**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 PACKAGE CustomerManagement

AS

PROCEDURE AddCustomer(p\_CustomerID INT, p\_Name VARCHAR(100), p\_DOB DATE, p\_Balance DECIMAL(10,2));

PROCEDURE UpdateCustomerDetails(p\_CustomerID INT, p\_Name VARCHAR(100), p\_DOB DATE);

FUNCTION GetCustomerBalance(p\_CustomerID INT) RETURNS DECIMAL(10,2);

END CustomerManagement;

CREATE PACKAGE BODY CustomerManagement

AS

PROCEDURE AddCustomer(p\_CustomerID INT, p\_Name VARCHAR(100), p\_DOB DATE, 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;

PROCEDURE UpdateCustomerDetails(p\_CustomerID INT, p\_Name VARCHAR(100), p\_DOB DATE)

BEGIN

UPDATE Customers

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

WHERE CustomerID = p\_CustomerID;

END;

FUNCTION GetCustomerBalance(p\_CustomerID INT) RETURNS DECIMAL(10,2)

BEGIN

RETURN (SELECT Balance FROM Customers WHERE CustomerID = p\_CustomerID);

END;

END CustomerManagement //

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 PACKAGE EmployeeManagement

AS

PROCEDURE HireEmployee(p\_EmployeeID INT, p\_Name VARCHAR(100), p\_Position VARCHAR(50), p\_Salary DECIMAL(10,2), p\_Department VARCHAR(50), p\_HireDate DATE);

PROCEDURE UpdateEmployeeDetails(p\_EmployeeID INT, p\_Name VARCHAR(100), p\_Position VARCHAR(50), p\_Salary DECIMAL(10,2), p\_Department VARCHAR(50));

FUNCTION CalculateAnnualSalary(p\_EmployeeID INT) RETURNS DECIMAL(10,2);

END EmployeeManagement;

CREATE PACKAGE BODY EmployeeManagement

AS

PROCEDURE HireEmployee(p\_EmployeeID INT, p\_Name VARCHAR(100), p\_Position VARCHAR(50), p\_Salary DECIMAL(10,2), p\_Department VARCHAR(50), 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;

PROCEDURE UpdateEmployeeDetails(p\_EmployeeID INT, p\_Name VARCHAR(100), p\_Position VARCHAR(50), p\_Salary DECIMAL(10,2), 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;

FUNCTION CalculateAnnualSalary(p\_EmployeeID INT) RETURNS DECIMAL(10,2)

BEGIN

RETURN (SELECT Salary \* 12 FROM Employees WHERE EmployeeID = p\_EmployeeID);

END;

END EmployeeManagement //

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 PACKAGE AccountOperations

AS

PROCEDURE OpenAccount(p\_AccountID INT, p\_CustomerID INT, p\_AccountType VARCHAR(20), p\_Balance DECIMAL(10,2));

PROCEDURE CloseAccount(p\_AccountID INT);

FUNCTION GetTotalBalance(p\_CustomerID INT) RETURNS DECIMAL(10,2);

END AccountOperations;

CREATE PACKAGE BODY AccountOperations

AS

PROCEDURE OpenAccount(p\_AccountID INT, p\_CustomerID INT, p\_AccountType VARCHAR(20), 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;

PROCEDURE CloseAccount(p\_AccountID INT)

BEGIN

DELETE FROM Accounts WHERE AccountID = p\_AccountID;

END;

FUNCTION GetTotalBalance(p\_CustomerID INT) RETURNS DECIMAL(10,2)

BEGIN

RETURN (SELECT SUM(Balance) FROM Accounts WHERE CustomerID = p\_CustomerID);

END;

END AccountOperations //

DELIMITER ;