

## Lab 4 (Part 1). PL/SQL Control Structures

### Submission:

- If you decide to skip the lab, make sure you submit the **check-off questions** (highlighted with green background) in a text document named **<lastname>\_<firstname>\_checkoff.sql or .txt** before **Wednesday, 7:30 am to Brightspace**.
- **All students are expected to submit your answers to all lab questions in a text document with the name <lastname>\_<firstname>\_lab4\_part1.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.”

### Objectives:

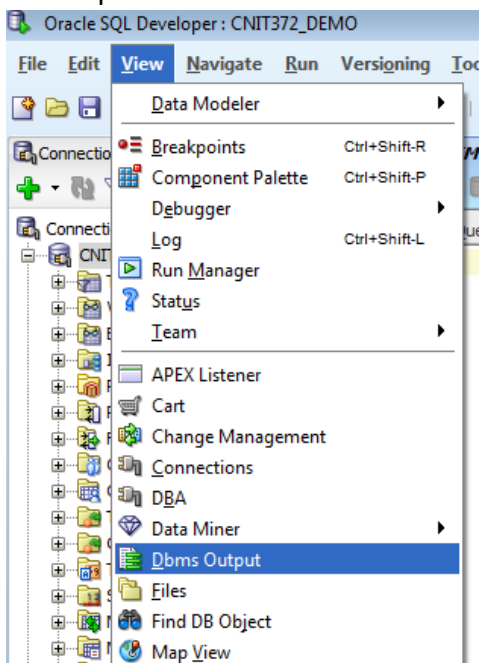
- Practice implementing PL/SQL Control Structures

### Notes:

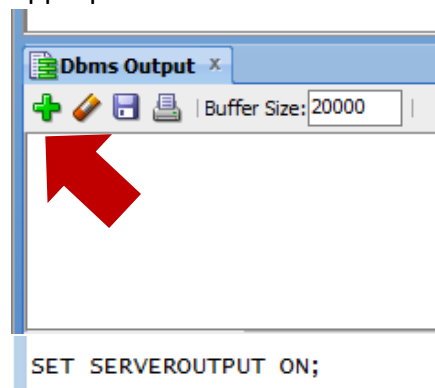
- With the upcoming deadlines of your Homework 1 and Milestone 1, I split the original Lab 4 into two parts, to reduce your workload this week. This part covers the first **10 points** of your Lab 4.

### Preparation: 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.



Conditional Selection Statements	
Basic IF ... THEN conditional structure	
<b>1a</b> (0.5 pt)	<p>Run the following SQL query:</p> <pre>select jobtitle, count(employeeid) from employee group by jobtitle;</pre> <p>What are the results of this query?</p>
<b>1b</b> (0.5 pt)	<p>Run the following PL/SQL code:</p> <pre>DECLARE   V_NUMBER_EMPLOYEES    NUMBER;   V_JOBTITLE             VARCHAR2(50) := '&amp;v_JOBTITLE';  BEGIN   select count(employeeID)     into V_NUMBER_EMPLOYEES   from employee   where jobtitle = V_JOBTITLE;    IF V_NUMBER_EMPLOYEES &lt; 1 THEN     DBMS_OUTPUT.PUT_LINE ('There are no employees with the Job Title: '    V_JOBTITLE);   ELSIF V_NUMBER_EMPLOYEES &lt; 4 THEN     DBMS_OUTPUT.PUT_LINE ('There are between 1 and 3 employees with the Job Title: '    V_JOBTITLE);   ELSIF V_NUMBER_EMPLOYEES &lt; 7 THEN     DBMS_OUTPUT.PUT_LINE ('There are between 4 and 6 employees with the Job Title: '    V_JOBTITLE);   ELSE     DBMS_OUTPUT.PUT_LINE ('There are 7 or more employees with the Job Title: '    V_JOBTITLE);   END IF;  END;</pre> <p>When prompted for Job Title, provide the following value: CIO (this is case sensitive) What is the output (e.g., what is printed to the screen)?</p>
<b>1c</b> (0.5 pt)	<p>Again, run the PL/SQL code from 1b. When prompted for Job Title, provide the following value: Accountant (this is case sensitive) What is the output (e.g., what is printed to the screen)?</p>
<b>1d</b> (0.5 pt)	<p>Again, run the PL/SQL code from 1b. When prompted for Job Title, provide the following value: Engineer (this is case sensitive) What is the output (e.g., what is printed to the screen)?</p>
<b>1e</b> (0.5 pt)	<p>Again, run the PL/SQL code from 1b. When prompted for Job Title, provide the following value: Assembly (this is case sensitive) What is the output (e.g., what is printed to the screen)?</p>
Basic CASE conditional structure	
<b>2a</b> (0.5 pt)	<p>Run the following PL/SQL code:</p>

	<pre> DECLARE   V_NUMBER_EMPLOYEES  NUMBER;   V_JOBTITLE          VARCHAR2(50) := '&amp;v_JOBTITLE';  BEGIN   select count(employeeID)     into V_NUMBER_EMPLOYEES   from employee   where jobtitle = V_JOBTITLE;    CASE V_NUMBER_EMPLOYEES     WHEN 0 THEN       DBMS_OUTPUT.PUT_LINE ('There are no employees with the Job Title: '    V_JOBTITLE);     WHEN 1 THEN       DBMS_OUTPUT.PUT_LINE ('There are between 1 and 3 employees with the Job Title: '    V_JOBTITLE);     WHEN 2 THEN       DBMS_OUTPUT.PUT_LINE ('There are between 1 and 3 employees with the Job Title: '    V_JOBTITLE);     WHEN 3 THEN       DBMS_OUTPUT.PUT_LINE ('There are between 1 and 3 employees with the Job Title: '    V_JOBTITLE);     WHEN 4 THEN       DBMS_OUTPUT.PUT_LINE ('There are between 4 and 6 employees with the Job Title: '    V_JOBTITLE);     WHEN 5 THEN       DBMS_OUTPUT.PUT_LINE ('There are between 4 and 6 employees with the Job Title: '    V_JOBTITLE);     WHEN 6 THEN       DBMS_OUTPUT.PUT_LINE ('There are between 4 and 6 employees with the Job Title: '    V_JOBTITLE);     ELSE       DBMS_OUTPUT.PUT_LINE ('There are 7 or more employees with the Job Title: '    V_JOBTITLE);   END CASE;  END;</pre>
	<p>When prompted for Job Title, provide the following value: CIO.</p> <p>What is the output (e.g., what is printed to the screen)?</p>
<p><b>2b</b> (0.5 pt)</p>	<p>Again, run the PL/SQL code from 2a. When prompted for Job Title, provide the following value: Accountant.</p> <p>What is the output (e.g., what is printed to the screen)?</p>
<p><b>2c</b> (0.5 pt)</p>	<p>Again, run the PL/SQL code from 2a. When prompted for Job Title, provide the following value: Engineer.</p> <p>What is the output (e.g., what is printed to the screen)?</p>
<p><b>2d</b> (0.5 pt)</p>	<p>Again, run the PL/SQL code from 2a. When prompted for Job Title, provide the following value: Assembly.</p> <p>What is the output (e.g., what is printed to the screen)?</p>
<p><b>Refining the Code</b></p> <p>Refine your code to be more programmatically efficient in its structure through the use of variables in your <b>CASE</b> conditional statement.</p>	
<p><b>3</b> (1.5 pt)</p>	<p>Modify the PL/SQL code from 2a, such that</p> <ol style="list-style-type: none"> <li>the data type of the variable <b>V_JOBTITLE</b> has the same data type as the column <b>jobtitle</b>, no matter how the schema changes.</li> <li>you can use only a single call to the DBMS_OUTPUT.PUT_LINE procedure. Use the local variable V_STAFF_LEVEL to facilitate this, as shown below. Then run the code.</li> </ol>

	<pre> DECLARE   V_NUMBER_EMPLOYEES  NUMBER;   V_JOBTITLE          EMPLOYEE.JOBTITLE%TYPE := '&amp;v_jobtitle';   V_STAFF_LEVEL       VARCHAR2(100); BEGIN   select count(employeeID)     into V_NUMBER_EMPLOYEES   from employee   where jobtitle = V_JOBTITLE;    CASE V_NUMBER_EMPLOYEES     WHEN 0 THEN       V_STAFF_LEVEL := 'There are no employees with the Job Title: ';     WHEN 1 THEN       V_STAFF_LEVEL := 'There are between 1 and 3 employees with the Job Title: ';     WHEN 2 THEN       V_STAFF_LEVEL := 'There are between 1 and 3 employees with the Job Title: ';     WHEN 3 THEN       V_STAFF_LEVEL := 'There are between 1 and 3 employees with the Job Title: ';     WHEN 4 THEN       V_STAFF_LEVEL := 'There are between 4 and 6 employees with the Job Title: ';     WHEN 5 THEN       V_STAFF_LEVEL := 'There are between 4 and 6 employees with the Job Title: ';     WHEN 6 THEN       V_STAFF_LEVEL := 'There are between 4 and 6 employees with the Job Title: ';     ELSE       V_STAFF_LEVEL := 'There are 7 or more employees with the Job Title: ';   END CASE;    -- Output user-friendly response   DBMS_OUTPUT.PUT_LINE (V_STAFF_LEVEL    V_JOBTITLE); END; </pre> <p>When prompted for Job Title, provide the following value: Engineer. What is the output (e.g., what is printed to the screen)?</p>
--	---

**Loop Constructs****Creating a "Simple" Loop in PL/SQL**

<b>4a</b> (0.5 pt)	<p>Using the following code, create a simple loop.</p> <pre> 1 DECLARE 2     MY_COUNT INTEGER := '&amp;MY_COUNT'; 3     MY_COUNTER INTEGER := 1; 4     MY_NUMBER INTEGER; 5 BEGIN 6     LOOP 7         MY_NUMBER := dbms_random.value(1,MY_COUNT); 8         DBMS_OUTPUT.PUT (MY_NUMBER    ', '); 9         MY_COUNTER := MY_COUNTER + 1; 10        EXIT WHEN MY_COUNTER &gt; MY_COUNT; 11    END LOOP; 12    DBMS_OUTPUT.PUT_LINE(''); 13 END; </pre> <p>When prompted for MY_COUNT, provide the value 5. What is the output (e.g., what is printed to the screen)?</p>
<b>4b</b> (0.5 pt)	<p>Explain what the simple loop "does".</p>
	<b>Creating a WHILE Loop in PL/SQL</b>
<b>4a</b> (1 pt)	<p>Rewrite the above simple loop with WHILE Loop.</p>

<b>4b</b> (0.5 pt)	Verify your WHILE LOOP by running it. Include the resulting output in your submission.
<b>Creating a FOR Loop in PL/SQL</b>	
<b>5a</b> (1 pt)	Rewrite the simple loop in 4a with FOR loop.
<b>5b</b> (0.5 pt)	Verify your FOR LOOP by running it. Include the resulting output in your submission.