Lab Guide for RDBMS Essentials

Education and Research

Infosys®
POWERED BY INTELLECT
DRIVEN BY VALUES

Author(s)	Rengarajan Ramanujam, Shinu Thomas
Authorized by	Srikantan Moorthy
Creation/Revision Date	June 2010
Version	1.4

COPYRIGHT NOTICE

Education and Research Department
Infosys Technologies Limited
Electronic City
Hosur Road
Bangalore - 561 229, India.

Tel: 91 80 852 0261-270
Fax: 91 80 852 0362
www.infy.com
mailto:E&R@infy.com

Document Revision History

Version	Date	Author(s)	Reviewer(s)	Comments
1.0	16-Mar-09	Rengarajan, Shinu Thomas	Kiran RK	Initial Draft
1.1	12-May-09	Rengarajan, Shinu Thomas	Kiran RK	Review comments incorporated after Pilot
1.2	15-Jun-09	Rengarajan, Shinu Thomas	Kiran RK	Review comments incorporated after Pilot
1.3	24-Mar-2010	Shinu Thomas	Rengarajan	IP checking and incorporation
1.4	1-Jun-2010	Shinu Thomas	Rengarajan, Amit Behl, Manjeet Singh Juneja	Conversion of assignments to Devsquare platform

Contents

1 Background

This document contains assignments to be completed as part of the hands on for the subject RDBMS Essentials **(Course code: DB91).**

Note: In order to complete the course, assignments in this document must be completed in the sequence mentioned.

1 Assignments for Day 4 of RDBMS Essentials

1.1 Demo 1: Using Implicit Cursor Attributes.

Objective: To be able to use implicit cursor attributes in PL/SQL blocks.

Problem Description:

Write a PL/SQL program to update the hostelfee by 15% where hotstelid is 'H1' and verify whether any row is affected and if it has affected display how many rows are affected.

```
-- CODE 1.1

BEGIN
  UPDATE hostel SET hostelfee=hostelfee+hostelfee*0.15
  WHERE hostelid='H1';
  IF SQL%NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('No row to update');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Number of rows updated:'||SQL%ROWCOUNT);
  END IF;
END;
/

OUTPUT:
```

```
Number of rows updated:1
```

1.2 Assignment 1: Using Implicit Cursor Attributes.

Objective: To be able to use implicit cursor attributes in PL/SQL blocks.

Platform: Use SQL*PLUS to solve this assignment. Make sure that all the necessary tables are present in your login id.

Problem Description:

1. Modify the codes of Assignment 2 of Day3 and verify how many rows are affected and if any row gets affected, display the number of rows affected.

1.3 Demo 2: Using Explicit Cursors with simple Loop

Objective: To be able to use explicit cursors with simple Loop and its attributes in PL/SQL blocks.

Problem Description:

Write a PL/SQL program to display the details of instructors as shown below given the department id.

Department Name: <departmentname>

Instr ID	Instr Name	Joining Date	Hrs left
<instructorid>	<instructorname>	<dateofjoining>	<remaininghours>

```
-- CODE 1.3

DECLARE
  --Accept departmentid
  v_departmentid department.departmentid%TYPE:=&v_departmentid;
  --
  CURSOR cur_instructor IS
  SELECT instructorid,instructorname,dateofjoining,remaininghours
  FROM instructor WHERE departmentid =v_departmentid;
  v_curvar cur_instructor%ROWTYPE;
  v_departmentname department.departmentname%TYPE;
BEGIN
  SELECT departmentname INTO v_departmentname FROM department
  WHERE departmentid = v_departmentid;
  DBMS_OUTPUT.PUT_LINE('-------------------------------------');
  DBMS_OUTPUT.PUT_LINE('Department Name: '||v_departmentname);
  DBMS_OUTPUT.PUT_LINE('-------------------------------------');
```

```
  OPEN cur_instructor;
  DBMS_OUTPUT.PUT_LINE('-----------------------------------------------
----------------------------');
  DBMS_OUTPUT.PUT_LINE('Instr Id       Instr Name     Joining Date
Hrs left');
 DBMS_OUTPUT.PUT_LINE('-----------------------------------------------
----------------------------');

  LOOP
  FETCH cur_instructor INTO v_curvar;
  EXIT WHEN cur_instructor%NOTFOUND;

  DBMS_OUTPUT.PUT(v_curvar.instructorid||'    ');
  DBMS_OUTPUT.PUT(RPAD(v_curvar.instructorname,20) ||'    ');
  DBMS_OUTPUT.PUT(v_curvar. dateofjoining ||'    ');
  DBMS_OUTPUT.PUT(v_curvar. remaininghours );
  DBMS_OUTPUT.NEW_LINE;
  END LOOP;
  CLOSE cur_instructor;
END;

OUTPUT:
```

```
Enter value for v_departmentid: 10
-------------------------------------
Department Name: Computer Science
-------------------------------------
-----------------------------------------------------------------
Instr Id       Instr Name     Joining Date    Hrs left
-----------------------------------------------------------------
I101   Bob Hockins            12-JAN-00   0
I102   Suguru Zikovich        21-FEB-01   0
I103   Ritivoi                13-JAN-03   0
```

1.4 Assignment 2: Using Explicit Cursors with simple Loop

Objective: To be able to use explicit cursors with simple Loop and its attributes in PL/SQL blocks.

Platform: Use both SQL*PLUS and devsquare platforms to solve the following problems. Make use of the URL shared by the educator or Batch Owner to login to devsquare.

NOTE:

The devsquare platform does not understand the headers and banners. So please display only the data in the devsquare platform.

For e.g. **DO NOT use the headers** like:

```
-----------------------------------------------
Course Id  - Course Name – Semester
-----------------------------------------------
```

Problem Description:

1. Write a PL/SQL program to display the **courseid, coursename and semester** of courses which are elective as shown below given the branch id. At the end display the total number of electives in that branch.

Platform: Use SQL*PLUS to solve this assignment. Headers should be displayed.

Branch Name : <branchname>

Course Id - Course Name – Semester
<courseId> - <coursename>-<semester>
.
.
.
Total electives: [display the total number of records, **Hint:** use ROWCOUNT attribute]

Platform: Use devsquare platform to solve the above problem. Make use of the URL shared by the educator or Batch Owner to login to devsquare. Headers should NOT be displayed.

Use the variable ***v_branchid*** to accept 'B1' as the branchid and ***cur_course*** for the cursor name.

The following should be the format to display the data. The delimiter or column separator for courseid, coursename, semester is hyphen (-)

Branch Name : <branchname>
<courseId> - <coursename>-<semester>
.
.
.

NOTE:

a. Need not have to display the Total number of electives in that branch.
b. Accept 'B1' as input for branchid
c. Cursor SELECT statements MUST contain the required columns ONLY as shown in the display format.

2. Write a PL/SQL program to display the branch name and total number of applicants opted in each branch.

   Platform: Use SQL*PLUS to solve this assignment. Make sure that all the necessary tables are present in your login id.

   Branch Name **No of Applicants Opted**

   <branchname> <derived column>

   Platform: Use devsquare platform to solve the above problem. Make use of the URL shared by the educator or Batch Owner to login to devsquare.

   Use ***cur_branch*** as the cursor name.

   The following should be the format to display the data.

   <branchname>-<derived column>

   .
   .
   .

3. Write a PL/SQL program to display the **studentname, email** and **overallpercentage** of all students present in a particular branch. (Accept the branchid as input)

   Platform: Use SQL*PLUS to solve this assignment. Make sure that all the necessary tables are present in your login id.

   Display the details in the following format.

   Branch Name:<branchname>

   Student Name Email Overall Percentage
   ---- ---------- ---------- -- - ----------
   <studentname> <emailid> <overallpercentage>

   Total number of students : <display the total number of students>

Platform: Use devsquare platform to solve the above problem. Make use of the URL shared by the educator or Batch Owner to login to devsquare.

use the variable ***v_branchid*** to accept 'B1' as the branchid and ***cur_student*** for the cursor name.
use the following format to display the data.

Branch Name : <branchname>
<studentname>-<emailid>-<overallpercentage>
.
.
.

NOTE:
a. Need not have to display the Total number of students in that branch.
b. Accept 'B1' as input for branchid
c. Cursor SELECT statements MUST contain the required columns ONLY as shown in the display format.

1.5 Assignment 3: Using Explicit Cursors with FOR Loop

Objective: To be able to use explicit cursors with FOR Loop and its attributes in PL/SQL blocks.

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

Modify the above cursor assignments (Assignment 2) using FOR Loop.

1.6 Demo 3: Parameterized Cursors

Objective: To be able to use parameterized explicit cursors in PL/SQL blocks.

Problem Description:

Write a PL/SQL program to display the details of all the courses for all the semesters for a given branch as shown in the below format.

Branch Name: <branchname>

Sem: 1

Course Name Elective Duration in Hours

<coursename> <elective> <durationinhours>

…

Sem: 2

Course Name Elective Duration in Hours

<coursename> <elective> <durationinhours>

…

```
-- CODE 1.6

DECLARE
      v_branchid branch.branchid%TYPE:='&v_branchid';
      v_branchname branch.branchname%TYPE;
      CURSOR cur_param(p_sem NUMBER) IS
       SELECT coursename,elective,durationinhours FROM course WHERE
semester=p_sem AND branchid=v_branchid;
  BEGIN
```

```
      SELECT branchname INTO v_branchname FROM branch WHERE
branchid=v_branchid;
      DBMS_OUTPUT.PUT_LINE('-----------------------------------------
--');
      DBMS_OUTPUT.PUT_LINE('Branch Name : '||v_branchname);
      FOR v_sem IN 1..2
      LOOP
            DBMS_OUTPUT.PUT_LINE('-----------------------------------
--------');
            DBMS_OUTPUT.PUT_LINE('Sem: '||v_sem);
            DBMS_OUTPUT.PUT_LINE('-----------------------------------
---------');
            DBMS_OUTPUT.PUT_LINE('Course Name                Elective
Duration');
            DBMS_OUTPUT.PUT_LINE('-----------------------------------
---------');
            FOR i IN cur_param(v_sem)
            LOOP
                  DBMS_OUTPUT.PUT(RPAD(i.coursename,25) ||'   ');
                  DBMS_OUTPUT.PUT(i.elective ||'   ');
                  DBMS_OUTPUT.PUT(i.durationinhours ||'   ');
                  DBMS_OUTPUT.NEW_LINE;
            END LOOP;
            DBMS_OUTPUT.PUT_LINE('-----------------------------------
---------');
      END LOOP;
END;
```

OUTPUT:

```
Enter value for v_branchid: B1
-----------------------------------------
Branch Name : Computer Science
-----------------------------------------
Sem: 1
-----------------------------------------
Course Name               Elective  Duration
-----------------------------------------
Programming Fundamentals    R   40
Data Structures             R   40
Basics of RDBMS             E   20
System Software             E   20
-----------------------------------------
-----------------------------------------
Sem: 2
-----------------------------------------
```

```
Course Name                Elective  Duration
-------------------------------------------
Computer Hardware           R   40
File Structures             R   40
Data Warehousing            E   20
Neural Networks             E   20
-------------------------------------------
```

1.7 Assignment 4: Parameterized Cursors

Objective: To be able to use parameterized explicit cursors in PL/SQL blocks.

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

Write a PL/SQL program to display the details of all the instructors for all the departments as shown in the below format.

Department Id: <departmentid>

Instructor Name Date of Joining Remaining Hours

<instructorname> <dateofjoining> <remaininghours>

…

Department Id: <departmentid>

Instructor Name Date of Joining Remaining Hours

<instructorname> <dateofjoining> <remaininghours>

…

1.8 Assignment 5: FOR UPDATE Cursor

Objective: To be able to use FOR UPDATE to update a set of attributes in explicit cursors in PL/SQL blocks.

Problem Description:

1. Write a PL/SQL block to update the gent's hostel fee to 15% and ladies hostel fee to 16%. If successful display the message **"Successfully Updated"** else display the message **"Not Updated"**.

2. Write a PL/SQL block to update the grade of each student. Do the necessary validations for all the scores as shown in the NOTE below. The calculation of totalscore is as shown below.

   totalscore :=semesterscore + projectscore + assignmentscore + internalscore ;

   Update the grade based on the range of score (given below) secured in totalscore.

Range	Grade
80-100	A
73-79	B+
65-72	B
55-64	C
0-54	D

NOTE:

The data in the registration table will be present only if you have completed the **Assignment No:2.3 of Day3** properly