**PL/SQL programming(3.3)**

SET SERVEROUTPUT ON;BEGIN    EXECUTE IMMEDIATE 'DROP TABLE BANK\_ACCOUNTS';EXCEPTION WHEN OTHERS THEN NULL;END;/-- Create BANK\_ACCOUNTS tableCREATE TABLE BANK\_ACCOUNTS (    AccountID NUMBER PRIMARY KEY,    CustomerID NUMBER,    Balance NUMBER);-- Insert sample dataINSERT INTO BANK\_ACCOUNTS VALUES (1, 101, 10000);INSERT INTO BANK\_ACCOUNTS VALUES (2, 102, 3000);COMMIT;-- Create stored procedure to transfer fundsCREATE OR REPLACE PROCEDURE TransferFunds (    p\_FromAccount IN NUMBER,    p\_ToAccount IN NUMBER,    p\_Amount IN NUMBER) AS    v\_FromBalance NUMBER;BEGIN    -- Get balance of source account    SELECT Balance INTO v\_FromBalance    FROM BANK\_ACCOUNTS    WHERE AccountID = p\_FromAccount;    -- Check sufficient funds    IF v\_FromBalance < p\_Amount THEN        RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account.');    END IF;    -- Perform transfer    UPDATE BANK\_ACCOUNTS    SET Balance = Balance - p\_Amount    WHERE AccountID = p\_FromAccount;    UPDATE BANK\_ACCOUNTS    SET Balance = Balance + p\_Amount    WHERE AccountID = p\_ToAccount;    DBMS\_OUTPUT.PUT\_LINE('Transfer of ' || TO\_CHAR(p\_Amount, 'FM9999990.00') ||                          ' from Account ' || p\_FromAccount || ' to Account ' || p\_ToAccount || ' successful.');    -- Display updated balances    DBMS\_OUTPUT.PUT\_LINE('Updated balances:');    FOR rec IN (SELECT AccountID, CustomerID, Balance FROM BANK\_ACCOUNTS ORDER BY AccountID) LOOP        DBMS\_OUTPUT.PUT\_LINE('AccountID: ' || rec.AccountID ||                              ', CustomerID: ' || rec.CustomerID ||                              ', Balance: ' || TO\_CHAR(rec.Balance, 'FM9999990.00'));    END LOOP;END;/-- Run the procedureBEGIN    TransferFunds(1, 2, 2000);END;/