**ECE 656 – Winter 2015 Assignment #2**

Zeya Liu(20551072),Anqi Yao (20557257), Hanning Zhu (20564432),Yuantao Ji (20511320)

**Part 1. SQL DDL and Advanced SQL Applications**

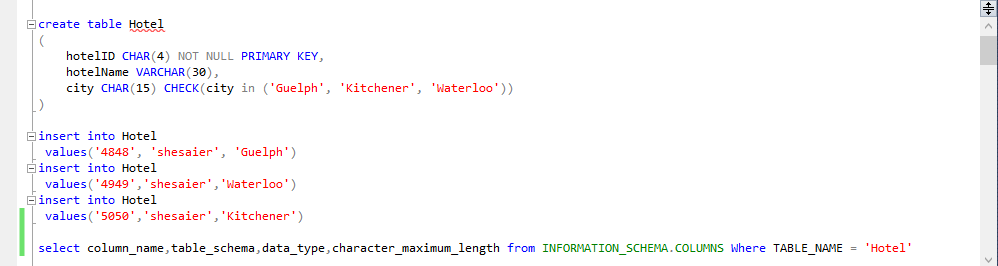
In this part, we used Microsoft SQL Server as implementation of the DBMS.

Firstly, we created a database named HotelDB.

C:\Users\Hannah\Desktop\creat table\creat db.png

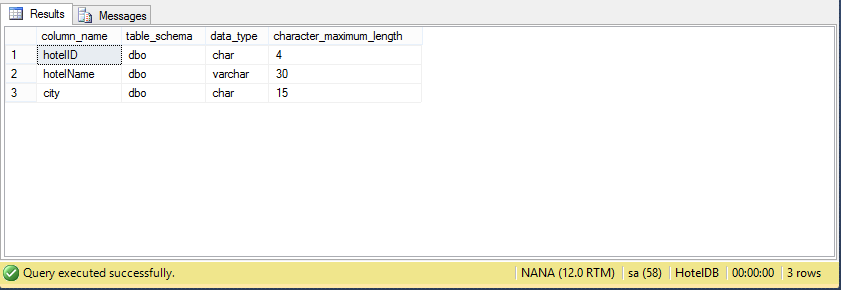
**The tables we created are as below:**

1. **Table Hotel**

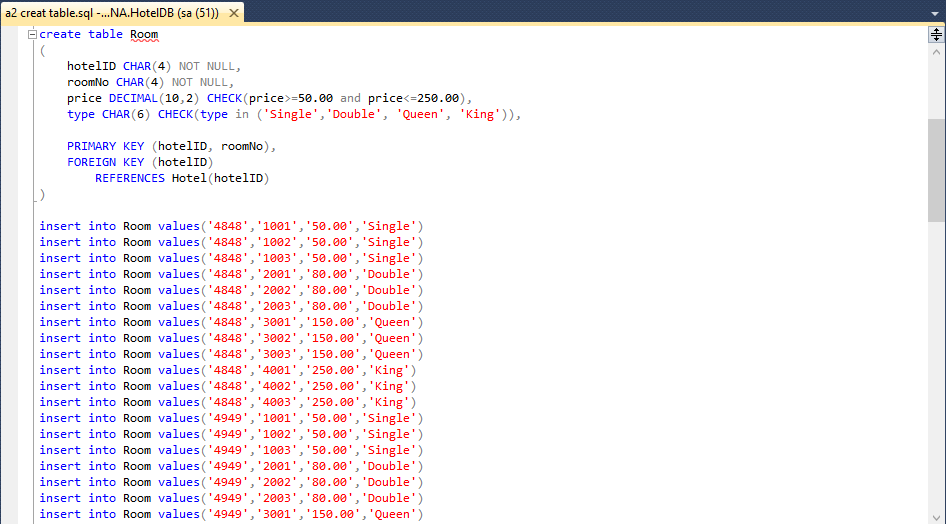


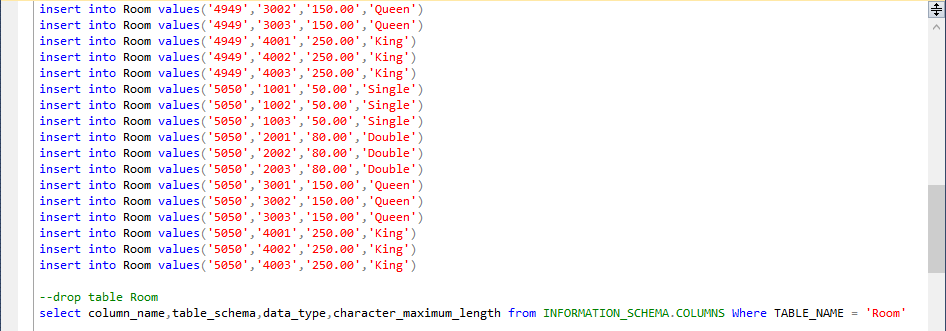
The parameters of this table: hotelID, hotelName, city. The primary key: hotelID.

The Description output:



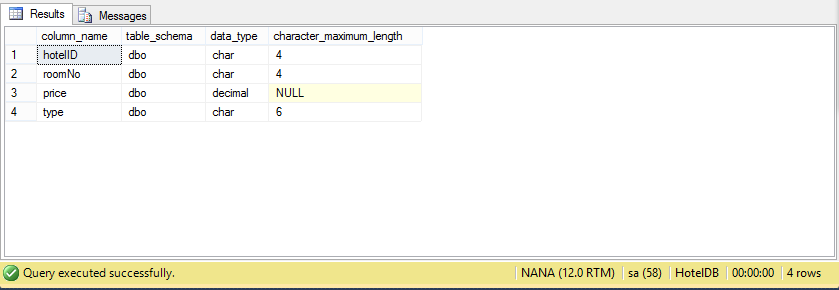
1. **Table Room**



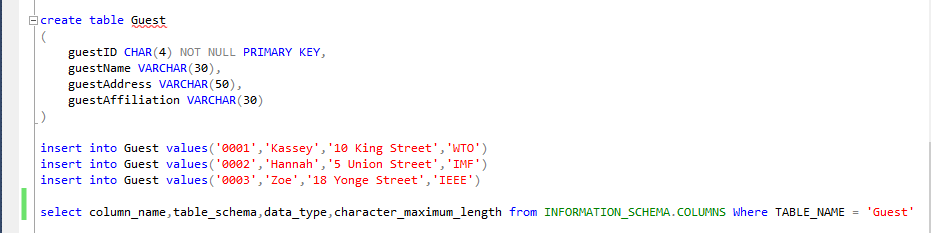


The parameters of this table: hotelID, roomNo, type, price. The primary key: hotelID, roomNo.

The description output:

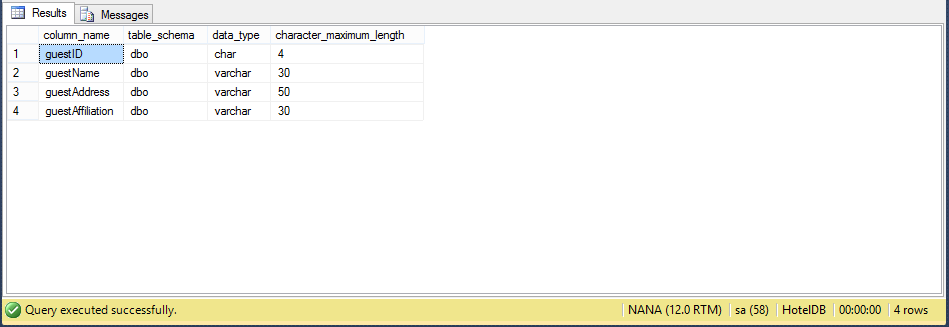


1. **Table Guest**

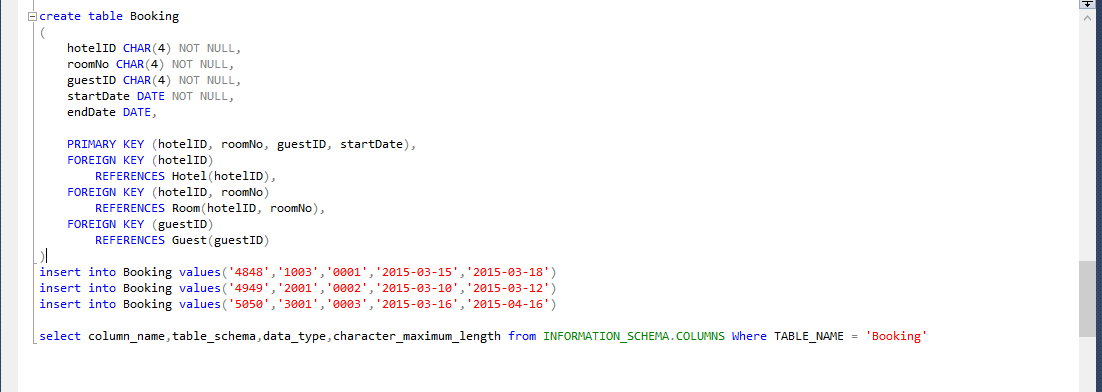


The parameters of this table: guestID, guestName, guestAddress, guestAffiliation. The primary key: guestID.

The description output:

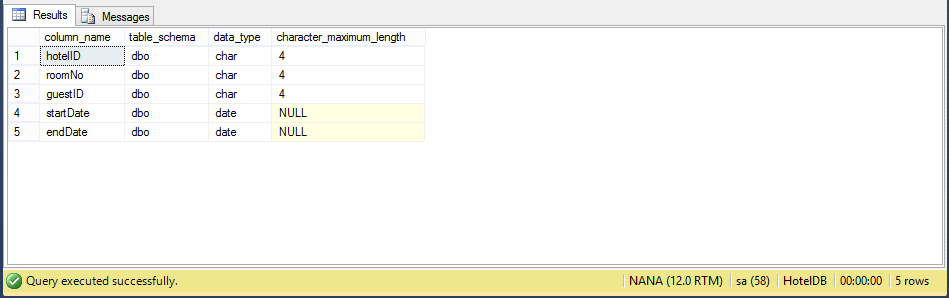


1. **Table Booking**



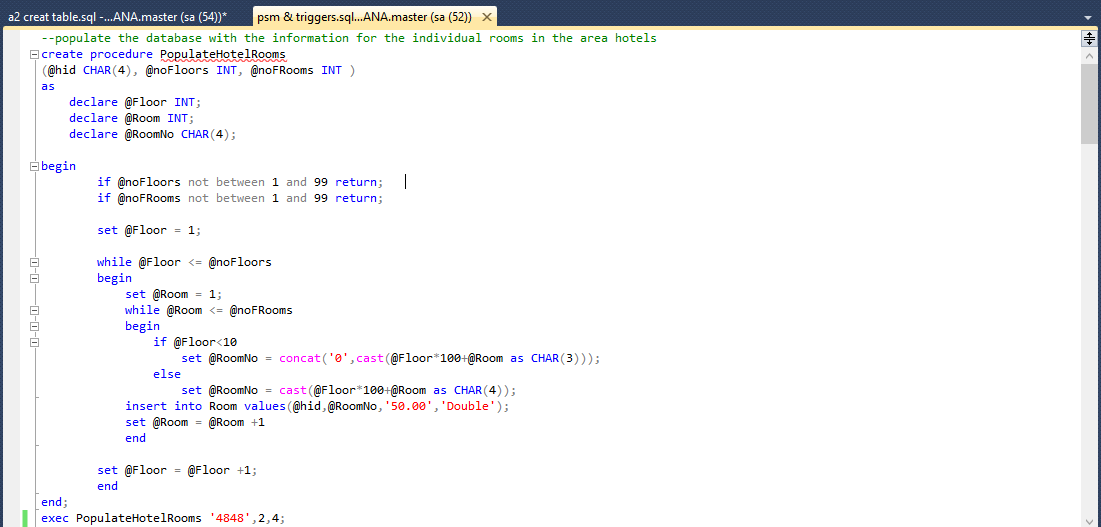
The parameters of this table: hotelID, roomNo, guestID, startDate, endDate. The primary key: hotelID, roomNo, guestID, startDate.

The description output:



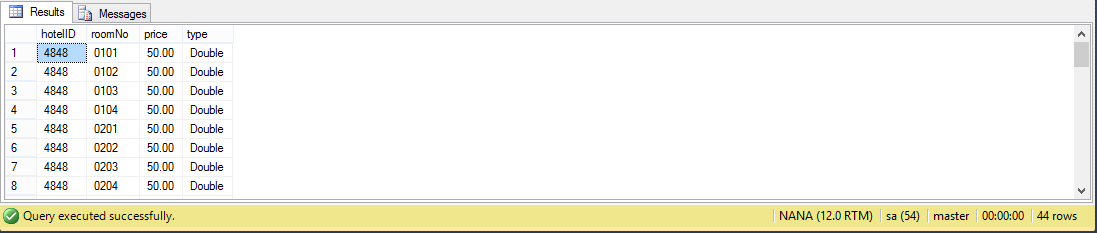
**Step 3:**

1. **PSM**



In this PSM PROCEDURE, we can insert the information for any individual rooms in certain hotel.

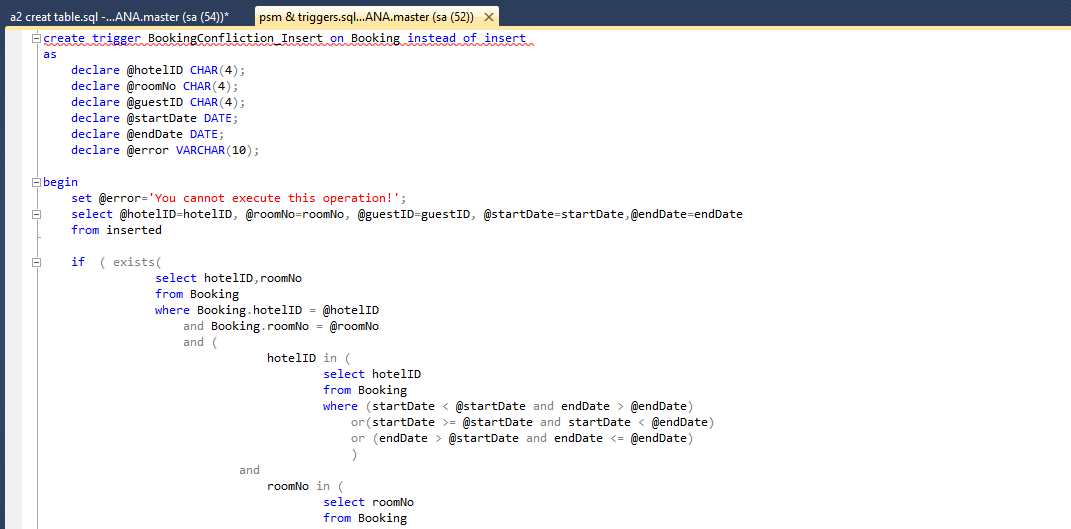
Using the input ‘4848’, 2, 4 as an example (where ‘4848’ is hotelID, 2 is number of floors, 4 is number of rooms per each floor), the output showed as below:

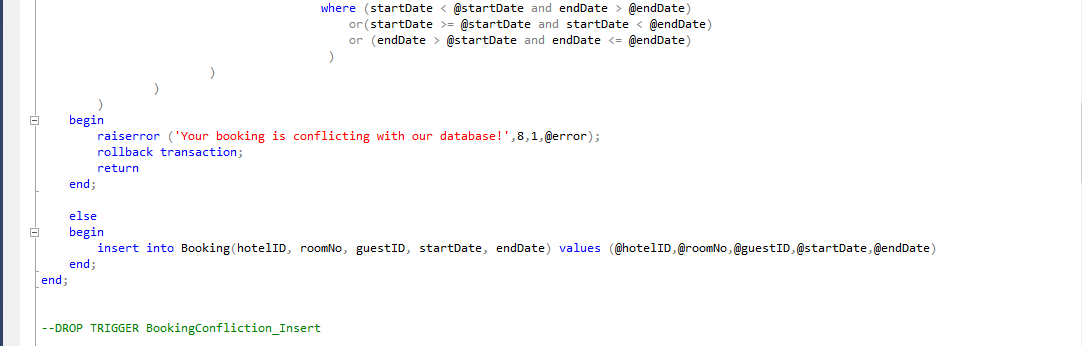


According to this output, after executed the PSM PROCEDURE, all expected rooms were added to database with the specified format.

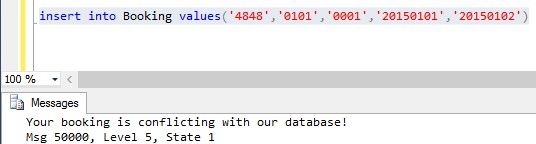
1. **TRIGGERS**

The **first trigger** we created is using for figure out the conflicts in booking function, which means when we want to book an unavailable room in this system, the system will produces an error message prompt user cannot do such operation as below:

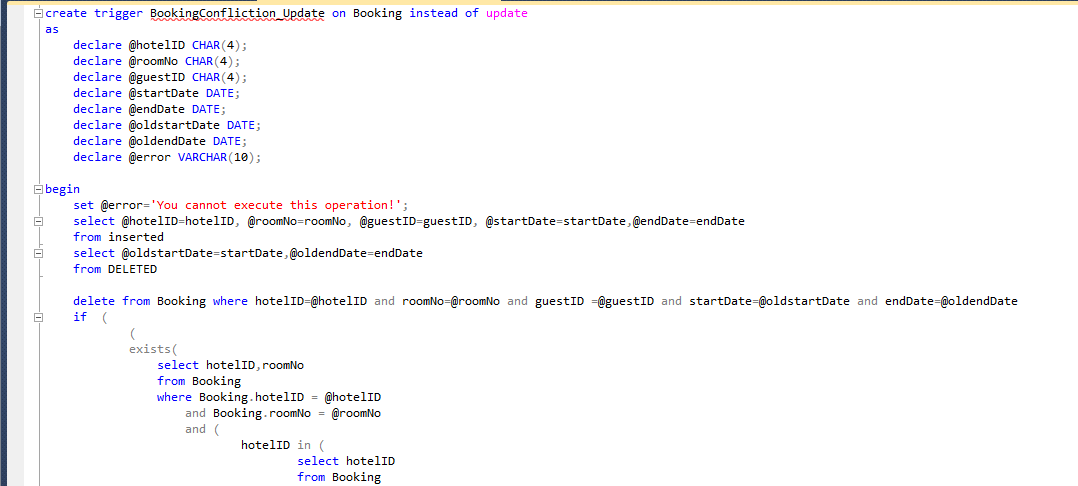
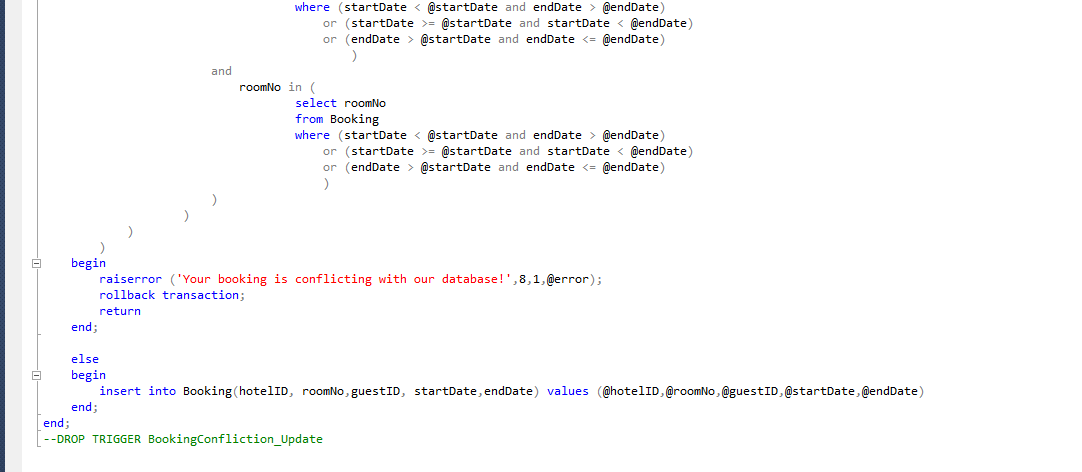




This trigger is aimed to avoid that when new booking is conflicting with exist one. The prompt message when we try to insert an existing booking message is as below:

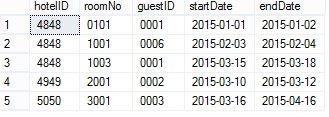
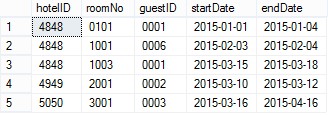


The **second trigger** we make aims to change booing information in the database system. When user want to change information for instance change the start booking date, this part will execute such operation as below:

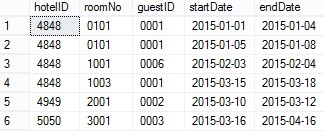
For instance, in our existed database, we already have a booking in the period: 2015-01-01 to 2015-01-02, then we update our table using the command code:” UPDATE Booking set endDate='20150104' where hotelID='4848' and roomNo='0101'”, which change the first record, the changing output is as below:

The original record is: The update record is:

If we update a new record is conflict with rest several data records, there will produces an error message prompt user the booking operation is conflict with the database system, the output is as below.

The command code is” UPDATE Booking set endDate='20150107' where hotelID='4848' and roomNo='0101' and startDate='20150101'”, the exist table is



The system output is:



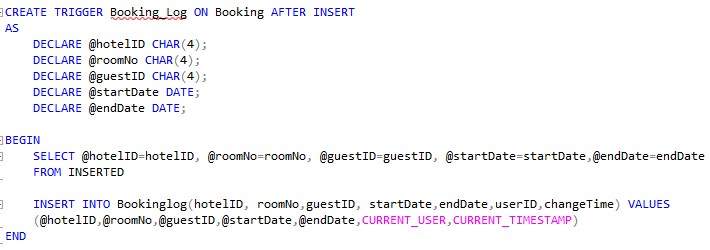
Which shows the changing information operation cannot be executed.

1. **Maintenance table**

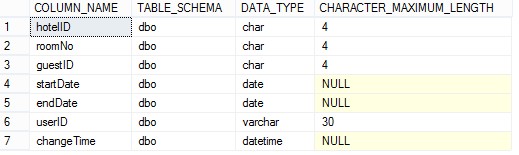
This table aims to record every change added to the table Booking, the source code is as follow:



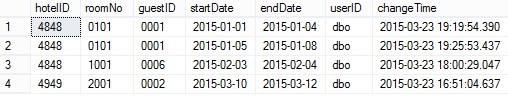
The trigger source code is:



The description output is:



Based on the previous operation, the changed process were record as follow:



**Part 2. SQL API Applications**

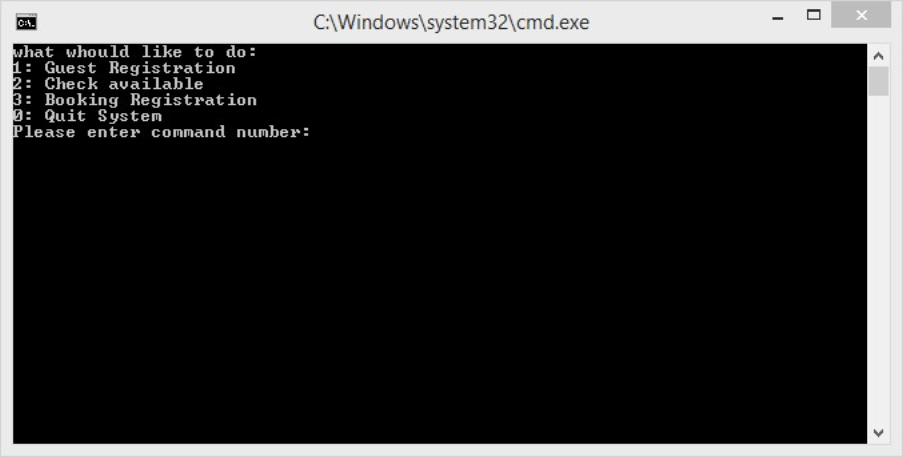
In this SQL API Applications we have three main functions to achieve the requirement. We have one main class named hotel system and four functionality classes, which are: the database\_connection class, guest class, check\_available class and booking\_registration class.

The source codes and their operation windows are as below:

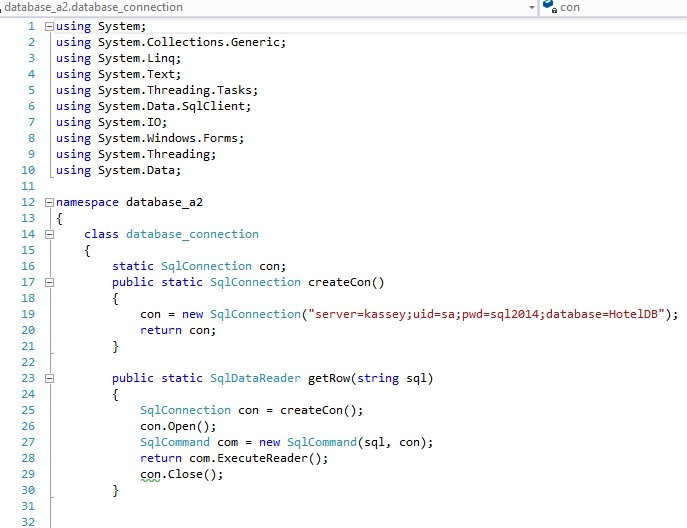
1. **The main class hotel system:**



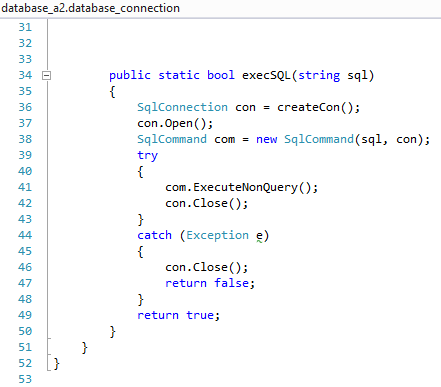
The operation window:



1. **database\_connection class:**



In this static we connect our c# project to our SQL Sever database management system, and we can also get the information from the database system.

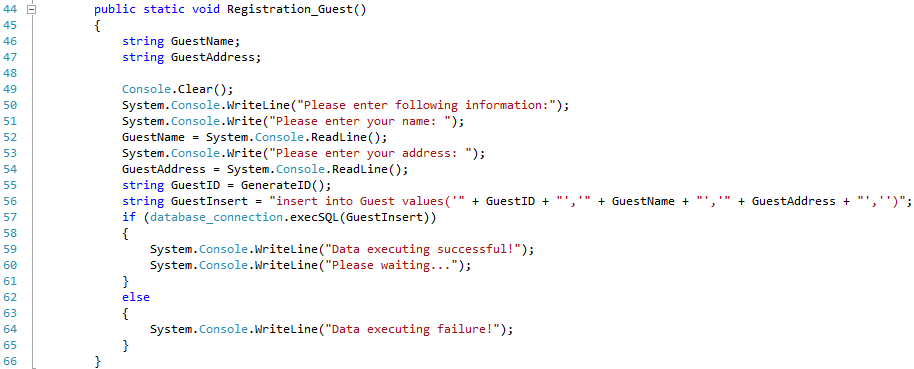


In the second static, we execute the SQL in our program.

1. **guest \_registration class:**

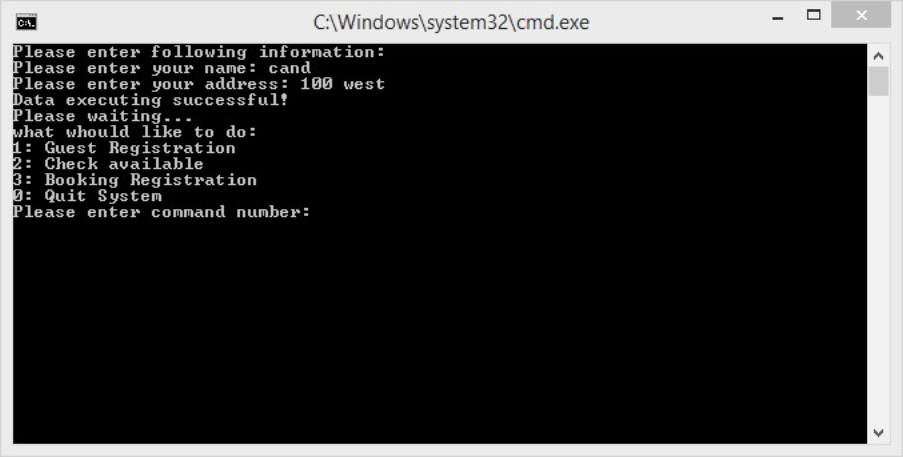


In this static we generate the guest ID by get the amount of original data form database, we use the amount of the record plus 1 to generate the new guest ID, then make the ID to the standard format like “0001” to “9999” with a fixed-length string of 4 characters.



After generate the new guest ID for the customer, we insert the information include guest ID, guest name, and guest address to our database system for the table guest.

The operation window of this function is as follow:



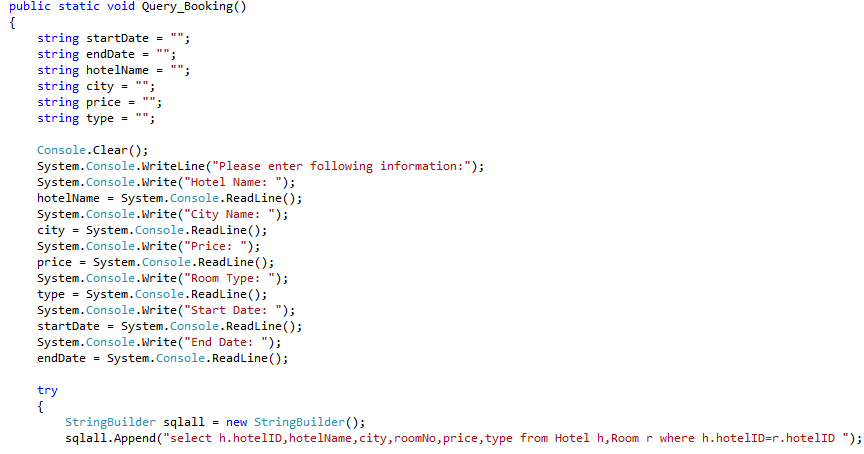
The records changing in the database is as below, we add a new guest called cand:

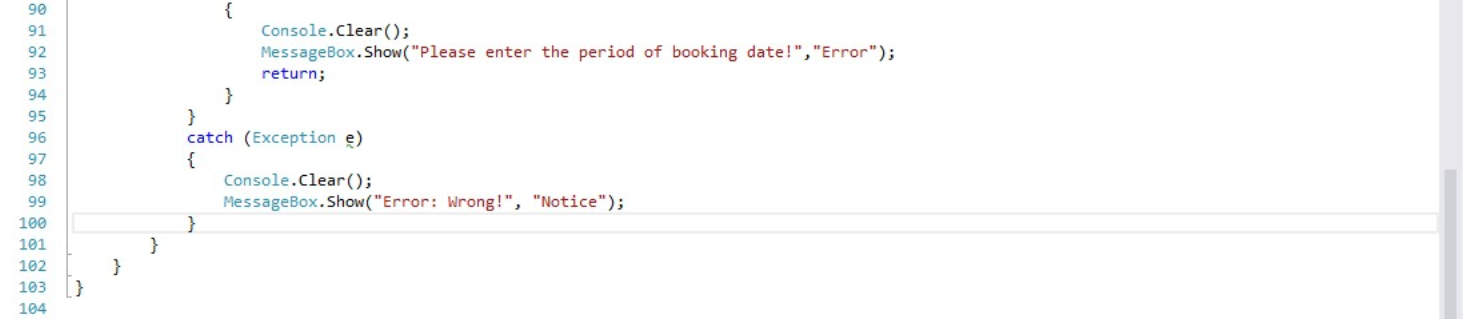


1. **check\_available class:**



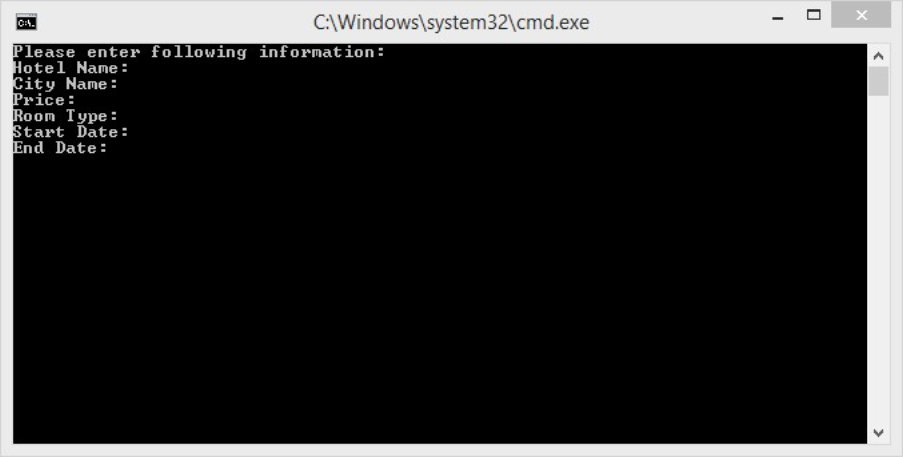
In this static, we list the records in database system, which satisfied the conditions that the user entered to the system. The conditions need to be entered are in the next static.

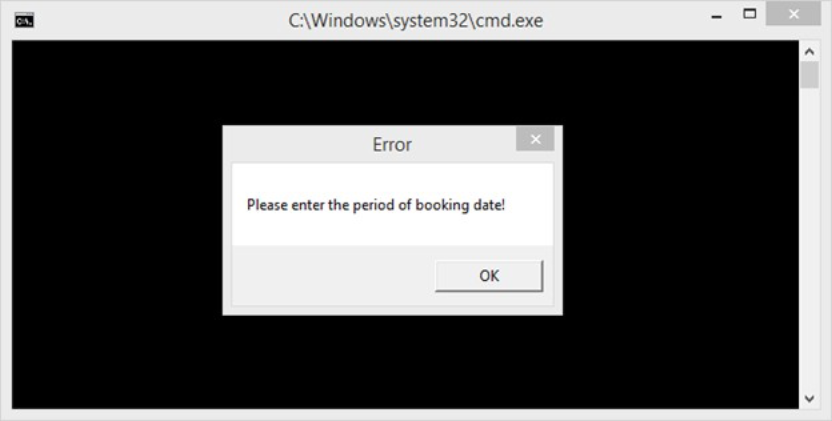


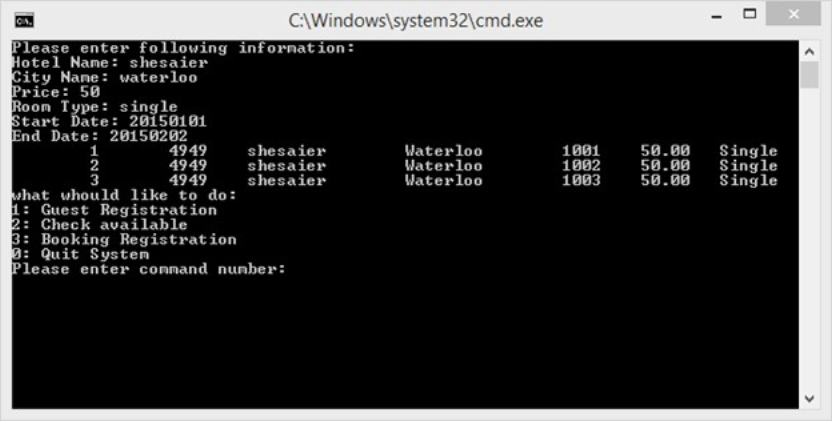
In this static, we search the records by using the restricted parameter such as the hotel name, the city name, the price of the room, the room type, and the start date and the end date. Then we use the entered parameters to search records in the database system. In this static, we use a logistic as the start date and end date is the restricted conditions, then when user enter one parameter, the system automatically add one parameter to the query language which connecting the system to database system, until the last parameter been added.

If the user does not enter the start date and end date, the system will produce a prompt message as below:

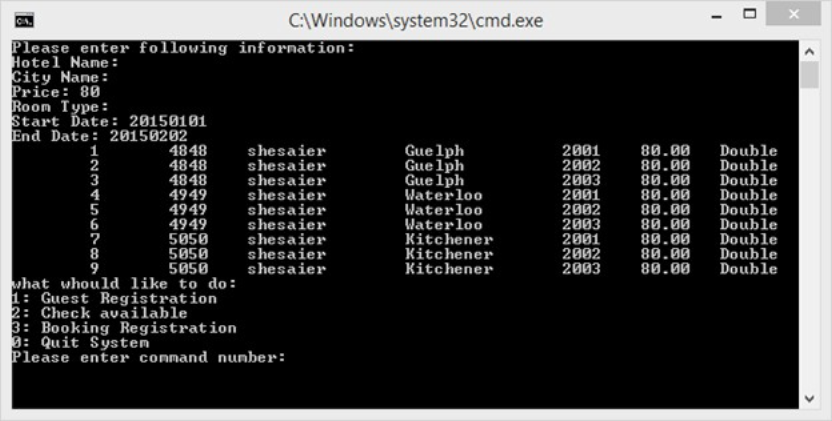




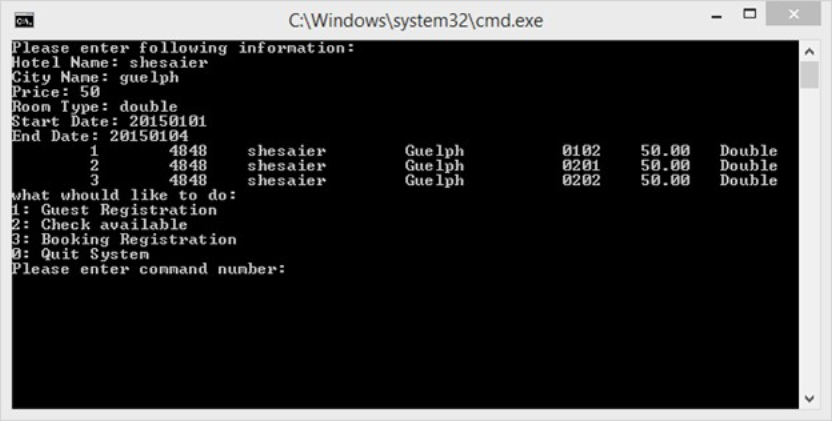
If the user enters all the information that system asked, the operation window is:



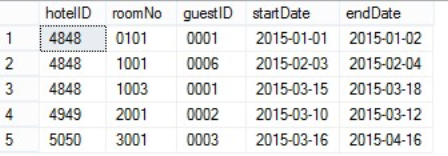
If the user only enters part of the required information, the system can still work well, the operation window is,

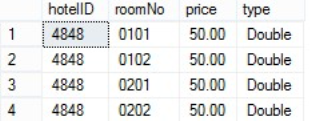


If the user enters conflict information, the system will only shows the available room but not conclude the conflict room:



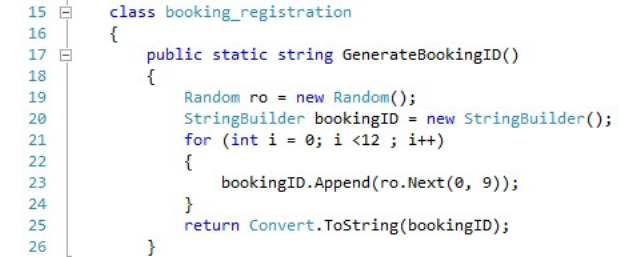
In this example, we enter a conflict time period between “2015-01-01” to “2015-01-04” but there exist in the period from “2015-01-01” to “2015-01-02” for room “0101” in the database, so it not shown in our output.





In the database system we have 4 rooms that satisfied the entered conditions, but room “0101” has been booked, so it did not shown in the result.

1. **booking\_registration class:**



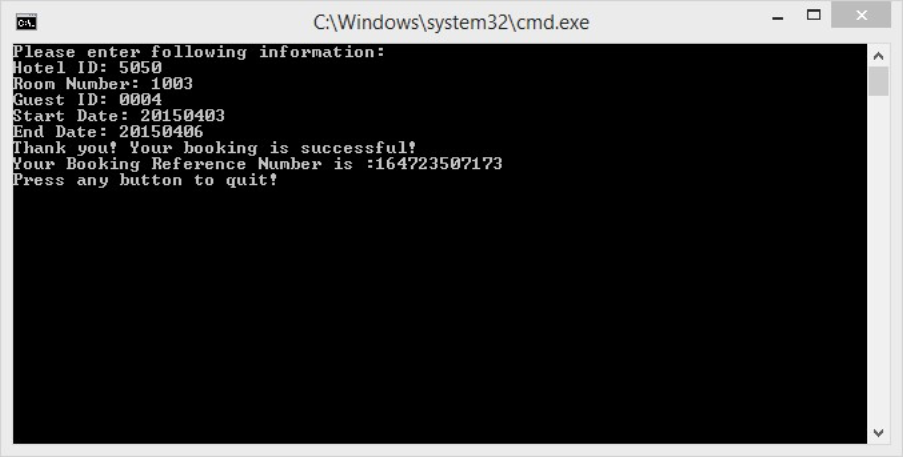
In this static, we create an 12-digits random number as booking ID which give to the customer for tracking the booking status.



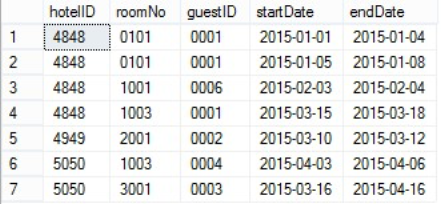


In this static, we will register a new room if the booking information does not conflict with the records in database system.

The operation window is:

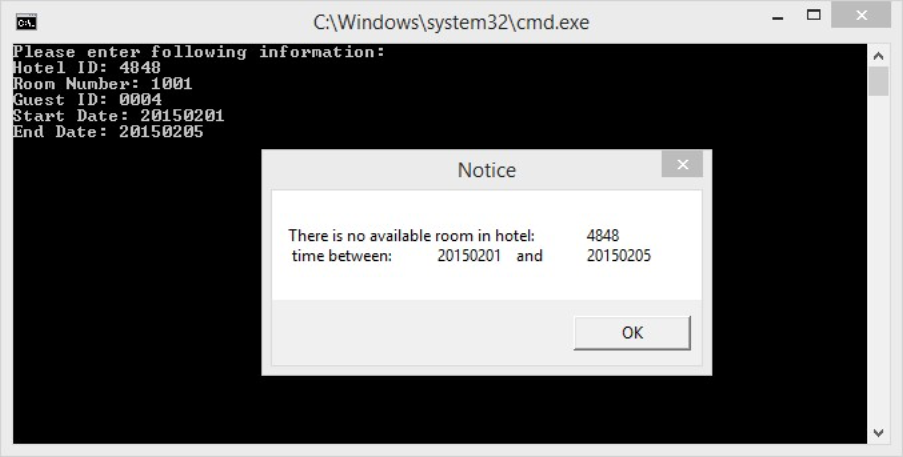


This window shows we add an new booking for guest 0004, the tracking ID is 164723507173,



This is the new booking record that adding to the database system.

If the booking information is conflict with the records in database system, the operation window will shows the prompt message as below:



If the customer did not enter all the information that the system required, the booking will failed, the system will produce a error message shows all input need to be not empty.

