**Exercise 7: Packages**

**ANS1.**

CREATE OR REPLACE PACKAGE CustomerManagement IS

PROCEDURE ADDCUST(

CID IN CUSTOMERS.CUSTOMERID%TYPE,

NAME IN CUSTOMERS.NAME%TYPE,

DOB IN CUSTOMERS.DOB%TYPE,

BALANCE IN CUSTOMERS.BALANCE%TYPE,

LASTMODIFIED IN CUSTOMERS.LASTMODIFIED%TYPE,

ISVIP IN CUSTOMERS.ISVIP%TYPE);

PROCEDURE UPDATECUST(CID IN CUSTOMERS.CUSTOMERID%TYPE,

CNAME IN CUSTOMERS.NAME%TYPE);

FUNCTION GETBALANCE(CID IN CUSTOMERS.CUSTOMERID%TYPE) RETURN CUSTOMERS.BALANCE%TYPE;

END CustomerManagement;

CREATE OR REPLACE PACKAGE BODY CustomerManagement IS

PROCEDURE ADDCUST(

CID IN CUSTOMERS.CUSTOMERID%TYPE,

NAME IN CUSTOMERS.NAME%TYPE,

DOB IN CUSTOMERS.DOB%TYPE,

BALANCE IN CUSTOMERS.BALANCE%TYPE,

LASTMODIFIED IN CUSTOMERS.LASTMODIFIED%TYPE,

ISVIP IN CUSTOMERS.ISVIP%TYPE) IS

BEGIN

INSERT INTO CUSTOMERS VALUES(CID,NAME,DOB,BALANCE,LASTMODIFIED,ISVIP);

END ADDCUST;

PROCEDURE UPDATECUST(CID IN CUSTOMERS.CUSTOMERID%TYPE,

CNAME IN CUSTOMERS.NAME%TYPE) IS

BEGIN

UPDATE CUSTOMERS

SET NAME=CNAME WHERE

CUSTOMERID=CID;

END UPDATECUST;

FUNCTION GETBALANCE(CID IN CUSTOMERS.CUSTOMERID%TYPE) RETURN CUSTOMERS.BALANCE%TYPE

IS BAL CUSTOMERS.BALANCE%TYPE;

BEGIN

SELECT BALANCE INTO BAL

FROM CUSTOMERS WHERE

CUSTOMERID=CID;

RETURN BAL;

END GETBALANCE;

END CustomerManagement;

DECLARE

Z NUMBER;

BEGIN

CustomerManagement.ADDCUST(30,'AUZI','17-FEB-1995',66000,'03-AUG-2024','T');

CustomerManagement.UPDATECUST(30,'RAUZI');

Z:=CustomerManagement.GETBALANCE(30);

END;

/

**ANS2:**

CREATE OR REPLACE PACKAGE EmployeeManagement IS

PROCEDURE ADDEMP(

EID IN EMPLOYEES.EMPLOYEEID%TYPE,

NAME IN EMPLOYEES.NAME%TYPE,

POSITION IN EMPLOYEES.POSITION%TYPE,

SALARY IN EMPLOYEES.SALARY%TYPE,

DEPARTMENT IN EMPLOYEES.DEPARTMENT%TYPE,

HIREDATE IN EMPLOYEES.HIREDATE%TYPE);

PROCEDURE UPDATEEMP(EID IN EMPLOYEES.EMPLOYEEID%TYPE,

ENAME IN EMPLOYEES.NAME%TYPE);

FUNCTION GETANNUALSAL(EID IN EMPLOYEES.EMPLOYEEID%TYPE) RETURN EMPLOYEES.SALARY%TYPE;

END EmployeeManagement;

CREATE OR REPLACE PACKAGE BODY EmployeeManagement IS

PROCEDURE ADDEMP(

EID IN EMPLOYEES.EMPLOYEEID%TYPE,

NAME IN EMPLOYEES.NAME%TYPE,

POSITION IN EMPLOYEES.POSITION%TYPE,

SALARY IN EMPLOYEES.SALARY%TYPE,

DEPARTMENT IN EMPLOYEES.DEPARTMENT%TYPE,

HIREDATE IN EMPLOYEES.HIREDATE%TYPE) IS

BEGIN

INSERT INTO EMPLOYEES VALUES(EID,NAME,POSITION,SALARY,DEPARTMENT,HIREDATE);

END ADDEMP;

PROCEDURE UPDATEEMP(EID IN EMPLOYEES.EMPLOYEEID%TYPE,

ENAME IN EMPLOYEES.NAME%TYPE) IS

BEGIN

UPDATE EMPLOYEES

SET NAME=ENAME WHERE

EMPLOYEEID=EID;

END UPDATEEMP;

FUNCTION GETANNUALSAL(EID IN EMPLOYEES.EMPLOYEEID%TYPE) RETURN EMPLOYEES.SALARY%TYPE

IS SAL EMPLOYEES.SALARY%TYPE;

BEGIN

SELECT SALARY INTO SAL

FROM EMPLOYEES WHERE

EMPLOYEEID=EID;

SAL:=SAL\*12;

RETURN SAL;

END GETANNUALSAL;

END EmployeeManagement;

**ANS3:**

CREATE OR REPLACE PACKAGE AccountOperations AS

PROCEDURE OpenAccount(

p\_CustomerID IN Accounts.CustomerID%TYPE,

P\_AccountType IN Accounts.AccountType%TYPE,

p\_InitialBalance IN Accounts.Balance%TYPE

);

PROCEDURE CloseAccount(

P\_AccountID IN Accounts.AccountID%TYPE

);

FUNCTION GetTotalBalance(

p\_CustomerID IN Accounts.CustomerID%TYPE

) RETURN Accounts.Balance%TYPE;

END AccountOperations;

/

CREATE OR REPLACE PACKAGE BODY AccountOperations AS

PROCEDURE OpenAccount(

p\_CustomerID IN Accounts.CustomerID%TYPE,

p\_AccountType IN Accounts.AccountType%TYPE,

p\_InitialBalance IN Accounts.Balance%TYPE

) IS

BEGIN

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

VALUES (Accounts\_SEQ.NEXTVAL, p\_CustomerID, p\_AccountType, p\_InitialBalance, SYSDATE);

EXCEPTION

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'ERROR WHILE OPENING NEW ACCOUNT: ' || SQLERRM);

END OpenAccount;

PROCEDURE CloseAccount(

p\_AccountID IN Accounts.AccountID%TYPE

) IS

BEGIN

DELETE FROM Accounts WHERE AccountID = AccountID;

EXCEPTION

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'ERROR WHILE CLOSING ACCOUNT: ' || SQLERRM);

END CloseAccount;

FUNCTION GetTotalBalance(

P\_CustomerIDIN Accounts.CustomerID%TYPE

) RETURN Accounts.Balance%TYPE IS

v\_TotalBalance Accounts.Balance%TYPE := 0;

BEGIN

SELECT SUM(Balance) INTO v\_TotalBalance

FROM Accounts

WHERE CustomerID= p\_CustomerID;

RETURN v\_TotalBalance;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 0;

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Error while getting total balance: ' || SQLERRM);

END GetTotalBalance;

END AccountOperations;

/