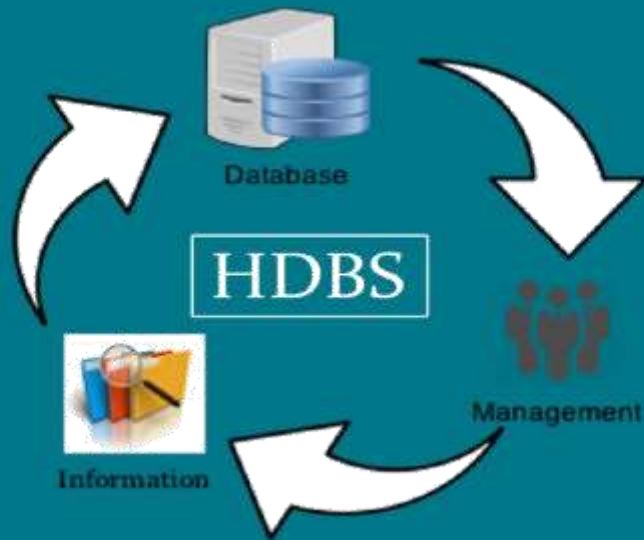


# HOTEL DATABASE MANAGEMENT SYSTEM



PREPARED BY: ANURAN MONDAL

BITS ID: 2022mt13022

DOCUMENTATION BY: ANURAN MONDAL

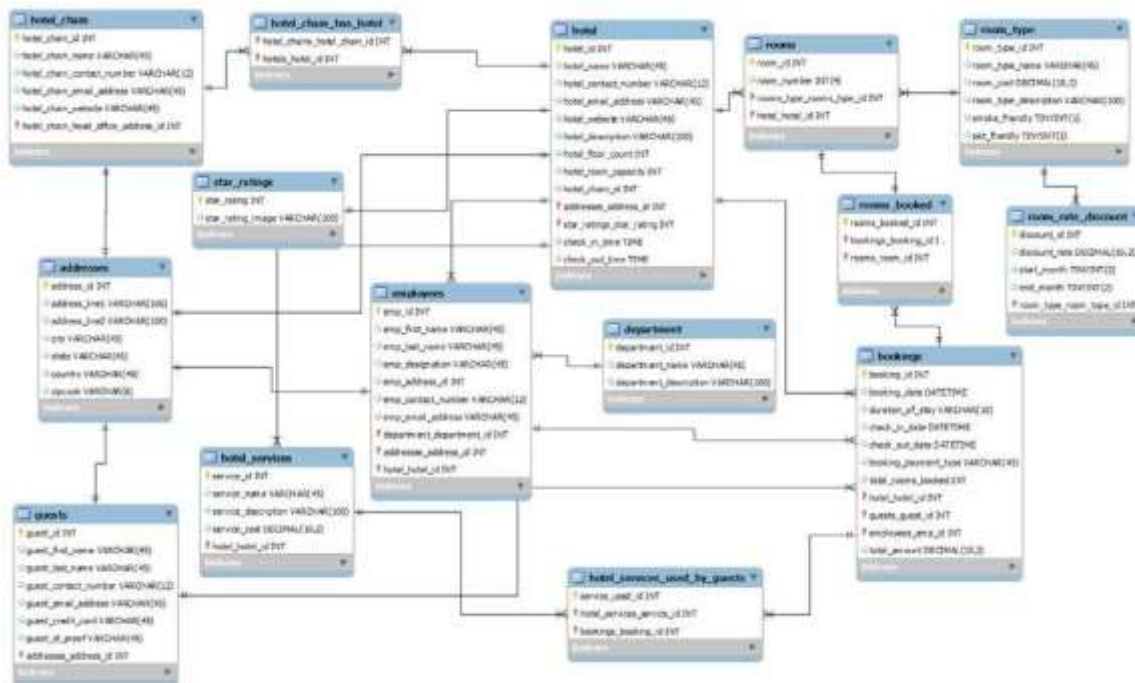
## INTRODUCTION: Overview of project & analyze the requirements

The main objective of this project is to create a database management system for a hotel. The hotel can have multiple chains, which can further have multiple hotels. Therefore, we need an organized management system, which can easily manage all the operations and data of the hotel chains and hotels respectively. We will be managing the below areas of the hotel database management system.

- The hotel chains, their details.
- The hotels in each chain and their details and other information like the rooms and their description and discounts, etc.
- Information about employees and departments they work in.
- Information about guests.
- Managing bookings and other services used by the guests.

## DATABASE DESIGNING: Starting with drawing the ERR diagram

We started with designing the structure of the database. We drew the ERR diagram on a paper, noting down all the tables required. We designed each table with columns and attributes respectively and trying to make an idea about the relationships between tables. We tried to keep the tables in a form where we can reduce the data redundancy and tried to make it accessible in an easy and simple way. The ERR diagram created is as below:



## DEFINE TABLES: Design the tables, columns and relationships

We started with designing the structure of the database. We drew the ERR diagram on a paper, noting down all the tables required. We designed each table with columns and attributes respectively. I have designed the below tables:



**bookings** table contains data about the booking made for rooms. The primary key for this table is booking\_id.

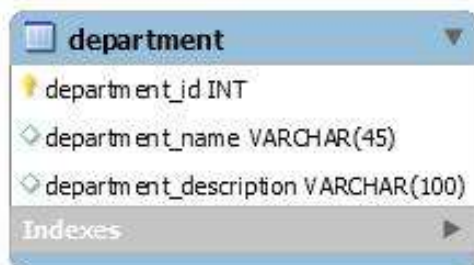
The table has the following foreign keys:

hotel\_hotel\_id which has a many-to-one relationship with the hotel table.

guests\_guest\_id which has a many-to-one relationship with the guests table.

employees\_emp\_id which has a many-to-one relationship with the employees 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.



**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)
Indexes	

**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
Indexes	

rooms_booked	
rooms_booked_id	INT
bookings_booking_id	INT
rooms_room_id	INT
Indexes	

**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.

## **NORMALIZATION: Tables should be normalized till 3<sup>rd</sup> norm**

Database normalization is the process of restructuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. Generally, if a database is normalized until third normal form then it is considered to be normalized. We tried to normalize the database until third normal form.

## **WRITING QUERIES AND TRIGGERS AND VIEWS:**

Writing down the queries was the easiest part for us. As we put a lot of effort in designing the database and creating relationships in a manner that, it will be easier to fetch the data from two or more tables. We wrote the queries as per the requirements and check in twice with valid as well as invalid data. We created two views and two triggers as below:

### **Views –**

GuestsDetails – This view shows all the information about the guests.

EmployeeDetails – This view shows all the information about the employees and their respective departments.

### **Triggers –**

BookingAudit\_OnInsert – When a new booking is generated, this trigger will create a booking audit table and insert the data into it.

BookingAudit\_OnDelete – When a booking is deleted, this trigger will be called and a row will be inserted on the Booking Audit table with the information of the booking that is being deleted.

## **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.

## **SUMMARY:**

This was an attempt to create a database management system for hotel where a DBA can easily manage the hotels, rooms, bookings, guests, employees, departments, services, etc. and other things as well, easily and quickly. Overall, it is huge area and we tried to cover few of the parts of it. Thank you!

DOCUMENTATION BY: **ANURAN MONDAL**