;

LOOP

FETCH cur\_loans INTO v\_loan\_id, v\_loan\_interest\_rate;

EXIT WHEN cur\_loans%NOTFOUND;

-- Calculate the new interest rate based on the policy (example: increase by 0.5%)

v\_new\_interest\_rate := v\_loan\_interest\_rate + 0.5;

-- Update the loan interest rate

UPDATE loans

SET loan\_interest\_rate = v\_new\_interest\_rate

WHERE loan\_id = v\_loan\_id;

DBMS\_OUTPUT.PUT\_LINE('Updated interest rate for loan ID: ' || v\_loan\_id);

END LOOP;

CLOSE cur\_loans;

COMMIT;

END;

/

**Example Data Insertion Scripts**

-- Creating tables

CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

customer\_name VARCHAR2(100)

);

CREATE TABLE transactions (

transaction\_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

customer\_id NUMBER,

transaction\_date DATE,

transaction\_type VARCHAR2(10),

amount NUMBER,

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

);

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

balance NUMBER

);

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

loan\_interest\_rate NUMBER

);

-- Inserting example data

INSERT INTO customers (customer\_id, customer\_name) VALUES (1, 'John Doe');

INSERT INTO customers (customer\_id, customer\_name) VALUES (2, 'Jane Smith');

INSERT INTO transactions (customer\_id, transaction\_date, transaction\_type, amount) VALUES (1, SYSDATE, 'Deposit', 500);

INSERT INTO transactions (customer\_id, transaction\_date, transaction\_type, amount) VALUES (2, SYSDATE, 'Withdrawal', 200);

INSERT INTO accounts (account\_id, balance) VALUES (1, 1000);

INSERT INTO accounts (account\_id, balance) VALUES (2, 2000);

INSERT INTO loans (loan\_id, loan\_interest\_rate) VALUES (1, 3.5);

INSERT INTO loans (loan\_id, loan\_interest\_rate) VALUES (2, 4.0);

COMMIT;