

# Oracle MOOC: PL/SQL Fundamentals

Week 3

## Homework for Lesson 3: Using Composite Datatypes and Explicit Cursors

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 completed the [setup instructions](#) provided on the course page, before attempting the homework.
- The solutions to the homework are NOT provided. We encourage you to try it out and discuss in the course forum for further learning.
- The homework is NOT mandatory to get the course completion award.
- Post your questions, comments, or suggestions (if any) in the course forum @ [https://community.oracle.com/community/technology\\_network\\_community/moocs/plsql-fundamentals](https://community.oracle.com/community/technology_network_community/moocs/plsql-fundamentals)
- It is suggested to save your solution scripts for each assignment.

### Watch out for:



- Reference video that discussed the corresponding concept in this MOOC.



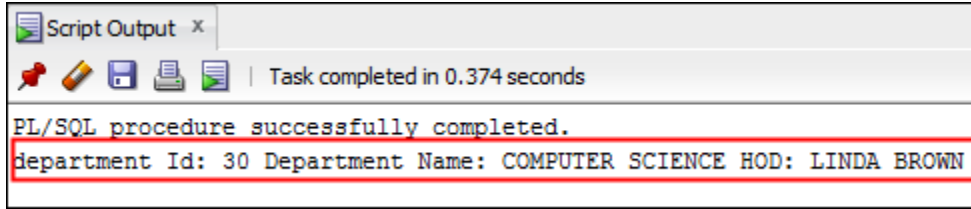
- Hints that can help you solve the assignment.

---

**Assignment 1:** Write a PL/SQL block to print information about department 30. Use a PL/SQL record based on the structure of the `AD_DEPARTMENTS` table.

# Oracle MOOC: PL/SQL Fundamentals

## Sample Output:



```
Script Output x
Task completed in 0.374 seconds
PL/SQL procedure successfully completed.
department Id: 30 Department Name: COMPUTER SCIENCE HOD: LINDA BROWN
```

 See [3-2: Understanding PL/SQL Records](#) for reference.

 Hints:

- Use the `%ROWTYPE` attribute on `ad_departments` table to declare the PL/SQL record variable.
- Use the `record.field` notation to print the record values.

**Assignment 2:** Create a PL/SQL block to retrieve the **names of some courses** from the `AD_COURSE_DETAILS` table and print each course name on the screen using an associative array.

- Declare an `INDEX BY table` `course_table_type` of type `ad_course_details.course_name`.
- Declare a variable `my_course_table` of type `course_table_type` to temporarily store the names of the courses.
- Declare two variables: `f_loop_count` and `v_course_id` of type `NUMBER`. Assign 6 to `f_loop_count` and 186 to `v_course_id`.
- Using a loop, retrieve the names of six courses and store the names in the associative array. Start with `course_id` 186. Increase `v_course_id` by 2 for every loop iteration. The following table shows the `course_id` for which you should retrieve the `course_name` and store in the associative array.

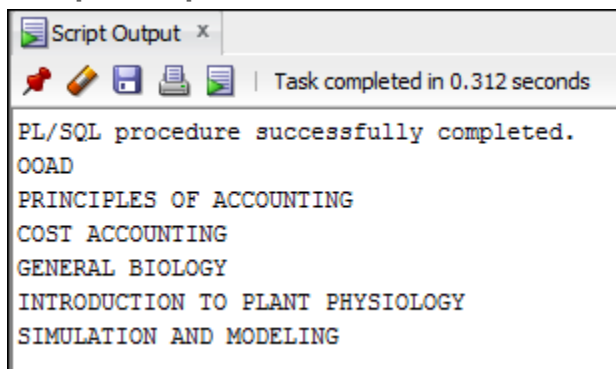
COURSE_ID	COURSE_NAME
188	OOAD
190	PRINCIPLES OF ACCOUNTING
192	COST ACCOUNTING

# Oracle MOOC: PL/SQL Fundamentals

194	GENERAL BIOLOGY
196	INTRODUCTION TO PLANT PHYSIOLOGY
198	SIMULATION AND MODELING

- e. Using another loop, retrieve the course names from the associative array and display them.
- f. Execute and save your script as `sol_03_02_soln.sql`.

## Sample Output:



```
Script Output x
Task completed in 0.312 seconds
PL/SQL procedure successfully completed.
OOAD
PRINCIPLES OF ACCOUNTING
COST ACCOUNTING
GENERAL BIOLOGY
INTRODUCTION TO PLANT PHYSIOLOGY
SIMULATION AND MODELING
```

 See [3-3: Understanding PL/SQL Collections](#) for reference.

---

**Assignment 3:** Create a PL/SQL block to retrieve and print the **complete details** of some courses from the `AD_COURSE _DETAILS` table using an associative array.

# Oracle MOOC: PL/SQL Fundamentals

## Sample Output:

```
Script Output x
Task completed in 0.343 seconds

PL/SQL procedure successfully completed.
Course ID: 188 Course Name: OOAD Session Id: 300 Department Id: 30
Course ID: 190 Course Name: PRINCIPLES OF ACCOUNTING Session Id: 100 Department Id: 10
Course ID: 192 Course Name: COST ACCOUNTING Session Id: 100 Department Id: 10
Course ID: 194 Course Name: GENERAL BIOLOGY Session Id: 200 Department Id: 20
Course ID: 196 Course Name: INTRODUCTION TO PLANT PHYSIOLOGY Session Id: 200 Department Id: 20
Course ID: 198 Course Name: SIMULATION AND MODELING Session Id: 300 Department Id: 30
```

 See [3-3: Understanding PL/SQL Collections](#) for reference.

 Hints:

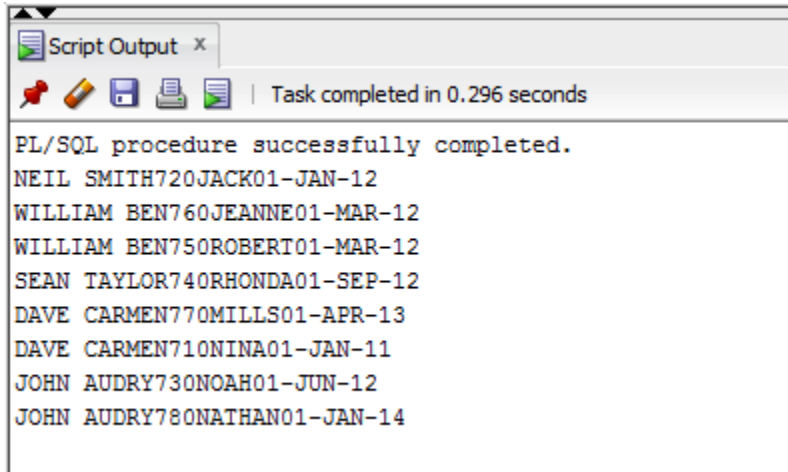
- Load `sol_03_02.sql` (created in assignment 2) and modify the declaration of the associative array to all details of all the courses. Use the `%ROWTYPE` attribute.
- Use an associative array with the `INDEX BY` table of records method.
- Modify the `SELECT` statement to retrieve all course information currently in the `AD_COURSE_DETAILS` table and store it in the associative array.
- Using another loop, retrieve the course information from the associative array and display the information.

---

**Assignment 4:** Create a PL/SQL block to declare a cursor named `c_parent_cursor`, that retrieves the parent ID, father's name, student ID, first name, and student registration year. Print the records from the cursor using a cursor `FOR` loop.

# Oracle MOOC: PL/SQL Fundamentals

## Sample Output:



```
Script Output x
Task completed in 0.296 seconds

PL/SQL procedure successfully completed.
NEIL SMITH720JACK01-JAN-12
WILLIAM BEN760JEANNE01-MAR-12
WILLIAM BEN750ROBERT01-MAR-12
SEAN TAYLOR740RHONDA01-SEP-12
DAVE CARMEN770MILLS01-APR-13
DAVE CARMEN710NINA01-JAN-11
JOHN AUDRY730NOAH01-JUN-12
JOHN AUDRY780NATHAN01-JAN-14
```



See [3-4: Working with Cursors](#) and [3-5: Exploring Explicit Cursors Further](#) for reference.



Hints:

- Declare the cursor using a JOIN query on the AD\_STUDENT\_DETAILS and AD\_PARENT\_INFORMATION tables.

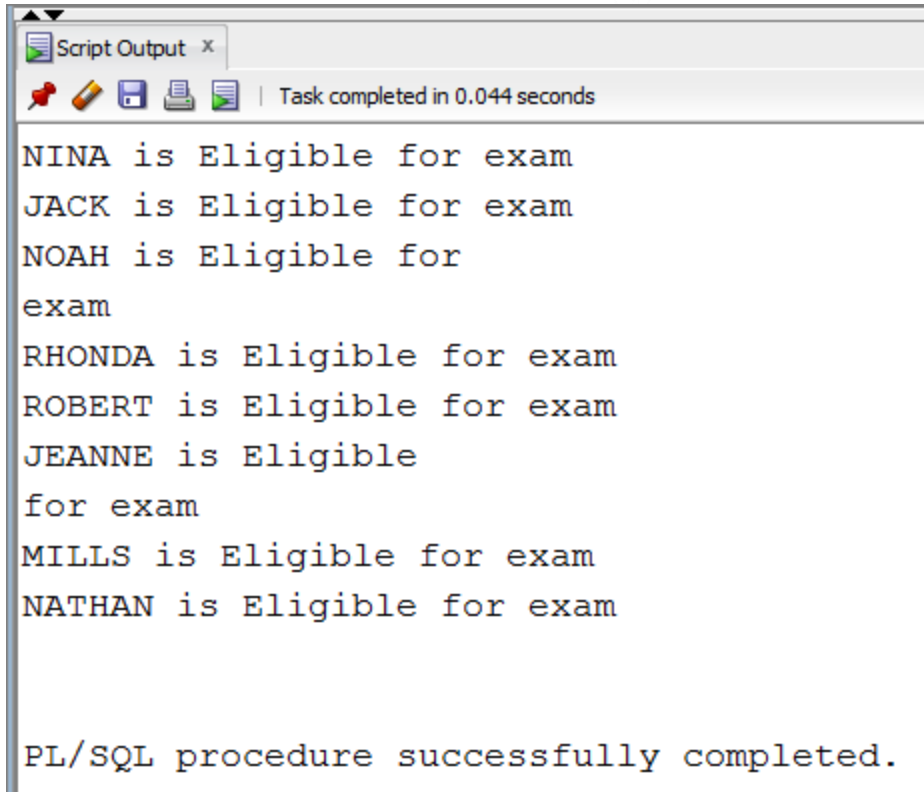
---

### Assignment 4:

- Create a PL/SQL block to declare a cursor named `c_student_cursor`, retrieves the `student_id`, `first_name`, and `no_of_days_off`.
- Use the cursor FOR loop to operate on the data retrieved. If the `no_of_days_off` is greater than 30, display the message “<<first\_name>> is Not Eligible for exam.” Otherwise, display the message “<<first\_name>> is Eligible for exam.”

# Oracle MOOC: PL/SQL Fundamentals

## Sample Output:



```
Script Output x
Task completed in 0.044 seconds

NINA is Eligible for exam
JACK is Eligible for exam
NOAH is Eligible for
exam
RHONDA is Eligible for exam
ROBERT is Eligible for exam
JEANNE is Eligible
for exam
MILLS is Eligible for exam
NATHAN is Eligible for exam

PL/SQL procedure successfully completed.
```



See [3-4: Working with Cursors](#) and [3-5: Exploring Explicit Cursors Further](#) for reference.



Hints:

- Declare the cursor using a `JOIN` query on the `AD_STUDENT_DETAILS` and `AD_STUDENT_ATTENDANCE` tables.

---

**Assignment 5:** Write a PL/SQL block that declares and uses two cursors—one without a parameter and one with a parameter.

- The first cursor retrieves the department number, department name, and HOD from the `AD_DEPARTMENTS` table for all departments whose department id is less than 50.
- The second cursor takes the department number as a parameter. It retrieves the following data from the `AD_COURSE_DETAILS` table: `course_id`, `course_name`, and `session_id` of the courses with `course_id` less than 190.

# Oracle MOOC: PL/SQL Fundamentals

## Sample Output:

```

Script Output x
Task completed in 0.327 seconds

PL/SQL procedure successfully completed.
Department Number : 10  Department Name : ACCOUNTING  HOD : MARK SMITH
-----
Department Number : 20  Department Name : BIOLOGY  HOD : DAVE GOLD
-----
Department Number : 30  Department Name : COMPUTER SCIENCE  HOD : LINDA BROWN
187  DATA STRUCTURES  300
188  COAD  300
-----
Department Number : 40  Department Name : LITERATURE  HOD : ANITA TAYLOR
175  AMERICAN LITERATURE  300
176  BUSINESS WRITING  200
189  COLLEGE READING  100
-----

```



See [3-4: Working with Cursors](#) and [3-5: Exploring Explicit Cursors Further](#) for reference.



### Hints:

- Declare a cursor `c_dept_cursor` to retrieve `department_id`, `department_name`, and `hod` for those departments with a `department_id` of less than 50. Order by `department_id`.
- Declare another cursor `c_course_cursor` that takes the department number as parameter and retrieves the following data from the `AD_COURSE_DETAILS` table: `course_id`, `course_name`, and `session_id` for those courses with a `course_id` less than 190.
- Declare variables to hold the values retrieved from each cursor. Use the `%TYPE` attribute while declaring variables.
- Open `c_dept_cursor` and use a simple loop to fetch values into the variables declared. Display the department number, department name, and hod. Use the appropriate cursor attribute to exit the loop.
- Open `c_course_cursor` by passing the current department number as a parameter. Start another loop and fetch the values of `course_cursor` into variables, and print all the details retrieved from the `AD_COURSE_DETAILS` table.

# Oracle MOOC: PL/SQL Fundamentals

- End the first loop and close `c_dept_cursor`. Then end the executable section.

Congratulations! You successfully practiced the concepts discussed in week 3.