# PROJECT TITLE

# **Database Model for "Hotel Booking System"**

#### **SCOPE OF THE PROJECT:**

A hotel booking system is a type of service that allows hotels to manage and organize their reservations, guest check-ins, and check-outs. This type of system typically includes features such as the ability to view available rooms, book reservations, and manage guest information. Some hotel booking systems may also include additional features such as the ability to manage room rates, generate reports, and integrate with other hotel management systems. The use of a hotel booking system can help hotels to streamline their operations and improve their efficiency. It can also help to provide a better experience for guests by allowing them to easily book and manage their reservations online.

By deploying Oracle 9i Enterprise versions, the focus is put into practice. The Standard Query Language is MYSQL. The full set of business rules is created using database objects, recorded procedures, and data trigger points in SQL.

This can display the proprietor of hotel, what type of rooms are most popular, and it can prompt adjustments to the building's infrastructure, such the addition of more of a particular room type.

#### **OBJECTIVE OF THE DATABASE:**

- Efficiency hotel operations will be improved.
- User friendly to access the data.
- Provide large storage for important information.
- Secure the information from physical systems and unauthorized persons.
- Remove redundant information.
- Authenticate database enlargement.
- Do the recent changes to the available database easy and quick.
- Multiple users can be activated at a time.
- When user requests the data response should be prompted.

#### **USER REQUIREMENTS:**

The basic user requirements for each hotel management system should contain, both functional and nonfunctional requirements. They include:

#### **Functional: -**

- An account will be associated with each online booking, and only one person can use one account.
- Users must be able to find for the best outcome.
- Accept the timings for checking room availability for that time period.

- To specified contact booking confirmation must be sent.
- Display and calculate other utilities and accommodation charges.
- Undo bookings.
- Guest's records should able to change and displayed.
- Changing rooms.

#### Non-functional: -

- Encryption should be used for booking to avoid bots.
- Within time limit search results should be prompted.
- In case of incorrect details, users should get help to complete the mandatory fields
- All payment methods should be acceptable.
- Efficient, accessible and easy to use.
- Maintain the record of responses, documentation and activities.

# **DESCRIPTION OF THE OBJECTIVES:**

- The database table contains data related to the transactions that go happen in the hotel all day.
- Some miscellaneous charges will be added to the transcations. Those should be separated into 3, it can also be added if needed.
- It will also display which transactions contain miscellaneous charges in booking.

- By connecting it to transaction table via the payment\_id column, we can also learn how the customer is paying for the reserved room.
- It holds entire people who uses with the hotel room and its database.
- This table lists every employee employed there along with their schedules.
- It also contains data about the various rooms.
- It includes all information regarding reservations made by a customer and given to a front desk agent.
- This table contains all the information regarding the hotel's numerous rooms.
- It also contains details about the various accommodation kinds that are offered.
- The availability of the room, as well as whether it is being sterilized or not, may also be seen.
- After calculating the total cost of the reservation, the payment options will be used to complete the transaction.
- This table provide the dashboard of all the information to front desk to know if customer needs to book the room.
- This table will show all the customers who made bookings and their information.
- This is very helpful for front desk employee to look after the information based on reservation id.
- It can display how many total numbers of bookings are booked in every room type. This will display the owners of the hotel about which types of rooms are in the most used and are in demand, and can progress to differences within the hotel's framework, such as including most of a particular room type of the hotel.

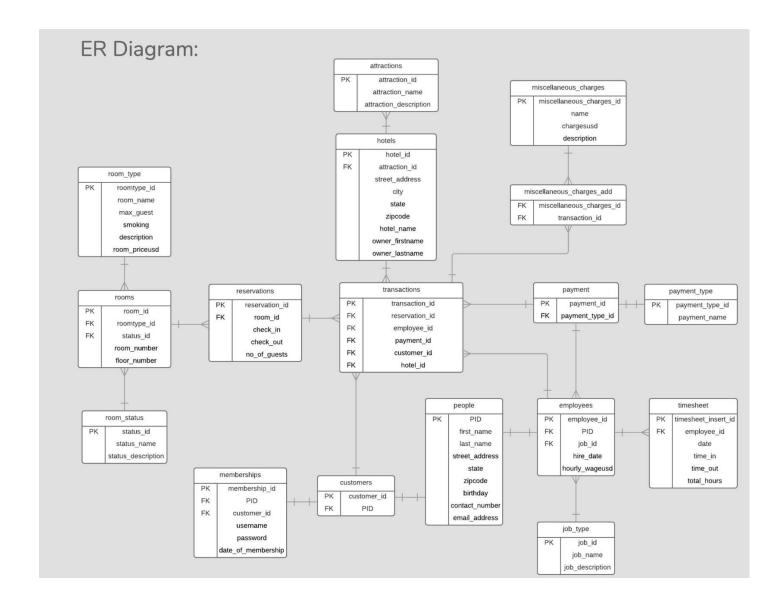
- This could be either the shareholder or a person who must have way in access to all the data base within it.
- The Hotel Administrator has more access to the record list, as they need to append all the different types of data which can fit into the database.
- The reception member must have the access for the bookings and customer record database and schedule the reservations.
- The employees are the ones who have the least control over the database. They need to identify which rooms needs sanitization.

# **BUSINESS RULES:**

- Only one customer can do any number of reservations.
- Employees under the age of eighteen are not permitted for the room booking.
- A customer can only make 1 reservation at a time.
- Only employee will contain the count of reservations.
- Customer can pay any number of payments.
- One customer can only pay for one payment.
- Employee under the age of 18 are not allowed to work.
- With one transaction, customer can do only one reservation.
- Payment can be done for more than one room.
- Customers under the age of 18 are not allowed to book the room.
- Only one room is associated with one reservation.
- Zero or more reservations can belong to each room.
- Only one room can be associated in one time frame.

- Customers can choose the type of room they require.
- Customers should enter or check-in into the room for the total number of guests they booked.
- Customers must check in for the number of visitors they reserved
- Guests under the age of 18 are not permitted to work.

# **ER-DIAGRAM (CROW'S FOOT MODEL):**



# **DATA DICTIONARY:-**

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	REQUIRED	PK OR F
	EMPLOYEE_ID	employee id	number(1-9)	9	Υ	PK
	P_ID	Person ID	number	999	Υ	FK
EMPLOYEES	JOB_ID	job id	number	999	Υ	FK
	HIRE_DATE	hired date	date	MMDDYYYY	Υ	
	HOURLY_WAGE_USD	wage in usd for hourly	number	99	Υ	
	P ID	person ID	char(3)	999	Υ	PK
	FIRST_NAME	first name	char	Xxxx	Y	
	LAST_NAME	last name	char	Xxxx	Υ	
	STREET ADDRESS	address of street	varchar(40)	xxx234x	Υ	
PEOPLE	STATE	state	char(20)	Xxxx	Υ	
	ZIPCODE	zipcode	number	9999	Υ	
	BIRTHDAY	b-day	date	MMDDYYYY		
	CONTACT_NUMBER	phone number	number(10)	999999999		
	EMAIL_ADDRESS	email address	varchar(40)	xxx234x	Υ	
	MISCELLANEOUS CHARGES ID	charges id	Varchar(30)	Xxx12x	Υ	FK
MISCELLANEOUS_CHARGES_ADD	TRANSACTION ID	transaction id	Varchar(30)	Xxx12x Xxx12x	Y	FK
	TRANSACTION_ID	transaction id	varchar(30)	AAA12A	'	FK
PAYMENT	PAYMENT_ID	payment id	number	9999	Υ	PK
FAIMENT	PAYMENT_TYPE_ID	type in payment id	varchar	xxx	Υ	FK
	PAYMENT TYPE ID	type in payment id	varchar	99x9x9xx	Υ	PK
PAYMENT_TYPE	PAYMENT NAME	payment name	char(20)	XXXX	-	

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	REQUIRED	PK OR I
	TIMESHEET_INSERT_ID	timesheet id	number(11)	999		PK
TIMESHEET	EMPLOYEE_ID	employee id	number(5)	999		FK
	DATE	date	date	MMDDYYYY	Y	
	TIME_IN	in time out time	time	HH:MM:SS	Y	
	TIME_OUT		time	HH:MM:SS		
	TOTAL_HOURS	total hours	number(3)	99	Y	
	JOB ID	job id	number(3)	999	Υ	PK
JOB_TYPE	JOB_NAME	name of the job	char(20)	xxx	Υ	
	JOB_DESCRIPTION	job description	varchar(30)	xxx9x	Y	
						nu.
	MEMBERSHIP_ID	membership id	number(3)	999		PK
	P_ID	Pid	number(3)	999		FK
MEMBERSHIPS	CUSTOMER_ID	customer id	number(3)	999		FK
	USER_NAME	user name	char(20)	XXX	Υ	
	PASSWORD	password	varchar(20)	xxx9x!	Y	
	DAT_OF_MEMBERSHIP	date of membership	date	MMDDYYYY	Υ	
	TRANSACTION_ID	transaction id	Varchar(30)	Xxx12x	Υ	PK
	RESERVATION_ID	reservation id	number	999	Υ	FK
	EMPLOYEE_ID	employee id	number(5)	999	Υ	FK
TRANSACTIONS	PAYMENT_ID	payment id	number	999	Υ	FK
	CUSTOMER_ID	customer id	number(3)	999	Υ	FK
	HOTEL_ID	hotel id	number	999	Υ	FK

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	REQUIRED	PK OR FK
	RESERVATION ID	reservation id	number	999	Υ	PK
	ROOM ID	room id	number	999	Υ	FK
RESERVATIONS	CHECK IN	in time	time	HH:MM:SS	Υ	
	CHECK OUT	out time	time	HH:MM:SS	Υ	
	NO_OF_GUESTS	total members staying	number	999	Υ	
	ROOM ID	room id	number	999	v	PK
	ROOMTYPE ID	type in room id	varchar	xxx9x	Y	FK
ROOMS	STATUS_ID	status id	char	XXX	Y	FK
ROOMO	ROOM_NUMBER	room number	number	999		110
	FLOOR NUMBER	floor number	number	999		
	. 2001_10111211	TOO HUITOO	Tidili Dol	000		
	ROOMTYPE ID	type of room id	varchar	xxx9x	Y	PK
	ROOM NAME	room name	varchar	xxx9x	Υ	
ROOM_TYPE	MAX_GUEST	max number of guests	number	999	Υ	
KOOM_TTPE	SMOKING DESCRIPTION	smoking /non smoking zone	char	xxx	Υ	
	ROOM_PRICE_USD	price per room in USD	number	999	Υ	
	STATUS_ID	status id	char	xxx	Υ	PK
ROOM_STATUS	STATUS_NAME	status name	char	xxx	Υ	
	STATUS_DESCRIPTION	confirmation status	char	xxx	Υ	

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	REQUIRED	PK OR FK	
	HOTEL_ID	hotel id	number	999	Y	PK	
	ATTRACTION ID	attraction id	number	999		FK	
	STREET_ADDRESS	address of street	varchar(40)	xxx234x	Y		
	CITY	city	char(20)	Xxxx	Y		
HOTELS	STATE	state	char(20)	Xxxx	Y		
	ZIPCODE	zipcode	number	999	Y		
	HOTEL_NAME	name of the hotel	char	XXX	Y		
	OWNER FIRSTNAME	first name of the owner	char	Xxxx	Y		
	OWNER_LASTNAME	last name of the owner	char	Xxxx	Υ		
	ATTRACTION_ID	attraction id	number	999	Υ	PK	
ATTRACTIONS	ATTRACTION_NAME	attraction name	char	XXX	Υ		
ATTRACTIONS	ATTRACTION_DESCRIPTION	description of the attraction	char	xxx	Υ		
	CHAR	Fixed character length data (	1 - 255 characters)				
	VARCHAR	Variable character length data	a (1 - 2,000 characters	s)			
	NUMBER	Numeric data. NUMBER (9,2)	is used to specify nur	nbers with up to nine digits	, including two dig	its to the right of the	decim
	DATE	format of date(year, month, d	ay) has to be given.				
	TIME	The time format	has to be(hours:minu	itesseconds)			

Through the project submission link, the excel file for the Data Dictionary is uploaded from a remote location.

# **Data Entry and Update: -**

#### Table name: - payment\_type

This will hold the different types of payment methods that the hotel accepts.

```
use mysql;
create table payment_type(
payment_type_id char(2) not null,
payment_name text not null,
primary key (payment_type_id)
);
insert into payment_type(payment_type_id, payment_name)
values(t1, 'cash');
insert into payment_type(payment_type_id, payment_name)
values(t2, 'card');
insert into payment_type(payment_type_id, payment_name)
values(t3, 'bitcoin');
select * from payment_type
```

#### Output:-

4	payment_type_id character (2)	payment_name text
1	t1	Cash
2	t2	Card
3	t3	Bitcoin

#### **Table name: - hotels**

This will contain the information about 1 of the hotel attractions in that nearby area.

```
📵 🏡 🥩 🔍 🖺 🖃
  1 • use mysql;
  hotel_id
                          char,
      street_address
                        varchar(50),
                          varchar(30),
      city
      state
                          varchar(20),
      zipcode
                          int(7),
      hotel_name
 8
                          varchar(20),
     owner_firstname
                         varchar(20),
 10    owner_lastname
                         varchar(20),
 11
      attraction_id
                          char(3),
 12
      primary key (hotel_id),
 13
     foreign key (attraction_id) references attraction1 (attraction_id)
 14
 15
 16 • insert into hotels1(hotel_id,street_address,city,state,zipcode,hotel_name,owner_firstname,owner_lastname,attraction_id)
 17
       values(h1,'123 Gateway Lane','LiverPool','texas','76201','The Dunn Hotel Inn','Taylor','Dunn',a1);
 18 • insert into hotels1(hotel_id,street_address,city,state,zipcode,hotel_name,owner_firstname,owner_lastname,attraction_id)
 19
      values(h2, '134 Hello Street', 'Hamstred', 'New York', '12345', 'The Hilly Hall', 'John', 'Cena', a1);
 20
 21
 22
```

#### **Output:-**

4	hotel_id character (7)	street_address text	<b>city</b> text	<b>state</b> text	<b>zipcode</b> integer	hotel_name text	owner_firstname text	owner_lastname text	attraction_id character (3)
1	h1	123 Galway Lane	Live	Texas	12894	The Dunn H	Taylor	Dunn	a1
2	h2	134 Hello Street	На	New	12345	The Hilly Hall	John	Cena	a1

#### **Table name: - Attractions**

This table contains different attractions in Texas, Liverpool including the hotel that has the database focuses on.

```
use mysql;
 2 • — create table attractions(
       attraction id
 3
                              char(3),
       attraction_name
                              varchar(20),
 4
       attraction description
 5
                                   varchar(20),
       primary key (attraction_id)
 7
        insert into attractions(attraction_id, attraction_name, attraction_description)
 8
        values(a1, 'Hotel', 'places to stay near by the attraction');
 9
        insert into attractions(attraction_id, attraction_name, attraction_description)
10 •
        values(a2,' FDR Museum', 'places to check out near by the attraction');
12 •
        insert into attractions(attraction_id, attraction_name, attraction_description)
        values(a3,'Riverwalk','places to explore near by the attraction');
13
        insert into attractions(attraction_id, attraction_name, attraction_description)
14 •
        values(a4, 'Restaurants', 'places to eat near by the attraction');
15
        select * from attractions;
16 •
```

4	attraction_id character (3)	attraction_name text	attraction_description text
1	a1	Hotel	Places to stay in the ar
2	a2	FDR Museum	Places to check out his
3	a3	Riverwalk	Places to explore the a
4	a4	Restaurants	Places to eat in the area

### **Table name:- room\_status**

This will display about whether the room is vacant, cleaned or being booked.

```
🛅 🔒 🦻 🖟 👰 🕛 🜇 🕢 🔞 Limit to 1000 rows
                                                     8 🔅 🥩 Q ¶ 🖃
      use mysql;
  2 • ⊖ create table room_status(
       status_id
                              char(2),
                             varchar(30),
  4
       status_name
       status_description
  5
                              varchar(50),
  6
       primary key (hotel_id)
 7
 9 • insert into room_status(status_id,status_name,status_description)
       values(s1, 'Booked', 'This room is booked');
 11 • insert into room_status(status_id,status_name,status_description)
       values(s2,'Vacant','This room is completely available');
 13 • insert into room_status(status_id,status_name,status_description)
       values(s3, 'Being Cleaned', 'This room is in the Cleaning State');
 14
 15
 16
```

4	status_id character (2)	status_name text	status_description text
1	s1	Booked	This room is booked.
2	s2	Vacant	This room is compl
3	s3	Being Cleaned	This room is in the

#### Table name:- transactions

This will contain all the data about the transactions that will happen via this hotel day in and day out.

```
1 • use mysql;
2 • ⊖ create table transactions(
3
       transaction_id
                              char(20),
4
       reservation_id
                              char(20),
5
       employee id
                              char(10),
 6
       payment_id
                               char(10),
7
       customer id
                               char(10),
       hotel_id
                               char(10),
8
9
       primary key (transaction_id),
       foreign key (hotel_id) references hotels (hotel_id),
10
11
       foreign key (reservation_id) references reservations (reservation_id),
12
       foreign key (employee_id) references employees (employee_id),
13
       foreign key (payment_id) references payment (payment_id)
14
      - );
15 • insert into transactions(transaction_id, reservation_id, employee_id, payment_id, customer_id, hotel_id)
16
       values(11111, 'rv1', 'e1', 'b1', 'c1', 'h1');
17 •
       insert into transactions(transaction_id, reservation_id, employee_id, payment_id, customer_id, hotel_id)
       values(12222,'rv2','e2','b2','c2','h1');
18
19 •
      insert into transactions(transaction_id, reservation_id, employee_id, payment_id, customer_id, hotel_id)
       values(13333,'rv3','e3','b3','c3','h1');
20
21 • insert into transactions(transaction_id, reservation_id, employee_id, payment_id, customer_id, hotel_id)
22
       values(14444, 'rv4', 'e4', 'b4', 'c1', 'h1');
23 • insert into transactions(transaction_id, reservation_id, employee_id, payment_id, customer_id, hotel_id)
       values(155555, 'rv5', 'e5', 'b5', 'c5', 'h1');
25 • insert into transactions(transaction_id, reservation_id, employee_id, payment_id, customer_id, hotel_id)
26
       values(166666,'rv6','e6','b6','c1','h1');
27 • insert into transactions(transaction_id, reservation_id, employee_id, payment_id, customer_id, hotel_id)
28
       values(177777, 'rv7', 'e7', 'b7', 'c3', 'h1');
29 • insert into transactions(transaction_id, reservation_id, employee_id, payment_id, customer_id, hotel_id)
```

4	transaction_id character (8)	reservation_id character (8)	employee_id character (3)	payment_id character (3)	customer_id character (3)	hotel_id character (3)
1	11111	rv1	e1	b1	c1	h1
2	12222	rv2	e1	b2	c2	h1
3	13333	rv4	e3	b3	c3	h1
4	14444	rv3	e1	b4	c1	h1
5	155555	rv5	e3	b5	c5	h1
6	166666	rv6	e3	b6	c1	h1
7	177777	rv7	e1	b7	c3	h1
8	188888	rv8	e3	b8	c3	h1
9	199999	rv9	e1	b9	c4	h1
10	112222	rv10	e1	b10	c4	h1

#### Table name:- room\_type

This will hold all types of the rooms and its information.

```
1 • use mysql;
 2 • ⊖ create table room_type(
      roomtype__id
      room_name
                       varchar(30),
5
     max_guest
                       int,
     smoking
                      boolean,
7
     description
                      varchar(30),
     room_priceUSD decimal(15,2),
8
9
     primary key (roomtype_id)
    1;
10
11
12 • insert into room_type(roomtype_id,room_name,max_guest,smoking,description,room_priceUSD)
13
      values(type1, 'Double Queen',5,false,'Two doubled sized', 150.00);
14 • insert into room_status(status_id,status_name,status_description)
      values(type2, 'Single King', 2, false, 'One king sized', 120.00);
16 • insert into room_status(status_id,status_name,status_description)
17
      values(type3, 'Suite Style',8,true, 'Two bedroom',300.00);
```

### Output:-

4	roomtype_id character (8)	room_name text	max_guest integer	<b>smoking</b> boolean	description text	room_priceusd numeric (15,2)
1	type1	Double Quee	5	false	Two double	150.00
2	type2	Single King	2	false	One king si	120.00
3	type3	Suite Style	8	true	Two bedroo	300.00

#### Table name:- payment

This will show how the customer paid for their booking, and is linked with the transaction data through the payment\_id.

```
1
       use mysql;
 2 • ⊖ create table payment(
       payment_id
                           char(3),
 4
       payment_type_id
                           char(2),
       primary key (attraction_id),
 6
       foreign key (payment_type_id) references payment_type (payment_type_id)
 7
        insert into payment(payment_id, payment_type_id)
 8 .
 9
        values(b1,t1);
        insert into payment(payment_id, payment_type_id)
10 •
11
        values(b2,t2);
12 •
        insert into payment(payment_id, payment_type_id)
        values(b3,t1);
13
14 •
        insert into payment(payment_id, payment_type_id)
        values(b4,t3);
15
        insert into payment(payment_id, payment_type_id)
16 •
        values(b5,t2);
17
        insert into payment(payment_id, payment_type_id)
18 •
19
        values(b6,t3);
        insert into payment(payment_id, payment_type_id)
20 •
21
        values(b7,t1);
22 •
        insert into payment(payment_id, payment_type_id)
```

4	payment_id character (3)	payment_type_id character (2)
1	b1	t1
2	b2	t2
3	b3	t1
4	b4	t3
5	b5	t2
6	b6	t3
7	b7	t1
8	b8	t3
9	b9	t1
10	b10	t2

Table name:- rooms

This will hold all data about various rooms in the hotel.

```
🛅 🔒 🦻 👰 🜔 🚷 🕢 🔯 Limit to 1000 rows
                                                       📵 🐆 🥩 🔍 🗻 🖃
  1 • use mysql;
 2 • 😑 create table rooms(
       room_id
                           char(6),
  4
       roomtype_id
                          char(10),
  5
       room_number
                           char(5),
  6
       floor_number
                           integer,
  7
       status_id
                           char(2),
       primary key (room_id),
 9
       foreign key (roomtype_id) references room_type (roomtype_id),
 10
       foreign key (status_id) references room_status (status_id)
 11
      · );
 12
13 • insert into rooms(room_id,roomtype_id,room_number,floor_number,status_id)
       values(rm1, type1, 100, 1, s1);
15 • insert into rooms(room_id,roomtype_id,room_number,floor_number,status_id)
       values(rm2, type2, 200, 2, s2);
17 • insert into rooms(room_id,roomtype_id,room_number,floor_number,status_id)
       values(rm3, type3, 300, 3, s1);
19 • insert into rooms(room_id,roomtype_id,room_number,floor_number,status_id)
       values(rm4, type3, 120, 1, s1);
21 • insert into rooms(room_id,roomtype_id,room_number,floor_number,status_id)
       values(rm5, type2, 220, 2, s2);
23 • insert into rooms(room_id,roomtype_id,room_number,floor_number,status_id)
       values(rm6, type3, 305, 3, s2);
25
```

4	room_id character (6)	roomtype_id character (10)	room_number character (5)	floor_number integer	status_id character (2)
1	rm1	type1	100	1	s1
2	rm2	type2	200	2	s2
3	rm3	type3	300	3	s1
4	rm4	type3	120	1	s1
5	rm5	type2	220	2	s2
6	rm6	type3	305	3	s2
7	rm7	type1	205	2	s1

Table name: - miscellaneous\_charges\_add

This will show which transaction has miscellaneous charges for their orders.

```
1 • use mysql;
 2 ● ⊖ create table miscellaneous_charges_add(
       miscellaneous_charges_id
      transaction id
                                        char(8),
      foreign key (miscellaneous_charges_id) references miscellaneous_charges (miscellaneous_charges_id),
      foreign key (transaction_id) references transactions (transaction_id)
       insert into miscellaneous_charges_add(miscellaneous_charges_id, miscellaneous_charges_id)
 8 •
 9
       values(m1,11111);
       insert into miscellaneous_charges_add(miscellaneous_charges_id, miscellaneous_charges_id)
10 .
12 • insert into miscellaneous_charges_add(miscellaneous_charges_id, miscellaneous_charges_id)
      values(m3,199999);
13
14 • select * from miscellaneous_charges_add
```

#### Output:-

4	miscellaneous_charges_id character (3)	transaction_id character (8)
1	m1	11111
2	m2	155555
3	m3	199999

#### Table name:- reservations

This will contain all data about the bookings a customer submits.

```
1 • use mysql;
2 • — create table reservations(
      reservation_id
3
                         char(8),
4
     room_id
                          char(6),
5
     check_in
                         date,
6
     check_out
                         date,
7
     no_of_guests
                          char(8),
8
     smoking
                          boolean,
     primary key (reservation_id),
9
10
     foreign key (room_id) references rooms (room_id)
11
    1;
12
13
14 • insert into rooms(reservation_id,room_id,check_in,check_out,no_of_guests,smoking)
      values(rv1,rm1,2017-09-04,2017-09-05,4,true);
16 • insert into rooms(reservation_id,room_id,check_in,check_out,no_of_guests,smoking)
17
      values(rv2,rm3,2016-03-24,2016-03-25,1,false);
18 • insert into rooms(reservation_id,room_id,check_in,check_out,no_of_guests,smoking)
      values(rv3,rm4,2017-05-27,2017-05-28,5,false);
20 • insert into rooms(reservation_id,room_id,check_in,check_out,no_of_guests,smoking)
21
      values(rv4, rm4, 22017-09-13, 2017-09-14, 5, true);
22 • insert into rooms(reservation_id,room_id,check_in,check_out,no_of_guests,smoking)
      values(rv5,rm5,2017-07-27,2017-07-28,4,false);
24 • insert into rooms(reservation_id,room_id,check_in,check_out,no_of_guests,smoking)
      values(rv6,rm2,2013-08-22,2013-08-23,5,true);
25
26
27
```

4	reservation_id character (8)	room_id character (6)	check_in date	check_out date	no_of_guests character (8)	smoking boolean
1	rv1	rm1	2017-09	2017-09-05	4	true
2	rv2	rm3	2016-03	2016-03-25	1	false
3	rv3	rm4	2017-05	2017-05-28	5	false
4	rv4	rm4	2017-09	2017-09-14	5	true
5	rv5	rm5	2017-07	2017-07-28	4	false
6	rv6	rm2	2013-08	2013-08-23	1	false
7	rv7	rm7	2012-06	2012-07-01	2	false
8	rv8	rm1	2015-09	2015-09-07	4	true
9	rv9	rm3	2011-03	2011-03-18	3	false
10	rv10	rm6	2014-06	2014-06-22	5	true

#### Table name :- Employees

```
0 % ØQ ¶ 🖘
                                 Limit to 1000 rows
 28 • 

CREATE TABLE employees (
 29
        employee_id
                       char(3),
 30
       PID
                       char(5),
 31
       job_id
                      char(2),
 32
       hire_date
                      date,
 33
       hourly_wageusd decimal(15,2),
 34
       primary key (employee_id),
 35
       foreign key (PID) references people (pid), foreign key (job_id) references job_type (job_id)
 36
 37
 38 • insert into employees(employee_id,PID,job_id,hire_date,hourly_wageusd)
        values(e1,p6,90,2017-03-14,10.00);
 39
 40 • insert into employees(employee_id,PID,job_id,hire_date,hourly_wageusd)
        values(e2,p7,91,2012-08-19,10.00);
 42 • insert into employees(employee_id,PID,job_id,hire_date,hourly_wageusd)
        values(e3, p8, 91, 2011-07-10, 10.00);
 44 • insert into employees(employee_id,PID,job_id,hire_date,hourly_wageusd)
        values(e4,p9,92,2009-05-19,20.00);
 46 • insert into employees(employee_id,PID,job_id,hire_date,hourly_wageusd)
 47
       values(e5,p10,93,2017-08-11,15.00);
 48
```

4	employee_id character (3)	pid character (5)	job_id character (2)	hire_date date	hourly_wageuse numeric (15,2)
1	e1	p6	90	2017-03	10.00
2	e2	p7	91	2012-08	10.00
3	e3	p8	91	2011-07	10.00
4	e4	p9	92	2009-05	20.00
5	e5	p10	93	2017-08	15.00

Table name: - Customers

```
2 • ○ CREATE TABLE customers (
     PID char(3),
3
4
   customer_id char(3),
5
   primary key (customer_id),
6
7
    foreign key (pid) references people (pid)
8
9 • insert into customers(PID, customer_id)
0
     values(p1,c1);
1 • insert into customers(PID, customer_id)
     values(p2,c2);
3 • insert into customers(PID, customer_id)
4
    values(p3,c3);
5 • insert into customers(PID, customer_id)
     values(p4,c4);
7 • insert into customers(PID, customer_id)
8
    values(p5,c5);
9
```

4	pid character (3)	customer_id character (3)
1	p1	c1
2	p2	c2
3	р3	c3
4	p4	c4
5	p5	c5

# Table name :- miscellaneous\_charges

This will hold the miscellaneous charges that are added to a booking transaction. Those charges will be broken down into 3 options, but more can also be added.

```
use mysql;
 2 • 😑 create table miscellaneous_charges(
       miscellaneous_charges_id
       name
                                       varchar(10),
       chargesUSD
                                      decimal(15,2),
      description
                                       varchar(20),
       primary key (miscellaneous_charges_id)
 8
9 •
        insert into miscellaneous_charges(miscellaneous_charges_id, name, chargesUSD, description)
10
        values(m1,'Stolen Item',50.00,'Something from the room is missing');
        insert into miscellaneous charges(miscellaneous charges id, name, chargesUSD, description)
        values(m2, 'Broken Furniture', 200.00, 'Something from the room is broken and needs to be replaced');
12
13 •
        insert into miscellaneous_charges(miscellaneous_charges_id, name, chargesUSD, description)
        values(m3, 'Food', 25.00, 'All room service for food is under a $25 buffet, all you can eat style');
14
       select * from miscellaneous charges
```

#### Output:-

4	miscellaneous_charges_id character (3)	name text	chargesusd numeric (15,2)	description text
1	m1	Stolen Item	50.00	Something from the room is missing.
2	m2	Broken Furniture	200.00	Something from the room is broken and needs to be replaced.
3	m3	Food	25.00	All room service for food is under a \$25 buffet, all you can eat style.

Table name: - People

This will hold data of all the people that interacted with the hotel and hotel database.

```
1 • use mysql;
2 ● ⊖ create table people(
       PID
                            char(5),
       first_name
                             text,
       last_name
       street_address
                              text,
                              text,
8
      zipcode
                            integer,
     birthday
                             date.
     contact_number
                              text,
      email_address
     primary key (PID)
12
13
14 • insert into people(PID,first_name, last_name, street_address, state, zipcode, birthday, contact_number, email_address)
15
        values(p1, 'Jason', 'Haley', ' 13 School Street', 'New York', 11946, 1978-11-04, 4587390869, 'jason.haley@gmail.com');
       insert into people(PID, first_name, last_name, street_address, state, zipcode, birthday, contact_number, email_address)
17
        values(p2, 'Scott', 'Fritsch', '10 Emerson Court', 'New York',11946,1989-06-06,1234567890, 'scott.fritsch@gmail.com');
      insert into people(PID,first_name, last_name, street_address, state, zipcode, birthday, contact_number, email_address)
18 •
19
        values(p3,'Jami','Domencio','15 Maple Court','New York',18977,1997-04-07,6312546789,'jami.domencio@gmail.com');
20 •
      insert into people(PID,first_name, last_name, street_address, state, zipcode, birthday, contact_number, email_address)
21
       values(p4, 'Alan', 'Laboseur', '255 Honey Drive', 'New York', 12607, 1985-09-07, 1118675301, 'alan.lab@coolgmail.com');
22 • insert into people(PID,first_name, last_name, street_address, state, zipcode, birthday, contact_number, email_address)
       values(p5, 'Jack', 'Heuber', '123 Talk Road', 'New York', 12445, 1998-10-03,1345879978, 'talkingguy@talking.com');
24 • insert into people(PID,first_name, last_name, street_address, state, zipcode, birthday, contact_number, email_address)
     values(p6, 'Dave', 'Connelly', '15 Bae Court', 'Rhod Island',12366,1997-11-10,1879087890, 'jefferyjeffery@gmail.com');
```

## Output: -

4	pid character (5)	first_name text	last_name text	street_address text	state text	<b>zipcode</b> integer	birthday date	contact_number text	email_address text
1	p1	Jason	Haley	13 School Street	New	11946	1978-11	4587390869	jason.haley@g
2	p2	Scott	Fritsch	10 Emerson Co	New	11946	1989-06	1234567890	scott.fritsch@g
3	р3	Jami	Domenico	15 Maple Court	New	18977	1997-04	6312546789	jami.domenico
4	p4	Alan	Laboseur	255 Honey Drive	New	12601	1985-09	1118675301	alan.lab@coolg
5	p5	Jack	Heuber	123 Talk Road	New	12445	1998-10	1345879978	talkingguy@tal
6	p6	Dave	Connelly	15 Bae Court	Rhod	12366	1997-11	1879087890	jefferyjeffery@
7	p7	Taylor	Connelly	17 Harbor Road	River	14577	1997-11	6316805787	taylor.kathryn
8	p8	John	Sasso	40 Bestfriend L	New	12889	1997-12	2267897765	john.sasso@be
9	p9	Shannon	Cover	33 Oak Ave	New	89059	1990-06	0987654321	shannon.cover
10	p10	Sreya	Sobti	1334 Linda Lane	Penn	37890	1995-10	7778987654	sreyasobti@ind

# Data Retrieval and Simple Reports: -

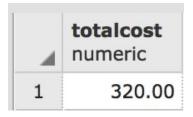
#### Queries: -

1. Writing a Query to find the total price for the customer that must be paid for their visit, based on reservation\_id.

```
use mysql;
select (( select chargesusd from miscellaneous charges
where miscellaneous charges_ id in (select miscellaneous_ charges id
from miscellaneous charges add where transaction_id in (select transaction_id
from transactions where reservation_id = 'rv5')))

(select room_priceusd from room_type where roomtype_id in (select roomtype_id
from rooms
where room_id in (select room_id from reservations
where reservation_id in (select reservation_id
from transactions
where reservation_id = 'rv5'))))) as totalCost;
```

## Output: -



2. Writing a query to display how much an employee will get paid for number of hours he works for.

```
use mysql;
select ( (select hourly_wageusd from employees
where pid in (select pid from people
where first_name = 'Taylor' AND last_name = 'Connelly'))
(select total_hours from timesheet
where employee_id in (select employee_id from employees
where pid in (select pid from people
where first_name = 'Taylor' AND last_name = 'Connelly')))) as TotalPay;
```

4	totalpay numeric
1	70.00

### 3. Writing a Query to show the details of the room.

```
1 • use mysql;
2 • select room_id, room_number, floor_number, status_description, room_name, room_priceusd, max_guest
3    from rooms inner join room status on rooms.status_id = room_status.status_id
4    inner join room_type on rooms. roomtype_id = room_type. roomtype_id;
5
```

# Output: -

4	room_id character (6)	room_number character (5)	floor_number integer	status_description text	room_name text	room_priceusd numeric (15,2)	max_guest integer
1	rm1	100	1	This room is booked.	Double Quee	150.00	5
2	rm2	200	2	This room is compl	Single King	120.00	2
3	rm3	300	3	This room is booked.	Suite Style	300.00	8
4	rm4	120	1	This room is booked.	Suite Style	300.00	8
5	rm5	220	2	This room is compl	Single King	120.00	2

# 4. Writing a Query to show the Customer Information where it must show customers reservations, and their information.

```
use mysql;
select first_name, last_name, contact_number

from people where pid in (select pid
from customers

where customer_id in (select customer_id
from transactions

where reservation_id in (select reservation_id
from reservations FULL outer join people ON people.pid = reservations. reservation_id)));
```

#### Output: -

4	first_name text	last_name text	contact_number text
1	Jason	Haley	4587390869
2	Scott	Fritsch	1234567890
3	Jami	Domenico	6312546789
4	Alan	Laboseur	1118675301
5	Jack	Heuber	1345879978

5. Write a Query to display the room status for the employees.

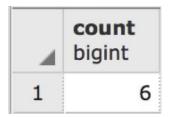
```
use mysql;
select room_id, room_number, floor_number, status_description
from rooms inner join room_status on rooms.status_id = room_status.status_id;
```

4	room_id character (6)	room_number character (5)	floor_number integer	status_description text
1	rm1	100	1	This room is booked.
2	rm2	200	2	This room is compl
3	rm3	300	3	This room is booked.
4	rm4	120	1	This room is booked.
5	rm5	220	2	This room is compl
6	rm6	305	3	This room is compl

## 6. Write a Query to display Total number of reservations after 2015

```
use mysql;
select count(reservation_id)
from reservations
where check_in > = '2015-01-01';
```

# Output: -



7. Write a Query to display the total no. of employees who worked for eight hours.

```
1 • use mysql;
2 • select count (employee id)
3  from timesheet
4  where total_hours >= '8';
5
```

4	<b>count</b> bigint
1	5

# 8. Write a Query to display for how many reservations are being booked in each room type

```
1 • use mysql;
2 • SELECT rooms. roomtype_id, COUNT (reservations. room_id)
3    AS NumberofRooms FROM reservations
4    LEFT JOIN rooms ON reservations.room_id = rooms. room_id
5    GROUP BY roomtype_id;
6
```

## **Output:**

4	roomtype_id character (10)	<b>numberofrooms</b> bigint
1	type2	2
2	type1	3
3	type3	5