**Exercise 6: Cursors**

**Scenario 1:** Generate monthly statements for all customers.

**CODE:** DECLARE

-- Cursor to retrieve transactions for the current month

CURSOR cur\_monthly\_transactions IS

SELECT t.account\_id, a.customer\_name, t.transaction\_date, t.amount

FROM transactions t

JOIN accounts a ON t.account\_id = a.account\_id

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

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

ORDER BY a.customer\_name, t.transaction\_date;

-- Record to hold cursor data

rec\_transaction cur\_monthly\_transactions%ROWTYPE;

BEGIN

-- Open the cursor and fetch each record

OPEN cur\_monthly\_transactions;

LOOP

FETCH cur\_monthly\_transactions INTO rec\_transaction;

EXIT WHEN cur\_monthly\_transactions%NOTFOUND;

-- Print statement for each transaction

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || rec\_transaction.customer\_name);

DBMS\_OUTPUT.PUT\_LINE('Transaction Date: ' || TO\_CHAR(rec\_transaction.transaction\_date, 'DD-MON-YYYY'));

DBMS\_OUTPUT.PUT\_LINE('Amount: ' || TO\_CHAR(rec\_transaction.amount, 'FM999,999.00'));

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

END LOOP;

-- Close the cursor

CLOSE cur\_monthly\_transactions;

EXCEPTION

WHEN OTHERS THEN

-- Handle any unexpected errors

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

END;

**Scenario 2:** Apply annual fee to all accounts.

**CODE: DECLARE**

DECLARE

annual\_fee\_amount CONSTANT NUMBER := 50.00;

CURSOR cur\_accounts IS

SELECT account\_id, balance

FROM accounts

FOR UPDATE;

rec\_account cur\_accounts%ROWTYPE;

BEGIN

OPEN cur\_accounts;

LOOP

FETCH cur\_accounts INTO rec\_account;

EXIT WHEN cur\_accounts%NOTFOUND;

IF rec\_account.balance >= annual\_fee\_amount THEN

UPDATE accounts

SET balance = rec\_account.balance - annual\_fee\_amount

WHERE account\_id = rec\_account.account\_id;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Account ' || rec\_account.account\_id || ' has insufficient balance for fee deduction.');

END IF;

END LOOP;

COMMIT;

CLOSE cur\_accounts;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

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

END;

**Scenario 3:** Update the interest rate for all loans based on a new policy.

**CODE:**

DECLARE

new\_interest\_rate CONSTANT NUMBER := 5.00;

CURSOR cur\_loans IS

SELECT loan\_id, interest\_rate

FROM loans

FOR UPDATE;

rec\_loan cur\_loans%ROWTYPE;

BEGIN

OPEN cur\_loans;

LOOP

FETCH cur\_loans INTO rec\_loan;

EXIT WHEN cur\_loans%NOTFOUND;

UPDATE loans

SET interest\_rate = new\_interest\_rate

WHERE loan\_id = rec\_loan.loan\_id;

END LOOP;

COMMIT;

CLOSE cur\_loans;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error updating loan interest rates: ' || SQLERRM);

END;