Oracle Massive Open Online Course

Oracle MOOC: PL/SQL Fundamentals

Week 1

Homework for Lesson 2: Executing SQL Statements, Conditional Programming, and Looping

Homework is your chance to put what you've learned in this lesson into practice. This homework is not "graded" and you are encouraged to write additional code beyond what is asked.

Note:

- Ensure you - complete the setup instructions provided on the course page before attempting the homework.
- The solutions to the homework are NOT provided. We encourage you to develop your own solutions and also collaborate through the course community. Collaborating with co-learners is a fun way of learning!
- Completing the homework is NOT mandatory to get the course completion award.
- Post your questions, comments, or suggestions (if any) in the community @ https://community.oracle.com/community/technology_network_community/moocs/plsql-fundamentals
- We suggest you save your solution scripts for each assignment.

Watch out for:

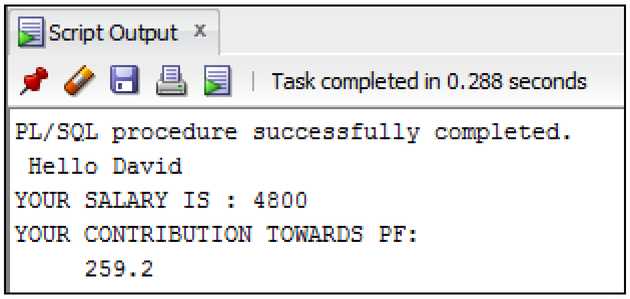
💻 - Reference video that teaches the corresponding concept in this MOOC.

💡 - Hints that can help you solve the assignment.

Assignment 1: Create an anonymous block that calculates the Provident Fund (PF) contribution amount for the faculty whose `faculty_id` is 105.

Oracle Massive Open Online Course

Sample Output:



```
PL/SQL procedure successfully completed.
 Hello David
YOUR SALARY IS : 4800
YOUR CONTRIBUTION TOWARDS PF:
     259.2
```

See 2-2: Using SQL Statements in PL/SQL Programs for reference.

Hints:

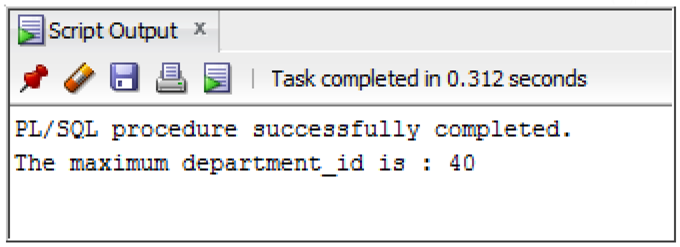
- Provident Fund (PF) is 12% of the base salary, and the base salary is 45% of the salary.
- Declare PL/SQL variables `v_basic_percent` and `v_pf_percent` to hold these percentage values, and assign the values 45 and 12 respectively.
- Declare two additional variables to hold the values of faculty's first name and salary. Use `%TYPE` attribute in this declaration.
- Write a `SELECT` query to retrieve the first name and salary of the faculty whose `faculty_ id` is 105.
- Calculate the contribution of the faculty towards PF. Try to use only one expression to calculate the PF.
- Write statements to print the values as per the expected output.

Assignment 2: Create a PL/SQL block that selects the maximum department ID in the `ad_departments` table and displays it. Save this script as `soln_02_02.sql`.

Oracle Massive Open Online Course

Oracle MOOC: PL/SQL Fundamentals

Sample output:



```
PL/SQL procedure successfully completed.
The maximum department_id is : 40
```

See 2-2: Using SQL Statements in PL/SQL Programs for reference.

Hint:

- Use the SQL group function `MAX` in the `SELECT` query.

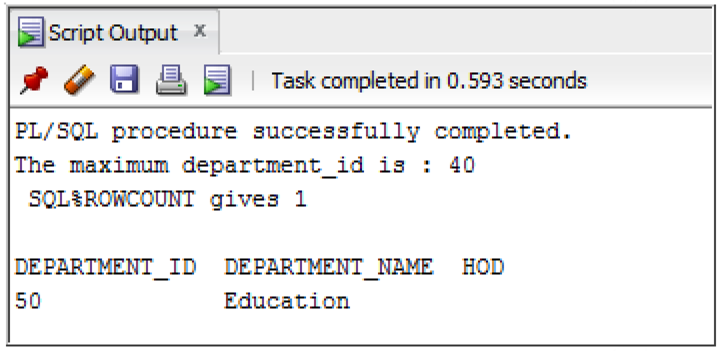
Assignment 3: Modify `soln_02_02.sql` (created in the above assignment) to insert a new record in the `ad_departments` table with the following values:

Department ID	maximum department ID in the `ad_departments` table + 10
Department Name	Education
HOD	null

Print the following in the output:

- Value of the maximum department id
- Number of rows inserted into the `ad_departments` table.
- New department record that is inserted.

Sample Output:



Script Output | Task completed in 0.593 seconds

```
PL/SQL procedure successfully completed.
The maximum department_id is : 40
 SQL%ROWCOUNT gives 1

DEPARTMENT_ID  DEPARTMENT_NAME  HOD
50             Education
```

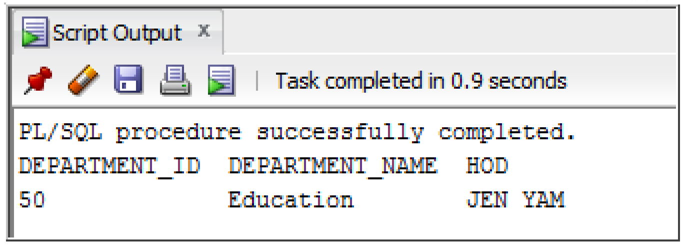
See 2-3: Executing DML Statements in PL/SQL for reference.

Hints:

- Use `SQL%ROWCOUNT` to find the number of records inserted into `ad_departments` table.
- Execute a select query with `max(dept_id)` in the `WHERE` clause to display the newly inserted record.

Assignment 4: Update the `HOD` value for the new record (inserted into the `ad_departments` table in the above assignment) to `'JEN YAM'`. Print the updated record.

Sample Output:



Script Output | Task completed in 0.9 seconds

```
PL/SQL procedure successfully completed.
DEPARTMENT_ID  DEPARTMENT_NAME  HOD
50             Education        JEN YAM
```

See 2-3: Executing DML Statements in PL/SQL for reference.

Oracle Massive Open Online Course

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Oracle Massive Open Online Course

Oracle MOOC: PL/SQL Fundamentals

Hints:

- Use the department id value calculated in assignment 3 (above) in the `WHERE` clause of the update statement.

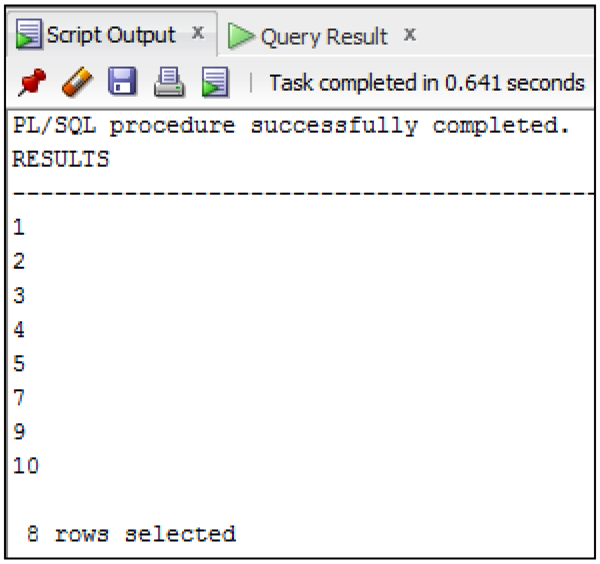
Assignment 5:

- Execute the following command to create the `messages` table.

  ```
  CREATE TABLE messages (results VARCHAR2(80));
  ```

- Write a PL/SQL block to insert numbers 1 through 10 excluding 6 and 8, into the `messages` table.
- Write a select query outside the PL/SQL block to print the numbers from the `messages` table.

Sample Output:



```
PL/SQL procedure successfully completed.
RESULTS
--------------------------------------------
1
2
3
4
5
7
9
10

 8 rows selected
```

See 2-4: Conditional Programming with PL/SQL for reference.

Hints:

- Commit before the end of the block.

Oracle Massive Open Online Course

Oracle MOOC: PL/SQL Fundamentals

Assignment 6:

- Execute the following code to create a `results` table that is based on the `ad_exam_results` table.

```
CREATE TABLE results AS

SELECT a.first_name, b.exam_id, c.course_name, b.marks

FROM ad_student_details a, ad_exam_results b,
ad_course_details c

WHERE a.student_id = b.student_id

AND c.course_id = b.course_id;
```

- Alter the results table to add a new column, `stars`, of `VARCHAR2` data type and size `50`.

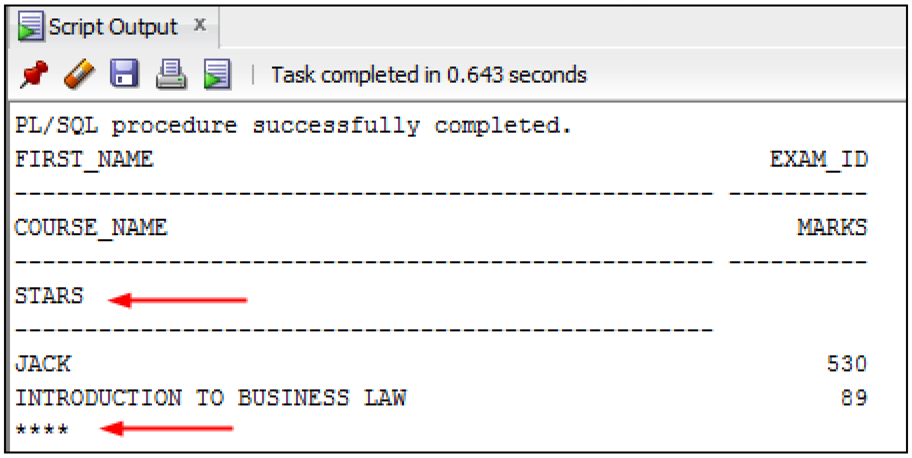
```
ALTER TABLE  results ADD stars VARCHAR2(50);
```

- Create a PL/SQL block that inserts an asterisk in the `stars` column for every 20 marks scored in an exam by the student whose first name is `JACK`.
- Print the updated record.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Oracle MOOC: PL/SQL Fundamentals

Sample Output:



```
PL/SQL procedure successfully completed.
FIRST_NAME                                          EXAM_ID
-------------------------------------------------- ----------
COURSE_NAME                                            MARKS
-------------------------------------------------- ----------
STARS
--------------------------------------------------
JACK                                                    530
INTRODUCTION TO BUSINESS LAW                             89
****
```

See 2-5: Implementing Loops in PL/SQL for reference.

Hints:

- In the declarative section of the block, declare a variable `v_fname` of type `results.first_name` and initialize it to 'JACK'. Declare a variable `v_exam_id` of type `results.exam_id` and initialize it to `530`.Declare a variable `v_asterisk` of type `results.stars` and initialize it to `NULL`. Create a variable `v_marks` of type `results.marks`.
- In the executable section, write logic using a `FOR` loop to append an asterisk (`*`) to the string for every 20 marks. For example, if the student scores 80 marks, the string of asterisks should contain four asterisks. If the student scores 100 marks, the string of asterisks should contain 5 asterisks. Update the `stars` column for the student with the string of asterisks. Commit before the end of the block.
- Display the row from the `results` table to verify whether your PL/SQL block has executed successfully.

Congratulations! You successfully practiced the concepts discussed in week 2.

Oracle Massive Open Online Course