SRS DOCUMENTATION FOR

Hotel Chain Management

GROUP NAME - P*

ARCHIT JAIN (2019101053)

PULKIT GUPTA (2019101078)

AADITYA SHARMA (2019113009)

===================================

OVERVIEW

Describing mini-world

The mini world chosen for this project is a HOTEL CHAIN.

This is a database for a hotel brand having multiple hotels. All information pertaining to the functioning of the hotel is stored in the database. The entire database model represents real time booking of rooms and events and also has information of recreational activities used by the guests.

Purpose

This database is made for hotel chain management . This database gives the details about the functioning of this chain . And allows its maintenance and modification of the details about hotels.

Users

The users of this database include hotel managing staff, their owners, their shareholders, even guests can also use it to see for availability of rooms and other hotel services .It can also be used by other investors who want to invest in this hotel chain.

What will the user do with the database

It will allow them to check availability of rooms when a visitor visits and once the visitor leaves the hotel staff (receptionist) could change the database accordingly. Gives information about different recreational activities provided by the hotel for extra services including spas,restaurants, entertainment , gaming, swimming pools etc.

Our database accounts for all the human interactions that take place with the hotels.And it allows operators and owners to streamline their administrative tasks while also increasing their bookings in both the short and long term.

Database Requirements

Entities

1) Hotel
2) Booking
3) Staff
4) Guest
5) Expenditure
6) Recreational Activity

Attributes of all Entities

Strong entity :-

a) Hotel

- Hotel ID : Primary key, INTEGER , NOT NULL
- Hotel Name : VARCHAR(50), NOT NULL
- Location : VARCHAR(50) , NOT NULL
- Contact number : INTEGER , NOT NULL
- Number of rooms available : INTEGER , NOT NULL

b) Booking (superclass)

- Booking Id : Primary key, INTEGER , NOT NULL
- Hotel Id : Foreign key, INTEGER , NOT NULL
- Email ID : VARCHAR(50), NOT NULL
- Phone Number : INTEGER , NOT NULL
- Payment mode (cash/online) : VARCHAR(50), NOT NULL
- Final Charges : FLOAT
- Status (confirmed/canceled) : VARCHAR(50)
  - Event booking (subclass1)
    - Hall number : VARCHAR (50) , NOT NULL
    - Number of Guest : INTEGER, NOT NULL
    - Organized by :VARCHAR (50) , NOT NULL
    - Datetime : VARCHAR (50) , NOT NULL
    - Duration : VARCHAR (50) , NOT NULL
    - Hall charges : FLOAT, NOT NULL
  - Room booking (subclass2)
    - Check in Date : VARCHAR (50) , NOT NULL
    - Check out Date : VARCHAR (50) , NOT NULL

- No. of members : INTEGER , NOT NULL
- Number of rooms : INTEGER , NOT NULL
- Room charges : FLOAT, NOT NULL

c) Staff (Superclass)

- Employee Id : Primary key, INTEGER , NOT NULL
- Hotel Id : Foreign key, INTEGER , NOT NULL
- Name : VARCHAR (50), NOT NULL
- Phone Number : INTEGER , NOT NULL
- Date of Birth : VARCHAR (50) , NOT NULL
- Address : VARCHAR(50), NOT NULL
- Salary : INTEGER , NOT NULL

d) Guest (Primary Key as **Guest id** = booking id + i + j **)**

- Guest Id* : Primary key, VARCHAR(100) , NOT NULL
- Booking id : Foriegn key ,INTEGER , NOT NULL
- Name : VARCHAR (50) , NOT NULL
- Age : INTEGER , NOT NULL
- Gender : VARCHAR (50) , NOT NULL

* *where i is i th room booked by guest and j is j th guest in i th room like if booking id = 123; no. of rooms booked 2; each has {1,3} members respectively, than guestid will be 12311, 12321, 12322 ,12323*

Weak entity :-

e) Expenditure

- Hotel Id : Foreign key , INTEGER , NOT NULL
- Date* : VARCHAR(50) , NOT NULL
- Income : FLOAT , NOT NULL
- Salary of staff : FLOAT , NOT NULL
- Extra expenses : FLOAT , NOT NULL
- Total profit : FLOAT , NOT NULL

* *date only includes month and year (MM-YYYY)*

f) Recreational Activity

- Hotel Id : Foreign key , INTEGER , NOT NULL
- Activity Name : partial key, VARCHAR(50) , NOT NULL
- Price per activity : FLOAT , NOT NULL
- Guest id : Foreign key , INTEGER , NOT NULL

Constraints

- Number of members in a room (0<i<=4)

- Number of rooms that can be booked over a booking id (0<i<=min (5, number of room available))
- Checkout date > check in date.
- Minimum number of staff members must be >=6

Relationships

There exists many relationships amongst the entities . For eg-

1. Manager MANAGES the (whole) staff forming a recursive relationship of cardinality 1:N. Manager (subclass in Staff) can manage minimum 0 and maximum n staff members whereas Staff Members can be managed by at most 1 manager. For Manager -> (0,N) and Staff -> (0,1)

2. Expenditure BELONGS_TO Hotel is an identifying relationship of cardinality 1:1 where Expenditure -> (1,1) and Hotel -> (1,1) here both expenditure and hotel has a total participation.

3. Staff WORKS_FOR Hotel is a binary relationship of cardinality N:1 where one staff member can work for only one hotel at a time but a hotel can have any number of staff members(at least 6) and represented as Staff -> (1,1) and Hotel -> (6,N)

4. Booking IN Hotel is a binary relationship of cardinality N:1 where a booking can be made for atmost one hotel at a time and a hotel can have any number of bookings and can also be represented as booking -> (0,1) and Hotel -> (0,N).

n > 3 Relationships (n=4)

- A recreational activity is performed by a guest under supervision of staff for which he has done booking.

  Recreational activity → Guest →Staff → Booking

  **Not an actual representation of relation in b/w them*

Subclass

- Booking entity is further divided into two subclasses

1) Room booking
2) Event booking

- A disjoint subclass of staff can be divided into following subclasses on the basis of type of job.

1) Receptionist
2) Room service
3) Cook
4) Security guard
5) Manager.
6) Gardener
7) Extra Services.

** Staff has a total participation .i.e. Each staff member must belong to at least one subclass of staff.*

Attribute type

Multivalued:-

- Contact number of the hotel can be more than one.
- Staff addresses can be more than one.

Derived:-

- Age of staff can be derived using Date of Birth and current date.
- Final charges attribute in Booking is addition of (i) room charges ((Room cost(K) + services used)*(check out date - check in date)) and (ii) Hall charges ((Duration)*(hourly cost(K))).
- Total profit is calculated as[income - salary of staff - extra expenses].

Composite:-

- Location of a hotel can be broken down into area , city and state.
- Staff name and guest name can be broken down into first name and last name.

Functional requirements

Insert :-

1. Insertion of new booking entries.
2. Insertion of new staff member.
3. Insertion of new guests.
4. Insertion of new recreational activity added.

Delete :-

1. Deletion of staff member (in case a staff member resigns).
2. Deletion of recreational activity (if it is dropped or removed from hotel) .

Update :-

1. Updating hotel contact number (in case it changes).
2. Update booking details.(both status and price)
3. Updating salary of staff.(if there is any increment).
4. Updating recreational activity attributes.

Retrieval :-

1. **Search** : Getting booking details by name of guest for which booking was made .
2. **Projection** : List of all hotel having available room > 10
3. **Aggregate** : Profit earned every Year by each hotel.
4. **Selection** : List of all staff member in a particular hotel
5. **Reports generated**:
   - Monthly financial report of a hotel.
   - Most popular activity in a hotel.

==

HERE WE MARK THE END OF OUR GROUP PROJECT PHASE#1

==