SCM22CD053:
EXP 9

1.

```
SQL> CREATE or REPLACE FUNCTION fact(n in integer)
  2  return integer
  3  is
  4  f number(32):=1;
  5  BEGIN
  6   for i in 1..n loop
  7    f:=f*i;
  8   end loop;
  9  return f;
 10  END;
 11  /

Function created.


SQL>  declare
  2    f2 number(32);
  3   begin
  4    f2:=fact(&f1);
  5    dbms_output.put_line('Factorial is: '||f2);
  6   end;
  7   /
Enter value for f1: 5
old   4:   f2:=fact(&f1);
new   4:   f2:=fact(5);
Factorial is: 120

PL/SQL procedure successfully completed.
```

2.

```
SQL> select * from stockco52;

    ITEMNO INAME                                                     QTY
---------- -------------------------------------------------- ----------
       100 pencil                                                     20
       101 scale                                                      23
       102 pen                                                        21
       103 eraser                                                     12
       104 sharpner                                                   22

SQL> create or replace function checkstock(productno in number)
  2  return number
  3  is
  4  Q number(4);
  5  begin
  6  select qty into Q from stockco52 where itemno=productno;
  7  return Q;
  8  end;
  9  /

Function created.

SQL> declare
  2  INO stockco52.itemno%type:=&INO;
  3  QTY stockco52.qty%type;
  4  begin
  5  qty:=checkstock(ino);
  6  dbms_output.put_line(ino||' '||qty);
  7  end;
  8  /
Enter value for ino: 103
old   2: INO stockco52.itemno%type:=&INO;
new   2: INO stockco52.itemno%type:=103;
103 12

PL/SQL procedure successfully completed.
SQL> SELECT * FROM STOCKCO52;

    ITEMNO INAME                                                     QTY
---------- -------------------------------------------------- ----------
       100 pencil                                                     20
       101 scale                                                      23
       102 pen                                                        21
       103 eraser                                                     12
       104 sharpner                                                   22
```

3.

```
SQL> CREATE OR REPLACE FUNCTION ITEMCHECK(P_ITEMNO IN NUMBER)
  2     RETURN BOOLEAN
  3  IS
  4     ITEM_NOT_FOUND EXCEPTION;
  5     ITEM_COUNT NUMBER;
  6  BEGIN
  7     SELECT COUNT(*) INTO ITEM_COUNT FROM STOCKCO52 WHERE ITEMNO = P_ITEMNO;
  8     IF ITEM_COUNT>0 THEN
  9        RETURN TRUE;
 10      ELSE
 11         RAISE ITEM_NOT_FOUND;
 12      END IF;
 13  EXCEPTION
 14      WHEN ITEM_NOT_FOUND THEN
 15         DBMS_OUTPUT.PUT_LINE('Item number ' || P_ITEMNO || ' does not exist in STOCKCO52.');
 16         RETURN FALSE;
 17      WHEN OTHERS THEN
 18         DBMS_OUTPUT.PUT_LINE('An unexpected error occurred: ' || SQLERRM);
 19         RETURN FALSE;
 20  END ITEMCHECK;
 21  /

Function created.

SQL> DECLARE
  2     ITEM_EXISTS BOOLEAN;
  3     INO STOCKCO52.ITEMNO%TYPE:=&INO;
  4  BEGIN
  5     ITEM_EXISTS := ITEMCHECK(INO);
  6     IF ITEM_EXISTS THEN
  7        DBMS_OUTPUT.PUT_LINE('Item exists in stock.');
  8     ELSE
  9        DBMS_OUTPUT.PUT_LINE('Item does not exist in stock.');
 10      END IF;
 11  END;
 12  /
Enter value for ino: 101
old   3:     INO STOCKCO52.ITEMNO%TYPE:=&INO;
new   3:     INO STOCKCO52.ITEMNO%TYPE:=101;
Item exists in stock.

PL/SQL procedure successfully completed.

SQL> /
Enter value for ino: 105
old   3:     INO STOCKCO52.ITEMNO%TYPE:=&INO;
new   3:     INO STOCKCO52.ITEMNO%TYPE:=105;
Item number 105 does not exist in STOCKCO52.
Item does not exist in stock.

PL/SQL procedure successfully completed.
```

4.

```
SQL> SELECT * FROM EMPCO52;

       ENO NAME                     SALARY
---------- -------------------- ----------
       100 ANIL                      25000
       101 JOS                       23000
       102 RESHMA                    10000
       103 MANOJ                      5000

SQL> CREATE OR REPLACE PROCEDURE DISPEMP(EMP_NO IN NUMBER)
  2  IS
  3     N EMPCO52.NAME%TYPE;
  4     S EMPCO52.SALARY%TYPE;
  5  BEGIN
  6     SELECT NAME,SALARY INTO N,S FROM EMPCO52 WHERE ENO=EMP_NO;
  7     DBMS_OUTPUT.PUT_LINE('NAME : ' || N);
  8     DBMS_OUTPUT.PUT_LINE('SALARY : ' || S);
  9  EXCEPTION
 10      WHEN NO_DATA_FOUND THEN
 11          DBMS_OUTPUT.PUT_LINE('NO SUCH EMPLOYEE');
 12  END;
 13  /

Procedure created.

SQL> DECLARE
  2     EN EMPCO52.ENO%TYPE:=&EMP_NO;
  3  BEGIN
  4     DISPEMP(EN);
  5  END;
  6  /
Enter value for emp_no: 102
old   2:     EN EMPCO52.ENO%TYPE:=&EMP_NO;
new   2:     EN EMPCO52.ENO%TYPE:=102;
NAME : RESHMA
SALARY : 10000

PL/SQL procedure successfully completed.

SQL> SELECT * FROM EMPCO52;

       ENO NAME                     SALARY
---------- -------------------- ----------
       100 ANIL                      25000
       101 JOS                       23000
       102 RESHMA                    10000
       103 MANOJ                      5000
```

5.

```
SQL> CREATE OR REPLACE PROCEDURE INCREASE_SALARY(EMP_NO IN NUMBER, R
IN NUMBER)
  2  IS
  3      V_UPDATED_SALARY EMPCO52.SALARY%TYPE;
  4  BEGIN
  5      UPDATE EMPCO52 SET SALARY=SALARY+(SALARY*R/100) WHERE
ENO=EMP_NO;
  6      IF SQL%ROWCOUNT>0 THEN
  7          SELECT SALARY INTO V_UPDATED_SALARY FROM EMPCO52 WHERE
ENO=EMP_NO;
  8          DBMS_OUTPUT.PUT_LINE('Updated Salary for Employee No ' || EMP_NO || ' : ' ||
V_UPDATED_
SALARY);
  9      ELSE
 10          DBMS_OUTPUT.PUT_LINE('No employee found with Employee No ' || EMP_NO);
 11      END IF;
 12  EXCEPTION
 13      WHEN OTHERS THEN
 14          DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
 15  END;
 16  /

Procedure created.

SQL> DECLARE
  2      EMP_NO EMPCO52.ENO%TYPE := &EMP_NO;
  3      R NUMBER := &PERCENTAGE;
  4  BEGIN
  5      INCREASE_SALARY(EMP_NO, R);
  6  END;
  7  /
Enter value for emp_no: 103
old   2:     EMP_NO EMPCO52.ENO%TYPE := &EMP_NO;
new   2:     EMP_NO EMPCO52.ENO%TYPE := 103;
Enter value for percentage: 3
old   3:     R NUMBER := &PERCENTAGE;
new   3:     R NUMBER := 3;
Updated Salary for Employee No 103 : 5150

PL/SQL procedure successfully completed.

SQL> SELECT * FROM EMPCO52;

       ENO NAME                     SALARY
---------- -------------------- ----------
       100 ANIL                      25000
       101 JOS                       23000
       102 RESHMA                    10000
       103 MANOJ                      5150
```

6.

```
SQL> SELECT * FROM ACCOUNTCO52;

    ACCNO NAME                 BALANCE
---------- -------------------- ----------
      100 JOHN                 1000
      101 JAMES                 2000
      102 SMITH                 3000

SQL> CREATE OR REPLACE PROCEDURE UPDATE_ACCOUNT(P_ACCNO IN
NUMBER,P_AMOUNT IN NUMBER,P_OPERATION IN CHAR)
  2  IS
  3      V_CURRENT_BALANCE ACCOUNTCO52.BALANCE%TYPE;
  4  BEGIN
  5      SELECT BALANCE INTO V_CURRENT_BALANCE FROM ACCOUNTCO52 WHERE
ACCNO=P_ACCNO;
  6      IF P_OPERATION='D' THEN
  7          UPDATE ACCOUNTCO52 SET BALANCE=V_CURRENT_BALANCE+P_AMOUNT
WHERE ACCNO=P_ACCNO;
  8          DBMS_OUTPUT.PUT_LINE('Deposited ' || P_AMOUNT || ' into account ' || P_ACCNO);
  9      ELSIF P_OPERATION='W' THEN
 10          IF V_CURRENT_BALANCE<P_AMOUNT THEN
 11              DBMS_OUTPUT.PUT_LINE('Insufficient funds for withdrawal from account ' ||
P_ACCNO);
 12              RETURN;
 13          ELSE
 14              UPDATE ACCOUNTCO52 SET BALANCE=V_CURRENT_BALANCE-P_AMOUNT
WHERE ACCNO=P_ACCNO;
 15              DBMS_OUTPUT.PUT_LINE('Withdrew ' || P_AMOUNT || ' from account ' ||
P_ACCNO);
 16          END IF;
 17      ELSE
 18          DBMS_OUTPUT.PUT_LINE('Invalid operation. Use "D" for deposit or "W" for
withdraw.');
 19          RETURN;
 20      END IF;
 21      DBMS_OUTPUT.PUT_LINE('Updated balance for account ' || P_ACCNO || ': ' ||
 22          (V_CURRENT_BALANCE+CASE WHEN P_OPERATION='D' THEN P_AMOUNT
ELSE-P_AMOUNT END));
 23  EXCEPTION
 24      WHEN NO_DATA_FOUND THEN
 25          DBMS_OUTPUT.PUT_LINE('Account not found: ' || P_ACCNO);
 26      WHEN OTHERS THEN
 27          DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
 28  END;
 29  /

Procedure created.

SQL> DECLARE
  2      V_ACCNO NUMBER;
  3      V_AMOUNT NUMBER;
  4      V_OPERATION CHAR;
  5  BEGIN
  6      V_ACCNO := &ACCNO;
  7      V_OPERATION := '&OPERATION';
  8      V_AMOUNT := &AMOUNT;
  9      UPDATE_ACCOUNT(V_ACCNO, V_AMOUNT, V_OPERATION);
 10  END;
 11  /
Enter value for accno: 101
old   6:     V_ACCNO := &ACCNO;
new   6:     V_ACCNO := 101;
Enter value for operation: W
old   7:     V_OPERATION := '&OPERATION';
new   7:     V_OPERATION := 'W';
Enter value for amount: 200
old   8:     V_AMOUNT := &AMOUNT;
new   8:     V_AMOUNT := 200;
Withdrew 200 from account 101
Updated balance for account 101: 1800

PL/SQL procedure successfully completed.

SQL> SELECT * FROM ACCOUNTCO52;

     ACCNO NAME                    BALANCE
---------- -------------------- ----------
       100 JOHN                       1000
       101 JAMES                      1800
       102 SMITH                      3000

SQL> /
Enter value for accno: 101
old   6:     V_ACCNO := &ACCNO;
new   6:     V_ACCNO := 101;
Enter value for operation: D
old   7:     V_OPERATION := '&OPERATION';
new   7:     V_OPERATION := 'D';
Enter value for amount: 600
old   8:     V_AMOUNT := &AMOUNT;
new   8:     V_AMOUNT := 600;
Deposited 600 into account 101
Updated balance for account 101: 2400

PL/SQL procedure successfully completed.

SQL> SELECT * FROM ACCOUNTCO52;

     ACCNO NAME                    BALANCE
---------- -------------------- ----------
       100 JOHN                       1000
```

```
101 JAMES                    2400
102 SMITH                    3000
```

7.

```
SQL> CREATE OR REPLACE PROCEDURE ADD_MULTIPLY(P_NUM1 IN
NUMBER,P_NUM2 IN NUMBER,P_SUM OUT NUMBER,P_PRODUCT IN OUT NUMBER)
  2  IS
  3  BEGIN
  4     P_SUM:=P_NUM1+P_NUM2;
  5     P_PRODUCT:=P_PRODUCT*P_NUM1*P_NUM2;
  6  EXCEPTION
  7     WHEN OTHERS THEN
  8        DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
  9  END;
 10  /

Procedure created.

SQL> DECLARE
  2     V_NUM1 NUMBER;
  3     V_NUM2 NUMBER;
  4     V_SUM NUMBER;
  5     V_PRODUCT NUMBER := 1;
  6  BEGIN
  7     V_NUM1:=&NUM1;
  8     V_NUM2:=&NUM2;
  9     ADD_MULTIPLY(V_NUM1,V_NUM2,V_SUM,V_PRODUCT);
 10      DBMS_OUTPUT.PUT_LINE('Sum: ' || V_SUM);
 11      DBMS_OUTPUT.PUT_LINE('Product: ' || V_PRODUCT);
 12  END;
 13  /
Enter value for num1: 2
old   7:     V_NUM1:=&NUM1;
new   7:     V_NUM1:=2;
Enter value for num2: 4
old   8:     V_NUM2:=&NUM2;
new   8:     V_NUM2:=4;
Sum: 6
Product: 8

PL/SQL procedure successfully completed.
```