Lab 3. PL/SQL Basics

Submission:

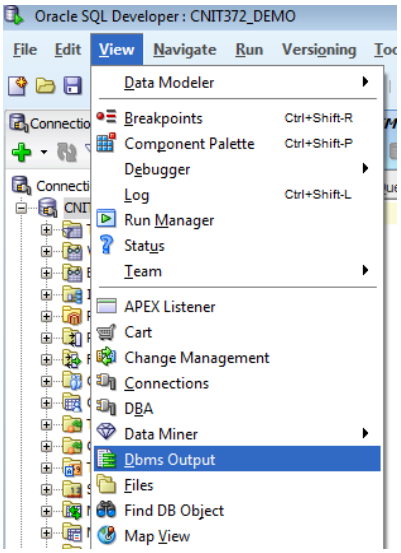
- If you decide to skip the lab, make sure you submit the **check-off questions** (highlighted with green background) in the text box of the check-off assignment item for this Lab before **Wednesday, noon to Brightspace**.
- **All students are expected to submit your answers to all lab questions in a text document with the name <lastname>_<firstname>_lab3.sql or .txt by the due date to Gradescope.**
- Please include both your code and the results in the **.txt** or **.sql** documents for full credits. For detailed requirements, please refer to the "Lab and Homework Submission Guideline."
- This Lab has Part A and Part B. Part A walks you through the PL/SQL basics, while Part B lets you apply what we've learned to solve some problems.
- You may submit either Part A or Part B for credit. If you submit both, only Part A will be graded.

Objectives:

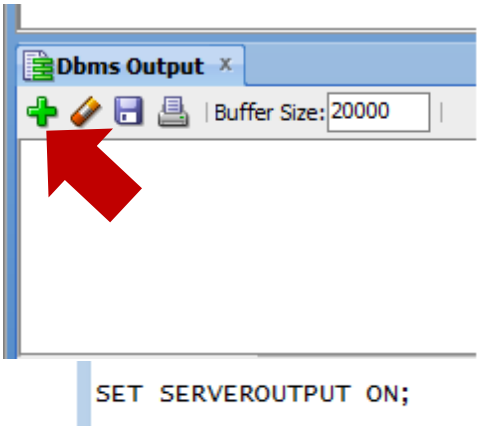
- Practice using simple PL/SQL commands
- Practice implementing PL/SQL Anonymous Blocks
- Practice using scalar variables
- Familiarization with the environmental variables common to PL/SQL Development

Environment Setup: Using the Oracle SQL Developer DBMS Output Pane

1. Turn on the DBMS Output Pane in SQL Developer



2. Set the pane to display the DBMS Output from your account by clicking the green plus sign (+) and selecting the appropriate connection.



Part A

Warm-up ***Put your answers in the comments. Each question is worth 0.7 point. (Total: 20.3 pts)***	
The SERVEROUTPUT environmental attribute	
1	Run the following PL/SQL code: `begin` `  DBMS_OUTPUT.PUT_LINE ('Hello World');` `end;` What is the output (e.g., what is printed to the screen)?
2	To have the result of a DBMS_OUTPUT statement written to the screen, you need to set the environmental attribute: serveroutput to 'ON'. Run the following command (there's no output for this question): `SET SERVEROUTPUT ON;`
3	Run the PL/SQL code from Question 1 again. What is the output now (e.g., what is printed to the screen)? Close SQL Developer. Then restart it. (Do not change anything this time) Run the PL/SQL code from Question 1 one more time. What is the output now (e.g., what is printed to the screen)? Does SQL Developer "remember" your SERVEROUTPUT preference? Now, set the environmental preference for SERVEROUTPUT to be back 'ON'.
4	Provide a definition and /or description of the SERVEROUTPUT attribute from appropriate Oracle documentation. Cite the source of your answer.
The DBMS_OUTPUT.PUT and the DBMS_OUTPUT.PUT_LINE	
5	Run the following PL/SQL code: `begin` `  DBMS_OUTPUT.PUT_LINE ('The Ubiquitous Hello World');` `end ;` What is the output (e.g., what is printed to the screen)?
6	Run the following PL/SQL code:

	`` begin DBMS_OUTPUT.PUT ('The') ; DBMS_OUTPUT.PUT (' ') ; DBMS_OUTPUT.PUT ('Ubiquitous') ; DBMS_OUTPUT.PUT (' ') ; DBMS_OUTPUT.PUT ('Hello') ; DBMS_OUTPUT.PUT (' '); DBMS_OUTPUT.PUT_LINE ('World') ; end ; ``` What is the output (e.g., what is printed to the screen)?
7	Based on questions 5 and 6, what do *you* believe the difference is between the DBMS_OUTPUT. PUT procedure and the DBMS_OUTPUT.PUT_LINE procedure?
8	Provide a definition and /or description of the DBMS_OUTPUT. PUT procedure from appropriate Oracle documentation. Cite the source of your answer.
9	Provide a definition and /or description of the DBMS_OUTPUT. PUT_LINE procedure from appropriate Oracle documentation. Cite the source of your answer.
10	Based on the definitions provided in 8 and 9, what is the difference between the DBMS_OUTPUT. PUT procedure and the DBMS_OUTPUT.PUT_LINE procedure?
Prompting User Input	
11	Run the following PL/SQL code: ``` begin DBMS_OUTPUT.PUT_LINE ('My name is ' || '&sv_YourName') ; end ; ``` What happens initially? What is the subsequent output (e.g., what is printed to the screen)?
The VERIFY environmental attribute	
12	Run the following PL/SQL code (there's no output for this question): ``` SET VERIFY ON; ```
13	Run the PL/SQL code from Q11 again. What is the output (e.g., what is printed to the screen)? Include everything outputted by the database.
14	Run the following PL/SQL code (there's no output for this question): ``` SET VERIFY OFF; ```
15	Run the PL/SQL code from Question 11 again. What is the output now (e.g., what is printed to the screen)? Include everything outputted by the database.
16	Provide a definition and /or description of the VERIFY attribute from appropriate Oracle documentation. Cite the source of your answer.
Default persistence of user input	

The prefix 'sv_' is a naming convention used to indicate that it is a PL/SQL **s**ubstitution **v**ariable.
The ampersand **'&'** indicates to the PL/SQL engine that the object being defined is a substitution variable.

17 Run the following PL/SQL code twice:

```
begin
        DBMS_OUTPUT.PUT_LINE ('My name is ' || '&sv_YourName') ;
end ;
```

How many times are you prompted for the value of **sv_YourName**? What does this imply about the persistence of inputted values?

Reuse of variables

18 Run the following PL/SQL code:

```
begin
        DBMS_OUTPUT.PUT_LINE ('Today is ' || '&sv_day') ;
        DBMS_OUTPUT.PUT_LINE ('Tomorrow is ' || '&sv_day') ;
end ;
```

Can you use the same variable name twice within the same unnamed block? What occurs if you attempt to do so?

Local persisting of user input

19 Run the following PL/SQL code:

```
begin
  DBMS_OUTPUT.PUT_LINE ('Today is ' || '&&sv_day') ;
  DBMS_OUTPUT.PUT_LINE ('Tomorrow is ' || '&sv_day') ;
end ;
```

What is the output (e.g., what is printed to the screen)?

Were you prompted to provide a value for the variable sv_day when the second DBMS_OUTPUT was executed? What is different that caused this?

20 Run the PL/SQL code from Question 19 again.
What is the output now (e.g., what is printed to the screen)?
Were you prompted for the variable sv_day? What does this imply about the persistence of variable input captured using the && command?

Variables

Variables are typically declared in the *declaration* section of unnamed blocks.
Initial values are often assigned in this section as well (including those supplied by users).

Variables: Simple Usage

21 Run the following PL/SQL code:

```
declare
        V_DAY  varchar2(10) := '&sv_day1' ;
begin
        DBMS_OUTPUT. PUT_LINE ('Today is ' || V_DAY) ;
end ;
```

What is the output (e.g., what is printed to the screen)?

Variables & Type Conversions

22 Run the following PL/SQL code:

```
declare
   V_DAY  varchar2(10) ;

begin
  V_DAY := to_char (sysdate, 'Day');

  DBMS_OUTPUT. PUT_LINE ('Today is  ' || V_DAY);
  DBMS_OUTPUT. PUT_LINE ('Tomorrow is ' || to_char (sysdate +1, 'Day'))

end ;
```

What is the output (e.g., what is printed to the screen)?

Number of Values a Scalar Variable Can Contain

23 Run the following SQL query:

```
select  employeeID
from    employee
where   employeeID = '100001';
```

How many rows are returned by this query?

24 Run the following PL/SQL code:

```
declare
   V_EMPLOYEEID EMPLOYEE.EMPLOYEEID%TYPE;
   V_LASTNAME     EMPLOYEE.LASTNAME%TYPE;
   V_FIRSTNAME   EMPLOYEE.FIRSTNAME%TYPE;

begin
   select       EMPLOYEEID,   LASTNAME,   FIRSTNAME
      into      V_EMPLOYEEID, V_LASTNAME, V_FIRSTNAME
   from EMPLOYEE
   where EMPLOYEEID = '100001';

   DBMS_OUTPUT.PUT_LINE ('Employee ID       LASTNAME       FIRSTNAME');
   DBMS_OUTPUT.PUT_LINE ('========================================');
   DBMS_OUTPUT.PUT (V_EMPLOYEEID);
   DBMS_OUTPUT.PUT ('          ');
   DBMS_OUTPUT.PUT (V_LASTNAME);
   DBMS_OUTPUT.PUT ('         ');
   DBMS_OUTPUT.PUT_LINE (V_FIRSTNAME);

end;
```

What is the output (e.g., what is printed to the screen)?

25 Run the following SQL query. How many rows are returned?

```
select  employeeID
from    employee;
```

26

Run the following PL/SQL code:

```
declare
   V_EMPLOYEEID EMPLOYEE.EMPLOYEEID%TYPE;
   V_LASTNAME     EMPLOYEE.LASTNAME%TYPE;
   V_FIRSTNAME   EMPLOYEE.FIRSTNAME%TYPE;

begin
   select       EMPLOYEEID,   LASTNAME,   FIRSTNAME
      into      V_EMPLOYEEID, V_LASTNAME, V_FIRSTNAME
   from EMPLOYEE;

   DBMS_OUTPUT.PUT_LINE ('Employee ID       LASTNAME       FIRSTNAME');
   DBMS_OUTPUT.PUT_LINE ('=========================================');
   DBMS_OUTPUT.PUT (V_EMPLOYEEID);
   DBMS_OUTPUT.PUT ('           ');
   DBMS_OUTPUT.PUT (V_LASTNAME);
   DBMS_OUTPUT.PUT ('         ');
   DBMS_OUTPUT.PUT_LINE (V_FIRSTNAME);

end;
```

Note: there is no WHERE clause in the embedded query above.
What is the output (e.g., what is printed to the screen)?
Hint: You should see an error.

27

Based on Q25, state how many values you are attempting to assign to each of the variables (V_EMPLOYEEID, V_LASTNAME, V_FIRSTNAME) in Q26.
Can you successfully do this? Or does PL/SQL throw an exception?

28

Based on your answer to Question 27, how many values can be held at a given time in a scalar-type variable?

29

Modify the code in Q24 to use EMPLOYEE%ROWTYPE. To do that, declare a variable V_EMPLOYEE of type EMPLOYEE%ROWTYPE. Modify select and as shown below.
What is the output (e.g., what is printed to the screen)?
Explain the code differences between Q24 and the code below.

```
declare
    V_EMPLOYEE employee%ROWTYPE;
begin
    select * into V_EMPLOYEE from EMPLOYEE where EMPLOYEEID='100001';
    DBMS_OUTPUT.PUT_LINE('Employee ID   LASTNAME    FIRSTNAME');
    DBMS_OUTPUT.PUT_LINE('==================================');
    DBMS_OUTPUT.PUT_LINE(V_EMPLOYEE.EMPLOYEEID||'        '||V_EMPLOYEE.LASTNAME||'     '||V_EMPLOYEE.FIRSTNAME);
end;
/
```

Writing Simple Anonymous Blocks (Total: 4.7 points)

30 1pt

How would you declare a variable to store telephone area codes (the first 3 digits of the phone number)?
(No need to include DECLARE/BEGIN/END keywords, just put one line of declaration.)

31a 2 pt

Write a simple anonymous block to store the largest number of customers (grouped by area code) in a variable.

31b 1.7 pt

Write a simple anonymous block to print the area code with the largest number of customers to dbms output. (Hint: extend your block in 31a)

Part B

1. Create a block that will print current date to the screen using PUT_LINE command. (2 pts)
2. Create a block that will separately print current day of the week, using PUT command, and a day of the week of the same date a year ago, also using PUT command. (2 pts)
3. Outline the difference between PUT and PUT_LINE commands. (1 pts)
4. Declare a variable that can store your birth date. Calculate your age, using the today's date, and print the output to the screen, including your name, your birth date and how old you are. Use type conversion functions in your code. (2 pts)
5. In the code you wrote for Q4, which variables can be declared as a constant? (0.5 pts)
6. Create an SQL query that displays an employee who has the most people that (s)he supervises directly. (1 pts)
7. Now write a block that will do a similar operation as Q6. Use %type to declare the needed variables. Print this employee's first name, last name, and the number of people that (s)he supervises. (1.5 pts)
8. Modify the query in the Q6 to list all supervisors and the number of people that they supervise. Interpret the results in your own words. (1.5 pts)
9. Create a block that performs the query in Q8. Attempt to store the results in the current variables. You should see an error – why do you think it happens? (1.5 pts)
10. Try running the following code in your SQL developer. What is the output? How can you correct the code to make it work? (1.5 pt)

```
DECLARE
     myname varchar2(10) NOT NULL := '';
BEGIN
     dbms_output.put_line('my name is '|| myname);
END;
```

11. Try running the following code. What's the output? Briefly explain. (1.5 pt)

```
DECLARE
     myname CONSTANT varchar2(10) NOT NULL := '';
BEGIN
     dbms_output.put_line('my name is '|| myname);
END;
```

12. Is your correction for Q10 going to correct the code in Q11 as well? Briefly explain why. (1 pt)

13. Give an example of an **in**valid variable name for PL/SQL. (1 pt)

Consider the following nested blocks:

```
<<OUTER_BLOCK>>
declare
      c_id VARCHAR2(20) := 'C-300001';
      o_id VARCHAR2(20);
begin
      <<INNER_BLOCK1>>
      declare
            c_id INTEGER;
      begin
      SELECT substr(CUSTOMERID,3,6) INTO c_id FROM CUSTOMER
      WHERE CUSTOMERID = c_id;
      [Q15.a]
      end INNER_BLOCK1;
      [Q15.b]

      <<INNER_BLOCK2>>
      declare
            o_id INTEGER;
      begin
         SELECT max(orderid) INTO o_id FROM CUSTORDER
         WHERE CUSTOMERID = c_id;
        [Q16.a]
      end INNER_BLOCK2;
      [Q16.b]
end;
```

14. Describe which part of the code above will cause an error (because of which the code won't run), and how you can fix it. (2 pts)
15. Which part of the code above will not cause an error but should be improved? (1 pt)
16. After your correction in Q14, what are the values of c_id at [Q15.a] and [Q15.b]? (1 pt)
17. What are the values of o_id at [Q16.a] and [Q16.b]? (1 pt)
18. Write a block to display the weights of the heaviest part in the Power and Software categories, using the format "The heaviest part in the Power category is XYZ, and the heaviest part in the Software category is ABC." (2 pts)