**Exercise 2: Error Handling**

**-- Scenario 1**

CREATE OR REPLACE PROCEDURE SafeTransferFunds(p\_FromAcc NUMBER, p\_ToAcc NUMBER, p\_Amount NUMBER) IS

v\_FromBalance NUMBER;

BEGIN

SELECT Balance INTO v\_FromBalance FROM Accounts WHERE AccountID = p\_FromAcc FOR UPDATE;

IF v\_FromBalance < p\_Amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds');

END IF;

UPDATE Accounts SET Balance = Balance - p\_Amount WHERE AccountID = p\_FromAcc;

UPDATE Accounts SET Balance = Balance + p\_Amount WHERE AccountID = p\_ToAcc;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (Transactions\_SEQ.NEXTVAL, p\_FromAcc, SYSDATE, p\_Amount, 'Failed');

END;

/

**-- Scenario 2**

CREATE OR REPLACE PROCEDURE UpdateSalary(p\_EmployeeID NUMBER, p\_Percent NUMBER) IS

v\_Salary Employees.Salary%TYPE;

BEGIN

SELECT Salary INTO v\_Salary FROM Employees WHERE EmployeeID = p\_EmployeeID;

UPDATE Employees SET Salary = Salary + (Salary \* p\_Percent / 100) WHERE EmployeeID = p\_EmployeeID;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (Transactions\_SEQ.NEXTVAL, NULL, SYSDATE, NULL, 'Invalid EmployeeID');

END;

/

**-- Scenario 3**

CREATE OR REPLACE PROCEDURE AddNewCustomer(p\_CustomerID NUMBER, p\_Name VARCHAR2, p\_DOB DATE, p\_Balance NUMBER) IS

BEGIN

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

VALUES (p\_CustomerID, p\_Name, p\_DOB, p\_Balance, SYSDATE);

COMMIT;

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (Transactions\_SEQ.NEXTVAL, NULL, SYSDATE, NULL, 'Duplicate Customer');

END;

/