

DBS501, FALL 20, ASSIGNMENT ONE, due on Monday, Nov 02nd by 7pm
e-mail the ANSWER as an ATTACHMENT, it must contain both Codes and Outputs
Your code must encompass ALL Test cases given here.

1) Write the PL/SQL block that will add a new row to the Departments table like shown:

Department_id → Highest value of all Id's increased by 50

Department_name → Testing

Manager_id → One with most people under supervision (do NOT assume it is a President, prove it with the code)

Location_id → You will be asked to input the City name where you want to place your new department. And that value will be converted to its related Location_id.

You will then display that new row and later undo the insert.

Your input value should be a VALID city name without any department. You need also to code for following cases: There is already ONE department in that city, there is MORE THAN ONE department there and the city is NOT listed in the Locations table.

You do NOT need to use cursor for this question, Here are the outputs:

Please provide the valid city without department:

	EMBED
--	-------

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
320	Testing	100	1100

Rollback complete.

Please provide the valid city without department:

	EMBED
--	-------

This city already contains department: Toronto

PL/SQL procedure successfully completed.

no rows selected

Rollback complete.

Please provide the valid city without department:

	EMBED
--	-------

This city has MORE THAN ONE department: Seattle

PL/SQL procedure successfully completed.

no rows selected

Rollback complete.

Please provide the valid city without department:

	EMBED
--	-------

This city is NOT listed: Belgrade

PL/SQL procedure successfully completed.

no rows selected

Rollback complete.

2) Write the PL/SQL block that will as input accept the start of the Course Description (one or more words) that is **definitely** used as a PREREQUISITE course. Then for each section of all courses that have this course (or courses) as a Prerequisite, determine the number of students registered. If this number is equal to or greater than 7, raise the user-defined exception and display message like “There are too many students for that section”. Otherwise, display how many students are in that section. **Use For Cursor Loop and Exception Handler as well.**

Make sure that your program is able to process all sections. Also, you need to deal with the situations where input points to a valid course that is NO prerequisite at all and also if input points to a non-existing course. Here are the possible outputs:

Enter the beginning of the Course Description in UPPER case:

	EMBED
--	-------

There are 5 students for section ID 85

There are 6 students for section ID 86

There are too many students for section 87

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

There are 5 students for section ID 88

There are too many students for section 89

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

There are 4 students for section ID 90

There are 2 students for section ID 91

There are 4 students for section ID 92

There are 0 students for section ID 93

There are 0 students for section ID 98

There are 4 students for section ID 146

There are too many students for section 147

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

There are 5 students for section ID 148

There are 1 students for section ID 149

There are 3 students for section ID 150

There are 2 students for section ID 151

PL/SQL procedure successfully completed.

Enter the beginning of the Course Description in UPPER case:

	EMBED
--	-------

**There is NO PREREQUISITE course that starts on: UNIX. Try again.
PL/SQL procedure successfully completed.**

Enter the beginning of the Course Description in UPPER case:

	EMBED
--	-------

**There is NO VALID course that starts on: SPORT. Try again.
PL/SQL procedure successfully completed.**

3) Write the PL/SQL block that will as input accept the start of the Course Description (one or more words) that possibly has a PREREQUISITE course. Then, it will display for each course that starts like the provided input the following components: Course Number, Description and Cost followed by the prerequisite Course Number, Description **and Cost.. Use For Cursor Loop. You should create a User-Defined Nested Record of four components, where the fourth component is Record again.**

Also, you need to deal with the situations where your input points to a valid course that has NO prerequisites at all and also if your input points to a non-existing course.

Here are the possible outputs:

Enter the beginning of the Course description in UPPER case:

	EMBED
--	-------

**Course: 144 - Database Design
Cost: 1195
Prerequisite: 420 - Database System Principles
Prerequisite Cost: 1195**

=====

**Course: 420 - Database System Principles
Cost: 1195
Prerequisite: 25 - Intro to Programming
Prerequisite Cost: 1195**

=====

PL/SQL procedure successfully completed.

Enter the beginning of the Course description in UPPER case:

	EMBED
--	-------

**There is NO prerequisite course for any course that starts on OPERATING. Try again.
PL/SQL procedure successfully completed.**

Enter the beginning of the Course description in UPPER case:

EMBED

**There is NO VALID course that starts on: SPORT. Try again.
PL/SQL procedure successfully completed.**

4) Write a PL/SQL block to display the course number and name and then for each section it will show the enrollment number. **Use Nested For Cursor Loops where the parent cursor will give information about each course and the child cursor will count the enrollment for each section. The child cursor will contain ONE input argument.**

You will be prompted to enter TWO key words within the course description and here are your possible outputs:

I) If you entered words JAVA and PROGRAM, the output looks like:

20 Intro to Java Programming

Section: 1 has an enrollment of: 4

Section: 2 has an enrollment of: 8

Section: 5 has an enrollment of: 3

Section: 4 has an enrollment of: 1

Section: 3 has an enrollment of: 5

Section: 7 has an enrollment of: 2

122 Intermediate Java Programming

Section: 1 has an enrollment of: 4

Section: 2 has an enrollment of: 3

Section: 4 has an enrollment of: 5

Section: 5 has an enrollment of: 8

Section: 3 has an enrollment of: 4

124 Advanced Java Programming

Section: 1 has an enrollment of: 5

Section: 2 has an enrollment of: 1

Section: 3 has an enrollment of: 2

146 Java for C/C++ Programmers

Section: 1 has an enrollment of: 1

Section: 2 has an enrollment of: 2

450 DB Programming in Java

Section: 1 has an enrollment of: 1
PL/SQL procedure successfully completed.

II) If you entered words INTRO and C, the output looks like:

20 Intro to Computers

Section: 2 has an enrollment of: 3

Section: 4 has an enrollment of: 2

Section: 8 has an enrollment of: 2

Section: 7 has an enrollment of: 2

240 Intro to the Basic Language

Section: 1 has an enrollment of: 12

Section: 2 has an enrollment of: 1

PL/SQL procedure successfully completed.

III) If you entered words INTRO and SOCCER, the output looks like:
There is NO course containing these 2 words. Try again.
PL/SQL procedure successfully completed.