**Hotel management system**

**Problem statement:-**

* Hotel management system is dived into many departments like reservation, food and beverages, services, guests, rooms, employees, events.

* Each department has list of services to be offered to customers. Each department has its own id, name and employees working as well as services offered to guests.

* **Employees** are the one who works in the hotel, like managers, front desk, housekeeping, chefs … etc. Each employee has unique id, name, salary.

* **Manager**  who manages entire  hotel.

* **Guests** are identified by their id. Each guest has a name, contact info, purpose of visit, nationality, date of birth and allotted rooms based on their reservations.

* **Rooms** are identified by their unique id. Each room has room number, type, capacity, status. Each guest can reserve upto 5 room(1,5). Each room can have only (1,1)guests.

* **Reservations:** - info about guest reservations, including guest id, cate of arrival and departure, room number.

* **Service** category. Like Spa, laundry, room service. Etc. Each service has its id, name description, cost.

* **Events:** - info about events held in hotel. Type of event, date, location, no of attendees.

* **Food and beverages**: - types of food and drinks offered (south tali, north tali, Chinese cusins, hot drinks, soft drinks.)

ENTITY’S: -

Person: - employee, guest.

Place: - room.

Event: - function\_hall, conference\_hall, services

Concept: - reservation.

Object: - food and beverages.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.no | Entity | Attributes | Type |  | S.no | Entity | Attributes | Type |
| 1 | Employee | Emp\_name | Composite |  | 5 | Reservation | Guest name | Composite |
| Emp\_id | Single |  | Room type | Simple |
| Salary | Simple |  | Check in date | Simple |
| Position | Simple |  | Chec out date | Simple |
| Age | Derived |  | Cost | Single |
|  |  |  |  |  |  |  |
|  |  |  |  |  | 6 | Service | Service\_id | Single |
| 2 | Guest | Guest\_name | Simple |  | Service\_type | Simple |
| Guest\_id | Simple |  | Cost | Single |
| Contact info | Multivalued |  | Location | Multivalued |
| Nationality | Simple |  |  |  |  |  |
| Age | Simple |  | 7 | Food & bevarages | Food\_id | Single |
| Purpose of visit | Required |  |  | Guest\_id | Multivalued |
| Religion | Optional |  |  | Emp\_id | Single |
|  |  |  |  |  |  | Flavor | Multi valued |
| 3 | Room | Room \_no | Simple |  |  | Type | Single |
| Room\_type | Simple |  |  | Cooking\_method | Single |
| Capacity | Simple |  |  | Flavour | Single |
| Status | Simple |  |  |  |  |
| Cost | Simple |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 4 | Function & conference | Type | Simple |  |  |  |  |  |
| Attendees | Simple |  |  |  |  |  |
| Theme | Simple |  |  |  |  |  |
| Organizers | Simple |  |  |  |  |  |
| Cost |  |  |  |  |  |  |

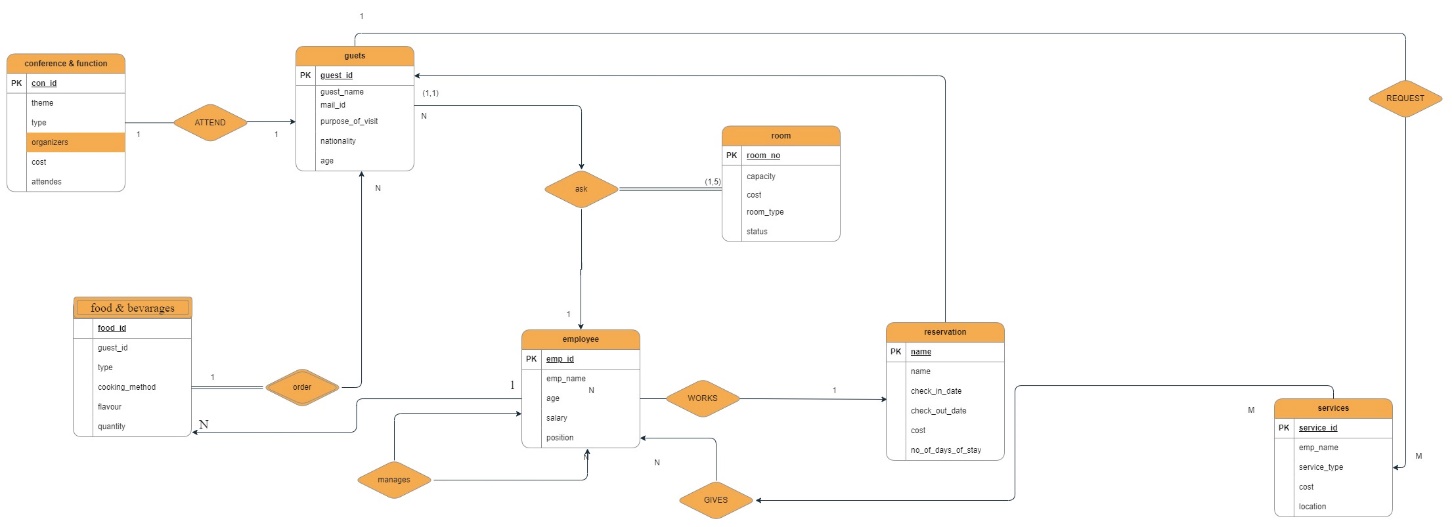
**Relationship for my entity’s: -**

**Relations ships:**

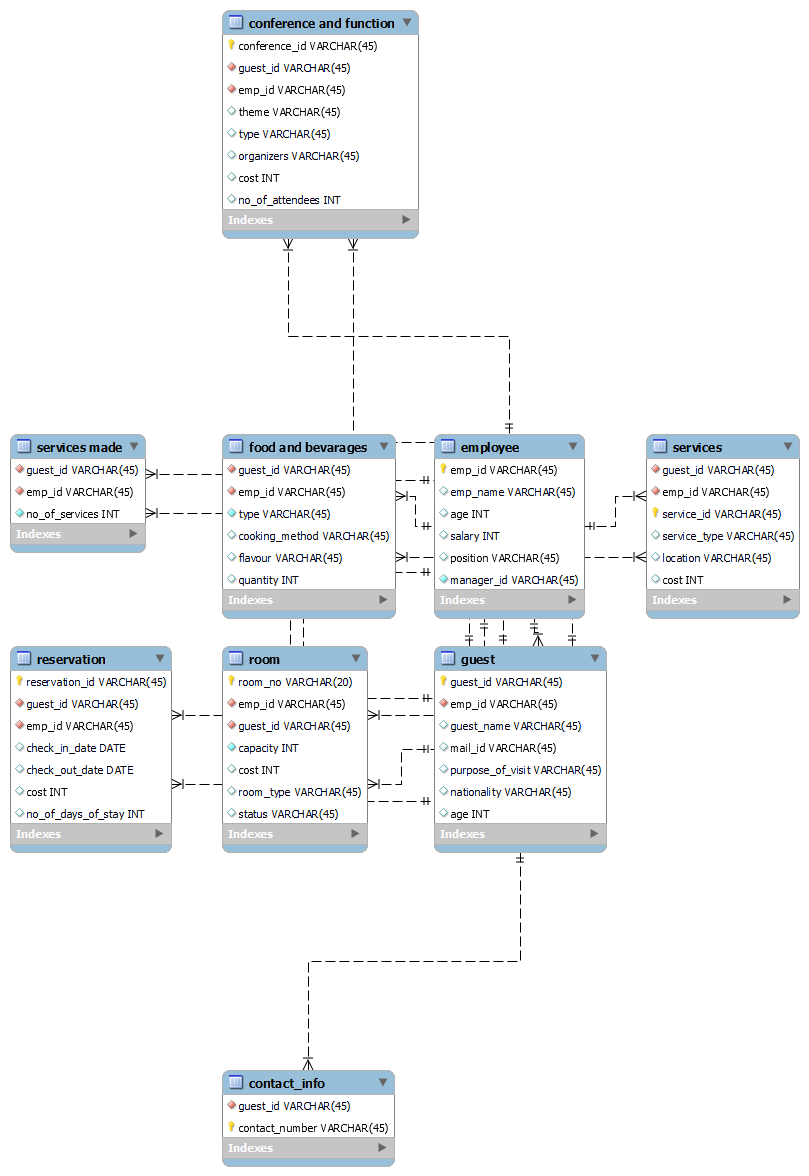
**Attend , ask ,  order , works , gives .**

ENTITY’S: -

1. Person: - employee, guest.
2. Place: - room.
3. Event: - function\_conference, services
4. Concept: - reservation.
5. Object: - food and beverages.

**** **ER-DIAGRAM**

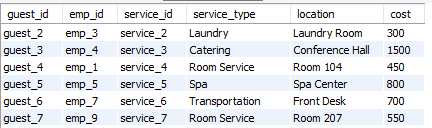
**ER-DIAGRAM GENERATED FROM MYSQL WORKBENCH**



**NORMALIZATION**

Table selected to find normalization is **Services made**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F |



**1nf Form: -**

The Above Relation Room Is Already in 1nf Form Because There Are No Multivalued Attributes (Atomic in Nature).

**Converting into 2nf Form: -**

**Steps:**

1.identify the prime and non-prime attributes

2.check if the candidate key is a composite key.

3.check if the non-prime attributes are fully dependent on the prime attributes.

4.check if the non-prime attribute is partially dependent on the prime attributes.

* The key for the given relation is (Guest\_id, emp\_id, service\_id) where (emp\_id, service\_id) are foreign keys and guest\_id is a primary key of relation room where three of them combine to form candidate key.
* The set of functional dependencies are f = {abc -> def, abc -> d,}
* {Def} may have redundant values so they cannot form the key.
* From the relation room prime attributes are {abc}
* From the relation non-prime attributes are{def}

**Checking For Partial Dependency**

{A -> D  ,  B -> D  , C -> D}

{A -> E  ,  B -> E  , C -> E}

{A -> F, B -> F, C -> F}

The Non-Prime Attributes {D, E, F} Are Partially Dependent On the Part of the Primary Key.

**2nf Decomposition**

{A -> D  ,  B -> D  , C -> D}

{A -> E  ,  B -> E  , C -> E}

{A -> F  ,  B -> F  , C -> F}

All The Above 9 Functional Dependencies Are Creating Problems Because There Are Non-Prime Attributes That Are Partially Dependent On The Candidate Key.

So All These 9 Functional Dependencies Are Broken into 9 New Different Relations.

So The New 9 Relations are

1. (Guest\_id,Service\_type)
2. (Guest\_id,Location)
3. (Guest\_id, Cost)
4. (Emp\_id,Service\_type)
5. (Emp\_id,Location)
6. (Emp\_id , Cost)
7. (Service\_id,Service\_type)
8. (Service\_id,Location)
9. (Service\_id , Cost)

**3NF FORM: -**

* **Already in 2NF**
* **NO transitive Dependency**

In the above relation there are no transitive dependency relations so the given relation is in 3nf form.