**Normalization Project on Online Hotel Management**

**Topic-Online Hotel Booking**

**Submitted to:- Prof. Ashok Harnal**

**TERM - III**

**Submitted By:- Group 4(Section G)**

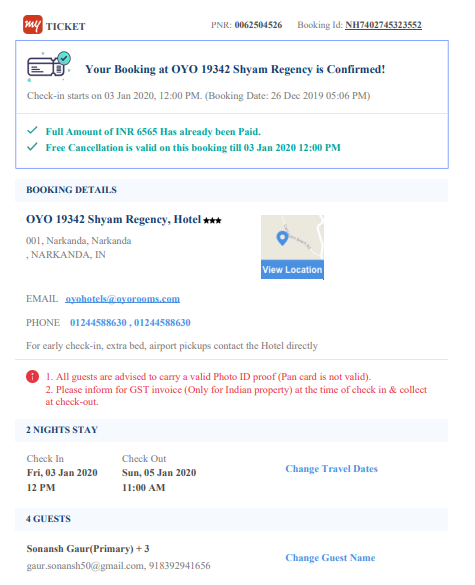
**Sonansh Gaur -025025**

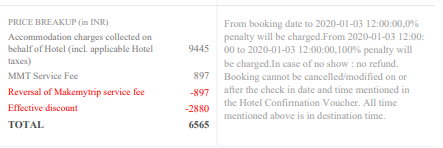
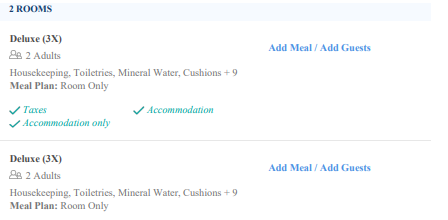
**Gitika Jain- 025044**

**Mebal Chaudhary- 025050**

**CASE:-**

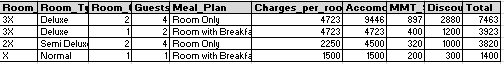
Below is the online generated **Hotel booking ticket** that we had taken in our project.





Let us capture the above information in a tabular form.





## 

## 

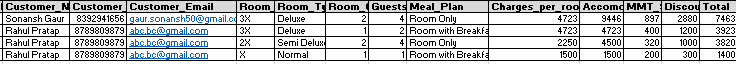
## 

## 

## **First Normal Form: No nested tables or Groups of Elements**

In no cell we should be having multiple elements or a table.





### 

### **Define primary key**

Before we proceed further, we need to specify our primary key. Each row has an Booking\_id for every Room\_id. Therefore, Primary Key: [*Booking\_id,Room\_id*]. It is a composite primary key.

The underlying structure of the orders table can be represented as in the figure below. We identify the columns that make up the primary key with the PK notation.

**Booking**

**Booking\_ID(PK)**

Booking\_Date

Booking\_Time

Hotel\_ID

Hotel\_Name

Hotel\_Address

Hotel\_Email

Hotel\_Contact

CheckIN\_Date

CheckIN\_Time

CheckOUT\_Date

CheckOUT\_Time

Customer\_PNR

Customer\_Name

Customer\_contact

Customer\_Email

**Room\_ID(PK)**

Room\_Type

Room\_Qty

Guests

Meal\_Plan

Charges\_per\_room

Accomodation\_charges

MMT\_Service\_FEE

Discount

Total

Our database schema now satisfies the two requirements of First Normal Form: atomicity and uniqueness. Thus it fulfills the most basic criterion of a relational database.

## 

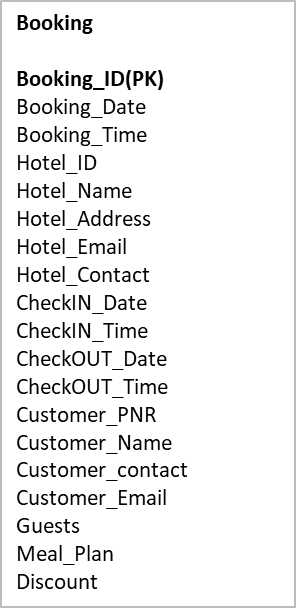
## **Second Normal Form: No Dependencies on a part of a Composite Primary Key**

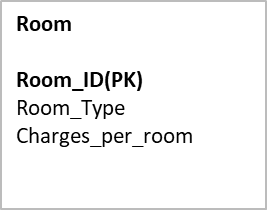
Next, we test each field for partial dependencies on the composite PK. This means that for a table that has a composite primary key, each column in the table that is not part of the primary key must depend upon the entire composite PK for its existence. If any column only depends upon one part of the composite key, then we say that the entire table has failed Second Normal Form and we must create another table to rectify the failure.

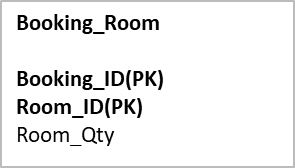
**Booking\_Date,Booking\_Time,CheckIN\_Date,CheckOUT\_Date,Guestsand Meal Plan depends on Booking\_ID**

**Room\_Type and Charges\_per\_room are dependent on Room\_ID**

All the columns that depend on Room\_id - whether in whole or in part - follow it into the new table. We call this new table Booking\_Room.





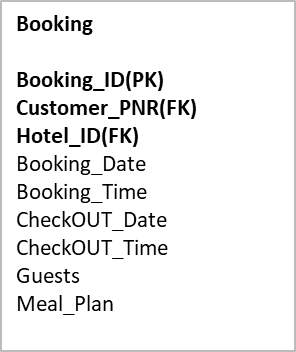


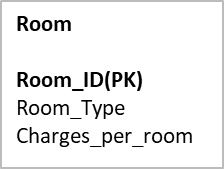
## 

## **Third Normal Form: No Dependencies on Non-Key Attributes**

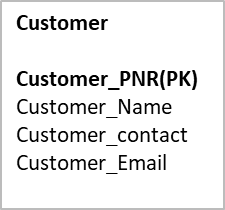
Fields like Hotel\_Name,Hotel\_Address,Hotel\_Email,Hotel\_Contact,CheckOUT\_Time,CheckIN\_Time rely on Hotel\_ID which is not a part of Primary Key

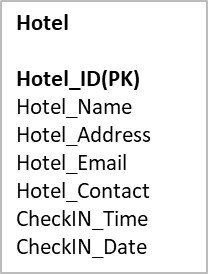
Fields like Customer\_Name,Customer\_Email,Customer\_Contact,Discount rely on Customer\_PNR which is also not a part of primary key





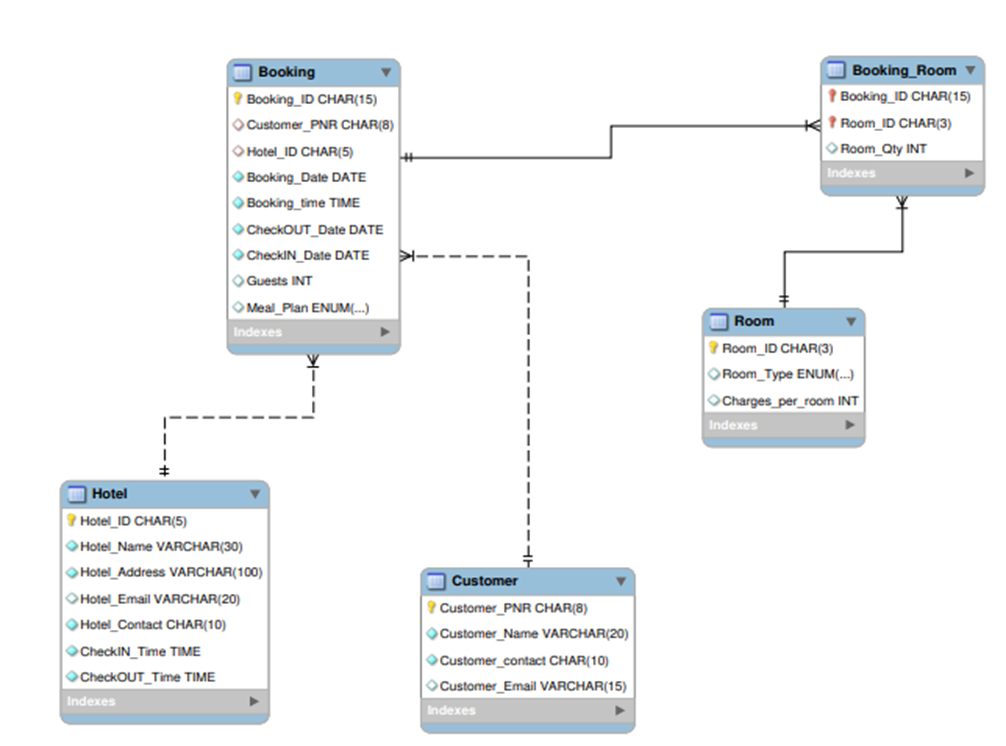






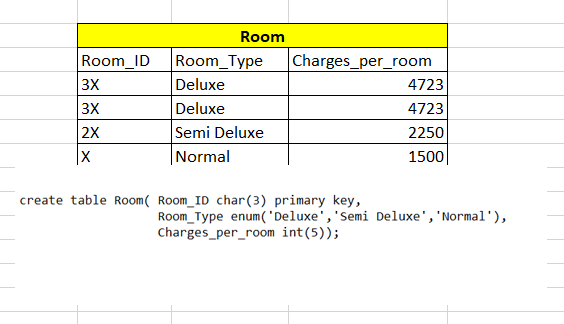
**Entity Relationship Diagrams**

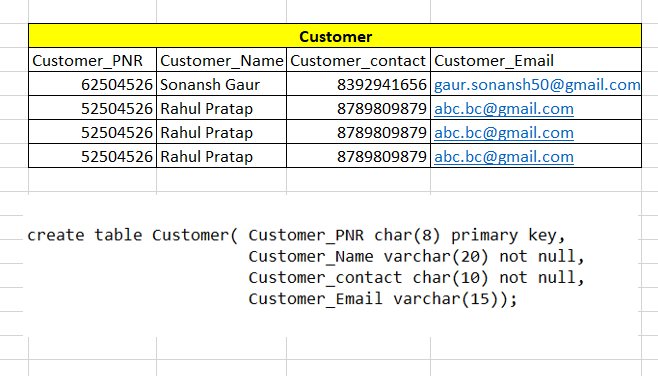
Entity Relationship Diagram (ERD) or Entity Relationship Diagram (ERD) is a diagram that shows the relationship between entity sets contained in a database. In other words, ER diagrams aid in the explanation of database logical structure. Entities, attributes, and relationships are the three main notions that ER diagrams are built on.

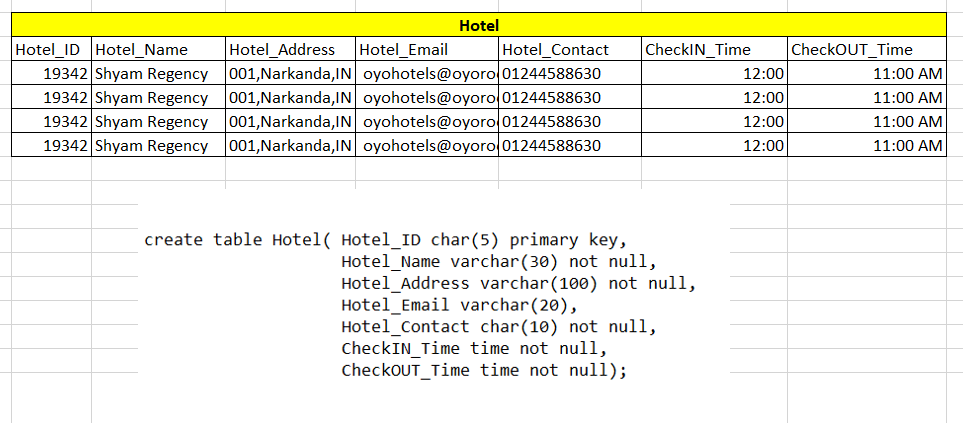


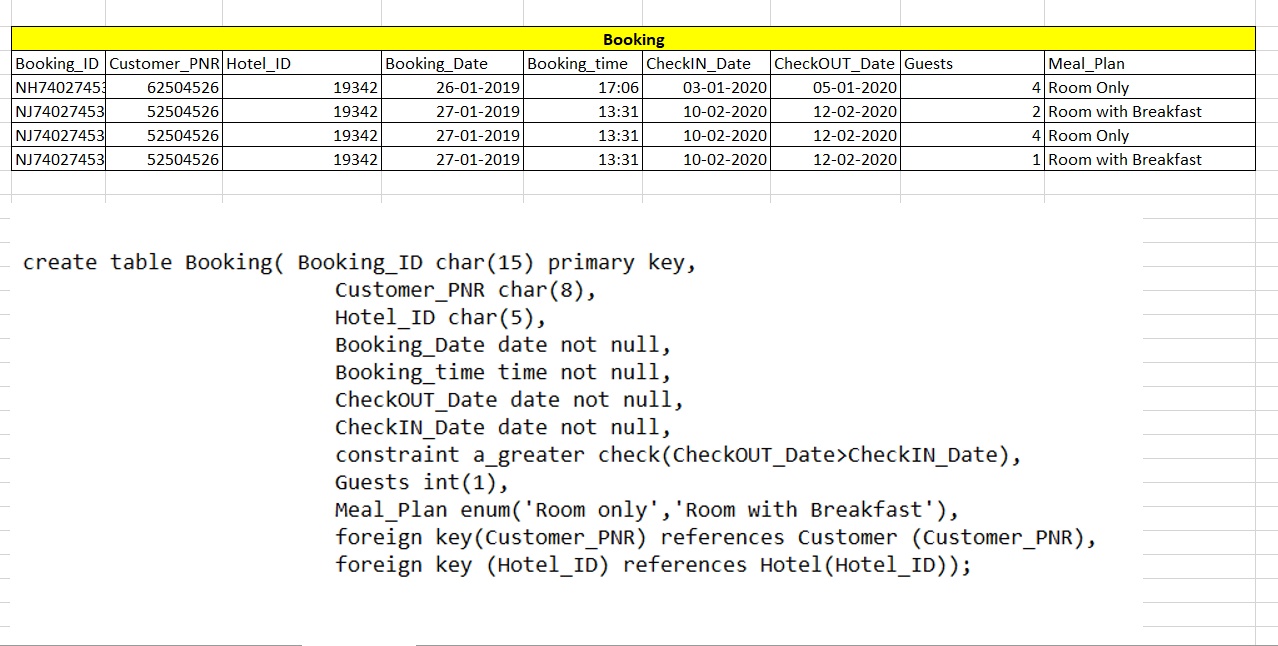
**MYSQL Code**

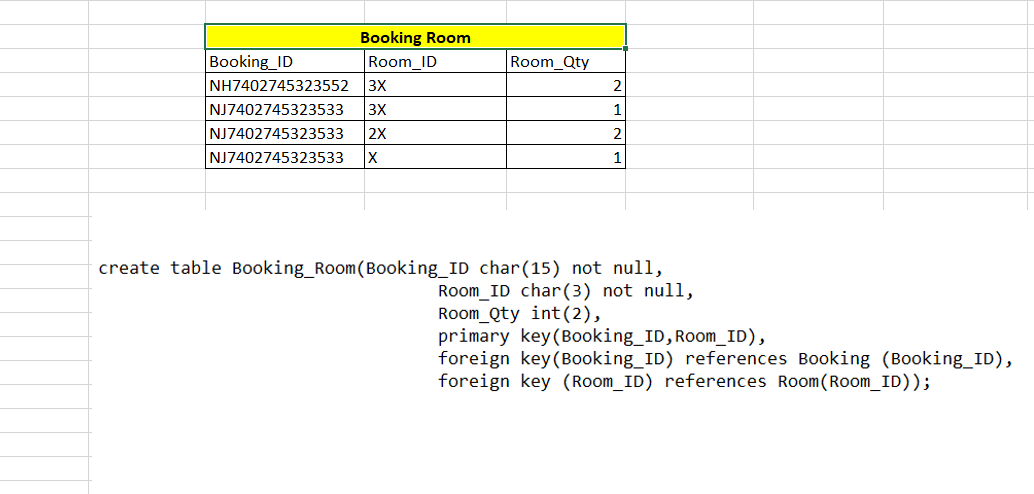
The codes that have been used in the formulation of tables have been shown below.









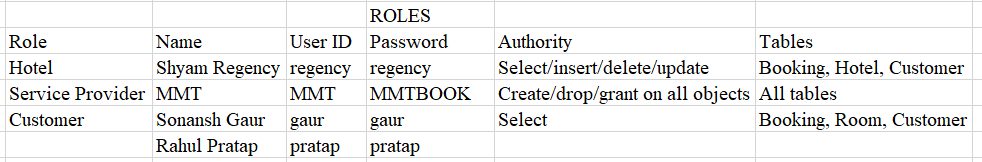


**GRANT AUTHORISATION:**

The database manager obtains information about the authenticated user through authorization. Part of that information is determining which database actions and data items the user is allowed to access. Providing them with the permission for the same is called granting authorization.

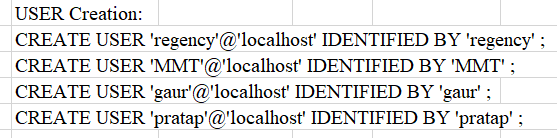
ROLES:

This tells what part which user plays in the database management system, what kind of authority in the database management system they would be provided and in which tables they can use those authorities.



USER CREATION:

This creates a separate id for every user associated with the database management system.



CODES:

These are the codes that would be used in proving these users an access in the database management system and through these codes only they would get the permission to access the database management system.

