**Scenario 1: Generate Monthly Statements for All Customers**

This PL/SQL block uses an explicit cursor GenerateMonthlyStatements to retrieve all transactions for the current month and print a statement for each customer.

DECLARE

CURSOR transaction\_cursor IS

SELECT customer\_id, transaction\_date, transaction\_amount

FROM transactions

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

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

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

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

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

BEGIN

OPEN transaction\_cursor;

LOOP

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

EXIT WHEN transaction\_cursor%NOTFOUND;

-- Print the statement (this example just outputs to console, adjust as needed)

DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' || v\_customer\_id || ', Date: ' || v\_transaction\_date || ', Amount: ' || v\_transaction\_amount);

END LOOP;

CLOSE transaction\_cursor;

END;

/

**Scenario 2: Apply Annual Fee to All Accounts**

This PL/SQL block uses an explicit cursor ApplyAnnualFee to deduct an annual maintenance fee from the balance of all accounts.

DECLARE

CURSOR account\_cursor 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 account\_cursor;

LOOP

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

EXIT WHEN account\_cursor%NOTFOUND;

-- Deduct the annual fee

UPDATE accounts

SET balance = balance - v\_annual\_fee

WHERE account\_id = v\_account\_id;

END LOOP;

CLOSE account\_cursor;

COMMIT;

END;

/

**Scenario 3: Update Interest Rates for All Loans Based on a New Policy**

This PL/SQL block uses an explicit cursor UpdateLoanInterestRates to fetch all loans and update their interest rates based on the new policy.

DECLARE

CURSOR loan\_cursor IS

SELECT loan\_id, interest\_rate

FROM loans;

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

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

v\_new\_interest\_rate NUMBER; -- Define the new interest rate policy

BEGIN

-- Example: Increase the interest rate by 0.5%

v\_new\_interest\_rate := 0.5;

OPEN loan\_cursor;

LOOP

FETCH loan\_cursor INTO v\_loan\_id, v\_current\_interest\_rate;

EXIT WHEN loan\_cursor%NOTFOUND;

-- Update the interest rate based on the new policy

UPDATE loans

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

WHERE loan\_id = v\_loan\_id;

END LOOP;

CLOSE loan\_cursor;

COMMIT;

END;

/

**Summary**

* **Scenario 1**: GenerateMonthlyStatements uses an explicit cursor to retrieve and print all transactions for the current month.
* **Scenario 2**: ApplyAnnualFee uses an explicit cursor to deduct an annual maintenance fee from the balance of all accounts.
* **Scenario 3**: UpdateLoanInterestRates uses an explicit cursor to update the interest rates for all loans based on a new policy.