BRITISH COUNCIL
INTERNATIONAL SCHOOL AWARD
2018–21

MERIDIAN SCHOOL
Connecting the Mind and Heart
MADHAPUR

HOTEL RESORT GUEST DATA ORGANIZER

A Project Report Submitted
in Partial Fulfilment of the Requirements

AISSCE - All India Senior School Certificate Examination
2020-2021: SCIENCE – XII A

In
COMPUTER SCIENCE (083)

NAMES	CLASS:	ROLL NO
1. TAMAN KOLANUPAKA	12 A	12
2. VENKATA NARAYANA	12 A	8
3. VISWA TEJA THOTA	12 A	5

MERIDIAN SCHOOL, MADHAPUR
NOVEMBER, 2020
UNDERTAKING

WE take great pride to declare that this project "**HEALTH RESORT GUEST DATA ORGANIZER**" submitted to **Mrs. SIRISHA GUNNALA for** the **CBSE - AISSE class XII certificate**. We have not plagiarized the Project for the Certificate. In case found guilty of any wrong doing, we accept that our Certificates may be unconditionally withdrawn.

November, 2020
Place: MERIDIAN SCHOOL
MADHAPUR

CERTIFICATE

I, Sirisha gunnala, Certify the project titled "**HEALTH RESORT GUEST DATA ORGANIZER"** by:

"**Taman Kolanupaka, Venkata Narayana, Viswa teja thota**", has been carried out under my guidance.

Sirisha Gunnala
Computer Science Teacher
Meridian school Madhapur

Acknowledgements

We are deeply indebted to our teacher and supporter**,** **Mrs. Sirisha Gunnala**

We would want to take this Chance to thank our Parents, our school and cbse for providing us with this opportunity.

ATLAST we would like to thank all our teachers, Mrs.Sirisha Gunnala and my team.

Contents

INTRODUCTION OF THE PROJECT

We the students of **CLASS XII A of Meridian School, Madhapur** have been assigned the Project **HEALTH RESORT GUEST DATA ORGANIZER**

Together, we worked as a Group to do the programming, coding and Debugging.

The project starts with –

1--->ENTER CUSTOMER DETAILS

2--->SEARCH FOR A CUSTOMER

3--->BOOKING RECORD

4--->CALCULATE ROOM RENT

5--->CALCULATE RESTUARANT BILL

6--->DISPLAY ALL CUSTOMER DETAILS

7--->GENERATE TOTAL BILL AMOUNT

8--->DELETE A CUSTOMER

9--->ANALYSE THE AMOUNT SPEND BY ALL CUSTOMERS CATEGORY WISE

10--->UPDATE CUSTOMER DETAILS

11--->EXIT

We are grateful to be given such an opportunity which wouldn't have been possible without the much-needed help from our teacher Mrs. Sirisha gunnala.

The Project has been a valid insight of how programming works in the future. We are hopeful that the code will help Hotel groups, Organize guest data more efficiently and easily.

PROCESS

1-We have done the planning in a paper work related to the project **HOTEL RESORT GUEST DATA ORGANIZER**.

2- We discussed our plan with our subject teacher

3-We started our project's foundation with our subject teacher.

4-We started the code input stage known as programming the project.

5-We debugged the code.

6-We prepared the project format as shown above.

System Requirements of the Project

Recommended System Requirements

Processors: Intel® Core™ i3 processor 4300M at 2.60 GHz.

Disk space: 2 to 4 GB.

Operating systems: Windows® 10, MACOS, and UBUNTU.

Python Versions: 3.X.X or Higher.

Minimum System Requirements

Processors: Intel Atom® processor or Intel® Core™ i3 processor.

Disk space: 1 GB.

Operating systems: Windows 7 or later, MACOS, and UBUNTU.

Python Versions: 2.7.X, 3.6.X.

Prerequisites before installing MySQL Connector Python

You need root or administrator privileges to perform the installation process.

Python must be installed on your machine.

Note: – MySQL Connector Python requires python to be in the system's PATH. Installation fails if it doesn't find Python.

On Windows, If Python doesn't exist in the system's PATH, please manually add the directory containing python.exe yourself.

PYTHON CODING

```
import pandas as pd
import matplotlib.pyplot as plt
import sqlalchemy as sq
from sqlalchemy import create_engine
engine = create_engine("mysql+mysqlconnector://root:#########@localhost/hotel")
connection = engine.connect()
con=sq.engine

#MAIN SCREEN
print("\n************************ MERIDIAN SCHOOL ***************")

print("\n************************ HOTEL MANAGEMENT ***************")

print("\n*****Designed and Maintained By: Venkata Narayana , Taman Kolanupaka , Vishwa
Teja" )

print("*****")

#STARTING POINT OF THE SYSTEM


# MODULE TO CREATE NEW CUSTOMER

def newCustomer():
    global cid
    if connection:
        createTable ="CREATE TABLE IF NOT EXISTS CUSTOMER_DETAILS( CID
varchar(10) NOT NULL,C_NAME VARCHAR(20) NOT NULL,C_AGE varchar(5) NOT
NULL,C_ADDRESS VARCHAR(60) NOT NULL,C_COUNTRY varchar(15) NOT
NULL,C_EMAIL VARCHAR(30) NOT NULL,C_CONTACT VARCHAR(12) NOT NULL);"
        engine.execute(createTable)
        print("\nPlease Fill All The Information Carefully !")
        cid = input("Enter the Customer ID Number    : ")
        name = input("Enter Customer Name           : ")
        age= input("Enter Customer Age              : ")
        address = input("Enter Customer Address      : ")
        country = input("Enter Customer Country      : ")
        email = input("Enter Customer Email          : ")
        phone= input("Enter Customer Contact Number  : ")
        df=
pd.DataFrame({'CID':[cid],'C_Name':[name],'C_AGE':[age],'C_ADDRESS':[address],'C_COU
NTRY':[country],'C_EMAIL':[email],'C_CONTACT':[phone]})
        df.to_sql('CUSTOMER_DETAILS', con=engine, index=False,if_exists='append')
        custs= engine.execute("SELECT * FROM CUSTOMER_DETAILS").fetchall()
        print(custs)
        print("\n New Customer Added Successfully !")


#MODULE TO SEARCH FOR CUSTOMER

def searchCustomer():
```

```
    global cid
    if connection:
        cid=input("ENTER CUSTOMER ID : ")
        query="SELECT * FROM CUSTOMER_DETAILS WHERE CID= %s"
        result = engine.execute(query,(cid,)).fetchall()
        if result:
            print(result)

df=pd.DataFrame(result,columns=['CID','C_Name','C_AGE','C_ADDRESS','C_COUNTRY','C
_EMAIL','C_CONTACT'])
            print(df)
            return True

        else:
            print("Record Not Found Try Again !")
            return False


#MODULE TO BOOK ROOM FOR CUSTOMER

def bookingDetails():
    global cid
    customer=searchCustomer()
    if customer:
        if connection:
            createTable ="CREATE TABLE IF NOT EXISTS BOOKING_DETAILS(CID
varchar(10) NOT NULL,CHECK_IN  date NOT NULL ,CHECK_OUT date NOT NULL)"
            engine.execute(createTable)
            checkin=input("\n Enter Customer CheckIN Date [ YYYY-MM-DD ]   : ")
            checkout=input("\n Enter Customer CheckOUT Date [ YYYY-MM-DD ] : ")
            df= pd.DataFrame({'CID':[cid],'CHECK_IN':[checkin],'CHECK_OUT':[checkout]})
            df.to_sql('BOOKING_DETAILS', con=engine, index=False,if_exists='append')
            booking_details= engine.execute("SELECT * FROM BOOKING_DETAILS").fetchall()
            print(booking_details)
            print("\nCHECK-IN AND CHECK-OUT ENTRY SUCCESSFULLY UPDATED !")


# MODULE TO CALCULATE ROOM_RENT

def RoomRent():
    global cid
    customer=searchCustomer()
    if customer:
        global roomrent
        if connection:
            createTable ="CREATE TABLE IF NOT EXISTS ROOM_RENT(CID VARCHAR(10)
NOT NULL,ROOM_CHOICE int(11) NOT NULL,NO_OF_DAYS int(11) NOT
NULL,ROOM_NO int(11) NOT NULL,ROOM_RENT int(11) NOT NULL);"
            engine.execute(createTable)
            print ("\n ##### We have The Following Rooms For You #####")
            print (" 1. Presidential ----> 25000 Rs.")
            print (" 2. Royal        ----> 10000 Rs. ")
            print (" 3. Elite        ----> 5000 Rs. ")
            print (" 4. Budget       ----> 3000 Rs. ")
            roomchoice =int(input("Enter Your Option : "))
```

```
        roomno=int(input("Enter Customer Room No : "))
        noofdays=int(input("Enter No. Of Days : "))
        if roomchoice==1:
            roomrent = noofdays * 25000
            print("\nPresidential Room Rent : ",roomrent)

        elif roomchoice==2:
            roomrent = noofdays * 10000
            print("\nRoyal Room Rent : ",roomrent)

        elif roomchoice==3:
            roomrent = noofdays * 5000
            print("\nElite Royal Room Rent : ",roomrent)

        elif roomchoice==4:
            roomrent = noofdays * 3000
            print("\nBudget Room Rent : ",roomrent)

        else:
            print("Sorry ,May Be You Are Giving Me The Wrong Input, Please Try Again !!! ")
            return
        df=
pd.DataFrame({'CID':[cid],'ROOM_CHOICE':[roomchoice],'NO_OF_DAYS':[noofdays],'ROO
M_NO':[roomno],'ROOM_RENT':[roomrent]})
        df.to_sql('ROOM_RENT', con=engine, index=False,if_exists='append')
        rooms= engine.execute("SELECT * FROM ROOM_RENT").fetchall()
        print(rooms)
        print("\nThank You , Your Room Has Been Booked For : ",noofdays , "Days" )
        print("\nYour Total Room Rent is : Rs. ",roomrent)


# MODULE TO CALCULATE RESTUARANT BILL

def RestaurantBill():
    global cid
    customer=searchCustomer()
    if customer:
        global restaurantbill
        if connection:
            createTable ="CREATE TABLE IF NOT EXISTS RESTAURANT_DETAILS(CID
VARCHAR(10) NOT NULL,CUISINE VARCHAR(30) NOT NULL,QUANTITY int(11) NOT
NULL,BILL int(11) NOT NULL)"
        engine.execute(createTable)
        print("""\n ######## We Welcome you to our restaurant-'FOOD ON FIRE' ########
        Here,We strive to serve you food for all your needs,be it
             The Proper Indian Thali
          The Masala filld Barbeque combo and many more
    Choose from our wide range of gourmet style food and appease your tastebuds""")
        print("1. INDIAN VEG THALI      -----> 800 Rs.")
        print("2. INDIAN NON-VEG THALI  -----> 1000 Rs.")
        print("3. SIZZLER               -----> 700 Rs.")
        print("4. BARBEQUE COMBO        -----> 1500 Rs.")
        choice_dish = int(input("Enter Your Cusine : "))
        quantity=int(input("Enter Quantity : "))
        if choice_dish==1:
```

```
            print("\nSO YOU HAVE ORDER: INDIAN VEG THALI ")
            restaurantbill = quantity * 800

        elif choice_dish==2:
            print("\nSO YOU HAVE ORDER: INDIAN NON-VEG THALI ")
            restaurantbill = quantity * 1000

        elif choice_dish==3:
            print("\nSO YOU HAVE ORDER: SIZZLER ")
            restaurantbill = quantity * 700

        elif choice_dish==4:
            print("\nSO YOU HAVE ORDER: BARBEQUE COMBO ")
            restaurantbill = quantity * 1500

        else:
            print("Sorry ,May Be You Are Giving Me The Wrong Input, Please Try Again !!! ")
            return


df=pd.DataFrame({'CID':[cid],'CUISINE':[choice_dish],'QUANTITY':[quantity],'BILL':[restauran
tbill]})
        df.to_sql('RESTAURANT_DETAILS', con=engine, index=False,if_exists='append')
        res_bill= engine.execute("SELECT * FROM RESTAURANT_DETAILS").fetchall()
        print(res_bill)
        print("Your Total Bill Amount Is : Rs. ",restaurantbill)
        print("\n\n**** WE HOPE YOU ENJOYED YOUR MEAL ***\n\n" )


#MODULE for displaying all customers

def displayAllCustomers():
    global cid
    if connection:
        result = engine.execute("SELECT * FROM CUSTOMER_DETAILS").fetchall()
        if result:

df=pd.DataFrame(result,columns=['CID','C_NAME','C_AGE','C_ADDRESS','C_COUNTRY','
C_EMAIL','C_CONTACT'])
            print(df)

        else:
            print("Sorry ! No Record Found , Please Try Again ! ")


#MODULE FOR GENERATING THE TOTAL BILL

def totalCustomerBill():
    global cid
    customer=searchCustomer()
    if customer:
        global grandTotal
        global cname
        global roomrent
        global restaurantbill
```

```
        if connection:
            query = "SELECT C.C_NAME NAME,R.ROOM_RENT STAYBILL,RES.BILL FROM
CUSTOMER_DETAILS C JOIN ROOM_RENT R ON (C.CID=R.CID) JOIN
RESTAURANT_DETAILS RES ON (C.CID=RES.CID) WHERE C.CID= %s"
            result = engine.execute(query,(cid,)).fetchall()
            if result:
                df=pd.DataFrame(result,columns=['C_Name','ROOM_RENT','RES_BILL'])
                print(df)
                cname=df.iloc[0]['C_Name']
                roomrent=df.iloc[0]['ROOM_RENT']
                restaurantbill=df.iloc[0]['RES_BILL']
                grandTotal=roomrent + restaurantbill
                print("\n **** THE GRAND TRIDENT HOTEL  **** CUSTOMER BIILING ****")
                print("\n CUSTOMER NAME      : " , cname)
                print("\n ROOM RENT          : Rs. ", roomrent)
                print("\n RESTAURANT BILL    : Rs. ", restaurantbill)
                print("_____________________________________________________")
                print("\nTOTAL AMOUNT : Rs. ",grandTotal)
                return True

            else:
                print("Record Not Found Try Again !")
                return False


# MODULE FOR DELETING A CUSTOMER RECORD

def deleteCustomer():
    global cid
    customer=searchCustomer()
    if customer:
        if connection:
            engine.execute('DELETE from CUSTOMER_DETAILS where CID =%s;',(cid,))
            engine.execute('DELETE from ROOM_RENT where CID =%s;',(cid,))
            engine.execute('DELETE from RESTAURANT_DETAILS where CID =%s;',(cid,))
            print("\n***** CUSTOMER AND RELATED DETAILS DELETED SUCCESSFULLY !
*****")


# MODULE TO UPDATE CUSTOMER DETAILS

def updateCustomerDetails():
    global cid
    customer=searchCustomer()
    if customer:
        if connection:
            print("1. UPDATE CUSTOMER NAME")
            print("2. UPDATE CUSTOMER AGE")
            print("3. UPDATE CUSTOMER ADDRESS")
            print("4. UPDATE CUSTOMER COUNTRY")
            print("5. UPDATE CUSTOMER EMAIL")
            print("6. UPDATE CUSTOMER CONTACT")
            change_detail = int(input("Enter Your choice for updation : "))
            if change_detail==1:
                change_name=input("PLEASE ENTER THE CORRECT NAME : ")
```

```
            sql='UPDATE CUSTOMER_DETAILS SET C_NAME = %s where CID =%s'
            engine.execute(sql,(change_name,cid))
            print("\n******* CUSTOMER NAME UPDATED SUCCESSFULLY! *******")

        elif change_detail==2:
            change_age=input("PLEASE ENTER THE CORRECT AGE : ")
            sql='UPDATE CUSTOMER_DETAILS SET C_AGE = %s where CID =%s'
            engine.execute(sql,(change_age,cid))
            print("\n******* CUSTOMER AGE UPDATED SUCCESSFULLY! *******")

        elif change_detail==3:
            change_address=input("PLEASE ENTER THE CORRECT ADDRESS : ")
            sql='UPDATE CUSTOMER_DETAILS SET C_ADDRESS = %s where CID =%s'
            engine.execute(sql,(change_address,cid))
            print("\n******* CUSTOMER ADDRESS UPDATED SUCCESSFULLY! *******")

        elif change_detail==4:
            change_country=input("PLEASE ENTER THE CORRECT COUNTRY : ")
            sql='UPDATE CUSTOMER_DETAILS SET C_COUNTRY = %s where CID =%s'
            engine.execute(sql,(change_country,cid))
            print("******* CUSTOMER COUNTRY UPDATED SUCCESSFULLY! *******")

        elif change_detail==5:
            change_email=input("PLEASE ENTER THE CORRECT EMAIL : ")
            sql='UPDATE CUSTOMER_DETAILS SET C_EMAIL = %s where CID =%s'
            engine.execute(sql,(change_email,cid))
            print("******* CUSTOMER EMAIL UPDATED SUCCESSFULLY! *******")

        elif change_detail==6:
            change_contact=input("PLEASE ENTER THE CORRECT CONTACT : ")
            sql='UPDATE CUSTOMER_DETAILS SET C_CONTACT = %s where CID =%s'
            engine.execute(sql,(change_contact,cid))
            print("******* CUSTOMER CONTACT UPDATED SUCCESSFULLY! *******")

        else:
            print("Sorry ,May Be You Are Giving Me The Wrong Input, Please Try Again !!! ")
            return


# MODULE FOR GENERATING GRAPH BY ALL CUSTOMERS ON RESTUARANT AND
ROOM

def customersWallet():
    if connection:
        query = "SELECT C.C_NAME NAME,R.ROOM_RENT STAYBILL,RES.BILL FROM
CUSTOMER_DETAILS C JOIN ROOM_RENT R ON (C.CID=R.CID) JOIN
RESTAURANT_DETAILS RES ON (C.CID=RES.CID)"
        result = engine.execute(query).fetchall()
        if result:
            df=pd.DataFrame(result,columns=['C_Name','ROOM_RENT','RES_BILL'])
            cname =  df['C_Name'].values.tolist()
            rent = df['ROOM_RENT'].values.tolist()
            res_bill = df['RES_BILL'].values.tolist()
```

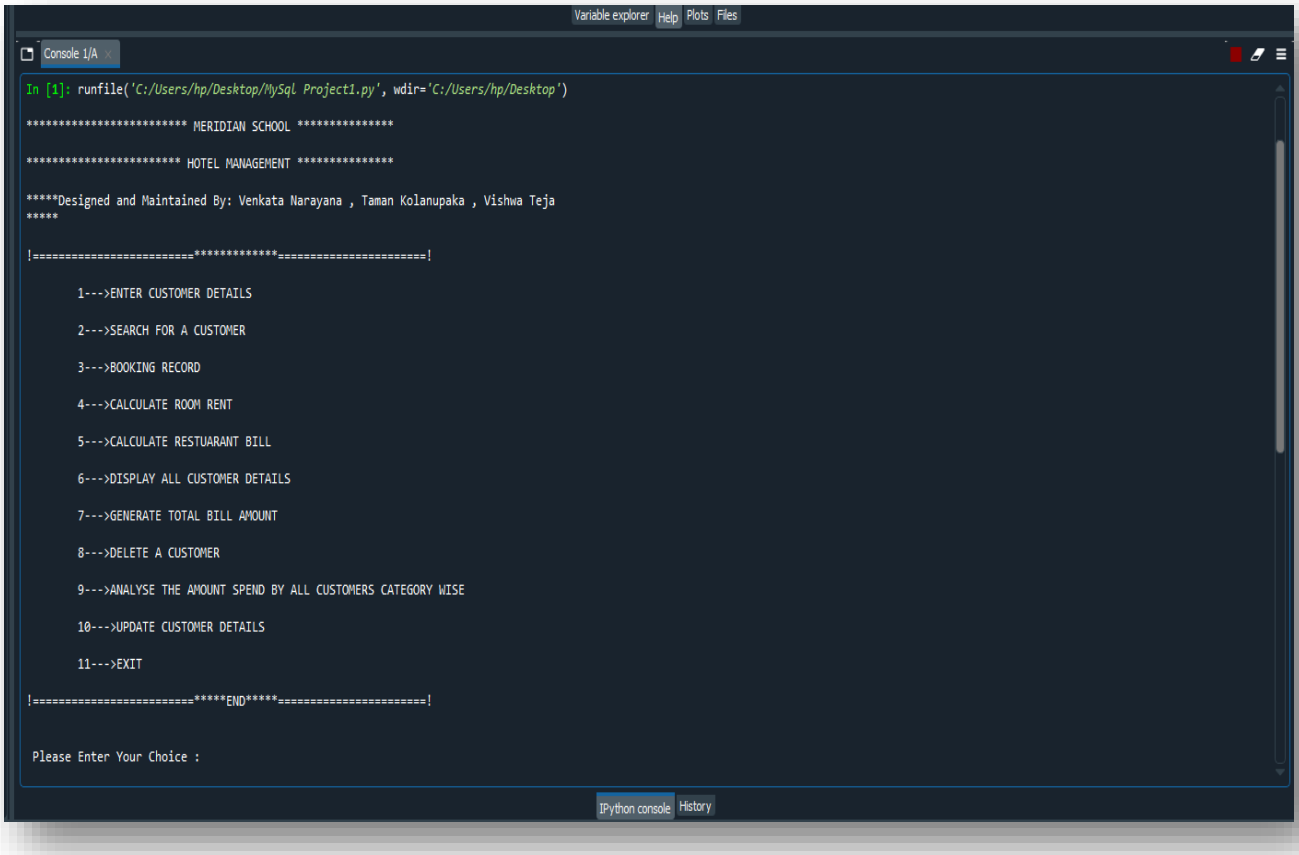
```
            plt.subplot(2,1,1)
            plt.bar(cname, rent, color='orange')
            plt.xlabel('Customer Name')
            plt.ylabel('Room Rent')
            plt.title('Customer vs Room Rent')
            plt.grid(True)
            plt.subplots_adjust(hspace=0.7,wspace=0.7)


            plt.subplot(2,1,2)
            plt.bar(cname, res_bill, color='cyan')
            plt.xlabel('Customer Name')
            plt.ylabel('Restaurant Bill')
            plt.title('Customer vs Restaurant Bill')
            plt.grid(True)
            plt.show()

        else:
            print("Record Not Found Try Again !")
            return False


def MenuSet():
    c='y'
    while(c=='y'):
        print("\n!=========================**********=======================!")
        print("""
        1--->ENTER CUSTOMER DETAILS

        2--->SEARCH FOR A CUSTOMER

        3--->BOOKING RECORD

        4--->CALCULATE ROOM RENT

        5--->CALCULATE RESTUARANT BILL

        6--->DISPLAY ALL CUSTOMER DETAILS

        7--->GENERATE TOTAL BILL AMOUNT

        8--->DELETE A CUSTOMER

        9--->ANALYSE THE AMOUNT SPEND BY ALL CUSTOMERS CATEGORY WISE

        10--->UPDATE CUSTOMER DETAILS

        11--->EXIT""")
        print("\n!=========================*****END*****=======================!")
        choice = int(input("\n Please Enter Your Choice : "))
        if choice == 1:
            newCustomer()

        elif choice == 2:
            searchCustomer()

        elif choice == 3:
```

```
            bookingDetails()

        elif choice == 4:
            RoomRent()
        elif choice == 5:
            RestaurantBill()

        elif choice==6:
            displayAllCustomers()

        elif choice==7:
            totalCustomerBill()

        elif choice==8:
            deleteCustomer()

        elif choice==9:
            customersWallet()

        elif choice==10:
            updateCustomerDetails()

        elif choice==11:
            print("Exiting")
            break

        else:
            print("Sorry ,May Be You Are Giving Me Wrong Input, Please Try Again !!! ")

Menuset()
```

OUTPUT OF THE CODE

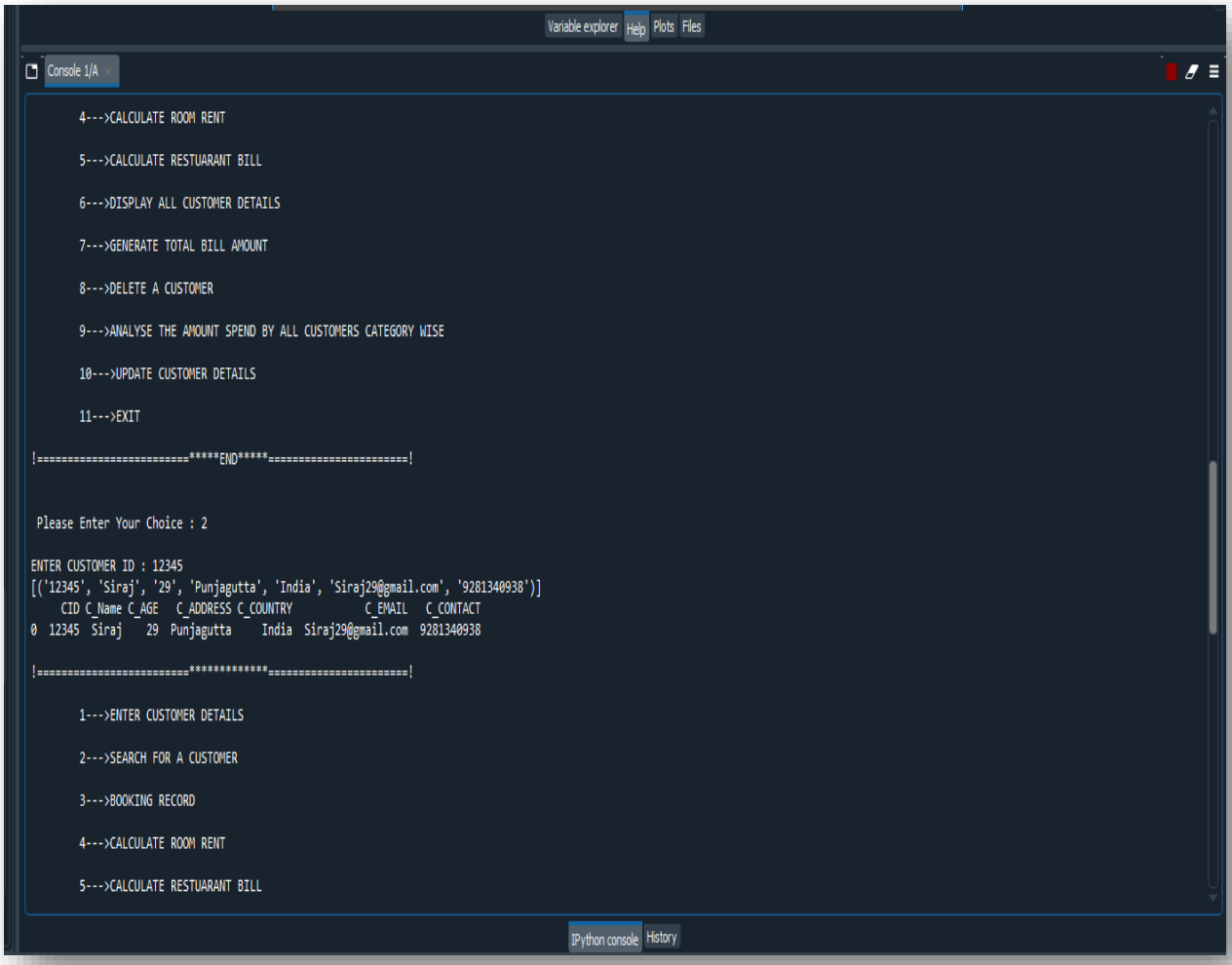
Finally, we conclude our work and present the output of the Project.

MAIN SCREEN

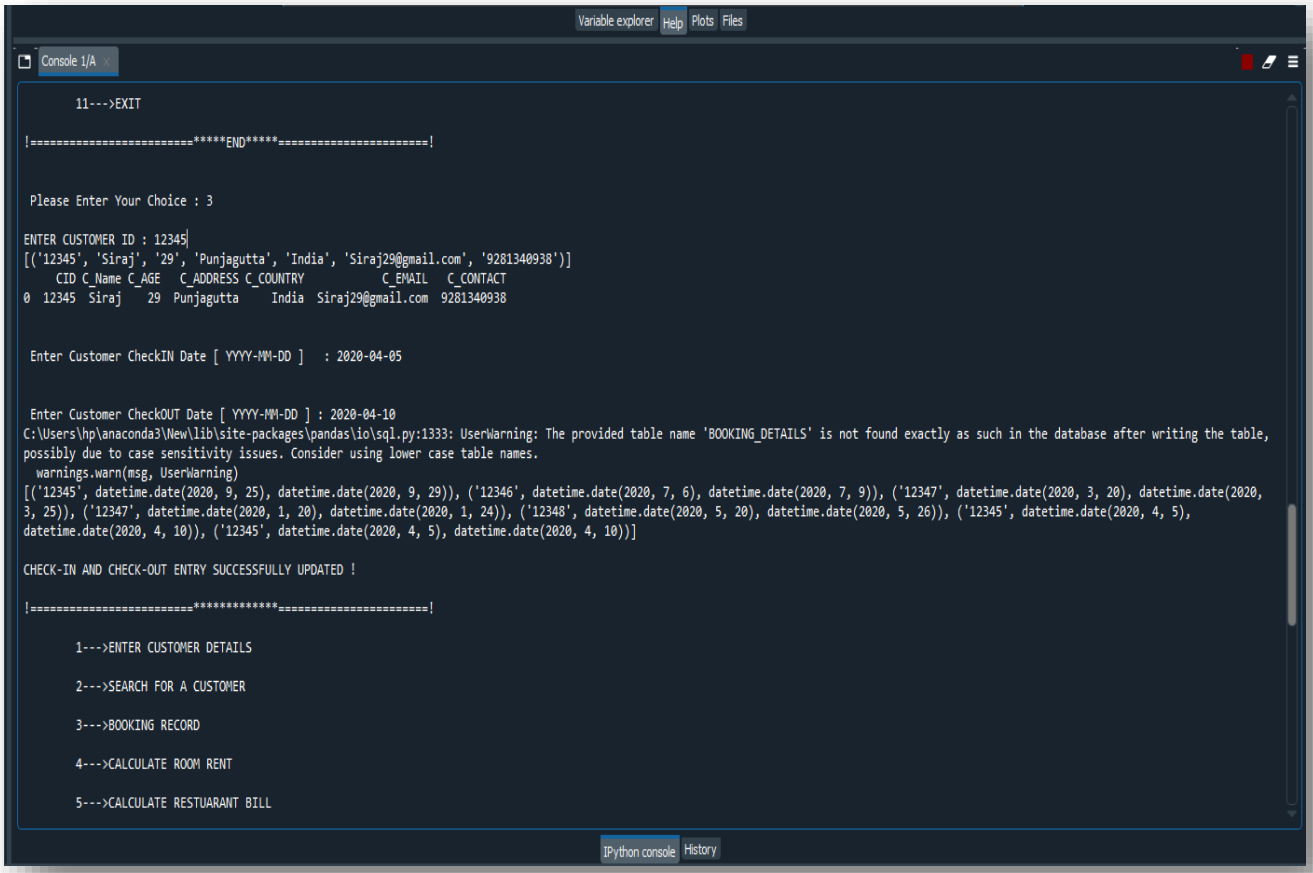


CUSTOMER DETAILS

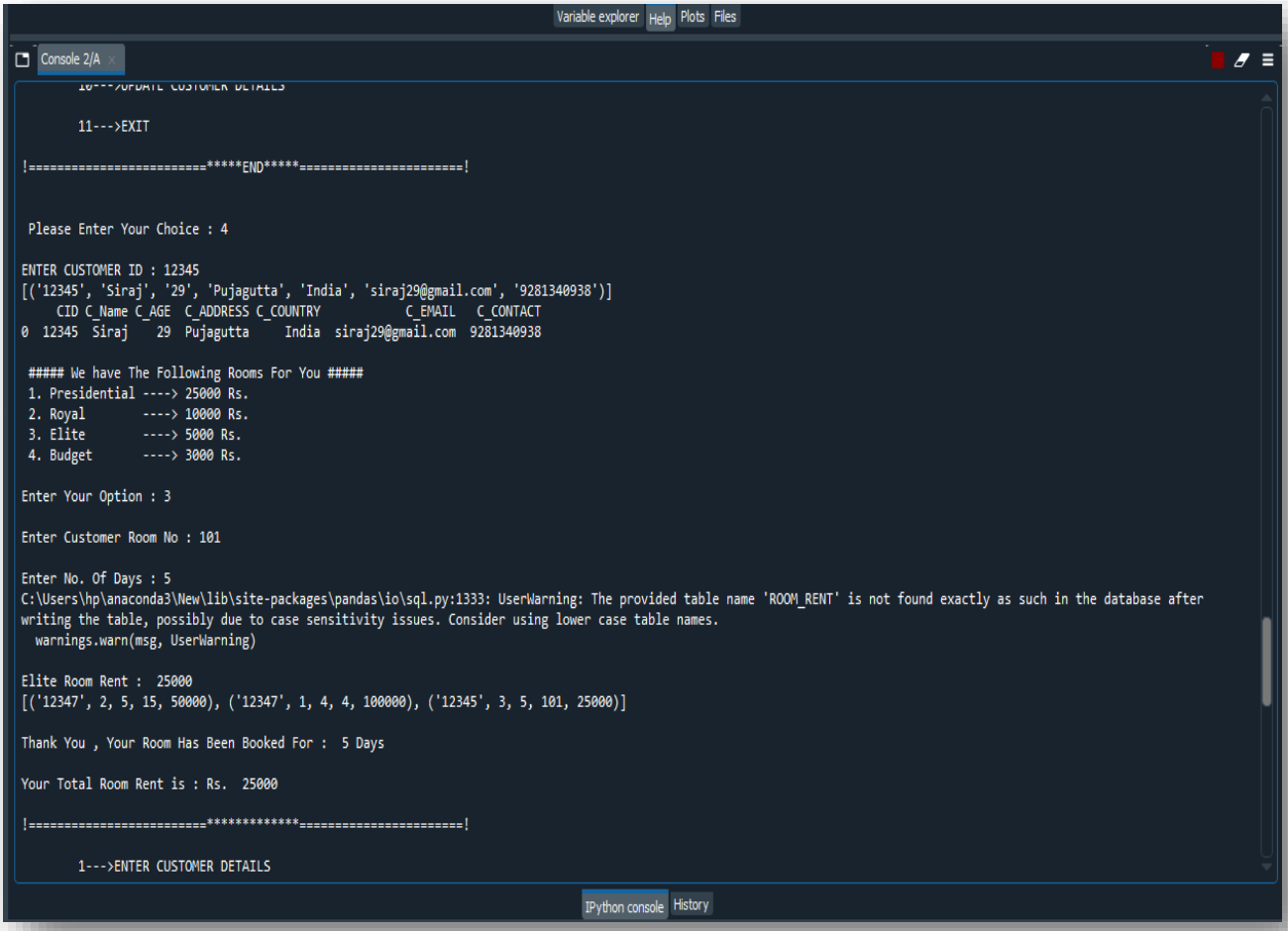
DETAILS OF A CUSTOMER



BOOKING RECORD

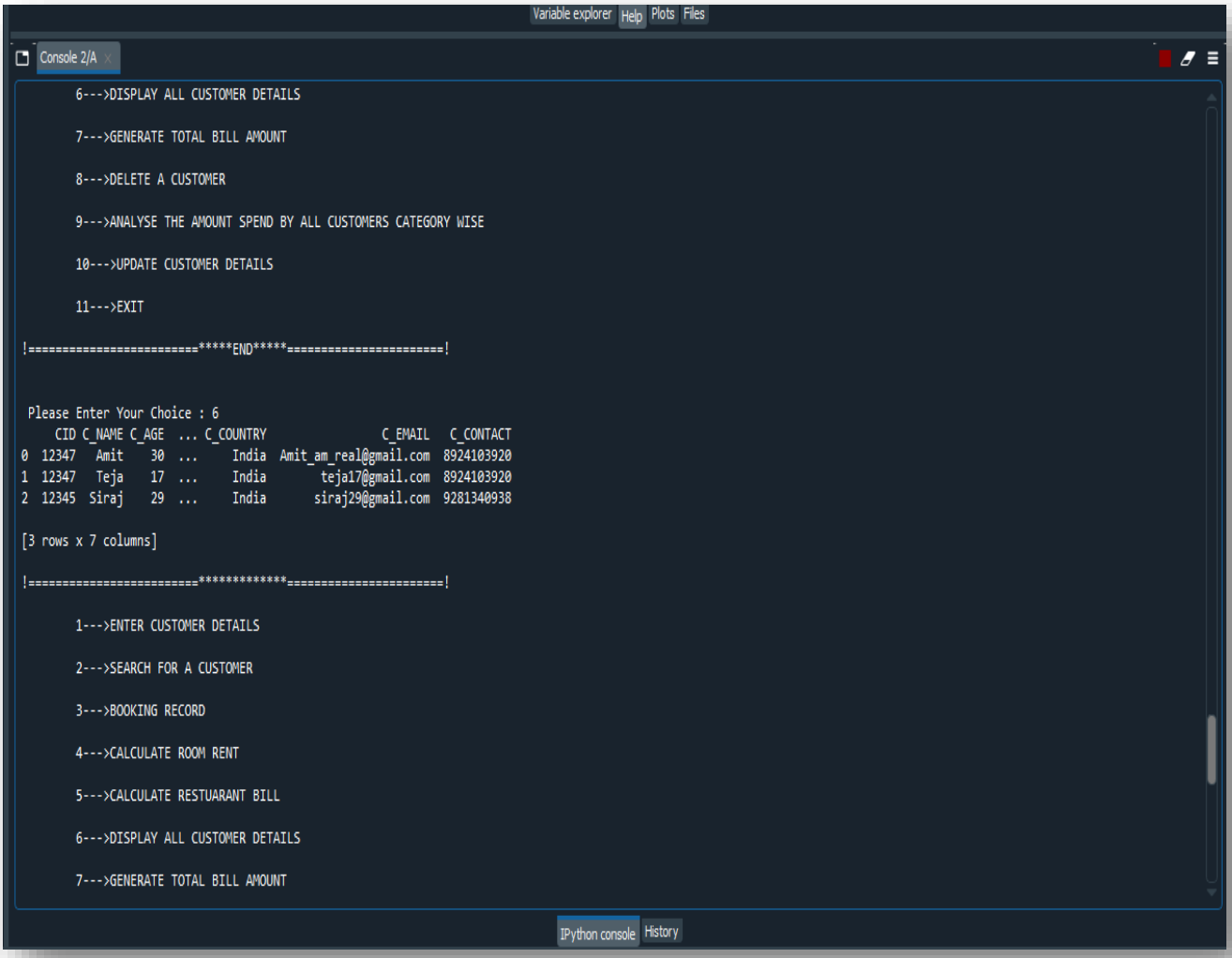


ROOM RENT

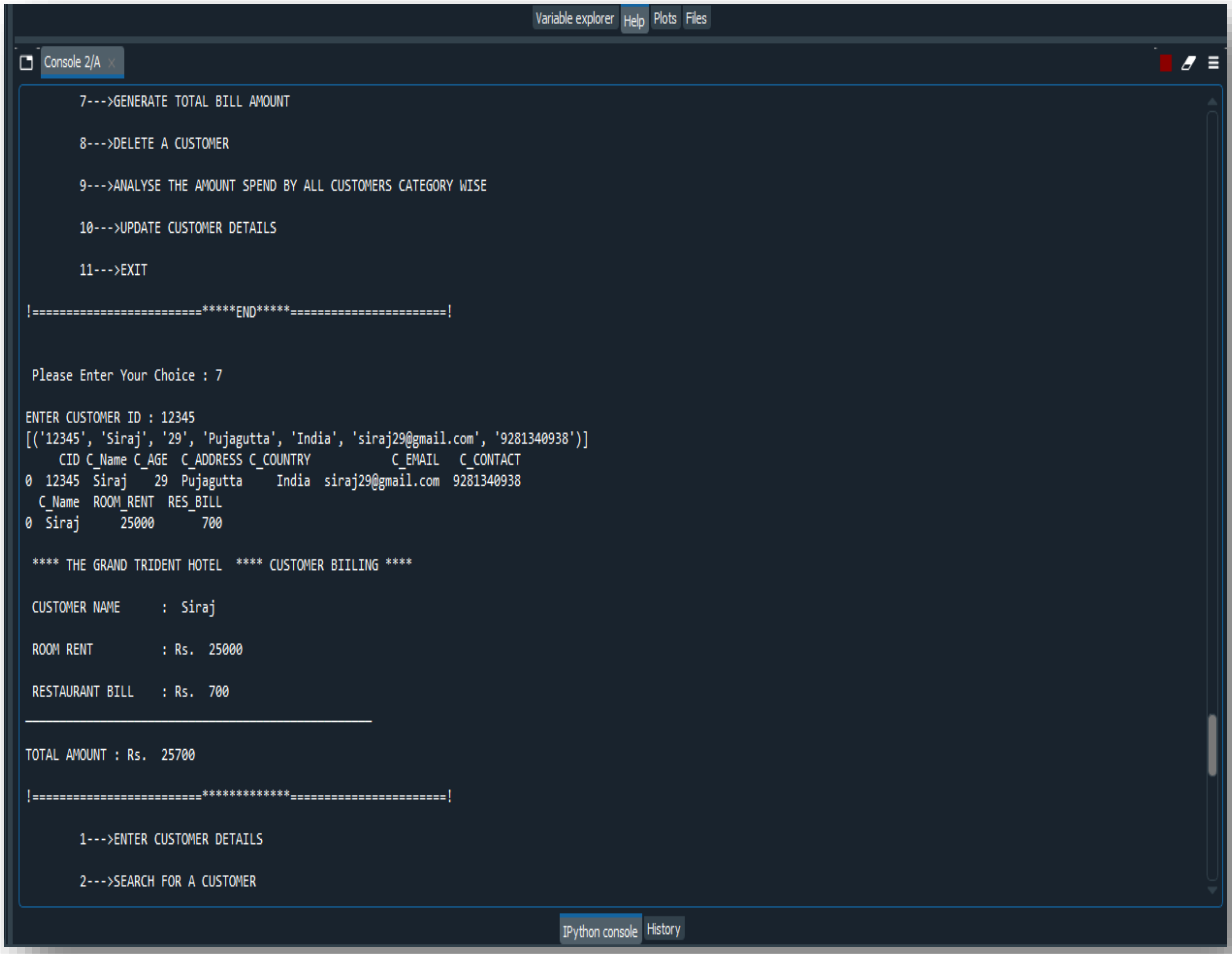

Variable explorer
Help
Plots
Files
Console 2/A
11--->EXIT
!=========*****END*****==========!
Please Enter Your Choice : 4
ENTER CUSTOMER ID : 12345
[('12345', 'Siraj', '29', 'Pujagutta', 'India', 'siraj29@gmail.com', '9281340938')]
CID C_Name C_AGE C_ADDRESS C_COUNTRY C_EMAIL C_CONTACT
0 12345 Siraj 29 Pujagutta India siraj29@gmail.com 9281340938
We have The Following Rooms For You
1. Presidential ----> 25000 Rs.
2. Royal ----> 10000 Rs.
3. Elite ----> 5000 Rs.
4. Budget ----> 3000 Rs.
Enter Your Option : 3
Enter Customer Room No : 101
Enter No. Of Days : 5
C:\Users\hp\anaconda3\New\lib\site-packages\pandas\io\sql.py:1333: UserWarning: The provided table name 'ROOM_RENT' is not found exactly as such in the database after writing the table, possibly due to case sensitivity issues. Consider using lower case table names.
warnings.warn(msg, UserWarning)
Elite Room Rent : 25000
[('12347', 2, 5, 15, 50000), ('12347', 1, 4, 4, 100000), ('12345', 3, 5, 101, 25000)]
Thank You , Your Room Has Been Booked For : 5 Days
Your Total Room Rent is : Rs. 25000
!=========**********==========!
1--->ENTER CUSTOMER DETAILS
IPython console
History

RESTAURANT BILL

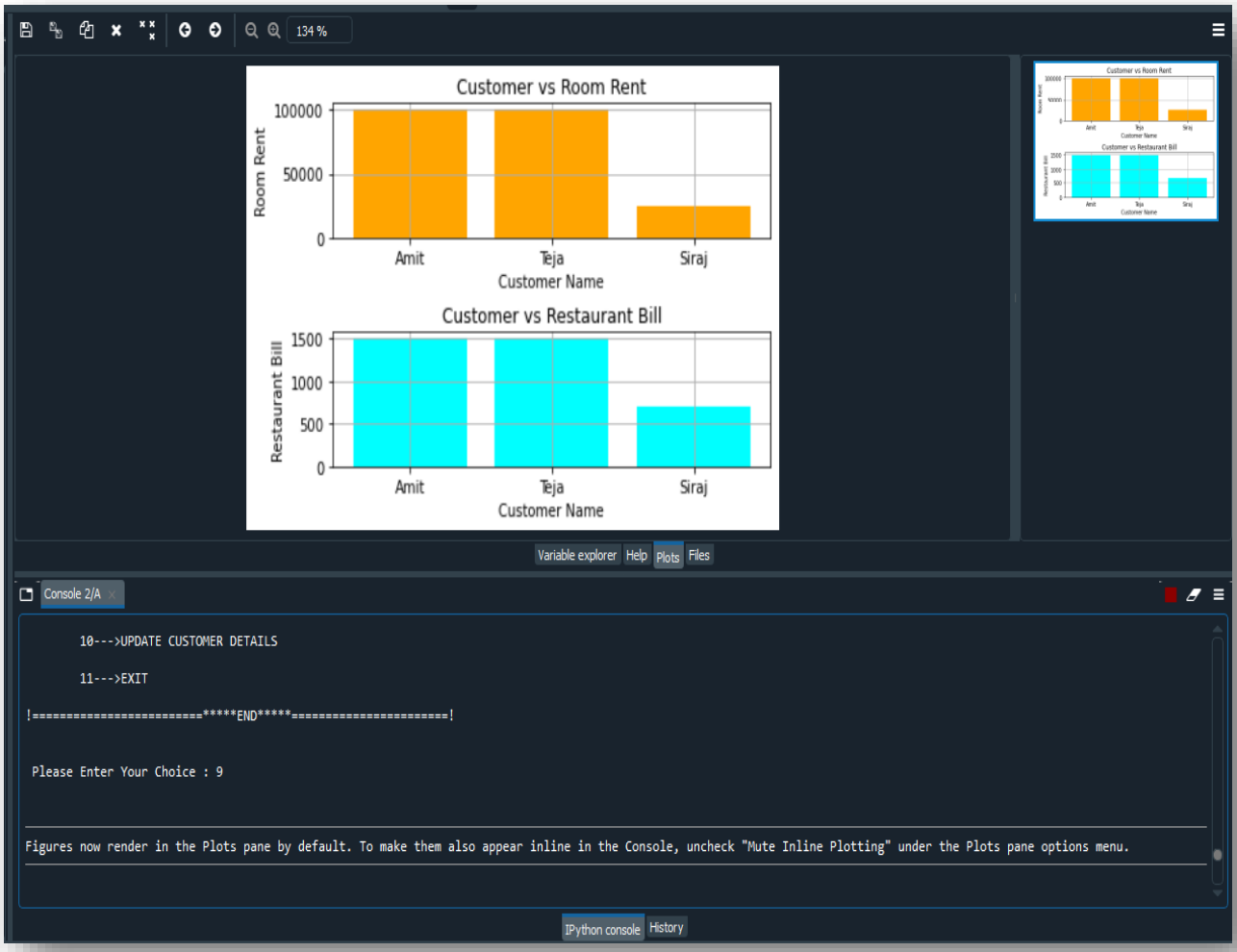
DETAILS OF ALL CUSTOMERS



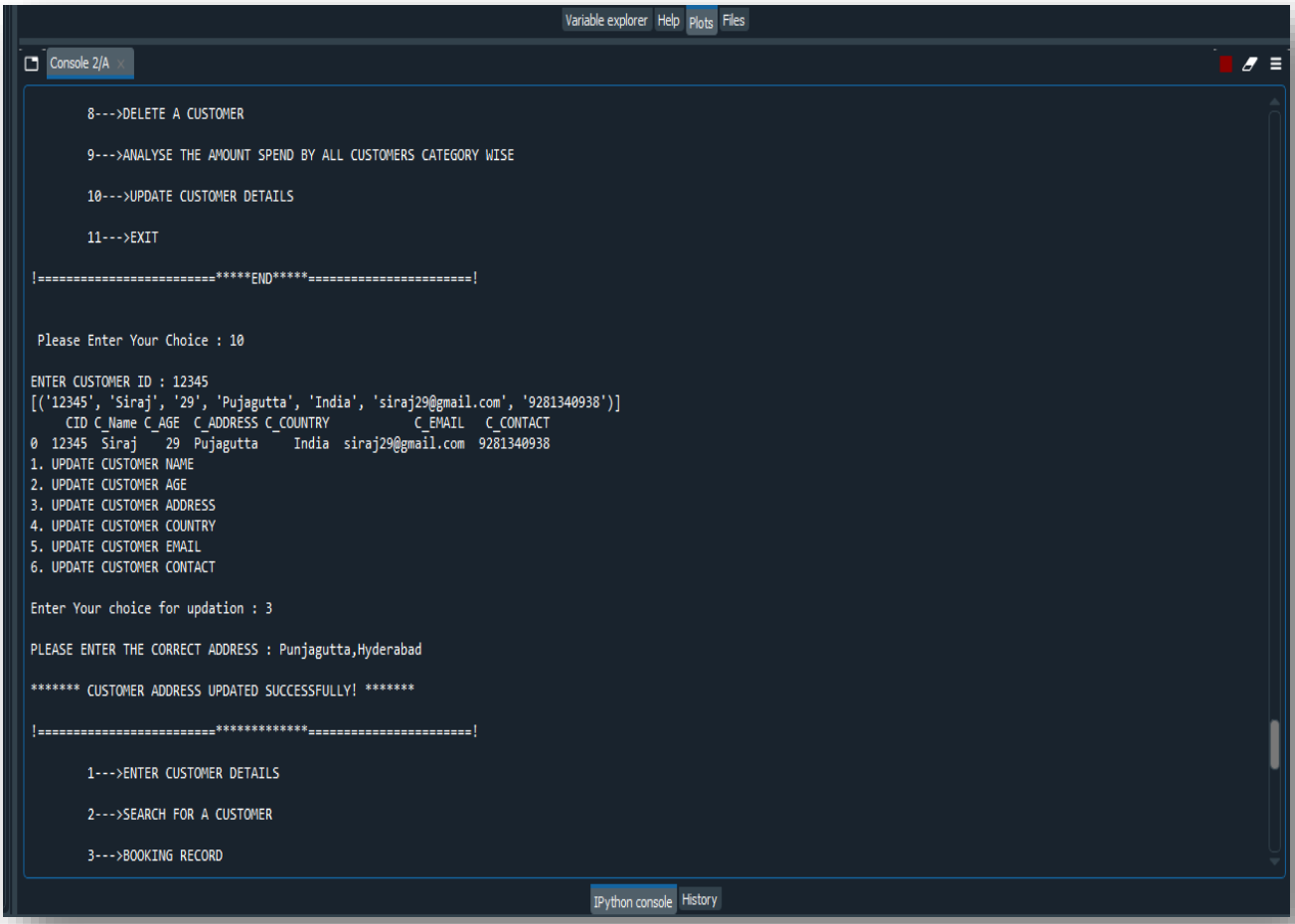
TOTAL BILL


Variable explorer
Help
Plots
Files
Console 2/A
7--->GENERATE TOTAL BILL AMOUNT
8--->DELETE A CUSTOMER
9--->ANALYSE THE AMOUNT SPEND BY ALL CUSTOMERS CATEGORY WISE
10--->UPDATE CUSTOMER DETAILS
11--->EXIT
!=========*****END*****==========!
Please Enter Your Choice : 7
ENTER CUSTOMER ID : 12345
[('12345', 'Siraj', '29', 'Pujagutta', 'India', 'siraj29@gmail.com', '9281340938')]
CID C_Name C_AGE C_ADDRESS C_COUNTRY C_EMAIL C_CONTACT
0 12345 Siraj 29 Pujagutta India siraj29@gmail.com 9281340938
C_Name ROOM_RENT RES_BILL
0 Siraj 25000 700
**** THE GRAND TRIDENT HOTEL **** CUSTOMER BIILING ****
CUSTOMER NAME : Siraj
ROOM RENT : Rs. 25000
RESTAURANT BILL : Rs. 700
TOTAL AMOUNT : Rs. 25700
!=========*********==========!
1--->ENTER CUSTOMER DETAILS
2--->SEARCH FOR A CUSTOMER
IPython console
History

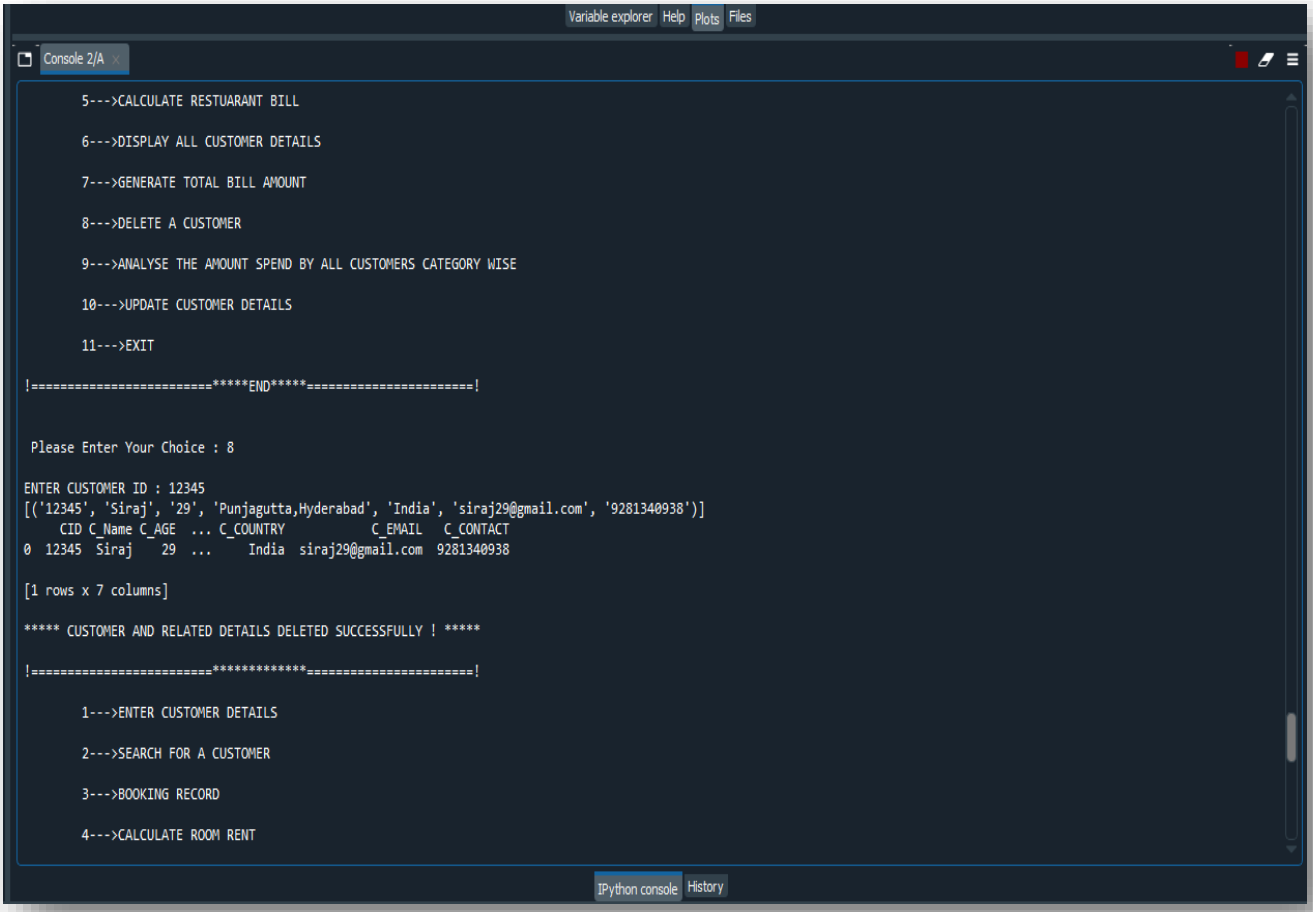
ANALYSIS OF AMOUNT SPENT BY THE CUSTOMERS CATEGORY-WISE



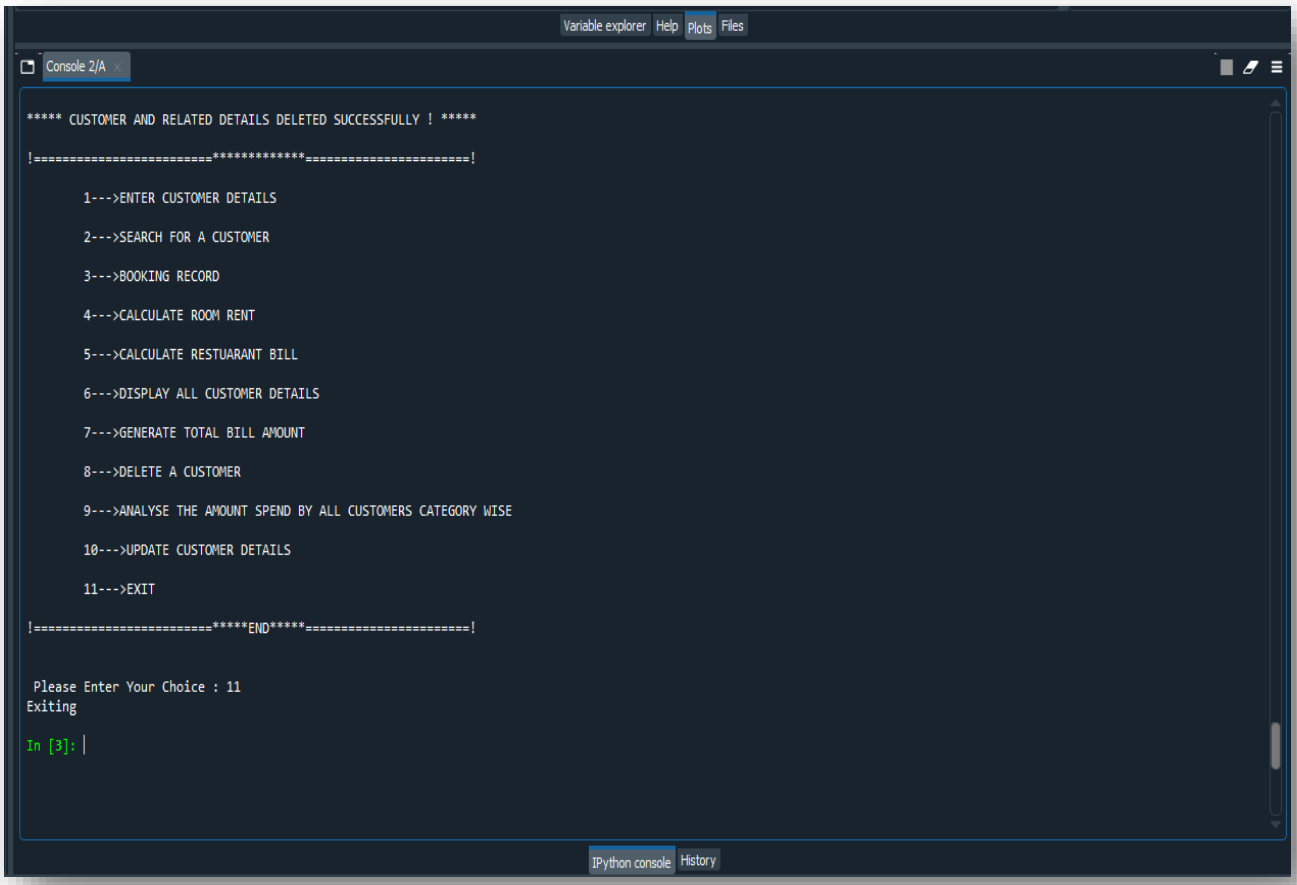
UPDATING CUSTOMER DETAILS



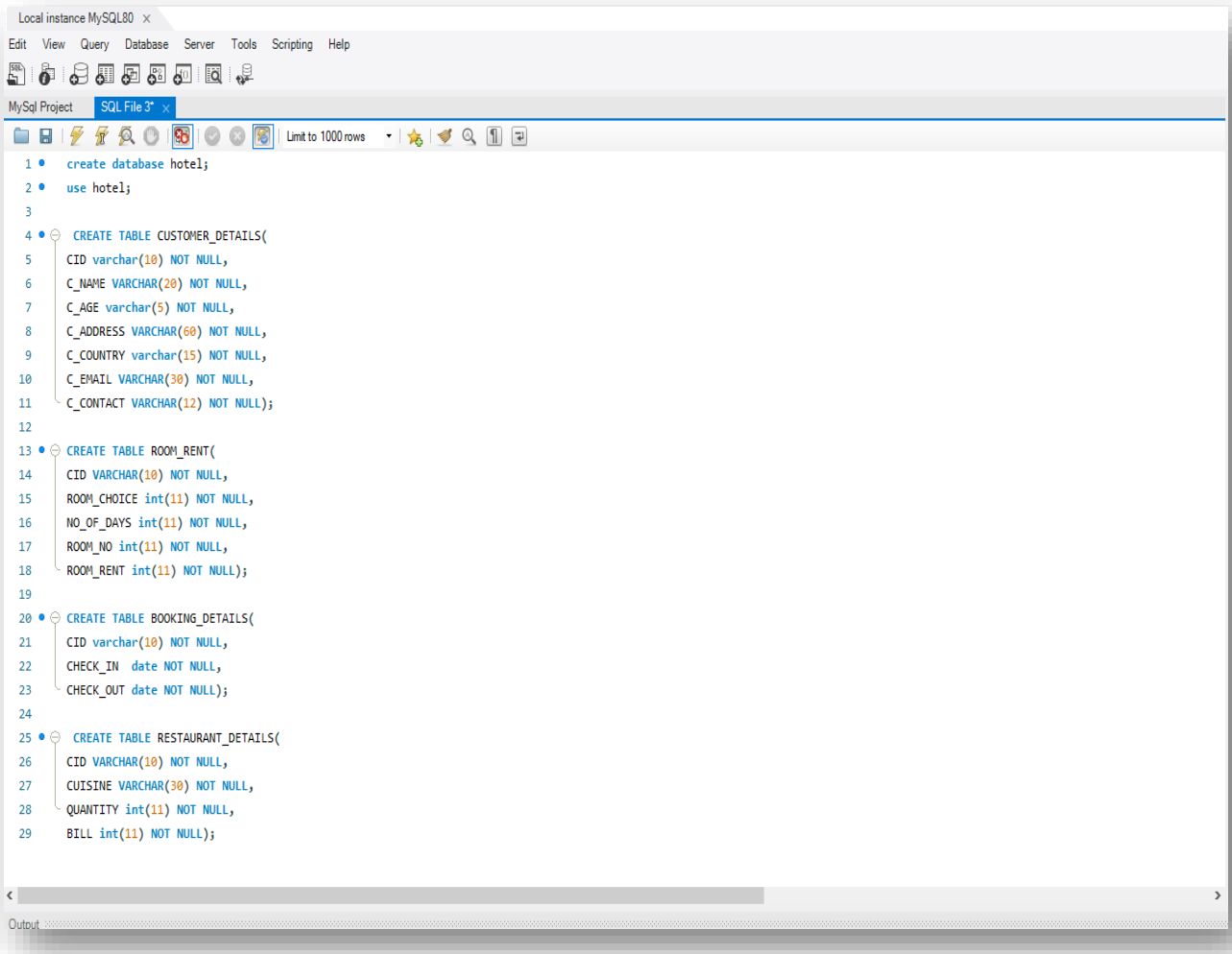
DELETING A CUSTOMER



EXITING


Variable explorer
Help
Plots
Files
Console 2/A
***** CUSTOMER AND RELATED DETAILS DELETED SUCCESSFULLY ! *****
!=========================**********=======================!
1--->ENTER CUSTOMER DETAILS
2--->SEARCH FOR A CUSTOMER
3--->BOOKING RECORD
4--->CALCULATE ROOM RENT
5--->CALCULATE RESTUARANT BILL
6--->DISPLAY ALL CUSTOMER DETAILS
7--->GENERATE TOTAL BILL AMOUNT
8--->DELETE A CUSTOMER
9--->ANALYSE THE AMOUNT SPEND BY ALL CUSTOMERS CATEGORY WISE
10--->UPDATE CUSTOMER DETAILS
11--->EXIT
!=========================*****END*****=======================!
Please Enter Your Choice : 11
Exiting
In [3]:
IPython console
History

WORKBENCH CODE




```
create database hotel;
use hotel;

 CREATE TABLE CUSTOMER_DETAILS(
CID varchar(10) NOT NULL,
C_NAME VARCHAR(20) NOT NULL,
C_AGE varchar(5) NOT NULL,
C_ADDRESS VARCHAR(60) NOT NULL,
C_COUNTRY varchar(15) NOT NULL,
C_EMAIL VARCHAR(30) NOT NULL,
C_CONTACT VARCHAR(12) NOT NULL);

CREATE TABLE ROOM_RENT(
CID VARCHAR(10) NOT NULL,
ROOM_CHOICE int(11) NOT NULL,
NO_OF_DAYS int(11) NOT NULL,
ROOM_NO int(11) NOT NULL,
ROOM_RENT int(11) NOT NULL);

CREATE TABLE BOOKING_DETAILS(
CID varchar(10) NOT NULL,
CHECK_IN  date NOT NULL,
CHECK_OUT date NOT NULL);

 CREATE TABLE RESTAURANT_DETAILS(
CID VARCHAR(10) NOT NULL,
CUISINE VARCHAR(30) NOT NULL,
QUANTITY int(11) NOT NULL,
BILL int(11) NOT NULL);
```

MYSQL DATABASE AND TABLES USED IN THIS PROJECT

DATABASE

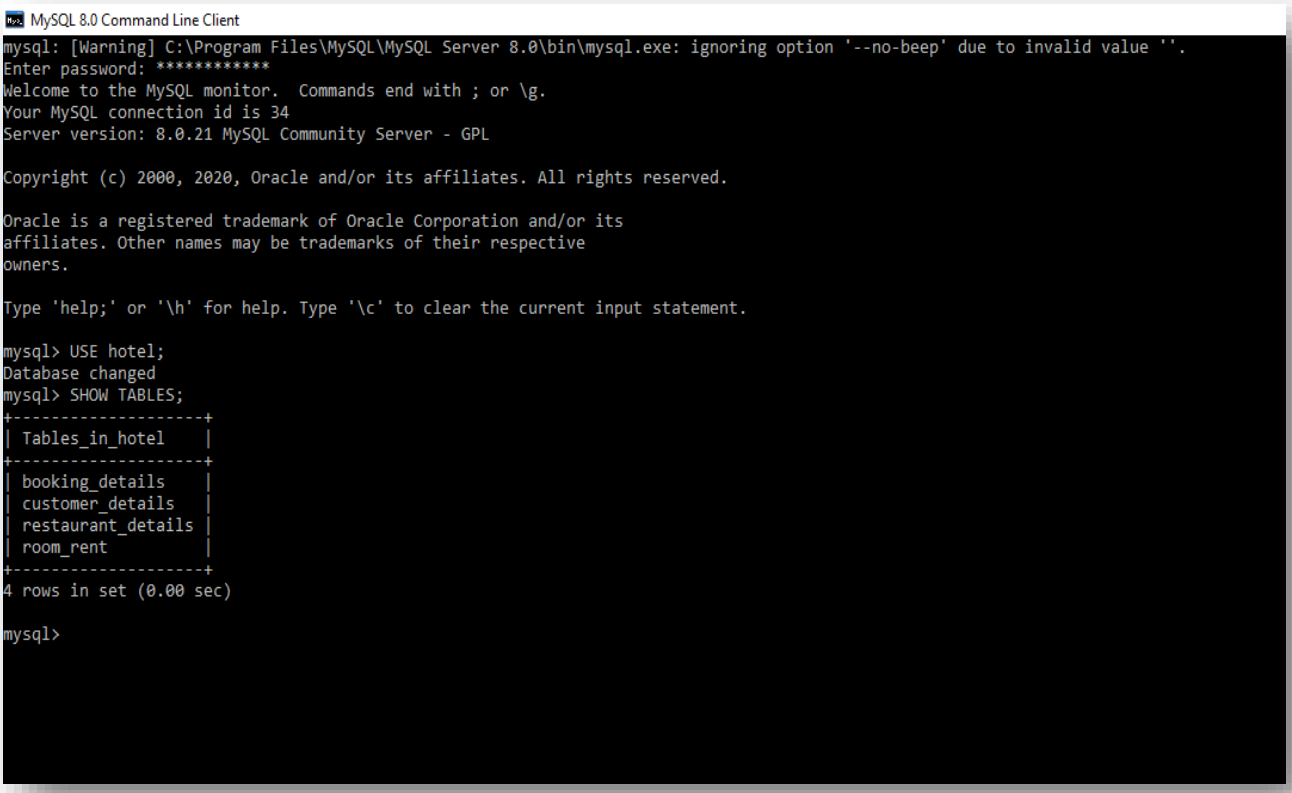
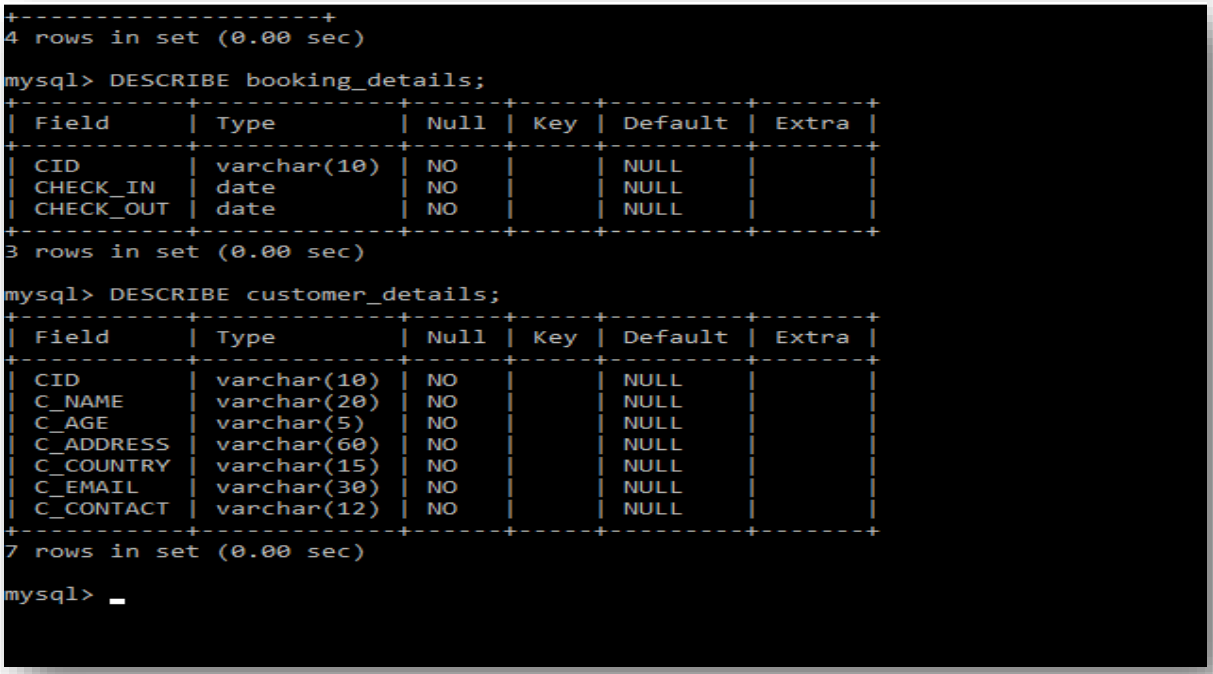
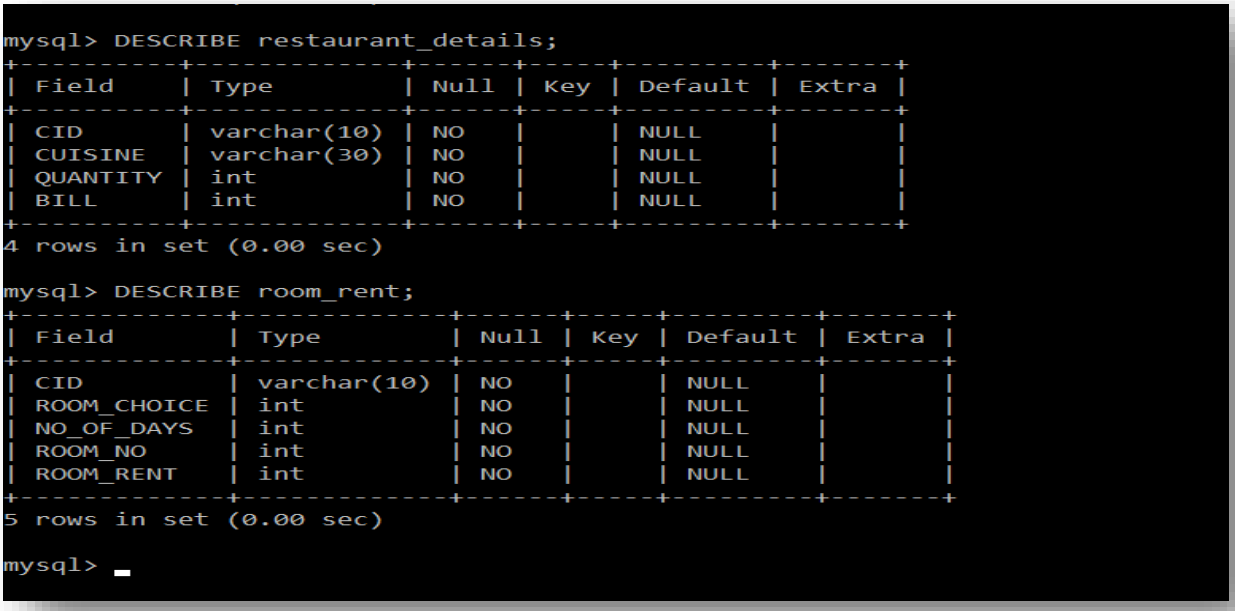


TABLE STRUCTURE 1 AND 2



```
+--------------------+
4 rows in set (0.00 sec)

mysql> DESCRIBE booking_details;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| CID       | varchar(10) | NO   |     | NULL    |       |
| CHECK_IN  | date        | NO   |     | NULL    |       |
| CHECK_OUT | date        | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> DESCRIBE customer_details;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| CID       | varchar(10) | NO   |     | NULL    |       |
| C_NAME    | varchar(20) | NO   |     | NULL    |       |
| C_AGE     | varchar(5)  | NO   |     | NULL    |       |
| C_ADDRESS | varchar(60) | NO   |     | NULL    |       |
| C_COUNTRY | varchar(15) | NO   |     | NULL    |       |
| C_EMAIL   | varchar(30) | NO   |     | NULL    |       |
| C_CONTACT | varchar(12) | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

mysql>
```

TABLE STRUCTURE 3 AND 4



```
mysql> DESCRIBE restaurant_details;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| CID      | varchar(10) | NO   |     | NULL    |       |
| CUISINE  | varchar(30) | NO   |     | NULL    |       |
| QUANTITY | int         | NO   |     | NULL    |       |
| BILL     | int         | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> DESCRIBE room_rent;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| CID         | varchar(10) | NO   |     | NULL    |       |
| ROOM_CHOICE | int         | NO   |     | NULL    |       |
| NO_OF_DAYS  | int         | NO   |     | NULL    |       |
| ROOM_NO     | int         | NO   |     | NULL    |       |
| ROOM_RENT   | int         | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql>
```

BACKEND DATA GENERATED THROUGH SOFTWARE

```
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM booking_details;
+-------+------------+------------+
| CID   | CHECK_IN   | CHECK_OUT  |
+-------+------------+------------+
| 12345 | 2020-09-25 | 2020-09-29 |
| 12346 | 2020-07-06 | 2020-07-09 |
| 12347 | 2020-03-20 | 2020-03-25 |
| 12347 | 2020-01-20 | 2020-01-24 |
| 12348 | 2020-05-20 | 2020-05-26 |
| 12345 | 2020-04-05 | 2020-04-10 |
| 12345 | 2020-04-05 | 2020-04-10 |
| 12346 | 2020-05-03 | 2020-05-05 |
| 12347 | 2020-05-20 | 2020-05-24 |
| 12348 | 2020-06-06 | 2020-06-12 |
+-------+------------+------------+
10 rows in set (0.00 sec)

mysql>
```

BACKEND DATA GENERATED THROUGH SOFTWARE

```
+-------+-----------+-----------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM customer_details;
+-------+--------+-------+-----------------------+----------+--------------------+-----------+
| CID   | C_NAME | C_AGE | C_ADDRESS             | C_COUNTRY | C_EMAIL            | C_CONTACT |
+-------+--------+-------+-----------------------+----------+--------------------+-----------+
| 12345 | Siraj  | 29    | Punjagutta            | India     | Siraj29@gmail.com  | 9281340938 |
| 12346 | Arjun  | 34    | Pragati Nagar,Hyderabad | India     | arjun34@gmail.com  | 9201372932 |
| 12347 | Chitra | 24    | Jubilee Hills,Hyderabad | India     | spchitra@gmail.com | 9562819376 |
| 12348 | Sarah  | 28    | Gachibowli,Hyderabad  | India     | sarahjazz@gmail.com | 9568201773 |
+-------+--------+-------+-----------------------+----------+--------------------+-----------+
4 rows in set (0.00 sec)

mysql>
```

BACKEND DATA GENERATED THROUGH SOFTWARE

```
mysql> SELECT * FROM restaurant_details;
+-------+---------+----------+------+
| CID   | CUISINE | QUANTITY | BILL |
+-------+---------+----------+------+
| 12346 | 4       |        1 | 1500 |
| 12347 | 1       |        1 |  800 |
| 12348 | 4       |        2 | 3000 |
| 12345 | 3       |        1 |  700 |
+-------+---------+----------+------+
4 rows in set (0.00 sec)

mysql>
```

BACKEND DATA GENERATED THROUGH SOFTWARE

```
+-------+---------+----------+------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM room_rent;
+-------+-------------+------------+---------+-----------+
| CID   | ROOM_CHOICE | NO_OF_DAYS | ROOM_NO | ROOM_RENT |
+-------+-------------+------------+---------+-----------+
| 12346 |           4 |          2 |     102 |      6000 |
| 12347 |           2 |          4 |     103 |     40000 |
| 12348 |           1 |          6 |     104 |    150000 |
| 12345 |           3 |          5 |     101 |     25000 |
+-------+-------------+------------+---------+-----------+
4 rows in set (0.00 sec)

mysql>
```