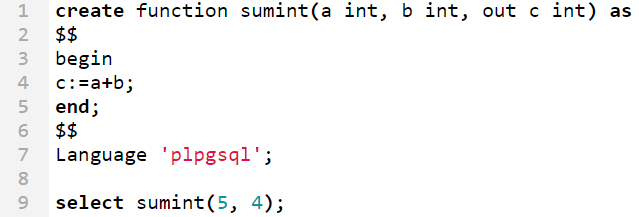
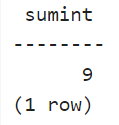
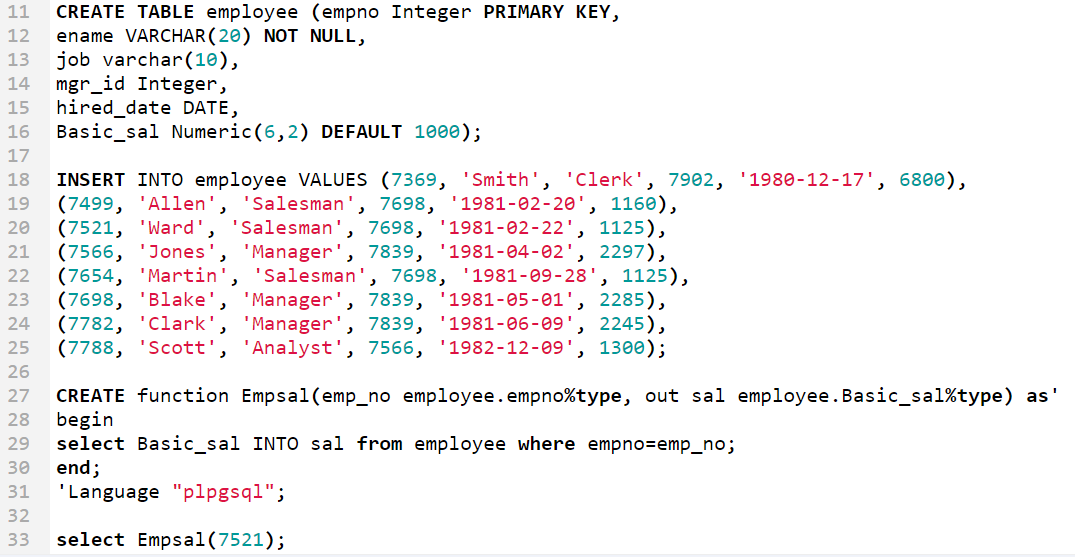
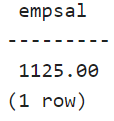
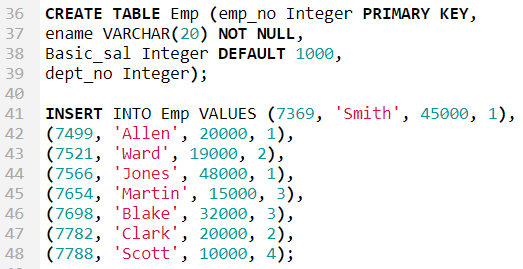
**database mangement systems**

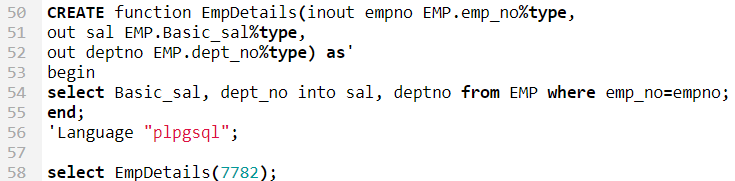
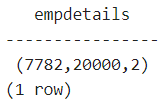
# **PLSQL Lab1**

## Write a function that accepts two integers as inputs and returns the sum of integers

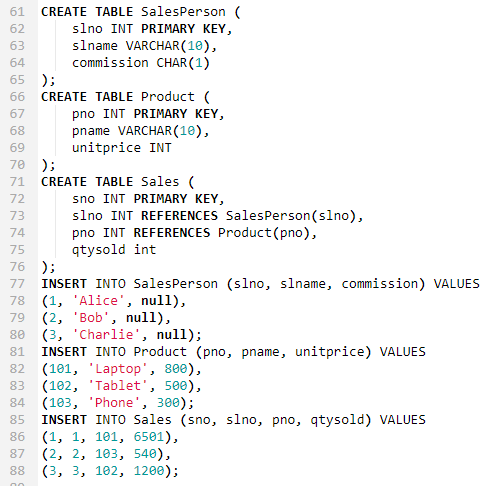


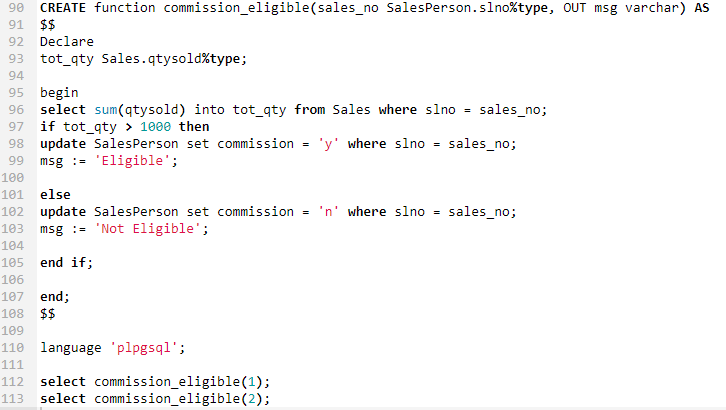


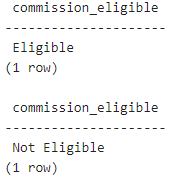
1. Write a function that accepts the employee no as input and returns the salary of that employee  
     
   
2. Write a function that accepts the employee no of an employee and returns the salary and department no as output  
   

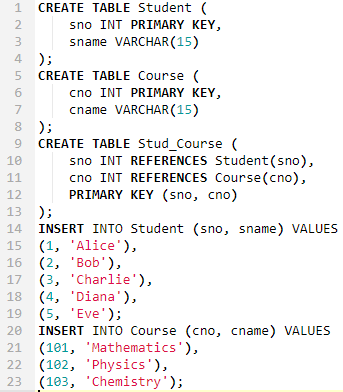
1. Write a function that accepts sales person id as input and check if that sales person is eligible for commission, eligibility criteria for commission is that the salesperson must have sold more than 1000 products combined all products together. If eligible, update the commission entry to Y.

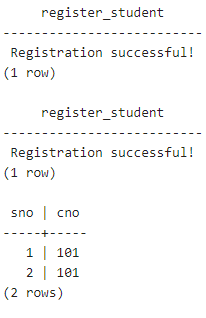
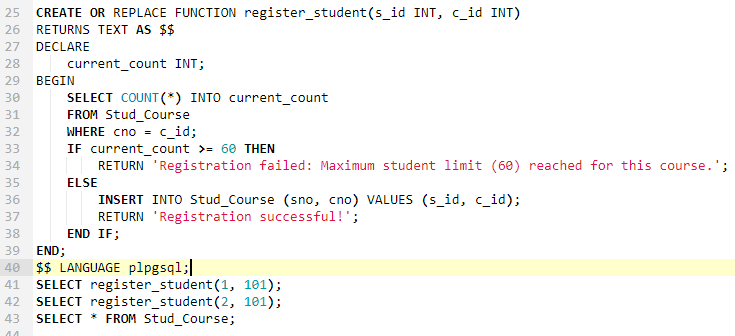






1. For all courses maximum sixty students can be registered write a function to register a student for a particular course only if current Number of students registered for that course is not exceeding the limit.

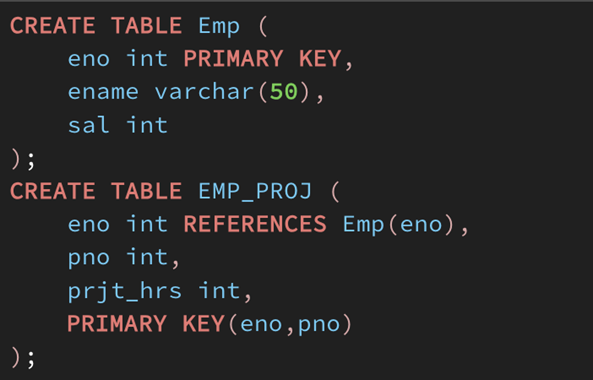


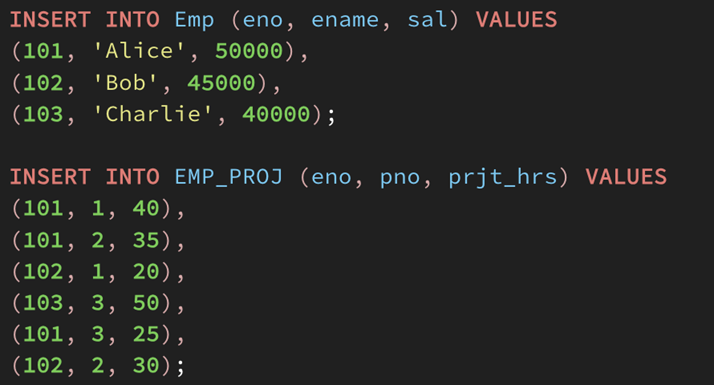
  
**Question 2**

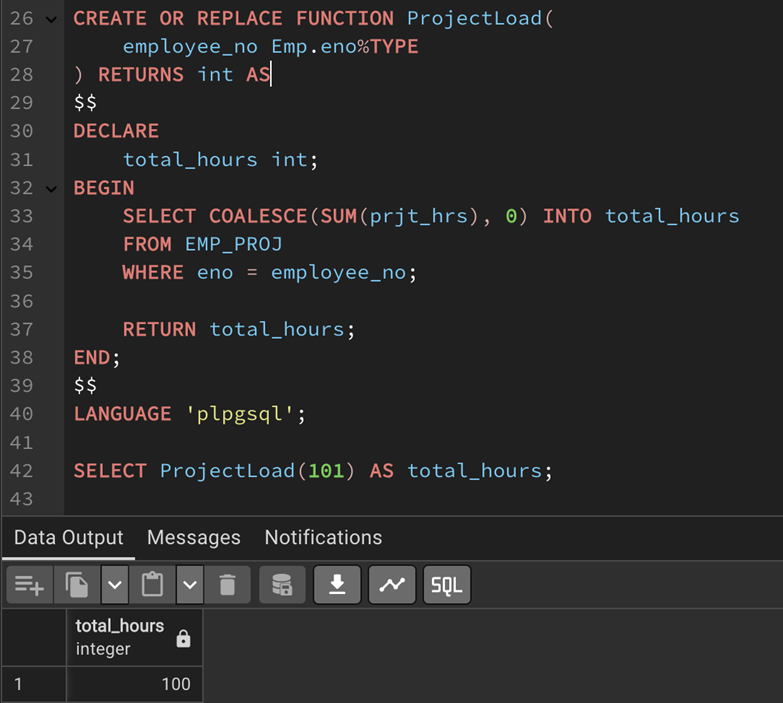
Consider the following schema .

**Emp(eno, ename, sal)**

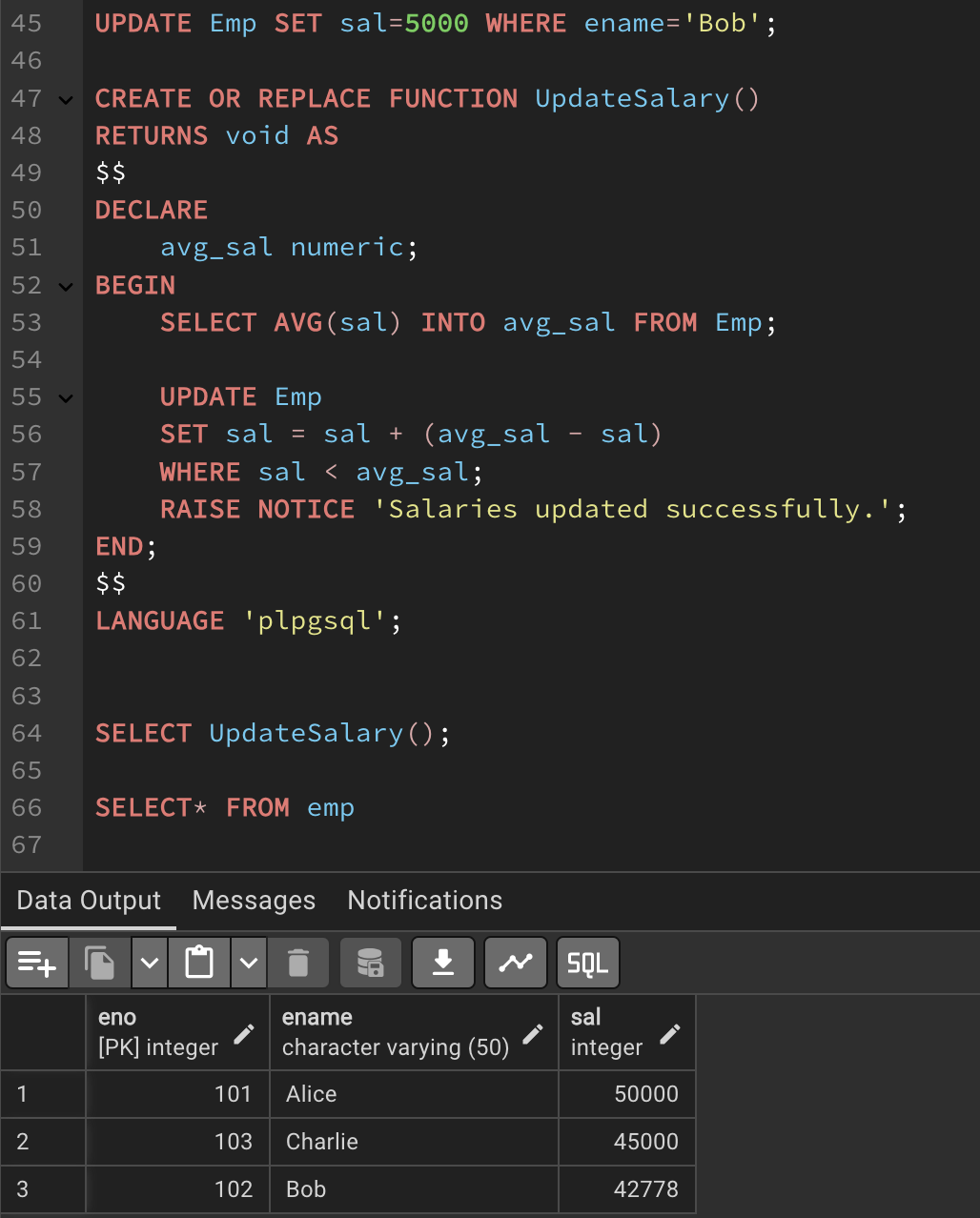
**EMP\_PROJ(eno, pno, prjt\_hrs).**





1. Write a function ProjectLoad that returns the total project working hours for the given eno. 
2. Write a PL/SQL code to update the salary of an employee if the employee earn less than the average salary.

New salary is current sal + difference between currrent sal and average salary

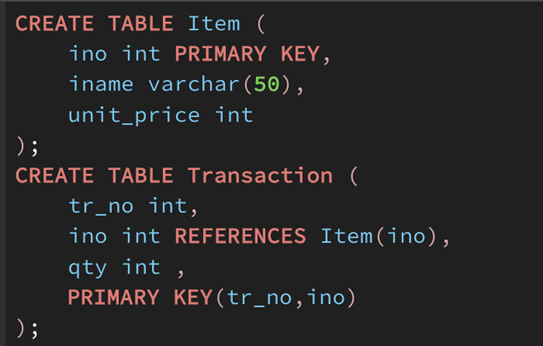


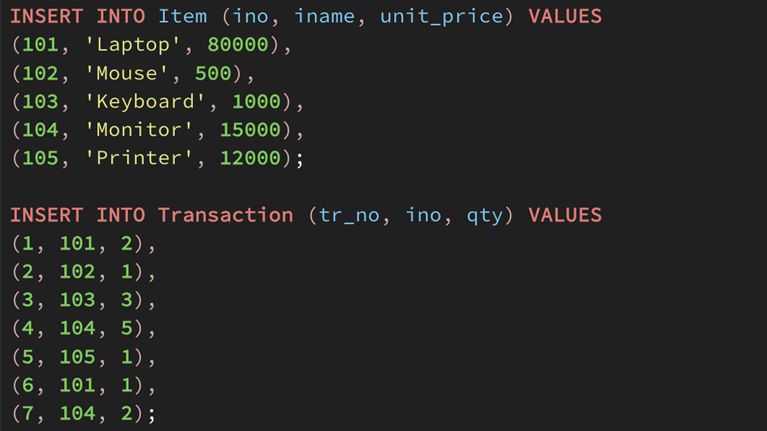
**Question 3**

Consider the following tables

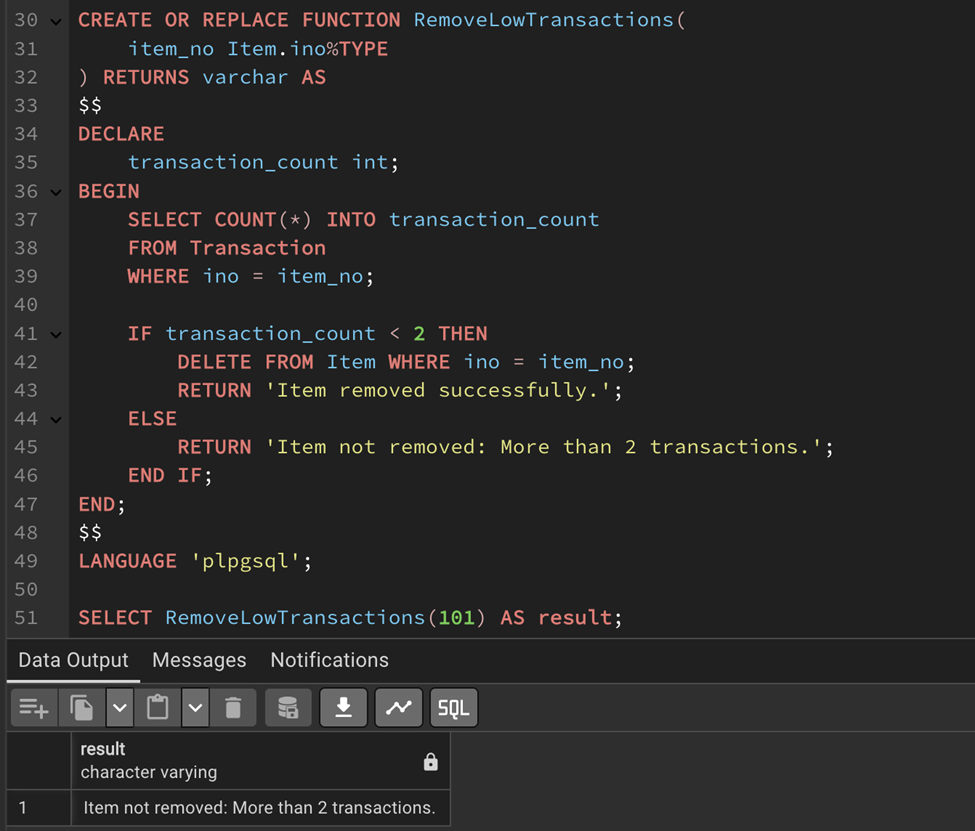
**Item(ino, iname, unit\_price)**

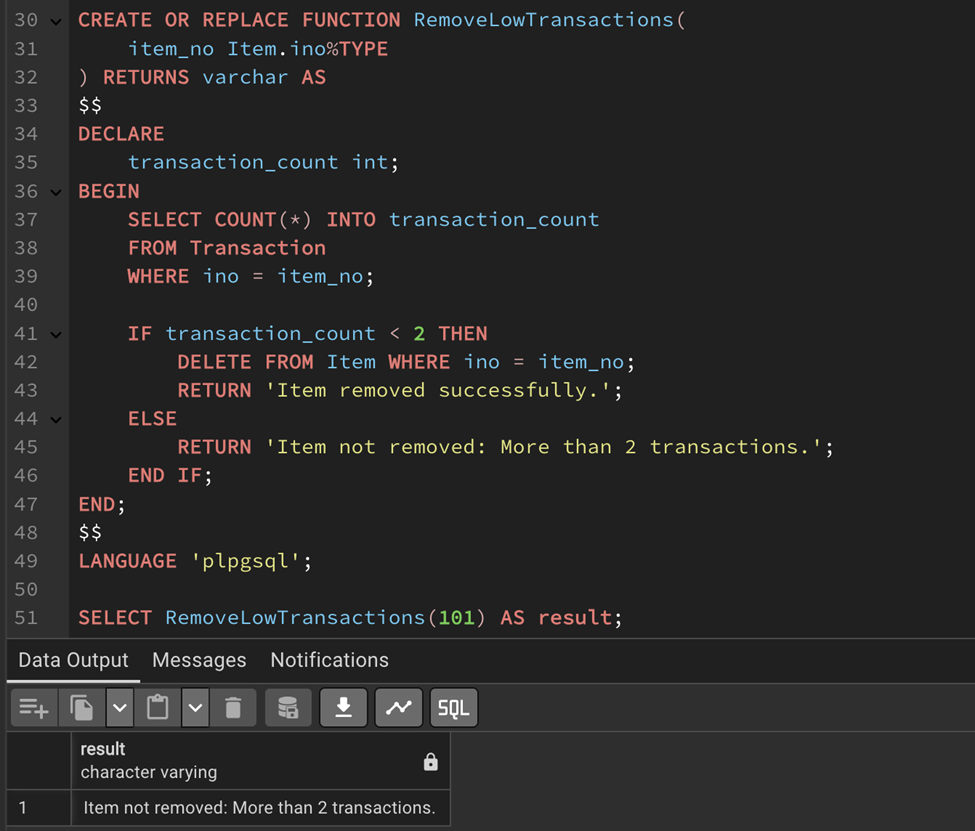
**Transaction(tr\_no,ino,qty)**

****

****

Write a function to accept an item no from the user. If transaction has been made for less than 2 times for that item (check from transaction table), delete the item from the Item table.





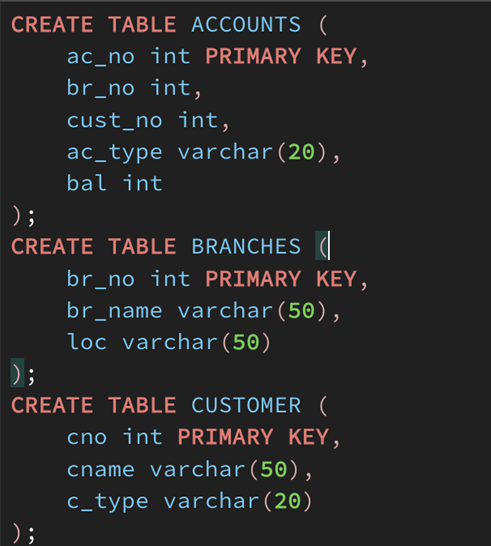
**Question 4**

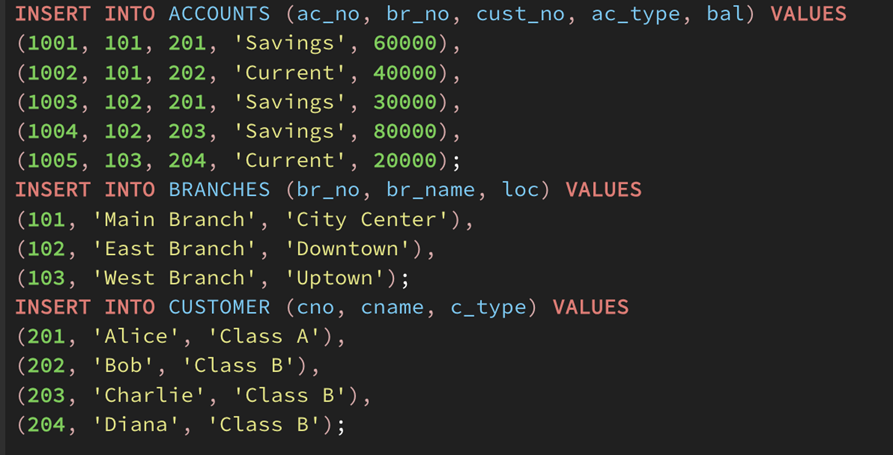
Consider a Bank database which includes the following tables.

**ACCOUNTS(ac\_no,br\_no, cust\_no,ac\_type,bal)**

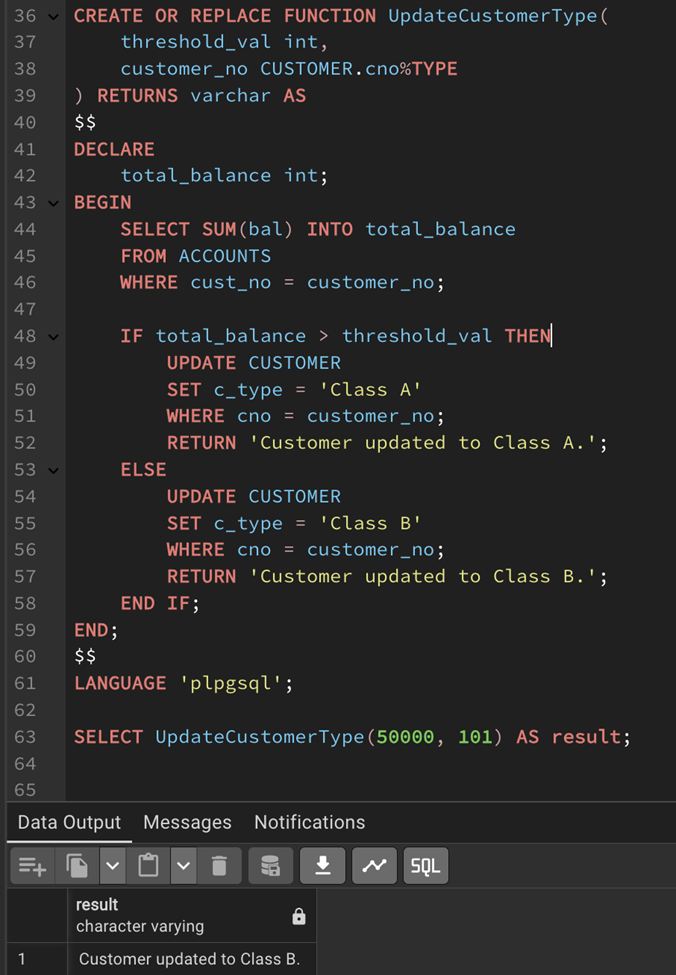
**BRANCHES(br\_no, br\_name, loc)**

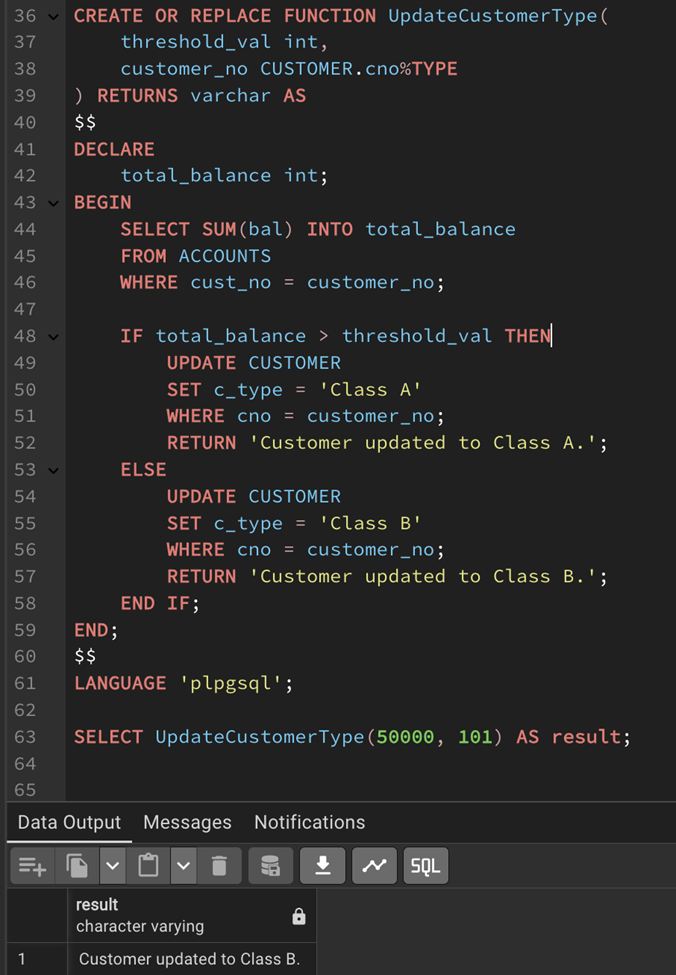
**CUSTOMER(cno, cname, c\_type)**

****

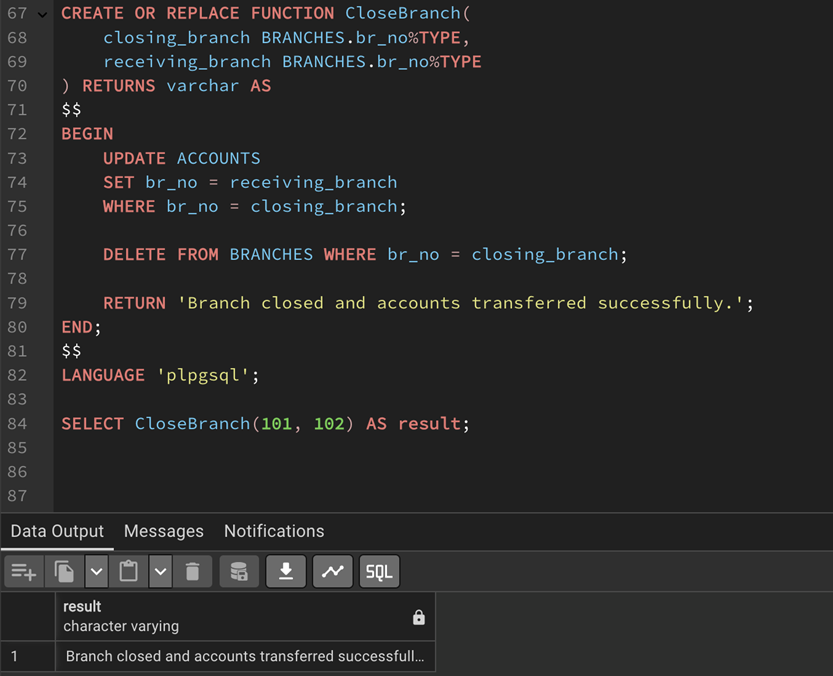
****

1. Write a function that accepts a threshold value and a customer number. The program updates the c\_type based on the threshold value. Ie. If balance > threshold then class A, else class B.





1. Write a function called CloseBranch that takes two arguments (the branch to be closed and the branch to take over the accounts) and transfers all accounts at the closing branch to the new branch and removes the closing branch.



1. Write a function that implements a "safe" withdrawal operation, that only permits a withdraw if there are sufficient funds in the account to cover it.

