**EXERCISE 6:**

Write a PL/SQL block using an explicit cursor GenerateMonthlyStatements that retrieves all transactions for the current month and prints a statement for each customer.

DECLARE

CURSOR transactions\_cursor IS

SELECT customer\_id, transaction\_id, transaction\_type, amount, transaction\_date

FROM transactions

WHERE TRUNC(transaction\_date, 'MM') = TRUNC(SYSDATE, 'MM');

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

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

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

v\_amount transactions.amount%TYPE;

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

BEGIN

OPEN transactions\_cursor;

LOOP

FETCH transactions\_cursor INTO v\_customer\_id, v\_transaction\_id, v\_transaction\_type, v\_amount, v\_transaction\_date;

EXIT WHEN transactions\_cursor%NOTFOUND;

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

', Transaction ID: ' || v\_transaction\_id ||

', Type: ' || v\_transaction\_type ||

', Amount: ' || v\_amount ||

', Date: ' || v\_transaction\_date);

END LOOP;

CLOSE transactions\_cursor;

END;

Write a PL/SQL block using an explicit cursor ApplyAnnualFee that deducts an annual maintenance fee from the balance of all accounts.

DECLARE

CURSOR accounts\_cursor IS

SELECT account\_id, balance

FROM accounts;

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

v\_balance accounts.balance%TYPE;

v\_annual\_fee CONSTANT NUMBER := 50; -- Assuming an annual fee of $50

BEGIN

OPEN accounts\_cursor;

LOOP

FETCH accounts\_cursor INTO v\_account\_id, v\_balance;

EXIT WHEN accounts\_cursor%NOTFOUND;

UPDATE accounts

SET balance = balance - v\_annual\_fee

WHERE account\_id = v\_account\_id;

END LOOP;

CLOSE accounts\_cursor;

COMMIT;

END;

Write a PL/SQL block using an explicit cursor UpdateLoanInterestRates that fetches all loans and updates their interest rates based on the new policy.

DECLARE

CURSOR loans\_cursor IS

SELECT loan\_id, interest\_rate

FROM loans;

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

v\_interest\_rate loans.interest\_rate%TYPE;

v\_new\_interest\_rate loans.interest\_rate%TYPE;

BEGIN

OPEN loans\_cursor;

LOOP

FETCH loans\_cursor INTO v\_loan\_id, v\_interest\_rate;

EXIT WHEN loans\_cursor%NOTFOUND;

-- Apply new policy for interest rate update

v\_new\_interest\_rate := v\_interest\_rate + 0.5; -- Example: increasing interest rate by 0.5%

UPDATE loans

SET interest\_rate = v\_new\_interest\_rate

WHERE loan\_id = v\_loan\_id;

END LOOP;

CLOSE loans\_cursor;

COMMIT;

END;