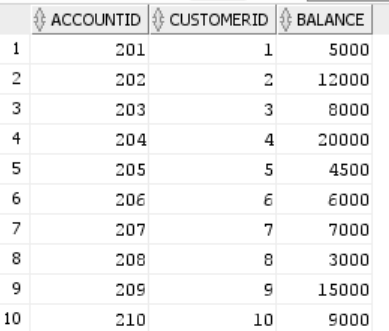
**PLSQL Exercise**

**Exercise 3: Stored Procedures**

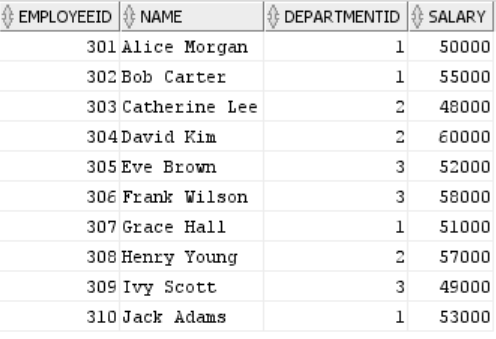
**Scenario 1:** The bank needs to process monthly interest for all savings accounts.

* + **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.

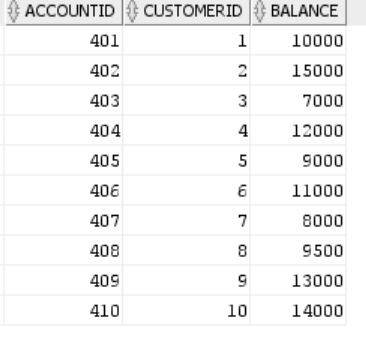
**SAVINGS\_ACCOUNT Table**



**EMPLOYEES Table**



**ACCOUNTS Table**



**Code:**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

UPDATE SAVINGS\_ACCOUNTS

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

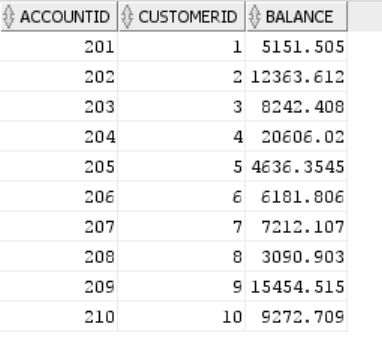
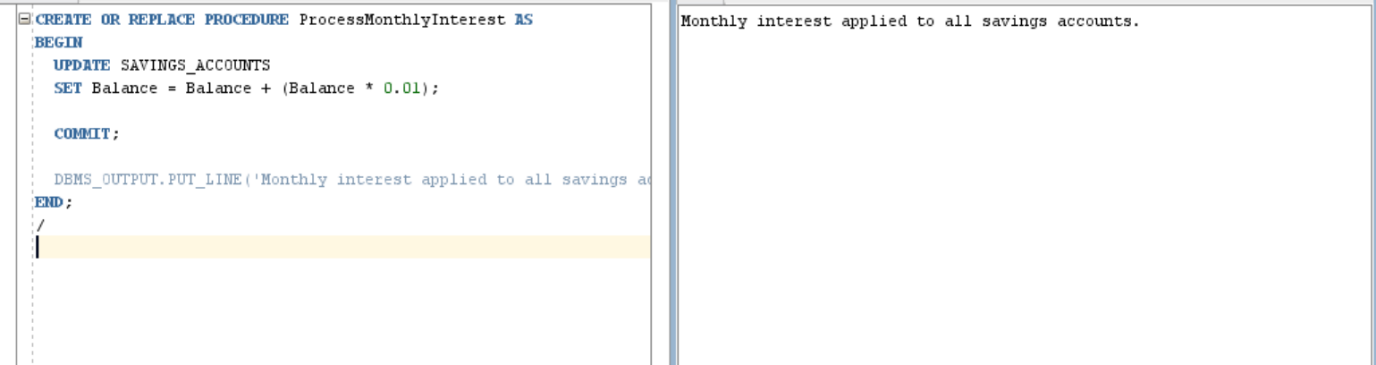
COMMIT;

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

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.

**Code:**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_DepartmentID IN NUMBER,

p\_BonusPercent IN NUMBER -- e.g., pass 10 for 10%

) AS

BEGIN

UPDATE EMPLOYEES

SET Salary = Salary + (Salary \* p\_BonusPercent / 100)

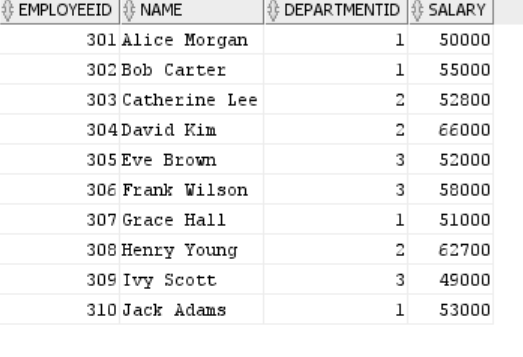
WHERE DepartmentID = p\_DepartmentID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Bonuses applied to employees in department ' || p\_DepartmentID);

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.

**Code:**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_FromAccountID IN NUMBER,

p\_ToAccountID IN NUMBER,

p\_Amount IN NUMBER

) AS

v\_SourceBalance NUMBER;

BEGIN

-- Get source account balance

SELECT Balance INTO v\_SourceBalance

FROM ACCOUNTS

WHERE AccountID = p\_FromAccountID

FOR UPDATE; -- Lock row for update

IF v\_SourceBalance < p\_Amount THEN

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

END IF;

-- Deduct from source

UPDATE ACCOUNTS

SET Balance = Balance - p\_Amount

WHERE AccountID = p\_FromAccountID;

-- Add to destination

UPDATE ACCOUNTS

SET Balance = Balance + p\_Amount

WHERE AccountID = p\_ToAccountID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || p\_Amount || ' from account ' || p\_FromAccountID || ' to account ' || p\_ToAccountID);

END;

/

call TransferFunds(401,402,500);

**Output:**

