**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 IS

BEGIN

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'Savings';

DBMS\_OUTPUT.PUT\_LINE('Monthly interest (1%) applied to all Savings accounts.');

END;

/

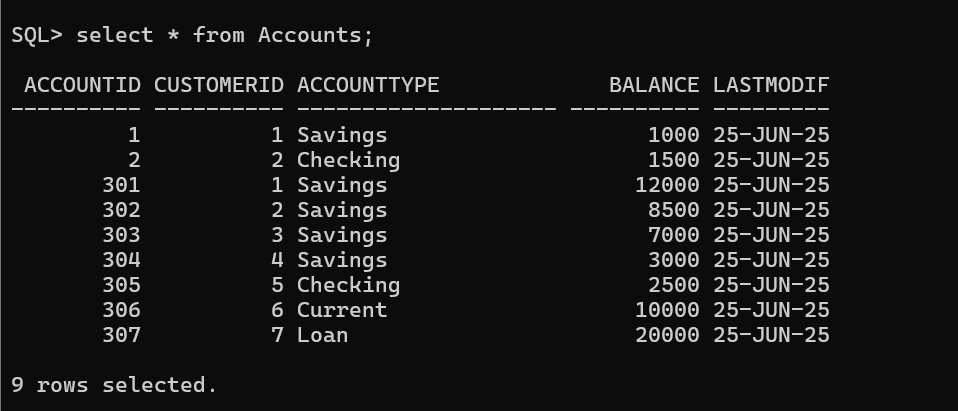
BEGIN

ProcessMonthlyInterest;

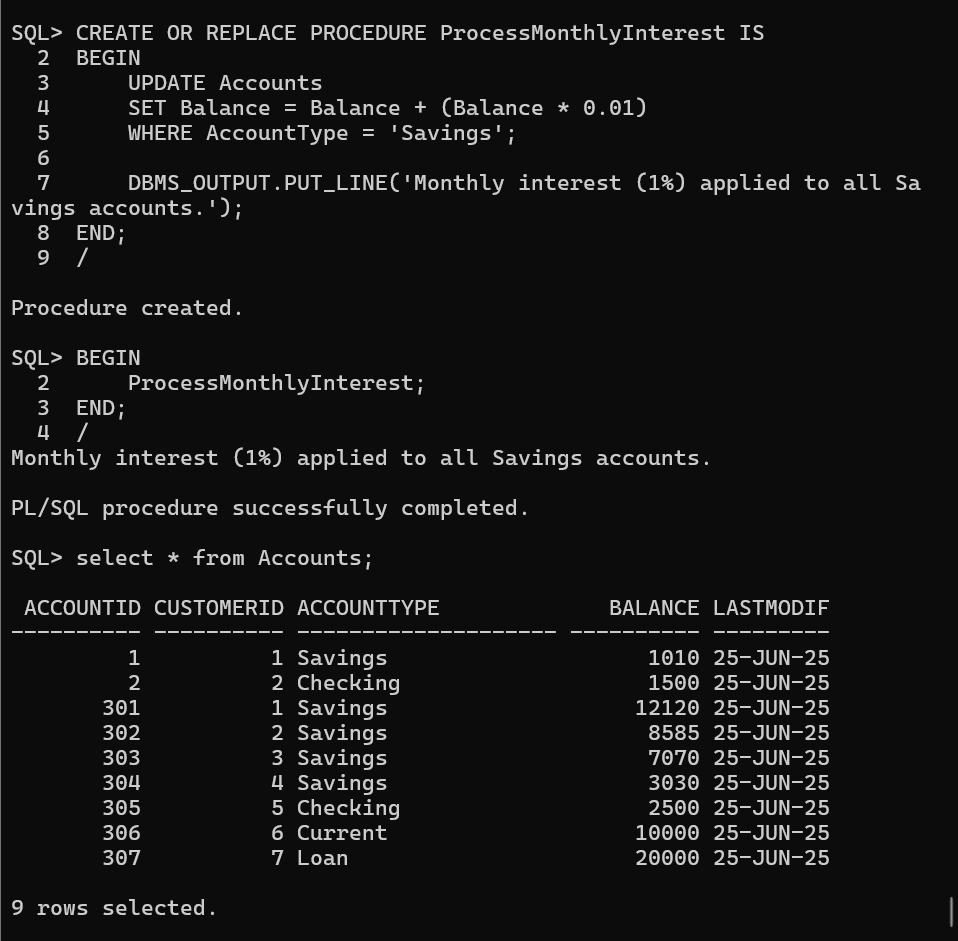
END;

/

**Table :-**

****

**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.

**PL/SQL Block :-**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_department IN VARCHAR2,

p\_bonus\_percent IN NUMBER

)

AS

v\_count NUMBER;

BEGIN

UPDATE Employees

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

WHERE UPPER(Department) = UPPER(p\_department);

v\_count := SQL%ROWCOUNT;

DBMS\_OUTPUT.PUT\_LINE(v\_count || ' employee(s) in department "' || p\_department || '" received a bonus of ' || p\_bonus\_percent || '%.');

END;

/

BEGIN

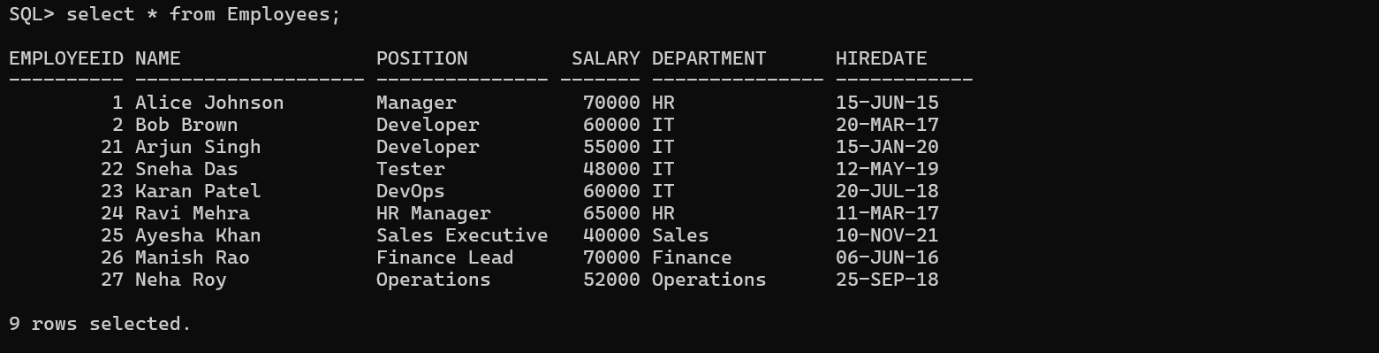
UpdateEmployeeBonus('IT', 10);

UpdateEmployeeBonus('Sales', 20);

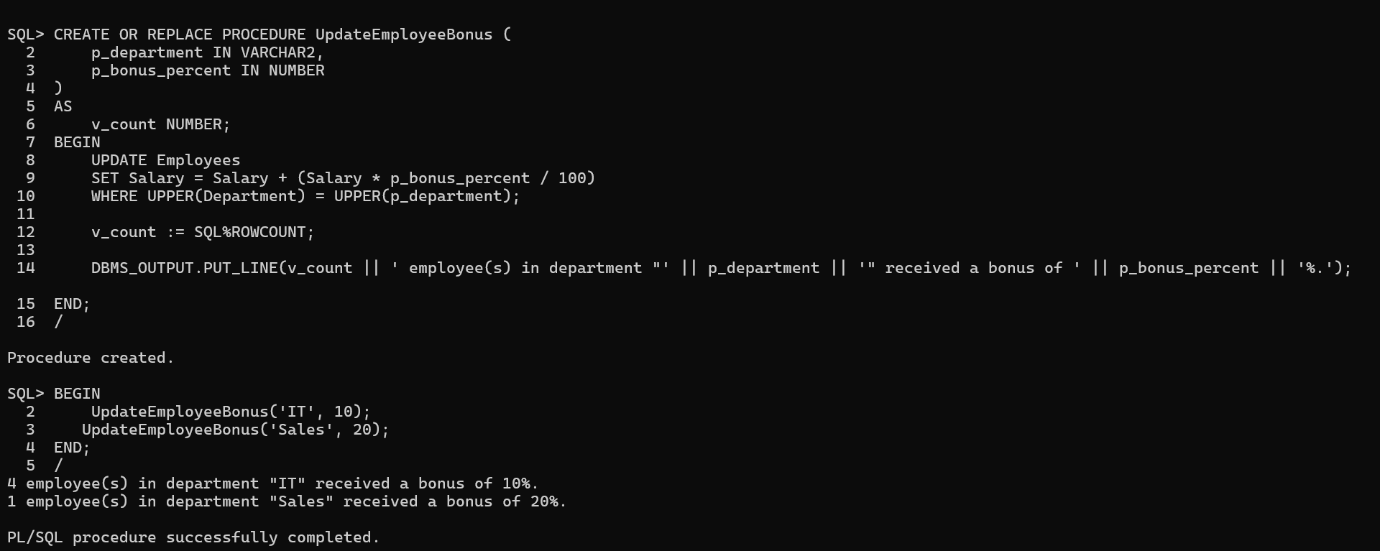
END;

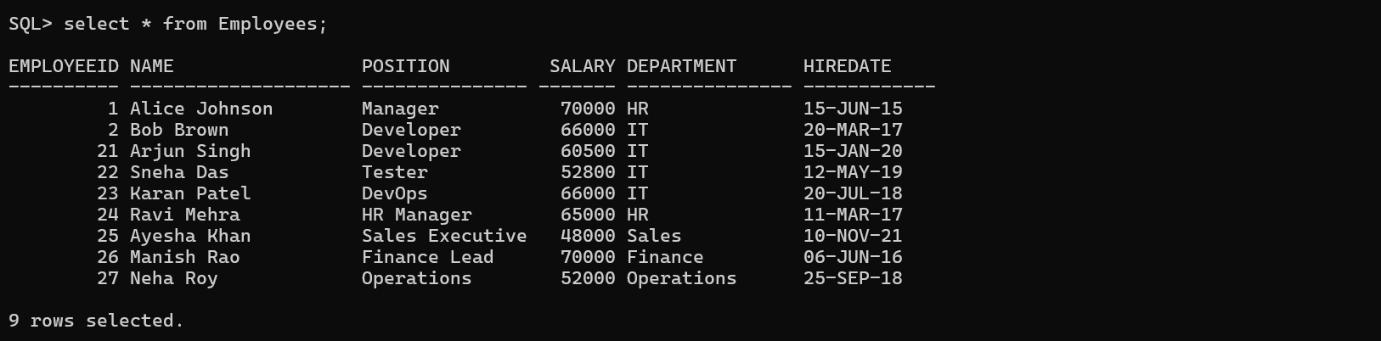
/

**Table :-**



**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.

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

)

AS

v\_balance NUMBER;

BEGIN

-- Step 1: Check if source account exists and get balance

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = p\_from\_account

FOR UPDATE;

-- Step 2: Check if sufficient balance exists

IF v\_balance < p\_amount THEN

DBMS\_OUTPUT.PUT\_LINE('Insufficient balance in Account ID: ' || p\_from\_account);

RETURN;

END IF;

-- Step 3: Deduct amount from source account

UPDATE Accounts

SET Balance = Balance - p\_amount

WHERE AccountID = p\_from\_account;

-- Step 4: Add amount to destination account

UPDATE Accounts

SET Balance = Balance + p\_amount

WHERE AccountID = p\_to\_account;

-- Step 5: Success message

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || p\_amount || ' from Account ' || p\_from\_account || ' to Account ' || p\_to\_account);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

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

WHEN OTHERS THEN

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

END;

/

BEGIN

TransferFunds(2, 1, 500); -- Transfers ₹500 from Account 2 to Account 1

END;

/

**Table :-**



**Output :-**

