**Exercise 7: Packages**

**Scenario 1:** Group all customer-related procedures and functions into a package.

**Question:** Create a package **CustomerManagement** with procedures for adding a new customer, updating customer details, and a function to get customer balance.

**Answer:**

DELIMITER //

CREATE PROCEDURE AddCustomer(

IN p\_CustomerID INT,

IN p\_Name VARCHAR(100),

IN p\_DOB DATE,

IN p\_Balance DECIMAL(10,2)

)

BEGIN

INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)

VALUES (p\_CustomerID, p\_Name, p\_DOB, p\_Balance, CURRENT\_DATE());

END//

DELIMITER ;

DELIMITER //

CREATE PROCEDURE UpdateCustomerDetails(

IN p\_CustomerID INT,

IN p\_Name VARCHAR(100),

IN p\_DOB DATE

)

BEGIN

UPDATE Customers

SET Name = p\_Name, DOB = p\_DOB, LastModified = CURRENT\_DATE()

WHERE CustomerID = p\_CustomerID;

END//

DELIMITER ;

DELIMITER //

CREATE FUNCTION GetCustomerBalance(p\_CustomerID INT)

RETURNS DECIMAL(10,2)

BEGIN

DECLARE balance DECIMAL(10,2);

SELECT Balance INTO balance

FROM Customers

WHERE CustomerID = p\_CustomerID;

RETURN balance;

END//

DELIMITER ;

**Scenario 2:** Create a package to manage employee data.

**Question:** Write a package **EmployeeManagement** with procedures to hire new employees, update employee details, and a function to calculate annual salary.

**Answer:**

DELIMITER //

CREATE PROCEDURE HireEmployee(

IN p\_EmployeeID INT,

IN p\_Name VARCHAR(100),

IN p\_Position VARCHAR(50),

IN p\_Salary DECIMAL(10,2),

IN p\_Department VARCHAR(50),

IN p\_HireDate DATE

)

BEGIN

INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)

VALUES (p\_EmployeeID, p\_Name, p\_Position, p\_Salary, p\_Department, p\_HireDate);

END//

DELIMITER ;

DELIMITER //

CREATE PROCEDURE UpdateEmployeeDetails(

IN p\_EmployeeID INT,

IN p\_Name VARCHAR(100),

IN p\_Position VARCHAR(50),

IN p\_Salary DECIMAL(10,2),

IN p\_Department VARCHAR(50)

)

BEGIN

UPDATE Employees

SET Name = p\_Name, Position = p\_Position, Salary = p\_Salary, Department = p\_Department

WHERE EmployeeID = p\_EmployeeID;

END//

DELIMITER ;

DELIMITER //

CREATE FUNCTION CalculateAnnualSalary(p\_EmployeeID INT)

RETURNS DECIMAL(10,2)

BEGIN

DECLARE annualSalary DECIMAL(10,2);

SELECT Salary \* 12 INTO annualSalary

FROM Employees

WHERE EmployeeID = p\_EmployeeID;

RETURN annualSalary;

END//

DELIMITER ;

**Scenario 3:** Group all account-related operations into a package.

**Question:** Create a package **AccountOperations** with procedures for opening a new account, closing an account, and a function to get the total balance of a customer across all accounts.

**Answer:**

DELIMITER //

CREATE PROCEDURE OpenAccount(

IN p\_AccountID INT,

IN p\_CustomerID INT,

IN p\_AccountType VARCHAR(20),

IN p\_Balance DECIMAL(10,2)

)

BEGIN

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)

VALUES (p\_AccountID, p\_CustomerID, p\_AccountType, p\_Balance, CURRENT\_DATE());

END//

DELIMITER ;

DELIMITER //

CREATE PROCEDURE CloseAccount(

IN p\_AccountID INT

)

BEGIN

DELETE FROM Accounts WHERE AccountID = p\_AccountID;

END//

DELIMITER ;

DELIMITER //

CREATE FUNCTION GetTotalBalance(

p\_CustomerID INT

)

RETURNS DECIMAL(10,2)

BEGIN

DECLARE totalBalance DECIMAL(10,2);

SELECT SUM(Balance) INTO totalBalance

FROM Accounts

WHERE CustomerID = p\_CustomerID;

RETURN totalBalance;

END//

DELIMITER ;