**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.

**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.

**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.

Tables:

SavingsAccounts Table:

CREATE TABLE SavingsAccounts (

    AccountID    NUMBER PRIMARY KEY,

    CustomerID   NUMBER,

    Balance      NUMBER(12,2)

);

Employee table:

CREATE TABLE Employees (

    EmpID        NUMBER PRIMARY KEY,

    Name         VARCHAR2(100),

    Department   VARCHAR2(50),

    Salary       NUMBER(10,2)

);

Accounts Table:

CREATE TABLE Accounts (

    AccountID    NUMBER PRIMARY KEY,

    CustomerID   NUMBER,

    Balance      NUMBER(12,2)

);

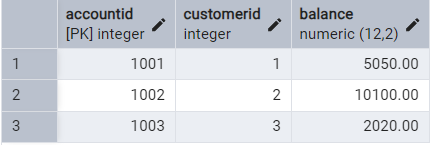
Insertion of data into tables:

Into SavingsAccounts:

INSERT INTO SavingsAccounts VALUES (1001, 1, 5000);

INSERT INTO SavingsAccounts VALUES (1002, 2, 10000);

INSERT INTO SavingsAccounts VALUES (1003, 3, 2000);



Into Employees:

INSERT INTO Employees VALUES (1, 'Alice', 'Sales', 50000);

INSERT INTO Employees VALUES (2, 'Bob', 'Sales', 60000);

INSERT INTO Employees VALUES (3, 'Charlie', 'HR', 55000);

A screenshot of a phone

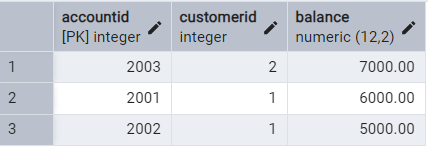
AI-generated content may be incorrect.

Into Accounts :

INSERT INTO Accounts VALUES (2001, 1, 8000);

INSERT INTO Accounts VALUES (2002, 1, 3000);

INSERT INTO Accounts VALUES (2003, 2, 7000);



COMMIT;

Scenario 1: Monthly Interest (1%) on SavingsAccounts

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

  FOR acc IN (SELECT AccountID, Balance FROM SavingsAccounts) LOOP

    UPDATE SavingsAccounts

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

    WHERE AccountID = acc.AccountID;

  END LOOP;

  COMMIT;

END;

To run:

BEGIN

  ProcessMonthlyInterest;

END;

Scenario 2: Employee Bonus by Department

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

  dept\_name IN VARCHAR2,

  bonus\_pct IN NUMBER

) IS

BEGIN

  UPDATE Employees

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

  WHERE Department = dept\_name;

  COMMIT;

END;

To run:

BEGIN

  UpdateEmployeeBonus('Sales', 10);

END;

Scenario 3: Transfer Funds Between Accounts

CREATE OR REPLACE PROCEDURE TransferFunds (

  from\_account IN NUMBER,

  to\_account IN NUMBER,

  amount IN NUMBER

) IS

  insufficient\_balance EXCEPTION;

  v\_balance NUMBER;

BEGIN

  -- Check balance

  SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = from\_account;

  IF v\_balance < amount THEN

    RAISE insufficient\_balance;

  END IF;

  -- Deduct from source

  UPDATE Accounts

  SET Balance = Balance - amount

  WHERE AccountID = from\_account;

  -- Add to target

  UPDATE Accounts

  SET Balance = Balance + amount

  WHERE AccountID = to\_account;

  COMMIT;

EXCEPTION

  WHEN insufficient\_balance THEN

    DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Insufficient balance.');

  WHEN NO\_DATA\_FOUND THEN

    DBMS\_OUTPUT.PUT\_LINE('Transfer failed: Account not found.');

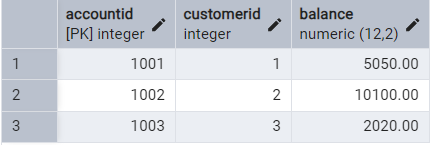
END;

BEGIN

  TransferFunds(2001, 2002, 2000);

END;

SELECT \* FROM SavingsAccounts;



SELECT \* FROM Employees;

A screenshot of a phone

AI-generated content may be incorrect.

SELECT \* FROM Accounts;

