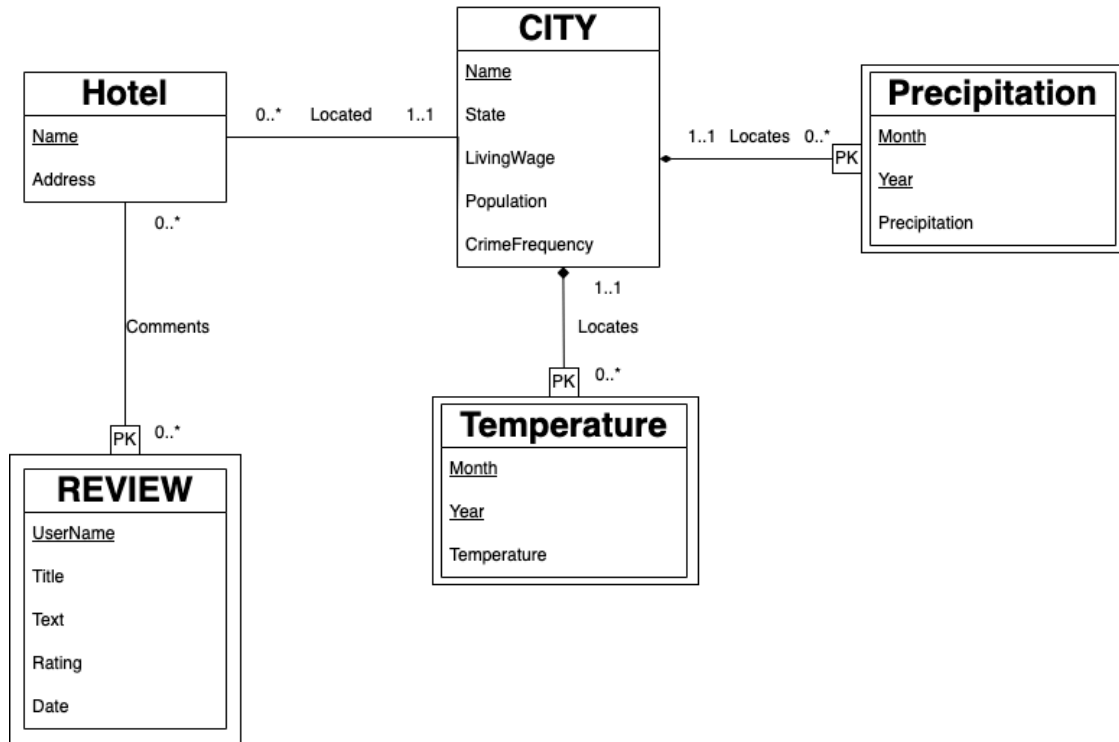


1. UML Diagram:



2. Assumptions:

The database stores information about **Hotels**, **Cities**, **Reviews**, **Temperature**, and **Precipitation**.

1. CITY

This is an entity regarding the locations, with 5 attributes about basic information.

(1) Name: The name of the city referred to geographic areas.

The primary key of the entity.

(2) State: The name of state referred to geographic areas

(3) LivingWage: The average of Wage referred to the city

(4) Population: The amount of people referred to the city

(5) CrimeFrequency: The frequency of crime referred to the city

Description:

- A **City** is uniquely identified by its Name. Other city attributes are State, LivingWage, Population, and CrimeFrequency.

2. HOTEL

This is an entity regarding the hotels with the address, with 2 attributes.

(1) Name: The name of the hotel given a name designation.

The primary key of the entity.

(2) Address: The location of the hotel referred to geographic areas.

Description:

- A **Hotel** is uniquely identified by its Name. It has Address as another attribute.

Relationship:

- A **City** may have multiple **Hotels** located within it, but a **Hotel** is located in only one **City**, which we denote as **1-many relationship**.

3. REVIEW

This is a weak entity regarding the comments about user experience, with 5 attributes.

- (1) **UserName**: The name of the user given a name designation.
The primary key of the entity.
- (2) **Title**: The title of the comment referred to the hotel.
- (3) **Text**: The content of the comment referred to the user experience.
- (4) **Rating**: The score of the user experience.
- (5) **Date**: The time stamp of the user experience.

Description:

- A **Review** is uniquely identified by its **UserName** and the hotel it comments on. Other review attributes are Title, Text, Date and Rating.

Relationship:

- A **Hotel** may have **Reviews** by multiple users, and one user can make **Reviews** on multiple **Hotels**, which we denote as **many-many relationship**.

4. TEMPERATURE

This is a weak entity regarding the average temperature in a city, with 3 attributes.

- (1) **Month**: The month of the average temperature. The primary key of the entity.
- (2) **Year**: The year of the average temperature. The primary key of the entity.
- (3) **Temperature**: The average temperature in a city with the time designation.

Description:

- The **Temperature** is uniquely identified by Month, Year and the city it locates at. It has Temperature as another attribute.

Relationship:

- A **Temperature** can only be located at a single **City**, but a **City** may have multiple **Temperature** according to the month and year, which we denote as **1-many relationship**.

5. PRECIPITATION

This is a weak entity regarding the average precipitation in a state, with 3 attributes.

- (1) **Month**: The month of the average precipitation. The primary key of the entity.
- (2) **Year**: The year of the average precipitation. The primary key of the entity.
- (3) **Temperature**: The average precipitation in a city with the time designation.

Description:

- The **Precipitation** is uniquely identified by Month, Year and the state it locates at. It has Precipitation as another attribute.

Relationship:

- A **Precipitation** can only be located at a single state, which belongs to the **City**, but a **City** may have multiple **Precipitation** according to the month and year, which we denote as **1-many relationship**.

3. Normalization: (BCNF vs 3NF)

We choose 3rd Normal Form (3NF) since:

- It can eliminate redundancy in the relational schema.
- It can preserve functional dependencies when decomposing.

Functional Dependency

[FD1] HotelName -> Address, CityName
[FD2] UserName, HotelName -> Title, Text, Rating, Date
[FD3] CityName -> State, LivingWage, Population, CrimeFrequency
[FD4] State, Month, Year -> Precipitation
[FD5] CityName, Month, Year -> Temperature

3NF

1. HotelName is a super-key for **Hotel**.
2. UserName and HotelName compose super-key for **Review**.
3. CityName is a super-key for **City**.
4. CityName, Month, Year compose super-key for **Precipitation**.
5. State, Month, Year compose super-key for **Temperature**.

Explanation for something modeled as an entity rather than an attribute

- Instead of viewing components of Review as attributes for Hotel, we construct an entity set for Review since it will be decomposed according to **FD1** and **FD2**.
- Instead of viewing components of City as attributes for Hotel, we construct an entity set for City since it will be decomposed according to **FD1** and **FD3**.
- Instead of viewing components of Precipitation as attributes for City, we construct an entity set for Precipitation since it will be decomposed according to **FD3** and **FD4**.
- Instead of viewing components of Temperature as attributes for City, we construct an entity set for Temperature since it will be decomposed according to **FD3** and **FD5**.

4. Relational Schema:

1. Hotel(

Name: VARCHAR(255) [PK],
Address: VARCHAR(255),
CityName: VARCHAR(255) [FK to City.Name]

)

2. Review(

UserName: VARCHAR(255) [PK],
Title: VARCHAR(255),
Text: VARCHAR(255)
Rating: INT,
Date: VARCHAR(255),
HotelName: VARCHAR(255) [PK] [FK to Hotel.Name]

)

3. City(

Name: VARCHAR(255) [PK],
State: VARCHAR(255),
LivingWage: REAL,
Population: INT,
CrimeFrequency: INT

)

4. Temperature(

Month: INT [PK],
Year: INT [PK],
Temperature: REAL,
CityName: VARCHAR(255) [PK] [FK to City.Name]

)

5. Precipitation(

Month: INT [PK],
Year: INT [PK],
Precipitation: REAL,
CityState: VARCHAR(255) [PK] [FK to City.State]

)