**COMP421 Database Systems Project 3**

**Group 27**

**Zhong Guan 260758587**

**Runze Zhang**

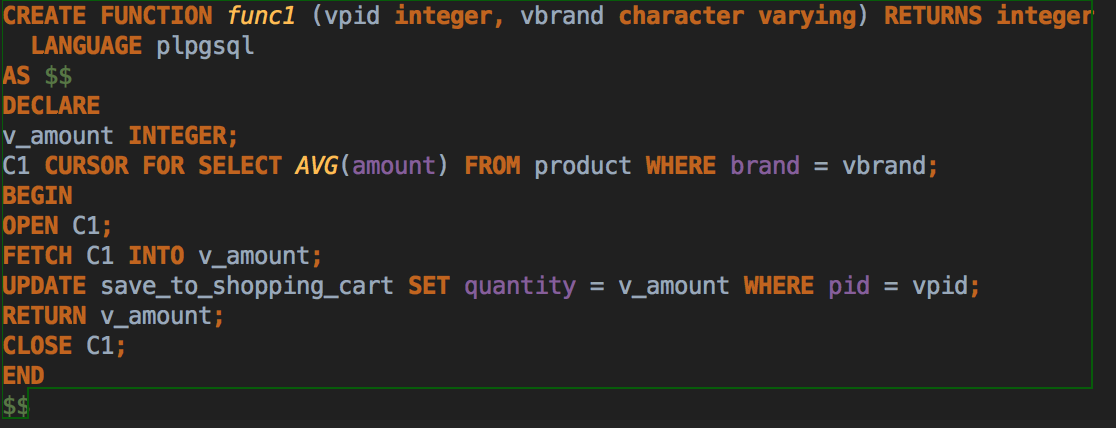
**Yining Zhou**

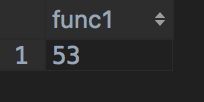
1. Stored Procedure

The stored procedure we designed is to compute the average amount of a brand in the *product* table and then use this value to modify the quantity of a product in *save\_to\_shopping\_cart* table. Both brand and product to be modified are all input provided by user.

Besides the stored procedure, we also created a trigger associated with this procedure. Whenever the quantity of a product in the *save\_to\_shopping\_cart* table changes, we will log it into a separate table named *shoppingcart\_audits* through this trigger.

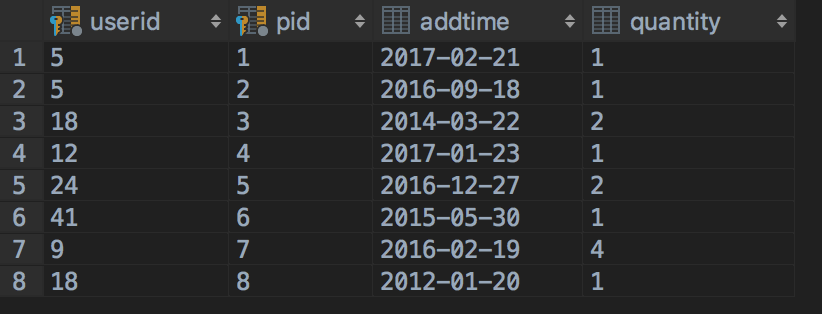
The stored procedure is as follows:



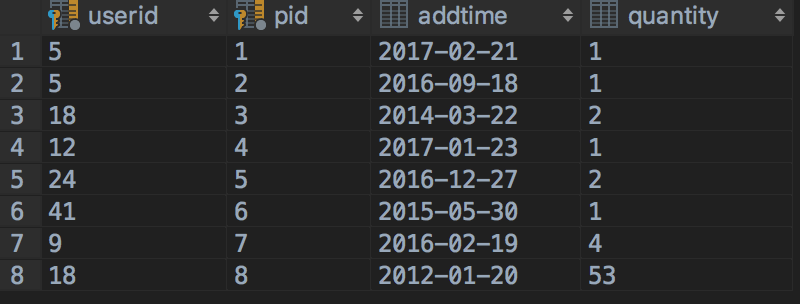
When we execute the function by /Users/apple/Desktop/Stored Procedure/Snip20170322_6.png, it will return the average amount of product of bran Microsoft in the shopping cart: 

If we query the relation *save\_to\_shopping\_cart* by using /Users/apple/Desktop/Stored Procedure/Snip20170322_8.png before and after the stored procedure was executed, we can see its difference.

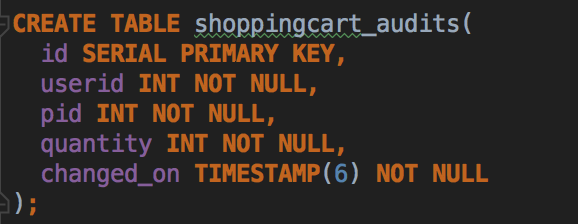
*Before*



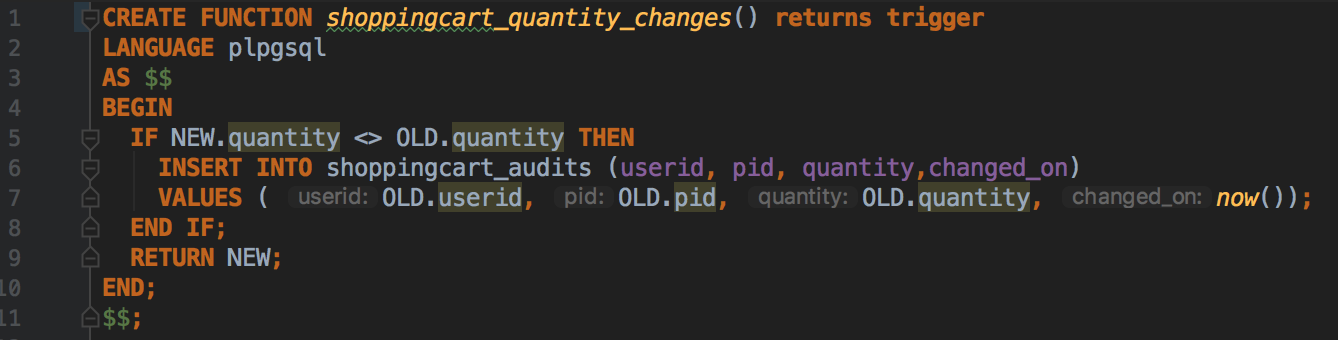
*After*



Before we define the trigger function, we created a table named *shoppingcart\_audits* to store the changes.

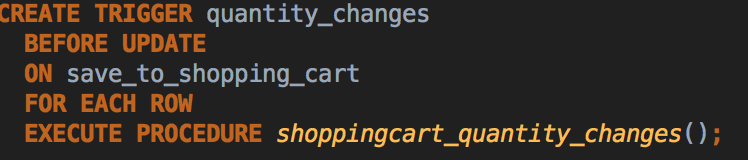


The trigger function is as follows:

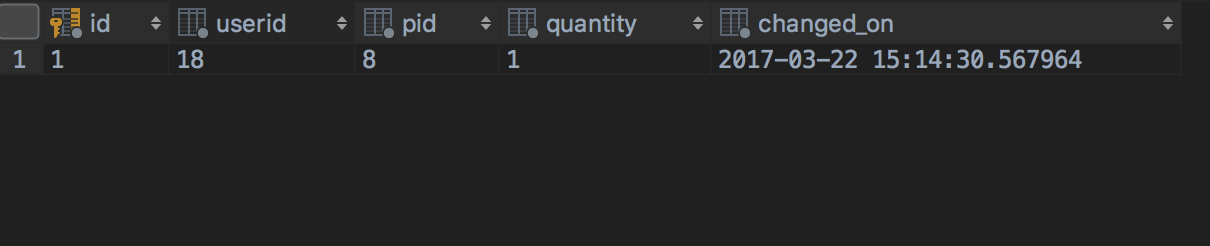


The function checks if the quantity of product in the shopping cart changes, it will insert the old quantity into the *shoppingcart\_audits* that including userid, pid and the time of change.

Second, we bind the trigger function to the *save\_to\_shopping\_cart* table. The trigger name is *quantity\_changes*. Before the value of the quantity column is updated, the trigger function is automatically invoked into log the changes.



Third, we can use the stored procedure for test.



Then, we update the quantity again, the result indicates the trigger function works successfully.

/Users/apple/Desktop/Snip20170322_18.png

