**Exercise 2: Error Handling**

**Scenario 1: Handle exceptions during fund transfers between accounts**

Query:

CREATE OR REPLACE PROCEDURE SafeTransferFunds(p\_from\_account\_id NUMBER, p\_to\_account\_id NUMBER, p\_amount NUMBER)

IS

v\_from\_balance NUMBER;

v\_to\_balance NUMBER;

BEGIN

SELECT Balance

INTO v\_from\_balance

FROM Accounts

WHERE AccountID = p\_from\_account\_id;

IF v\_from\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in the source account');

END IF;

BEGIN

UPDATE Accounts

SET Balance = Balance - p\_amount

WHERE AccountID = p\_from\_account\_id;

UPDATE Accounts

SET Balance = Balance + p\_amount

WHERE AccountID = p\_to\_account\_id;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error transferring funds: ' || SQLERRM);

END;

END SafeTransferFunds;

**Scenario 2: Manage errors when updating employee salaries**

Query:

CREATE OR REPLACE PROCEDURE UpdateSalary(p\_employee\_id NUMBER, p\_percentage NUMBER)

IS

v\_employee\_id NUMBER;

v\_salary NUMBER;

BEGIN

SELECT EmployeeID, Salary

INTO v\_employee\_id, v\_salary

FROM Employees

WHERE EmployeeID = p\_employee\_id;

UPDATE Employees

SET Salary = v\_salary \* (1 + p\_percentage / 100)

WHERE EmployeeID = v\_employee\_id;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Employee ID ' || p\_employee\_id || ' does not exist');

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error updating salary: ' || SQLERRM);

END UpdateSalary;

**Scenario 3: Ensure data integrity when adding a new customer**

Query:

CREATE OR REPLACE PROCEDURE AddNewCustomer(p\_customer\_id NUMBER, p\_name VARCHAR2, p\_dob DATE, p\_initial\_balance NUMBER)

IS

v\_customer\_id NUMBER;

BEGIN

BEGIN

SELECT CustomerID

INTO v\_customer\_id

FROM Customers

WHERE CustomerID = p\_customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Error: Customer ID ' || p\_customer\_id || ' already exists');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

BEGIN

INSERT INTO Customers (CustomerID, Name, DOB, InitialBalance, CurrentBalance, LastModified)

VALUES (p\_customer\_id, p\_name, p\_dob, p\_initial\_balance, p\_initial\_balance, SYSDATE);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || p\_customer\_id || ' added successfully');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error adding customer: ' || SQLERRM);

END;

END;

END AddNewCustomer;