Lab Submission: 8

Student Name: Michael Daly

Course and Section: CST8215 – 362

========================================================================

Lab 8 – Creating a database function

Always refer to the database documentation for the version of software you are running to be sure of the exact syntax ….

<http://www.postgresqltutorial.com/>

<https://www.postgresql.org/docs/9.5/plpgsql-declarations.html>

CREATE FUNCTION *function\_name* (parameter1 type, parameter2 type)

RETURNS type AS $$

BEGIN

---- logic

END; $$

LANGUAGE **PLPGSQL**

* First, specify the name of the function after the CREATE FUNCTION keywords.
* Then, put a comma-separated list of parameters with their datatype inside the parentheses following the function name.
* Next, specify the return datatype of the function after the RETURNS keyword.
* After that, place the code inside the BEGIN and END block. The function always ends with a semicolon (;) followed by the END keyword.
* Finally, indicate the procedural language of the function e.g., plpgsql in case PL/pgSQL is used.

**Important Note: --- the RETURN type has a another option in the later releases of Postgres…..see the lab exercise. (OUT parameters – these work like the RETURN parameter except you can define multiple ‘out’ values which allow the developer to select which parameter(s) they need for a given program.**

**Grading: Lab 8 is worth 5 marks**

========================================================================

In this Lab, you will create a new ‘function’….

Before beginning…..read this link so you have a better understanding of how ‘parameters’ work.

<http://www.postgresqltutorial.com/plpgsql-function-parameters/>

Using your Query Tool, create the function below using the ‘create function….’

CREATE FUNCTION invoice\_details(prod\_cost numeric(8,2),

quantity int,

sales\_tax numeric(4,2),

OUT Prod\_cost\_by\_quantity numeric(8,2),

OUT tax\_charged numeric(8,2),

OUT invoice\_total numeric(8,2))

AS $$

BEGIN

Prod\_cost\_by\_quantity := prod\_cost \* quantity;

Tax\_charged := prod\_cost\_by\_quantity \* sales\_tax;

Invoice\_total := prod\_cost\_by\_quantity + tax\_charged;

END;$$

LANGUAGE plpgsql;

Now, display the values stored in the ‘out’ parameters created by your function. You will notice that there are four different ‘select’ statements that call upon your new function ---- look at the differences.

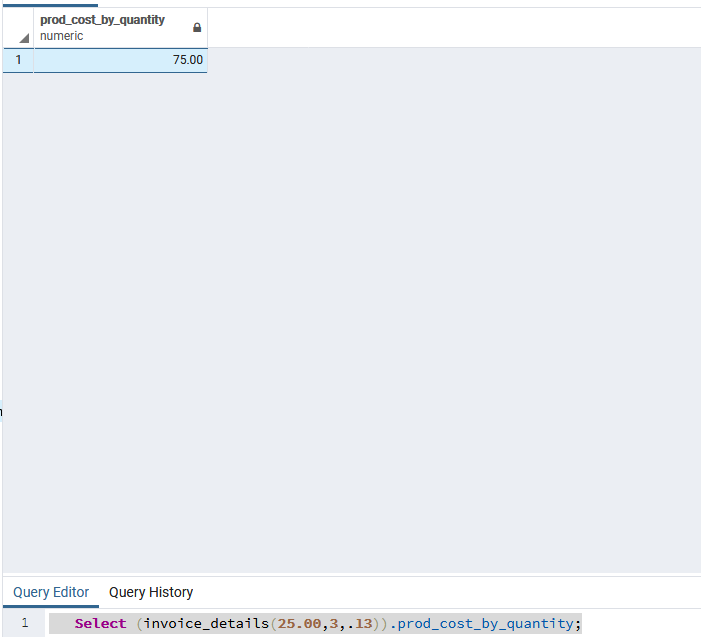
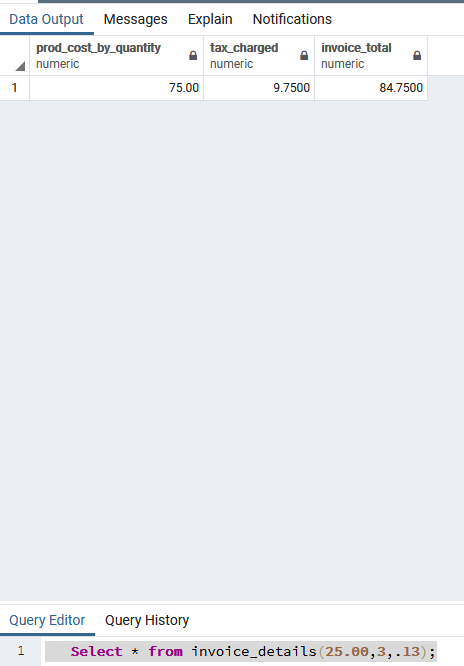
Select \* from invoice\_details(25.00,3,.13);

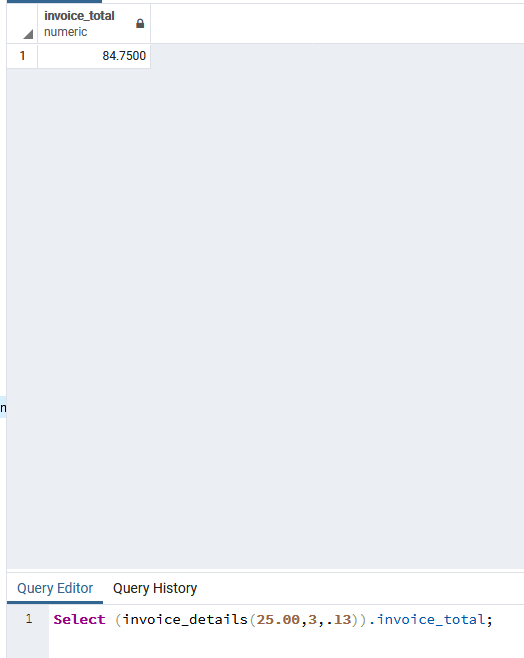
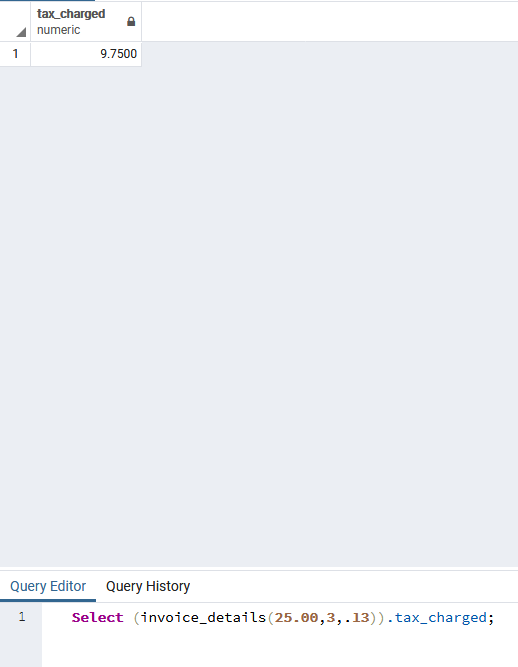
Select (invoice\_details(25.00,3,.13)).prod\_cost\_by\_quantity;

Select (invoice\_details(25.00,3,.13)).tax\_charged;

Select (invoice\_details(25.00,3,.13)).invoice\_total;

*Provide a screen shot showing each query and its result set*





Using your own words, describe an example of why you, as a developer, would create database functions.

*Type your response here*

As a developer anytime I need a specific set of data about the metadata I would use a function. They follow the same similar definitions of a function in any coding language except they can return but not alter. So if I need a set of data from the database but would not want to alter the database, or have to write repeating query's(or have a database object for each specific return I need) I would create a function and just feed in the values I need.

If I needed to find the net sales of an item based on price and quantity, I would write a function with the two as the parameters. I could use that function when presenting data to show all orders of items, or each item specifically.