**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.

CREATE TABLE transactions (

txn\_id NUMBER PRIMARY KEY,

acc\_no NUMBER,

txn\_type VARCHAR2(20),

amount NUMBER,

txn\_date DATE

);

INSERT INTO transactions VALUES (2001, 401, 'deposit', 1000, SYSDATE);

INSERT INTO transactions VALUES (2002, 401, 'withdraw', 500, SYSDATE - 2);

INSERT INTO transactions VALUES (2003, 402, 'deposit', 2000, SYSDATE - 40);

COMMIT;

DECLARE

CURSOR txn\_cursor IS

SELECT acc\_no, txn\_type, amount, txn\_date

FROM transactions

WHERE TO\_CHAR(txn\_date, 'MMYYYY') = TO\_CHAR(SYSDATE, 'MMYYYY');

v\_acc\_no transactions.acc\_no%TYPE;

v\_type transactions.txn\_type%TYPE;

v\_amt transactions.amount%TYPE;

v\_date transactions.txn\_date%TYPE;

BEGIN

OPEN txn\_cursor;

LOOP

FETCH txn\_cursor INTO v\_acc\_no, v\_type, v\_amt, v\_date;

EXIT WHEN txn\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Account: ' || v\_acc\_no || ' | Type: ' || v\_type || ' | Amount: ' || v\_amt || ' | Date: ' || TO\_CHAR(v\_date, 'DD-Mon-YYYY'));

END LOOP;

CLOSE txn\_cursor;

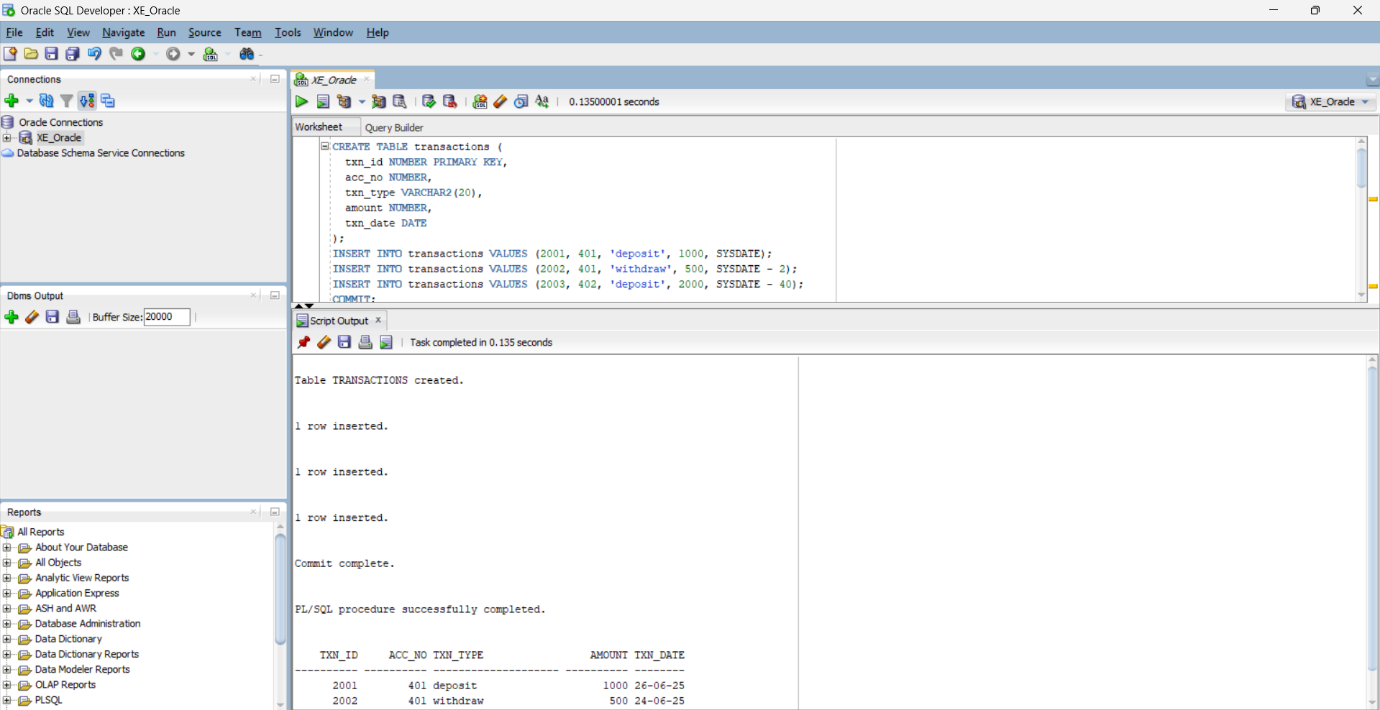
END;

/

--example usage

SELECT \* FROM transactions

WHERE TO\_CHAR(txn\_date, 'MMYYYY') = TO\_CHAR(SYSDATE, 'MMYYYY');



**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.

DECLARE

CURSOR acc\_cursor IS

SELECT acc\_no, balance FROM accounts;

v\_acc\_no accounts.acc\_no%TYPE;

v\_balance accounts.balance%TYPE;

annual\_fee CONSTANT NUMBER := 500;

BEGIN

OPEN acc\_cursor;

LOOP

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

EXIT WHEN acc\_cursor%NOTFOUND;

UPDATE accounts

SET balance = balance - annual\_fee

WHERE acc\_no = v\_acc\_no;

DBMS\_OUTPUT.PUT\_LINE('Annual fee applied to Account: ' || v\_acc\_no);

END LOOP;

CLOSE acc\_cursor;

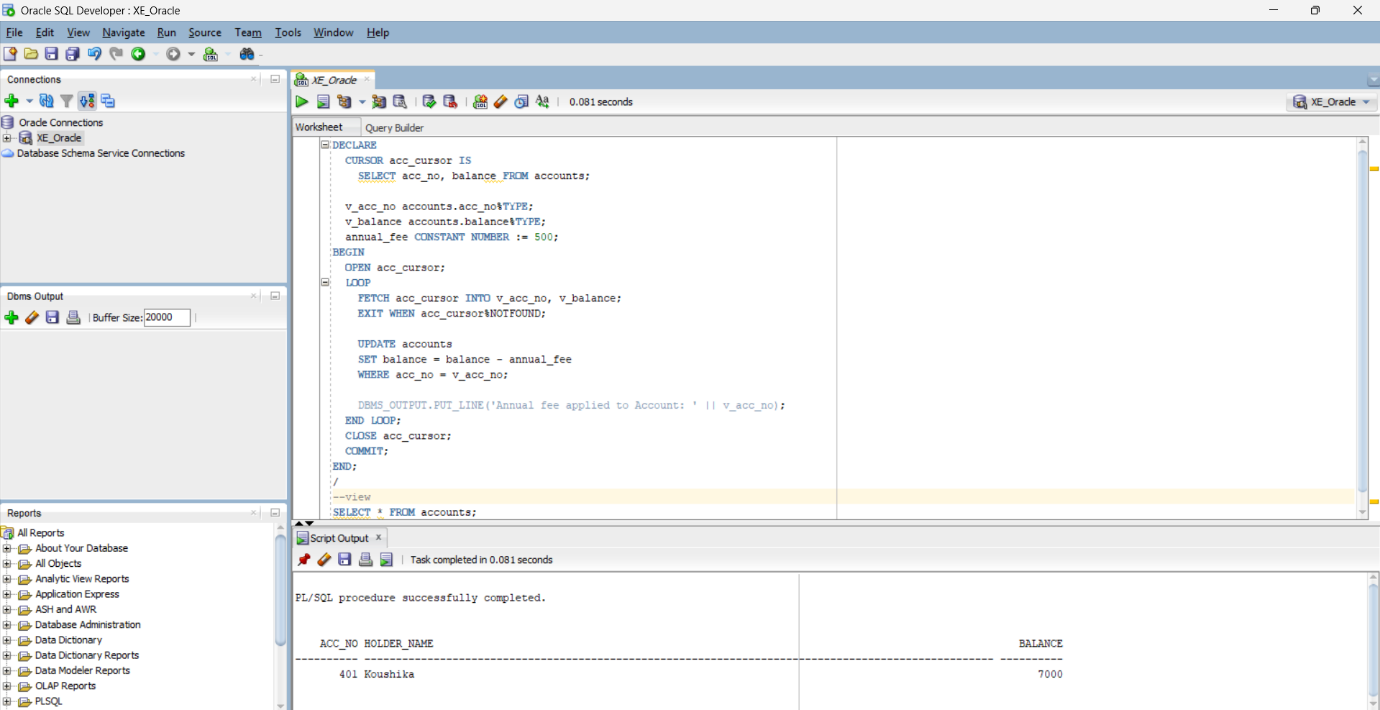
COMMIT;

END;

/

--view

SELECT \* FROM accounts;



**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.

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

interest\_rate NUMBER,

due\_date DATE

);

INSERT INTO loans VALUES (301, 1, 9.5, SYSDATE + 180);

INSERT INTO loans VALUES (302, 2, 8.0, SYSDATE + 90);

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;

UPDATE loans

SET interest\_rate = v\_rate + 0.5

WHERE loan\_id = v\_loan\_id;

DBMS\_OUTPUT.PUT\_LINE('Updated interest for Loan ID: ' || v\_loan\_id);

END LOOP;

CLOSE loan\_cursor;

COMMIT;

END;

/

--view

SELECT \* FROM loans;  
