

# Hotel Database

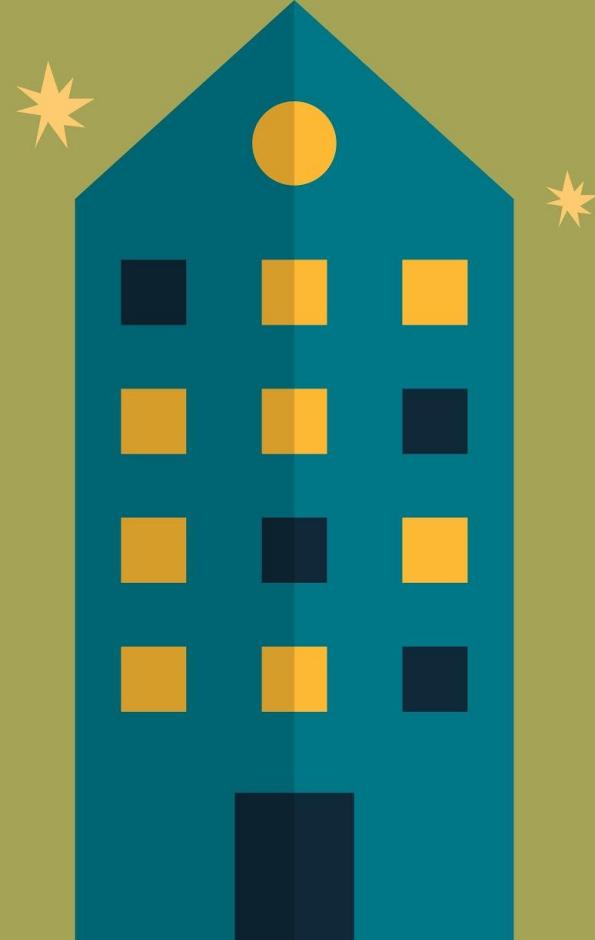
**Management System**



# Main objective

to create a hotel management system

This software is aimed at facilitating the prompt, accurate and easy management of a hotel. It will be very helpful to the manager on the one hand and to the customers as they would be able to get good and fast service.



# Hotel Management System

A Hotel consist of a wide areas to manage. We tried to include the main areas for a hotel management system in this project.



Rooms



Bookings

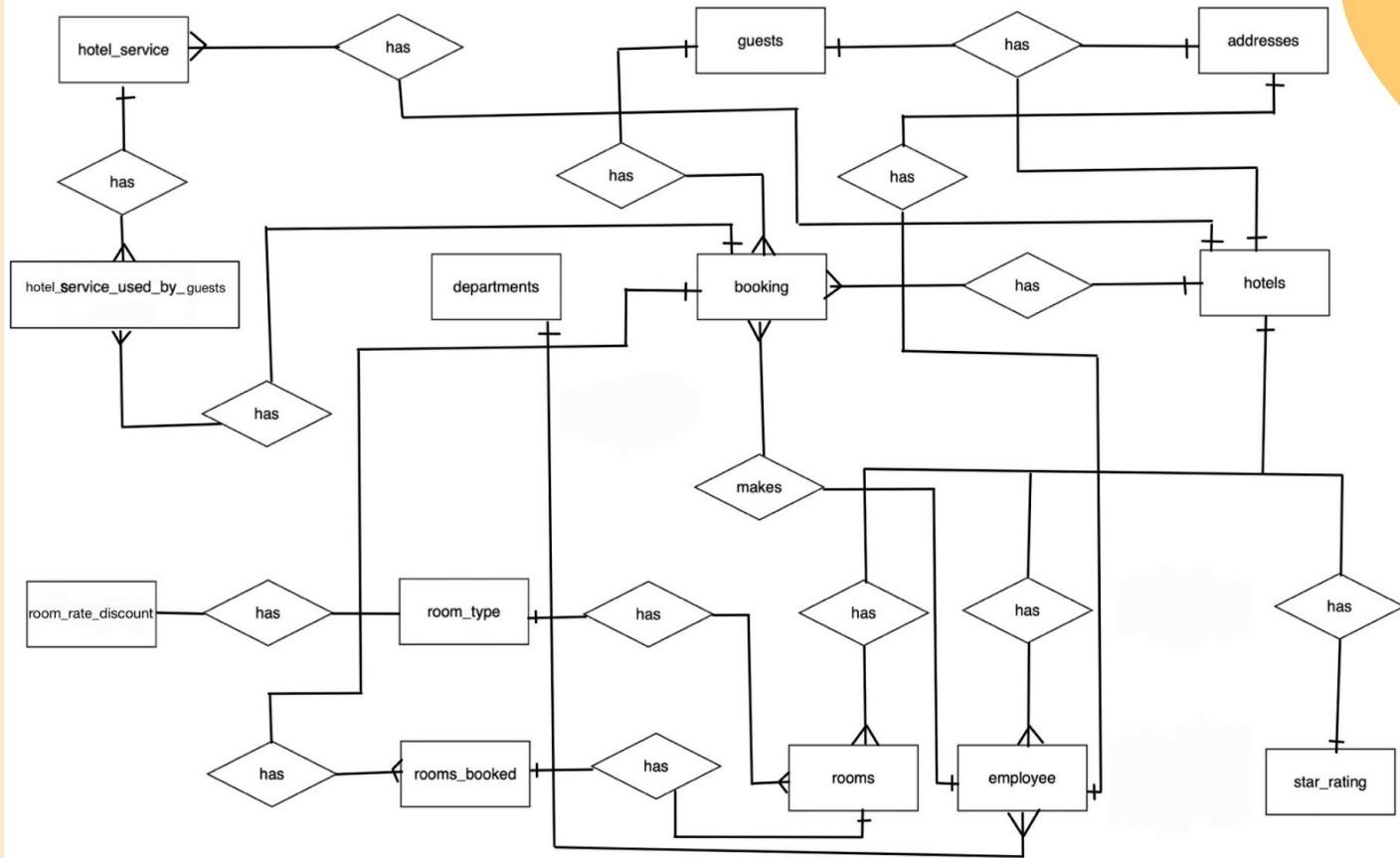


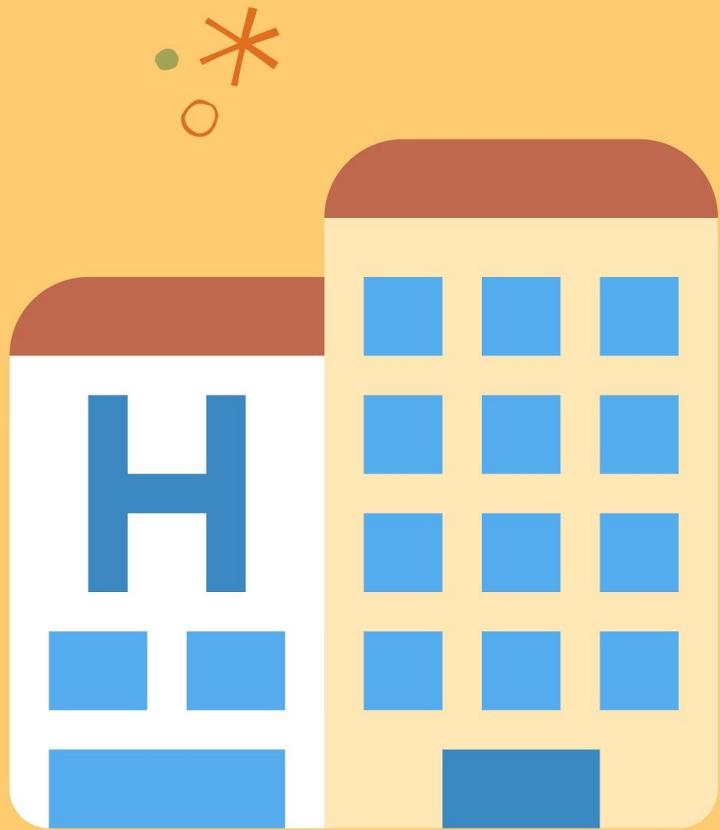
Employees



Guests







# Schema

addresses(address\_id, address\_line1, address\_line2, city, state, country, zip code)

star\_rating(star\_rating)

hotel(hotel\_id, name, contact\_number, email\_address, website, description, floor\_count, room\_capacity, address\_id,  
star\_rating, check\_in\_time, check\_out\_time)

FK(hotel.address\_id, addresses.address\_id)

FK(hotel.star\_rating, star\_rating.star\_rating)

room\_type(room\_type\_id, name, cost, description, smoke\_friendly, pet\_friendly)

rooms(room\_id, room\_number, room\_type\_id, hotel\_id)

FK(rooms.room\_type\_id, room\_type.room\_type\_id)

FK(rooms.hotel\_id, hotel.hotel\_id)

guests(guest\_id, first\_name, last\_name, contact\_number, email\_address, address\_id)

FK(guests.address\_id, addresses.address\_id)

department(department\_id, name, description)

employees(employee\_id, first\_name, last\_name, designation, contact\_number, email\_address, department\_id, address\_id,  
hotel\_id)

FK(employees.department\_id, department.department\_id)

FK(employees.address\_id, addresses.address\_id)

FK(employees.hotel\_id, hotel.hotel\_id)

`bookings(booking_id, booking_date, duration_of_stay, check_in_date, check_out_date, booking_payment_type, total_rooms_booked, hotel_id, guest_id, employee_id, total_amount)`

`FK(booking.hotel_id, hotel.hotel_id)`

`FK(booking.guest_id, guest.guest_id)`

`FK(booking.employee_id, employees.employee_id)`

`room_rate_discount(discount_id, discount_rate, start_month, end_month, room_type_id)`

`FK(room_rate_discount.room_type_id, room_type.room_type_id)`

`rooms_booked(rooms_booked_id, booking_id, room_id)`

`FK(room_booked.booking_id, booking.booking_id)`

`FK(room_booked.room_id, rooms.room_id)`

`hotel_services(service_id, service_name, service_description, service_cost, hotel_id)`

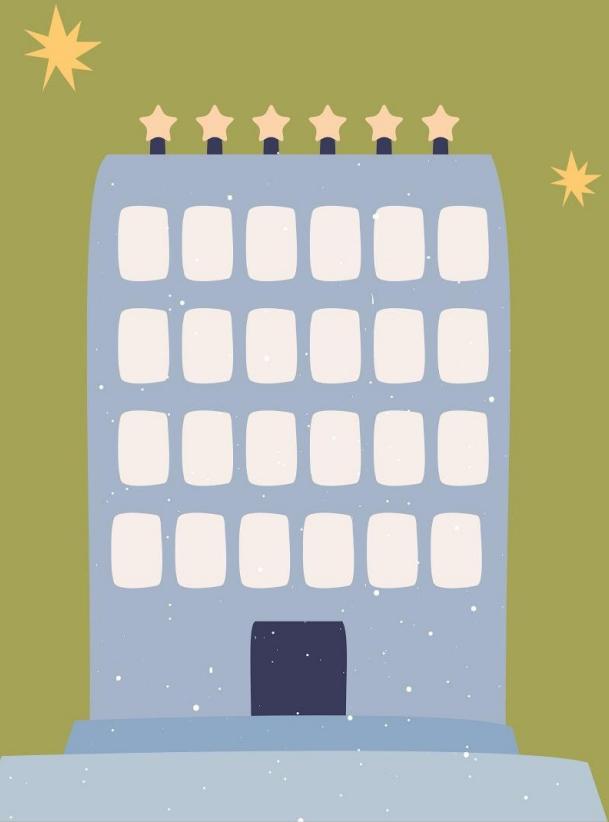
`FK(hotel_service.hotel_id, hotel.hotel_id)`

`hotel_service_used_by_guests(service_used_id, service_id, booking_id)`

`FK(hotel_service_used_by_guests.service_id, hotel_services.service_id)`

`FK(hotel_service_used_by_guests.booking_id, bookings.booking_id)`

# Normalization



**1NF:**

hotel\_management\_system (address\_id, address\_line1, address\_line2, city, state, country, zipcode, star\_rating, hotel\_id, hotel\_name, hotel\_contact\_number, hotel\_email\_address, website, hotel\_description, floor\_count, room\_capacity, check\_in\_time, check\_out\_time, room\_type\_id, room\_type\_name, cost, description, smoke\_friendly, pet\_friendly, room\_id, room\_number, guest\_id, guest\_first\_name, guest\_last\_name, guest\_contact\_number, guest\_email\_address, department\_id, department\_name, department\_description, employee\_id, employee\_first\_name, employee\_last\_name, designation, employee\_contact\_number, employee\_email\_address, booking\_id, booking\_date, duration\_of\_stay, check\_in\_date, check\_out\_date, booking\_payment\_type, total\_rooms\_booked, total\_amount, discount\_id, discount\_rate, start\_month, end\_month, rooms\_booked\_id, service\_id, service\_name, service\_description, service\_cost, serviced\_used\_id)

**FD:**

hotel\_id, room\_id → name, contact\_number, email\_address, website, description, floor\_count, room\_capacity, address\_id, star\_rating, check\_in\_time, check\_out\_time, room\_number, rooms\_type\_id, hotel\_id, room\_type\_id, name, cost, description, smoke\_friendly, pet\_friendly

Hotel\_id → name, contact\_number, email\_address, website, description, floor\_count, room\_capacity, address\_id, star\_rating, check\_in\_time, check\_out\_time

Room\_id → room\_number, rooms\_type\_id, hotel\_id, room\_type\_id, name, cost, description, smoke\_friendly, pet\_friendly

## 2NF:

hotel\_management\_system (hotel\_id, room\_id, star\_rating, serviced\_used\_id)

Hotel\_Info(hotel\_id, name, contact\_number, email\_address, website, description, floor\_count, room\_capacity, address\_id, star\_rating, check\_in\_time, check\_out\_time)

Room\_Info(room\_id, room\_number, rooms\_type\_id, hotel\_id, room\_type\_id, name, cost, description, smoke\_friendly, pet\_friendly)

## FD:

Hotel\_id → name, contact\_number, email\_address, website, description, floor\_count, room\_capacity, address\_id, star\_rating, check\_in\_time, check\_out\_time

Name → contact\_number

Contact\_number → email\_address

Email\_address → website

Website → description

Description → floor\_count

Floor\_count → room\_capacity

Room\_capacity → address\_id

Address\_id → check\_in\_time

Check\_in\_time → check\_out\_time

Address\_id → address\_line1, address\_line2, city, state, country, zipcode

Address\_line1 → address\_line2

Address\_line2 → city

City → state

State → country

Country → zip code

Room\_id → room\_number, rooms\_type\_id, hotel\_id, name, cost, description, smoke\_friendly, pet\_friendly

Room\_number → rooms\_type\_id

Rooms\_type\_id → hotel\_id

Hotel\_id → name

Name → cost

Cost → description

Description → smoke\_friendly

Smoke\_friendly → pet\_friendly

Guest\_id → guest\_first\_name, guest\_last\_name, guest\_contact\_number, guest\_email\_address, department\_id, department\_name, department\_description

Guest\_first\_name → guest\_last\_name

Guest\_last\_name → guest\_contact\_number

Guest\_contact\_number → guest\_email\_address

Guest\_email\_address → department\_id

Department\_id → department\_name

Department\_name → department\_description

Employee\_id → employee\_first\_name, employee\_last\_name, designation, employee\_contact\_number, employee\_email\_address

Employee\_first\_name → employee\_last\_name

Employee\_last\_name → designation

Designation → employee\_contact\_number

Employee\_contact\_number → employee\_email\_address

Booking\_id → booking\_date, duration\_of\_stay, check\_in\_date, check\_out\_date, booking\_payment\_type, total\_rooms\_booked, total\_amount

Booking\_date → duration\_of\_stay

Duration\_of\_stay → check\_in\_date

Check\_in\_date → check\_out\_date

Check\_out\_date → booking\_payment\_type

Booking\_payment\_type → total\_rooms\_booked

Total\_rooms\_booked → total\_amount

Discount\_id → discount\_rate, start\_month, end\_month, rooms\_booked\_id

Discount\_rate → start\_month

Start\_month → end\_month

End\_month → rooms\_booked\_id

Service\_id → service\_name, service\_description, service\_cost, serviced\_used\_id

Service\_name → service\_description

Service\_description → service\_cost

Service\_cost → serviced\_used\_id

### 3NF:

addresses(address\_id, address\_line1, address\_line2, city, state, country, zipcode)

star\_ratings(star\_rating)

hotel(hotel\_id, hotel\_name, hotel\_contact\_number, hotel\_email\_address, website, hotel\_description, floor\_count, room\_capacity, check\_in\_time, check\_out\_time)

room\_type(room\_type\_id, room\_type\_name, cost, description, smoke\_friendly, pet\_friendly)

rooms( room\_id, room\_number)

guests(guest\_id, guest\_first\_name, guest\_last\_name, guest\_contact\_number, guest\_email\_address)

department(department\_id, department\_name, department\_description)

employees(employee\_id, employee\_first\_name, employee\_last\_name, designation, employee\_contact\_number, employee\_email\_address)

bookings(booking\_id, booking\_date, duration\_of\_stay, check\_in\_date, check\_out\_date, booking\_payment\_type, total\_rooms\_booked, total\_amount)

room\_rate\_discount(discount\_id, discount\_rate, start\_month, end\_month)

room\_booked(rooms\_booked\_id)

hotel\_services(service\_id, service\_name, service\_description, service\_cost)

hotel\_services\_used\_by\_guests(serviced\_used\_id)

# Programming to database

---



```
1 • - CREATE TABLE addresses (
2     address_id INT ,
3     address_line1 VARCHAR(80),
4     address_line2 VARCHAR(80),
5     city VARCHAR(60),
6     state VARCHAR(60),
7     country VARCHAR(60),
8     zipcode VARCHAR(10),
9     PRIMARY KEY (address_id)
10 );
11
12 • - CREATE TABLE star_ratings (
13     star_rating INT,
14     PRIMARY KEY (star_rating))
```

```
INSERT INTO addresses (address_id, address_line1, address_line2, city, state, country, zipcode) VALUES  
    (1,1106, 'San Martin', 'Lujan de Cuyo','Mendoza','Argentina',54358),  
    (2,111, 'Supalai Village', 'Bangna', 'Bangkok', 'Thailand', '10260'),  
    (3,7598, 'Atomic Street', 'Buffalo', 'New York', 'USA', 145),  
    (4,45, 'Nawamin Road', 'Bang Kapi', 'Bangkok', 'Thailand', '10240'),  
    (5,879, 'Lock Street', 'Galt', 'Ontario', 'Canada', 'N1R 3Y8'),  
    (6,4312,'Eglinton Avenue', 'Toronto', 'Ontario', 'Canada', 'M4P 1A6'),  
    (7,87, 'Sainr Germain', 'Gagny', 'lle-de-France', 'France', 77330),  
    (8,1565, 'E South St', 'Globe', 'AZ', 'USA',85501),  
    (9,161, 'Hannah Road', 'Victoria', 'Rotorua', 'New Zealand',3010),  
    (10,240, 'Byron Street', 'Hohonu', 'Grey', 'New Zealand',7805),  
    (11,220, 'House Village', 'Chonburi', 'Chonburi', 'Thailand', '20000'),  
    (12,33, 'Kaoseng-Chana Road', 'Muang Songkhla', 'Songkhla', 'Thailand',90000),  
    (13,408, 'Nakoenchaisi Road', 'Dusit', 'Bangkok', 'Thailand',10500),  
    (14,191, 'Ratchaparop Road', 'Bangkok', 'Bangkok', 'Thailand',10400),  
    (15,3767, 'Yaowarat Road', 'Muang', 'Phuket', 'Thailand',83000),  
    (16,135, 'Moo 3 Khlongyai', 'Trat', 'Trat', 'Thailand', '23110'),  
    (17,642, 'Phetchaburi Road', 'Ratchathewi', 'Bangkok', 'Thailand',10400),  
    (18,284, 'Bangla Road', 'Phuket', 'Phuket', 'Thailand',83150),  
    (19,101, 'Pang-Nga Road', 'Muang', 'Phuket', 'Thailand',83000),  
    (20,50, 'Bandsaen Sai 1', 'Chonburi', 'Chonburi', 'Thailand',20130),  
    (21,2257, 'Cha-am Beach Road', 'Cha-am', 'Phetchaburi', 'Thailand',76120),  
    (22,771, 'Bang Sarae', 'Sattahip', 'Chonburi', 'Thailand',20250),  
    (23,107, 'Petchkasem Road', 'Hua Hin', 'Prachuap Khiri Khan', 'Thailand',77110),  
    (24,333, 'Beach Road', 'Bang Lamung', 'Chonburi', 'Thailand',20260),  
    (25,99, 'Muen-Ngoen Road', 'Phuket', 'Phuket', 'Thailand',83150),  
    (26,389, 'Bann Klong Prao', 'Ko Chang', 'Trat', 'Thailand',23170),
```

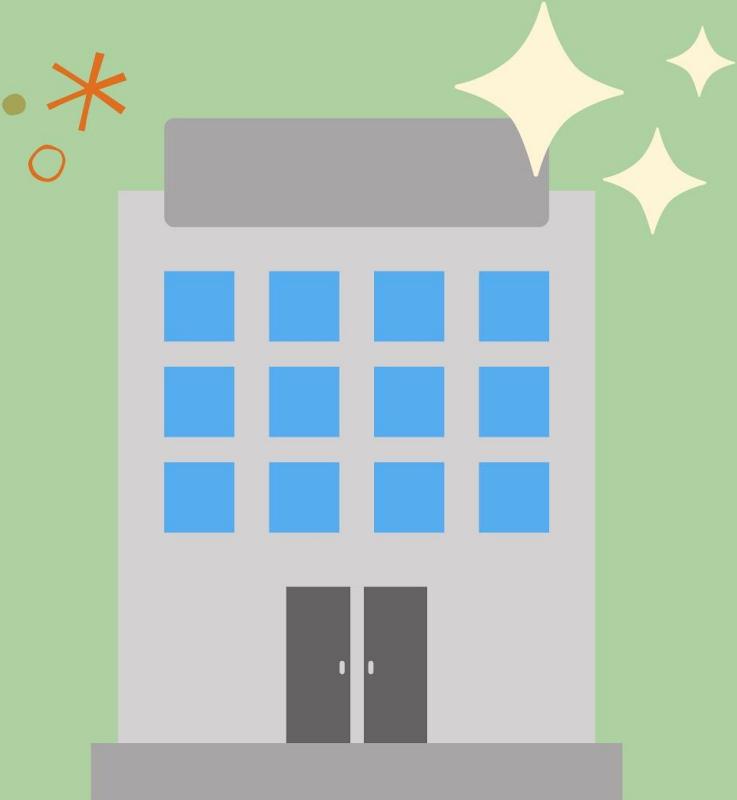


```
import mysql.connector
mydb = mysql.connector.connect(
    host = "localhost",
    user = "root",
    password = "",
    database = "hotel"
)

mycursor = mydb.cursor ()
sql = "Select * from addresses"
mycursor.execute(sql)
myresult = mycursor.fetchall()
for x in myresult :
    print(x)
```



```
 7 from tkinter import *
 8 import sqlite3
 9
10 win=Tk()
11 def set_text():
12     conn = sqlite3.connect('test.db')
13     print ("Opened database successfully")
14     cursor = conn.execute("SELECT booking_id, booking_date, check_in_date, check_out_date, booking_payment_type, hotel_id, guest_id, total_amount from bookings where booking_id=?")
15     for row in cursor:
16         txt1.delete(0,END)
17         txt1.insert(0,str(row[0]))
18         txt2.delete(0,END)
19         txt2.insert(0,str(row[1]))
20         txt3.delete(0,END)
21         txt3.insert(0,str(row[2]))
22         txt4.delete(0,END)
23         txt4.insert(0,str(row[3]))
24         txt5.delete(0,END)
25         txt5.insert(0,str(row[4]))
26         txt6.delete(0,END)
27         txt6.insert(0,str(row[5]))
28         txt7.delete(0,END)
29         txt7.insert(0,str(row[6]))
30         txt8.delete(0,END)
31         txt8.insert(0,str(row[7]))
32         label1['text']=str(row[0])
33     conn.close()
34     return
35
36 def updateval():
37     conn = sqlite3.connect('test.db')
38     s=str(txt1.get())
39     booking_date=str(txt2.get())
40     check_in_date=str(txt3.get())
41     check_out_date=str(txt4.get())
42     booking_payment_type=str(txt5.get())
43     hotel_id=str(txt6.get())
44     guest_id=str(txt7.get())
45     total_amount=str(txt8.get())
46     cursor=conn.cursor()
47     cursor.execute("update bookings set booking_date=?,check_in_date=?,check_out_date=?,booking_payment_type=?,hotel_id=?, guest_id=?,total_amount=? where booking_id=?",( booking_id,s,check_in_date,check_out_date,booking_payment_type,hotel_id,guest_id,total_amount))
48     conn.commit()
```



# Results

# ADDRESS :

address_id	address_line1	address_line2	city	state	country	zipcode
1	1106	San Martin	Lujan de Cuyo	Mendoza	Argentina	54358
2	111	Supalai Village	Bangna	Bangkok	Thailand	10260
3	7598	Atomic Street	Buffalo	New York	USA	145
4	45	Nawamin Road	Bang Kapi	Bangkok	Thailand	10240
5	879	Lock Street	Galt	Ontario	Canada	N1R 3Y8
6	4312	Eglinton Avenue	Toronto	Ontario	Canada	M4P 1A6
7	87	Sainr Germain	Gagny	Ile-de-France	France	77330
8	1565	E South St	Globe	AZ	USA	85501
9	161	Hannah Road	Victoria	Rotorua	New Zealand	3010
10	240	Byron Street	Hohonu	Grey	New Zealand	7805
11	220	House Village	Chonburi	Chonburi	Thailand	20000
12	33	Kaoseng-Chan...	Muang Song...	Songkhla	Thailand	90000
13	408	Nakoenchaisi...	Dusit	Bangkok	Thailand	10500
14	191	Ratchaparop R...	Bangkok	Bangkok	Thailand	10400
15	3767	Yaowarat Road	Muang	Phuket	Thailand	83000
16	135	Moo 3 Khlongyai	Trat	Trat	Thailand	23110
17	642	Phetchaburi R...	Ratchathewi	Bangkok	Thailand	10400
18	284	Bangla Road	Phuket	Phuket	Thailand	83150
► 19	101	Pang-Nga Road	Muang	Phuket	Thailand	83000
20	50	Bandsaen Sai 1	Chonburi	Chonburi	Thailand	20130
21	2257	Cha-am Beach...	Cha-am	Phetchaburi	Thailand	76120
22	771	Bang Sarae	Sattahip	Chonburi	Thailand	20250
23	107	Petchkasem R...	Hua Hin	Prachuap K...	Thailand	77110
24	333	Beach Road	Bang Lamung	Chonburi	Thailand	20260
25	99	Muen-Ngoen...	Phuket	Phuket	Thailand	83150
26	389	Bann Klong Prao	Ko Chang	Trat	Thailand	23170
27	416	Nathon Road	Koh Samui	Surat Thani	Thailand	84140
28	666	Koh Road	Bang Lamung	Chonburi	Thailand	20150
29	995	Pae-Klaeng-Kr...	Klaeng	Rayong	Thailand	21190
30	820	Sukprayoon R...	Chonburi	Chonburi	Thailand	20000
31	433	Phetch Road	Muang Song...	Songkhla	Thailand	90000
32	508	Nakoenchaisi...	Dusit	Bangkok	Thailand	10500
33	991	Ratchaparop R...	Bangkok	Bangkok	Thailand	10400
34	867	Ya Road	Muang	Phuket	Thailand	83000
35	195	Khlongyai Road	Trat	Trat	Thailand	23110

```
In [2]: runfile('D:/2year/table.py', wdir='D:/2year')
(1, '1106', 'San Martin', 'Lujan de Cuyo', 'Mendoza', 'Argentina', '54358')
(2, '111', 'Supalai Village', 'Bangna', 'Bangkok', 'Thailand', '10260')
(3, '7598', 'Atomic Street', 'Buffalo', 'New York', 'USA', '145')
(4, '45', 'Nawamin Road', 'Bang Kapi', 'Bangkok', 'Thailand', '10240')
(5, '879', 'Lock Street', 'Galt', 'Ontario', 'Canada', 'N1R 3Y8')
(6, '4312', 'Eglinton Avenue', 'Toronto', 'Ontario', 'Canada', 'M4P 1A6')
(7, '87', 'Sainr Germain', 'Gagny', 'Ile-de-France', 'France', '77330')
(8, '1565', 'E South St', 'Globe', 'AZ', 'USA', '85501')
(9, '161', 'Hannah Road', 'Victoria', 'Rotorua', 'New Zealand', '3010')
(10, '240', 'Byron Street', 'Hohonu', 'Grey', 'New Zealand', '7805')
(11, '220', 'House Village', 'Chonburi', 'Chonburi', 'Thailand', '20000')
(12, '33', 'Kaoseng-Chana Road', 'Muang Songkhla', 'Songkhla', 'Thailand', '90000')
(13, '408', 'Nakoenchaisi Road', 'Dusit', 'Bangkok', 'Thailand', '10500')
(14, '191', 'Ratchaparop Road', 'Bangkok', 'Bangkok', 'Thailand', '10400')
(15, '3767', 'Yaowarat Road', 'Muang', 'Phuket', 'Thailand', '83000')
(16, '135', 'Moo 3 Khlongyai', 'Trat', 'Trat', 'Thailand', '23110')
(17, '642', 'Phetchaburi Road', 'Ratchathewi', 'Bangkok', 'Thailand', '10400')
(18, '284', 'Bangla Road', 'Phuket', 'Phuket', 'Thailand', '83150')
(19, '101', 'Pang-Nga Road', 'Muang', 'Phuket', 'Thailand', '83000')
(20, '50', 'Bandsaen Sai 1', 'Chonburi', 'Chonburi', 'Thailand', '20130')
(21, '2257', 'Cha-am Beach Road', 'Cha-am', 'Phetchaburi', 'Thailand', '76120')
(22, '771', 'Bang Sarae', 'Sattahip', 'Chonburi', 'Thailand', '20250')
(23, '107', 'Petchkasem Road', 'Hua Hin', 'Prachuap Khiri Khan', 'Thailand', '77110')
(24, '333', 'Beach Road', 'Bang Lamung', 'Chonburi', 'Thailand', '20260')
(25, '99', 'Muen-Ngoen Road', 'Phuket', 'Phuket', 'Thailand', '83150')
(26, '389', 'Bann Klong Prao', 'Ko Chang', 'Trat', 'Thailand', '23170')
(27, '416', 'Nathon Road', 'Koh Samui', 'Surat Thani', 'Thailand', '84140')
(28, '666', 'Koh Road', 'Bang Lamung', 'Chonburi', 'Thailand', '20150')
(29, '995', 'Pae-Klaeng-Kram Road', 'Klaeng', 'Rayong', 'Thailand', '21190')
(30, '820', 'Sukprayoon Road', 'Chonburi', 'Chonburi', 'Thailand', '20000')
(31, '433', 'Phetch Road', 'Muang Songkhla', 'Songkhla', 'Thailand', '90000')
(32, '508', 'Nakoenchaisi Road', 'Dusit', 'Bangkok', 'Thailand', '10500')
(33, '991', 'Ratchaparop Road', 'Bangkok', 'Bangkok', 'Thailand', '10400')
(34, '867', 'Ya Road', 'Muang', 'Phuket', 'Thailand', '83000')
(35, '195', 'Khlongyai Road', 'Trat', 'Trat', 'Thailand', '23110')
```

# BOOKING :

```
In [3]: runfile('D:/2year/table.py', wdir='D:/2year')
(1, datetime.datetime(2020, 10, 26, 17, 30, 24), '3', datetime.datetime(2020, 10, 31, 14, 0), datetime.datetime(2020, 11, 3, 12, 0), 'cash', 1, 3, 1, 2, Decimal('7500.00'))
(2, datetime.datetime(2020, 10, 26, 21, 58, 6), '2', datetime.datetime(2020, 10, 31, 14, 0), datetime.datetime(2020, 11, 2, 12, 0), 'card', 1, 3, 2, 4, Decimal('19000.00'))
(3, datetime.datetime(2020, 10, 27, 5, 52, 52), '1', datetime.datetime(2020, 10, 31, 14, 0), datetime.datetime(2020, 11, 1, 12, 0), 'card', 1, 3, 3, 1, Decimal('7500.00'))
(4, datetime.datetime(2020, 10, 28, 12, 0), '4', datetime.datetime(2020, 11, 1, 14, 0), datetime.datetime(2020, 11, 5, 12, 0), 'card', 1, 3, 4, 1, Decimal('22000.00'))
(5, datetime.datetime(2020, 10, 29, 14, 10, 20), '2', datetime.datetime(2020, 11, 1, 14, 0), datetime.datetime(2020, 11, 3, 12, 0), 'card', 1, 3, 5, 2, Decimal('9000.00'))
(6, datetime.datetime(2020, 10, 31, 15, 40), '1', datetime.datetime(2020, 11, 3, 14, 0), datetime.datetime(2020, 11, 4, 12, 0), 'card', 1, 3, 6, 2, Decimal('3500.00'))
(7, datetime.datetime(2020, 10, 31, 19, 10), '5', datetime.datetime(2020, 11, 12, 14, 0), datetime.datetime(2020, 11, 17, 12, 0), 'cash', 2, 3, 7, 4, Decimal('37500.00'))
(8, datetime.datetime(2020, 10, 31, 19, 12, 30), '3', datetime.datetime(2020, 11, 6, 14, 0), datetime.datetime(2020, 11, 9, 12, 0), 'card', 1, 3, 8, 2, Decimal('13500.00'))
(9, datetime.datetime(2020, 10, 31, 12, 48, 12), '4', datetime.datetime(2020, 11, 12, 14, 0), datetime.datetime(2020, 11, 16, 12, 0), 'card', 1, 3, 9, 4, Decimal('10000.00'))
(10, datetime.datetime(2020, 11, 1, 17, 40), '2', datetime.datetime(2020, 11, 5, 14, 0), datetime.datetime(2020, 11, 7, 12, 0), 'cash', 2, 3, 10, 4, Decimal('7000.00'))
```

booking_id	booking_date	duration_of_stay	check_in_date	check_out_date	booking_payment_ty...	total_rooms_booked	hotel_id	guest_id	employee_id	total_amou
1	2020-10-26 17:30:24	3	2020-10-31 14:00:00	2020-11-03 12:00:00	cash	1	3	1	2	7500.00
2	2020-10-26 21:58:06	2	2020-10-31 14:00:00	2020-11-02 12:00:00	card	1	3	2	4	19000.00
3	2020-10-27 05:52:52	1	2020-10-31 14:00:00	2020-11-01 12:00:00	card	1	3	3	1	7500.00
4	2020-10-28 12:00:00	4	2020-11-01 14:00:00	2020-11-05 12:00:00	card	1	3	4	1	22000.00
5	2020-10-29 14:10:20	2	2020-11-01 14:00:00	2020-11-03 12:00:00	card	1	3	5	2	9000.00
6	2020-10-31 15:40:00	1	2020-11-03 14:00:00	2020-11-04 12:00:00	card	1	3	6	2	3500.00
7	2020-10-31 19:10:00	5	2020-11-12 14:00:00	2020-11-17 12:00:00	cash	2	3	7	4	37500.00
8	2020-10-31 19:12:30	3	2020-11-06 14:00:00	2020-11-09 12:00:00	card	1	3	8	2	13500.00
9	2020-10-31 12:48:12	4	2020-11-12 14:00:00	2020-11-16 12:00:00	card	1	3	9	4	10000.00
10	2020-11-01 17:40:00	2	2020-11-05 14:00:00	2020-11-07 12:00:00	cash	2	3	10	4	7000.00



## GUEST :

```
In [129]: runfile('D:/untitled20.py', wdir='D:')

(1, 'George', 'Weasley', '093-8362937', 'George.w@gmail.com', 1)
(2, 'Harry', 'Potter', '086-8363529', 'Harry.p@gmail.com', 2)
(3, 'Susan', 'Bones', '089-8273635', 'Susan.b@gmail.com', 3)
(4, 'Sasipim', 'Kunaruk', '080-897375', 'sasipim.pim@gmail.com', 4)
(5, 'Severus', 'Snape', '082-8452937', 'Severus.s@gmail.com', 5)
(6, 'Dudly ', 'Dursley', '097-3946274', 'Dudly.d@gmail.com', 6)
(7, 'Draco', 'Malfoy', '085-9365284', 'Draco.m@gmail.com', 7)
(8, 'Tom', 'Riddle', '092-3249642', 'Tom.r@gmail.com', 8)
(9, 'Newt', 'Scamander', '095-3946294', 'Newt.s@gmail.com', 9)
(10, 'Hermione', 'Granger', '080-0284643', 'Hermione.g@gmail.com', 10)
```

booking_id	booking_date	duration_of_stay	check_in_date	check_out_date	booking_payment_ty...	total_rooms_booked	hotel_id	guest_id	employee_id	total_amou...
1	2020-10-26 17:30:24	3	2020-10-31 14:00:00	2020-11-03 12:00:00	cash	1	3	1	2	7500.00
2	2020-10-26 21:58:06	2	2020-10-31 14:00:00	2020-11-02 12:00:00	card	1	3	2	4	19000.00
3	2020-10-27 05:52:52	1	2020-10-31 14:00:00	2020-11-01 12:00:00	card	1	3	3	1	7500.00
4	2020-10-28 12:00:00	4	2020-11-01 14:00:00	2020-11-05 12:00:00	card	1	3	4	1	22000.00
5	2020-10-29 14:10:20	2	2020-11-01 14:00:00	2020-11-03 12:00:00	card	1	3	5	2	9000.00
6	2020-10-31 15:40:00	1	2020-11-03 14:00:00	2020-11-04 12:00:00	card	1	3	6	2	3500.00
7	2020-10-31 19:10:00	5	2020-11-12 14:00:00	2020-11-17 12:00:00	cash	2	3	7	4	37500.00
8	2020-10-31 19:12:30	3	2020-11-06 14:00:00	2020-11-09 12:00:00	card	1	3	8	2	13500.00
9	2020-10-31 12:48:12	4	2020-11-12 14:00:00	2020-11-16 12:00:00	card	1	3	9	4	10000.00
10	2020-11-01 17:40:00	2	2020-11-05 14:00:00	2020-11-07 12:00:00	cash	2	3	10	4	7000.00

## ROOM TYPE :

```
In [14]: runfile('D:/2year/table.py', wdir='D:/2year')
(1, 'Superior', Decimal('2500.00'), '1 King Bed with Resort view', 'no', 'no')
(2, 'Deluxe Sea View', Decimal('3500.00'), ' 1 King Bed with Ocean view', 'no', 'no')
(3, 'Deluxe Pool Terrace', Decimal('4500.00'), '1 King Bed with Poolside view', 'no', 'no')
(4, 'Deluxe Pool Access', Decimal('5500.00'), '2 King Bed with Pool view', 'yes', 'no')
(5, 'Family room', Decimal('6500.00'), '1 King Bed and 2 Double Bed with Resort view', 'no', 'yes')
(6, 'Garden Suite', Decimal('7500.00'), '1 Bedroom Suite, 1 King Bed and Whirlpool with Ocean view', 'no', 'yes')
(7, 'Presidential Suite', Decimal('9500.00'), '1 Bedroom Suite, 1 King Bed, Balcony and Whirlpool with Ocean view', 'yes', 'yes')
```

room_type_id	name	cost	description	smoke_friendly	pet_friendly
1	Superior	2500.00	1 King Bed with Resort view	no	no
2	Deluxe Sea View	3500.00	1 King Bed with Ocean view	no	no
3	Deluxe Pool Terrace	4500.00	1 King Bed with Poolside view	no	no
4	Deluxe Pool Access	5500.00	2 King Bed with Pool view	yes	no
5	Family room	6500.00	1 King Bed and 2 Double Bed with Resort view	no	yes
6	Garden Suite	7500.00	1 Bedroom Suite, 1 King Bed and Whirlpool with... no		yes
7	Presidential Suite	9500.00	1 Bedroom Suite, 1 King Bed, Balcony and Whir... yes		yes

## ROOM RATE DISCOUNT :

```
In [13]: runfile('D:/2year/table.py', wdir='D:/2year')
(1, '10%', 'January', 'March', 1)
(2, '20%', 'April', 'June', 1)
(3, '40%', 'July', 'September', 1)
(4, 'no', 'October', 'December', 1)
(5, 'no', 'January', 'March', 2)
(6, '15%', 'April', 'June', 2)
(7, '35%', 'July', 'September', 2)
(8, '10%', 'October', 'December', 2)
(9, '15%', 'January', 'March', 3)
(10, '10%', 'April', 'June', 3)
(11, '40%', 'July', 'September', 3)
(12, '10%', 'October', 'December', 3)
(13, '10%', 'January', 'March', 4)
(14, '20%', 'April', 'June', 4)
(15, '25%', 'July', 'September', 4)
(16, '10%', 'October', 'December', 4)
(17, '10%', 'January', 'March', 5)
(18, '15%', 'April', 'June', 5)
(19, '15%', 'July', 'September', 5)
(20, 'no', 'October', 'December', 5)
(21, 'no', 'January', 'March', 6)
(22, '15%', 'April', 'June', 6)
(23, '20%', 'July', 'September', 6)
(24, '5%', 'October', 'December', 6)
```

discount_id	discount_rate	start_month	end_month	room_type_id
1	10%	January	March	1
2	20%	April	June	1
3	40%	July	September	1
4	no	October	December	1
5	no	January	March	2
6	15%	April	June	2
7	35%	July	September	2
8	10%	October	December	2
9	15%	January	March	3
10	10%	April	June	3
11	40%	July	September	3
12	10%	October	December	3
13	10%	January	March	4
14	20%	April	June	4
15	25%	July	September	4
16	10%	October	December	4
17	10%	January	March	5
18	15%	April	June	5
19	15%	July	September	5
20	no	October	December	5
21	no	January	March	6
22	15%	April	June	6
23	20%	July	September	6
24	5%	October	December	6

## EMPLOYEES :

```
In [6]: runfile('D:/2year/table.py', wdir='D:/2year')
(1, 'Steve', 'Rogers', 'Reception', '096-3689640', 'CaptainAmerica@gmail.com', 1, 11, 3)
(2, 'Tony', 'Stark', 'Manager', '092-9275298', 'IamIronman@gmail.com', 1, 12, 3)
(3, 'Natasha', 'Romanoff', 'Cashier', '082-8741329', 'Natasha@gmail.com', 5, 13, 3)
(4, 'Bruce', 'Banner', 'Reservation', '086-1408257', 'Hulk@gmail.com', 1, 14, 3)
(5, 'Peter', 'Parker', 'Porter', '085-9275420', 'SpiderMan@gmail.com', 1, 15, 3)
(6, 'Clint', 'Barton', 'Waiter', '086-6251986', 'Hawkeye@gmail.com', 3, 16, 3)
(7, 'Scott', 'Lang', 'Waiter', '084-4028685', 'AntMan@gmail.com', 3, 17, 3)
(8, 'T', 'Challa', 'House Keeper', '085-2975410', 'BlackPanther@gmail.com', 2, 18, 3)
(9, 'Stephen', 'Strange', 'House Keeper', '098-7252061', 'DoctorStrange@gmail.com', 2, 19, 3)
(10, 'Carol', 'Danvers', 'House Keeper', '080-9264197', 'CaptainMarvel@gmail.com', 3, 30, 3)
(11, 'Loki', 'Laufeyson', 'Chef', '098-1652806', 'Loki@gmail.com', 4, 31, 3)
(12, 'Nicholas', 'Joseph', 'Chef', '089-2789646', 'whooo@gmail.com', 4, 32, 3)
(13, 'Pepper', 'Potts', 'Cashier', '086-9265427', 'wife.of.ironman@gmail.com', 5, 33, 3)
(14, 'Groot', 'Cute', 'Security', '087-4186708', 'Groot@gmail.com', 6, 34, 3)
(15, 'Rocket', 'Cutemuk', 'Manager', '083-0187167', 'rockett@gmail.com', 7, 35, 3)
```

employee_id	first_name	last_name	designation	contact_number	email_address	department_id	address_id	hotel_id
1	Steve	Rogers	Reception	096-3689640	CaptainAmerica@gmail.com	1	11	3
2	Tony	Stark	Manager	092-9275298	IamIronman@gmail.com	1	12	3
3	Natasha	Romanoff	Cashier	082-8741329	Natasha@gmail.com	5	13	3
4	Bruce	Banner	Reservation	086-1408257	Hulk@gmail.com	1	14	3
5	Peter	Parker	Porter	085-9275420	SpiderMan@gmail.com	1	15	3
6	Clint	Barton	Waiter	086-6251986	Hawkeye@gmail.com	3	16	3
7	Scott	Lang	Waiter	084-4028685	AntMan@gmail.com	3	17	3
8	T	Challa	House Keeper	085-2975410	BlackPanther@gmail.com	2	18	3
9	Stephen	Strange	House Keeper	098-7252061	DoctorStrange@gmail.com	2	19	3
10	Carol	Danvers	House Keeper	080-9264197	CaptainMarvel@gmail.com	3	30	3
11	Loki	Laufeyson	Chef	098-1652806	Loki@gmail.com	4	31	3
12	Nicholas	Joseph	Chef	089-2789646	whooo@gmail.com	4	32	3
13	Pepper	Potts	Cashier	086-9265427	wife.of.ironman@gmail.com	5	33	3
14	Groot	Cute	Security	087-4186708	Groot@gmail.com	6	34	3
15	Rocket	Cutemuk	Manager	083-0187167	rockett@gmail.com	7	35	3

## DEPARTMENT :

```
In [4]: runfile('D:/2year/table.py', wdir='D:/2year')
(1, 'Front Office Department', 'reservation,reception')
(2, 'Housekeeping Department', 'responsible for cleanliness')
(3, 'Food and Beverage Service Department', 'responsible for the service of food and drinks')
(4, 'Kitchen and Food Production Department', 'cooking')
(5, 'Accounts and Credits Department', 'maintains all the financial transactions')
(6, 'Security Department', 'responsible for the overall security of the hotel')
(7, 'Sales and Marketing Department', 'bring om business and increase the sales of hotel')
```

department_id	department_name	description
1	Front Office Department	reservation,reception
2	Housekeeping Department	responsible for cleanliness
3	Food and Beverage Service Department	responsible for the service of food and drinks
4	Kitchen and Food Production Department	cooking
5	Accounts and Credits Department	maintains all the financial transactions
6	Security Department	responsible for the overall security of the hotel
7	Sales and Marketing Department	bring om business and increase the sales of hotel

## HOTEL SERVICE :

```
In [9]: runfile('D:/2year/table.py', wdir='D:/2year')
(1, 'Room Service', 'There will be 24-hour Room Service to take care of customers needs', Decimal('50.00'), 3)
(2, 'Laundry and Dry Cleaning', ' Send laundry and dry cleaning with the hotel.', Decimal('100.00'), 3)
(3, 'Spa and Salon', ' Spa and Salon services', Decimal('300.00'), 3)
(4, 'Game Room', ' There are various games available such as card game, Ping Pong', Decimal('50.00'), 3)
(5, ' Fitness Center', '24 hour fitness', Decimal('50.00'), 3)
```

service_id	service_name	service_description	service_cost	hotel_id
1	Room Service	There will be 24-hour Room Service to take car...	50.00	3
2	Laundry and Dry Cleaning	Send laundry and dry cleaning with the hotel.	100.00	3
3	Spa and Salon	Spa and Salon services	300.00	3
4	Game Room	There are various games available such as car...	50.00	3



## ROOMS :

```
In [127]: runfile('D:/untitled20.py', wdir='D:')

(1, 1011, 1, 3)
(2, 102, 1, 3)
(3, 103, 1, 3)
(4, 104, 1, 3)
(5, 105, 1, 3)
(6, 106, 1, 3)
(7, 201, 2, 3)
(8, 202, 2, 3)
(9, 203, 2, 3)
(10, 204, 2, 3)
(11, 205, 2, 3)
(12, 206, 2, 3)
(13, 301, 3, 3)
(14, 302, 3, 3)
(15, 303, 3, 3)
(16, 304, 3, 3)
(17, 305, 3, 3)
(18, 306, 3, 3)
(19, 401, 4, 3)
(20, 402, 4, 3)
(21, 403, 4, 3)
(22, 404, 4, 3)
(23, 405, 4, 3)
(24, 406, 4, 3)
(25, 501, 5, 3)
(26, 502, 5, 3)
(27, 503, 5, 3)
(28, 504, 5, 3)
(29, 505, 5, 3)
(30, 601, 6, 3)
(31, 602, 6, 3)
(32, 603, 6, 3)
(33, 604, 6, 3)
(34, 703, 7, 3)
(35, 702, 7, 3)
```

room_id	room_number	room_type_id	hotel_id
1	1011	1	3
2	102	1	3
3	103	1	3
4	104	1	3
5	105	1	3
6	106	1	3
7	201	2	3
8	202	2	3
9	203	2	3
10	204	2	3
11	205	2	3
12	206	2	3
13	301	3	3
14	302	3	3
15	303	3	3
16	304	3	3
17	305	3	3
18	306	3	3
19	401	4	3
20	402	4	3
21	403	4	3
22	404	4	3
23	405	4	3
24	406	4	3
25	501	5	3
26	502	5	3
27	503	5	3
28	504	5	3
29	505	5	3
30	601	6	3
31	602	6	3
32	603	6	3
33	604	6	3
34	703	7	3
35	702	7	3
NULL	NULL	NULL	NULL

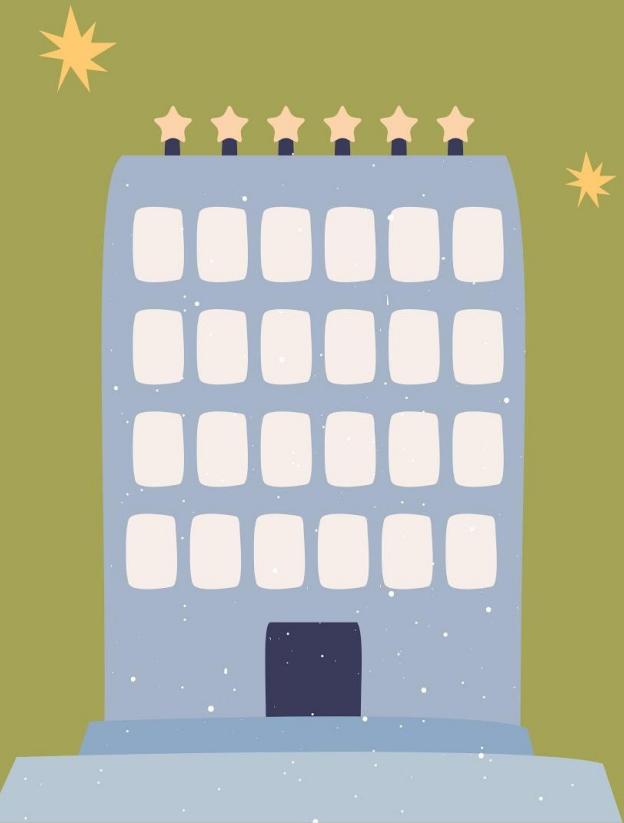
# GUI :

```
Editor - D:\bookings2.py
temp.py bookings1.py bookings2.py
7 from tkinter import *
8 import sqlite3
9
10 win=Tk()
11 def set_text():
12     conn = sqlite3.connect('test.db')
13     print ("Opened database successfully")
14     cursor = conn.cursor()
15     for row in cursor:
16         txt1.delete(0,END)
17         txt1.insert(0,str(row[0]))
18         txt2.delete(0,END)
19         txt2.insert(0,str(row[1]))
20         txt3.delete(0,END)
21         txt3.insert(0,str(row[2]))
22         txt4.delete(0,END)
23         txt4.insert(0,str(row[3]))
24         txt5.delete(0,END)
25         txt5.insert(0,str(row[4]))
26         txt6.delete(0,END)
27         txt6.insert(0,str(row[5]))
28         txt7.delete(0,END)
29         txt7.insert(0,str(row[6]))
30         txt8.delete(0,END)
31         txt8.insert(0,str(row[7]))
32         label1['text']=str(row[0])
33     print ("Operation done successfully")
34     conn.close()
35     return
36
37 def updateval():
38     conn = sqlite3.connect('test.db')
39     s=str(txt1.get())
40     booking_date=str(txt2.get())
41     check_in_date=str(txt3.get())
42     check_out_date=str(txt4.get())
43     booking_payment_type=str(txt5.get())
44     hotel_id=str(txt6.get())
45     guest_id=str(txt7.get())
46     total_amount=str(txt8.get())
47     cursor=conn.cursor()
48     cursor.execute("update bookings set booking_date=?,check_in_date=?,check_out_date=?,booking_payment_type=?,ho
```

A screenshot of a Python Tkinter application window titled "tk". The window contains eight text input fields labeled "booking\_id", "booking\_date", "check\_in\_date", "check\_out\_date", "booking\_payment\_type", "hotel\_id", "guest\_id", and "total\_amount". Below the fields are two buttons: "Click" and "Update".

A screenshot of the same Python Tkinter application window as above, but with data populated into the fields. The "booking\_id" field contains "1", "booking\_date" contains "2020-10-26 17:30:24", "check\_in\_date" contains "2020-10-31 14:00:00", "check\_out\_date" contains "2020-11-03 12:00:00", "booking\_payment\_type" contains "cash", "hotel\_id" contains "3", "guest\_id" contains "1", and "total\_amount" contains "7500". The "Click" and "Update" buttons remain at the bottom.

JOIN TABLE



```
In [18]: runfile('C:/Users/Acer/.spyder-py3/temp.py', wdir='C:/Users/Acer/.spyder-py3')
```

```
(1, 'Front Office Department', 1, 'Steve', 'Rogers', 'Reception', 11, 'Thailand')
(1, 'Front Office Department', 2, 'Tony', 'Stark', 'Manager', 12, 'Thailand')
(1, 'Front Office Department', 4, 'Bruce', 'Banner', 'Reservation', 14, 'Thailand')
(1, 'Front Office Department', 5, 'Peter', 'Parker', 'Porter', 15, 'Thailand')
(2, 'Housekeeping Department', 8, 'T', 'Challa', 'House Keeper', 18, 'Thailand')
(2, 'Housekeeping Department', 9, 'Stephen', 'Strange', 'House Keeper', 19, 'Thailand')
(3, 'Food and Beverage Service Department', 6, 'Clint', 'Barton', 'Waiter', 16, 'Thailand')
(3, 'Food and Beverage Service Department', 7, 'Scott', 'Lang', 'Waiter', 17, 'Thailand')
(3, 'Food and Beverage Service Department', 10, 'Carol', 'Danvers', 'House Keeper', 30, 'Thailand')
(4, 'Kitchen and Food Production Department', 11, 'Loki', 'Laufeyson', 'Chef', 31, 'Thailand')
(4, 'Kitchen and Food Production Department', 12, 'Nicholas', 'Joseph', 'Chef', 32, 'Thailand')
(5, 'Accounts and Credits Department', 3, 'Natasha', 'Romanoff', 'Cashier', 13, 'Thailand')
(5, 'Accounts and Credits Department', 13, 'Pepper', 'Potts', 'Cashier', 33, 'Thailand')
(6, 'Security Department', 14, 'Groot', 'Cute', 'Security', 34, 'Thailand')
(7, 'Sales and Marketing Department', 15, 'Rocket', 'Cutemuk', 'Manager', 35, 'Thailand')
```

```
In [19]: department.department_id, department.department_name, employees.employee_id, employees.first_name, employees.last_name, employees.designation,
addresses.address_id, addresses.country
from department inner join employees on department.department_id = employees.department_id
inner join addresses on
employees.address_id = addresses.address_id
```

```
# In [21]: runfile('C:/Users/Acer/.spyder-py3/temp.py', wdir='C:/Users/Acer/.spyder-py3')
(1, 'George', 'Weasley', '093-8362937', 1, 'Argentina')
(2, 'Harry', 'Potter', '086-8363529', 2, 'Thailand')
(3, 'Susan', 'Bones', '089-8273635', 3, 'USA')
(4, 'Sasipim', 'Kunaruk', '080-897375', 4, 'Thailand')
(5, 'Severus', 'Snape', '082-8452937', 5, 'Canada')
(6, 'Dudly', 'Dursley', '097-3946274', 6, 'Canada')
(7, 'Draco', 'Malfoy', '085-9365284', 7, 'France')
(8, 'Tom', 'Riddle', '092-3249642', 8, 'USA')
(9, 'Newt', 'Scamander', '095-3946294', 9, 'New Zealand')
(10, 'Hermione', 'Granger', '080-0284643', 10, 'New Zealand')

In [22]: guests.guest_id, first_name, last_name, contact_number, addresses.address_id, country from guests inner join addresses on guests.address_id =
addresses.address_id
```

```
In [24]: runfile('C:/Users/Acer/.spyder-py3/temp.py', wdir='C:/Users/Acer/.spyder-py3')
(1, 'Room Service', Decimal('50.00'), 3, 'Bangsarey Beach Resort', '091-6427583', 5, datetime.timedelta(seconds=50400), datetime.timedelta(seconds=43200))
(2, 'Laundry and Dry Cleaning', Decimal('100.00'), 3, 'Bangsarey Beach Resort', '091-6427583', 5, datetime.timedelta(seconds=50400), datetime.timedelta(seconds=43200))
(3, 'Spa and Salon', Decimal('300.00'), 3, 'Bangsarey Beach Resort', '091-6427583', 5, datetime.timedelta(seconds=50400), datetime.timedelta(seconds=43200))
(4, 'Game Room', Decimal('50.00'), 3, 'Bangsarey Beach Resort', '091-6427583', 5, datetime.timedelta(seconds=50400), datetime.timedelta(seconds=43200))
(5, 'Fitness Center', Decimal('50.00'), 3, 'Bangsarey Beach Resort', '091-6427583', 5, datetime.timedelta(seconds=50400), datetime.timedelta(seconds=43200))

In [25]: hotel_services.service_id, service_name, service_cost, hotel.hotel_id, name, contact_number, star_rating, check_in_time, check_out_time from hotel_services inner
join hotel on hotel_services.hotel_id = hotel.hotel_id
```

```
t
In [27]: runfile('C:/Users/Acer/.spyder-py3/temp.py', wdir='C:/Users/Acer/.spyder-py3')
(1, '10%', 'January', 'March', 1, 'Superior', Decimal('2500.00'), 'no', 'no')
(2, '20%', 'April', 'June', 1, 'Superior', Decimal('2500.00'), 'no', 'no')
(3, '40%', 'July', 'September', 1, 'Superior', Decimal('2500.00'), 'no', 'no')
(4, 'no', 'October', 'December', 1, 'Superior', Decimal('2500.00'), 'no', 'no')
(5, 'no', 'January', 'March', 2, 'Deluxe Sea View', Decimal('3500.00'), 'no', 'no')
(6, '15%', 'April', 'June', 2, 'Deluxe Sea View', Decimal('3500.00'), 'no', 'no')
(7, '35%', 'July', 'September', 2, 'Deluxe Sea View', Decimal('3500.00'), 'no', 'no')
(8, '10%', 'October', 'December', 2, 'Deluxe Sea View', Decimal('3500.00'), 'no', 'no')
(9, '15%', 'January', 'March', 3, 'Deluxe Pool Terrace', Decimal('4500.00'), 'no', 'no')
(10, '10%', 'April', 'June', 3, 'Deluxe Pool Terrace', Decimal('4500.00'), 'no', 'no')
(11, '40%', 'July', 'September', 3, 'Deluxe Pool Terrace', Decimal('4500.00'), 'no', 'no')
(12, '10%', 'October', 'December', 3, 'Deluxe Pool Terrace', Decimal('4500.00'), 'no', 'no')
(13, '10%', 'January', 'March', 4, 'Deluxe Pool Access', Decimal('5500.00'), 'yes', 'no')
(14, '20%', 'April', 'June', 4, 'Deluxe Pool Access', Decimal('5500.00'), 'yes', 'no')
(15, '25%', 'July', 'September', 4, 'Deluxe Pool Access', Decimal('5500.00'), 'yes', 'no')
(16, '10%', 'October', 'December', 4, 'Deluxe Pool Access', Decimal('5500.00'), 'yes', 'no')
(17, '10%', 'January', 'March', 5, 'Family room', Decimal('6500.00'), 'no', 'yes')
(18, '15%', 'April', 'June', 5, 'Family room', Decimal('6500.00'), 'no', 'yes')
(19, '15%', 'July', 'September', 5, 'Family room', Decimal('6500.00'), 'no', 'yes')
(20, 'no', 'October', 'December', 5, 'Family room', Decimal('6500.00'), 'no', 'yes')
(21, 'no', 'January', 'March', 6, 'Garden Suite', Decimal('7500.00'), 'no', 'yes')
(22, '15%', 'April', 'June', 6, 'Garden Suite', Decimal('7500.00'), 'no', 'yes')
(23, '20%', 'July', 'September', 6, 'Garden Suite', Decimal('7500.00'), 'no', 'yes')
(24, '5%', 'October', 'December', 6, 'Garden Suite', Decimal('7500.00'), 'no', 'yes')

In [28]: room_rate_discount.discount_id, discount_rate, start_month, end_month, room_type.room_type_id, name, cost, smoke_friendly, pet_friendly from room_rate_discount
inner join room_type on room_rate_discount.room_type_id = room_type.room_type_id
```

```
In [14]: runfile('C:/Users/Acer/.spyder-py3/temp.py', wdir='C:/Users/Acer/.spyder-py3')
(1, 'George', 'Weasley', '093-8362937', 1, datetime.datetime(2020, 10, 31, 14, 0), datetime.datetime(2020, 11, 3, 12, 0), Decimal('7500.00'), 3, 2)
(2, 'Harry', 'Potter', '086-8363529', 2, datetime.datetime(2020, 10, 31, 14, 0), datetime.datetime(2020, 11, 2, 12, 0), Decimal('19000.00'), 3, 4)
(3, 'Susan', 'Bones', '089-8273635', 3, datetime.datetime(2020, 10, 31, 14, 0), datetime.datetime(2020, 11, 1, 12, 0), Decimal('7500.00'), 3, 1)
(4, 'Sasipim', 'Kunarak', '080-897375', 4, datetime.datetime(2020, 11, 1, 14, 0), datetime.datetime(2020, 11, 5, 12, 0), Decimal('22000.00'), 3, 1)
(5, 'Severus', 'Snape', '082-8452937', 5, datetime.datetime(2020, 11, 1, 14, 0), datetime.datetime(2020, 11, 3, 12, 0), Decimal('9000.00'), 3, 2)
(6, 'Dudly ', 'Dursley', '097-3946274', 6, datetime.datetime(2020, 11, 3, 14, 0), datetime.datetime(2020, 11, 4, 12, 0), Decimal('3500.00'), 3, 2)
(7, 'Draco', 'Malfoy', '085-9365284', 7, datetime.datetime(2020, 11, 12, 14, 0), datetime.datetime(2020, 11, 17, 12, 0), Decimal('37500.00'), 3, 4)
(8, 'Tom', 'Riddle', '092-3249642', 8, datetime.datetime(2020, 11, 6, 14, 0), datetime.datetime(2020, 11, 9, 12, 0), Decimal('13500.00'), 3, 2)
(9, 'Newt', 'Scamander', '095-3946294', 9, datetime.datetime(2020, 11, 12, 14, 0), datetime.datetime(2020, 11, 16, 12, 0), Decimal('10000.00'), 3, 4)
(10, 'Hermione', 'Granger', '080-0284643', 10, datetime.datetime(2020, 11, 5, 14, 0), datetime.datetime(2020, 11, 7, 12, 0), Decimal('7000.00'), 3, 4)

In [15]: Select distinct guests.guest_id, guests.first_name, guests.last_name, guests.contact_number,
bookings.booking_id, bookings.check_in_date, bookings.check_out_date, bookings.total_amount, bookings.hotel_id, employees.employee_id from guests inner join bookings on
guests.guest_id = bookings.guest_id inner join employees on bookings.employee_id = employees.employee_id
```

```
In [2]: runfile('D:/2year/table.py', wdir='D:/2year')
(3, 'Bangsarey Beach Resort', 1, 1011, 1, 'Superior', Decimal('2500.00'), '1 King Bed with Resort view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 2, 102, 1, 'Superior', Decimal('2500.00'), '1 King Bed with Resort view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 3, 103, 1, 'Superior', Decimal('2500.00'), '1 King Bed with Resort view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 4, 104, 1, 'Superior', Decimal('2500.00'), '1 King Bed with Resort view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 5, 105, 1, 'Superior', Decimal('2500.00'), '1 King Bed with Resort view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 6, 106, 1, 'Superior', Decimal('2500.00'), '1 King Bed with Resort view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 7, 201, 2, 'Deluxe Sea View', Decimal('3500.00'), '1 King Bed with Ocean view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 8, 202, 2, 'Deluxe Sea View', Decimal('3500.00'), '1 King Bed with Ocean view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 9, 203, 2, 'Deluxe Sea View', Decimal('3500.00'), '1 King Bed with Ocean view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 10, 204, 2, 'Deluxe Sea View', Decimal('3500.00'), '1 King Bed with Ocean view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 11, 205, 2, 'Deluxe Sea View', Decimal('3500.00'), '1 King Bed with Ocean view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 12, 206, 2, 'Deluxe Sea View', Decimal('3500.00'), '1 King Bed with Ocean view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 13, 301, 3, 'Deluxe Pool Terrace', Decimal('4500.00'), '1 King Bed with Poolside view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 14, 302, 3, 'Deluxe Pool Terrace', Decimal('4500.00'), '1 King Bed with Poolside view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 15, 303, 3, 'Deluxe Pool Terrace', Decimal('4500.00'), '1 King Bed with Poolside view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 16, 304, 3, 'Deluxe Pool Terrace', Decimal('4500.00'), '1 King Bed with Poolside view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 17, 305, 3, 'Deluxe Pool Terrace', Decimal('4500.00'), '1 King Bed with Poolside view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 18, 306, 3, 'Deluxe Pool Terrace', Decimal('4500.00'), '1 King Bed with Poolside view', 'no', 'no')
(3, 'Bangsarey Beach Resort', 19, 401, 4, 'Deluxe Pool Access', Decimal('5500.00'), '2 King Bed with Pool view', 'yes', 'no')
(3, 'Bangsarey Beach Resort', 20, 402, 4, 'Deluxe Pool Access', Decimal('5500.00'), '2 King Bed with Pool view', 'yes', 'no')
(3, 'Bangsarey Beach Resort', 21, 403, 4, 'Deluxe Pool Access', Decimal('5500.00'), '2 King Bed with Pool view', 'yes', 'no')
(3, 'Bangsarey Beach Resort', 22, 404, 4, 'Deluxe Pool Access', Decimal('5500.00'), '2 King Bed with Pool view', 'yes', 'no')
(3, 'Bangsarey Beach Resort', 23, 405, 4, 'Deluxe Pool Access', Decimal('5500.00'), '2 King Bed with Pool view', 'yes', 'no')
(3, 'Bangsarey Beach Resort', 24, 406, 4, 'Deluxe Pool Access', Decimal('5500.00'), '2 King Bed with Pool view', 'yes', 'no')
(3, 'Bangsarey Beach Resort', 25, 501, 5, 'Family room', Decimal('6500.00'), '1 King Bed and 2 Double Bed with Resort view', 'no', 'yes')
(3, 'Bangsarey Beach Resort', 26, 502, 5, 'Family room', Decimal('6500.00'), '1 King Bed and 2 Double Bed with Resort view', 'no', 'yes')
(3, 'Bangsarey Beach Resort', 27, 503, 5, 'Family room', Decimal('6500.00'), '1 King Bed and 2 Double Bed with Resort view', 'no', 'yes')
(3, 'Bangsarey Beach Resort', 28, 504, 5, 'Family room', Decimal('6500.00'), '1 King Bed and 2 Double Bed with Resort view', 'no', 'yes')
(3, 'Bangsarey Beach Resort', 29, 505, 5, 'Family room', Decimal('6500.00'), '1 King Bed and 2 Double Bed with Resort view', 'no', 'yes')
(3, 'Bangsarey Beach Resort', 30, 601, 6, 'Garden Suite', Decimal('7500.00'), '1 Bedroom Suite, 1 King Bed and Whirlpool with Ocean view', 'no', 'yes')
(3, 'Bangsarey Beach Resort', 31, 602, 6, 'Garden Suite', Decimal('7500.00'), '1 Bedroom Suite, 1 King Bed and Whirlpool with Ocean view', 'no', 'yes')
(3, 'Bangsarey Beach Resort', 32, 603, 6, 'Garden Suite', Decimal('7500.00'), '1 Bedroom Suite, 1 King Bed and Whirlpool with Ocean view', 'no', 'yes')
(3, 'Bangsarey Beach Resort', 33, 604, 6, 'Garden Suite', Decimal('7500.00'), '1 Bedroom Suite, 1 King Bed and Whirlpool with Ocean view', 'no', 'yes')
(3, 'Bangsarey Beach Resort', 34, 703, 7, 'Presidential Suite', Decimal('9500.00'), '1 Bedroom Suite, 1 King Bed, Balcony and Whirlpool with Ocean view', 'yes', 'yes')
(3, 'Bangsarey Beach Resort', 35, 702, 7, 'Presidential Suite', Decimal('9500.00'), '1 Bedroom Suite, 1 King Bed, Balcony and Whirlpool with Ocean view', 'yes', 'yes')
```

```
In [3]: Select hotel.hotel_id,hotel.name, rooms.room_id, rooms.room_number, room_type.room_type_id, room_type.name, room_type.cost, room_type.description, room_type.smoke_friendly, room_type.pet_friendly from hotel inner join rooms on hotel.hotel_id = rooms.hotel_id inner join room_type on rooms.room_type_id = room_type.room_type_id
```