

Lab Guide for RDBMS Essentials



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

COPYRIGHT NOTICE

All ideas and information contained in this document are the intellectual property of Education and Research Department, Infosys Technologies Limited. This document is not for general distribution and is meant for use only for the person they are specifically issued to. This document shall not be loaned to anyone, within or outside Infosys, including its customers. Copying or unauthorized distribution of this document, in any form or means including electronic, mechanical, photocopying or otherwise is illegal.

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	30-Jun-2010	Shinu Thomas	Rengarajan	Updated and checked for devsquare compatibility

Contents

COPYRIGHT NOTICE	II
DOCUMENT REVISION HISTORY	I
CONTENTS.....	II
1 BACKGROUND	3
1.1 DEMO1: WRITING A SIMPLE PL/SQL BLOCK	3
1.2 ASSIGNMENT 1: WRITING A SIMPLE PL/SQL BLOCK	4
1.3 DEMO2: ACCEPTING THE VALUES IN DECLARE SECTION.....	4
1.4 ASSIGNMENT 2: ACCEPTING THE VALUES IN DECLARE SECTION	5
1.5 DEMO3: ACCEPTING THE VALUES IN EXECUTION SECTION.....	5
1.6 ASSIGNMENT 3: ACCEPTING THE VALUES IN EXECUTION SECTION	6
1.7 ASSIGNMENT 4: USAGE OF BOOLEAN DATA TYPE.....	6
1.8 DEMO 4: USAGE OF ANCHORED DECLARATION (%TYPE)	7
1.9 ASSIGNMENT 5: USAGE OF ANCHORED DECLARATION (%TYPE)	8
1.10 ASSIGNMENT 6: SCOPE OF VARIABLES.....	9
1.11 ASSIGNMENT 7: BIND VARIABLE.....	10
1.12 ASSIGNMENT 8: PL/SQL OPERATORS	10
1.13 ASSIGNMENT 9: PL/SQL CONDITIONAL CONSTRUCTS	11

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.

1.1 Demo1: Writing a simple PL/SQL block

Objective: To be able to write a simple PL/SQL block and get familiarized with data types.

Platform: Use SQL*PLUS to solve this assignment. Create the tables and insert the data using the “CourseRegistrationDBDesign.sql” script. If you have all the tables already created in SQL*PLUS then read the problem description.

Problem Description:

Course Registration application has a department table with departmentid, departmentname and headofdepartment.

Write a PL/SQL block which declares variables for assigning departmentid, departmentname and headofdepartment details and display the same in execution section.

```
-- CODE 1.1
SET SERVEROUTPUT ON;

DECLARE
    v_departmentid NUMBER(2);
    v_departmentname VARCHAR2(30);
    v_headofdepartment VARCHAR2(4):='I101';
BEGIN
    v_departmentid := 10;
    v_departmentname := 'Computer Science';
    v_headofdepartment := 'I101';
    DBMS_OUTPUT.PUT_LINE('Department Id: '||v_departmentid);
    DBMS_OUTPUT.PUT_LINE('Department Name: '||v_departmentname);
    DBMS_OUTPUT.PUT_LINE('Department Head: '||v_headofdepartment);
END;
/

OUTPUT
Department Id: 10
Department Name: Computer Science
```

Department Head: I101

1.2 Assignment 1: Writing a simple PL/SQL block

Objective: To be able to write a simple PL/SQL block and get familiarized with data types.

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

1. Write a PL/SQL block to display the details of a particular applicant.

[Hint: Please refer the applicant table for appropriate variable names and data types. Assign some data in the declare section as shown in the Demo1 and display the details.]

1.3 Demo2: Accepting the values in DECLARE section

Objective: To be able to accept values and assign the values in the DECLARE section of PL/SQL

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

Write a PL/SQL block to accept the departmentid and departmentname based on its data type in DECLARE section and display the same in execution section of PL/SQL.

-- CODE 1.3

```
SET SERVEROUTPUT ON;
```

```
DECLARE
```

```
  v_departmentid NUMBER(2) := &v_deptid;
```

```
  v_departmentname VARCHAR2(30) := '&v_deptname';
```

```
BEGIN
```

```
  DBMS_OUTPUT.PUT_LINE('Department Id: '||v_departmentid);
```

```
  DBMS_OUTPUT.PUT_LINE('Department Name: '||v_departmentname);
```

```
END;
```

```
/
```

```
OUTPUT
```

```
Enter value for v_deptid: 10
```

```
old 2: v_departmentid NUMBER(2) :=&v_deptid;
new 2: v_departmentid NUMBER(2) :=10;
Enter value for v_deptname: Computer Science
old 3: v_departmentname VARCHAR2(30) := '&v_deptname';
new 3: v_departmentname VARCHAR2(30) := 'Computer Science';
Department Id: 10
Department Name: Computer Science
```

1.4 Assignment 2: Accepting the values in DECLARE section

Objective: To be able to accept values and assign the values in the DECLARE section of PL/SQL

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

1. Write a PL/SQL block to accept the branchid and branchname based on its data type in DECLARE section and display the same in EXECUTION section of PL/SQL.

1.5 Demo3: Accepting the values in EXECUTION section

Objective: To be able to accept values and assign the values in the EXECUTION section of PL/SQL

Problem Description:

Write a PL/SQL block to accept the departmentid and departmentname based on its data type in EXECUTION section and display the same in execution section of PL/SQL.

```
-- CODE 1.5

SET SERVEROUTPUT ON;

DECLARE
    v_departmentid NUMBER(2);
    v_departmentname VARCHAR2(30);
BEGIN
    v_departmentid :=&v_deptid;
    v_departmentname := '&v_deptname';
    DBMS_OUTPUT.PUT_LINE('Department Id: ' || v_departmentid);
    DBMS_OUTPUT.PUT_LINE('Department Name: ' || v_departmentname);
END;
```

```
/
```

OUTPUT

```
Enter value for v_deptid: 10
old 5: v_departmentid :=&v_deptid;
new 5: v_departmentid :=10;
Enter value for v_deptname: Computer Science
old 6: v_departmentname:='&v_deptname';
new 6: v_departmentname:='Computer Science';
Department Id: 10
Department Name: Computer Science
```

1.6 Assignment 3: Accepting the values in EXECUTION section

Objective: To be able to accept values and assign the values in EXECUTION section and displaying the same in the EXECUTION section of PL/SQL

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

1. Write a PL/SQL block to accept the branchid and branchname based on its data type in EXECUTION section and display the same in EXECUTION section of PL/SQL.

1.7 Assignment 4: Usage of BOOLEAN data type

Objective: To understand the usage of BOOLEAN data type

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

Analyze the following code and answer the questions given below.

```
-- CODE 1.7
```

```
DECLARE
    v_bool BOOLEAN;
BEGIN
    IF(v_bool IS NULL) THEN
        DBMS_OUTPUT.PUT_LINE('By default the value is NULL');
```



```

ELSIF(v_bool = TRUE) THEN
    DBMS_OUTPUT.PUT_LINE('By default the value is TRUE');
ELSE
    DBMS_OUTPUT.PUT_LINE('By default the value is FALSE');
END IF;
END;
/

```

Understanding the BOOLEAN:

What:

- 1) is the value of a BOOLEAN variable by default?
- 2) happens if we try to print the BOOLEAN variable?
- 3) are the values that can be assigned to a BOOLEAN variable?
- 4) are the values expected from a BOOLEAN variable comparison?

1.8 Demo 4: Usage of Anchored Declaration (%TYPE)

Objective: To be able to write a PL/SQL variable using anchored declarations.

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

Write a PL/SQL block to accept the departmentid and departmentname based on its data type (use %TYPE) in EXECUTION section and display the same in execution section of PL/SQL.

```

-- CODE 1.8

SET SERVEROUTPUT ON;

DECLARE
    v_departmentid department.departmentid%TYPE;
    v_departmentname department.departmentname%TYPE;
BEGIN
    v_departmentid :=&g_deptid;
    v_departmentname:='&g_deptname';
    DBMS_OUTPUT.PUT_LINE('Department Id: '||v_departmentid);
    DBMS_OUTPUT.PUT_LINE('Department Name: '||v_departmentname);
END;
/

OUTPUT

Enter value for g_deptid: 10
old   5:   v_departmentid :=&g_deptid;

```

```
new    5:    v_departmentid :=10;
Enter value for g_deptname: Computer Science
old    6:    v_departmentname:='&g_deptname';
new    6:    v_departmentname:='Computer Science';
Department Id: 10
Department Name: Computer Science
```

1.9 Assignment 5: Usage of Anchored Declaration (%TYPE)

Objective: To be able to write a PL/SQL variable using anchored declarations.

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

1. Write a PL/SQL block to accept the branchid and brabchname based on its data type (use %TYPE) in EXECUTION section and display the same in execution section of PL/SQL.
2. Analyze the following code and answer the questions given below.

-- CODE 1.9.1

```
DECLARE
  v_num1 NUMBER(7,2) :=2000.90;
  v_num2 v_num1%TYPE;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Num1: '||v_num1);
  DBMS_OUTPUT.PUT_LINE('Num2: '||v_num2);
END;
/
```

Explore:

State TRUE or FALSE

- i. The value of v_num2 = v_num1
- ii. %TYPE just maps the data type only
- iii. %TYPE assigns the data to the variable

-- CODE 1.9.2

```
DECLARE
  v_num1 NUMBER(7,2) NOT NULL:=2000.90;
  v_num2 v_num1%TYPE;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Num1: '||v_num1);
```

```

    DBMS_OUTPUT.PUT_LINE('Num2: '||v_num2);
END;
/

```

Explore:

State TRUE or FALSE

- i. The NOT NULL constraint of v_num1 is applicable to v_num2
- ii. %TYPE in this example maps the data type and the constraint if presents

-- CODE 1.9.3

```

DECLARE
    v_branchid branch.branchid%TYPE;
    v_branchname branch.branchname%TYPE;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Branch Id: '||v_branchid);
    DBMS_OUTPUT.PUT_LINE('Branch Name: '||v_branchname);
END;
/

```

**Note:**

branchname of branch table has NOT NULL constraint.

3. What are mapped to the variables if it refers to a column present in the table?
 - i. Only the value
 - ii. Only the data type
 - iii. Both value and data type
 - iv. Only data type and constraint
 - v. Only constraint
4. Modify the code 1.9.3 to refer to the column admissionstatus in applicant table, which has CHECK constraint and analyze what are mapped?

1.10 Assignment 6: Scope of variables

Objective: To understand the scope of variables.

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

Analyze the following code and answer the questions given below.

-- CODE 1.10

```

DECLARE
    v_projectscore NUMBER := 20;

```

```
BEGIN
DECLARE
    v_projectscore registration.projectscore%TYPE;
BEGIN
    v_projectscore:=15; -- this is the secured score
    DBMS_OUTPUT.PUT_LINE('Max score:'||v_projectscore);
    DBMS_OUTPUT.PUT_LINE('Secured score:'||v_projectscore);
END;
v_projectscore:=20; --this is the maximum score for the course.
DBMS_OUTPUT.PUT_LINE('Max score:'||v_projectscore);
END;
/
```

Explore:

1. In the inner block what will be the Max score?
2. Can the inner block's variable be displayed in the outer block?
3. Can the outer block's variable be displayed in the inner block?
4. How to display 20 in the place of Max score in the inner block?

[Hint: using labels (qualifiers), change the code accordingly]

1.11 Assignment 7: Bind variable

Objective: To be able to use bind variable in PL/SQL blocks.

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

1. Write a PL/SQL program to accept a value to a variable called v_studentid and assign it to a bind variable called g_studentid and display the same in the SQL prompt i.e. outside the PL/SQL program.
 2. Write a PL/SQL program to accept the value present in the bind variable g_studentid and display the value in the execution block of PL/SQL.
 3. What is the scope of the bind variable?
- [Hint: Try using the same variable in another instance of the SQL PLUS]

1.12 Assignment 8: PL/SQL Operators

Objective: To be able to use various operators in PL/SQL blocks.

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

1. Write a PL/SQL block to accept the registration details and score of a student (refer registration table for the structure) and display the total score scored by the student.
[Hint: totalscore = projectscore+ assignmentscore+ internalscore+ semesterscore]

**Note:**

- a. The projectscore should be between 0 and 20;
- b. The assignmentscore should be between 0 and 10;
- c. The internalscore should be between 0 and 20;
- d. The semester score should be between 0 and 50;

2. Write a PL/SQL program to display the result of a student in the following format.
[Hint: Use concatenation operator(||) as connector.]

Student Id	Course Id	Total
S001	C001	90

1.13 Assignment 9: PL/SQL Conditional Constructs

Objective: To be able to use conditional constructs in PL/SQL blocks.

Platform: Use SQL*PLUS to solve this assignment.

Problem Description:

1. Modify the Assignment 9(1) to calculate the grade of a student. The following is the business logic for calculating the grade from the totalscore obtained.

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

[Hint: if totalscore = 90 then the grade is A]

2. Implement the same code (Assignment 6(1)) using both IF-ELSE-END IF; and IF-ELSIF-END IF;.
3. Analyze the following code and predict the output for each input.

```
-- CODE 1.13
DECLARE
```

```
v_totalscore NUMBER:=&totalscore;
v_grade CHAR(2);
BEGIN
  IF v_totalscore >= 0 THEN
    v_grade:='F';
  ELSIF v_totalscore >=65 THEN
    v_grade:='B';
  ELSIF v_totalscore>=73 THEN
    v_grade:='B+';
  ELSIF v_totalscore >=80 THEN
    v_grade:='A';
  END IF;
  DBMS_OUTPUT.PUT_LINE('Grade:' || v_grade);
END;
/
```

4. Write a PL/SQL program to display 1 to 10 using FOR LOOP.
5. Write a PL/SQL program to display 1 to 10 using WHILE LOOP.
6. Write a PL/SQL program to display 10 to 1 using FOR LOOP. [Hint: use REVERSE]

Summary of this assignment:

You have learnt

- How to write a simple PL/SQL block.
- Various data types available and the differences.
- Accepting a value and displaying the same.
- How to use anchored declarations in PL/SQL
- The scope of variables in PL/SQL blocks.
- How to use bind variable in PL/SQL
- How to use operators in PL/SQL
- How to use conditional constructs in PL/SQL