

Group Project - 1

Mini-World : Hotel Franchise

Introduction

This Mini world design handles the data of any Hotel Franchise that has many branches, including data of food, staff, raw materials and customers with timely deletion of unnecessary data. For simplicity let us consider a Hotel Franchise named Haveli with five branches as our Mini-World database.

Purpose

The Main purpose of this database is to make the data handling easy and efficient for any Hotel Franchise management and deal with their services productively.

Users

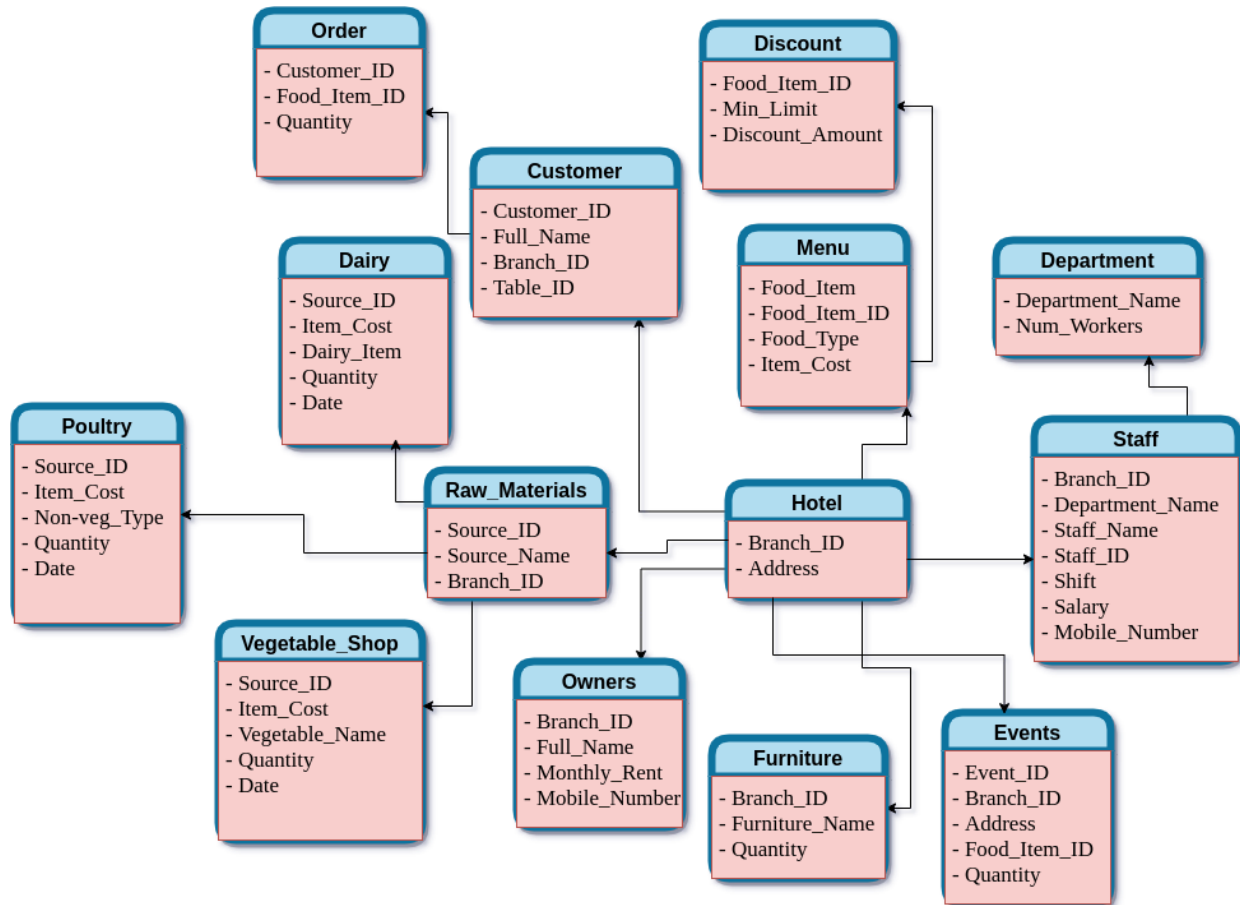
Users of the database:

- Administrators
- Technical staff
- Hotel Database Designers
- Financial Department

Applications

- To maintain a record of daily required raw materials, information of all working staff used by the hotel branches.
- To keep track of all the available food items for customer orders.
- To enable easy access to the Hotel services provided to customers

Database Model Diagram



Database Requirements

Entities

1. Hotel

Attributes:

- a. Branch_ID
 - I. Attribute-Type : Key Attribute
 - II. Data-Type: int (Takes values from 1 to 5)
- b. Address
 - I. Attribute-Type : Composite Attribute
 - II. Data-Type: int,int,string,int
 - III. Attributes:
 - Door_No
 - Street_No
 - Colony_Name
 - Pin-Code

2. Department

Attributes:

- a. Department_Name
 - I. Attribute-Type : Key Attribute
 - II. Data-Type: string(length ranges from 1 to 30 characters)
- b. Num_Workers
 - I. Attribute-Type: Simple Attribute
 - II. Data-Type: int (Minimum value is 1 and no bound on maximum)

3. Staff

Attributes:

- a. Branch_ID
 - I. Attribute-Type : Simple Attribute
 - II. Data-Type: int (Takes values from 1 to 5)
- b. Department_Name
 - I. Attribute-Type : Simple Attribute
 - II. Data-Type: string(length ranges from 1 to 30 characters)
- c. Staff_Name
 - I. Attribute-Type : Composite Attribute
 - II. Data-Type: int,int,string,int
 - III. Attributes:
 - First_Name
 - Last_Name
- d. Staff_ID
 - I. Attribute-Type: Key Attribute
 - II. Data-Type: Int (Takes values from 1)

e. Shift

- I. Attribute-Type: Simple Attribute
- II. Data-Type: string

f. Salary

- I. Attribute-Type: Simple Attribute
- II. Data-Type: Int (Takes values from 1000)

g. DOB

- I. Attribute-Type: Composite Attribute
- II. Data-Type: Int (Takes values from 1)
- III. Derived Attribute: Age
- IV. Attributes:
 - Day
 - Month
 - Year

h. Mobile_Number

- I. Attribute-Type: Multi valued Attribute
- II. Data-Type: string (number of characters 10)
- III. Attributes:
 - Mobile_Number_1
 - Mobile_Number_2...

4. Customer

Attributes:

a. Customer_ID

- I. Attribute-Type: Key Attribute
- II. Data-Type: Int (Takes values from 1)
- III. Derived Attribute: Bill

b. Customer_Name

- I. Attribute-Type: Composite Attribute
- II. Data-Type: String,String
- III. Attributes:
 - First_Name
 - Last_Name

c. Branch_ID

- I. Attribute-Type: Simple Attribute
- II. Data-Type: Int(Takes values from 1 to 5)

d. Table_ID

- I. Attribute-Type: Simple Attribute
- II. Data-Type: Int (Takes values from 1 to 5)

5. Order

Attributes:

- a. Customer_ID
 - I. Attribute-Type: Prime Attribute
 - II. Data-Type: int (Takes values from 1)
- b. Food_Item_ID
 - I. Attribute-Type: Prime Attribute
 - II. Data-Type: int (Takes values from 1 to 50)
- c. Quantity
 - I. Attribute-Type: Simple Attribute
 - II. Data-Type: float (Takes value from 0)

❖ Primary Key: Set of Customer_ID, Food_Item_ID

6. Menu

Attributes:

- a. Food_Item_ID
 - I. Attribute-Type: Primary Key
 - II. Data-Type: int (Takes values from 1 to 50)
- b. Food_Type
 - I. Attribute-Type: simple Attribute
 - II. Data-Type: string (length of string varies from 3 to 50)
- c. Food_Item
 - I. Attribute-Type: Alternate Key
 - II. Data-Type: String (length of string varies from 3 to 50)
- d. Item_Cost
 - I. Attribute-Type: simple attribute
 - II. Data-Type: Int (Takes values from 30)

7. Vegetable_Shop

Attributes:

- a. Source_ID
 - I. Attribute-Type: Simple Attribute
 - II. Data-Type: int (Takes values from 1 to 3)
- b. Vegetable_Name
 - I. Attribute-Type: Simple Attribute
 - II. Data-Type: string (length of string varies from 3 to 50)
- c. Item_Cost
 - I. Attribute-Type: Simple Attribute
 - II. Data-Type: int (Takes values from 10)
 - III. Derived Attribute: Bill
- d. Quantity
 - I. Attribute-Type: Simple Attribute

- II. Data-Type: Float(Takes values from 0)
- e. Date
 - I. Attribute-Type: Composite Attribute
 - II. Data-Type: int,int,int
 - III. Attributes:
 - Day
 - Month
 - Year

8. Poultry

Attributes:

- a. Source_ID
 - I. Attribute Type: Simple Attribute
 - II. Data Type: int (Takes values from 1 to 3)
- b. Non-veg_Item
 - I. Attribute-Type: Simple Attribute
 - II. Data-Type: String (length of string varies from 3 to 50)
- c. Item_Cost
 - I. Attribute-Type: Simple Attribute
 - II. Data-Type: int (Takes values from 100)
 - III. Derived Attribute: Bill
- d. Quantity
 - I. Attribute-Type: Simple Attribute
 - II. Data-Type: float(Takes values from 0)
- e. Date
 - I. Attribute-Type: Composed Attribute
 - II. Data-Type: int,int,int
 - III. Attributes:
 - Day
 - Month
 - Year

9. Dairy

Attributes:

- a. Source_ID
 - I. Attribute Type: Simple Attribute
 - II. Data Type: int (Takes values from 1 to 3)
- b. Dairy_Item
 - I. Attribute Type: Simple Attribute
 - II. Data Type: string (No.of characters can be from 1 to 30)
- c. Item_Cost
 - I. Attribute-Type: Simple Attribute
 - II. Data-Type: int (Takes values from 100)
 - III. Derived Attribute: Bill

- d. Quantity
 - I. Attribute Type: Simple Attribute
 - II. Data Type: Float (Takes values from 0)
- e. Date
 - I. Attribute-Type: Composite Attribute
 - II. Data-Type: int,int,int
 - III. Attributes:
 - Day
 - Month
 - Year

10. Furniture

Attributes:

- a. Branch_ID
 - I. Attribute Type: Simple Attribute
 - II. Data Type: int (Takes values from 1 to 5)
- b. Furniture_Name
 - I. Attribute Type: Simple Attribute
 - II. Data Type: String (No.of characters can range from 1 to 30)
- c. Quantity
 - I. Attribute Type: Simple Attribute
 - II. Data Type: int (Takes values from 1)

11. Owners

Attributes:

- a. Branch_ID
 - I. Attribute Type: Key Attribute
 - II. Data Type: int (Takes values from 1 to 5)
- b. Name
 - I. Attribute-Type: Composite Attribute
 - II. Data-Type: String,String
 - III. Attributes:
 - First_Name
 - Last_Name
- c. Monthly_Rent
 - I. Attribute-Type: Single valued attribute
 - II. Data-Type: int (Takes values from 1,00,000)
- d. Mobile_Number
 - I. Attribute-Type: Multi valued Attribute
 - II. Data-Type: string (number of characters 10)
 - III. Attributes:
 - Mobile_Number_1
 - Mobile_Number_2...

12. Events

Attributes:

- a. Branch_ID
 - I. Attribute Type: Simple Attribute
 - II. Data Type: int (Takes values from 1)
- b. Food_Item_ID
 - I. Attribute Type: Simple Attribute
 - II. Data Type: int (Takes values from 1)
- c. Quantity
 - I. Attribute Type: Simple Attribute
 - II. Data Type: int (Takes values from 1)
- d. Address
 - I. Attribute-Type : Composite Attribute
 - II. Data-Type: int,int,string,int
 - III. Attributes:
 - Door_No
 - Street_No
 - Colony_Name
 - Pin-Code

13. Discount

Attributes:

- a. Food_Item_ID
 - I. Attribute Type: Key Attribute
 - II. Data Type: int (Takes values from 1)
- b. Min_Cost
 - I. Attribute Type: Simple Attribute
 - II. Data Type: int (Takes values from 500)
- c. Discount_Amount
 - I. Attribute Type: Simple Attribute
 - II. Data Type: int (Takes values from 100)

14. Raw Materials

- a. Source_Name
 - I. Attribute Type: Simple Attribute
 - II. Data Type: String (No of Characters can range from 0 to 30)
- b. Source_ID
 - I. Attribute Type: Simple Attribute
 - II. Data Type: int (Takes values from 1 to 3)
- c. Branch_ID
 - I. Attribute Type: Key Attribute
 - II. Data Type: int (Takes values from 1 to 5)

Weak Entities:

1. **Furniture:** with respective to Hotel
2. **Vegetable_Shop:** with respective to raw-material
3. **Poultry:** with respective to raw-material
4. **Dairy:** with respective to raw-material
5. **Events:** with respective to Hotel

Subclasses:

1. **Raw Materials**
 - a. Dairy
 - b. Poultry
 - c. Vegetable_Shop

Relationships

N = 2 Relationships

1. **Relationship-1:** Organized
 - a. Degree = 2
 - b. Entities involved: Department, Staff
 - c. Cardinality ratios: M : N ($N \geq M$)
 - d. Statement: **Staff** are organized under their respective **departments**.
2. **Relationship-2:** Chose
 - a. Degree = 2
 - b. Entities involved: Discount, Menu
 - c. Cardinality ratios: M : N ($M \leq N$)
 - d. Statement: **Discount** is chosen for the food_item in **Menu**
3. **Relationship-3:** Owned
 - a. Degree = 2
 - b. Entities involved: Owners, Hotel
 - c. Cardinality ratios: 1 : 1
 - d. Statement: Every **hotel** building is owned by a **owner**

N = 3 Relationships

1. **Relationship-4:** Caterings
 - a. Degree = 3
 - b. Entities involved: Hotel,Events,order
 - c. Cardinality ratios: 1 : M : N ($N \geq M \geq 1$)
 - d. Statement: **Hotel** caterings for **orders** for **events**.

N = 4 Relationships

1. **Relationship-5:** Orders
 - a. Degree = 4
 - b. Entities involved: Customer,order,Menu,Hotel
 - c. Cardinality ratios: M:N:1:1
 - d. Statement: **Customer** *orders* an **order** from **Menu** in **Hotel**

N = 5 Relationships

1. **Relationship-6:** Offers
 - a. Degree = 5
 - b. Entities involved: Hotel,Discount,Customer,order,Menu
 - c. Cardinality ratios: 1:P:Q:1:R
 - d. Statement: **Hotel** offers **discount** on **order** from **Menu** to **Customer**

Functional Requirements

Modifications

1. **Insert:** Query to Insert new data to the database
Example:
 - a. Insert Panipuri which cost 50 per plate to the Menu of Haveli Hotels.
 - b. Insert the data of the new staff joined at Haveli Hotels with branch-ID 3.
 - c. Insert Veg-Spring rolls in the order of Customer-ID 9 of Haveli Hotels with Branch-ID as 5.

2. **Delete:** Query to delete particular data from database

Example:

- a. Delete Ice aroma from Menu of Haveli Hotels.
- b. Delete the ownership details of branch-ID 5.(To replace it with a new owner details)
- c. Delete Last 1 month information from the Dairies subclass of raw materials with Branch 1 of Haveli Hotels.

3. **Update:** Query to update particular data in database

Example:

- a. Update cost of Chicken dum biryani on the menu of Haveli Hotels.
- b. Update the last name of a staff with staff-ID 6 working at Haveli hotels with branch-ID 4.
- c. Update rent of branch ID 2 to Rs.1,20,000 per month for the owner.

Retrievals

1. **Selection:** Query to fetch data from database

Example:

- a. Retrieve starters from the menu of Haveli Hotels.
- b. Retrieve salary of Chef named Pallavi in Haveli Hotels with branch 2.
- c. Retrieve the number of tables in Haveli Hotels with Branch ID 1.

2. **Projection:** Query to enable the users to search the database by a particular attribute

Example:

- a. Names of all owners who rented Haveli hotels of cost greater than 1 Lakh.
- b. List all the food items with a cost greater than 200 per plate.
- c. List of dates with Non-veg supply more than 25 kgs from a Poultry of Haveli hotels with branch-ID 3.
- d. List all the staff with age greater than 60 years.

3. **Aggregate:** Query to derive group and subgroup data by analysis of a set of individual data entries.

Example:

- a. Branch of Haveli hotels paying highest rent.
- b. Maximum salary of a chef in Haveli Hotels with branch ID 1.

4. **Search:** Query to search partial text for entries in an entity, matching for subparts of the entries.

Example:

- a. Owners with Name “Bhargavi” renting the building to Haveli Hotels.
- b. Staff with last 3 digits of mobile number as 999 from Haveli Hotels.
- c. Most frequently ordered item at Havel hotels with branch-ID 3.

5. **Analysis:** Query of reports convey something about the relationship between entities.

Example:

- a. Average rent paid by the Haveli hotel group in a month.
- b. Average cost of starters in Menu from Haveli Hotels.
- c. Average number of orders in a week from “Madhapur” area for haveli hotels with branch-ID 2.

- By **TEAM**

❖ Kurukunda Bhargavi

❖ Pampulapati Pallavi

❖ Greeshma Amaraneni