Hotel Management System

Contributors

Jakasaniya Jainil

Subject: -Database Management System

Date: - 03/11/2020

Introduction to Hotel Management System

One of the most important sectors for the economic growth of the country is the Tourism sector, where no doubt the good facilities, good behavior and a good management plays a major role for their attraction. Hotels, Lodges, Resorts etc., being the important for their management during their stay, should have the proper manpower and equipment to provide a better service.

"Hotel" can simply, be defined as a place where a Bonafede traveler can receive food and shelter, provided he/she is in a position to pay for it and is in a fit condition to be received. So, in general terms, Hotel Management System is expertise in management, professional managers, technicians, manuals, systems, etc. on the basis of management fees and share of profits as incentive payment, leading to the prosperity and profit for the hotel.

The term Hotel Management System includes the management of each and every aspect related to the Hotel for the attraction, smoothness of handling things, and proper management. Especially the Reservations, Registrations, Accounts, Services, Payment modes, Extra activities should be handled and maintained carefully and in an effective way so as to satisfy the guest as well as to check there is no leakage of profit to the hotel. The better the service, more the attraction and more the profit not only to the Hotel but also indirectly to the nation.

Problems With Manual Hotel Management System

It is important to keep pace with time with the increasing competition in the market and to stand on the present environment of the modern world.

The various drawbacks of manually handling the operations in the hotel are: -

- *Though paper work acts as the basis for any activities by now-a-days this is considered as slower way of performing compared to the computerized system.*
- *Searching records of individual guest takes time.*
- *Record of guest and payment record can be inconsistent.*

- *the system is unreliable and inaccurate.*
- *Uneconomical due to the large number of manpower, stationeries, and time investment.*
- *to find the info and the current balance of the guest manually is burdensome.*
- *Manual receipt making is not efficient when the party is large.*
- *Retrieving information like reports and queries is very time consuming and almost impossible practicably if time has to be considered.*

Requirement of Hotel Management System

After the analysis on the problems associated with manual system of the hotel management, it is understood that the proper system for maintaining the information of the guest and retrieving the information, personal accounts, payment bill and the stock management are required. So, in order to overcome those problems, a system was needed to develop which would help to fulfill the following requirements of the hotel management system.

- *Storing the information of the guest in the computerized database.*
- *Generating receipt and other balance records*
- *Accessing information of the guest*
- *Updating the guest account according to the service used by him*
- *Making queries to know about the guests checking-in, checking-out, room occupancy, services available*
- *Generating bills at the time of check-out*
- *Knowing the foreign currency exchange rate for converting the currencies*
- *Managing the stocks at the Stock Department*

Introduction to Database Management System (DBMS)

A database system is essentially nothing more than a computerized record-keeping system. It is repository for collection of computerized data files. The user of system will be given facilities to perform a variety of operation data record such as:

- *Inserting new data into existing files*
- *Retrieving data from existing files*
- *Updating data in the existing files*
- *Deleting data from the existing files*

The collection of inter-related data referred to as the database contains information about one particular enterprise. The primary goal of DBMS is to provide in environment that is both convenient and efficient to use in retrieving and storing database information. The management of data involves both the definition of structures for the storage of information and the provision of mechanisms for the manipulation of information. It also provides safety of information stored, despite system crashes or attempt at unauthorized access.

The advantage of computerized database is:

- *It avoids the replication of the same data in many places in full based systems*
- *Consistency of data is avoided*
- *Fast accessing of data*
- *Multiple users access the data at the same time*
- *It provides security of data by giving privileges to access the database to the user according to their record*

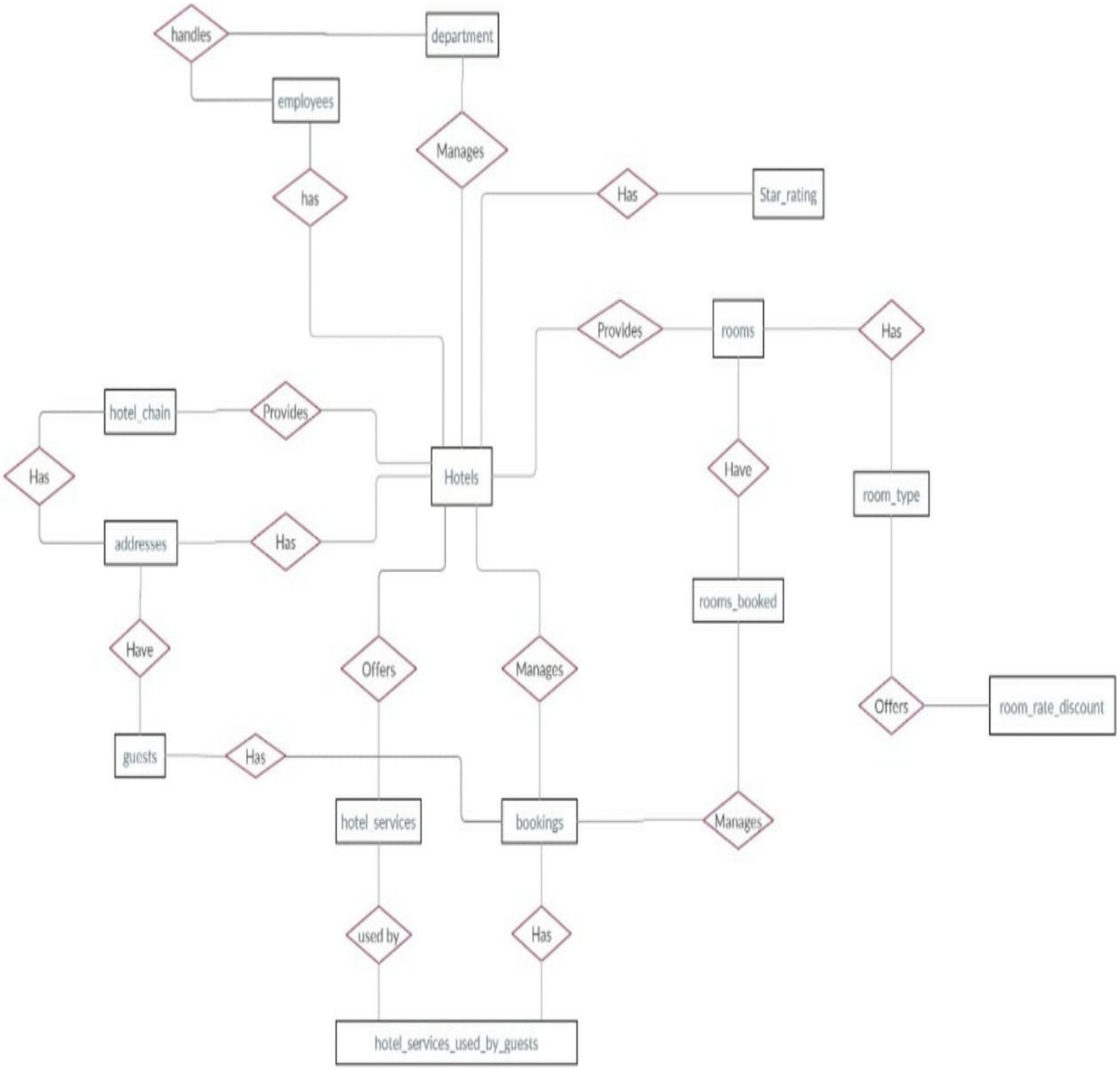
Overview

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.*

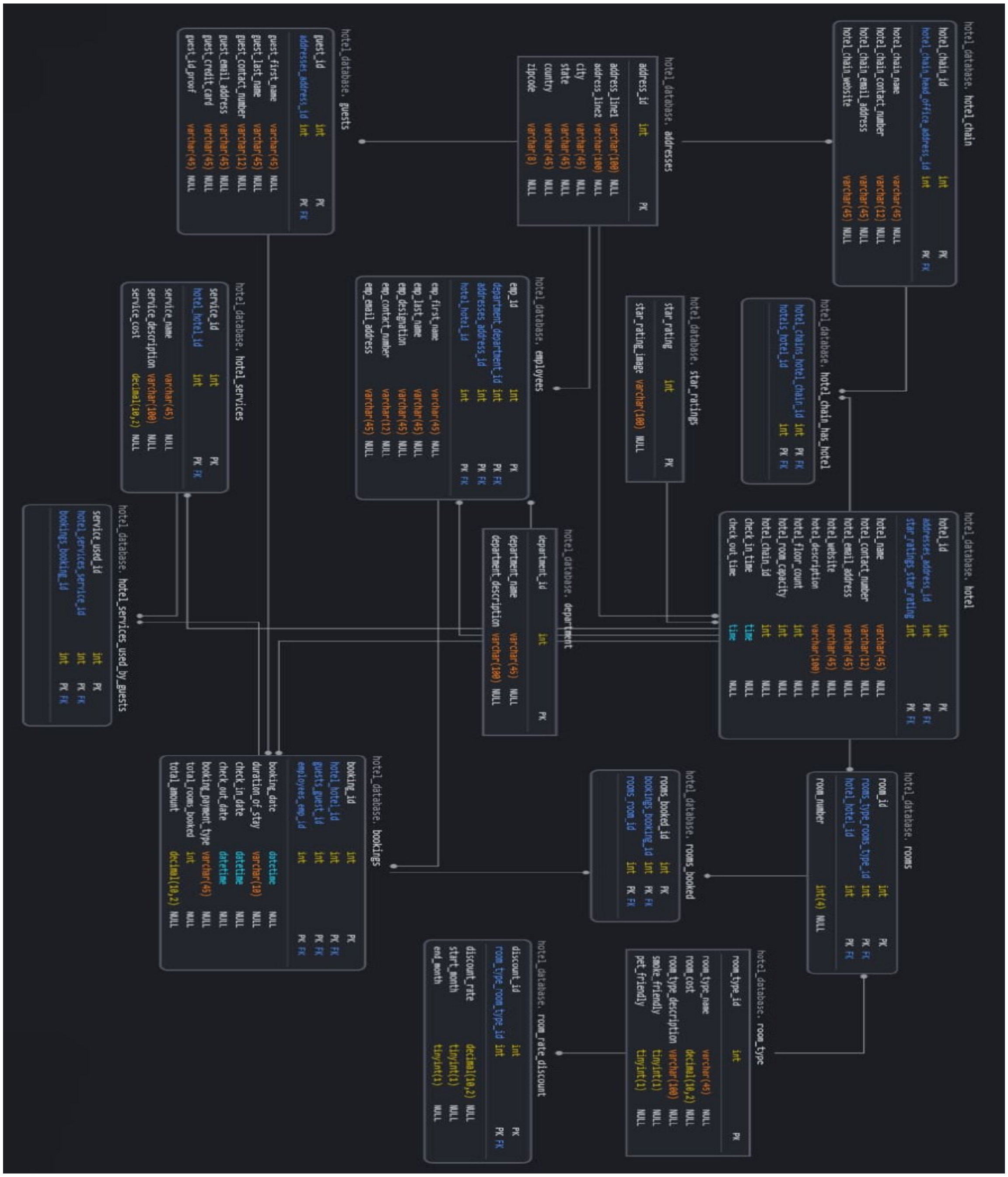
ER Diagram

The entity-relationship (ER) modeling i.e., based on the perception of a real world that consists of a set of basic objects called entities and of relationships among these objects the objects have their other attributes and once object is differentiated from the other by their attributes.



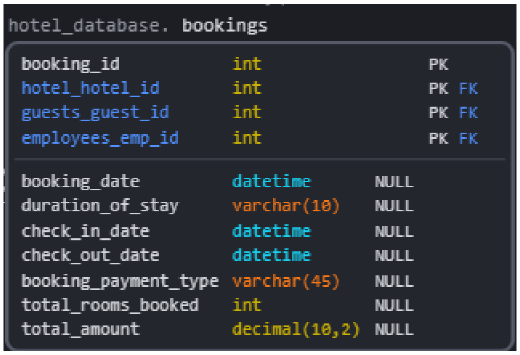
Schema

We started with designing the structure of the database. We drew the Schema 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 Schema created is as below:


hotel_database. hotel_chain
hotel_chain_id int PK
hotel_chain_head_office_address_id int PK FK
hotel_chain_name varchar(45) NULL
hotel_chain_contact_number varchar(12) NULL
hotel_chain_email_address varchar(45) NULL
hotel_chain_website varchar(45) NULL
hotel_database. hotel
hotel_id int PK
addresses_address_id int PK FK
star_ratings_star_rating int PK FK
hotel_name varchar(45) NULL
hotel_contact_number varchar(12) NULL
hotel_email_address varchar(45) NULL
hotel_website varchar(45) NULL
hotel_description varchar(100) NULL
hotel_floor_count int NULL
hotel_room_capacity int NULL
hotel_chain_id int NULL
check_in_time time NULL
check_out_time time NULL
hotel_database. rooms
room_id int PK
rooms_type_rooms_type_id int PK FK
hotel_hotel_id int PK FK
room_number int(4) NULL
hotel_database. hotel_chain_has_hotel
hotel_chains_hotel_chain_id int PK FK
hotels_hotel_id int PK FK
hotel_database. room_type
room_type_id int PK
room_type_name varchar(45) NULL
room_cost decimal(10,2) NULL
room_type_description varchar(100) NULL
smoke_friendly tinyint(1) NULL
pet_friendly tinyint(1) NULL
hotel_database. star_ratings
star_rating int PK
star_rating_image varchar(100) NULL
hotel_database. rooms_booked
rooms_booked_id int PK
bookings_booking_id int PK FK
rooms_room_id int PK FK
hotel_database. addresses
address_id int PK
address_line1 varchar(100) NULL
address_line2 varchar(100) NULL
city varchar(45) NULL
state varchar(45) NULL
country varchar(45) NULL
zipcode varchar(8) NULL
hotel_database. department
department_id int PK
department_name varchar(45) NULL
department_description varchar(100) NULL
hotel_database. employees
emp_id int PK
department_department_id int PK FK
addresses_address_id int PK FK
hotel_hotel_id int PK FK
emp_first_name varchar(45) NULL
emp_last_name varchar(45) NULL
emp_designation varchar(45) NULL
emp_contact_number varchar(12) NULL
emp_email_address varchar(45) NULL
hotel_database. room_rate_discount
discount_id int PK
room_type_room_type_id int PK FK
discount_rate decimal(10,2) NULL
start_month tinyint(1) NULL
end_month tinyint(1) NULL
hotel_database. guests
guest_id int PK
addresses_address_id int PK FK
guest_first_name varchar(45) NULL
guest_last_name varchar(45) NULL
guest_contact_number varchar(12) NULL
guest_email_address varchar(45) NULL
guest_credit_card varchar(45) NULL
guest_id_proof varchar(45) NULL
hotel_database. bookings
booking_id int PK
hotel_hotel_id int PK FK
guests_guest_id int PK FK
employees_emp_id int PK FK
booking_date datetime NULL
duration_of_stay varchar(10) NULL
check_in_date datetime NULL
check_out_date datetime NULL
booking_payment_type varchar(45) NULL
total_rooms_booked int NULL
total_amount decimal(10,2) NULL
hotel_database. hotel_services
service_id int PK
hotel_hotel_id int PK FK
service_name varchar(45) NULL
service_description varchar(100) NULL
service_cost decimal(10,2) NULL
hotel_database. hotel_services_used_by_guests
service_used_id int PK
hotel_services_service_id int PK FK
bookings_booking_id int PK FK

Define Tables

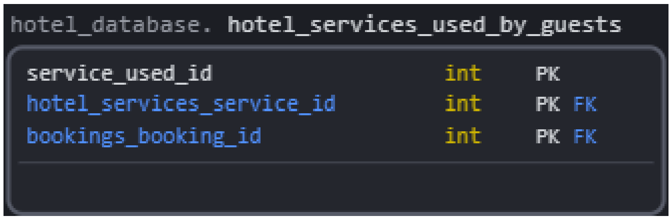
We started with designing the structure of the database. We drew the ER diagram on a paper, noting down all the tables required. We designed each table with columns and attributes respectively. We have designed the below tables:



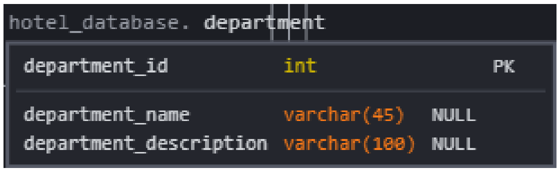
```
hotel_database. bookings
booking_id            int                   PK
hotel_hotel_id        int                   PK FK
guests_guest_id       int                   PK FK
employees_emp_id      int                   PK FK

booking_date          datetime       NULL
duration_of_stay      varchar(10)    NULL
check_in_date         datetime       NULL
check_out_date        datetime       NULL
booking_payment_type  varchar(45)    NULL
total_rooms_booked    int            NULL
total_amount          decimal(10,2)  NULL
```

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, whichrelates to hotel_services table & bookings_booking_id relates to bookings table.*



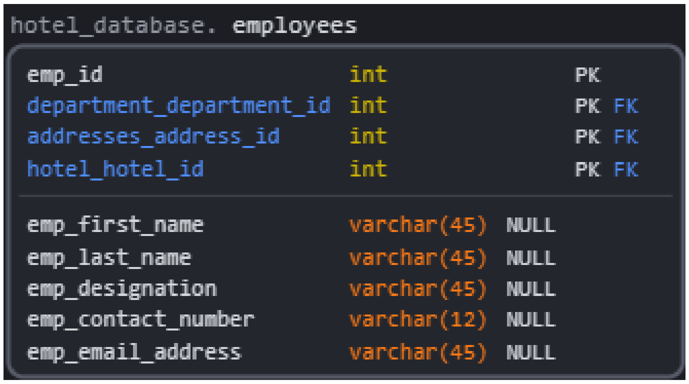
```
hotel_database. hotel_services_used_by_guests
service_used_id              int     PK
hotel_services_service_id    int     PK FK
bookings_booking_id          int     PK FK
```



```
hotel_database. department
department_id               int                 PK

department_name             varchar(45)   NULL
department_description      varchar(100)  NULL
```

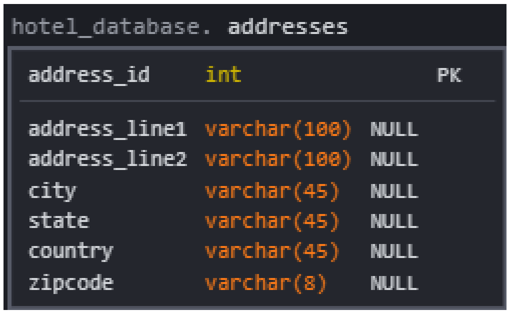
department *table contains the data about the different departments of the hotel. The primary key is department_id, which creates a one-to-many relationships with the employees' table.*

***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.*



hotel_database. employees

emp_id	int		PK
department_department_id	int		PK FK
addresses_address_id	int		PK FK
hotel_hotel_id	int		PK FK
emp_first_name	varchar(45)	NULL	
emp_last_name	varchar(45)	NULL	
emp_designation	varchar(45)	NULL	
emp_contact_number	varchar(12)	NULL	
emp_email_address	varchar(45)	NULL	

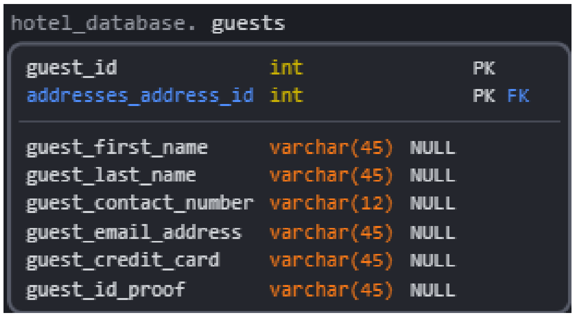


hotel_database. addresses

address_id	int		PK
address_line1	varchar(100)	NULL	
address_line2	varchar(100)	NULL	
city	varchar(45)	NULL	
state	varchar(45)	NULL	
country	varchar(45)	NULL	
zipcode	varchar(8)	NULL	

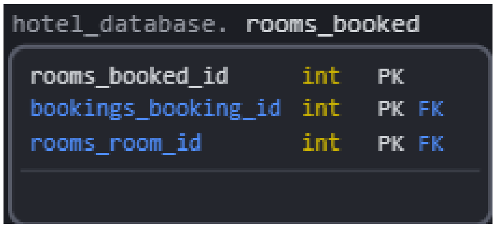
***addresses** table defines the information about the address of guests, hotels, hotel chains, employees. The primary key of the table is address_id. It maintains one-to-one relationship with tables, hotel_chain, hotel, employees and guests.*

***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.*



hotel_database. guests

guest_id	int		PK
addresses_address_id	int		PK FK
guest_first_name	varchar(45)	NULL	
guest_last_name	varchar(45)	NULL	
guest_contact_number	varchar(12)	NULL	
guest_email_address	varchar(45)	NULL	
guest_credit_card	varchar(45)	NULL	
guest_id_proof	varchar(45)	NULL	



hotel_database. rooms_booked

rooms_booked_id	int	PK
bookings_booking_id	int	PK FK
rooms_room_id	int	PK FK

***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

Normalization is the process of organizing data in the database. This includes creating tables and establishing the relationships between those tables according to the rules designed both to protect the data and to make the database more flexible by eliminating two factors: redundancy and dependency. We tried to normalize the database until third normal form.

There are some benefits of normalization.

- *Faster sorting and index creation because tables are narrower*
- *More clustered indices are allowed because there are no tables*
- *Narrower indices per table, helping INSERT, UPDATE, and DELETE performance*
- *Fewer nulls and less redundant data, increasing database compactness*

Writing Queries And Trigger And Cursor

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 also created trigger and cursor.

Queries

-- How many distinct guests have made bookings for a particular month?

```
SELECT guest_first_name, guest_last_name,guest_contact_number
FROM guests
WHERE guest_id IN (SELECT distinct guests_guest_id FROM
bookings WHERE MONTH (check_in_date) = 8);
```

-- How many available rooms are in a particular hotel for a given date?

```
SELECT h.hotel_room_capacity AS 'Total Rooms',
SUM(total_rooms_booked) AS 'Total Rooms Booked' ,
h.hotel_room_capacity - SUM(b.total_rooms_booked)    AS
'Available Rooms'
FROM `bookings` b JOIN hotel h
ON b.hotel_hotel_id = h.hotel_id
WHERE booking_date LIKE '2018-08-14%' AND hotel_hotel_id = 1;
```

-- How many hotels are in a hotel chain?

```
SELECT count (*) AS 'Total Hotels'
FROM hotel_chain_has_hotel
WHERE hotel_chains_hotel_chain_id = 1;
```

-- How many books has a customer made in one year?

```
SELECT count (*) AS 'Total Bookings'
FROM bookings
WHERE YEAR (booking_date) = 2018 AND guests_guest_id = 1;
```

-- How many rooms are booked in a particular hotel on a given date?

```
SELECT SUM (total_rooms_booked) AS 'Total Rooms Booked'
FROM `bookings`
WHERE booking_date LIKE '2018-06-08%' AND hotel_hotel_id = 1;
```

-- List all the unique countries hotels are located in.

```
SELECT DISTINCT country
FROM addresses
WHERE address_id IN (SELECT addresses_address_id FROM hotel);
```

-- How many rooms are available in a given hotel?

```
SELECT h.hotel_room_capacity - SUM(b.total_rooms_booked) AS 'Available Rooms'
FROM `bookings` b JOIN hotel h
ON b.hotel_hotel_id = h.hotel_id
WHERE booking_date LIKE '2018-06-08%' AND hotel_hotel_id = 1;
```

-- List all the hotels that have a URL available.

```
SELECT *
FROM `hotel`
WHERE hotel_website IS NOT NULL;
```

Cursor

Employee_Detail *– To display employee first name, last name and designation.*

```
DECLARE

s_fname VARCHAR2(45);

s_lname VARCHAR2(45);

s_designation VARCHAR2(45);

CURSOR Employee_Detail IS

SELECT emp_first_name,emp_last_name,emp_designation FROM employees;

BEGIN
```

```
OPEN Employee_Detail;

Loop

FETCH Employee_Detail INTO s_fname,s_lname,s_designation;

DBMS_OUTPUT.PUT_LINE (s_fname||'  '||s_lname||'
'||s_designation);

EXIT WHEN Employee_Detail %NOTFOUND;

END Loop;

CLOSE Employee_Detail;

END;
```

Trigger

***BookingAudit_OnInsert** – When a new booking is generated, this trigger will be fired.*

```
CREATE TABLE Bookings_audit( employees_emp_id NUMBER(10));

SET SERVEROUTPUT ON;

CREATE OR REPLACE TRIGGER B_audit

BEFORE INSERT ON Bookings_audit

FOR EACH ROW

ENABLE

DECLARE

   e_id NUMBER (10);

BEGIN

 SELECT employees_emp_id INTO e_id FROM bookings;

 DBMS_OUTPUT.PUT_LINE ('Inserted a new row by employee: '||e_id);

END;
```

Challenge 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!