**Exercise 3: Stored Procedures**

**Scenario 1:** The bank needs to process monthly interest for all savings accounts.

**Question:** Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

CREATE TABLE savings\_accounts (

acc\_no NUMBER PRIMARY KEY,

holder\_name VARCHAR2(100),

balance NUMBER

);

INSERT INTO savings\_accounts VALUES (201, 'Koushika', 10000);

INSERT INTO savings\_accounts VALUES (202, 'Madhu', 15000);

COMMIT;

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE savings\_accounts

SET balance = balance + (balance \* 0.01);

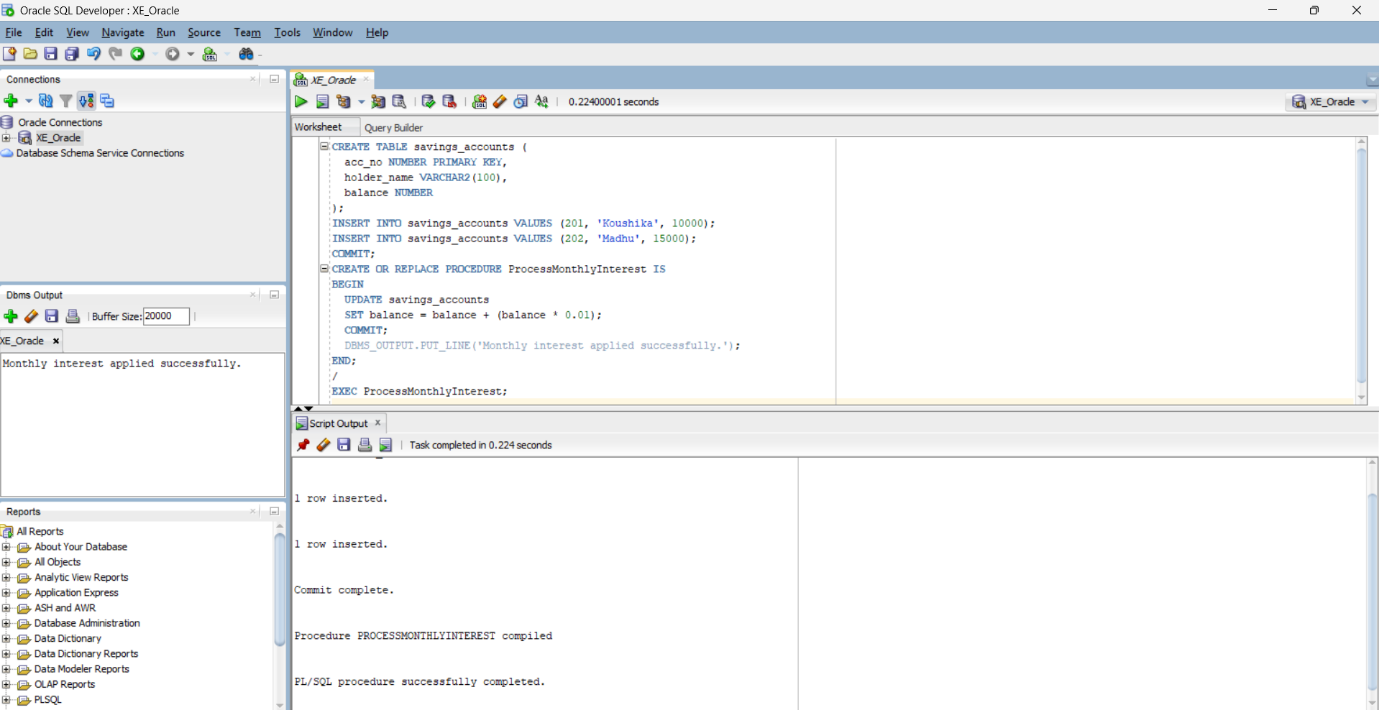
COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Monthly interest applied successfully.');

END;

/

EXEC ProcessMonthlyInterest;



**Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

**Question:** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.  
  
CREATE TABLE employees (

emp\_id NUMBER PRIMARY KEY,

emp\_name VARCHAR2(100),

department VARCHAR2(50),

salary NUMBER

);

INSERT INTO employees VALUES (301, 'Siva', 'IT', 50000);

INSERT INTO employees VALUES (302, 'Arun', 'HR', 40000);

INSERT INTO employees VALUES (303, 'Meena', 'IT', 55000);

COMMIT;

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

dept\_name IN VARCHAR2,

bonus\_percent IN NUMBER

) IS

BEGIN

UPDATE employees

SET salary = salary + (salary \* bonus\_percent / 100)

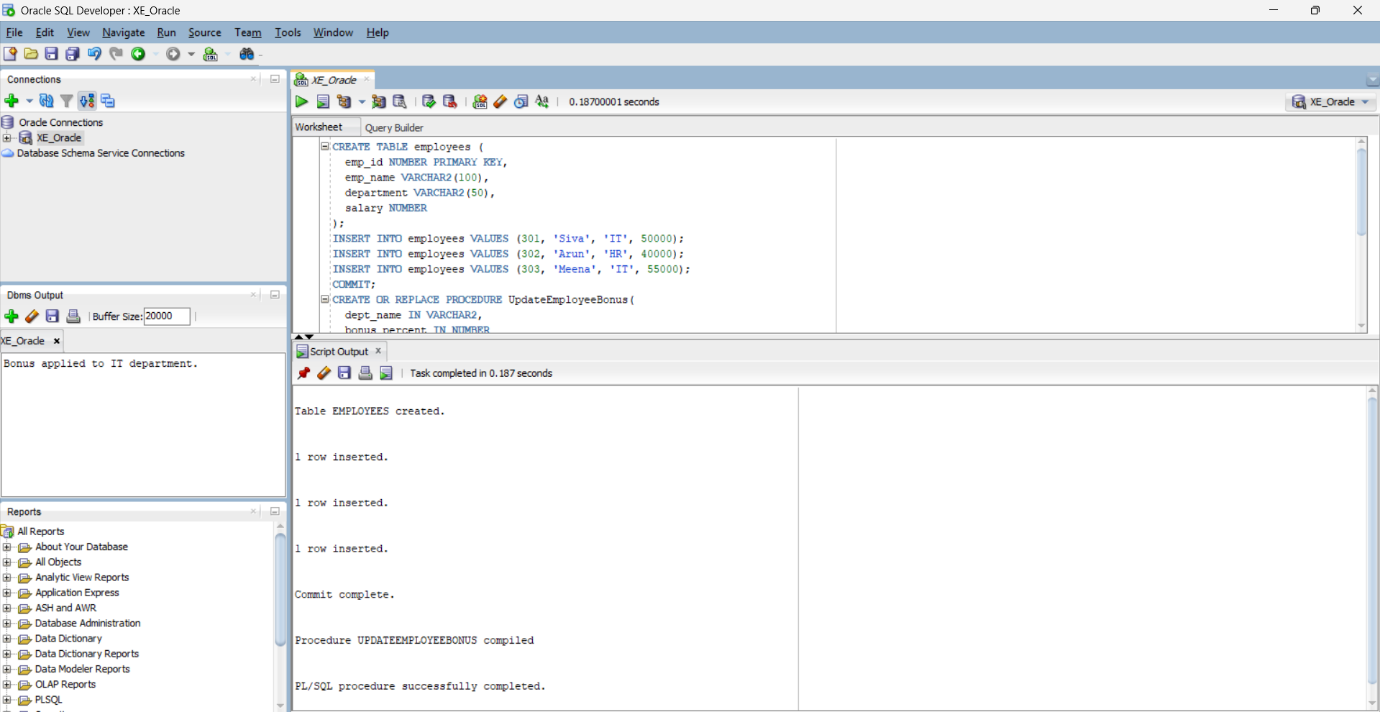
WHERE department = dept\_name;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to ' || dept\_name || ' department.');

END;

/

EXEC UpdateEmployeeBonus('IT', 10);  
  


**Scenario 3:** Customers should be able to transfer funds between their accounts.

**Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

CREATE TABLE accounts (

acc\_no NUMBER PRIMARY KEY,

holder\_name VARCHAR2(100),

balance NUMBER

);

INSERT INTO accounts VALUES (401, 'Koushika', 8000);

INSERT INTO accounts VALUES (402, 'Madhu', 3000);

COMMIT;

CREATE OR REPLACE PROCEDURE TransferFunds(

from\_acc IN NUMBER,

to\_acc IN NUMBER,

amount IN NUMBER

) IS

BEGIN

UPDATE accounts

SET balance = balance - amount

WHERE acc\_no = from\_acc AND balance >= amount;

IF SQL%ROWCOUNT = 0 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance or invalid source account.');

END IF;

UPDATE accounts

SET balance = balance + amount

WHERE acc\_no = to\_acc;

IF SQL%ROWCOUNT = 0 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Invalid destination account.');

END IF;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transfer completed successfully.');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Transfer failed: ' || SQLERRM);

END;

/

EXEC TransferFunds(401, 402, 1000);

