**Exercise 3: Error Handling:**

BEGIN

    EXECUTE IMMEDIATE 'DROP TABLE customers';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

BEGIN

    EXECUTE IMMEDIATE 'DROP TABLE employees';

EXCEPTION WHEN OTHERS THEN NULL;

END;

/

CREATE TABLE customerss (

    customer\_id NUMBER PRIMARY KEY,

    name VARCHAR2(100),

    balance NUMBER

);

CREATE TABLE employees1 (

    employee\_id NUMBER PRIMARY KEY,

    name VARCHAR2(100),

    salary NUMBER

);

CREATE OR REPLACE PROCEDURE addnewcustomer (

    p\_id       NUMBER,

    p\_name     VARCHAR2,

    p\_balance  NUMBER

)

AS

BEGIN

    INSERT INTO customerss (customer\_id, name, balance)

    VALUES (p\_id, p\_name, p\_balance);

    DBMS\_OUTPUT.PUT\_LINE('Customer added successfully.');

EXCEPTION

    WHEN OTHERS THEN

        DBMS\_OUTPUT.PUT\_LINE('Error in ADDNEWCUSTOMER: ' || SQLERRM);

END;

/

CREATE OR REPLACE PROCEDURE updatesalary (

    p\_empid     NUMBER,

    p\_increment NUMBER

)

AS

BEGIN

    UPDATE employees1

    SET salary = salary + p\_increment

    WHERE employee\_id = p\_empid;

    IF SQL%ROWCOUNT = 0 THEN

        DBMS\_OUTPUT.PUT\_LINE('No employee found.');

    ELSE

        DBMS\_OUTPUT.PUT\_LINE('Salary updated.');

    END IF;

EXCEPTION

    WHEN OTHERS THEN

        DBMS\_OUTPUT.PUT\_LINE('Error in UPDATESALARY: ' || SQLERRM);

END;

/

CREATE OR REPLACE PROCEDURE safetransferfunds (

    p\_from\_id NUMBER,

    p\_to\_id   NUMBER,

    p\_amount  NUMBER

)

AS

BEGIN

    -- Check if both customers exist and have enough balance

    DECLARE

        v\_from\_balance NUMBER;

    BEGIN

        SELECT balance INTO v\_from\_balance

        FROM customerss

        WHERE customer\_id = p\_from\_id;

        IF v\_from\_balance < p\_amount THEN

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

        END IF;

        UPDATE customerss SET balance = balance - p\_amount WHERE customer\_id = p\_from\_id;

        UPDATE customerss SET balance = balance + p\_amount WHERE customer\_id = p\_to\_id;

        DBMS\_OUTPUT.PUT\_LINE('Funds transferred successfully.');

    END;

EXCEPTION

    WHEN OTHERS THEN

        DBMS\_OUTPUT.PUT\_LINE('Error in SAFETRANSFERFUNDS: ' || SQLERRM);

END;

/

INSERT INTO employees1 VALUES (2001, 'Alice', 50000);

INSERT INTO employees1 VALUES (2002, 'Bob', 45000);

COMMIT;

BEGIN

    addnewcustomer(101, 'John Doe', 1000);

    addnewcustomer(102, 'Jane Smith', 500);

    updatesalary(2001, 5000);

    safetransferfunds(101, 102, 300);

END;

/

