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

* + **Question:** 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.

->**Solution:**

DECLARE

CURSOR cur\_Transactions IS

SELECT c.CustomerID, c.Name, t.TransactionDate, t.Amount, t.TransactionType

FROM Customers c

JOIN Accounts a ON c.CustomerID = a.CustomerID

JOIN Transactions t ON a.AccountID = t.AccountID

WHERE t.TransactionDate >= TRUNC(SYSDATE, 'MM')

AND t.TransactionDate < ADD\_MONTHS(TRUNC(SYSDATE, 'MM'), 1)

ORDER BY c.CustomerID, t.TransactionDate;

v\_CustomerID Customers.CustomerID%TYPE;

v\_Name Customers.Name%TYPE;

v\_TransactionDate Transactions.TransactionDate%TYPE;

v\_Amount Transactions.Amount%TYPE;

v\_TransactionType Transactions.TransactionType%TYPE;

BEGIN

OPEN cur\_Transactions;

LOOP

FETCH cur\_Transactions INTO v\_CustomerID, v\_Name, v\_TransactionDate, v\_Amount, v\_TransactionType;

EXIT WHEN cur\_Transactions%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || v\_CustomerID || ' Name: ' || v\_Name || ' Date: ' || v\_TransactionDate || ' Amount: ' || v\_Amount || ' Type: ' || v\_TransactionType);

END LOOP;

CLOSE cur\_Transactions;

END;

/

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

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

->**Solution:**

DECLARE

CURSOR cur\_Accounts IS

SELECT AccountID, Balance

FROM Accounts;

v\_AccountID Accounts.AccountID%TYPE;

v\_Balance Accounts.Balance%TYPE;

v\_AnnualFee CONSTANT NUMBER := 50; -- Annual fee amount

BEGIN

OPEN cur\_Accounts;

LOOP

FETCH cur\_Accounts INTO v\_AccountID, v\_Balance;

EXIT WHEN cur\_Accounts%NOTFOUND;

UPDATE Accounts

SET Balance = Balance - v\_AnnualFee

WHERE AccountID = v\_AccountID;

DBMS\_OUTPUT.PUT\_LINE('Applied annual fee to Account ID: ' || v\_AccountID || '. New Balance: ' || (v\_Balance - v\_AnnualFee));

END LOOP;

CLOSE cur\_Accounts;

COMMIT;

END;

/

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

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

->**Solution:**

DECLARE

CURSOR cur\_Loans IS

SELECT LoanID, InterestRate

FROM Loans;

v\_LoanID Loans.LoanID%TYPE;

v\_InterestRate Loans.InterestRate%TYPE;

v\_NewInterestRate Loans.InterestRate%TYPE;

BEGIN

OPEN cur\_Loans;

LOOP

FETCH cur\_Loans INTO v\_LoanID, v\_InterestRate;

EXIT WHEN cur\_Loans%NOTFOUND;

-- Example of new policy: increase interest rate by 0.5%

v\_NewInterestRate := v\_InterestRate + 0.5;

UPDATE Loans

SET InterestRate = v\_NewInterestRate

WHERE LoanID = v\_LoanID;

DBMS\_OUTPUT.PUT\_LINE('Updated Loan ID: ' || v\_LoanID || ' New Interest Rate: ' || v\_NewInterestRate);

END LOOP;

CLOSE cur\_Loans;

COMMIT;

END;

/