

Humberto Pedraza

Jesus Garcia

Adrian Lopez

## **Hotel Database Project**

### **1. Introduction**

We are the hotel group and we consist of three students: Humberto Pedraza, Jesus Garcia, and Adrian Lopez. Our project is to create a database for a hotel. The reason we chose a hotel is because we thought it would be a great topic to practice our database design and learning about database in general. Making a database out of a hotel did not seem too difficult but good enough to get our feet wet.

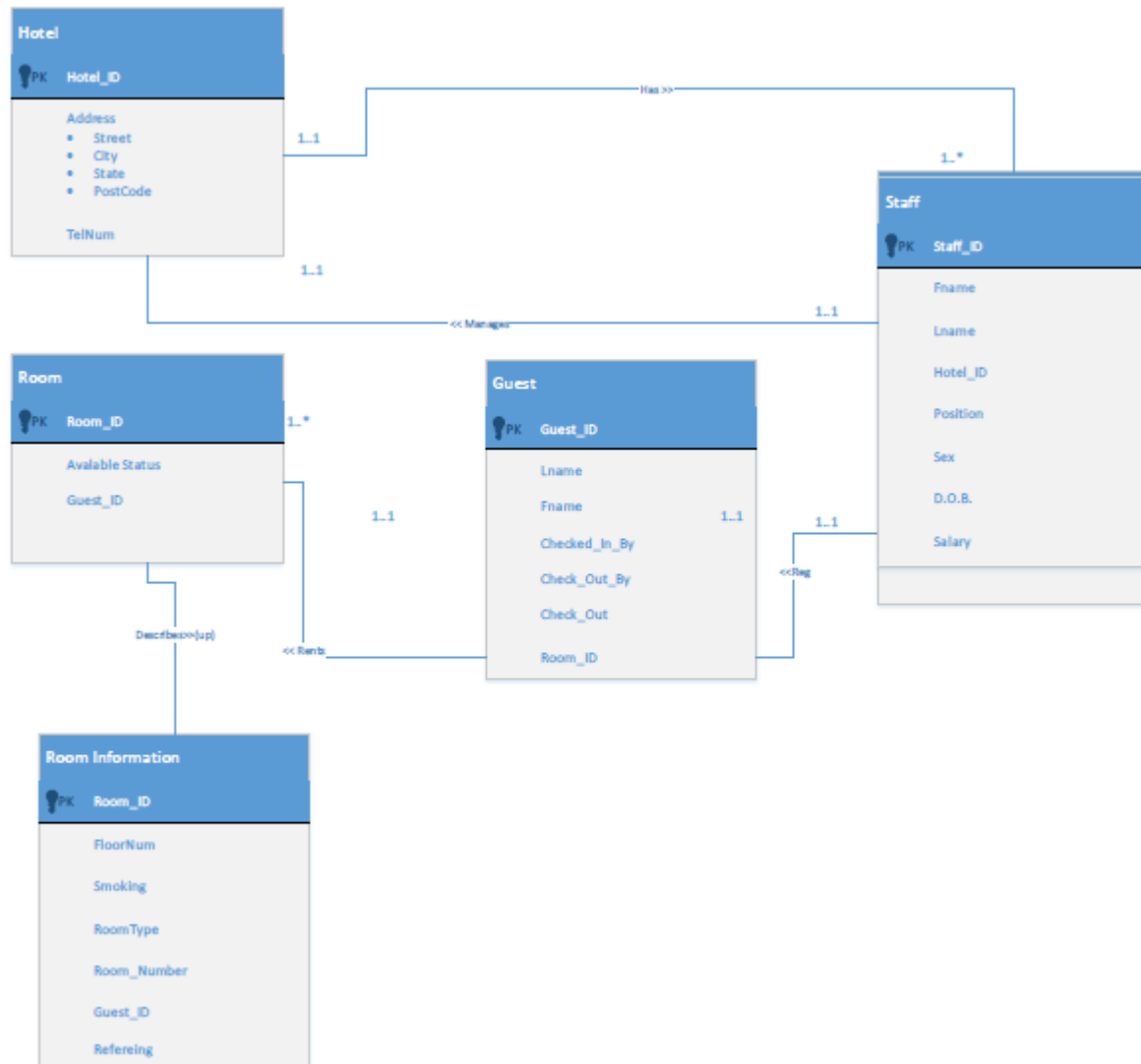
### **2. Basic Database Design**

The purpose of the hotel database system is to support the hotel's staff and so that managers can manage the hotel in an organized fashion. The database allows staff to quickly answer questions from potential guests about any rooms available, floors available, questions about room specifications such as beds, smoking/nonsmoking, etc (our database as of now doesn't support this). The database's primary mission is to help assist the staff give the hotel's guests the best experience possible.

the square is inside system boundaries and

the perfect room the guest can rent out. The inside boundaries is everything that is data/information and outside boundaries is everyone and everything the database helps out.

### **3. ER Diagram**



**Figure 2: Hotel Database ER diagram**

Entities: Hotel, Staff, Guest, Room, Room\_Information

Attribute of Hotel and its relationship with Staff entity:

Hotel\_ID (Primary Key) – the key that will uniquely identify each hotel Address

Address – Will specify the hotel's location

TelNum – Will hold the unique hotel's telephone number.

Hotel has a one to many relationship with entity staff. So each hotel will have several staff members working for it.

Attributes of Staff and its relationship with Guest entity:

Staff\_ID (Primary Key) – Will uniquely identify a staff member to its hotel.

Fname – This attribute is the staff member's first name

Lname – This attribute is the staff member's last name

Hotel\_Num (Foreign Key) – It is a key that identifies where the staff member works at

Position – This attribute holds the staff member's role to the hotel

Sex – Identifies the staff members gender

D.O.B. – This attribute holds the staff members birthdate

Salary – This attribute holds the staff members yearly income

Staff has a one to one relationship with guest because staff will check-in and check-out a guest

Attributes of Guest and its relationship with Room entity:

Guest\_ID(Primary Key) – It uniquely identifies a guest at a hotel

First\_Name – This attribute will hold guest first name

Last\_Name - This attribute will hold guest last name

Room\_Id(Foreign key) – This attribute identifies what room guest will be staying at

Checked\_In\_By(Foreign Key) – This attribute holds identifies which staff member checked them in

Check\_In – This is the time guest checks in

Check\_Out\_By – This is the time guest should check out

Check\_Out - The actual time guest checks out

The Guest Table has a one to many relationship with room because every guest can rent out one or several rooms at a time.

Attributes of Room and Room relationship with Room\_Information entity:

Room\_ID(Primary Key) – Uniquely identifies the a room

Available\_Status – Attribute to indicate if the room is available or not

Guest\_ID - Attribute that identifies who is renting out the room if NULL then no one is at that room

Room has a one to one relationship with Room\_Information entity because Room\_Information contains specific information of a physical room.

Attributes of Room\_Information:

Room\_ID – Uniquely identifies what room it is referring to

Floor\_Number – The floor number in which the room is located at

Smoking – Attribute that indicates whether a guest can smoke in their or not

Room\_Type – Attribute that describes the type of room it is such as suite, two bed room, one twin bed, etc.

Room\_Number – The room's room number to find where the room is at

Hotel\_ID(Foreign Key) - An attribute that identifies where that room belongs too.

#### 4. Relational Database Schema

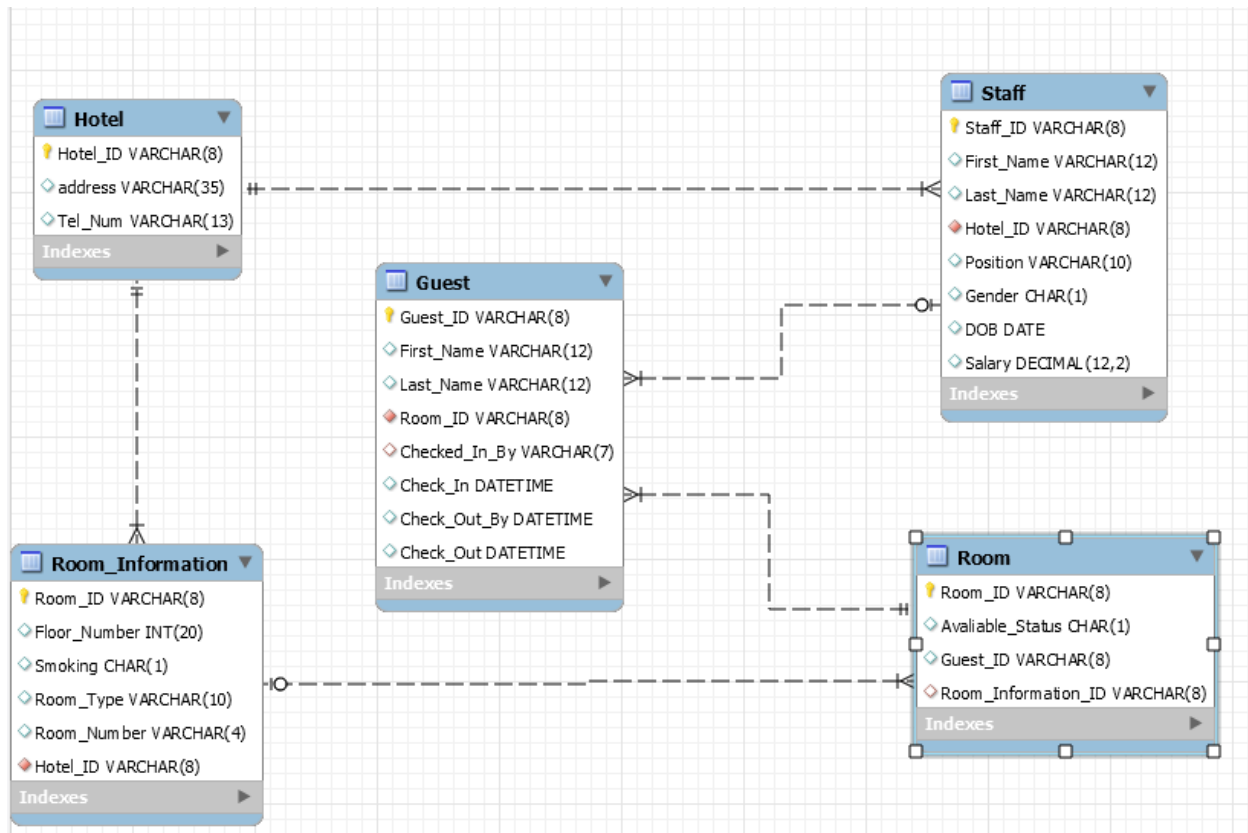


Figure 3: Schema diagram created out of our schema script

```
DROP DATABASE IF EXISTS `Hotel_Database`;
```

```
CREATE DATABASE IF NOT EXISTS `Hotel_Database` DEFAULT CHARACTER SET latin1  
COLLATE latin1_swedish_ci;
```

```
USE `Hotel_Database`;
```

```
DROP TABLE IF EXISTS `Hotel`;
```

```
CREATE TABLE IF NOT EXISTS `Hotel` ( Hotel_ID varchar(8) NOT NULL,  
                                     address varchar(35) ,  
                                     Tel_Num varchar(13));
```

```
DROP TABLE IF EXISTS `Staff`;
```

```
CREATE TABLE IF NOT EXISTS `Staff` ( Staff_ID varchar(8) NOT NULL,
```

```
First_Name varchar(12) ,  
Last_Name varchar(12) ,  
Hotel_ID varchar(8) NOT NULL,  
Position varchar(10) ,  
Gender char(1) ,  
DOB date ,  
Salary decimal(12 , 2));
```

```
DROP TABLE IF EXISTS `Room`;
```

```
CREATE TABLE IF NOT EXISTS `Room` ( Room_ID varchar(8) NOT NULL,  
Avaliable_Status char(1),  
Guest_ID varchar(8),  
Room_Information_ID varchar(8));
```

```
DROP TABLE IF EXISTS `Guest`;
```

```
CREATE TABLE IF NOT EXISTS `Guest` ( Guest_ID varchar(8) NOT NULL,  
First_Name varchar(12) ,  
Last_Name varchar(12) ,  
Room_ID varchar(8) NOT NULL,  
Checked_In_By varchar(7),  
Check_In DATETIME,  
Check_Out_By DATETIME,  
Check_Out DATETIME );
```

```
DROP TABLE IF EXISTS `Room_Information`;
```

```
CREATE TABLE IF NOT EXISTS `Room_Information` ( Room_ID varchar(8) NOT NULL,  
Floor_Number int(20) ,  
Smoking char(1),  
Room_Type varchar(10) ,  
Room_Number varchar(4),  
Hotel_ID varchar(8) NOT NULL);
```

```

alter table `Hotel` add PRIMARY KEY( Hotel_ID ) ;
alter table `Staff` add PRIMARY KEY( Staff_ID ) ;
alter table `Staff` add FOREIGN KEY(Hotel_ID) REFERENCES `Hotel`(Hotel_ID);
alter table `Guest` add PRIMARY KEY(Guest_ID);
alter table `Guest` add FOREIGN KEY(Checked_In_By) REFERENCES `Staff`(Staff_ID);
alter table `Room` add PRIMARY KEY(Room_ID);

```

```

ALTER TABLE `Guest` ADD FOREIGN KEY (Room_ID) REFERENCES `Room` (Room_ID);
alter table `Room_Information` add PRIMARY KEY(Room_ID);
alter table `Room_Information` add FOREIGN KEY(Hotel_ID) references Hotel(Hotel_ID);
alter table `Room` add FOREIGN KEY(Room_Information_ID) references
Room_Information(Room_ID);

```

The schema is very much like our ER diagram. It consist of 5 tables which are:

- 1) Hotel table which has three attributes Hotel\_ID which holds a hotel's unique identification (Primary Key), address which hold that hotel's physical location, and Tel\_Num which holds the hotel's telephone number.
- 2) Staff table which has eight attributes Staff\_ID which holds a unique identification for each staff member (Primary Key), Hotel\_ID which hold a unique key in which identifies where the staff member works at/ First\_Name attribute and Last\_Name attribute which holds staff members first name and last name. Position which holds a unique staff members position towards the hotel they work at. Gender which holds the staff member's sex. DOB attribute which holds staff member's date of birth. Salary attribute which holds information about how much the staff member makes annually.
- 3) Guest table which has 8 attributes. First\_Name attribute and Last\_Name attribute which holds the member's first and last name. Guest\_ID which is primary key and holds the guest's unique identification. Room\_ID (Foreign Key) attribute which holds a unique ID in which the guest is staying at. Checked\_In\_By (foreign key) which holds information about what staff member checked in the guest. Check\_In attribute which holds date and time in which guest checked in. Check\_Out\_By attribute which holds information on when guest is expected to check out by. Check\_Out attribute which is information about when guest actually left and checked out of the hotel.
- 4) Room table which has 3 attributes. Room\_ID which uniquely identifies a unique room. Available\_Status which holds information about whether the room is available to rent or



not. Guest\_ID(Foreign Key) attribute which holds information about what guest is staying in at the room.

- 5) Room\_Information table which has 6 attributes. Room\_ID (primary key) which uniquely identifies the room in which it is string information about. Floor\_Number attribute which stores information about what floor is the room located at. Smoking attribute which is information indicating whether smoking is allowed in the room or not. Room\_Type attribute which holds information about what kind of room it is, such as two bed room, suite, etc. Room\_Number attribute is data about what room is that room located at in the hotel. Hotel\_ID (foreign key) attribute which stores information about what hotel does that room belong too.

## 5. SQL Queries

Description: Query will return all rooms that are available to rent out.

Query: select \* from room where available\_status = 'Y';

Query Result:

			Room_ID	Available_Status	Guest_ID
<input type="checkbox"/>			R010000	Y	NULL
<input type="checkbox"/>			R010002	Y	NULL
<input type="checkbox"/>			R010003	Y	NULL
<input type="checkbox"/>			R010004	Y	NULL
<input type="checkbox"/>			R010005	Y	NULL
<input type="checkbox"/>			R010006	Y	NULL

**Table 1: Table shows all rooms available**

Description: Query will find all rooms available in all hotels who's available and are non-smoking.

Query: select \* from room\_information join room on room\_information.Room\_id = room.Room\_ID where Room.available\_status = 'Y' AND Room\_Information.smoking = 'N';

Query Result:

Room_ID	Floor_Number	Smoking	Room_Type	Room_Number	Room_ID	Available_Status	Guest_ID
R010000	5	N	One Bed Ro	5203	R010000	Y	NULL
R010002	5	N	Deluxe	5208	R010002	Y	NULL
R010003	5	N	Suite	5255	R010003	Y	NULL
R010006	5	N	Suite	5223	R010006	Y	NULL

**Table 2: Results from query returning all rooms available that are non-smoking.**

Description: Return all guests in floors higher than 5. Hotel will be giving free lunch to all guest in floors higher than 5.

Query: select \* from `guest` g join `room` on g.Room\_ID = room.Room\_ID join room\_information on room.Room\_ID = room\_information.Room\_ID where Floor\_Number > 5;

Query Result:

Guest_ID	First_Name	Last_Name	Room_ID	Checked_In_By	Room_ID	Available_Status	Guest_ID	Room_ID	Floor_Number	Smoking	Room_Type	Room_Number
G000001	Jesus	Gonzalez	R000001	S000001	R000001	N	G000001	R000001	20	N	Suite	2020
G000332	Bob	Garcia	R000332	S000002	R000332	N	G000332	R000332	25	Y	Tripe Room	2501

**Table 3: Query results showing all rooms on floor 5 and higher.**

Description: Return all guests checked in by a specific staff member. Manager wants to see how hard staff "S000001" has been working.

Query:

select \* from `room` r join `guest` on r.Room\_ID = guest.Room\_ID join `staff` on guest.Checked\_In\_By = staff.staff\_ID where staff.staff\_ID="S000001";

Query Result:

Room_ID	Available_Status	Guest_ID	Guest_ID	First_Name	Last_Name	Room_ID	Checked_In_By	Staff_ID	First_Name	Last_Name	Hotel_ID	Position	Gender	DOB	Salary
R000001	N	G000001	G000001	Jesus	Gonzalez	R000001	S000001	S000001	Adrian	Lopez	H000001	Manager	M	1987-07-15	100000.00
R000003	N	G000003	G000003	Charlie	Garcia	R000003	S000001	S000001	Adrian	Lopez	H000001	Manager	M	1987-07-15	100000.00
R000100	N	G000152	G000152	Bert	Ramirez	R000100	S000001	S000001	Adrian	Lopez	H000001	Manager	M	1987-07-15	100000.00

**Table 4: Query result showing all guest signed in by a certain staff member.**

Description: Query will find how many guest we have at all hotels.

Query:

SELECT H.address AS 'Hotel\_Location', COUNT(G.Guest\_ID) AS '# of Guest'  
FROM Hotel H , Guest G , Staff S  
WHERE G.Checked\_In\_By = S.Staff\_ID AND  
S.Hotel\_ID = H.Hotel\_ID  
GROUP BY(H.address);

Query Result:

Hotel_Location	# of Guest
123 Sesame St. ,MA	2
365 Sesame St. ,MA	2
502 Abraham Rd. ,FL	1
5554 Range Rd. ,MA	2
912 Hackberry Rd.,Palm Tow,TX	3

**Table 5: Query result showing how many guest we have at each hotel.**

Description: Query will find hotel that has most guest renting out suites.

Query:

```
SELECT H.address AS 'Location' , Count(I.Room_Type = 'Suite') AS '# of Guest with Suite'
FROM Hotel H, Room_Information I
WHERE I.Hotel_ID = H.Hotel_ID AND I.Room_Type = 'Suite'
GROUP BY H.address
HAVING Count(I.Room_Type = 'Suite') > 0
ORDER BY Count(I.Room_Type = 'Suite') DESC;
```

Query Result:

Location	# of Guest with Suite
912 Hackberry Rd.,Palm Tow,TX	2
365 Sesame St. ,MA	2
5554 Range Rd. ,MA	1

**Table 6: Shows query results showing all hotels that have rented out suites in descending order. The first tuple is the hotel with the most suites rented out tied with the second tuple.**

Description: Query will find all staff members working at a each hotel. So query should display staff members name (first and last) along with what hotel they work at.

Query:

```
SELECT S.First_Name , S.Last_Name, H.address AS 'Works At:'
FROM Staff S JOIN Hotel H
ON S.Hotel_ID = H.Hotel_ID;
```

Query Result:

First_Name	Last_Name	Works At:
Adrian	Lopez	912 Hackberry Rd.,Palm Tow,TX
Humberto	Lopez	123 Sesame St. ,MA
Jessica	Maine	5554 Range Rd. ,MA
Ray	Soto	502 Abraham Rd. ,FL
Lilly	Lopez	365 Sesame St. ,MA

**Table 7: Query result showing all staff members first and last name and what hotel they work at.**

Description: Query will find all rooms that are not of room type 'suite' or 'deluxe'

Query:

```
SELECT room_id
FROM Room
WHERE Room.Room_ID NOT IN
(Select DISTINCT room_id
FROM room_information
WHERE room_type = "Deluxe" or room_type = "Suite");
```

Query Result:

room_id
R000003
R000100
R000332
R0005236
R010000
R010004

**Table 8: Query result shows all rooms that are not of room type suite or deluxe.**

## **6. Conclusion**

The biggest challenges we had faced was a technical related challenge. We found it challenging to write SQL queries for a real world situation. At first we came up with queries that did not benefit our mission statement. We then got together at the CS lab and thought about the main issue hotels has to face about their business. The most meaningful query was a query to find available rooms for our potential guest. Every guest has different needs, some guest will be oil rig workers who want the cheapest room available, people visiting family, celebs who requiring the best suite; there are many types of potential guest and the most meaningful queries should be queries that revolve around quickly assisting-the guest in finding a room they specifically are looking floor. That way the database will allow the hotel to give the guest an even better customer service and ultimately have returning customers. Once we got our mind set we were able to write queries.

The most interesting thing we learned from this was that the first design is NOT what we are going to end up at the end. We thought about our project, came up with a design, and in the end we forgot to include attributes we did think of until we actually created the schema in SQL.

