

Project Report

Statement:

We are creating a Customer Management System for Hotel Le Meridien.

Hotel Le Meridien

Room Details

Customer ID

3

Floor Location

☐ 1st ☐ 2nd ☐ 3rd ☐ 4th ☐ 5th

Room Number

101

Room Type

☐ Deluxe ☐ Suite ☐ Twin ☐ King ☐ Queen

Price for selected room is:

Number of Bed's:

Parking Access

☐ YES ☐ NO

Balcony View

☐ Front view ☐ Central View ☐ Downtown View ☐ Pool-side View

Check In date

mm / dd / yyyy

Check Out date

mm / dd / yyyy

Description:

The customer management system of Hotel uses the HTML forms to collect the customer information, MySQL database to store the information and Python programming to connect data from HTML forms to the database. It will act as a middleware to return the result value as well.

Assumptions:

Please find the assumptions we have taken into account while creating our management system.

1) There will be 2 types of customers who will be coming to the Hotel i.e.

- The individual customers, who would be residing in the hotel alone and they have an entry in the main Customer table with Customer ID.
- Customers who coming with family or in groups, for them there will be one primary customer whose information will be recorded in the Customer table and a CustomerID will be generated for them.

And the rest of the members will be registered into the Dependents table related to main table with the Customer ID. These customer will take part in Room and Event but will not be allocated a Customer Id as they will be dependent once for main customer.

2) Only the primary customer is responsible for the

- Event booking,
- Room bookings
- Payments.

Hence all the tables are related to each other using the Customer ID.

3) In Customer table Customer ID is the primary key with auto increment starting from One.

4) Room table has all the details about the room and will generate a booking id which is unique key auto incrementing from 10000 value. In this table Checking date and Room No is composite primary key enabling a constraint where no two primary customer can book one room at a time.

5) Event table has event room and start date as composite key enabling a constraint where no 2 primary customers can book 1 event at a time.

Added Constraints :-

1) After booking if Any customer does not shows up then all entries related to this customer will be deleted.

2) Total Room amount would be calculated as number of days stayed by the customer multiplied by the individual room amount

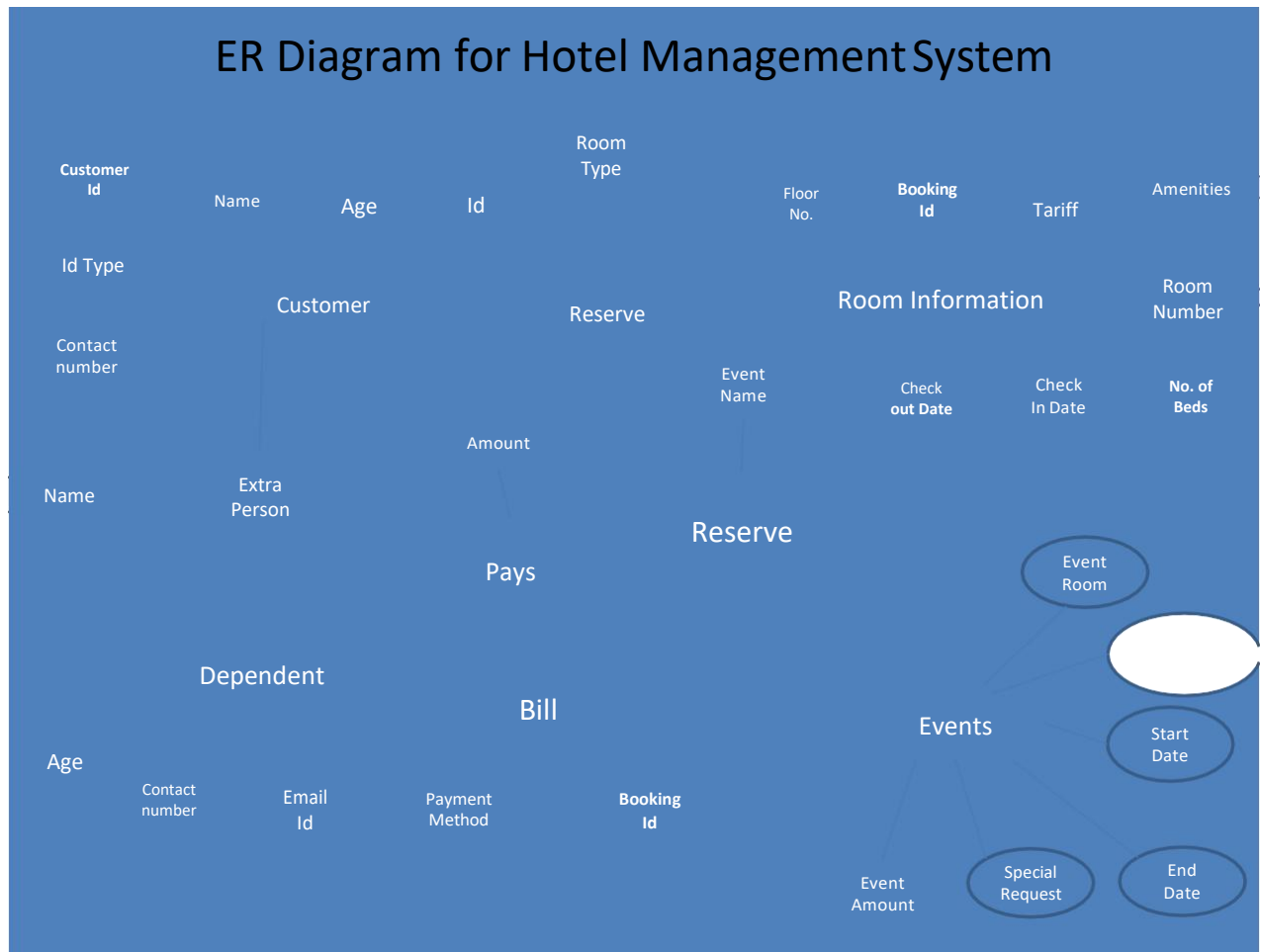
Solution

Part 1:-

Describe in at most one page (preferably less than half a page) any assumptions and constraints you made. To get full marks, in addition to any constraints mentioned in

“General description” you should describe at least 2 constraints more (in addition) to the ones specified above. All key constraints should be determined in the ER below.

Provide a complete ER diagram that models the “General description” and includes all constraints and assumptions you wrote in (1) above.



Assumptions:

Please find the assumptions we have taken into account while creating our management system.

1) There will be 2 types of customers who will be coming to the Hotel i.e.

- The individual customers, who would be residing in the hotel alone and they have an entry in the main Customer table with Customer ID.

- Customers who coming with family or in groups, for them there will be one primary customer whose information will be recorded in the Customer table and a CustomerID will be generated for them.

And the rest of the members will be registered into the Dependents table related to main table with the Customer ID. These customer will take part in Room and Event but will not be allocated a Customer Id as they will be dependent once for main customer.

2) Only the primary customer is responsible for the

- Event booking,
- Room bookings
- Payments.

Hence all the tables are related to each other using the Customer ID.

3) In Customer table Customer ID is the primary key with auto increment starting from One.

4) Room table has all the details about the room and will generate a booking id which is unique key auto incrementing from 10000 value. In this table Checking date and Room No is composite primary key enabling a constraint where no two primary customer can book one room at a time.

5) Event table has event room and start date as composite key enabling a constraint where no 2 primary customers can book 1 event at a time.

6) In reservation table amount payable will be calculated as the sum of room amount and the event amount for specific customer ID.

Added Constraints :-

1) After booking if Any customer does not shows up then all entries related to this customer will be deleted.

2) Total Room amount would be calculated as number of days stayed by the customer multiplied by the individual room amount

Part 2 :-

Implement the ER diagram as a Relational Model. Specify the tables you are using together with the domains (datatypes).

Remark: the relational model is not going to encode all of the constraints of the ER model. Realize the above database and express the following query in Relational Algebra and in

SQL: Find the customer (or customers) who paid the highest room rate in 2017 and is also related to at least one more non-primary customer.

When you implement the database in SQL, specify the tables together with all constraints.

What to report (same for (i) and (ii)) The Relational Algebra query. The schema of your relational DB, the SQL queries that create the relations (careful to include all ICs). We cannot provide data for DB tables because part of the question is you to design the DB schema (i.e. not everyone will have the same tables). Therefore, you should make up entries to your database and list the contents of the relation instances (tables) in a similar way that they were listed at the end of the Lab Exercise you did a few weeks ago. Make sure that the entries you make are indicative for the SQL queries you should present. Finally, you should report the SQL output on the example entries you made .

Solution :-

ER to Relational :-

Customers
CustomerID INT (Primary Key)
CustomerName VARCHAR(45)
Age INT
IDType Varchar(10)
IDNumber Int
Email VARCHAR(45)
Phone BIGINT
Dependents VARCHAR(5)

Rooms
BookingID INT (Primary Key)
CustomerID INT
RoomNo INT
RoomType VARCHAR(45)

Price float
Floor VARCHAR(10)
Parking varchar(5)
Balcony varchar(5)
Room_Amt as Price*Duration Double
CheckinDate date
CheckoutDate date
Beds INT
Duration INT
Unique Key(CheckingDate + RoomNo)

Events
CustomerID int (Foreign Key)
EventName VARCHAR(45)
Event_Room Varchar(45)
Max_Occ INT
StartDate VARCHAR(45)
EndDate VARCHAR(45)
Spl_Request VARCHAR(150)
Event_Amount INT
Primary Key(Event_Room + StartDate)

Dependents
Name VARCHAR(45)
CustomerID INT Foreign Key
Age INT
Email VARCHAR(45)
Phone bigint

Reservations
CustomerID int (Foreign Key)
BookingID int (Foreign Key)
Amount INT
Payment_Type Varchar(45)

Relation Query :-

Q. Find the customers who paid the highest room rate in 2017.
and is also related to atleast one more non-primary customers.

Answer:

Customers (CustomerID, CustomerName, Age, IDtype, IDNumber, Email, Phone, Dependents)
Room (BookingID, CustomerID, RoomNo., RoomType, Price, Floor, Parking, Balcony, CheckinDate, Check out Date, Beds, Duration)
Event (CustomerID, EventName, EventRoom, MaxOcc, Start Date, EndDate, Request, EventAmount)
Reservations (CustomerID, BookingID, Amount, PaymentType)
Dependents (Name, CustomerID, Age, Email, Phone)

Relational Algebra:

(i) $\pi_{\text{CustomerName, Price}} \left(\sigma_{\substack{\text{CustomerID} = \left(\text{CustomerX} \rho(\text{Room}) \right) \\ \text{CustomerID2} \\ \text{AND} \\ \text{Dependents} = \text{'Yes'} \\ \text{AND} \\ \text{CheckinDate(Year)} = 2017}} \right)$

$\pi_{\text{CustomerName}} (\text{Customers}) - \pi_{\text{CustomerName}} \left(\sigma_{\substack{\text{Price} > \\ \text{Price2}}} \left((i) \times \rho(i) \right) \right)$
 $\text{CustomerName} \rightarrow \text{CustomerName2}$
 $\text{Price} \rightarrow \text{Price2}$

SQL Script :-

```
CREATE TABLE hotel.Customers (  
  CustomerID INT NOT NULL AUTO_INCREMENT,  
  CustomerName VARCHAR(45) ,  
  Age INT,  
  IDType Varchar(10),  
  IDNumber Int,  
  Email VARCHAR(45),  
  Phone BIGINT,  
  Dependents VARCHAR(5),  
  PRIMARY KEY (CustomerID));
```

```
CREATE TABLE hotel.Rooms (  
  BookingID INT NOT NULL AUTO_INCREMENT,  
  CustomerID INT ,  
  RoomNo VARCHAR(45),  
  RoomType VARCHar(45),  
  Price INT,  
  Floor VARCHAR(10) NOT NULL ,  
  Parking varchar(5) ,  
  Balcony varchar(45),  
  CheckinDate date,  
  CheckoutDate date,  
  Beds int ,  
  Duration INT,  
  RoomAmount double as (Price * Duration),  
  Primary Key (BookingID),  
  Unique key (CheckinDate, RoomNo),  
  foreign key (CustomerID) references Customers(CustomerID)  
  on delete cascade  
  on update no action) AUTO_INCREMENT = 10000 ENGINE INNODB;
```

```
CREATE TABLE hotel.Events  
( CustomerID int NOT NULL ,  
  EventName VARCHAR(45) ,  
  Event_Room Varchar(45),  
  Max_Occ INT ,  
  StartDate date ,  
  EndDate date ,
```



```
Spl_Request VARCHAR(150) ,  
Event_Amount INT,  
Primary Key (Event_Room,StartDate),  
foreign key (CustomerID) references Customers(CustomerID)  
on delete cascade  
on update no action) ENGINE INNODB;
```

```
CREATE TABLE hotel.Dependents  
( Name VARCHAR(45) ,  
CustomerID INT Not Null,  
Age INT,  
Email VARCHAR(45) ,  
Phone bigint,  
foreign key (CustomerID) references Customers(CustomerID)  
on delete cascade  
on update no action) ENGINE INNODB;
```

```
CREATE TABLE hotel.Reservations  
( CustomerID int NOT NULL,  
BookingID int NOT NULL,  
Amount INT,  
Payment_Type Varchar(45),  
foreign key (CustomerID) references Customers(CustomerID),  
foreign key (BookingID) references Rooms(BookingID)  
on delete cascade  
on update no action) ENGINE INNODB;
```

Part 3

Integrate with Apache and CGI. The interface should be in HTML (very basic HTML is okay for getting full marks). For the CGI you can use Python or any other programming language you feel comfortable with.

Make three simple HTML pages where the user can add customers, rooms, and room reservations. Also make a fourth one which is a simple HTML page where the user can search for reservations based on any feature you want (make it as simple as you want it to be – searching with a single feature gets full). The results of the queries should be returned in tabular form (no need to use the specifics for HTML table – the result should be human readable).

HTML Pages :-

- 1) Home Page

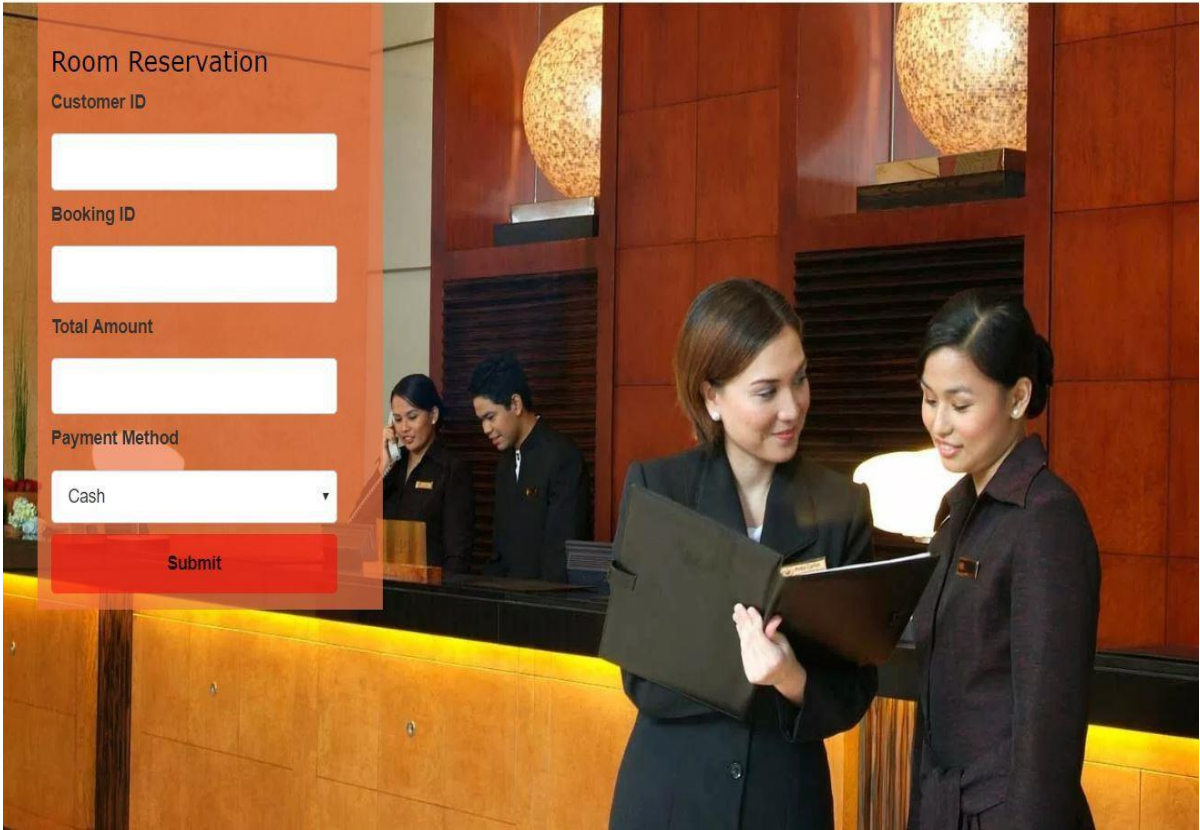
Welcome To Hotel Le Meridien

Le Meridien is an upscale, design-focused international hotel brand with a European perspective, formerly headquartered in France and in the United Kingdom. The brand was acquired by US-based Starwood Hotels & Resorts in 2005. Following the acquisition of Starwood by Marriott in 2015, it is now owned by Marriott International and has a portfolio of over 110 hotels.



2) Reservation :-

Hotel Le Meridien



Output(in Tabular form):

CustomerID	BookingID	Amount	Payment_Type
3	10000	20000	Cash
2	10001	3000	Cash
1	10001	29900	Cash

3) Add Customer :-

Hotel Le Meridien	Home	Add Customer	Add Dependents	Events	Rooms	Reservations	Search
-------------------	------	--------------	----------------	--------	-------	--------------	--------

Hotel Le Meridien

Customer form

Name:

Age:

Email:

Phone Number:

Enter Your State ID:

Enter State ID Number:

Dependents:

☐ Yes
☐ No

Submit

Output (in Tabular Form):

CustomerID	CustomerName	Age	IDType	IDNumber	Email	Phone	Dependents
1	Anand Chauhan	26	ssn	23451678	gangadhar@gmail.com	8080658672	No
2	Ruchi Moondra	30	license	3456754	ruchi@gmail.com	8624484855	Yes
3	Roman Rhodes	40	ssn	364829626	rhodes@jfk.com	6489272638	Yes
4	Etish Talreja	23	ssn	3452617	etish@gmail.com	8624484855	Yes

4) Add Dependents:

[Hotel Le Meridien](#) [Home](#) [Add Customer](#) [Add Dependents](#) [Events](#) [Rooms](#) [Reservations](#) [Search](#)

Hotel Le Meridien

Dependent Information

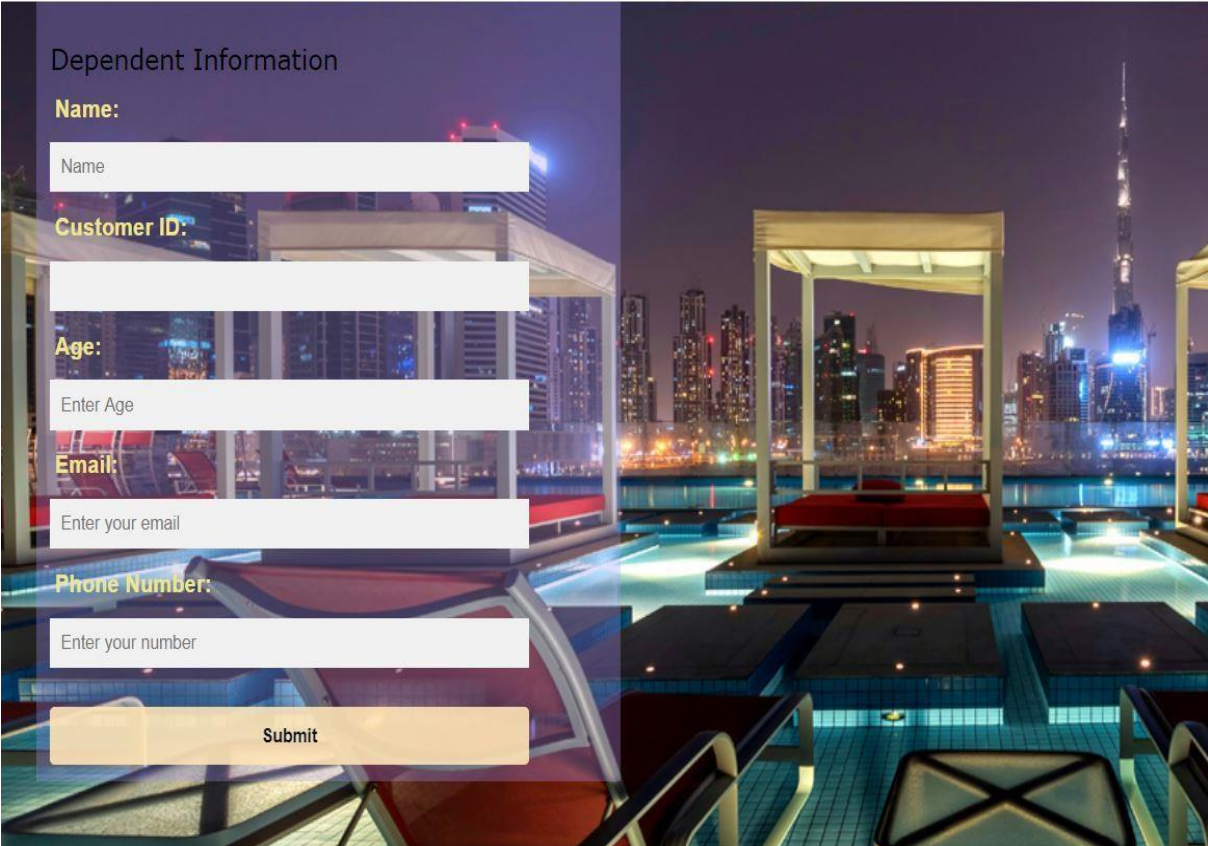
Name:

Customer ID:

Age:

Email:

Phone Number:



Output(in Tabular form):

localhost/cgi-bin/dep.py				
Name	CustomerID	Age	Email	Phone
Etish Talreja	2	1	etish@gmail.com	8624484855
Ganga	2	1	etish@gmail.com	8624484855
Orian	2	1	etish@gmail.com	8624484855
steffen	2	1	etish@gmail.com	8624484855
gangadhar	3	56	dyrduy@gmail.com	3561534
goku	3	40	dyrduy@gmail.com	3561534
vivian	3	40	dyrduy@gmail.com	3561534
Roman Rhodes	1	2	rhodes@jfk.com	6489272638

5) Add Event :-

Hotel Le Meridien
Home
Add Customer
Add Dependents
Events
Rooms
Reservations
Search

Hotel Le Meridien

Event Information

Customer ID

Event Name

Event Room

Maximum Occupancy

Start Date

End Date

Special Request

Event Amount

Submit

Output(in Tabular form):

CustomerID	EventName	Event_Room	Max_Occ	StartDate	EndDate	Spl_Request	Event_Amount
1	Wedding	10000	3	2017-11-14	None	Vegan food	4000
3	Ceremony	20	3	2017-09-13	None	Continental	2000
1	birthday	40	3	2017-02-20	None	Non-Veg	200
2	product	55	40	2017-02-28	None	Non-Veg	10000

6) Room Booking page:-

Hotel Le Meridien
Home
Add Customer
Add Dependents
Events
Rooms
Reservations
Search

Hotel Le Meridien

Room Details
Customer ID


Floor Location
☐ 1st ☐ 2nd ☐ 3rd ☐ 4th ☐ 5th
Room Number

Room Type
☐ Deluxe ☐ Suite ☐ Twin ☐ King ☐ queen
Price for selected room is:

Number of Bed's:

Parking Access
☐ YES ☐ NO
Balcony View
☐ Front view ☐ Central View ☐ Downtown View ☐ Pool-side View
Check In date

Check Out date



Output(in Tabular form):

BookingID	CustomerID	RoomNo	RoomType	Price	Floor	Parking	Balcony	CheckinDate	CheckoutDate	Beds	Duration
10000	3	101	150	150	third	YES	Central View	2017-04-12	2017-04-14	2	2
10001	2	204	150	150	second	NO	Front view	2017-04-12	2017-04-14	2	2
10002	2	303	200	200	third	NO	Central View	2017-11-14	2017-11-17	2	3

7) Search Page :-

[Hotel Le Meridien](#) [Home](#) [Add Customer](#) [Add Dependents](#) [Events](#) [Rooms](#) [Reservations](#) [Search](#)


Hotel Le Meridien

Customer ID

Submit

Max Amount for 2017

Submit



Output(in Tabular Form):

CustomerID	CustomerName	Age	IDType	IDNumber	Email	Phone	Dependents	BookingID	CustomerID	RoomNo	RoomType	Price	Floor	Parking	Balcony	CheckinDate	CheckoutDate	Beds	Duration
3	Roman Rhodes	40	ssn	364829626	rhodes@jfk.com	6489272638	Yes	10000	3	101	150	150	third	YES	Central View	2017-04-12	2017-04-14	2	2

Now Here we have the Screen shot of Python files :-

```
cust.py
1 #!C:\Users\gpiii\AppData\Local\Programs\Python\Python35\python.exe -u
2 print("Content-Type: text/html")
3 print()
4 import cgi, cgiib
5 cgiib.enable() ifor debugging
6 form = cgi.FieldStorage()
7 fname = form.getvalue('name')
8 age = form.getvalue('age')
9 idtype = form.getvalue('idtype')
10 idno = form.getvalue('idnumber')
11 email = form.getvalue('email')
12 phone = form.getvalue('phone')
13 depend = form.getvalue('depend')
14 import pymysql
15 db = pymysql.connect("localhost","root","Ganga@123","hotel")
16 cursor = db.cursor()
17 sql = "insert into customers (CustomerName, Age, IDType, IDNumber, Email, Phone, Dependents) VALUES (%s, %s, %s, %s, %s, %s, %s)"
18 cursor.execute(sql, (fname, age, idtype, idno, email, phone, depend))
19 db.commit()
20 cursor.execute("SELECT * FROM customers;")
21 data = cursor.fetchall()
22 field_names = [i[0] for i in cursor.description]
23 print("<table border-collapse: collapse; width: 100%; border: 1px solid #
24 #dddddd; text-align: left; padding: 8px; > tr>th-child(even) { background-color: #dddddd; } </style>")
25 print("<table><tr>")
26 for columns in field_names:
27     print("<th> %s </th>" % columns)
28 print("</tr>")
29 for rows in data:
30     print("<tr>")
31     for subrows in rows:
32         print("<td> %s </td>" % subrows)
33     print("</tr>")
34 print("</table>")
35 db.commit()
```

```
Apache24 cgi-bin dep.py
1 #!C:\Users\gpiii\AppData\Local\Programs\Python\Python35\python.exe -u
2 print("Content-Type: text/html")
3 print()
4 import cgi, cgiib
5 cgiib.enable() ifor debugging
6 form = cgi.FieldStorage()
7 fname = form.getvalue('Name')
8 cus_id = form.getvalue('customerid')
9 age = form.getvalue('age')
10 email = form.getvalue('email')
11 phone = form.getvalue('phone')
12 import pymysql
13 db = pymysql.connect("localhost","root","Ganga@123","hotel")
14 cursor = db.cursor()
15 sql = "insert into dependents (Name, CustomerID, Age, Email, Phone) VALUES (%s, %s, %s, %s, %s)"
16 cursor.execute(sql, (fname, cus_id, age, email, phone))
17 db.commit()
18 cursor.execute("SELECT * FROM dependents;")
19 data = cursor.fetchall()
20 print(data)
21 sql = ("firstname", "lastname", "CustomerID", "age", "email", "phone")
22 #data = array(data)
23 from fields = len(data)
24 field_names = [i[0] for i in cursor.description]
25 print("<table border-collapse: collapse; width: 100%; border: 1px solid #
26 #dddddd; text-align: left; padding: 8px; > tr>th-child(even) { background-color: #dddddd; } </style>")
27 print("<table><tr>")
28 for columns in field_names:
29     print("<th> %s </th>" % columns)
30 print("</tr>")
31 for rows in data:
32     print("<tr>")
33     for subrows in rows:
34         print("<td> %s </td>" % subrows)
35     print("</tr>")
36 print("</table>")
37 # Fetch all the rows using fetchall() method.
38 db.commit()
```



```
Apache24  cgi-bin  rom.py  Add Configuration...
rom.py x  que.py x  dep.py x  eve.py x  rev.py x  cust.py x  Question.py x
1  #!C:\Users\gpril\AppData\Local\Programs\Python\Python35\python.exe -u
2  print("Content-Type: text/html")
3  print()
4  import cgi, cgitb
5  cgitb.enable() #for debugging
6  form = cgi.FieldStorage()
7  cus_id = form.getvalue('cusid')
8  roomno = form.getvalue('Room_No')
9  roomtype = form.getvalue('roomtype')
10 no_of_beds = form.getvalue('no_of_beds')
11 price = form.getvalue('price')
12 floor = form.getvalue('floor')
13 parking = form.getvalue('parking')
14 balcony = form.getvalue('balcony')
15 chkin = form.getvalue('startdate')
16 chkout = form.getvalue('enddate')
17 duration = form.getvalue('ndays')
18 import pymysql
19 db = pymysql.connect("localhost","root","Ganga@123","hotel")
20 cursor = db.cursor()
21 sql = "insert into rooms (CustomerID, RoomNo, RoomType, Price, Floor, Parking, Balcony, CheckinDate, CheckoutDate, Beds, Duration) VALUES ("
22      "(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
23      cursor.execute(sql,(cus_id, roomno, roomtype, price, floor, parking, balcony, chkin, chkout, no_of_beds, duration))
24      db.commit()
25      cursor.execute("SELECT * FROM rooms")
26      data = cursor.fetchall()
27      #print (data)
28      #cols=(('firstname', 'lastname', 'CustomerID', 'age', 'email', 'phone'))
29      #dataToPrint=array(data)
30      #num_fields = len(data)
31      field_names = [i[0] for i in cursor.description]
32      print("<style>table { font-family: arial, sans-serif; border-collapse: collapse; width: 100%; text-align: left; border: 1px solid #000; }</style>")
33      print("<table><tr>")
34      for columns in field_names:
35          print("<th>%s</th>" % columns)
36      print("</tr>")
37      for rows in data:
38          print("<tr>")
```

```
Apache24  cgi-bin  Question.py  Add Configuration...
rom.py x  que.py x  dep.py x  eve.py x  rev.py x  cust.py x  Question.py x
1  #!C:\Users\gpril\AppData\Local\Programs\Python\Python35\python.exe -u
2  print("Content-Type: text/html")
3  print()
4  import cgi, cgitb
5  cgitb.enable() #for debugging
6  form = cgi.FieldStorage()
7  import pymysql
8  # Open connection to the database
9  db = pymysql.connect("localhost","root","Ganga@123","hotel")
10 # Start a cursor object using cursor() method
11 cursor = db.cursor()
12 # Execute a SQL query using execute() method. This should return all the columns of heroes that use swords.
13 sql = "insert into reservations (CustomerID, BookingID, Amount, PaymentType) VALUES (%s,%s,%s,%s)"
14 #cursor.execute(sql,(cus_id, booking_id, totamt, paymethod))
15 db.commit()
16 x = "select * from customers c, rooms r where c.CustomerID = r.CustomerID and r.CustomerID = (select r1.CustomerID from reservations r1 where r1.Amount in (select max(Amount) from reservations))"
17 cursor.execute(x)
18 data = cursor.fetchall()
19 # c.CustomerID, c.CustomerName, c.IDNumber, c.Phone, r.RoomNo, r.RoomType, r.Price, r.Duration
20 #print (data)
21 #cols=(('firstname', 'lastname', 'CustomerID', 'age', 'email', 'phone'))
22 #dataToPrint=array(data)
23 #num_fields = len(data)
24 field_names = [i[0] for i in cursor.description]
25 print("<style>table { font-family: arial, sans-serif; border-collapse: collapse; width: 100%; text-align: left; border: 1px solid #000; text-align: left; padding: 8px; }</style>")
26 print("<table><tr>")
27 for columns in field_names:
28     print("<th>%s</th>" % columns)
29 print("</tr>")
30 for rows in data:
31     print("<tr>")
32     for subrows in rows:
33         print("<td>%s</td>" % subrows)
34     print("</tr>")
35 print("</table>")
36 # Fetch all the rows using fetchall() method.
37
38 db.commit()
```