**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.

**Program:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

FOR acc IN (

SELECT AccountID, Balance

FROM Accounts

WHERE AccountType = 'Savings'

) LOOP

UPDATE Accounts

SET Balance = Balance + (acc.Balance \* 0.01)

WHERE AccountID = acc.AccountID;

DBMS\_OUTPUT.PUT\_LINE(

'1% interest added to Account ID: ' || acc.AccountID ||

', New Balance: ' || TO\_CHAR(acc.Balance \* 1.01, '9999.99')

);

END LOOP;

COMMIT;

END;

/

**Executing the Procedure :**

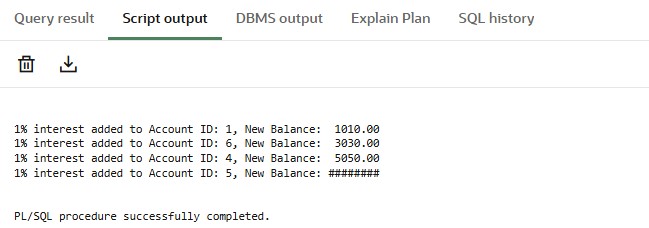
BEGIN

ProcessMonthlyInterest;

END;

/

**Output:**

****

**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.

**Program** :

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_dept IN VARCHAR2,

p\_bonus\_percent IN NUMBER

) IS

BEGIN

FOR emp IN (

SELECT EmployeeID, Name, Salary

FROM Employees

WHERE Department = p\_dept

) LOOP

UPDATE Employees

SET Salary = Salary + (emp.Salary \* (p\_bonus\_percent / 100))

WHERE EmployeeID = emp.EmployeeID;

DBMS\_OUTPUT.PUT\_LINE(

'Bonus of ' || p\_bonus\_percent || '% applied to ' || emp.Name ||

' (ID: ' || emp.EmployeeID || '), New Salary: ' ||

TO\_CHAR(emp.Salary \* (1 + p\_bonus\_percent / 100), '999999.99')

);

END LOOP;

COMMIT;

END;

/

**Executing the Procedure:**

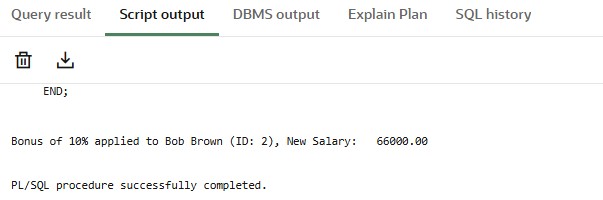
BEGIN

UpdateEmployeeBonus('IT', 10);

END;

/

**Output:**

****

**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.

**Program:**

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_account\_id IN NUMBER,

p\_to\_account\_id IN NUMBER,

p\_amount IN NUMBER

) IS

v\_from\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_from\_balance

FROM Accounts

WHERE AccountID = p\_from\_account\_id;

IF v\_from\_balance >= p\_amount THEN

UPDATE Accounts

SET Balance = Balance - p\_amount

WHERE AccountID = p\_from\_account\_id;

UPDATE Accounts

SET Balance = Balance + p\_amount

WHERE AccountID = p\_to\_account\_id;

DBMS\_OUTPUT.PUT\_LINE('Transfer of ₹' || p\_amount || ' successful from Account ' ||

p\_from\_account\_id || ' to Account ' || p\_to\_account\_id);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Insufficient funds in Account ' || p\_from\_account\_id);

END IF;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('One of the accounts does not exist.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An unexpected error occurred: ' || SQLERRM);

END;

/

**Executing the Procedure**:

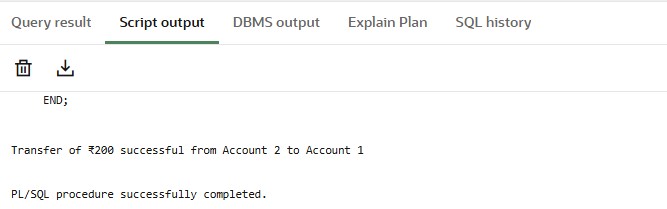
BEGIN

TransferFunds(2, 1, 200);

END;

/

**Output:**

****