***CSC375: Database Management Systems***

***Database for the Hotel The Four***

***Task 1***

*Yara Chahine*

*Nahla Baalbaki*

*Mira Mansour*

*Sabine Hamdoun*

***05/09/2021***

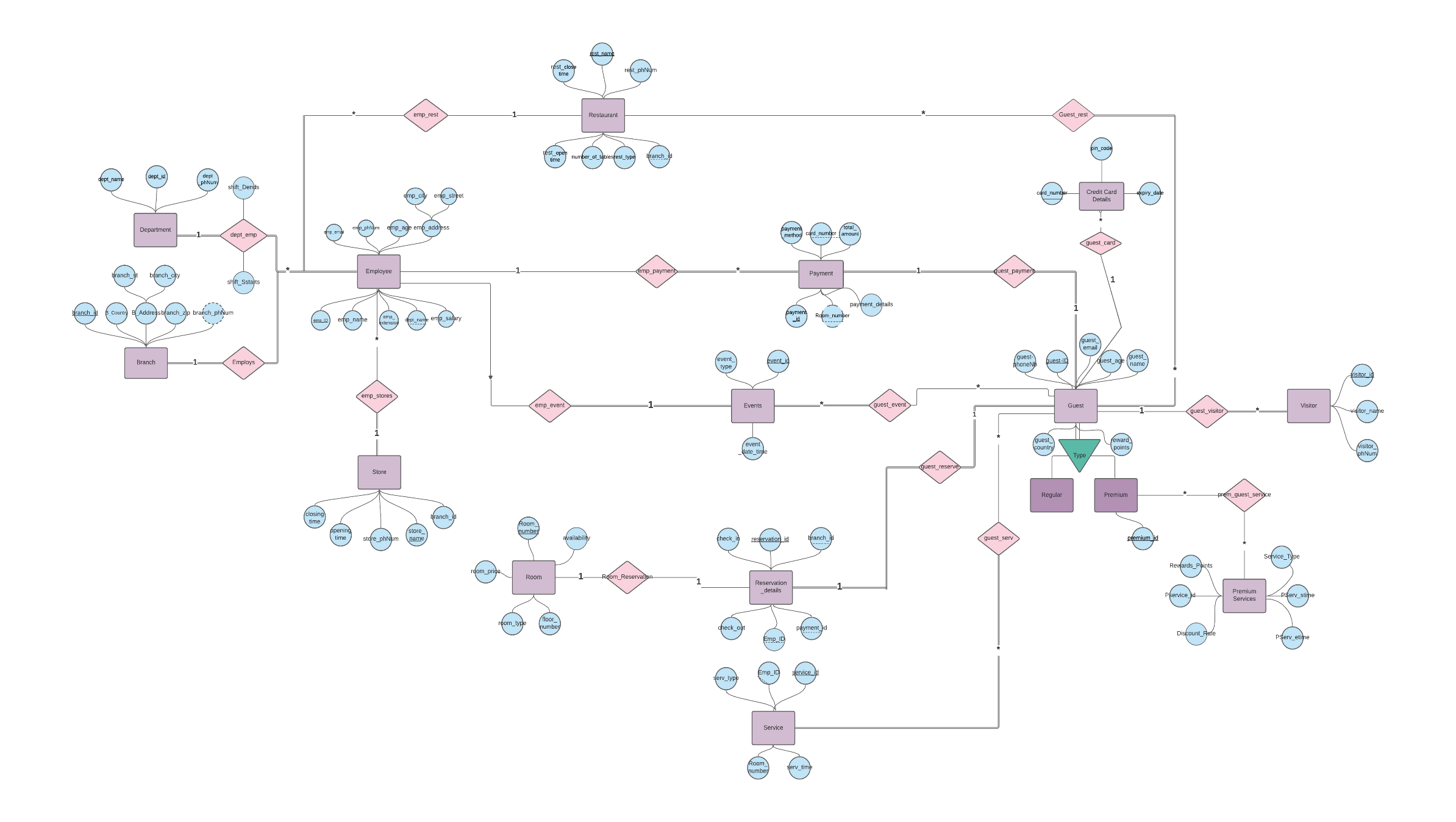
**Application**:

Our application’s main goal is to increase guest loyalty and satisfaction. By having this application, we will be able to show the guest the diverse facilities and activities found at our chain of hotels “The Four”. The facilities and activities at the hotel target people from different ages knowing that we aim to please our guests. Moreover, the staff at our hotels are well trained and professional at dealing with guests. They try their best to make sure our guests are content and pleased with their stay. In order to do so, we offer interesting daily offers alongside reward points whenever a guest books a facility through the application. Once the guest gains a certain number of reward points, they will be able to access premium features and services in the hotel. Additional features available on the application such as the guest at the hotel will be able to order room service 24/7, check out of the room, reserve their own parking spot, or even book car service.

As for the database we are currently creating, it will be accessible to 3 main departments: Front Office, Human Resources, and Management at the hotel with certain restrictions based on their job. For example, only HR staff have access to employee records, and are able to hire or fire an employee, or give a salary raise/cut. This database will allow front office employees to update reward points, add new guests and check room availability safely without losing other important data. To get all the data required to build this database, we will be generating random virtual data within the context.

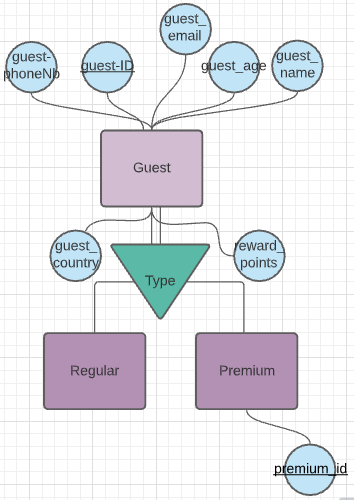
Moreover, from the guest’s perspective, they will be able to book a room or a service 24/7 form the application, which in its turn will be added to the hotel database. Premium guests will have access to premium services and can check how many reward points they have.

There are similar applications that allow guests to view their desired hotel and book a room such as Expedia, Booking.com and Trivago. However, the application for our Hotel has additional features such as collecting reward points, getting daily discounts on certain activities or facilities, ordering room service 24/7 using the application, as well as allowing premium guests to have access to special floors at our hotel.



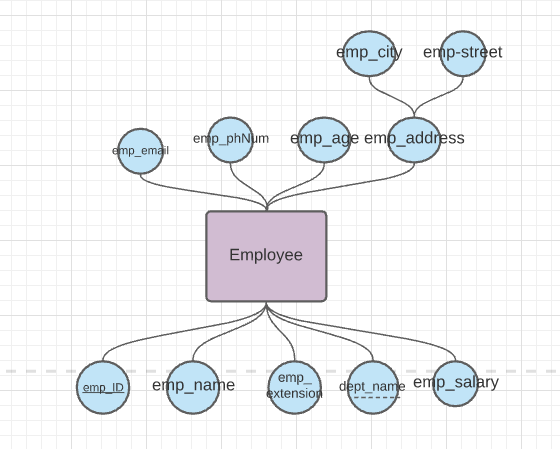
**Entities:**

**1) Guest**(Regular (guest\_ID, guest\_name, guest\_phoneNb, guest\_age, guest\_email, guest\_country, reward\_points) vs. Premium(premium\_id)

Since our application’s main goal is to increase guests’ satisfaction and loyalty, our guest is our most important entity, it includes guests’ details such as ID, name, phone number, age, email, country and reward points. The more a guest visits our hotel and uses its services, the more reward points they receive, thus upgrading them to a “Premium Guest”, having a premium ID, with access to premium services! Our guest entity is part of five relation entities: guest\_payment, guest\_reserve, guest\_visitor, guest\_event, guest\_services with one additional relation premium\_guest\_service for premium guests.

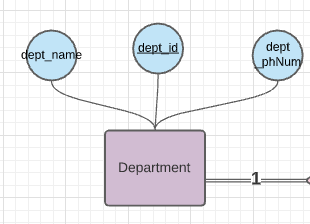
**2) Employees** (emp\_ID, emp\_name, emp\_email, emp\_phNum, emp\_age, emp\_address(emp\_city,emp\_street), emp\_extension, dept\_name, emp\_salary)

Our employees are the backbone of our hotel. They ensure that everything is going smoothly in all departments and hotel services. They assist our guests and ensure that they have a great experience at the hotel. Our employee entity consists of the employee’s unique ID, name, phone number, email, age, address, salar, phone number extension and the department to which they belong.

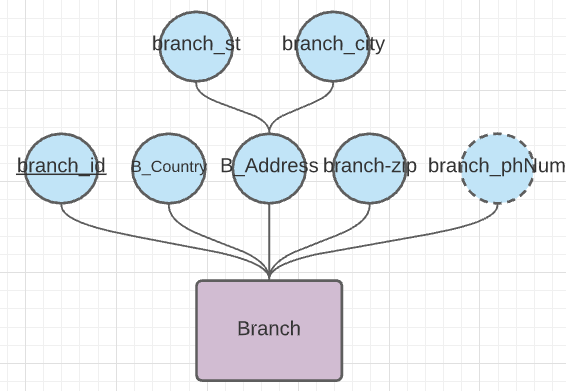


**3) Department** (dept\_id, dept\_name, dept\_phNum)

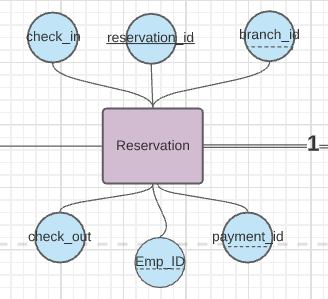
There are many departments at the hotel and together they ensure a variety of services in our hotel to please our guests. These departments vary between management, reception, kitchen, room service, valet parking, event planning and much more. Each entity is uniquely identified with a unique department ID, name of the department, its location and the phone number allocated to it.



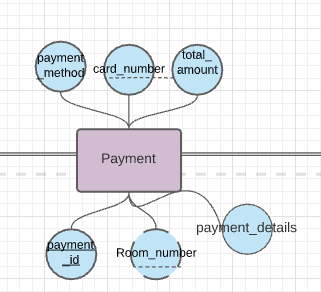
**4) Branch** (branch\_id, B\_Country,B\_Address(B\_City,B\_Street),branch\_zip, branch\_phNum)

Our hotel has many branches across many countries. The branch entity is characterized by a unique ID which is a primary key, address, multivalued phone numbers,, country and zip number. This entity specifies at which branch employees work and at which branch are guests staying.

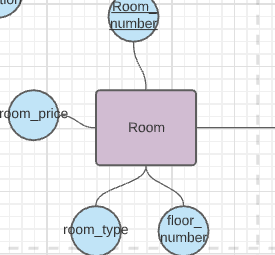
**5) Reservation\_details** (reservation\_id, branch\_id, check\_in, check\_out, payment\_id, emp\_ID)

When a guest books a room, the reservation details will be stored in the Reservation entity. This entity includes foreign keys such as room\_num, branch\_id to know where the guest is exactly staying, payment\_id to link it with the payment, guest\_id to know which guest made what reservation, and emp\_id to know which employee received the reservation details. This entity also includes the reservation\_ID that uniquely distinguishes each instance of Reservation, and finally check\_in and check\_out times. With all these details included, we assure each guest that their rooms will be available to them once they arrive.

6) **Payment** (payment\_id, payment\_method,Room\_number, total\_amount, card\_number, payment\_details)

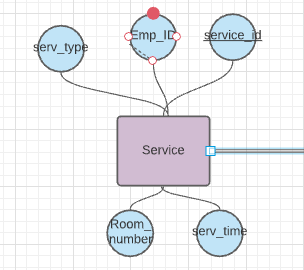
At our hotel, we have the most flexible payment method whereby we ensure that the guest feels comfortable when dealing with us. We accept the payments online or cash once the guest arrives. Moreover, we deal with authenticated monetary services thus the guest’s information is secured. 

7) **Room** (Room\_number, room\_type, floor\_number, room\_price, availability)

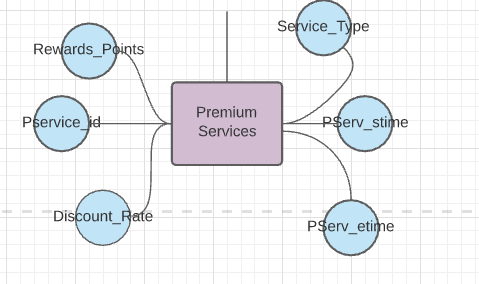
At our hotel we have the most luxurious and spacious rooms. We work on delivering the best form of hospitality to our guests so that they get to experience the highest level of comfort and most comfortable hotel experience. Each floor at our hotel has a certain theme to it. The rooms are designed by the best interior designers in the world in order to ensure that our guests are having a luxurious time at our hotel.

8) **Service** (service\_id, emp\_ID, serv\_type, Room\_number, serv\_time)

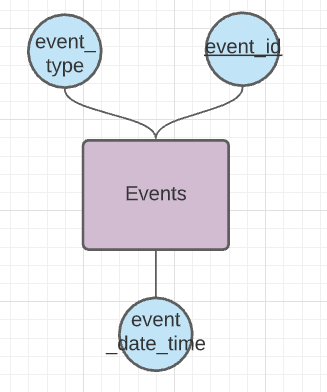
Services are available at any time of the day, the guest will be able to receive several services within a very short period of time. We also deliver various selections of services for the guest to benefit from multiple types of cuisines, laundry, valet parking, transportation, spa, swimming pool, gym and room services.



9) **Premium Services** (Pservice\_ID, Discount\_Rate, Rewards\_Points, Service\_Type, PServ\_stime, PServ\_etime)

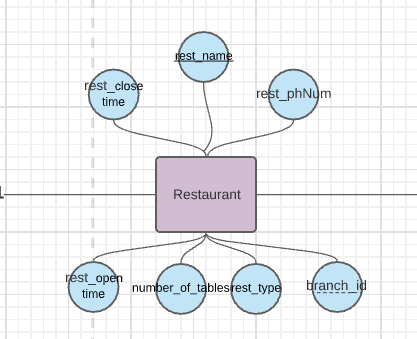
We offer Premium services, identified by its Premium\_ID, at our hotel to guests who maintain 10000 Rewards\_Points. Moreover, we keep track of the Rewards\_Points that can promote a guest from regular to Premium. The Premium services allow guest’s access to the VIP lounge, luxury breakfast, garden terrace, and authorize them to have Discount\_Rate on their booking fees. Finally, we hold data regarding the Service\_Type, PServ\_stime and PServ\_etime to provide them with the best possible guest service.

10) **Events** (event\_id, event\_date\_time, event\_type).

To keep our Hotel lively and exciting, we rotate different events across the entire year. Each event is identified by its event\_id and described by the description (such as New Year’s Eve, Easter ...). We also maintain the location, date and time of the event that are also classified by the event\_type that could be private or public.

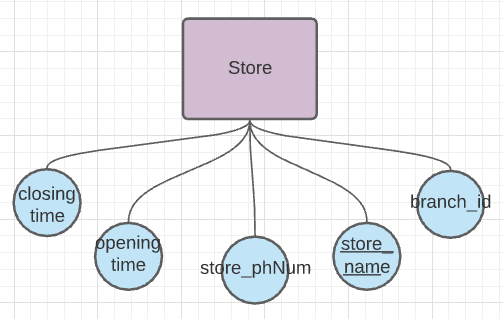
11) **Restaurant** (rest\_name, branch\_id, rest\_open\_time, rest\_close\_time, number\_of\_tables, rest\_phNum, rest\_type)

It is known that it is not a vacation without good food! That’s why at our hotel we have a variety of restaurants from Mediterranean to Japanese as well as Italian. This variety ensures that everyone is satisfied with the food as well as the ambience. Moreover, our restaurants have the best chefs from around the world that are well known for their culinary skills.

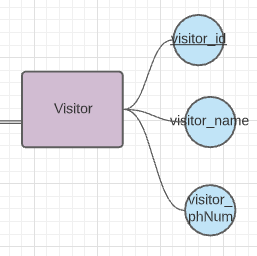


12) **Store** (store\_name, store\_phNum, opening\_time, closing\_time, branch\_id)

At our hotel, we would like for our guest to shop for a souvenir in order to have a piece of our hotel as a memory.Therefore, we have a variety of stores and antique shops that would spark our guests’s interest. Moreover, we have a pharmacy that is fully equipped with all needed medical drugs in case a guest needs an urgent medicine.

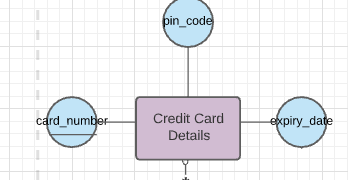


**13) Visitor:** (visitor\_id, room\_num, visitor\_name)

At our hotel, visitors are welcomed to come and spend their day or night in order to enjoy the activities we have. Moreover, visitors that come to visit our guests can spend their day with them as well.

**14) Credit\_Card:** (card\_number, pin\_code,expiry\_date)

At our hotel, knowing that reservations are online, payments take place through debit cards or credit cards. Of course, it goes without saying that our hotel deals with such matters confidentially.



**Queries:**

1. Select emp\_id as ID , emp\_name as Name, dept\_name as Department from employees where dept\_name= 'Management' or dept\_name= 'Reception';
2. Select guest\_ID as ID, guest\_name as Name, guest\_age as Age from guest where guest\_age > 30;
3. Select guest\_name as Name, reward\_points as 'Reward Points' from guest where reward\_points>600;
4. Select visitor\_id ,visitor\_name from Visitor where Room\_number>120;
5. Select Emp\_name as Name, emp\_salary as Salary from employees where emp\_salary > all (select emp\_salary from employees where dept\_name = 'Management');

**Relations :**

1. **dept\_emp:** This relation between department and Employee shows that the department has more than 1 employee however the employee has only one department where he belongs to. Moreover, this relation has two attributes shift\_Dstart and shift\_Dend which are related to the working hours of the employee.

2. **Employs** : This relation is between Branch and Employee. It is one to many since a branch can employ numerous employees but employees can only work at one branch. This relation ensures total participation since every employee works at a branch and every branch has employees.

3. **emp\_stores**: This relation between Employee and Stores shows that the store has more than 1 employee but the employee works only in one store.

4. **emp\_rest**: This relation between Employee and Restaurant shows that the Restaurant has more than 1 employee but the employee works only in one Restaurant. It has total participation from the restaurant’s side since every restaurant must have an employee working in it.

5. **emp\_event**: This relation is between Employee and Events, since it is necessary to know which employees are supervising which event. It is many to one since many employees can be at one event but no employee can be at two events at the same time. This relation has total participation from the events’ side since every event has to have employees overseeing it.

6. **emp\_payment:** This relation is between Employee and payment, it shows that an employee can do more than one payment but the payment has only one employee that certifies it .

7. **Guest\_rest:** This relation is between Guest and Restaurant which goes to show that restaurants can have many guests as well as guests have the choice to visit many restaurants.

8. **Guest\_payment** : This relation is necessary to link every guest with their payment details. It is one to one since every guest can have one single payment and every payment is done by one guest. It has total participation since every guest has to pay and every payment is done by a guest.

9. **Guest\_visitor** : This relation links guests and their visitors. It is one to many because a guest can have many visitors but a visitor can be invited by only one guest. This relation has total participation from the visitors’ side but not the guest’s since not all guests have to have visitors over.

10. **Prem\_guest\_service**: This relation is between premium customers and premium services. It is many to many since every guest can enjoy many services and each service is enjoyed by many guests. It does not ensure total participation since not all premium guests have to enjoy a given service.

11. **guest\_serv:** This relation is between guest and service and it shows that the services can be provided to many guests, and a service used and enjoyed by many guests..

12. **Room\_Reservation:** This relation is between Reservation and Room. It is one to one since every room can be reserved once and every reservation can be reserved for one room.

13**. Guest\_reserve**: This relation is between guests and their reservation, to ensure that all reservation details are linked to their respective guest. This is a one to one relation since every guest can only make one reservation at a given time. It ensures total participation from both sides since every guest has to make a reservation, and every reservation is made bby some guest.

14. **Guest\_event** :This relation is between guests and events. It is many to many since every guest can attend many events, and every event is attended by many guests. It ensures total participation from the event’s side since every event must be attended by some guest, unless it is a really, really bad one.

**1) Guest**(Regular (guest\_ID, guest\_name, guest\_phoneNum, guest\_age, guest\_email, guest\_country, reward points) vs.Premium(premium\_id)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| guest\_ID | guest\_name | guest\_  phNum | guest\_age | guest\_email | guest\_country | Reward  \_points |
| 2555 | Nahla Baalbaki | 7230145 | 20 | nahla.baalbaki@hotmail.com | Lebanon | 10 |
| 2556 | Ruba Baalbaki | 7894562 | 22 | ruba.baalbaki@hotmail.com | Lebanon | 22 |

**Functional Dependencies :**

* guest\_ID , guest\_phNum, guest\_email → guest\_name, guest\_age, guest\_country
* guest\_ID, guest\_phNum, guest\_email → guest\_phNum, guest\_email, reward points

**Normalization :**

* “Guest\_id” is the primary key for this schema,while guest\_phNum and guest\_email are candidate key since no two people can have the same phone number or email, therefore the three attributes form a **superkey**.
* As for the second FD, it is trivial. Thus, this table is in **BCNF**.

We did not include guest\_name in the superkey, since this is a large hotel we can have many guests with the same name.

**2) Employees** (emp\_ID, emp\_name, emp\_email, emp\_phNum, emp\_age, emp\_city,emp\_street, emp\_extension,dept\_name, emp\_salary)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| emp\_ID | emp\_name | emp\_email | emp\_phNUM | emp\_age | emp\_city | emp \_street | emp \_extension | dept \_name | emp \_salary |
| 12351 | Sabine Hamdoun | sabine.hamdoun@hotmail.com | 7625845 | 23 | Beirut | St12 | 51 | Management | 56000 |
| 12352 | Mira Mansour | mira.mansour@hotmail.com | 7895452 | 26 | Bekaa | St23 | 52 | Human Resources | 53000 |

**Functional Dependencies :**

* emp\_ID, emp\_name ,emp\_phNum, emp\_email,emp\_extension→ emp\_age, emp\_address, dept\_name, emp-salary
* emp\_ID, emp\_name, emp\_phNum,emp\_email, emp\_extension → emp\_name , emp\_phNum, emp\_email,emp\_extension

**Normalization :**

* “Emp\_name” , “Emp\_phNum”, emp\_phNum, emp\_extension are candidate keys for this schema, thus , with emp\_ID, they form a **superkey** for this schema.
* As for the second FD, the dependency is trivial ( emp\_name, emp\_phNum, emp\_extension are a subset of emp\_ID, emp\_name, emp\_phNum, emp\_extension). Thus, this schema is in **BCNF**.

We are assuming that no two employees have the same name.

**3) Department** (dept\_id, dept\_name, dept\_phNum)

|  |  |  |
| --- | --- | --- |
| dept\_id | dept\_name | dept\_phNum |
| 5878 | Management | 07745426 |

**Functional Dependencies:**

* Dept\_id, department\_name, dept\_phNum → dept\_phNum, dept\_name

**Normalization :**

This FD satisfies both conditions of **BCNF**, it is trivial and the left part’s attributes form a **superkey**.

**4)Branch** (branch\_id, B\_Country, B\_Address(B\_Street, B\_City), branch\_zip, branch\_phNum)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| branch\_id | B\_Country | B\_Street | B\_City | branch\_zip | branch\_phNum |
| 1 | Lebanon | Beirut | Hamra Street | 85786 | 247637858 |
| 2 | Netherlands | Amsterdam | King Street | 78584 | 374738263 |

**Functional Dependencies:**

* Branch\_id, branch\_phNum, branch\_zip,b\_St → B\_Country, B\_City
* Branch\_id,branch\_phNum, branch\_zip, b\_St → branch\_phNum, branch\_zip, b\_St

**Normalization:**

* For the first “branch\_id”, “branch\_phNum”, “branch\_zip” & “b\_St” form a **superkey.**
* As for the second, the attributes on the right form a **subset** of the attributes on the left.

Thus, this table is in **BCNF**

**5) Reservation details** (reservation\_id, branch\_id, check\_in, check\_out, payment\_id, emp\_ID)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| reservation \_id | branch \_id | check \_in | check \_out | payment \_id | emp\_ID |
| 3245 | 21 | 12:00 | 12:00 | 123345 | 3333 |

**Functional Dependencies:**

* reservation\_id, payment\_id → check\_in, check\_out, branch\_id, emp\_ID, payment\_id

**Normalization**:

The attributes on the right side of this dependency constitute a **superkey** of this table, thus this table is in **BCNF**.

6) **Payment** (payment\_id, payment\_method, Room\_number, total\_amount, card\_number, payment\_details)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| payment\_id | payment\_method | Room\_ number | total\_ amount | card\_number | payment\_details |
| 12 | Credit Card | 102 | 300 | 8977925484464 | Service |

**Functional Dependencies:**

* Payment\_id→payment\_method,Room\_number,total\_amout,card\_number,payment\_details

**Normalization:**

* This FD is trivial since Payment\_ID is the primary key, therefore, it is in BCNF

**7) Room**( Room\_Number,room\_type, floor\_number, room\_price, availability)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Room\_Number | room\_type | floor\_Number | room\_price | availability |
| 190 | Single | 3 | 200 | occupied |
| 204 | Double | 2 | 400 | available |

**Functional Dependency:**

* room\_number → room\_type, floor\_Number, room\_price, availability

**Normalization:**

* Room\_number is a primary/super key.Thus table is in **BCNF**

8) **Service** (service\_id, Emp\_ID, serv\_type, Room\_number, serv\_time)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| service\_id | Emp\_ID | serv\_type | Room  \_number | serv\_time |
| 1 | 1 | Food Delivery | 108 | 12:00:00 |

**Functional Dependencies:**

* service\_id →,Emp\_id,serve\_type, Room\_number, serv\_time

**Normalization:**

* In this schema, service\_id is the primary key accordingly, it is a **superkey**.However, Emp\_ID is not in the superkey, and the second FD is not trivial, thus it is not in BCNF. We only need to remove Emp\_name, we do not need to create a table with only emp\_ID because we already have the Employee entity. The new table is thus :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| service\_id | Emp\_ID | serv\_type | Room  \_number | serv\_time |
| 6987 | 23 | Reception | 898 | 12:56 |

9) **Premium Services** (Pservice\_ID, Discount\_Rate, Reward\_Points, Service\_Type, PServ\_stime, PServ\_etime)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pservice\_ID | Discount \_Rate | Reward\_ Points | Service\_Type | PServ\_ stime | PServ\_ etime |
| 1 | 0.20 | 40 | Car Parking | 12:00:00 | 12:10:00 |

**Functional Dependencies:**

* Pservice\_ID → Discount\_Rate, Rewards\_Point, Service\_Type, PServ\_stime, PServ\_etime.

**Normalization:**

* Premium\_ID is the primary key of this schema therefore it is the **superkey** and this schema is in **BCNF**

10) **Events** (event\_id, event\_date\_time, event\_type).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| event\_id | branch\_id | description | date | time | event\_type |
| 252 | 2 | Birthday | 13/5/2021 | 14:00 | Celebration |

**Functional Dependencies:**

* event\_id → branch\_id, description, date, time, event\_type.

**Normalization:**

* “Event\_id” is a primary key for this schema therefore it is the **superkey** and this schema is in **BCNF**.

11) **Restaurant** (rest\_name, branch\_id, rest\_open\_time, rest\_close\_time, number\_of\_tables, rest\_phNum, rest\_type)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| rest\_name | branch\_id | rest\_open\_time | rest\_close\_time | number\_of\_tables | rest\_phNum | rest\_type |
| Rosalind | 1 | 8:00 | 22:00 | 30 | 91846834 | Casual |

**Functional Dependencies:**

* rest\_name, rest\_phNum, →branch\_id, rest\_open\_time, rest\_close\_time, number\_of\_tables, rest\_type,rest\_phNum,

**Normalization:**

* Rest\_name is a primary key for this schema. Thus, it is a **superkey** and this relation schema is in **BCNF**.

12) **Store** (store\_name, store\_phNum, opening\_time, closing\_time, branch\_id)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| store\_name | store\_phNum | opening\_time | closing\_time | branch\_id |
| FOUR | 7584522 | 8:00 | 18:00 | 4 |

**Functional Dependencies:**

* Store\_name,store\_phNum → opening\_time, closing\_time, branch\_id, Store\_phNum

**Normalization:**

* “store\_name” is a **primary** key, and store\_phNum is a candidate key for this relational schema. Therefore, together they form a **superkey** and this relational schema is in **BCNF**.

**13) Visitor**(visitor\_id, visitor\_name, visitor\_phNum)

|  |  |  |
| --- | --- | --- |
| Visitor\_id | visitor\_name | visitor\_phNum |
| 4 | Nahla Baalbaki | 78234876 |

**Functional Dependencies:**

* Visitor\_id, visitor\_phNum, visitor\_name → visitor\_name, visitor\_phNum,.

**Normalization:**

* The above FD satisfies both **BCNF** properties; it is **trivial** and the left side is a **superkey**.

**14) Credit\_Card**(card\_number, pin\_code, expiry\_date)

|  |  |  |
| --- | --- | --- |
| card\_number | pin\_code | expiry\_date |
| 125478521225632 | 1321 | 2025-01-01 |

**Functional Dependencies:**

* card\_number→ pin\_code,expiry\_date.

**Normalization:**

* The above FD satisfies both **BCNF** properties; it is **trivial** and the left side is a **superkey**.

1. **dept\_emp:** This relation between department and Employee shows that the department has more than 1 employee however the employee has only one department where he belongs to.

|  |  |  |  |
| --- | --- | --- | --- |
| Emp\_ID | Dep\_Name | Shift\_Dstart | Shift\_Dend |

**Functional dependencies :**

* Emp\_ID → Dep\_Name, Shift\_Dstart, Shift\_Dend

**Normalization :**

* **“**Emp\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

2. **Employs** : This relation is between Branch and Employee. It is one to many since a branch can employ numerous employees but employees can only work at one branch. This relation ensures total participation since every employee works at a branch and every branch has employees.

|  |  |
| --- | --- |
| Emp\_ID | Branch\_ID |

**Functional dependencies :**

* Emp\_ID → Branch\_ID

**Normalization :**

* **“**Emp\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

3. **emp\_stores**: This relation between Employee and Stores shows that the store has more than 1 employee but the employee works only in one store.`

|  |  |
| --- | --- |
| Emp\_ID | store\_name |

**Functional dependencies :**

* Emp\_ID → store\_name

**Normalization :**

* **“**Emp\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

4. **emp\_rest**: This relation between Employee and Restaurant shows that the Restaurant has more than 1 employee but the employee works only in one Restaurant. It has total participation from the restaurant’s side since every restaurant must have an employee working in it.

|  |  |
| --- | --- |
| Emp\_ID | rest\_name |

**Functional dependencies :**

* Emp\_ID → rest\_name

**Normalization :**

* **“**Emp\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

5. **emp\_event**: This relation is between Employee and Events, since it is necessary to know which employees are supervising which event. It is many to one since many employees can be at one event but no employee can be at two events at the same time. This relation has total participation from the events’ side since every event has to have employees overseeing it.

|  |  |
| --- | --- |
| Emp\_ID | event\_id |

**Functional dependencies :**

* Emp\_ID → event\_id

**Normalization :**

* **“**Emp\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

6. **emp\_payment:** This relation is between Employee and payment, it shows that an employee can do more than one payment but the payment has only one employee that certifies it .

|  |  |
| --- | --- |
| payment\_id | Emp\_ID |

**Functional dependencies :**

* payment\_id → Emp\_ID

**Normalization :**

* **“**Emp\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

7. **Guest\_rest:** This relation is between Guest and Restaurant which goes to show that restaurants can have many guests as well as guests have the choice to visit many restaurants.

|  |  |
| --- | --- |
| guest\_ID | rest\_name |

**Functional dependencies :**

* guest\_ID → rest\_name

**Normalization :**

* **“**guest\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

8. **Guest\_payment** : This relation is necessary to link every guest with their payment details. It is one to one since every guest can have one single payment and every payment is done by one guest. It has total participation since every guest has to pay and every payment is done by a guest.

|  |  |
| --- | --- |
| guest\_ID | payment\_id |

**Functional dependencies :**

* guest\_ID → payment\_id

**Normalization :**

* **“**guest\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

9. **Guest\_visitor** : This relation links guests and their visitors. It is one to many because a guest can have many visitors but a visitor can be invited by only one guest. This relation has total participation from the visitors’ side but not the guest’s since not all guests have to have visitors over.

|  |  |
| --- | --- |
| visitor\_id | guest\_ID |

**Functional dependencies :**

* visitor\_id → guest\_ID

**Normalization :**

* **“**guest\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

10. **Prem\_guest\_service**: This relation is between premium customers and premium services. It is many to many since every guest can enjoy many services and each service is enjoyed by many guests. It does not ensure total participation since not all premium guests have to enjoy a given service.

|  |  |
| --- | --- |
| guest\_ID | Pservice\_id |

**Functional dependencies :**

* guest\_ID → Pservice\_id

**Normalization :**

* **“**guest\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

11. **guest\_serv:** This relation is between guest and service and it shows that the services can be provided to many guests, and a service used and enjoyed by many guests..

|  |  |
| --- | --- |
| guest\_ID | service\_id |

**Functional dependencies :**

* guest\_ID → service\_id

**Normalization :**

* **“**guest\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

12. **Room\_Reservation:** This relation is between Reservation and Room. It is one to one since every room can be reserved once and every reservation can be reserved for one room.

|  |  |
| --- | --- |
| Room\_number | reservation\_id |

**Functional dependencies :**

* Room\_number→ reservation\_id

**Normalization :**

* **“**room\_number” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

13**. Guest\_reserve**: This relation is between guests and their reservation, to ensure that all reservation details are linked to their respective guest. This is a one to one relation since every guest can only make one reservation at a given time. It ensures total participation from both sides since every guest has to make a reservation, and every reservation is made bby some guest.

|  |  |
| --- | --- |
| guest\_ID | reservation\_id |

**Functional dependencies :**

* guest\_ID → reservation\_id

**Normalization :**

* **“**guest\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

14. **Guest\_event** :This relation is between guests and events. It is many to many since every guest can attend many events, and every event is attended by many guests. It ensures total participation from the event’s side since every event must be attended by some guest, unless it is a really, really bad one.

|  |  |
| --- | --- |
| guest\_ID | event\_id |

**Functional dependencies :**

* guest\_ID → event\_id

**Normalization :**

* **“**guest\_ID” is the primary key, and thus the **superkey**, so the above table is in **BCNF**.

**MYSQL**

create database TheFour;

use TheFour;

create table guest(

guest\_ID int auto\_increment,

guest\_name varchar(100),

guest\_phoneNb long,

guest\_age int,

guest\_email varchar(100),

guest\_country varchar(50),

reward\_points int,

PRIMARY KEY (guest\_ID)

);

insert into guest(guest\_name, guest\_phoneNb, guest\_age, guest\_email, guest\_country, reward\_points) values

('Lara Idriss','01589746','25','lara.idriss@gmail.com','Lebanon','0'),

('Yasmine Samadi','01987456','21','yasmine.samadi@gmail.com','Russia','0'),

('Hala Mikkawi','01232324','28','hala.mikkawi@gmail.com','Lebanon','1500'),

('Karim Zeidan','01454545','30','karim.zeidan@gmail.com','UAE','300'),

('Mohamad Saamdi','0145658','22','mohamad.samadi@gmail.com','UAE','2000'),

('Rawad Zaatari','01363698','27','rawad.zaatari@gmail.com','Lebanon','360'),

('Faysal Sabaayoun','01544758','35','faysal.sabaayoun@gmail.com','Italy','0'),

('Riham Isamil','01369658','55','riham.isamil@gmail.com','Spain','980'),

('Solar Yaman','01234552','36','solar.yaman@gmail.com','Lebanon','320'),

('Jawad Barakat','01478963','40','jawad.barakat@gmail.com','Lebanon','0'),

('Liam Chamli', '76722322', '24', 'liam.chamli@outlook.com', 'Italy','0'),

('Olivia Johnson', '70550550', '29', 'olivia.johnson@outlook.com', 'America','500'),

('Lara Kanso', '03454777', '18', 'lara.kanso@outlook.com', 'Lebanon','0'),

('Isabel Brown', '76001101', '32', 'isabel.brown@outlook.com', 'Africa','1200'),

('Jad Ismail', '76231114', '25', 'jad.ismail@outlook.com', 'Lebanon','0'),

('Mia Dbouk', '03669978', '46', 'mia.dbouk@outlook.com', 'Paris','0'),

('Lucas Williams', '70814283', '21', 'lucas.williams@outlook.com', 'Brazil','0'),

('Celine Kobeissy', '76111200', '40', 'celine.kobeissy@outlook.com', 'China','600'),

('Bella Koleilat', '76098625', '24', 'bella.koleilat@outlook.com', 'Italy','900'),

('James Davis', '71872243', '36', 'james.davis@outlook.com', 'Greece','0'),

('Alaa Kiran', '89326489', '30', 'alaa.kiran@gmail.com', 'India','1000'),

('Tamara Morgan', '23975214', '40', 'tamara.morgan@hotmail.com', 'USA','520'),

('Laura Bryan', '96529245', '60', 'laura.bryan@outlook.com', 'Canada','960'),

('Iman Youssef', '23863253', '20', 'iman.youssef@gmail.com', 'Africa','780'),

('Issa Ismail', '76231114', '22', 'issa.ismail@hotmail.com', 'Algeria','0'),

('Omar Yousef', '32653544', '36', 'omar.youssef@outlook.com', 'Tunisia','0'),

('Lamia Jawad', '70814283', '45', 'lamia.jawad@hotmail.com', 'Brazil','0'),

('Wael Jarrah', '23576754', '32', 'wael.jarrah@gmail.com', 'Lebya','0'),

('Maria Tamer', '1246733', '22', 'MariaT@gmail.com', 'Lebanon','0'),

('Karim Dia', '39276214', '21', 'KarimD@outlook.com', 'Egypt','0'),

('Lina Jaber', '76712093', '34', 'lina.jaber@outlook.com', 'Lebanon','560'),

('Melissa Maatouk', '70520150', '29', 'melissa.maatouk@outlook.com', 'America','0'),

('Jana Kanso', '03453291', '40', 'jana.kanso@outlook.com', 'Lebanon','0'),

('Jane Austen', '76122290', '70', 'jane.austen@outlook.com', 'America','450'),

('Jad Chahine', '76331414', '35', 'jad.chahine@outlook.com', 'Lebanon','690'),

('Mia Chamli', '03663958', '46', 'mia.chamli@outlook.com', 'London','790'),

('Sara Aoun', '70233283', '29', 'sara.aoun@outlook.com', 'Lebanon','0'),

('Celine Mahjoub', '76126200', '30', 'celine.mahjoub@outlook.com', 'China','0'),

('Ali Jaber', '76098625', '64', 'ali.jaber@outlook.com', 'Turkey','0'),

('Fadi Ghandour', '71822213', '76', 'fadi.ghandour@outlook.com', 'Lebanon','0')

;

select \* from guest;

create table premium\_guest(

premium\_id int auto\_increment,

guest\_ID int,

guest\_name varchar(100),

guest\_phoneNb int,

guest\_age int,

guest\_email varchar(100),

guest\_country varchar(50),

primary key(premium\_id),

foreign key(guest\_ID) references guest(guest\_ID)

);

insert into premium\_guest(guest\_ID,guest\_name,guest\_phoneNb,guest\_age,guest\_email,guest\_country) values

('3','Hala Mikkawi','01232324','28','hala.mikkawi@gmail.com','Lebanon'),

('4','Karim Zeidan','01454545','30','karim.zeidan@gmail.com','UAE'),

('5','Mohamad Saamdi','0145658','22','mohamad.samadi@gmail.com','UAE'),

('6','Rawad Zaatari','01363698','27','rawad.zaatari@gmail.com','Lebanon'),

('8','Riham Isamil','01369658','55','riham.isamil@gmail.com','Spain'),

('9','Solar Yaman','01234552','36','solar.yaman@gmail.com','Lebanon'),

('12','Olivia Johnson', '01550550', '29', 'olivia.johnson@outlook.com', 'America'),

('14','Isabel Brown', '01001101', '32', 'isabel.brown@outlook.com', 'Africa'),

('18','Celine Kobeissy', '01111200', '40', 'celine.kobeissy@outlook.com', 'China'),

('19','Bella Koleilat', '01098625', '24', 'bella.koleilat@outlook.com', 'Italy'),

('22','Alaa Kiran', '01326489', '30', 'alaa.kiran@gmail.com', 'India'),

('23','Tamara Morgan', '01975214','40', 'tamara.morgan@hotmail.com', 'USA'),

('24','Laura Bryan', '01529245','60', 'laura.bryan@outlook.com', 'Canada'),

('25','Iman Youssef', '01863253', '20', 'iman.youssef@gmail.com', 'Africa'),

('31','Lina Jaber', '76712093', '34', 'lina.jaber@outlook.com', 'Lebanon'),

('34','Jane Austen', '76122290', '70', 'jane.austen@outlook.com', 'America'),

('35','Jad Chahine', '76331414', '35', 'jad.chahine@outlook.com', 'Lebanon'),

('36','Mia Chamli', '03663958', '46', 'mia.chamli@outlook.com', 'London')

;

select \* from premium\_guest;

create table employees(

emp\_ID int auto\_increment,

emp\_name varchar(100),

emp\_email varchar(100),

emp\_phNum long,

emp\_age int,

emp\_city varchar(100),

emp\_street varchar(100),

emp\_extension int,

dept\_name varchar(100),

emp\_salary int,

primary key(emp\_ID)

);

insert into employees(emp\_name,emp\_email,emp\_phNum,emp\_age,emp\_city,emp\_street,emp\_extension,dept\_name,emp\_salary) values

('Zad Kojok','zad.kojok@TheFour.com','02587836','26','Lebanon-Beirut','Airport Street','1478','Reception',9000),

('Yara Akoum','yara.akoum@TheFour.com','02874593','27','Lebanon-Saida','Daleaa Street','1485','HouseKeeping',4600),

('Jana Zwiya','jana.zwiya@TheFour.com','02366978','26','Lebanon-Jbeil','Rabieh Street','1463','Accounting',9100),

('Hazem Joeudi','hazem.joeudi@TheFour.com','02251478','23','Lebanon-Jbeil','Rabieh Street','1474','Managment',5400),

('Ali Zein','ali.zein@TheFour.com','02147896','36','Lebanon-Tripoli','Hames Street','1430','Security',3000),

('Ghadi Merhi','ghadi.merhi@TheFour.com','02365214','45','Lebanon-Tyr','Zein Street','1432','Accoutning',6700),

('Rolanda Chhade','rolanda.chhade@TheFour.com','02369869','35','Lebanon-Beirut','Verdun Street','1440','Security',5600),

('Angela Kechtban','angela.kechtban@TheFour.com','02741850','32','Lebanon-Tripoli','Hames Street','1498','Sales and Marketing',7800),

('Marwa Antar','marwa.antar@TheFour.com','02001473','32','Lebanon-Saida','Hilaliyeh Street','1423','Housekeeping',2500),

('Manar Dafer','manar.dafer@TheFour.com','02225873','34','Lebanon-Jbeil','Hames Street','1410','Reception',3600),

('Zein Hamade', 'zein.hamade@TheFour.com', '70809809', '21', 'Lebanon-Baabda', 'Airport Street', '1254', 'Housekeeping', 7800),

('Fatima Baalbaki', 'fatima.baalbaki@TheFour.com', '70123855', '23', 'Lebanon-Beirut', 'Verdun Street', '3222', 'Reception', 9100),

('Hadi Charaf', 'hadi.charaf@TheFour.com', '70060935', '35','Lebanon-Baabda', 'Airport Street', '1202', 'Management', 1200),

('Abbas Khalil', 'abbas.khalil@TheFour.com', '71205288', '39', 'Lebanon-Beirut', 'Dbayeh Strret', '2303', 'Accounting', 4200),

('Sarah Amhaz', 'sarah.amhaz@TheFour.com', '70435339', '21', 'Lebanon-Baabda', 'Baabda Street', '1233', 'Security', 2000),

('Rayan Ghandour', 'rayan.ghandour@TheFour.com', '71525844', '28', 'Lebanon-Beirut', 'Bir Hassan Street', '1444', 'Maintenance', 2500),

('Alex Hamad', 'alex.hamad@TheFour.com', '03772083', '19', 'Lebanon-Baabda', 'Baabda Strret', '4766', 'Food and Beverage', 3600),

('Zeinab Chahoud', 'zeinab.chahoud@TheFour.com', '76052673', '44','Lebanon-Beirut', 'Jnah Street', '5987', 'HouseKeeping', 4200),

('Omar Kanso', 'omar.kanso@TheFour.com', '76222123', '31','Lebanon-Beirut', 'Dbayeh Street', '3285', 'Reception', 3200),

('Ihab Jihad', 'ihab.jihad@TheFour.com', '28356353', '30', 'Lebanon-Beirut','Abdulla Street', '1234', 'Management', 5800),

('Fahed Rabih', 'fahed.rabih@TheFour.com', '23857932', '33', 'Lebanon-Baalback','Ali Street', '5678', 'Accounting', 4500),

('Kazem Wael', 'kazem.wael@TheFour.com', '32589632', '22', 'Lebanon-Beirut', 'Fouad Street' ,'2345', 'HouseKeeping', 4500),

('Ahmad Issam', 'ahmad.issam@TheFour.com', '32985635', '27', 'Lebanon-Saida', 'Sharif Street','3456', 'Reception', 1700),

('Ali Rabeh', 'ali.rabeh@TheFour.com', '329573287', '28', 'Lebanon-Beit Shama', 'Osman Street','1345', 'Food and Beverages', 4500),

('Pamela Yaacoub', 'pamela.yaacoub@TheFour.com', '327846823', '29', 'Lebanon-Koraytem', 'Ankara Street','1667', 'Maintenance', 1400),

('Esmail Ayoub', 'esmail.ayoub@TheFour.com', '39826432', '39', 'Lebanon-Bhamdoun', 'Holland Street','1889', 'Security', 7600),

('Zeina Moazzen', 'zeina.moazzen@TheFour.com', '382956893', '40', 'Lebanon-Sour','Abdulla Street' ,'1334', 'HouseKeeping', 2500),

('Ruba Kassem', 'ruba.kassem@TheFour.com', '389265983', '35', 'Lebanon-Damour','Jaber Street' ,'1889', 'Human Resources', 3300),

('Nasim Alyan', 'nasim.alyan@TheFour.com', '3295732035', '26', 'Lebanon-Tripoli', 'Karim Street','1445', 'Sales and Marketing', 2000),

('Tania Hatoum' , 'tania.hatoum@theFour.com', '70812779', '28', 'Lebanon-Nabatieh', 'Hay El Saray Street' , '1154', 'Management', 3200),

('Roua Charife', 'roua.charife@theFour.com', '70133455', '25', 'Lebanon-Verdun', '10th Avenu Street' , '3122', 'House Keeping', 6000),

('Rami Zeineddine', 'rami.zeineddine@theFour.com', '71060235', '30', 'Lebanon-Kfarjoz' , 'Abar Fakhreddine Street' , '1292', 'Reception', 5100),

('Abbas Jaafar', 'abbas.jaafar@theFour.com', '71233288', '29', 'Lebanon-Baalbak', 'El Sakhra street' , '5113', 'Accounting', 3200),

('Lynn Hajjar', 'lynn.hajjar@theFour.com', '70435111', '23', 'Lebanon-Baabda', 'Saydaliye street', '1633', 'Security', 2250),

('Marwa Faaour', 'marwa.faaour@theFour.com', '71333844', '31', 'Lebanon-Jbeil', 'Elysee street', '1294', 'Management', 3300),

('Yara Karam' , 'yara.karam@theFour.com', '03745583', '20', 'Lebanon-Furn El Chebbek', ' Knise street' , '4236', 'Food and Beverage', 5000),

('Mireille Kaouk', 'mireille.kaouk@theFour.com', '76332673', '34', 'Lebanon-Saida', 'El nejme street' , '1987', 'HouseKeeping', 2500),

('Reine Ezzedine', 'reine.ezzedine@theFour.com', '76290223', '38', 'Lebanon-Dbayeh', 'Lor street' , '6525', 'Reception', 3600),

('Karim Naser', 'karim.naser@theFour.com', '03934539', '22', 'Lebanon-Khalde', 'Kfarchoun Street', '8309', 'Accounting', 2560),

('Leila Naser', 'leila.naser@theFour.com', '03587527', '24', 'Lebanon-Khalde', 'Kfarchoun Street', '8319', 'Sales and Marketing', 2590)

;

select \* from employees;

drop table Room;

create table Room(

Room\_number int,

room\_type varchar(50),

floor\_number int,

room\_price double,

availability varchar(50),

PRIMARY KEY(Room\_number)

);

insert into room(Room\_number, room\_type,floor\_number,room\_price,availability) values

('101','Standard','1','200.00','Available'),

('102','Double','1','350.00','Occupied'),

('103','Standard','1','200.00','Available'),

('104','Family','1','450.00','Available'),

('105','Suite','1','550.00','Occupied'),

('106','Double','1','350.00','Available'),

('107','Standard','1','200.00','Occupied'),

('108','Suite','1','550.00','Occupied'),

('109','Family','1','450.00','Occupied'),

('110','Double','1','350.00','Occupied'),

('111', 'Family', '2', '450.00','Available'),

('112', 'Standard', '2', '200.00','Available'),

('113', 'Suite', '2', '550.00','Available'),

('114', 'Standard', '2', '200.00','Available'),

('115', 'Family', '2', '450.00','Occupied'),

('116', 'Standard', '2', '200.00','Available'),

('117', 'Standard', '2','200.00','Occupied'),

('118', 'Suite', '2', '550.00','Available'),

('119', 'Standard', '2', '200.00','Occupied'),

('120', 'Suite', '2', '550.00', 'Available'),

('121', 'Standard', '3', '200.00', 'Available'),

('122', 'Suite', '3', '550.00', 'Occupied'),

('123', 'Standard', '3', '200.00','Available'),

('124', 'Double', '3','350.00','Occupied'),

('125', 'Suite', '3', '550.00', 'Available'),

('126', 'Standard', '3', '200.00','Occupied'),

('127', 'Double', '3','350.00','Available'),

('128', 'Family', '3', '450.00', 'Occupied'),

('129', 'Family', '3', '450.00','Available'),

('131', 'Suite', '4', '550.00', 'Available'),

('132', 'Suite', '4', '550.00', 'Available'),

('133', 'Standard', '4','200.00','Occupied'),

('134', 'Double', '4','350.00','Occupied'),

('135', 'Standard', '4', '200.00', 'Available'),

('136', 'Standard', '4', '200.00','Occupied'),

('137', 'Double', '4','350.00','Occupied'),

('138', 'Family', '4', '450.00', 'Available'),

('139', 'Family', '4', '450.00','Occupied'),

('140', 'Suite', '4', '550.00','Occupied')

;

select \* from Room;

create table Service(

service\_id int auto\_increment,

emp\_ID int,

serv\_type varchar(50),

Room\_number int,

serv\_time time,

PRIMARY KEY(service\_id),

FOREIGN KEY(emp\_ID) references employees(emp\_ID)

);

insert Service(emp\_ID, serv\_type,Room\_number,serv\_time) values

('1','Food Delivery','108','12:00:00'),

('2','Room Cleaning','102','02:25:00'),

('3','Dry Cleaning','110','06:32:00'),

('4','Laundry','109','05:07:00'),

('5','Food Delivery','103','01:00:00'),

('6','Room Cleaning','107','03:58:00'),

('7','Laundry','106','10:21:00'),

('8','Food Delivery','105','09:32:00'),

('9','Dry Cleaning','102','08:55:00'),

('10','Laundry','101','01:58:00'),

('11', 'Food Delivery', '120', '12:45:00'),

('12', 'Room Service', '121', '15:32:00'),

('13', 'Reception Guests', '122', '18:21:00'),

('14', 'Laundry', '123', '21:01:00'),

('15', 'Dry cleaning', '124', '13:12:00'),

('16', 'Housekeeping Service', '125', '18:57:00'),

('17', 'Laundry', '126', '23:34:00'),

('18', 'Dry cleaning', '127','14:17:00'),

('19', 'Housekeeping Service', '128', '12:56:00'),

('20', 'Room Service', '129', '13:42:00'),

('21', 'Food Delivery', '130', '16:23:00'),

('22', 'Catering services', '131', '23:44:00'),

('23', 'Laundry', '132', '23:44:00'),

('24', 'Dry cleaning', '133', '23:44:00'),

('25', 'Room Service', '134', '12:50:10'),

('26', 'Concierge services', '126', '1:31:00'),

('27', 'Room Service', '135', '18:11:00'),

('28', 'Doctor on call', '138','22:05:00'),

('29', 'Dry cleaning', '136', '12:12:00'),

('30', 'Doctor on call', '137', '19:57:00'),

('31', 'Laundry', '130', '23:44:00'),

('32', 'Dry cleaning', '139' ,'18:17:00'),

('33', 'Reception Guests', '128', '10:45:00'),

('34', 'Room Service', '121','11:31:00'),

('35', 'Reception Guests','122','78:11:00'),

('36', 'Laundry', '128','22:05:00'),

('37', 'Dry cleaning', '124','12:12:00'),

('38', ' Room Service', '113','19:57:00'),

('39', 'Laundry', '120', '23:44:00'),

('40', 'Dry cleaning', '124' ,'18:17:00')

;

select \* from Service;

create table PremiumServices(

Pservice\_ID int auto\_increment,

Discount\_Rate decimal(3,2),

Reward\_Points int,

Service\_Type varchar(50),

PServ\_stime time,

PServ\_etime time,

PRIMARY KEY(Pservice\_ID)

);

insert PremiumServices( Discount\_Rate, Reward\_Points,Service\_Type, PServ\_stime, PServ\_etime) values

('0.20','40','Car Parking','12:00:00','12:10:00'),

('0.15','30','VIP lounge','02:25:00','03:25:00'),

('0.07','14','Car Parking','06:32:00','06:47:00'),

('0.11','22','Rooftop bar','05:07:00','07:07:00'),

('0.15','30','VIP buffet ','01:00:00','01:50:00'),

('0.10','20','VIP lounge','03:58:00','04:30:00'),

('0.10','20','Spa','10:21:00','10:50:00'),

('0.13','26','Beauty Salon','03:32:11','05:00:00'),

('0.12','24','Rooftop bar','08:54:00','10:00:00'),

('0.10','20','Spa','01:18:00','03:30:00'),

('0.10','20','Car Parking','13:00:00','13:30:00'),

('0.05','10','Spa','14:30:00','15:30:00'),

('0.12','24','Car Parking','12:35:00','12:50:00'),

('0.14','28','Beauty Salon','17:05:00','18:05:00'),

('0.20','40','Vip Lounge','20:00:00','20:30:00'),

('0.06','12','Spa','12:40:00','13:40:00'),

('0.07','14','Spa','10:50:00','11:50:00'),

('0.13','26','Beauty Salon','13:00:00','13:40:00'),

('0.16','32','Rooftop Bar','21:00:00','22:35:00'),

('0.10','20','Car Parking','13:52:00','14:00:00')

;

select \* from PremiumServices;

create table branch (

branch\_id int auto\_increment,

B\_Country varchar(100),

B\_City varchar(100),

B\_Street varchar(100),

branch\_zip int,

branch\_phNum long,

primary key(branch\_id)

);

insert branch(B\_Country,B\_City,B\_Street, branch\_zip, branch\_phNum) values

('Lebanon','Beirut', 'Hamra Street', '123', '96103118119' ),

('Netherlands', 'Amsterdam', 'Kings Street', '23', '9881243432949'),

('Egypt' , 'Cairo', 'Ajaa Street','56', '9662282374'),

('Turkey' , 'Istanbul' , 'Taksim' , '111', '2891838123')

;

select \* from branch;

create table restaurant(

rest\_name varchar(50),

branch\_id int,

rest\_open\_time time,

rest\_close\_time time,

number\_of\_tables int,

rest\_PhNum long,

rest\_type varchar(50),

primary key(rest\_name),

foreign key (branch\_id) references branch(branch\_id)

);

insert restaurant(rest\_name,branch\_id,rest\_open\_time, rest\_close\_time, number\_of\_tables, rest\_phNum, rest\_type) values

('Italiono1','1', '10:00:00', '24:00:00', '14', '01443223', 'Italian Food'),

('BeBabel1','1', '8:00:00', '24:00:00', '11', '01022987', 'Lebanese Food'),

('The Grill1','1', '11:00:00', '23:30:00', '12', '01765987', 'Mexican Food'),

('ChopSticks1','1', '13:00:00', '24:00:00', '10', '01027345', 'Chinese Food'),

('Italiono2','2', '10:00:00', '24:00:00', '14', '015874693', 'Italian Food'),

('BeBabel2','2', '8:00:00', '24:00:00', '11', '01257488', 'Lebanese Food'),

('The Grill2','2', '11:00:00', '23:30:00', '12', '01223336', 'Mexican Food'),

('ChopSticks2','2', '13:00:00', '24:00:00', '10', '01996688', 'Chinese Food'),

('Italiono3','3', '10:00:00', '24:00:00', '14', '01585745', 'Italian Food'),

('BeBabel3','3', '8:00:00', '24:00:00', '11', '015696584', 'Lebanese Food'),

('The Grill3','3' ,'11:00:00', '23:30:00', '12', '014745835', 'Mexican Food'),

('ChopSticks3','3', '13:00:00', '24:00:00', '10', '012525896', 'Chinese Food'),

('Italiono4','4', '10:00:00', '24:00:00', '14', '012585476', 'Italian Food'),

('BeBabel4','4', '8:00:00', '24:00:00', '11', '012363258', 'Lebanese Food'),

('The Grill4','4', '11:00:00', '23:30:00', '12', '012788874', 'Mexican Food'),

('ChopSticks4','4', '13:00:00', '24:00:00', '10', '012585964', 'Chinese Food')

;

select \* from restaurant;

create table store(

store\_name varchar(50),

store\_phNum long,

opening\_time time,

closing\_time time,

branch\_id int,

primary key(store\_name),

foreign key (branch\_id) references branch(branch\_id)

);

insert into store(store\_name, store\_phNum, opening\_time, closing\_time, branch\_id) values

('Sweet Store', '238956934','8:00:00','21:00:00','1'),

('Focal Feed','289357621','9:00:00','22:00:00','1'),

('Corner Cafe','839469349','8:00:00','16:30:00','2'),

('Healthy Treats','928357928','9:00:00','14:30:00','3'),

('Aqua','17432895','8:30:00','12:00:00','4'),

('Summer Gift Shop','928365235','9:00:00','12:00:00','4'),

('Stone Tech Zone','32846329','9:00:00','20:00:00','4'),

('Columbia','32958632','8:00:00','18:00:00','2'),

('Buffalo','32576945','9:00:00','19:00:00','3')

;

select \* from store;

drop table visitor;

Create table visitor(

visitor\_id int auto\_increment,

visitor\_name varchar(50),

visitor\_phNum long,

Room\_number int,

primary key (visitor\_id),

foreign key (Room\_number) references Room(Room\_number)

);

insert visitor(visitor\_name, visitor\_phNum, Room\_number) values

('Amjad Baalbaki','01258743','101'),

('Mahmoud Ali','01363696','102'),

('Jihad Mansour','01252525','103'),

('Rubina Ahamd','01236987','112'),

('Manal Dafer','01789654','112'),

('Perla Chahine','01369685','101'),

('Karim Maarouf','01363658','103'),

('Sarah Abdallah','01777888','106'),

('Bilal Sabayoun','01258593','107'),

('Hadi Saleh','01258796','106'),

('Ghadi Saleh','012578364','111'),

('Huda Merhi','014875962','117'),

('Nabila Yaman','01587496','101'),

('Hani Yehya','01259874','127'),

('Hana Chahine', '03970700','104'),

('Lea Hamdoun', '71482032','137'),

('Mira Hammoud', '76563212','121'),

('Louai Khalil', '78280101','125'),

('Mohammad Alama', '76250366','134'),

('Khaled Hamad', '81397414','136')

;

select \* from visitor;

create table credit\_card(

card\_number varchar(17),

pin\_code int,

expiry\_date date,

primary key (card\_number)

);

insert into credit\_card(card\_number, pin\_code, expiry\_date) values

('3697895872365258',7412,'2024-05-01'),

('6554445521289655' ,3698,'2025-06-01'),

('1258746936652874',1254,'2029-05-01'),

('2589634715896325',8562,'2025-02-01'),

('6987456745896321',7412,'2024-04-01'),

('3698574589632158',1235,'2025-01-01'),

('6985411225874123',2365,'2028-08-01'),

('6987532145698741',8547,'2023-01-01'),

('8745985636521478',2587,'2027-09-01'),

('3201874159632541',2739,'2025-07-01'),

('9635214785236954',1478,'2026-07-01'),

('8745963215478262',2367,'2024-05-01'),

('8547412536987452',7563,'2025-08-01'),

('3265987845122525',6987,'2027-03-01'),

('1234560325970243',2145,'2025-07-01'),

('4574456256439173',2587,'2028-07-01'),

('3295728459874254',4563,'2025-04-01')

;

select \* from credit\_card;

create table Payment(

payment\_id int auto\_increment,

payment\_method varchar(30),

Room\_number int,

total\_amount int,

card\_number varchar(17),

payment\_details varchar(50),

primary key (payment\_id),

foreign key (Room\_number) references Room(Room\_number),

foreign key (card\_number) references credit\_card(card\_number)

);

insert Payment (payment\_method,Room\_number, total\_amount, card\_number, payment\_details) values

('Debit Card', '101','250', '3697895872365258','Room Reservation'),

('Credit Card', '101','550', '6554445521289655','Service'),

('Debit Card', '110','450', '1258746936652874','Room Reservation'),

('Credit Card', '120','560', '2589634715896325','Room Reservation'),

('Debit Card', '132','690', '6987456745896321','Service'),

('Debit card', '111','200', '3698574589632158','Room Reservation'),

('Debit Card', '123','350', '6985411225874123','Service'),

('Credit card', '125','750', '6987532145698741','Room Reservation'),

('Credit Card','140', '300', '8745985636521478','Service'),

('Debit card','114', '300', '3201874159632541','Room Reservation'),

('Credit card','125', '350', '9635214785236954','Room Reservation'),

('Debit Card', '119','250', '8745963215478262','Room Reservation'),

('Credit card','103', '550', '8547412536987452','Room Reservation'),

('Credit card','136', '350', '3265987845122525','Room Reservation'),

('Credit Card ', '115','300', '1234560325970243','Room Reservation'),

('Credit card', '123','300', '4574456256439173','Service'),

('Credit Card', '101','200', '3295728459874254','Room Reservation')

;

select \* from Payment;

drop table Reservation\_details;

create table Reservation\_details(

reservation\_id int auto\_increment,

branch\_id int,

check\_in datetime,

check\_out datetime,

payment\_id int,

emp\_id int,

primary key( reservation\_id),

foreign key (branch\_id) references branch(branch\_id),

foreign key (payment\_id) references Payment(payment\_id),

foreign key (emp\_id) references employees(emp\_id)

);

insert Reservation\_details(branch\_id, check\_in, check\_out, payment\_id, emp\_id) values

('1', '2020-03-05 2:00:00', '2020-03-06 12:00:00', '1', '1'),

('3', '2020-03-07 2:00:00', '2020-03-08 12:00:00', '2', '2'),

('3', '2019-01-07 2:00:00', '2019-01-08 12:00:00', '3', '3'),

('4', '2018-07-01 2:00:00', '2018-07-02 12:00:00', '4', '4'),

('2', '2020-10-12 2:00:00', '2020-10-13 12:00:00', '5', '5'),

('1', '2005-11-17 2:00:00', '2005-11-18 12:00:00', '6', '6'),

('2', '2015-01-07 2:00:00', '2015-01-08 12:00:00', '7', '7'),

('3', '2015-01-09 2:00:00', '2015-01-10 12:00:00', '8', '8'),

('2', '2019-06-28 2:00:00', '2019-06-29 12:00:00', '9', '9'),

('4', '2019-09-20 2:00:00', '2019-09-21 12:00:00', '10', '10'),

('1', '2020-10-24 2:00:00', '2020-10-25 12:00:00', '11', '11'),

('3', '2020-11-25 2:00:00', '2020-11-26 12:00:00', '12', '12'),

('3', '2020-08-21 2:00:00', '2020-08-22 12:00:00', '13', '13'),

('4', '2020-09-12 2:00:00', '2020-09-13 12:00:00', '14', '14'),

('2', '2020-05-14 2:00:00', '2020-05-15 12:00:00', '15', '15'),

('1', '2020-10-11 2:00:00', '2020-10-12 12:00:00', '16', '16');

select \* from Reservation\_details;

create table department(

dept\_id int auto\_increment,

dept\_name varchar(100),

dept\_phNum long,

primary key(dept\_id)

);

insert department(dept\_name, dept\_phNum) values

('Management', '01373088'),

('Accounting', '01882333'),

('HouseKeeping', '01611194'),

('Reception', '01396369'),

('Reception', '01662709'),

('Food and Beverages', '01332082'),

('Security', '01240414'),

('Maintenance', '01446346'),

('Human Resource', '01581599'),

('Sales and Marketing', '01642035')

;

select \* from department;

create table Events(

event\_id int auto\_increment,

event\_date\_time datetime,

event\_type varchar(50),

primary key (event\_id)

);

insert Events(event\_type,event\_date\_time) values

('Workshop', '2021-12-15 20:00:00'),

('Ceminar', '2021-5-21 12:00:00'),

('Charity', '2021-7-13 14:00:00'),

('Birthday', '2021-7-11 20:30:00'),

('Workshop', '2021-5-21 18:30:00'),

('Birthday', '2021-10-25 17:00:00'),

('Workshop', '2021-8-26 13:00:00'),

('Valentines', '2022-2-14 20:00:00'),

('New Years Eve', '2022-12-31 23:59:59'),

('Ceminar', '2021-11-16 10:00:00'),

('New Years Eve', '2021-12-31 23:59:59')

;

Select \* from Events;