**Stored Procedures**

**Scenario 1:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

  UPDATE Accounts

  SET Balance = Balance + (Balance \* 0.01)

  WHERE AccountType = 'Savings';

  DBMS\_OUTPUT.PUT\_LINE('Interest added to all the savings accounts.');

  COMMIT;

EXCEPTION

  WHEN OTHERS THEN

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

    ROLLBACK;

END;

/

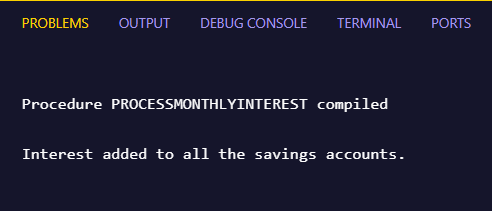
BEGIN

  ProcessMonthlyInterest;

END;

/

**Output:**

****

**Scenario 2:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

  dept IN VARCHAR2,

  pct  IN NUMBER

) AS

BEGIN

  UPDATE Employees

  SET Salary = Salary + (Salary \* pct / 100)

  WHERE Department = dept;

  IF SQL%ROWCOUNT = 0 THEN

    DBMS\_OUTPUT.PUT\_LINE('No employees found in the department ' || dept);

  ELSE

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

  END IF;

  COMMIT;

EXCEPTION

  WHEN OTHERS THEN

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

    ROLLBACK;

END;

/

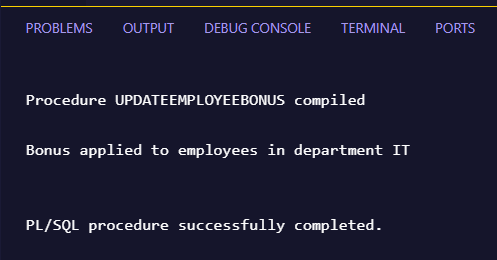
BEGIN

  UpdateEmployeeBonus('IT', 5);

END;

**/**

**Output:**

****

**Scenario 3:**

CREATE OR REPLACE PROCEDURE TransferFunds(

  from\_id IN NUMBER,

  to\_id   IN NUMBER,

  amt     IN NUMBER

) AS

  bal NUMBER;

BEGIN

  SELECT Balance INTO bal

  FROM Accounts

  WHERE AccountID = from\_id FOR UPDATE;

  IF bal < amt THEN

    RAISE\_APPLICATION\_ERROR(-20005, 'Insufficient balance.');

  END IF;

  UPDATE Accounts SET Balance = Balance - amt WHERE AccountID = from\_id;

  UPDATE Accounts SET Balance = Balance + amt WHERE AccountID = to\_id;

  DBMS\_OUTPUT.PUT\_LINE('Funds transferred.');

  COMMIT;

EXCEPTION

  WHEN OTHERS THEN

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

    ROLLBACK;

END;

/

**If balance is sufficient:**

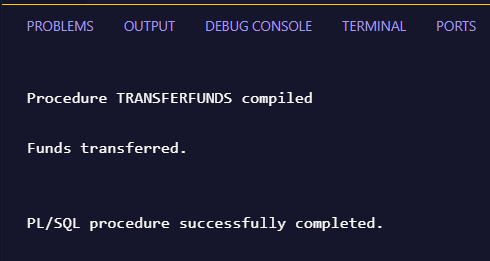
BEGIN

  TransferFunds(1, 2, 500);

END;

/

**Output:**

****

**If balance is Insufficient:**

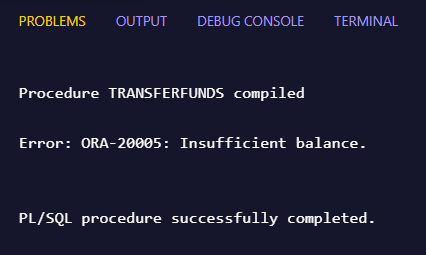
BEGIN

  TransferFunds(1, 2, 1500);

END;

/

**Output:**

****