6c

```
SQL> CREATE TABLE Employee_23BAI1242 (
  2    Emp_No NUMBER PRIMARY KEY,
  3    E_Name VARCHAR2(100),
  4    Experience NUMBER,
  5    City VARCHAR2(100),
  6    Dno NUMBER
  7  );

Table created.
```

```
SQL> CREATE TABLE Emp_Sal_23BAI1242 (
  2    Emp_No NUMBER PRIMARY KEY,
  3    Basic NUMBER,
  4    Deduction NUMBER,
  5    Gross_Salary NUMBER,
  6    FOREIGN KEY (Emp_No) REFERENCES Employee_23BAI1242(Emp_No
  7  );

Table created.
```

```
SQL> CREATE TABLE Emp_Sal_Deleted_23BAI1242 (
  2    Emp_No NUMBER,
  3    Basic NUMBER,
  4    Deduction NUMBER,
  5    Gross_Salary NUMBER,
  6    Deleted_At TIMESTAMP
  7  );

Table created.
```

```
SQL> CREATE OR REPLACE TRIGGER trg_show_name_change_23BAI1242
  2  AFTER UPDATE OF E_Name
  3  ON Employee_23BAI1242
  4  FOR EACH ROW
  5  BEGIN
  6    DBMS_OUTPUT.PUT_LINE('Old E_Name: ' || :OLD.E_Name);
  7    DBMS_OUTPUT.PUT_LINE('New E_Name: ' || :NEW.E_Name);
  8  END;
  9  /

Trigger created.
```

```
SQL> CREATE OR REPLACE TRIGGER trg_insert_deleted_23BAI1242
  2  AFTER DELETE
  3  ON Emp_Sal_23BAI1242
  4  FOR EACH ROW
  5  BEGIN
  6    INSERT INTO Emp_Sal_Deleted_23BAI1242 (Emp_No, Basic, Ded
  7    VALUES (:OLD.Emp_No, :OLD.Basic, :OLD.Deduction, :OLD.Gro
  8  END;
  9  /

Trigger created.
```

```
SQL> SET SERVEROUTPUT ON;
SQL> INSERT INTO Employee_23BAI1242 (Emp_No, E_Name, Experience,
  2  VALUES (1, 'John Doe', 5, 'New York', 101);

1 row created.
```

```
SQL> UPDATE Employee_23BAI1242
  2  SET E_Name = 'Jane Doe'
  3  WHERE Emp_No = 1;
Old E_Name: John Doe
New E_Name: Jane Doe

1 row updated.
```

```
SQL> INSERT INTO Emp_Sal_23BAI1242 (Emp_No, Basic, Deduction, Gr
  2  VALUES (1, 5000, 500, 4500);

1 row created.
```

```
SQL> DELETE FROM Emp_Sal_23BAI1242
  2  WHERE Emp_No = 1;

1 row deleted.
```

```
SQL> SELECT * FROM Emp_Sal_Deleted_23BAI1242;

    EMP_NO      BASIC  DEDUCTION GROSS_SALARY
---------- ---------- ---------- ------------
DELETED_AT
---------------------------------------------------------------------------
         1       5000        500         4500
10-MAR-25 05.02.31.000000 PM
```

6d

```
SQL*Plus: Release 11.2.0.2.0 Production on Mon Mar 10 17:08:03 2

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

SQL> connect system
Enter password:
Connected.
SQL> CREATE TABLE Student_23BAI1242 (
  2    Register_No NUMBER PRIMARY KEY,
  3    Name VARCHAR2(100),
  4    Marks_Subject1 NUMBER,
  5    Marks_Subject2 NUMBER,
  6    Marks_Subject3 NUMBER,
  7    Marks_Subject4 NUMBER,
  8    Marks_Subject5 NUMBER
  9  );

Table created.
```

```
SQL> INSERT INTO Student_23BAI1242 (Register_No, Name, Marks_Sub
  2  VALUES (1, 'Yash', 45, 50, 60, 55, 70);

1 row created.
```

```
SQL> INSERT INTO Student_23BAI1242 (Register_No, Name, Marks_Sub
  2  VALUES (2, 'Sanskar', 30, 45, 35, 60, 65);

1 row created.
```

```
SQL> INSERT INTO Student_23BAI1242 (Register_No, Name, Marks_Sub
  2  VALUES (3, 'Nitya', 50, 55, 45, 70, 75);

1 row created.
```

```
SQL> INSERT INTO Student_23BAI1242 (Register_No, Name, Marks_Subj
  2  VALUES (4, 'Nikhil', 20, 35, 40, 55, 60);

1 row created.
```

```
SQL> INSERT INTO Student_23BAI1242 (Register_No, Name, Marks_Sub
  2  VALUES (5, 'Soham', 55, 60, 50, 65, 80);

1 row created.
```

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2    CURSOR student_cursor_23BAI1242 IS
  3      SELECT Name, Marks_Subject1, Marks_Subject2, Marks_Subj
  4      FROM Student_23BAI1242
  5      WHERE Marks_Subject1 > 40 AND Marks_Subject2 > 40 AND M
  6
  7    student_record student_cursor_23BAI1242%ROWTYPE;
  8    average_marks NUMBER;
  9  BEGIN
 10    OPEN student_cursor_23BAI1242;
 11
 12    LOOP
 13      FETCH student_cursor_23BAI1242 INTO student_record;
 14      EXIT WHEN student_cursor_23BAI1242%NOTFOUND;
 15
 16      average_marks := (student_record.Marks_Subject1 + stude
 17
 18      DBMS_OUTPUT.PUT_LINE('Name: ' || student_record.Name);
 19      DBMS_OUTPUT.PUT_LINE('Marks: ' || student_record.Marks_
udent_record.Marks_Subject5);
 20      DBMS_OUTPUT.PUT_LINE('Average Marks: ' || average_marks
 21      DBMS_OUTPUT.PUT_LINE('----------------------------');
 22    END LOOP;
 23
 24    CLOSE student_cursor_23BAI1242;
 25  END;
 26  /
```

```
 24    CLOSE student_cursor_23BAI1242;
 25  END;
 26  /
Name: Yash
Marks: 45, 50, 60, 55, 70
Average Marks: 56
----------------------------
Name: Nitya
Marks: 50, 55, 45, 70, 75
Average Marks: 59
----------------------------
Name: Soham
Marks: 55, 60, 50, 65, 80
Average Marks: 62
----------------------------

PL/SQL procedure successfully completed.

SQL>
```

6b

```
SQL> CREATE TABLE Student_Course_BAI1242 (
  2    Roll_No NUMBER PRIMARY KEY,
  3    Course VARCHAR2(100),
  4    Course_Code VARCHAR2(20),
  5    Sem VARCHAR2(10),
  6    Total_Marks NUMBER,
  7    Percentage NUMBER
  8  );

Table created.

SQL>
SQL> CREATE TABLE Employee_BAI1242 (
  2    Emp_No NUMBER PRIMARY KEY,
  3    E_Name VARCHAR2(100),
  4    Experience NUMBER,
  5    City VARCHAR2(100),
  6    Dno NUMBER
  7  );

Table created.

SQL>
SQL> CREATE TABLE Emp_Sal_BAI1242 (
  2    Emp_No NUMBER PRIMARY KEY,
  3    Basic NUMBER,
  4    Deduction NUMBER,
  5    Gross_Salary NUMBER,
  6    FOREIGN KEY (Emp_No) REFERENCES Employee_BAI1242(Emp_No)
  7  );

Table created.

SQL>
SQL> CREATE TABLE Department_BAI1242 (
  2    Dname VARCHAR2(100),
  3    Dnumber NUMBER PRIMARY KEY,
  4    MGRSSN NUMBER
  5  );

Table created.
```

```
SQL> CREATE OR REPLACE PROCEDURE Get_Percentage_Ranges_BAI1242 (
  2    v_count_100_70 NUMBER;
  3    v_count_69_60 NUMBER;
  4    v_count_59_50 NUMBER;
  5    v_count_below_49 NUMBER;
  6  BEGIN
  7    SELECT COUNT(*) INTO v_count_100_70
  8    FROM Student_Course_BAI1242
  9    WHERE Course = p_course AND Percentage BETWEEN 70 AND 100
 10
 11    SELECT COUNT(*) INTO v_count_69_60
 12    FROM Student_Course_BAI1242
 13    WHERE Course = p_course AND Percentage BETWEEN 60 AND 69;
 14
 15    SELECT COUNT(*) INTO v_count_59_50
 16    FROM Student_Course_BAI1242
 17    WHERE Course = p_course AND Percentage BETWEEN 50 AND 59;
 18
 19    SELECT COUNT(*) INTO v_count_below_49
 20    FROM Student_Course_BAI1242
 21    WHERE Course = p_course AND Percentage < 50;
 22
 23    DBMS_OUTPUT.PUT_LINE('Percentage Range 70-100%: ' || v_co
 24    DBMS_OUTPUT.PUT_LINE('Percentage Range 60-69%: ' || v_cou
 25    DBMS_OUTPUT.PUT_LINE('Percentage Range 50-59%: ' || v_cou
 26    DBMS_OUTPUT.PUT_LINE('Percentage Below 50%: ' || v_count_
 27  END;
 28  /

Procedure created.
```

```
SQL> CREATE OR REPLACE FUNCTION Factorial_BAI1242 (n IN NUMBER)
  2  BEGIN
  3    IF n = 0 THEN
  4      RETURN 1;
  5    ELSE
  6      RETURN n * Factorial_BAI1242(n - 1);
  7    END IF;
  8  END;
  9  /

Function created.
```

```
SQL> CREATE OR REPLACE PROCEDURE Update_Dno_BAI1242 (p_emp_no IN
  2  BEGIN
  3    UPDATE Employee_BAI1242
  4    SET Dno = p_new_dno
  5    WHERE Emp_No = p_emp_no;
  6
  7    COMMIT;
  8  END;
  9  /

Procedure created.
```

```
SQL> CREATE OR REPLACE FUNCTION Count_Employees_BAI1242 (p_dno I
  2    v_emp_count NUMBER;
  3  BEGIN
  4    SELECT COUNT(*) INTO v_emp_count
  5    FROM Employee_BAI1242
  6    WHERE Dno = p_dno;
  7
  8    RETURN v_emp_count;
  9  END;
 10  /

Function created.
```

```
SQL> SELECT Factorial_BAI1242(5) FROM DUAL;

FACTORIAL_BAI1242(5)
--------------------
                 120
```

```
SQL> SELECT Count_Employees_BAI1242(10) FROM DUAL;

COUNT_EMPLOYEES_BAI1242(10)
---------------------------
                          0
```

```
SQL> INSERT INTO Student_Course_BAI1242 (Roll_No, Course, Course_
  2  VALUES (1, 'Computer Science', 'CS101', '1', 450, 80);

1 row created.

SQL>
SQL> INSERT INTO Student_Course_BAI1242 (Roll_No, Course, Course_
  2  VALUES (2, 'Computer Science', 'CS101', '1', 400, 65);

1 row created.

SQL>
SQL> INSERT INTO Student_Course_BAI1242 (Roll_No, Course, Course_
  2  VALUES (3, 'Computer Science', 'CS101', '1', 375, 55);

1 row created.

SQL>
SQL> INSERT INTO Student_Course_BAI1242 (Roll_No, Course, Course_
  2  VALUES (4, 'Computer Science', 'CS101', '1', 320, 45);

1 row created.

SQL>
SQL> INSERT INTO Student_Course_BAI1242 (Roll_No, Course, Course_
  2  VALUES (5, 'Computer Science', 'CS101', '1', 290, 30);

1 row created.
```

```
SQL> SELECT Count_Employees_BAI1242(10) FROM DUAL;

COUNT_EMPLOYEES_BAI1242(10)
---------------------------
                          2
```

```
SQL> SET SERVEROUTPUT ON;
SQL> EXEC Get_Percentage_Ranges_BAI1242('Computer Science');
Percentage Range 70-100%: 1
Percentage Range 60-69%: 1
Percentage Range 50-59%: 1
Percentage Below 50%: 2

PL/SQL procedure successfully completed.
```

```
SQL> SET SERVEROUTPUT ON;
SQL> EXEC Update_Dno_BAI1242(1, 10);

PL/SQL procedure successfully completed.
```

```
SQL> INSERT INTO Employee_BAI1242 (Emp_No, E_Name, Experience, C
  2  VALUES (1, 'Alice', 5, 'New York', 10);

1 row created.

SQL>
SQL> INSERT INTO Employee_BAI1242 (Emp_No, E_Name, Experience, C
  2  VALUES (2, 'Bob', 3, 'Los Angeles', 10);

1 row created.

SQL>
SQL> INSERT INTO Employee_BAI1242 (Emp_No, E_Name, Experience, C
  2  VALUES (3, 'Charlie', 8, 'Chicago', 20);

1 row created.

SQL>
SQL> INSERT INTO Employee_BAI1242 (Emp_No, E_Name, Experience, C
  2  VALUES (4, 'David', 6, 'Houston', 20);

1 row created.

SQL>
SQL> INSERT INTO Employee_BAI1242 (Emp_No, E_Name, Experience, C
  2  VALUES (5, 'Eve', 2, 'Phoenix', 30);

1 row created.
```

```
SQL> SELECT * FROM Employee_BAI1242 WHERE Emp_No = 1;

    EMP_NO
----------
E_NAME
--------------------------------------------------------------------------------
EXPERIENCE
----------
CITY
--------------------------------------------------------------------------------
       DNO
----------
         1
Alice
         5

    EMP_NO
----------
E_NAME
--------------------------------------------------------------------------------
EXPERIENCE
----------
CITY
--------------------------------------------------------------------------------
       DNO
----------
New York
        10
```

```
SQL> SELECT * FROM Employee_BAI1242;

    EMP_NO
----------
E_NAME
--------------------------------------------------------------------------------
EXPERIENCE
----------
CITY
--------------------------------------------------------------------------------
       DNO
----------
         1
Alice
         5

    EMP_NO
----------
E_NAME
--------------------------------------------------------------------------------
EXPERIENCE
----------
CITY
--------------------------------------------------------------------------------
       DNO
----------
New York
        10


    EMP_NO
----------
E_NAME
--------------------------------------------------------------------------------
EXPERIENCE
----------
CITY
--------------------------------------------------------------------------------
```

```
CITY
--------------------------------------------------------------------------------
       DNO
----------
```

```
         4
David
         6

    EMP_NO
----------
E_NAME
--------------------------------------------------------------------------------
EXPERIENCE
----------
CITY
--------------------------------------------------------------------------------
       DNO
----------
Houston
        20


    EMP_NO
----------
E_NAME
--------------------------------------------------------------------------------
EXPERIENCE
----------
CITY
--------------------------------------------------------------------------------
       DNO
----------
         5
Eve
         2

    EMP_NO
----------
E_NAME
--------------------------------------------------------------------------------
EXPERIENCE
----------
CITY
--------------------------------------------------------------------------------
       DNO
----------
Phoenix
        30
```

```
SQL>
```