**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.

**PL/SQL BLOCK:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

FOR acc IN (

SELECT AccountID, Balance

FROM Accounts

WHERE AccountType = 'Savings'

) LOOP

UPDATE Accounts

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

LastModified = SYSDATE

WHERE AccountID = acc.AccountID;

END LOOP;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Monthly interest processed for all Savings accounts.');

END;

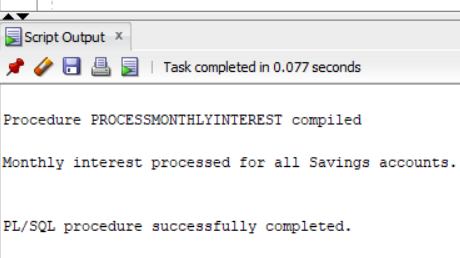
**To Execute the Procedure:**

BEGIN

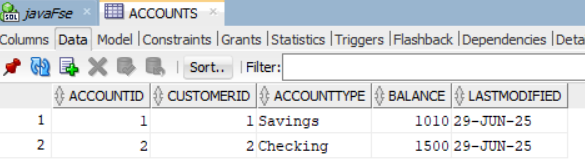
ProcessMonthlyInterest;

END;

**OUTPUT:**

****

**TABLE AFTER RUNNING THE BLOCK:**

****

**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.

**PL/SQL BLOCK:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

dept IN VARCHAR2,

bonus\_pct IN NUMBER

) AS

BEGIN

UPDATE Employees

SET Salary = Salary + (Salary \* bonus\_pct / 100)

WHERE Department = dept;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Bonus of ' || bonus\_pct || '% applied to employees in department: ' || dept);

END;

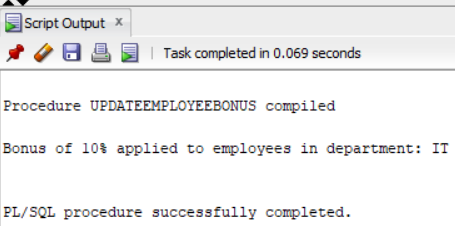
**TO EXECUTE THE PROCECURE:**

BEGIN

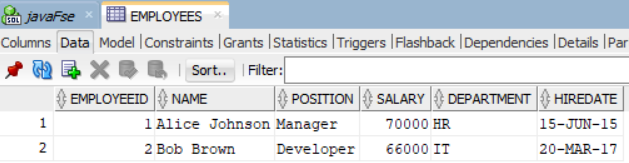
UpdateEmployeeBonus('IT', 10);

END;

**OUTPUT:**

****

**TABLE AFTER RUNNING THE BLOCK:**

****

**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.

**PL/SQL BLOCK:**

CREATE OR REPLACE PROCEDURE TransferFunds (

from\_account\_id IN NUMBER,

to\_account\_id IN NUMBER,

amount IN NUMBER

) AS

v\_from\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_from\_balance

FROM Accounts

WHERE AccountID = from\_account\_id;

IF v\_from\_balance < amount THEN

raise\_application\_error(-20001, 'Insufficient Balance in source account');

END IF;

UPDATE Accounts

SET Balance = Balance - amount,

LastModified = SYSDATE

WHERE AccountID = from\_account\_id;

UPDATE Accounts

SET Balance = Balance + amount,

LastModified = SYSDATE

WHERE AccountID = to\_account\_id;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transfer successful: ₹' || amount || ' transferred from AccountID ' ||

from\_account\_id || ' to AccountID ' || to\_account\_id);

END;

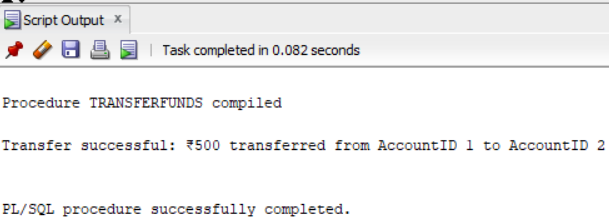
**TO EXECUTE THE PROCEDURE:**

BEGIN

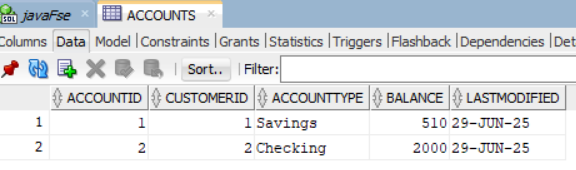
TransferFunds(1, 2, 500);

END;

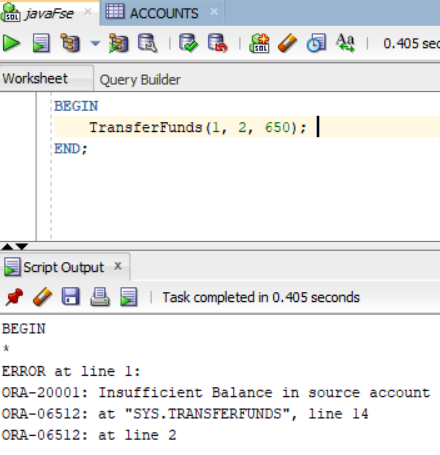
**OUPUT (SUCCESSFUL TRANSACTION):**

****

**TABLE AFTER EXECUTING THE PROCEDURE:**

****

**FAILED TRANSACTION:**

****