HOTEL DATABASE MANAGEMENT SYSTEM

Prepared By – Vaibhavi More & Sweta Gupta

Table of contents

In a nutshell, this shows what we are going to cover today in our presentation of "Hotel Database Management System".
It will be around 10 minutes presentation. We would be more than happy to answer your questions at the end of the presentation!

1. Introduction
2. ERR Diagram
3. Tables & Relationships
4. Normalization
5. Queries, Views, Triggers
6. Challenges
7. Questions

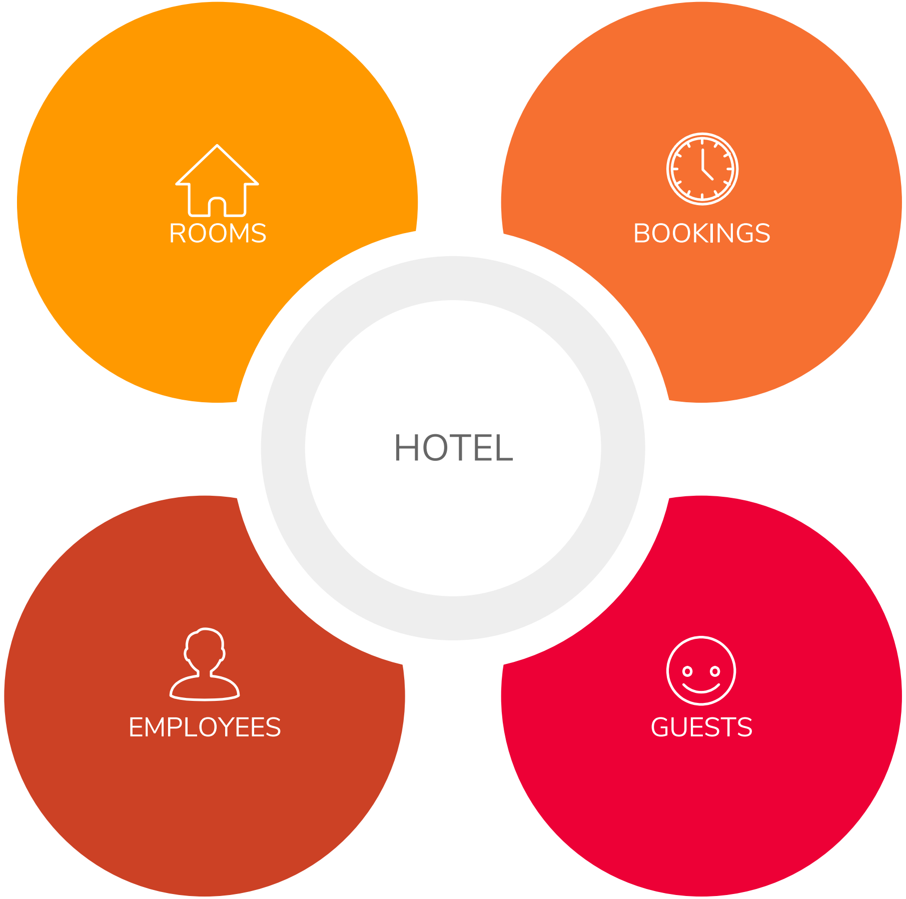
The main objective of this project is to create a database management system for a hotel.

We have build this project as a group of two.
The group members are:

1. Vaibhavi More
2. Sweta Gupta

It was fun working together, overcoming each other’s flaws together and learning from each other's strengths in respective areas of Database Design & Management.

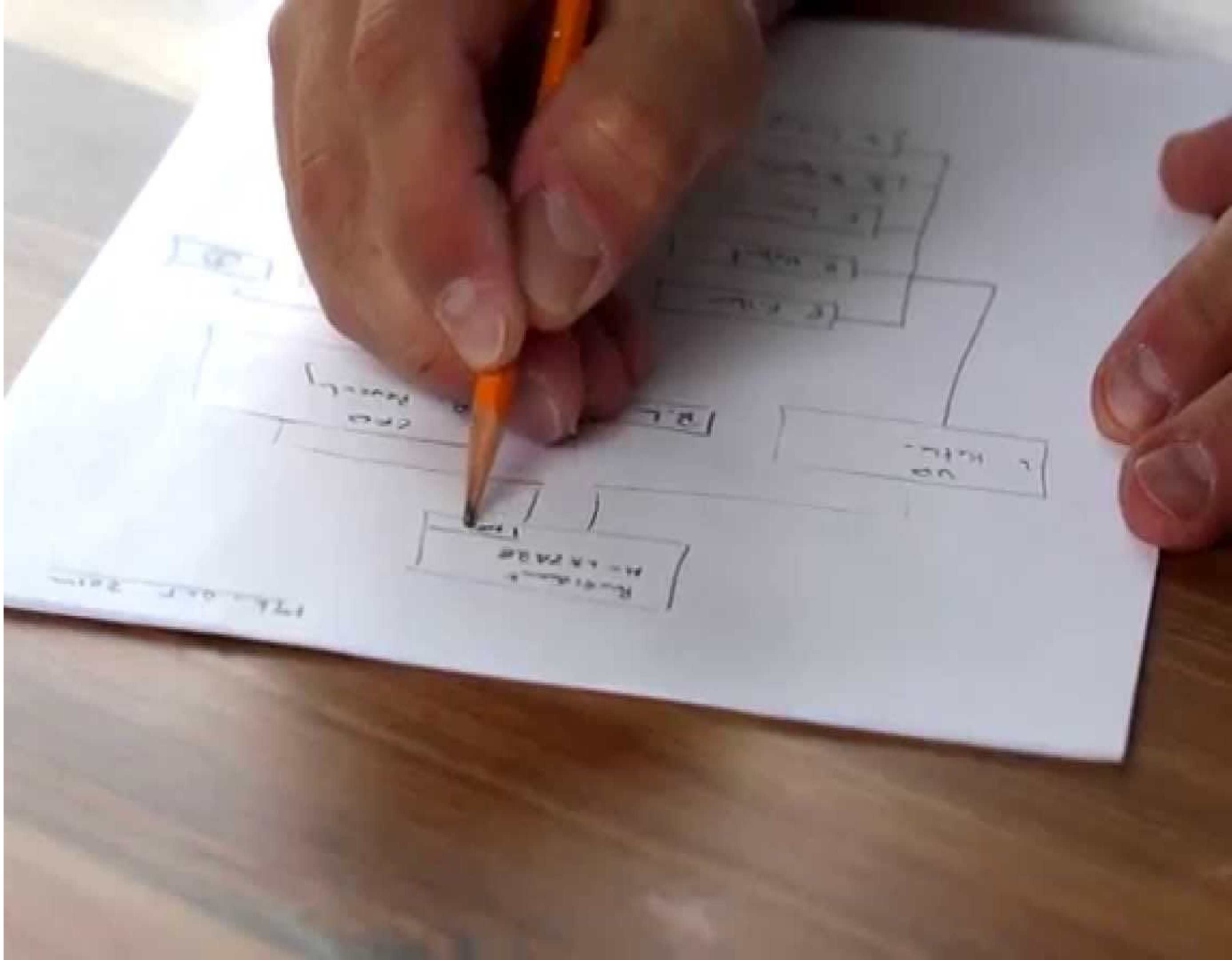
A Hotel consists of a wide areas to manage. We tried to include the main areas for a hotel management system in this project.

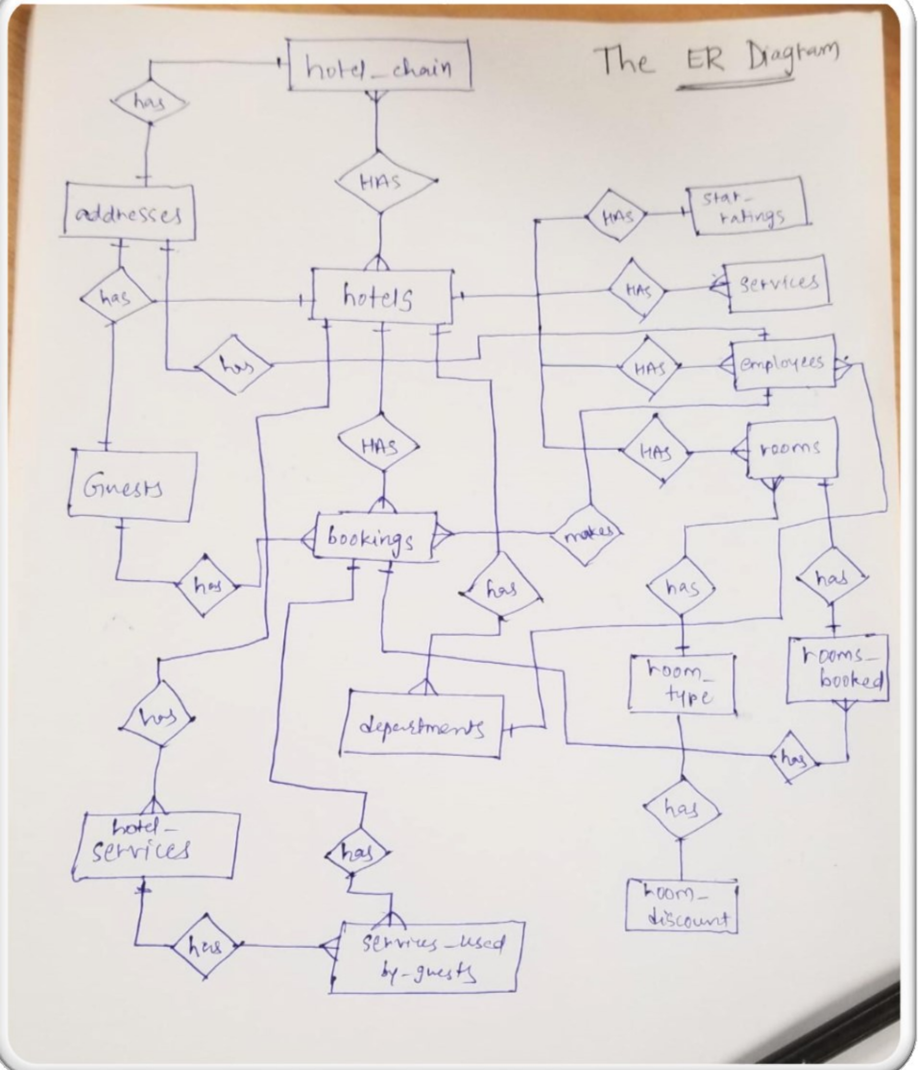


Step:1

ERR-Diagram

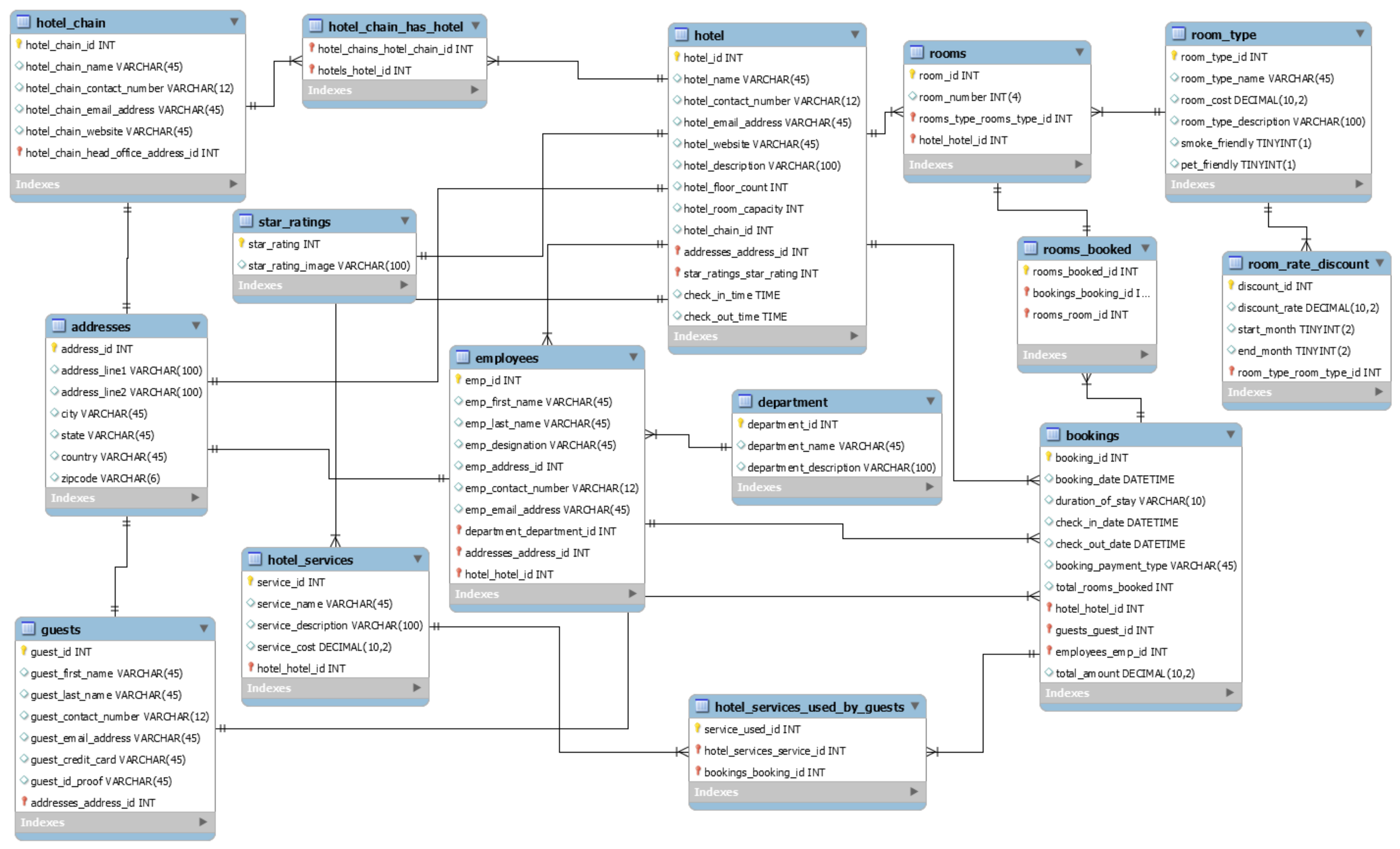
We drew the ERR diagram on a paper, noting down all the tables required





It looks messy, right?

Yeah, we better look at the ERR diagram !



hotel_chain
- hotel_chain_id INT
- hotel_chain_name VARCHAR(45)
- hotel_chain_contact_number VARCHAR(12)
- hotel_chain_email_address VARCHAR(45)
- hotel_chain_website VARCHAR(45)
- hotel_chain_head_office_address_id INT

Indexes

hotel_chain_has_hotel
- hotel_chains_hotel_chain_id INT
- hotels_hotel_id INT

Indexes

hotel
- hotel_id INT
- hotel_name VARCHAR(45)
- hotel_contact_number VARCHAR(12)
- hotel_email_address VARCHAR(45)
- hotel_website VARCHAR(45)
- hotel_description VARCHAR(100)
- hotel_floor_count INT
- hotel_room_capacity INT
- hotel_chain_id INT
- addresses_address_id INT
- star_ratings_star_rating INT
- check_in_time TIME
- check_out_time TIME

Indexes

rooms
- room_id INT
- room_number INT(4)
- rooms_type_rooms_type_id INT
- hotel_hotel_id INT

Indexes

room_type
- room_type_id INT
- room_type_name VARCHAR(45)
- room_cost DECIMAL(10,2)
- room_type_description VARCHAR(100)
- smoke_friendly TINYINT(1)
- pet_friendly TINYINT(1)

Indexes

star_ratings
- star_rating INT
- star_rating_image VARCHAR(100)

Indexes

rooms_booked
- rooms_booked_id INT
- bookings_booking_id I...
- rooms_room_id INT

Indexes

room_rate_discount
- discount_id INT
- discount_rate DECIMAL(10,2)
- start_month TINYINT(2)
- end_month TINYINT(2)
- room_type_room_type_id INT

Indexes

addresses
- address_id INT
- address_line1 VARCHAR(100)
- address_line2 VARCHAR(100)
- city VARCHAR(45)
- state VARCHAR(45)
- country VARCHAR(45)
- zipcode VARCHAR(6)

Indexes

employees
- emp_id INT
- emp_first_name VARCHAR(45)
- emp_last_name VARCHAR(45)
- emp_designation VARCHAR(45)
- emp_address_id INT
- emp_contact_number VARCHAR(12)
- emp_email_address VARCHAR(45)
- department_department_id INT
- addresses_address_id INT
- hotel_hotel_id INT

Indexes

department
- department_id INT
- department_name VARCHAR(45)
- department_description VARCHAR(100)

Indexes

bookings
- booking_id INT
- booking_date DATETIME
- duration_of_stay VARCHAR(10)
- check_in_date DATETIME
- check_out_date DATETIME
- booking_payment_type VARCHAR(45)
- total_rooms_booked INT
- hotel_hotel_id INT
- guests_guest_id INT
- employees_emp_id INT
- total_amount DECIMAL(10,2)

Indexes

hotel_services
- service_id INT
- service_name VARCHAR(45)
- service_description VARCHAR(100)
- service_cost DECIMAL(10,2)
- hotel_hotel_id INT

Indexes

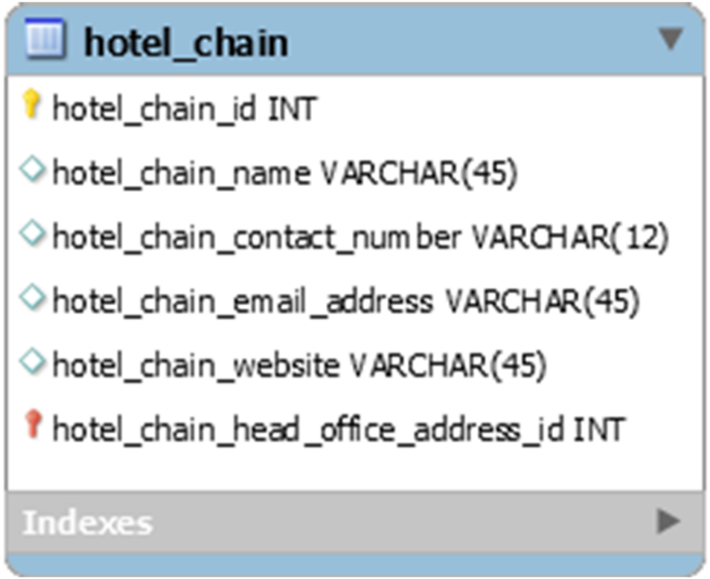
guests
- guest_id INT
- guest_first_name VARCHAR(45)
- guest_last_name VARCHAR(45)
- guest_contact_number VARCHAR(12)
- guest_email_address VARCHAR(45)
- guest_credit_card VARCHAR(45)
- guest_id_proof VARCHAR(45)
- addresses_address_id INT

Indexes

hotel_services_used_by_guests
- service_used_id INT
- hotel_services_service_id INT
- bookings_booking_id INT

Indexes

Let us begin exploring the TABLES



hotel_chain

hotel_chain_id INT

hotel_chain_name VARCHAR(45)

hotel_chain_contact_number VARCHAR(12)

hotel_chain_email_address VARCHAR(45)

hotel_chain_website VARCHAR(45)

hotel_chain_head_office_address_id INT

Indexes

hotel_chain table consists of information related to a hotel chain. hotel_chain_id – This is the primary key of the table. It has Not Null constraint and Unique constraint. hotel_chain_head_office_address_id – This is a foreign key which is related to the addresses table. hotel_chain table has one-to-one relationship with the addresses table and many-to-many relationship with the hotel table. This results into a linking table hotel_chain_has_hotel.

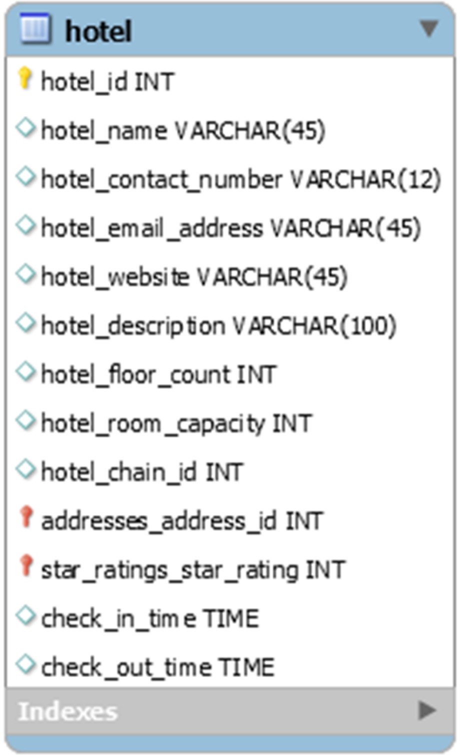
hotel table contains information about a particular hotel.

hotel_id – This is the primary key of the table. It has Not Null constraint and Unique constraint.

Addresses_address_id – This is the foreign key which is related to the addresses table.

Star_ratings_star_rating – This is the foreign key which is related to the star_ratings table.

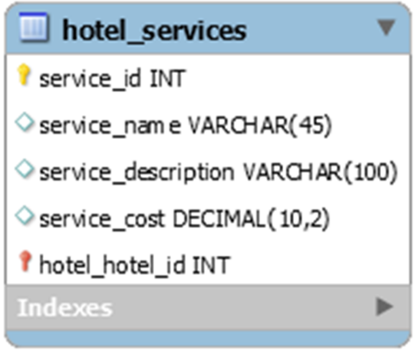
This table has many-to-many relationship with the hotel_chain table, one-to-one with star_rating, addresses, rooms_booked tables, one-to-many with employees, bookings, rooms, hotel_services tables.



hotel

hotel_id INT
hotel_name VARCHAR(45)
hotel_contact_number VARCHAR(12)
hotel_email_address VARCHAR(45)
hotel_website VARCHAR(45)
hotel_description VARCHAR(100)
hotel_floor_count INT
hotel_room_capacity INT
hotel_chain_id INT
addresses_address_id INT
star_ratings_star_rating INT
check_in_time TIME
check_out_time TIME

Indexes



hotel_services

service_id INT

service_name VARCHAR(45)

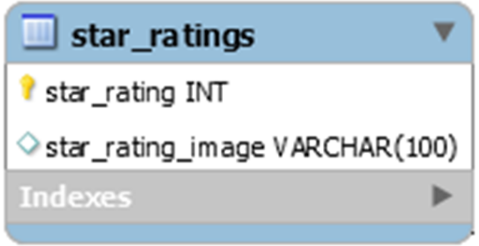
service_description VARCHAR(100)

service_cost DECIMAL(10,2)

hotel_hotel_id INT

hotel_services table consists of information for the services provided by the hotel like laundry, spa, sauna bath, gym, etc. It for service_id as the primary key and hotel_hotel_id as the foreign key which is related to the hotel table.

It holds many-to-one relationship with the hotel table.

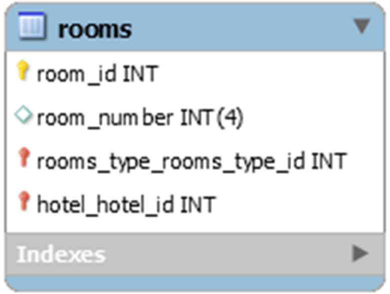


star_ratings

star_rating INT

star_rating_image VARCHAR(100)

star_ratings table consists only two columns. The star_rating column is a primary key consists of the rating of the hotel. And star_rating_image stores the image of the star_rating.
It has got one-to-one relationship with the hotels table.



rooms

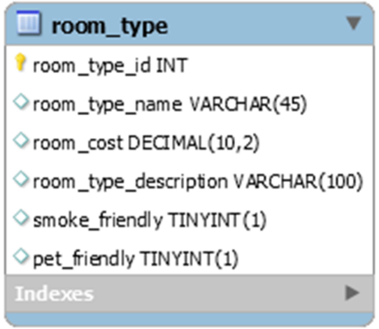
room_id INT

room_number INT(4)

rooms_type_rooms_type_id INT

hotel_hotel_id INT

Room_type table has the information about the room of each type. The primary key of the table is room_type_id.
It has got one-to-many relationship with rooms table.



room_type

room_type_id INT

room_type_name VARCHAR(45)

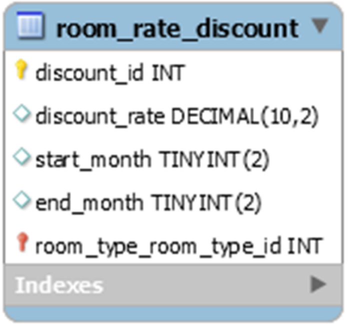
room_cost DECIMAL(10,2)

room_type_description VARCHAR(100)

smoke_friendly TINYINT(1)

pet_friendly TINYINT(1)

rooms table contains data about the rooms of the hotel. The primary key of this table is room_id and this table has two foreign keys, rooms_type_rooms_type_id and hotel_hotel_id. This table has many-to-one relationships with the hotel and room_type tables.



room_rate_discount

discount_id INT

discount_rate DECIMAL(10,2)

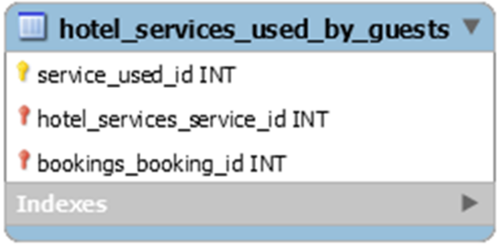
start_month TINYINT(2)

end_month TINYINT(2)

room_type_room_type_id INT

Room_rate_discount is the table that contains information about the discount depending on month of the year for each room type. The primary key of the table is discount_id and it has the foreign key rooms_type_rooms_type_id. This table has many-to-one relationship with the room_type table

hotel_services_used_by_guests table contains info about the services used by the guests. Primary key is service_used_id & two foreign keys, hotel_services_service_id, which relates to hotel_services table & bookings_booking_id relates to bookings table.

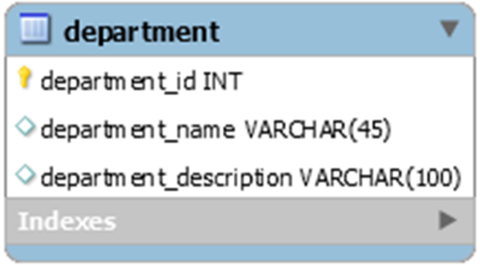


hotel_services_used_by_guests

service_used_id INT

hotel_services_service_id INT

bookings_booking_id INT



department

department_id INT

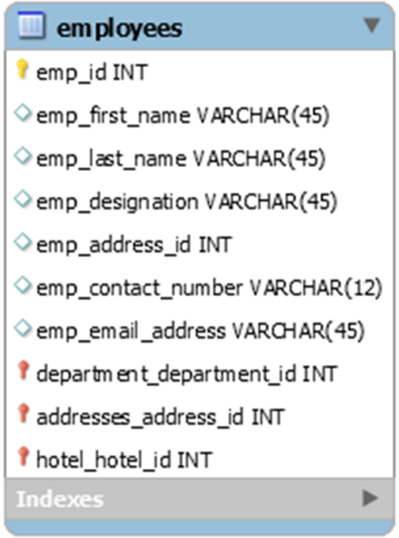
department_name VARCHAR(45)

department_description VARCHAR(100)

Indexes

department table contains the data about the different departments of the hotel. The primary key is department_id, which creates a one-to-many relationship with the employees table.

employees table consists of data related to the employees.
The primary key is employee_id. There are three foreign keys, service_id that denotes many-to-one relations with the department table.
address_id that denotes one-to-one relationship with the addresses table.
hotel_id that denotes many-to-one relationship with the hotel table.



employees

emp_id INT

emp_first_name VARCHAR(45)

emp_last_name VARCHAR(45)

emp_designation VARCHAR(45)

emp_address_id INT

emp_contact_number VARCHAR(12)

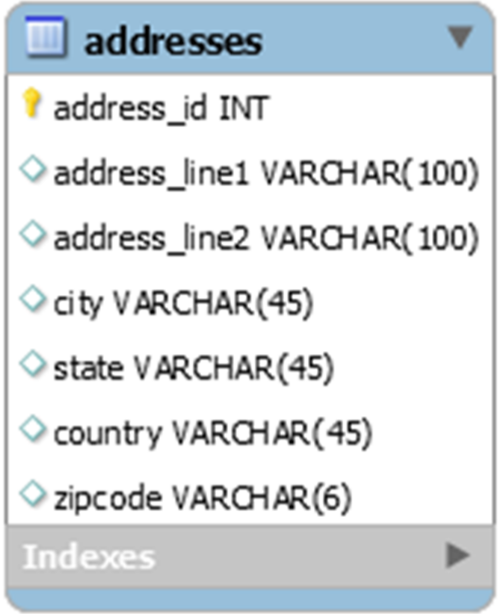
emp_email_address VARCHAR(45)

department_department_id INT

addresses_address_id INT

hotel_hotel_id INT

Indexes



addresses

address_id INT

address_line1 VARCHAR(100)

address_line2 VARCHAR(100)

city VARCHAR(45)

state VARCHAR(45)

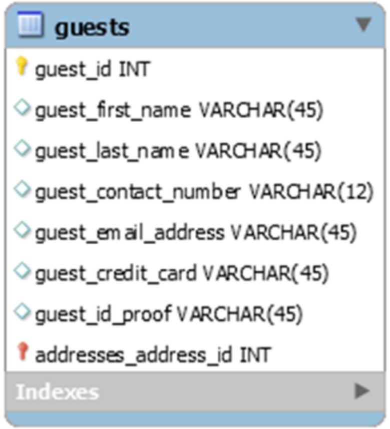
country VARCHAR(45)

zipcode VARCHAR(6)

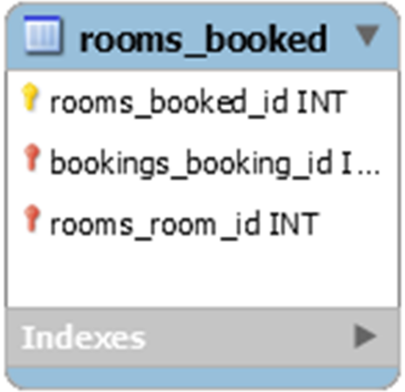
addresses table defines the information about the address of guests, hotels, hotel chains, employees.
The primary key of the table is address_id.
It maintains one-to-one relationship with tables, hotel_chain, hotel, employees and guests.

guests table has the data about the guests that check in to the hotel. The primary key of this table is guest_id.

There is one foreign key in this table, address_id that has one-to-one relationship with the address table.



```
guests
guest_id INT
guest_first_name VARCHAR(45)
guest_last_name VARCHAR(45)
guest_contact_number VARCHAR(12)
guest_email_address VARCHAR(45)
guest_credit_card VARCHAR(45)
guest_id_proof VARCHAR(45)
addresses_address_id INT
```



```
rooms_booked
rooms_booked_id INT
bookings_booking_id I...
rooms_room_id INT
```

rooms_booked table has one primary key, rooms_booked_id.

This table has 2 foreign keys, booking_id which has many-to-one relationship with the bookings table and room_id which has one-to-one-relationship with the rooms table.

“

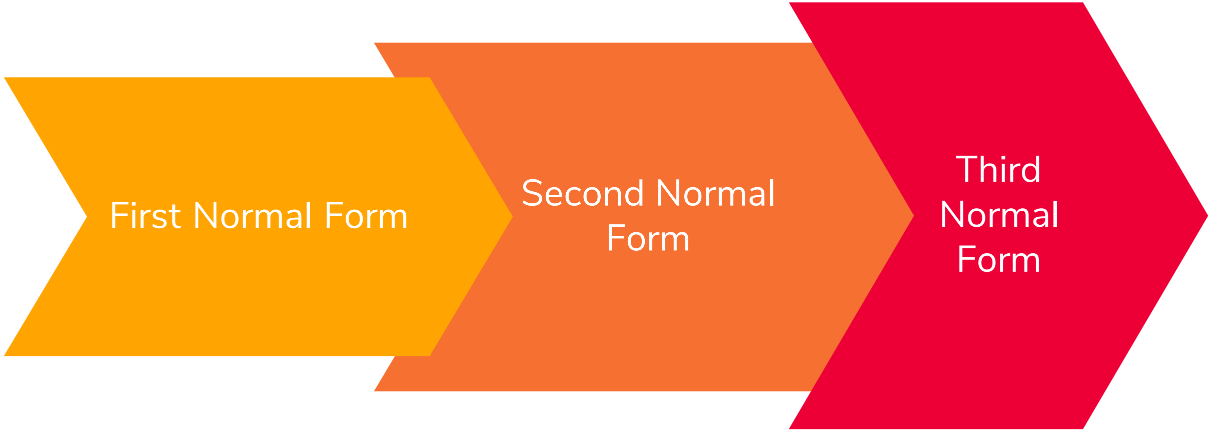
There were indeed a lot of tables to design and a lot of relationships to manage..

Relationships..

A tricky business, eh?

Normalization Process

We achieved the first normal form by keeping the data scalar.
Coming to the second normal form, we tried to make the relationships depend on the primary key.
On the third normal for, we made sure that all the dependencies are only on the primary key of the tables.



And the came our favourite part.. Writing queires was fun ☺ We also made two views and two triggers.

QUERIES

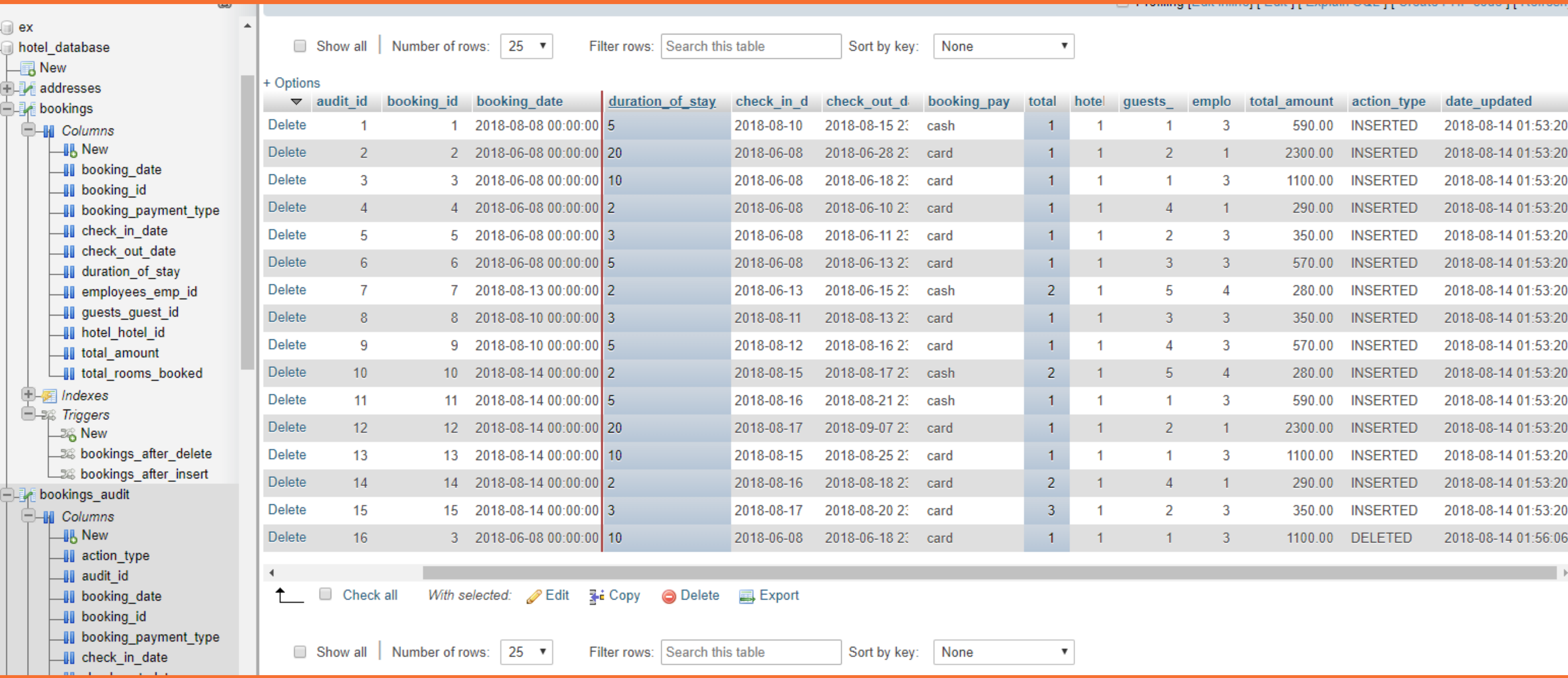
To execute the required tasks and fetch the data from one or more tables.

VIEWS

To view the details of employees along with their departments and also the details of the guests.

TRIGGERS

To create a Booking Audit table and store information about insert and delete bookings records.



- ex
- hotel_database
  - New
  - addresses
  - bookings
    - Columns
      - New
      - booking_date
      - booking_id
      - booking_payment_type
      - check_in_date
      - check_out_date
      - duration_of_stay
      - employees_emp_id
      - guests_guest_id
      - hotel_hotel_id
      - total_amount
      - total_rooms_booked
    - Indexes
    - Triggers
      - New
      - bookings_after_delete
      - bookings_after_insert
  - bookings_audit
    - Columns
      - New
      - action_type
      - audit_id
      - booking_date
      - booking_id
      - booking_payment_type
      - check_in_date

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

+ Options

	audit_id	booking_id	booking_date	duration_of_stay	check_in_d	check_out_d	booking_pay	total	hotel	guests_	emplo	total_amount	action_type	date_updated
Delete	1	1	2018-08-08 00:00:00	5	2018-08-10	2018-08-15 2	cash	1	1	1	3	590.00	INSERTED	2018-08-14 01:53:20
Delete	2	2	2018-06-08 00:00:00	20	2018-06-08	2018-06-28 2	card	1	1	2	1	2300.00	INSERTED	2018-08-14 01:53:20
Delete	3	3	2018-06-08 00:00:00	10	2018-06-08	2018-06-18 2	card	1	1	1	3	1100.00	INSERTED	2018-08-14 01:53:20
Delete	4	4	2018-06-08 00:00:00	2	2018-06-08	2018-06-10 2	card	1	1	4	1	290.00	INSERTED	2018-08-14 01:53:20
Delete	5	5	2018-06-08 00:00:00	3	2018-06-08	2018-06-11 2	card	1	1	2	3	350.00	INSERTED	2018-08-14 01:53:20
Delete	6	6	2018-06-08 00:00:00	5	2018-06-08	2018-06-13 2	card	1	1	3	3	570.00	INSERTED	2018-08-14 01:53:20
Delete	7	7	2018-08-13 00:00:00	2	2018-06-13	2018-06-15 2	cash	2	1	5	4	280.00	INSERTED	2018-08-14 01:53:20
Delete	8	8	2018-08-10 00:00:00	3	2018-08-11	2018-08-13 2	card	1	1	3	3	350.00	INSERTED	2018-08-14 01:53:20
Delete	9	9	2018-08-10 00:00:00	5	2018-08-12	2018-08-16 2	card	1	1	4	3	570.00	INSERTED	2018-08-14 01:53:20
Delete	10	10	2018-08-14 00:00:00	2	2018-08-15	2018-08-17 2	cash	2	1	5	4	280.00	INSERTED	2018-08-14 01:53:20
Delete	11	11	2018-08-14 00:00:00	5	2018-08-16	2018-08-21 2	cash	1	1	1	3	590.00	INSERTED	2018-08-14 01:53:20
Delete	12	12	2018-08-14 00:00:00	20	2018-08-17	2018-09-07 2	card	1	1	2	1	2300.00	INSERTED	2018-08-14 01:53:20
Delete	13	13	2018-08-14 00:00:00	10	2018-08-15	2018-08-25 2	card	1	1	1	3	1100.00	INSERTED	2018-08-14 01:53:20
Delete	14	14	2018-08-14 00:00:00	2	2018-08-16	2018-08-18 2	card	2	1	4	1	290.00	INSERTED	2018-08-14 01:53:20
Delete	15	15	2018-08-14 00:00:00	3	2018-08-17	2018-08-20 2	card	3	1	2	3	350.00	INSERTED	2018-08-14 01:53:20
Delete	16	3	2018-06-08 00:00:00	10	2018-06-08	2018-06-18 2	card	1	1	1	3	1100.00	DELETED	2018-08-14 01:56:06

□ Check all *With selected:* Edit Copy Delete Export

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

Challenges Faced

We faced most of the challenges in creating relationships among tables.

We need to make sure that all the relationships created among tables are logical and follow the normalization rules.

The most challenging part was creating the booking and the rooms table and its relationships with other respective tables.

> *A successful DBA*
> *makes the data*
> *easy to access*
> *and*
> *hard to lose!*

That’s it!
Thank you very much for your time!

If you have any questions regarding the presentation, please feel free to ask us!

We will be more than happy to answer you ☺