HOTEL DATABASE MANAGEMENT SYSTEM

Prepared By – Nandini Rathod

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!

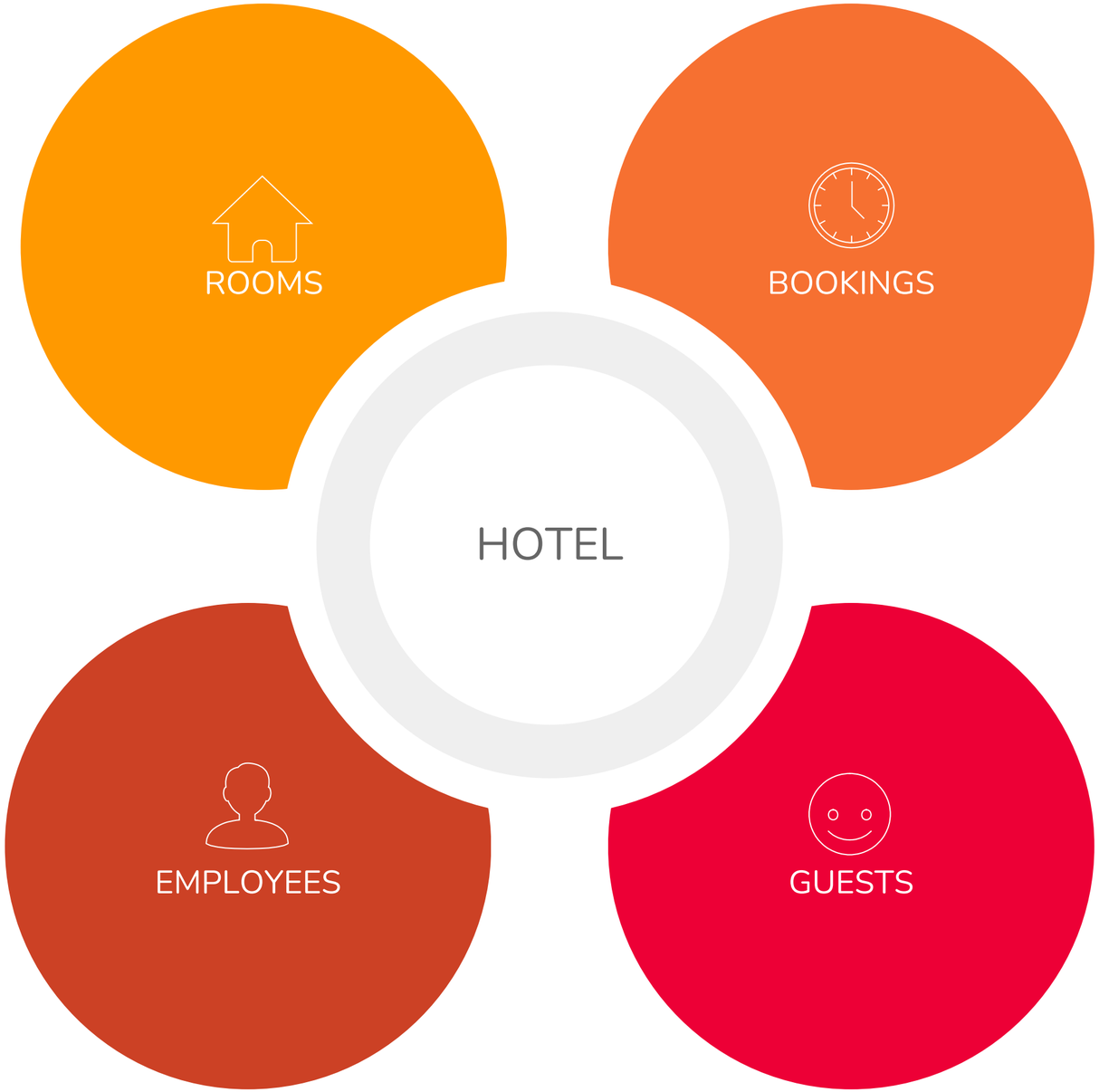
The main objective of this project is to create a database management system for a hotel.

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.

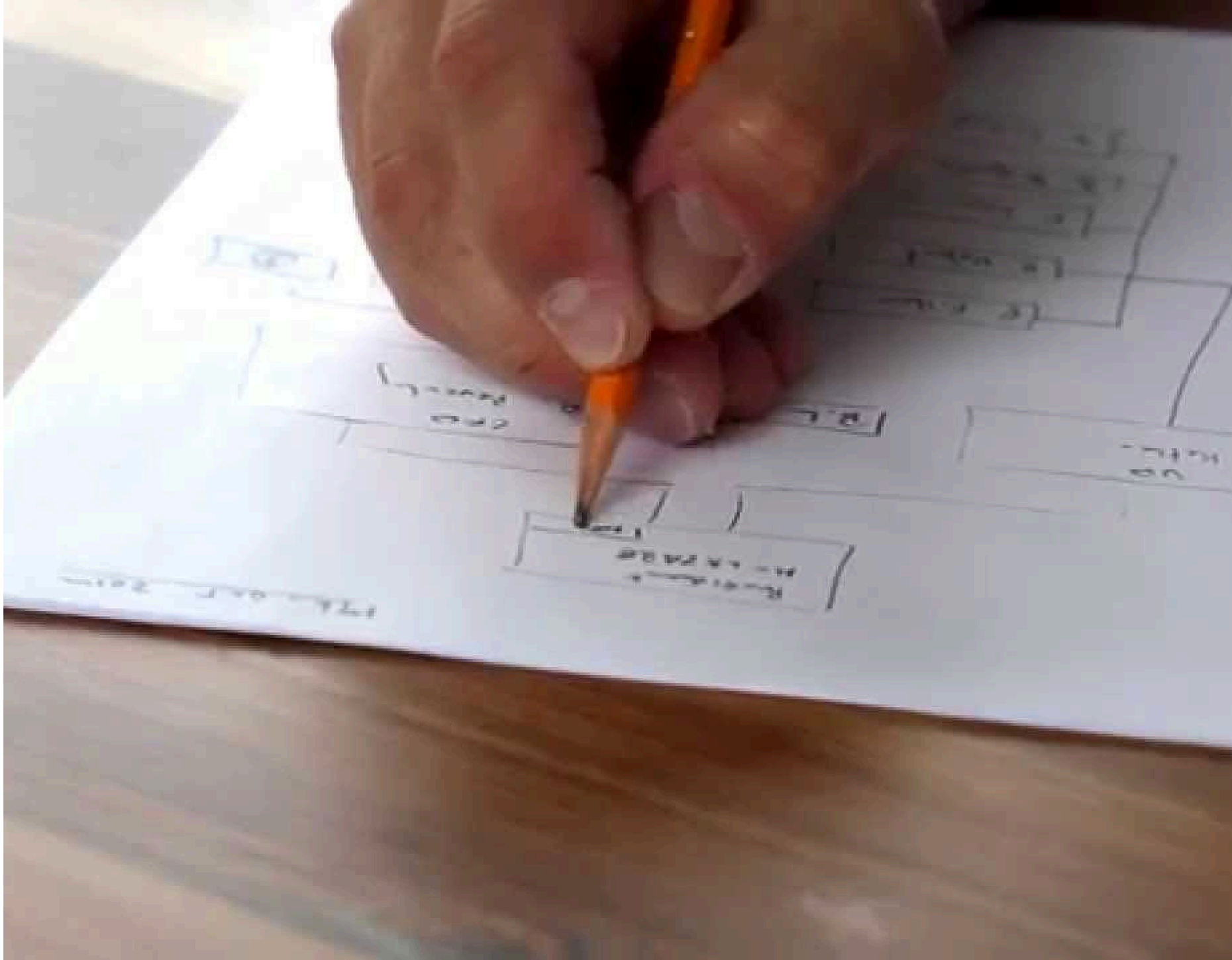
A Hotel consists of a wide areas to manage. We tried to include the main areas for a hotel management system in this project.

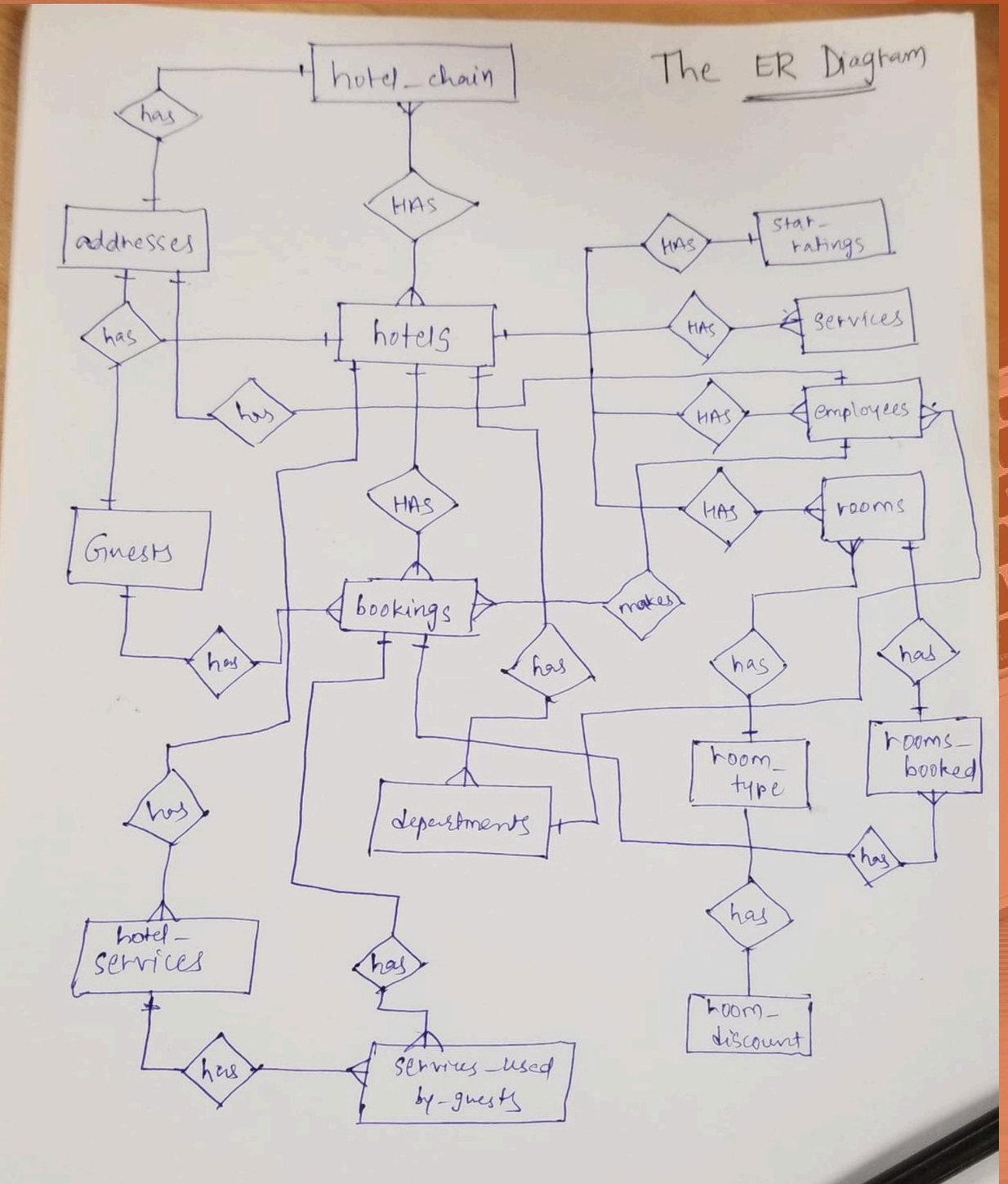


Step:1

ERR-Diagram

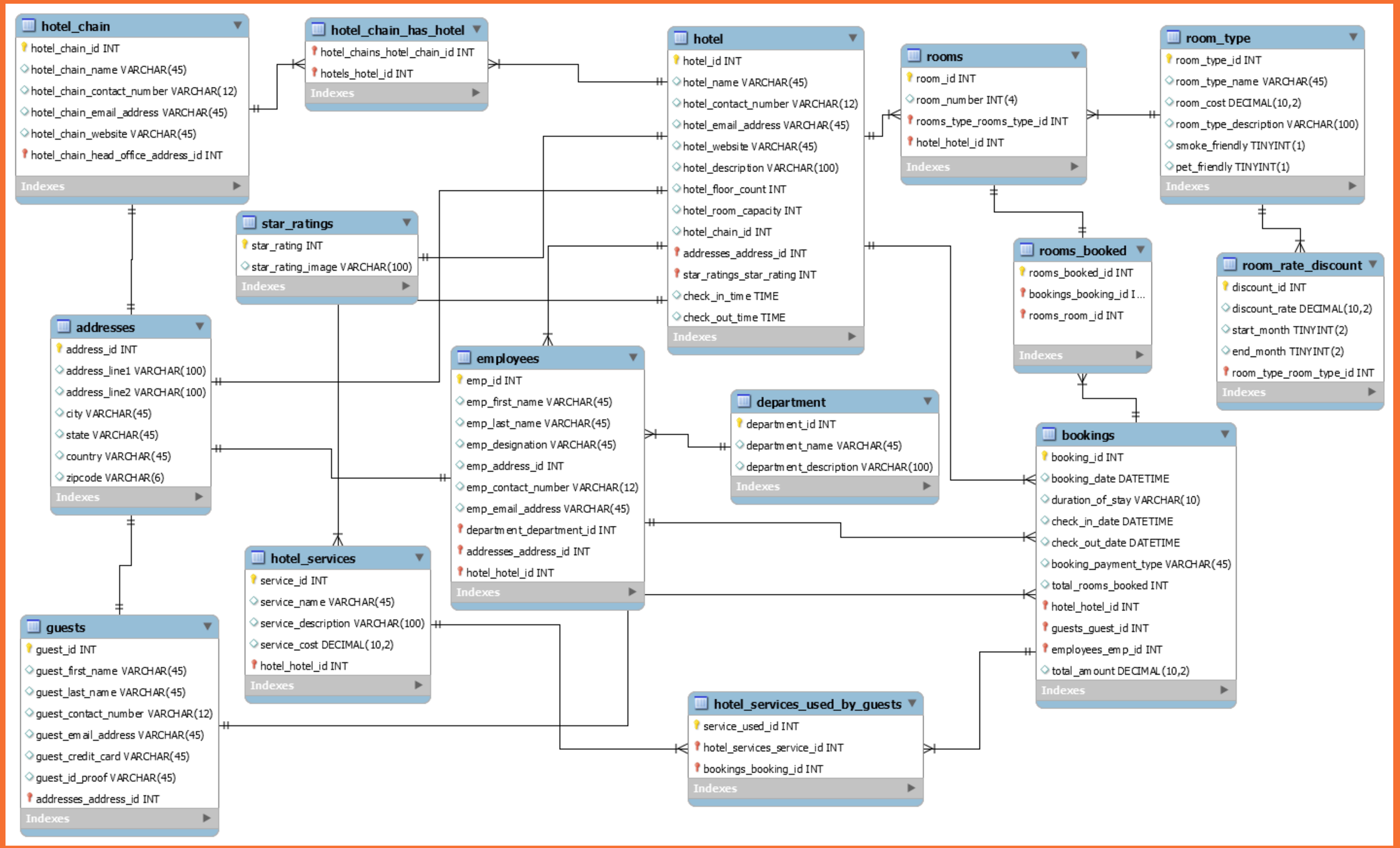
We drew the ERR diagram on a paper, noting down all the tables required





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

“

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

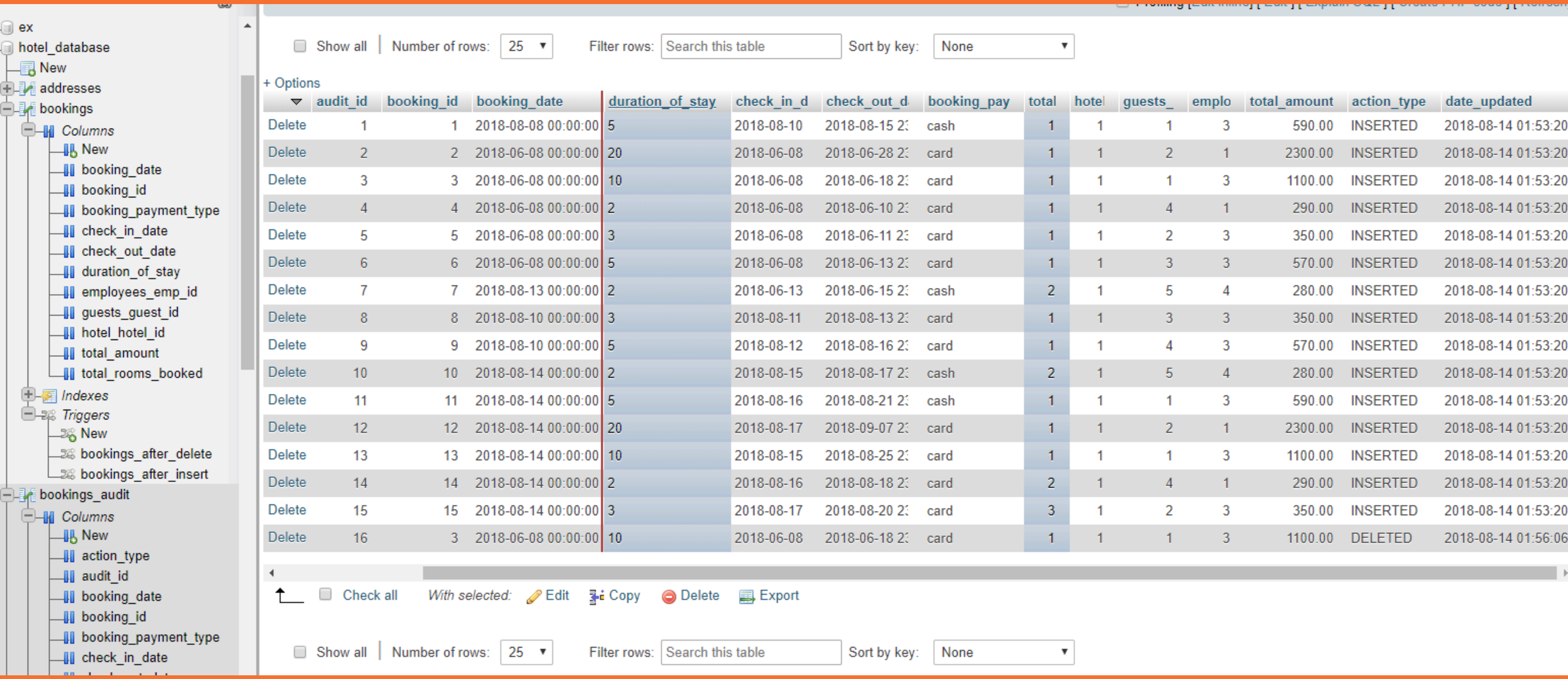
To execute the required tasks and fetch the data from one or more tables.

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