Week-2

PL SQL programming

**Exercise 3: Stored Procedures**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE Accounts';**

**EXCEPTION**

**WHEN OTHERS THEN IF SQLCODE != -942 THEN RAISE; END IF;**

**END;**

**/**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE Employees';**

**EXCEPTION**

**WHEN OTHERS THEN IF SQLCODE != -942 THEN RAISE; END IF;**

**END;**

**/**

**-- CREATE SCHEMA**

**CREATE TABLE Accounts (**

**AccountID NUMBER PRIMARY KEY,**

**CustomerID NUMBER,**

**AccountType VARCHAR2(20),**

**Balance NUMBER,**

**LastModified DATE**

**);**

**CREATE TABLE Employees (**

**EmployeeID NUMBER PRIMARY KEY,**

**Name VARCHAR2(100),**

**Position VARCHAR2(50),**

**Salary NUMBER,**

**Department VARCHAR2(50),**

**HireDate DATE**

**);**

**-- INSERT SAMPLE DATA**

**INSERT INTO Accounts VALUES (1, 1, 'Savings', 1000, SYSDATE);**

**INSERT INTO Accounts VALUES (2, 2, 'Checking', 2000, SYSDATE);**

**INSERT INTO Employees VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR', SYSDATE);**

**INSERT INTO Employees VALUES (2, 'Bob Brown', 'Developer', 60000, 'IT', SYSDATE);**

**COMMIT;**

**SET SERVEROUTPUT ON**

**-- Scenario 1: Process monthly interest**

**CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS**

**BEGIN**

**UPDATE Accounts**

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

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

**END;**

**/**

**EXEC ProcessMonthlyInterest;**

**-- See updated balances**

**BEGIN**

**FOR acc IN (SELECT \* FROM Accounts) LOOP**

**DBMS\_OUTPUT.PUT\_LINE('AccountID: '||acc.AccountID||' Balance: '||acc.Balance);**

**END LOOP;**

**END;**

**/**

**-- Scenario 2: Update employee bonus**

**CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(dept VARCHAR2, bonus\_pct NUMBER) IS**

**BEGIN**

**UPDATE Employees**

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

**WHERE Department = dept;**

**DBMS\_OUTPUT.PUT\_LINE('Bonus applied to department: '||dept);**

**END;**

**/**

**EXEC UpdateEmployeeBonus('IT', 10);**

**-- See updated salaries**

**BEGIN**

**FOR emp IN (SELECT \* FROM Employees) LOOP**

**DBMS\_OUTPUT.PUT\_LINE('Employee: '||emp.Name||' Salary: '||emp.Salary);**

**END LOOP;**

**END;**

**/**

**-- Scenario 3: Transfer funds between customer accounts**

**CREATE OR REPLACE PROCEDURE TransferFunds(from\_acc NUMBER, to\_acc NUMBER, amt NUMBER) IS**

**src\_balance NUMBER;**

**BEGIN**

**SELECT Balance INTO src\_balance FROM Accounts WHERE AccountID = from\_acc;**

**IF src\_balance < amt THEN**

**DBMS\_OUTPUT.PUT\_LINE('Insufficient balance in account '||from\_acc);**

**RETURN;**

**END IF;**

**UPDATE Accounts SET Balance = Balance - amt WHERE AccountID = from\_acc;**

**UPDATE Accounts SET Balance = Balance + amt WHERE AccountID = to\_acc;**

**DBMS\_OUTPUT.PUT\_LINE('Transferred '||amt||' from Account '||from\_acc||' to Account '||to\_acc);**

**END;**

**/**

**EXEC TransferFunds(1, 2, 500);**

**BEGIN**

**FOR acc IN (SELECT \* FROM Accounts) LOOP**

**DBMS\_OUTPUT.PUT\_LINE('AccountID: '||acc.AccountID||' Balance: '||acc.Balance);**

**END LOOP;**

**END;**

**/**

