**Exercise 6: Cursors**

**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.

**CODE:**

INSERT INTO transactions (txn\_id, account\_id, txn\_type, amount, txn\_date)

VALUES (5, 101, 'deposit', 500, SYSDATE);

INSERT INTO transactions (txn\_id, account\_id, txn\_type, amount, txn\_date)

VALUES (6, 101, 'withdraw', 200, SYSDATE);

INSERT INTO transactions (txn\_id, account\_id, txn\_type, amount, txn\_date)

VALUES (7, 102, 'deposit', 1000, SYSDATE);

COMMIT;

DECLARE

CURSOR txn\_cursor IS

SELECT account\_id, txn\_type, amount, txn\_date

FROM transactions

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

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

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

v\_type transactions.txn\_type%TYPE;

v\_amount transactions.amount%TYPE;

v\_date transactions.txn\_date%TYPE;

BEGIN

OPEN txn\_cursor;

LOOP

FETCH txn\_cursor INTO v\_account\_id, v\_type, v\_amount, v\_date;

EXIT WHEN txn\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Account: ' || v\_account\_id ||

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

', Amount: ₹' || v\_amount ||

', Date: ' || TO\_CHAR(v\_date, 'DD-MON-YYYY'));

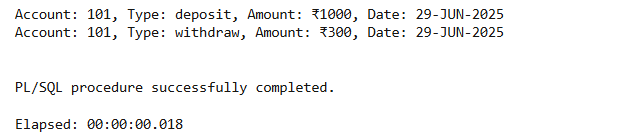
END LOOP;

CLOSE txn\_cursor;

END;

/

**OUTPUT:**



**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.

**CODE:**

DECLARE

CURSOR acc\_cursor IS

SELECT accountid, balance FROM accounts;

v\_acc\_id accounts.accountid%TYPE;

v\_balance accounts.balance%TYPE;

BEGIN

OPEN acc\_cursor;

LOOP

FETCH acc\_cursor INTO v\_acc\_id, v\_balance;

EXIT WHEN acc\_cursor%NOTFOUND;

UPDATE accounts

SET balance = balance - 100

WHERE accountid = v\_acc\_id;

DBMS\_OUTPUT.PUT\_LINE('₹100 annual fee deducted from Account: ' || v\_acc\_id);

END LOOP;

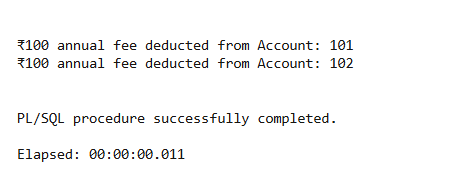
CLOSE acc\_cursor;

COMMIT;

END;

/

**OUTPUT:**

****

**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.

**CODE:**

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE loans';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

amount NUMBER(10,2),

interest\_rate NUMBER(5,2)

);

INSERT INTO loans VALUES (1, 1, 50000, 9.5);

INSERT INTO loans VALUES (2, 2, 80000, 10.0);

INSERT INTO loans VALUES (3, 3, 100000, 11.5);

COMMIT;

DECLARE

CURSOR loan\_cursor IS

SELECT loan\_id, interest\_rate FROM loans;

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

v\_rate loans.interest\_rate%TYPE;

BEGIN

OPEN loan\_cursor;

LOOP

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

EXIT WHEN loan\_cursor%NOTFOUND;

IF v\_rate < 10 THEN

UPDATE loans

SET interest\_rate = v\_rate + 1

WHERE loan\_id = v\_loan\_id;

ELSE

UPDATE loans

SET interest\_rate = v\_rate + 0.5

WHERE loan\_id = v\_loan\_id;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Updated Loan ID ' || v\_loan\_id || ' to new rate.');

END LOOP;

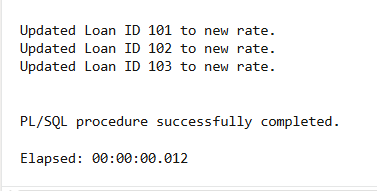
CLOSE loan\_cursor;

COMMIT;

END;

/

**OUTPUT:**

****