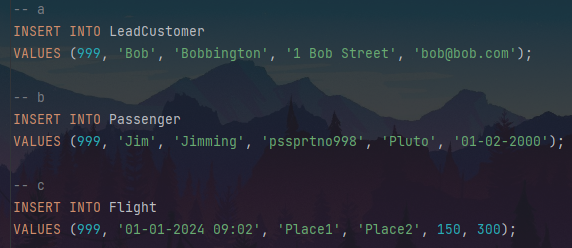
**Transactions of Interest**

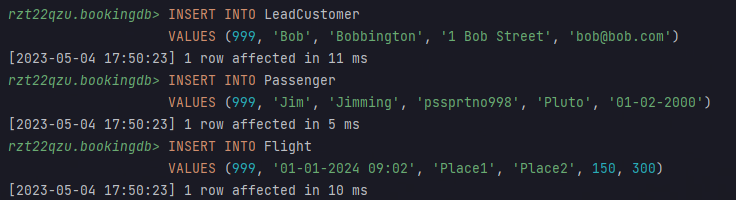
**A. Insert a new record. This could be**

**a. Given a lead customer ID number, name, and contact details, create a**  
**new customer record.**  
**b. Given a passenger with an ID, name, date of birth, etc., create a new**  
**passenger record.**  
**c. Given a flight ID number, origin, destination, flight date, capacity of the**  
**aircraft, and price per seat create a new flight record.**

SQL:

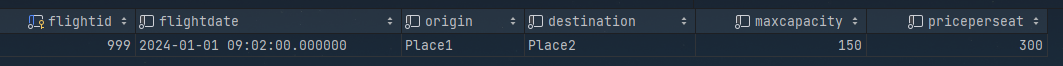


Output:

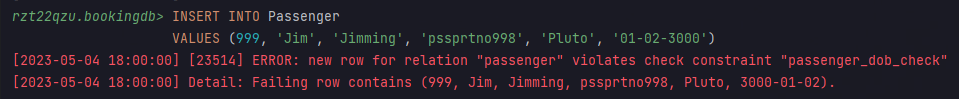


Result:

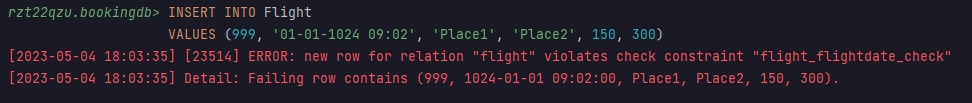
  

Erroneous testing: DoB in the future:

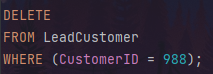


Erroneous testing: Flight in the past:

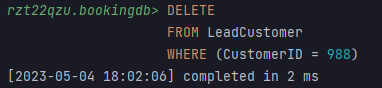


**B. Given a customer ID number, remove the record for that customer. It should**  
**not be possible to remove customers that have active (i.e., reserved) flight**  
**bookings. A customer that has only cancelled bookings could be removed; the**  
**associated bookings should also be removed along with all the seat bookings.**

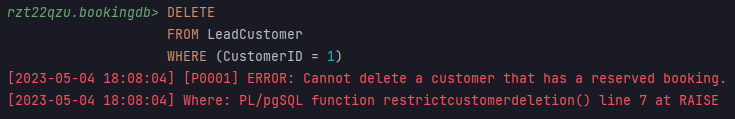
SQL:



Output:

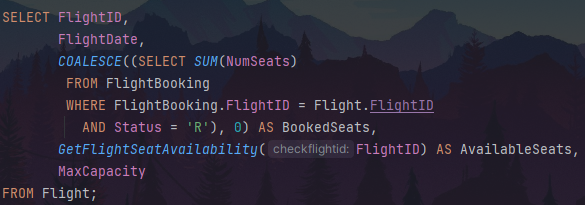


Erroneous testing: Delete customer with reserved booking:

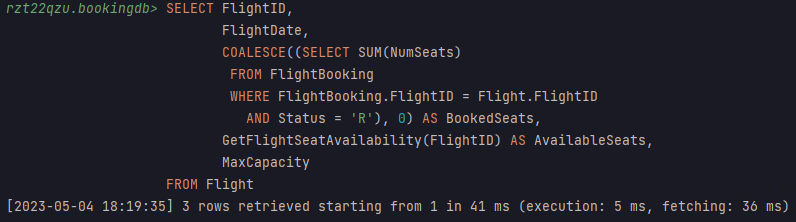


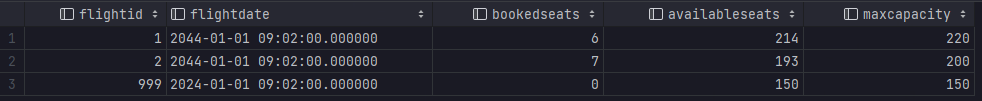
**C. Check the availability of seats on all flights by showing the flight ID number,**  
**flight date along with the number of booked seats, number of available seats and**  
**maximum capacity.**

SQL:



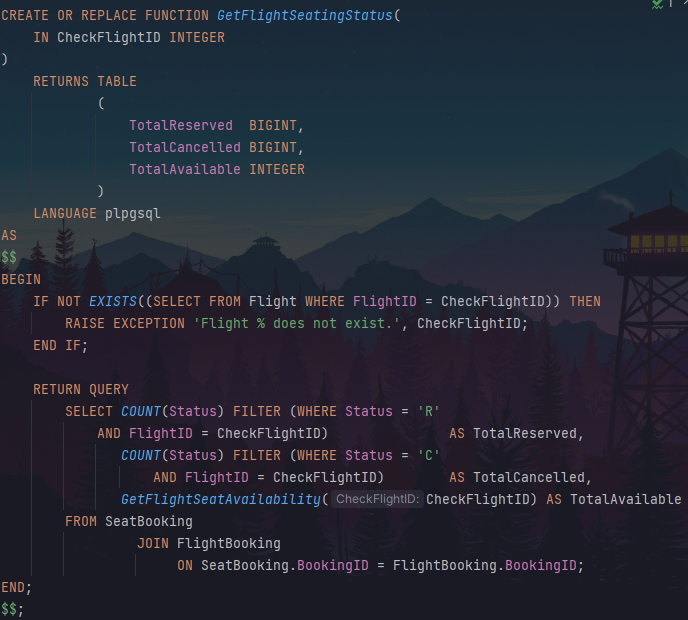
Output:





**D. Given a flight ID number, check the status of all seats currently allocated to**  
**that flight, i.e., return the total number of reserved/ cancelled/ available seats.**

SQL:



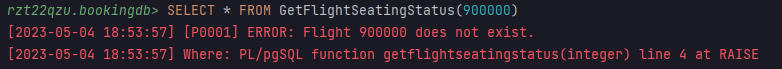


Output:



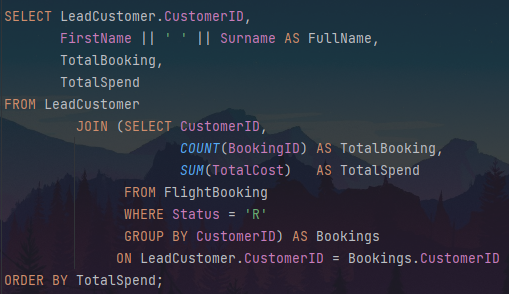


Erroneous testing: Check a flight ID that doesn’t exist

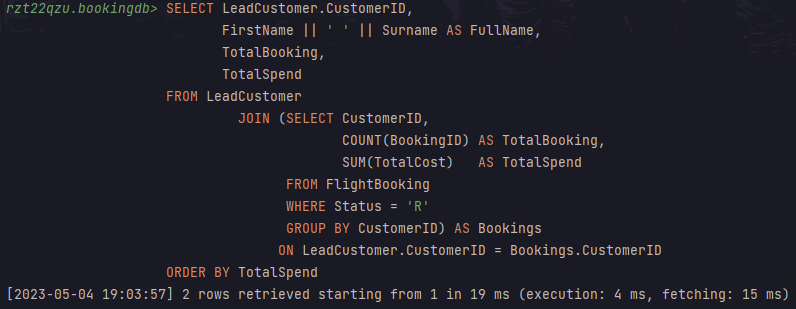


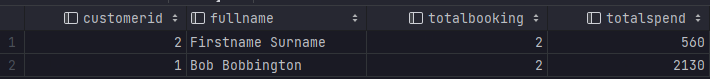
**E. Produce a ranked list of all lead customers, showing their ID, their full name,**  
**the total number of bookings made, and the total** **spend made for all bookings.**  
**The list should be sorted by decreasing total value.**

SQL:



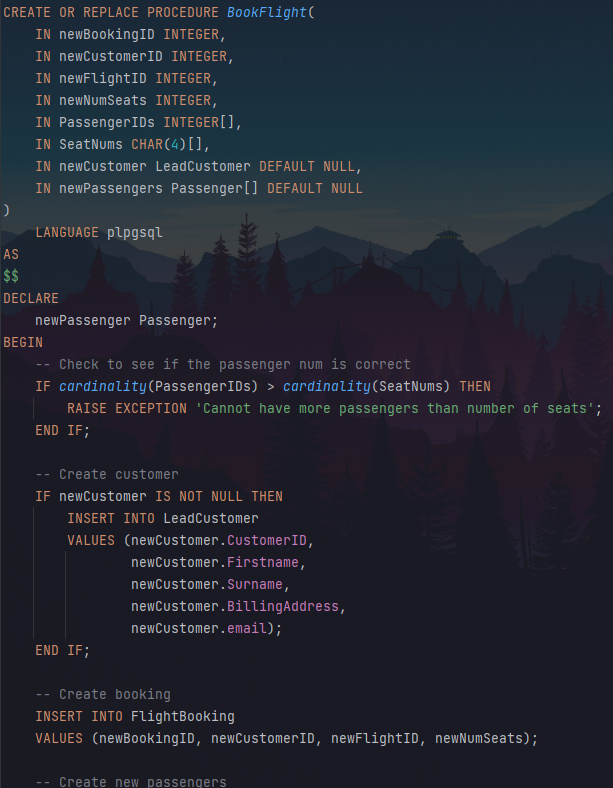
Output:

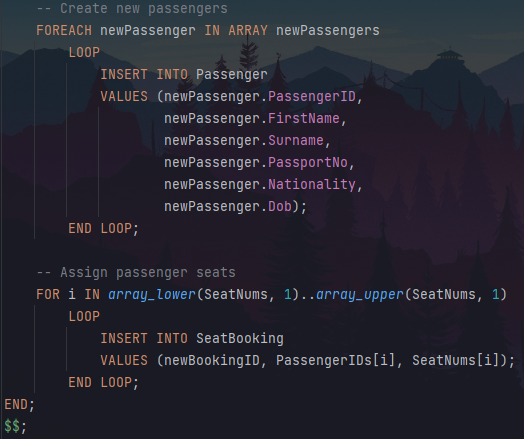


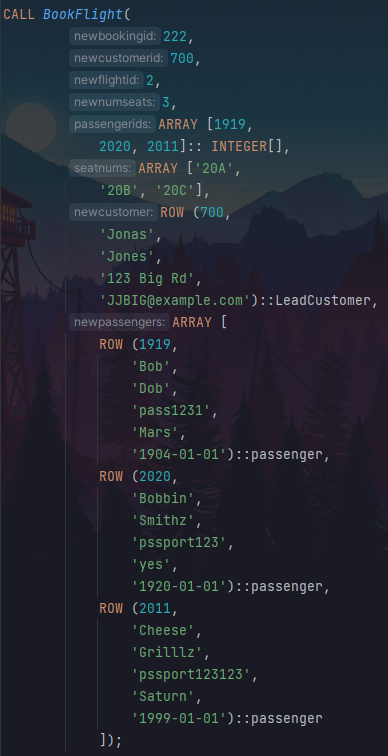


**F. Given a booking ID, customer ID number, flight ID number, number of seats**  
**required and passenger details, make a booking for a given flight. This procedure**  
**should first show seats available in a given flight and then proceed to insert**  
**booking, if there are sufficient seats available. The customer could be an existing**  
**customer or a new customer, in which case it should be entered first into the**  
**database. Seats numbers can be allocated at the time of booking or later on.**  
**The making of a booking with all the steps outlined should work as an atomic**  
**operation.**

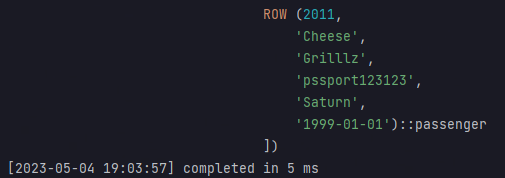
SQL:

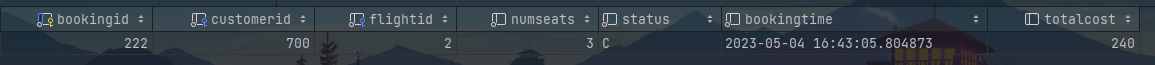




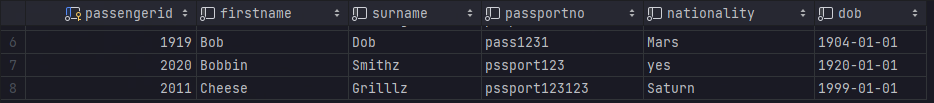


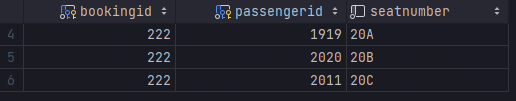
Output:







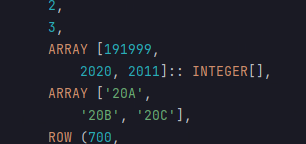




Erroneous testing: Provide too many seat numbers:  



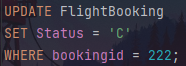

Erroneous testing: Provide a non-existent passenger ID





**G. Given a booking ID number, cancel the booking. Note that cancelling a**  
**booking only changes the status and should not delete the historical details of the**  
**original booking. However, cancelled seats should be viewed as available.**

SQL:



Output:

