

Lab 7. PL/SQL Packages

Point Distribution:

Question Number	Points
1a, 1b, 1c, 1d, 1f, 2a, 2b, 2c, 2d, 2e, 2f, 2i	0.75pt x 12
1e, 2g, 2h, 3c, 3f	1pt x 5
3a, 3b, 3d, 3e	1.5pt x 4
Total	20

Submission: A typed document with extension “.txt” OR “.sql” must be uploaded to the <<lab7>> submission link on Gradescope.

Objectives:

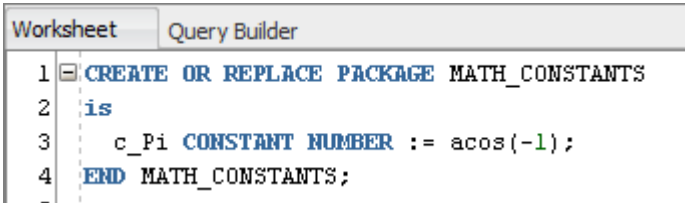
- Practice implementing PL/SQL Packages
- Practice implementing overloaded subroutines within PL/SQL packages
- There are no check offs for this lab. Lab attendance is optional as well.

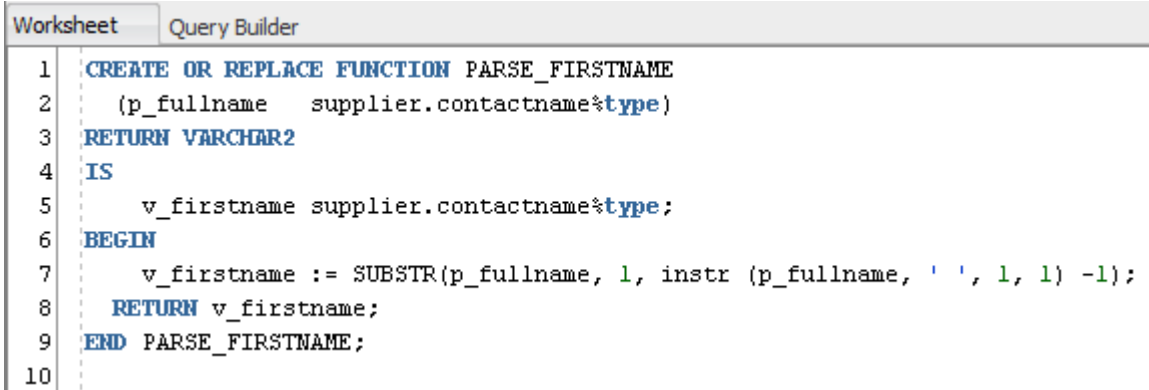
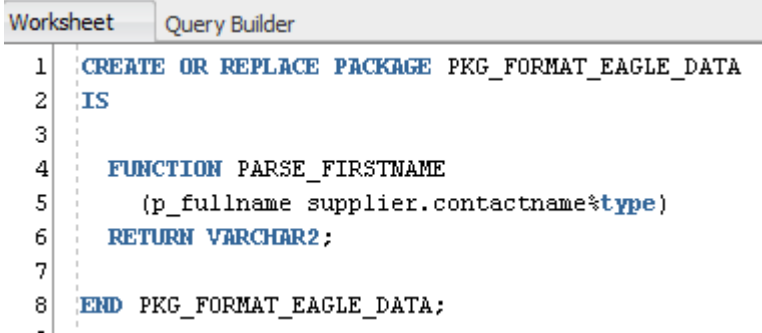
Questions

Specification-only Packages

Some Packages do not contain a body. Instead, they possess only a specification section.

These packages are typically used to hold values for constants.

1a	<p>Using the code provided below, create the MATH_CONSTANTS specification-only package.</p>  <pre> 1 CREATE OR REPLACE PACKAGE MATH_CONSTANTS 2 IS 3 c_Pi CONSTANT NUMBER := acos(-1); 4 END MATH_CONSTANTS; </pre>
1b	<p>What is the acos() function? What is the value of acos(-1)? What is this value typically referred to?</p> <p>NOTE: acos stands for arc-cosine</p>
1c	<p>What is the code necessary to retrieve the value of c_Pi and print it to the screen as output? Include both the code and the results in your submission.</p>
1d	<p>Try retrieving the value of c_Pi as you would call a function (using a SELECT statement). Can you do so?</p> <p>Include both the code and the results in your submission.</p>
1e	<p>Create a PL/SQL function named CALCULATE_AREA that accepts a single numeric parameter: p_radius. This module should:</p> <ul style="list-style-type: none"> • calculate the area of a circle of the specified radius (if you don't remember the formula, please look it up)

	<ul style="list-style-type: none"> use the value of c_Pi stored in the MATH_CONSTANTS package when calculating the area of the circle return a numeric value representing the area of the circle rounded to 3 digits
1f	<p>Test CALCULATE_AREA with the following values:</p> <ul style="list-style-type: none"> radius = 5 radius = 10 radius = 15 <p>Include both the code and the results in your submission.</p>
<h3>General PL/SQL Packages</h3> <p>Most PL/SQL packages contain both a specification section and a body section. However, you may wish to test your functions and procedures prior to inserting them into the package.</p>	
2a	<p>Create the function PARSE_FIRSTNAME shown below.</p>  <pre> 1 CREATE OR REPLACE FUNCTION PARSE_FIRSTNAME 2 (p_fullname supplier.contactname%type) 3 RETURN VARCHAR2 4 IS 5 v_firstname supplier.contactname%type; 6 BEGIN 7 v_firstname := SUBSTR(p_fullname, 1, instr (p_fullname, ' ', 1, 1) -1); 8 RETURN v_firstname; 9 END PARSE_FIRSTNAME; 10 </pre>
2b	<p>What is the code necessary to use the PARSE_FIRSTNAME function to retrieve the first name of each contact in the Eagle Electronics SUPPLIER table?</p> <p>Include both the code and the results in your submission.</p>
2c	<p>Once we are confident that the function is working correctly, we can remove (drop) the “naked” function and subsequently incorporate it into a package.</p> <p>Drop the PARSE_FIRSTNAME function.</p>
2d	<p>As shown below, create the package specification for a package named PKG_FORMAT_EAGLE_DATA. This package (at least initially) contains a single function named PARSE_FIRSTNAME.</p>  <pre> 1 CREATE OR REPLACE PACKAGE PKG_FORMAT_EAGLE_DATA 2 IS 3 4 FUNCTION PARSE_FIRSTNAME 5 (p_fullname supplier.contactname%type) 6 RETURN VARCHAR2; 7 8 END PKG_FORMAT_EAGLE_DATA; 9 </pre> <p>Notice the interface for the PARSE_FIRSTNAME function is the module type (function), module name (PARSE_FIRSTNAME), parameter list, and return type.</p>

2e	<p>Subsequently create the package body for the PKG_FORMAT_EAGLE_DATA package as shown below.</p> <div data-bbox="282 247 1451 743"> <div>Worksheet Query Builder</div> <pre> 1 CREATE OR REPLACE PACKAGE BODY PKG_FORMAT_EAGLE_DATA 2 IS 3 FUNCTION PARSE_FIRSTNAME 4 (p_fullname supplier.contactname%type) 5 RETURN VARCHAR2 6 IS 7 v_firstname supplier.contactname%type; 8 BEGIN 9 v_firstname := SUBSTR(p_fullname, 1, instr (p_fullname, ' ', 1, 1) -1); 10 RETURN v_firstname; 11 END PARSE_FIRSTNAME; 12 13 END PKG_FORMAT_EAGLE_DATA; </pre> </div> <p>Notice that the package body contains the full implementation details for the PARSE_FIRSTNAME function (unlike the package spec which contains no implementation details).</p>
2f	<p>Test the PARSE_FIRSTNAME functionality of the PKG_FORMAT_EAGLE_DATA package using the contact name of the suppliers to Eagle Electronics.</p> <p>Include both the code and the results in your submission.</p>
2g	<p>Extend the package specification of the PKG_FORMAT_EAGLE_DATA package. Specifically, incorporate a 2nd public function named PARSE_LASTNAME into the package.</p> <ul style="list-style-type: none"> • The function should accept a character input in 'FirstName LastName' format • The input should have the same data type with contact name from supplier table • The function should return ONLY the last name of each contact
2h	<p>Extend the package body of the PKG_FORMAT_EAGLE_DATA package with the implementation details for the PARSE_LASTNAME function.</p>
2i	<p>Using the contact name for each supplier to Eagle Electronics, test your PKG_FORMAT_EAGLE_DATA package.</p> <p>Make certain to test both the PARSE_LASTNAME functionality (to confirm it works) as well as your PARSE_FIRSTNAME functionality (to confirm you have not broken anything).</p> <p>Include both the code and the results in your submission.</p>
<p>Implement Your Own PL/SQL Package</p>	
3	<p>Create a package named PKG_SHAPE_CALCULATION. Include the following modules in this package, each of which is described in the sub-questions.</p> <ul style="list-style-type: none"> • A public function named RECTANGLE_AREA (see part a) • A public function named RECTANGLE_AREA (see part b) • A private constant named c_Pi (see part c) • A public function named CIRCLE_AREA (see part d) • A private function CHECK_VALUE that takes a numeric value and returns a boolean value that indicates whether the input is acceptable (see part e)

3a	<p>A public function named RECTANGLE_AREA that:</p> <ul style="list-style-type: none"> • accepts 2 numeric parameters: length and width • verifies that the length and width are physically possible – use private function CHECK_VALUE • calculates the area of a rectangle of the specified dimensions (if you don't remember the formula, please look it up) • returns a numeric value representing this area • incorporates a general exception handler to gracefully deal with exceptions
3b	<p>A public function named RECTANGLE_AREA that:</p> <ul style="list-style-type: none"> • accepts 1 numeric parameter: length • verifies that the length is physically possible – use private function CHECK_VALUE • calculates the area of a square (e.g., a rectangle having sides of equal length) of the specified dimension (<i>NOTE: the function is named the same as in 3a, keep it this way</i>) • returns a numeric value representing this area • incorporates a general exception handler to gracefully deal with exceptions
3c	<p>A private constant named c_Pi that uses the arccosine to determine the value of π. What is the benefit of making it private?</p>
3d	<p>A public function named CIRCLE_AREA that:</p> <ul style="list-style-type: none"> • accepts 1 numeric parameter: radius • verifies that the radius is physically possible — use private function CHECK_VALUE • calculates the area of a circle having the specified radius • returns a numeric value representing this area • incorporates a general exception handler to gracefully deal with exceptions
3e	<p>A private function named CHECK_VALUE that:</p> <ul style="list-style-type: none"> • accepts 1 numeric parameter which can be used as a length, a width, or a radius • returns a boolean value (True or False) <ul style="list-style-type: none"> ○ True should be returned if the value is above 0 ○ False should be returned if the value is less than or equal to 0 • incorporates a general exception handler to gracefully deal with exceptions
3f	<p>Test each of the modules in the PKG_SHAPE_CALCULATION package at least 3 times. Make certain to include the following cases in the test for each module</p> <ul style="list-style-type: none"> • Positive number(s) • Negative number(s) • Zero(s) <p>Include both the code and the results in your submission.</p>