UNIVERSITY
OF ONTARIO
INSTITUTE OF TECHNOLOGY

SOFE 3700U
Data Management Systems
Final Project
Title: Hotel DBMS

Allan Santosh - 100557518
Thanushan Rameswaran - 100462146
Dhanushga Lionel - 100616831

Abstract

In this report, we discuss our work on creating a Hotel database. While some may think that this project lack creativity, the way we implemented it shows a lot of thought and consideration. Our idea for a centralized hotel database fills a niche that both hotels and customers will appreciate.

Our goal for this project is to create a Hotel Reservation website, where customers can book and view their bookings in a plain and simplified looking website. We will accomplish this by implementing some of concepts and theory that we learned in class such as using queries to get proper information required by the user and updating the database based on selected queries.

We will initially discuss the thought behind our decision to build a hotel database. We will then discuss our goals for the project and how it relates to what we have learned in the course.

Coming up with an idea for a database system involved a lot of brainstorming. We ultimately decided to thinking of an existing service that we could improve upon and after a couple options we decided on a Hotel database. We believe that our system will offer advantages to both customers and Hotel companies.

Introduction

According to Statistica, the US hotel industry generated revenue of over $200 billion USD during the 2017 fiscal year alone. The hotel industry has also been growing every year since 2009, and the trend is expecting to continue. There many hotel booking websites. Sites like TripAdvisor, Kayak, Expedia, Seatguru, Google Flights, Priceline, Bookings.com and hotel.com are being used by millions of people. People book for vacation, work trips, honeymoons and so on and so forth. Since the hotel booking industry is so oversaturated, competition is fierce between the big companies, leaving smaller booking sites to eat the dust.

Relation to Other Work

It is true that there are many other hotel DBMS. All those companies listed in the Introduction will most certainly have their own database. This new centralized hotel-booking database differs from those company databases. Instead of each company having their own database, all major companies will use the same database.

There are multiple advantages to this. Firstly, for the customer. When deciding on a hotel, instead of visiting and researching potential hotels in the area, they will have all of the information compiled in one place. They can compare different features, amenities, locations, prices, room types, etc. that the hotels have to offer. This way, not only will the customer easily be able to find exactly what suits their needs but they will also find it for the best prices.

The advantage to hotels is that as the network grows for our Hotel database system they will be presenting their information to a wider audience. A database system like ours has the potential to grow in popularity because it compiles the data of many hotels into one place. As our Hotel database system grows in popularity, it may become the go to spot for people looking to book a hotel. This gives hotels a much wider client base and potential customers. This will allow hotels to save on advertising and client acquisition. The money that would otherwise have been put into advertising can be better utilized in repairs and renovations to the hotel itself. They can also add more features, special packages and reduced pricing.

Our hotel database system fosters a healthy sense of competition amongst the hotels. Because all hotel companies are listed in the same place, there is incentive to offer better pricing or better features to win customers. The end result is that money and resources are being spent on the customer rather than on advertising.

Main Body of Work

Goals of the Project

We aim to build a database that all booking companies will share. The goal of this centralized hotel-booking database is to allow for quick data access in a highly visual manner. We want to decrease the amount of storage space to book a hotel, and make this tremendous volume of information seem simple. This will be accomplished by segmenting the system into several different views which would detail a specific aspect of the information provided. One view will show the breakdown of the booking hierarchy, so that customers can book what room(s) they want and retrieve that information. Another view is to graphically locate the hotels using geoinformatics.

Relationship to the Course

Many of the database design concepts that we learned in class lectures and labs were implemented in our project. We initially had to come up with a conceptual idea for the project. We did this by brainstorming as a group and thinking of services that already exist that could be improved upon with the use of a database system. We came up with a couple ideas and finally decided on a hotel database system.

Once we had our idea of a service, we had to start with the design. For this we made a list of entities we would need along with the attributes for each of the entities. We modeled this in a relational schema diagram. Similar to the examples we did in class, we labeled each entity and the attributes for each entity. Each entity needed to have an identifying attribute, that is an attribute that would be unique to each row or tuple in that entity. This identifying attribute was underlined and used as the primary key. Foreign keys were then identified. That is if an attribute from one entity referenced another attributes primary key, it was made a foreign key. In the relational schema, foreign keys had an arrow pointing from the referencing attribute to the primary key attribute that was being referenced.

Having completed the relational schema diagram we were then able to make the ER diagram. We used the model we learned in the lectures. We modeled each entity as a rectangle and its attributes were connected ovals. Key entities were underlined. Relationships were modeled with a diamond shape and were connected to relating entities. We made a legend similar to the one in the lectures labeling each shape and what it represented. This way anybody viewing the diagram would be able to follow it.

These diagrams were very useful in the creation of the ER diagram. In the ER diagram we included the same entities and the attributes. We also added the appropriate types for the attributes.

Design and Implementation

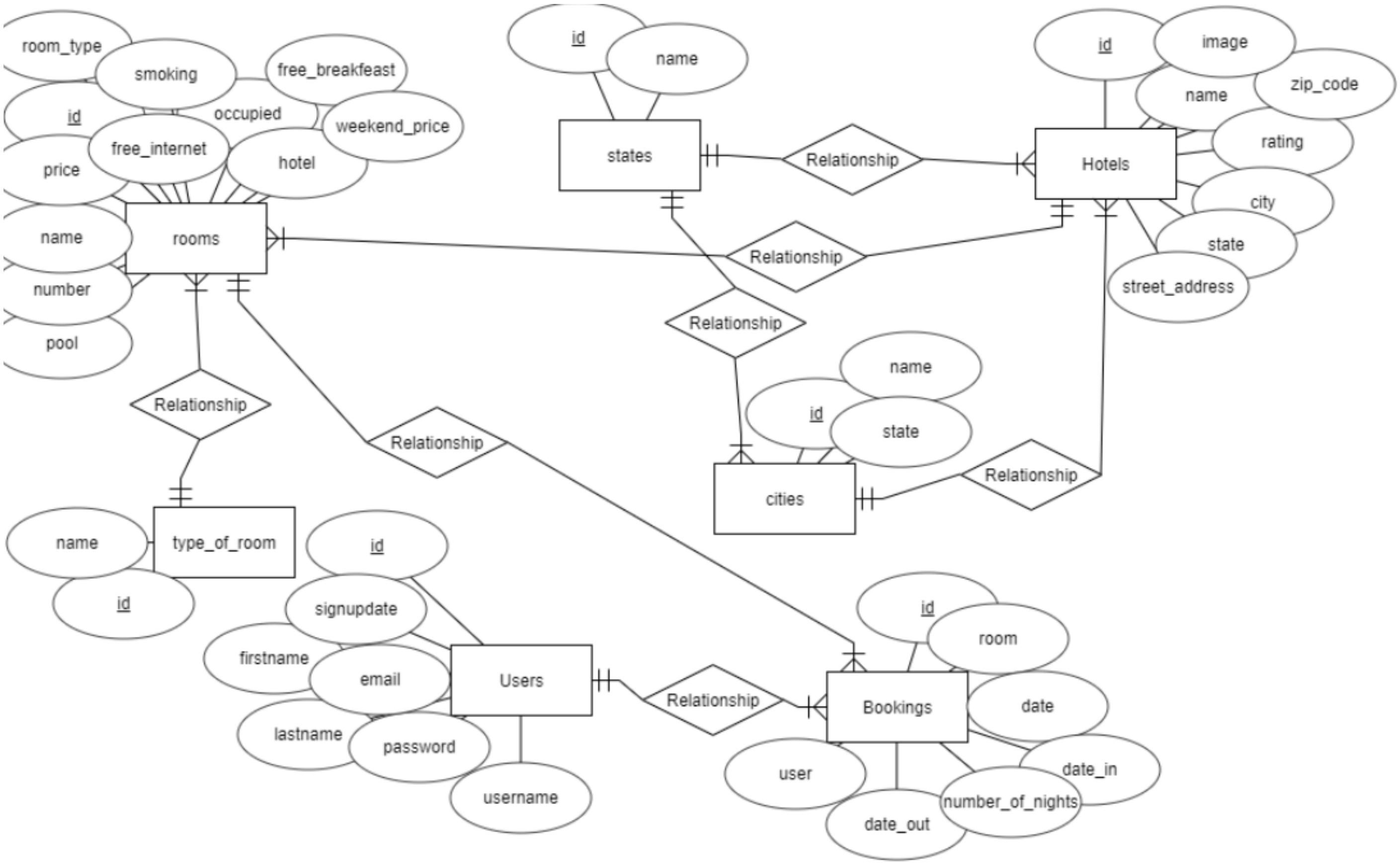
Tools Used: XML, PHP, HTML, PHPMYADMIN, MYSQL DATABASE

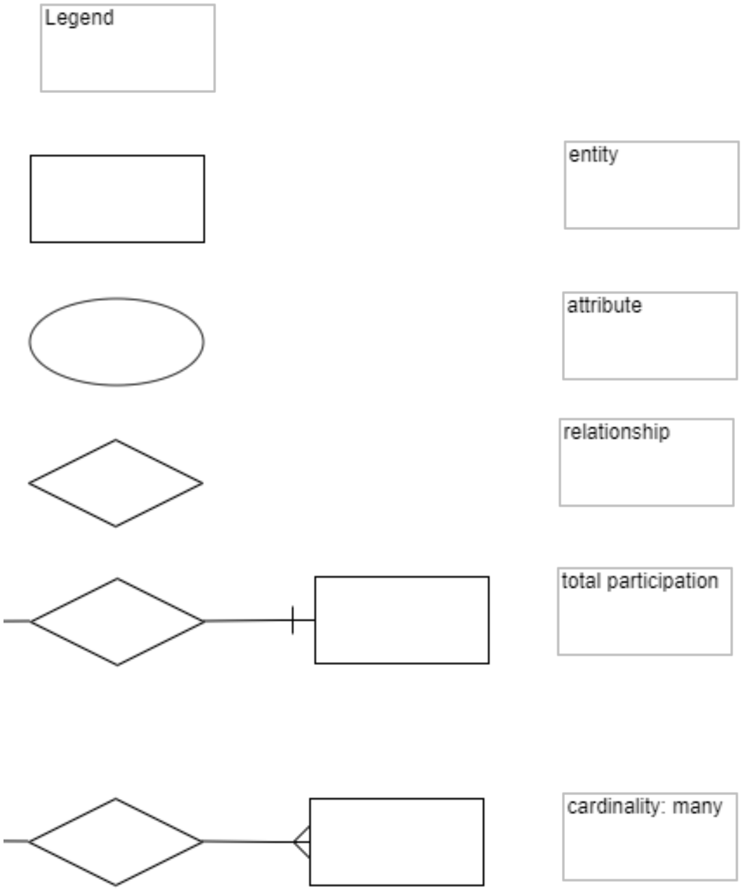
In order to fulfil the goals of the project, a proper website had to be designed which not only had to be simple to use but also satisfy all the user requirements. The website design was chosen to be the most simplistic as possible. The users are greeted with a interactive home page where the user is prompted to select the state and the city the user choses to stay at. The user is then prompted with the options which asks the user about the number of guests and the number of days user wants to stay. Sql queries are run within the php script that pulls information from the database and displays the hotels that are relevant to the city and state the user chooses. An html code that is run within the php script creates the tables that are necessary for the hotels to be seen. The user is then given the option to select the hotel of his or her choice and then view the rooms available. Another query is run in order to find the available rooms of that particular hotel and the results are displayed on the screen. The user gets a choice to select that particular room and book the room. Once the user confirms a php script is run to update that database to mark that room as booked and registers that room under the current user that was logged in. The user can go under his booking history to view all the bookings that were made by that user.

The website would offer an XML import option to constantly update the database with new hotel information. Also able to generate its own XML file in order for the website to share the information with a third party.

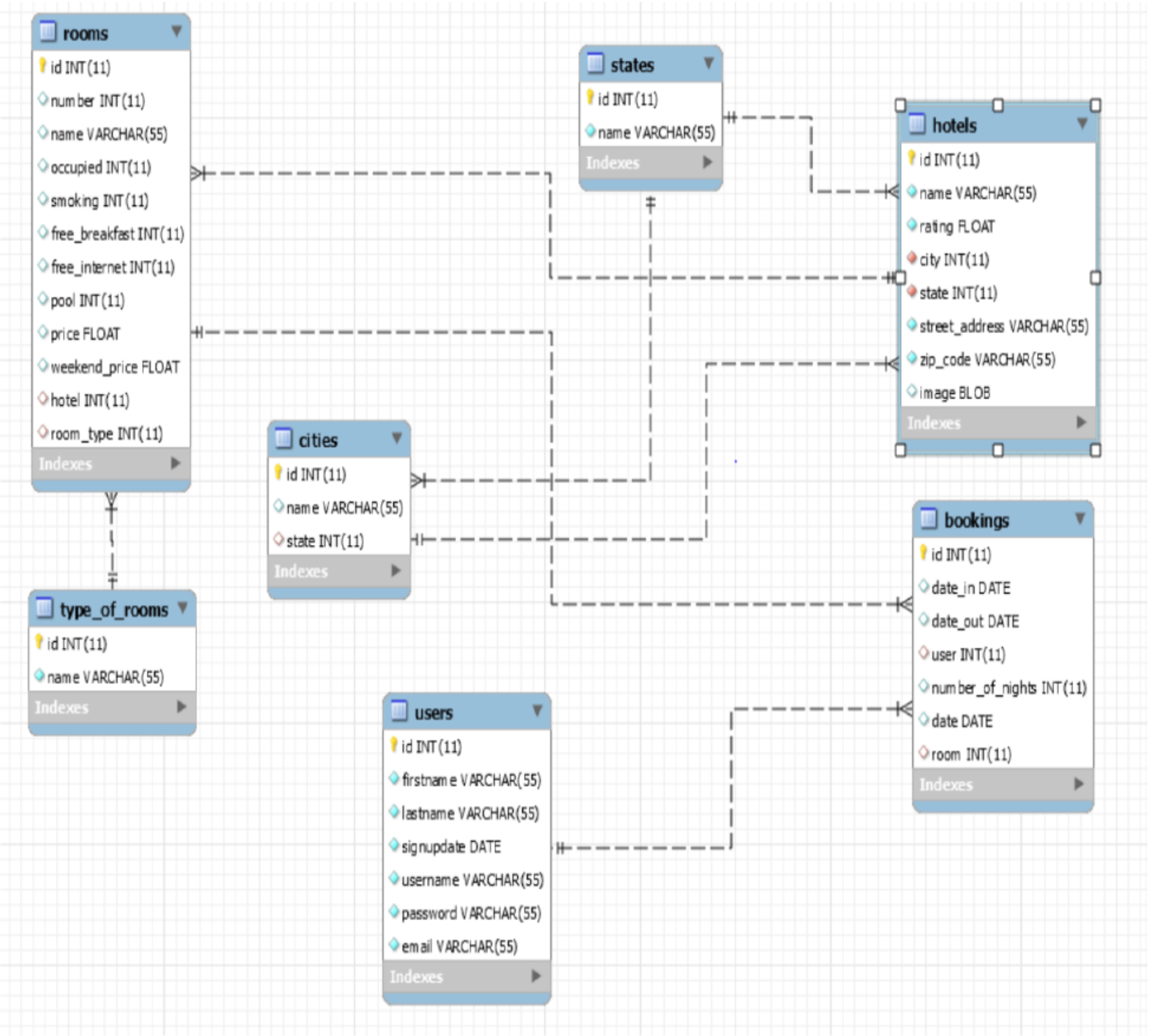
Schematics

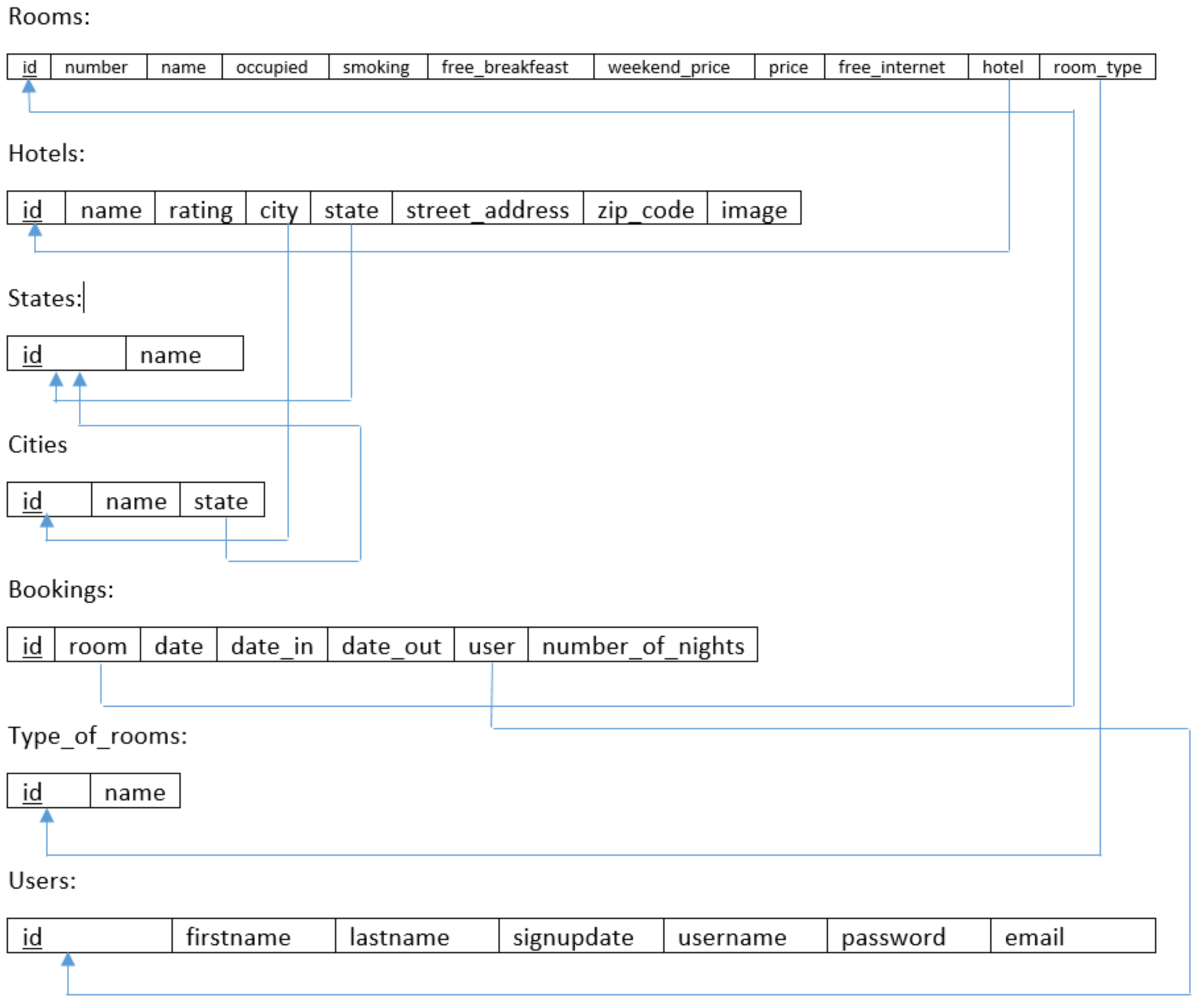
ER Schema Diagram:


room_type
smoking
free_breakfeast
id
occupied
weekend_price
price
free_internet
hotel
name
rooms
number
pool
Relationship
type_of_room
name
id
id
name
states
Relationship
Hotels
id
image
name
zip_code
rating
city
state
street_address
Relationship
Relationship
name
id
state
cities
Relationship
Relationship
id
signupdate
firstname
email
Users
lastname
password
username
Relationship
id
room
Bookings
date
date_in
user
number_of_nights
date_out


Legend
entity
attribute
relationship
total participation
cardinality: many

Relational Diagram





Rooms:

id	number	name	occupied	smoking	free_breakfeast	weekend_price	price	free_internet	hotel	room_type

Hotels:

id	name	rating	city	state	street_address	zip_code	image

States:

id	name

Cities

id	name	state

Bookings:

id	room	date	date_in	date_out	user	number_of_nights

Type_of_rooms:

id	name

Users:

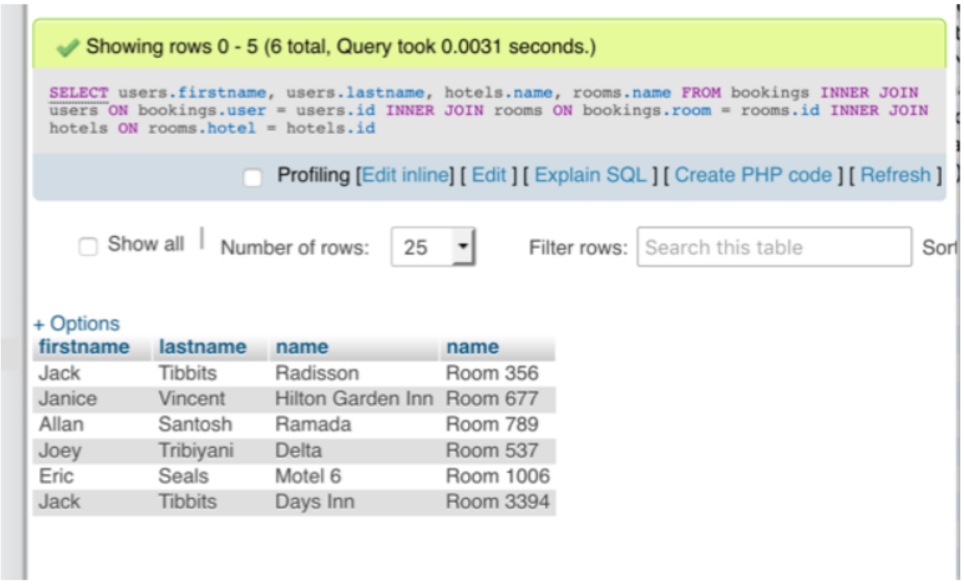
id	firstname	lastname	signupdate	username	password	email

Results of the Queries Used

View 1: Computes a join of at least three tables

The following view shows the first name, last name, the hotel name and the name of the room of all the bookings in the booking table. This is retrieved by joining users, room and hotels table.

```
SELECT users.firstname, users.lastname, hotels.name, rooms.name
FROM bookings
INNER JOIN users ON bookings.user = users.id
INNER JOIN rooms ON bookings.room = rooms.id
INNER JOIN hotels ON rooms.hotel = hotels.id;
```



✓ Showing rows 0 - 5 (6 total, Query took 0.0031 seconds.)

```
SELECT users.firstname, users.lastname, hotels.name, rooms.name FROM bookings INNER JOIN
users ON bookings.user = users.id INNER JOIN rooms ON bookings.room = rooms.id INNER JOIN
hotels ON rooms.hotel = hotels.id
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Filter rows: Search this table
Sor
+ Options

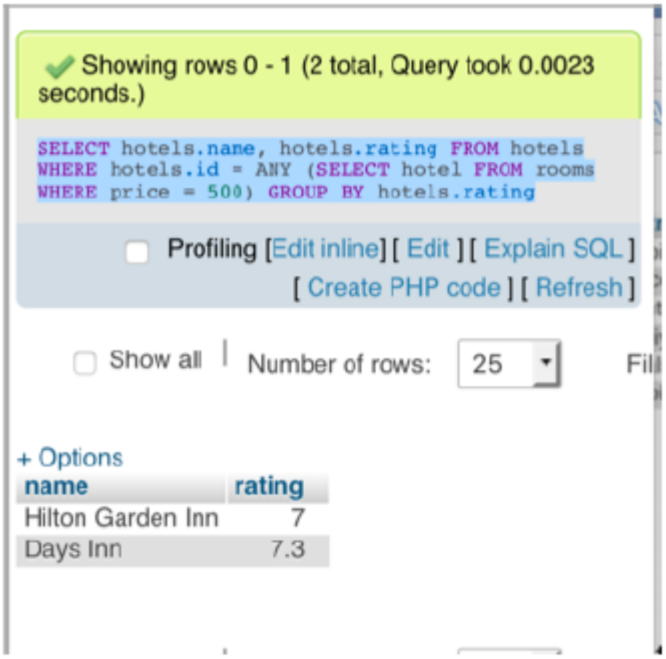
Number of rows: 25

firstname	lastname	name	name
Jack	Tibbits	Radisson	Room 356
Janice	Vincent	Hilton Garden Inn	Room 677
Allan	Santosh	Ramada	Room 789
Joey	Tribiyani	Delta	Room 537
Eric	Seals	Motel 6	Room 1006
Jack	Tibbits	Days Inn	Room 3394

View 2: Uses nested queries with ANY or ALL operator and uses a GROUP BY clause.

The following view shows the name of the hotel and the rating of a hotel (grouped by rating) where the price of any room is equal to 500.

```
SELECT hotels.name, hotels.rating FROM hotels WHERE hotels.id = ANY (SELECT hotel FROM
rooms WHERE price = 500) GROUP BY hotels.rating;
```



✔ Showing rows 0 - 1 (2 total, Query took 0.0023 seconds.)

```
SELECT hotels.name, hotels.rating FROM hotels
WHERE hotels.id = ANY (SELECT hotel FROM rooms
WHERE price = 500) GROUP BY hotels.rating
```

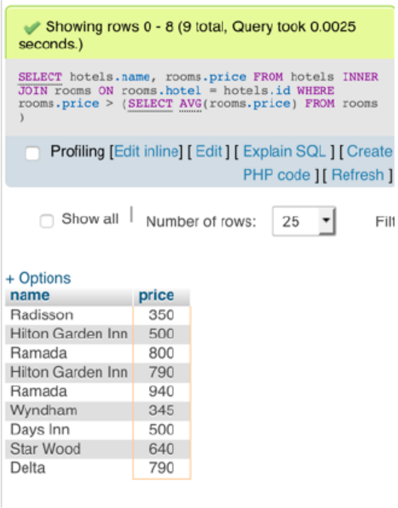
□ Profiling [Edit inline] [Edit] [Explain SQL]
[Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Fil
+ Options

name	rating
Hilton Garden Inn	7
Days Inn	7.3

View 3: A correlated nested query

The following view shows the name of the hotel and prices of rooms. The list only shows the name of the hotels where the price of the room is greater than the average price of all the rooms in all the hotels. Here the nested query is the calculation of the average price of all the rooms in the hotel.

SELECT hotels.name, rooms.price FROM hotels INNER JOIN rooms ON rooms.hotel = hotels.id WHERE rooms.price > (SELECT AVG(rooms.price) FROM rooms)



✓ Showing rows 0 - 8 (9 total, Query took 0.0025 seconds.)

```
SELECT hotels.name, rooms.price FROM hotels INNER
JOIN rooms ON rooms.hotel = hotels.id WHERE
rooms.price > (SELECT AVG(rooms.price) FROM rooms
)
```

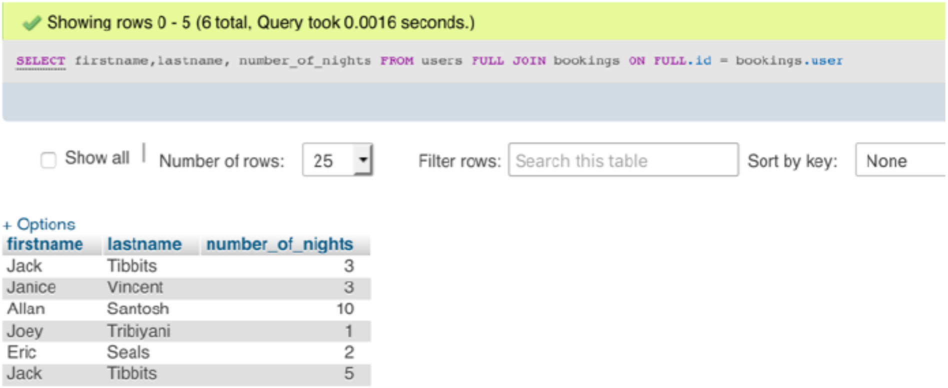
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filt
+ Options

name	price
Radisson	350
Hilton Garden Inn	500
Ramada	800
Hilton Garden Inn	790
Ramada	940
Wyndham	345
Days Inn	500
Star Wood	640
Delta	790

View 4: Uses a FULL JOIN

The following view shows the firstname last name of the users and the number of nights spending in hotel.

SELECT firstname,lastname, number_of_nights
FROM users
FULL JOIN bookings ON FULL.id = bookings.user;



✓ Showing rows 0 - 5 (6 total, Query took 0.0016 seconds.)

```
SELECT firstname,lastname, number_of_nights FROM users FULL JOIN bookings ON FULL.id = bookings.user
```

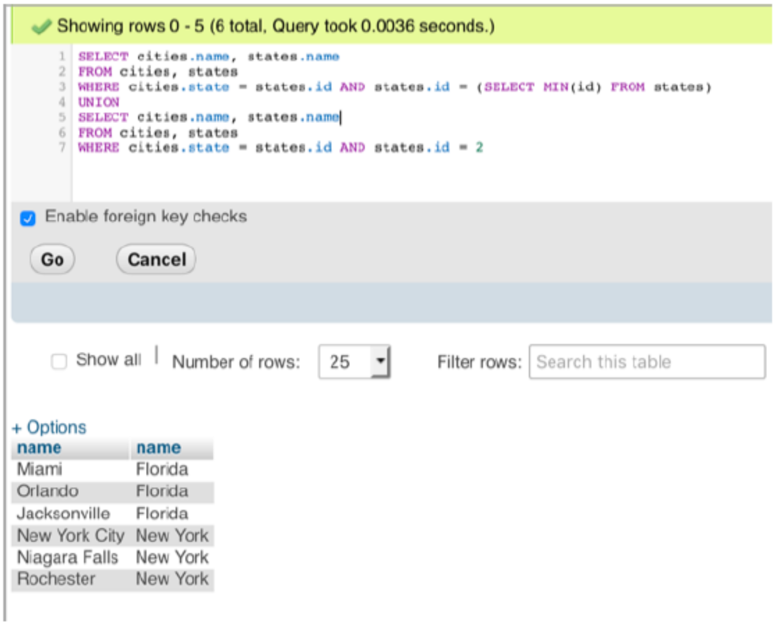
□ Show all | Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options

firstname	lastname	number_of_nights
Jack	Tibbits	3
Janice	Vincent	3
Allan	Santosh	10
Joey	Tribiyani	1
Eric	Seals	2
Jack	Tibbits	5

View 5: Uses nested queries with any of the set operations UNION, EXCEPT, or INTERSECT

The following is the view of city names and the corresponding state name where the state id is the min value of state.ID which is 1, unioned with the names of the cities and states where state id = 2

```
SELECT cities.name, states.name
FROM cities, states
WHERE cities.state = states.id AND states.id = (SELECT MIN(id) FROM states)
UNION
SELECT cities.name, states.name
FROM cities, states
WHERE cities.state = states.id AND states.id = 2
```



✓ Showing rows 0 - 5 (6 total, Query took 0.0036 seconds.)

```
SELECT cities.name, states.name
FROM cities, states
WHERE cities.state = states.id AND states.id = (SELECT MIN(id) FROM states)
UNION
SELECT cities.name, states.name
FROM cities, states
WHERE cities.state = states.id AND states.id = 2
```

☑ Enable foreign key checks

□ Show all | Number of rows: 25

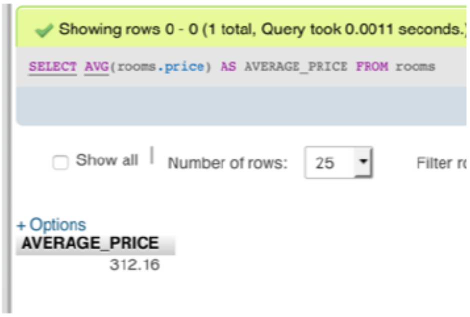
Filter rows: Search this table

name	name
Miami	Florida
Orlando	Florida
Jacksonville	Florida
New York City	New York
Niagara Falls	New York
Rochester	New York

View 6: Using the AVG

The following view shows the average price of all the room under the title of AVERAGE_PRICE.

SELECT AVG(rooms.price) AS AVERAGE_PRICE FROM rooms;



✓ Showing rows 0 - 0 (1 total, Query took 0.0011 seconds.)

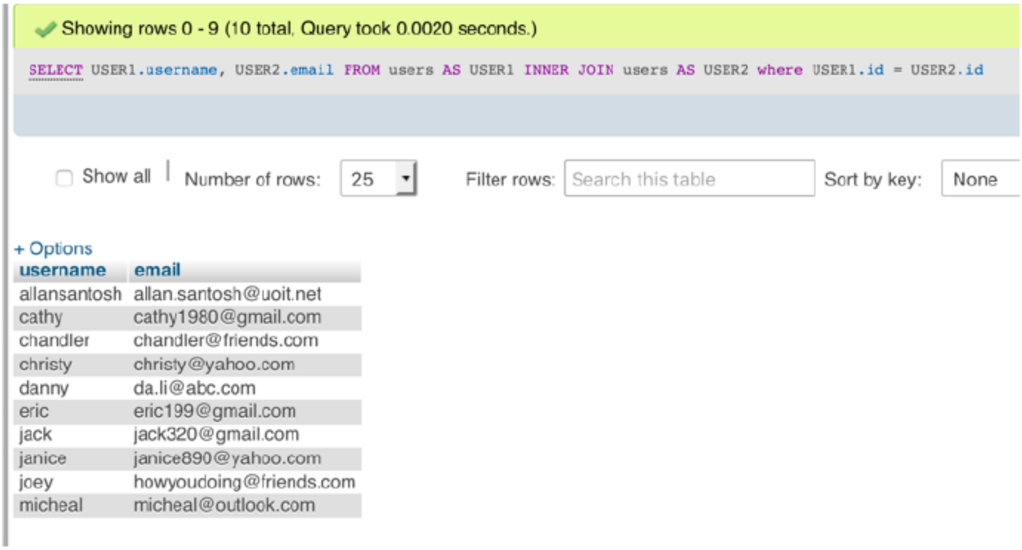
```
SELECT AVG(rooms.price) AS AVERAGE_PRICE FROM rooms
```

AVERAGE_PRICE
312.16

View 7: Use INNER JOIN to join the same table and retrieve results

Username and email is retrieved from the users table using the INNER JOIN

SELECT USER1.username, USER2.email
FROM users AS USER1 INNER JOIN users AS USER2
where USER1.id = USER2.id;



✓ Showing rows 0 - 9 (10 total, Query took 0.0020 seconds.)

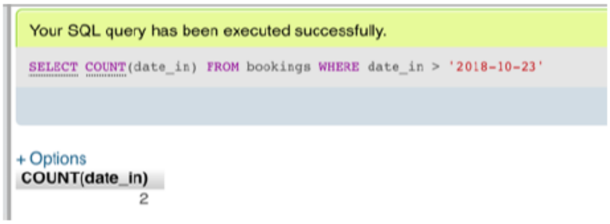
```
SELECT USER1.username, USER2.email FROM users AS USER1 INNER JOIN users AS USER2 where USER1.id = USER2.id
```

username	email
allansantosh	allan.santosh@uoit.net
cathy	cathy1980@gmail.com
chandler	chandler@friends.com
christy	christy@yahoo.com
danny	da.li@abc.com
eric	eric199@gmail.com
jack	jack320@gmail.com
janice	janice890@yahoo.com
joey	howyoudoing@friends.com
micheal	micheal@outlook.com

View 8: Use COUNT

The following view counts the number of times there is a booking in the bookings table where the date someone checked in is 23-08-2018.

```
SELECT COUNT(date_in)
FROM bookings
WHERE date_in > '2018-10-23';
```



Your SQL query has been executed successfully.

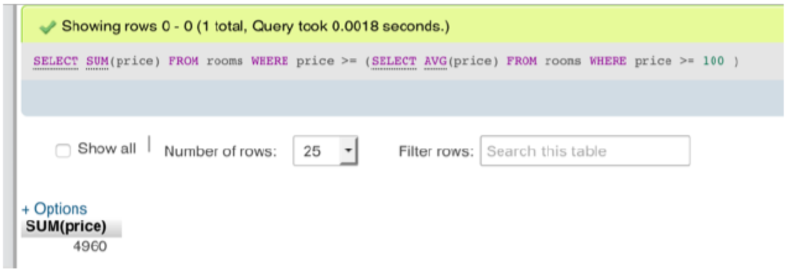
```
SELECT COUNT(date_in) FROM bookings WHERE date_in > '2018-10-23'
```

COUNT(date_in)
2

View 9: Use SUM

The following view sums the prices of all the prices greater than or equal to the average prices of all the rooms in all the hotels where price is greater than equal to 100.

```
SELECT SUM(price) FROM rooms WHERE price >= (SELECT AVG(price) FROM rooms WHERE price
>= 100 )
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0018 seconds.)

```
SELECT SUM(price) FROM rooms WHERE price >= (SELECT AVG(price) FROM rooms WHERE price >= 100 )
```

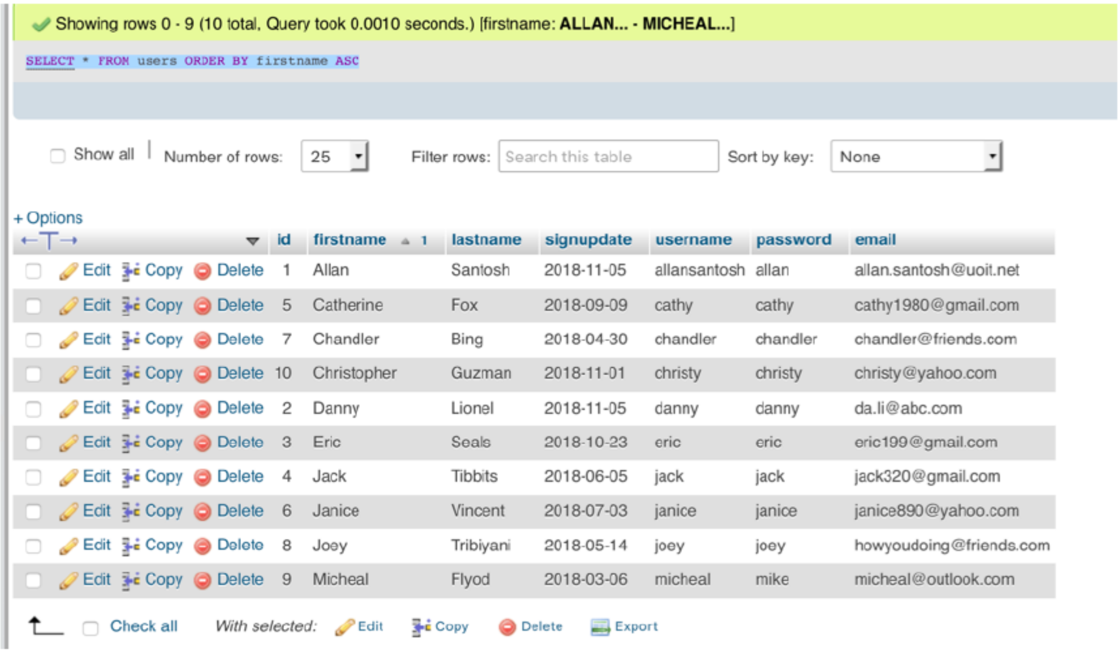
□ Show all | Number of rows: 25 Filter rows: Search this table

SUM(price)
4960

View 10: List all users details in ascending order

Using ORDER BY column name ASC we can list all the users in ascending order based on the specified column name.

SELECT * FROM users ORDER BY firstname ASC



✓ Showing rows 0 - 9 (10 total, Query took 0.0010 seconds.) [firstname: **ALLAN... - MICHEAL...**]

```
SELECT * FROM users ORDER BY firstname ASC
```

	id	firstname ▲ 1	lastname	signupdate	username	password	email
□ Edit Copy Delete	1	Allan	Santosh	2018-11-05	allansantosh	allan	allan.santosh@uoit.net
□ Edit Copy Delete	5	Catherine	Fox	2018-09-09	cathy	cathy	cathy1980@gmail.com
□ Edit Copy Delete	7	Chandler	Bing	2018-04-30	chandler	chandler	chandler@friends.com
□ Edit Copy Delete	10	Christopher	Guzman	2018-11-01	christy	christy	christy@yahoo.com
□ Edit Copy Delete	2	Danny	Lionel	2018-11-05	danny	danny	da.li@abc.com
□ Edit Copy Delete	3	Eric	Seals	2018-10-23	eric	eric	eric199@gmail.com
□ Edit Copy Delete	4	Jack	Tibbits	2018-06-05	jack	jack	jack320@gmail.com
□ Edit Copy Delete	6	Janice	Vincent	2018-07-03	janice	janice	janice890@yahoo.com
□ Edit Copy Delete	8	Joey	Tribiyani	2018-05-14	joey	joey	howyoudoing@friends.com
□ Edit Copy Delete	9	Micheal	Flyod	2018-03-06	micheal	mike	micheal@outlook.com

Thoughts About Any Future Work

With our initial design and implementation complete, we have more refining to do before we can try and present it to hotel brands and attempt to gather clients. We also have to find a way of advertising our product or another method of client acquisition. With a polished proof of concept we will be able to demonstrate what our database can do and why hotel companies will benefit from adopting our product.

Improvements:

There are entities and attributes we may want to add in the future. Certain hotels may offer packages which include access to certain amenities within the hotel or car rentals, etc. We can either add these as attributes to the hotel or as a separate entity or relation.

Another possibility is to add the option for hotels to add their own entities pending our approval. This way if their hotel offers features that are not currently in our database they will be able to add them.

Proposal to Hotel companies:

Without final working database application we will be able to present it to hotels in a proposal presentation. In the presentation we will be able to discuss with them the improvements that our database system has over their current system. For this we will research the specific hotels beforehand and what system they are currently using.

One of the main selling points of our database system is that it will give hotel companies exposure to a large client base. Hotels typically spend 4-5% of revenue on advertising and marketing. Their aim, amongst other things, is to establish their brand and also to acquire customers. Our database system will allow hotels to display their information to a large audience. This will allow hotels to save money on marketing and allow them to allocate their resources towards improving their hotels and offering better prices.

Conclusion with Contributions Made

Overall, we are satisfied with the quick and effective access that our Hotel database can offer. Our wide selection of data provides us with the opportunity to see many different views, which can be useful to see trends and patterns of the hotel industry.

Equal Contributions were made by all member of the group

References

https://www.w3schools.com/

https://www.tutorialspoint.com/php/php_and_xml.htm