An Analysis on Women’s Safety while travelling

Project Report

Group 10

Stephy Romichan

Meghana Sreenivasa Rao

(575)-842-9980

(857)-421-9496

romichan.s@northeastern.edu

rao.meg@northeastern.edu

Percentage of Effort Contributed by Student 1: 50

Percentage of Effort Contributed by Student 2: 50

Signature of Student 1: Stephy Romichan

Signature of Student 2: Meghana Sreenivasa Rao

Submission Date: 12/08/2024

Executive Summary

The primary objective of this study was to design and implement a relational database focused on enhancing women's safety while traveling, ensuring it is ready for practical use in addressing real-world challenges. The project aimed to analyze data related to traveler demographics, safety ratings of destinations, crime statistics, and traveler feedback to identify risks and patterns. The database serves as a tool for understanding the safety concerns faced by women, especially solo travelers, and equips them with insights to plan their trips more confidently.

The relational database was built using MySQL, with a total of 21 tables connected through primary and foreign keys to ensure consistency and accuracy. It was designed to handle various types of data, including traveler profiles, destination safety ratings, and reported incidents, allowing for comprehensive analysis. To explore its flexibility in handling unstructured data, the database was also implemented in MongoDB, where the tables were imported, and queries were executed using the shell and Compass tools. This step tested the feasibility of using NoSQL for storing and querying the data efficiently.

The database's potential was demonstrated through integration with Python for advanced analytics and data visualization. This enabled the identification of travel behavior patterns, risky destinations, and safety measures available at different locations. For example, it could help highlight the safest modes of travel, identify regions with effective women-focused services, and provide emergency support details. Future work will focus on implementing data governance practices to improve its reliability, making it a robust tool for addressing women's safety concerns while traveling.

Introduction

Traveling for women, while exciting, often comes with unique safety challenges. Issues such as navigating public transport, selecting safe accommodations, and dealing with regional risks like theft or harassment can create significant concerns, especially for solo female travelers. Despite growing awareness, information on these risks is often scattered across traveler reviews, surveys, and reports, making it difficult for women to plan trips confidently. This project seeks to address these gaps by building a centralized system to collect and analyze data about women's travel experiences, with a specific focus on safety insights.

The aim is to use data-driven analysis to empower women with actionable information. This includes identifying safe destinations, understanding patterns of crime or harassment, and highlighting the availability of women-focused services like female-only transport or dedicated helplines. Additionally, the project integrates feedback from travelers to reflect real-world concerns and experiences, ensuring that the insights are both relevant and practical for trip planning.

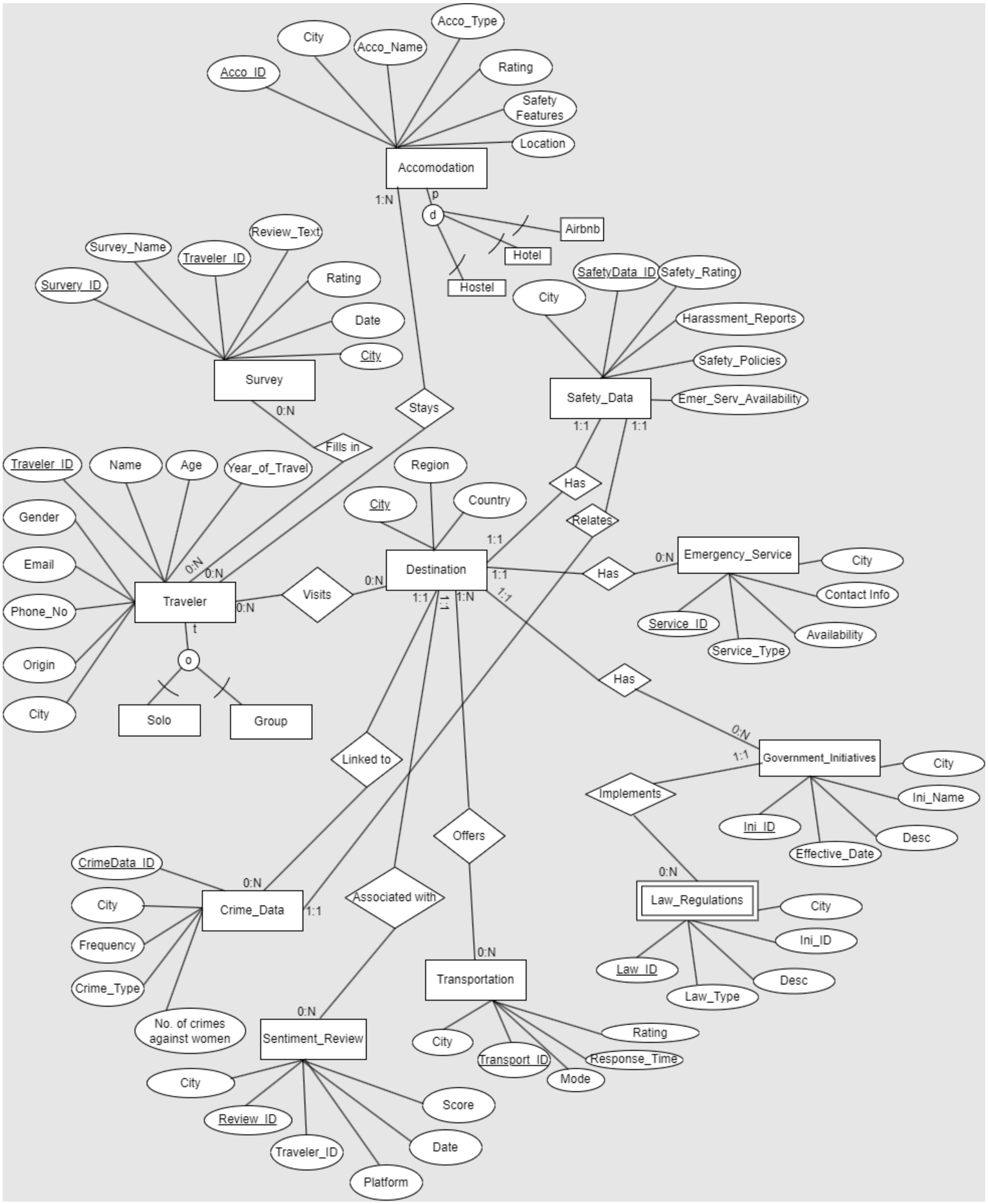
If data from web scraping and other real-world sources were used to populate the database, certain aspects must be carefully addressed to ensure the platform's effectiveness and reliability. A few key factors considered while developing this platform are:

1. **Data Accuracy:** Ensuring reliable data collection and verification through consistent cross-referencing and feedback from trusted sources like reviews, surveys, and safety reports.
2. **User Privacy:** Since sensitive information such as traveler demographics and personal details are involved, strict privacy measures are applied to protect user data.
3. **Real-Time Updates:** Incorporating frequent updates to safety ratings, crime statistics, and traveler reviews ensures the database remains current and relevant for users.
4. **Community Insights:** Introducing a feature for users to share feedback and insights about their travel experiences helps enhance the database's relevance and usability over time.

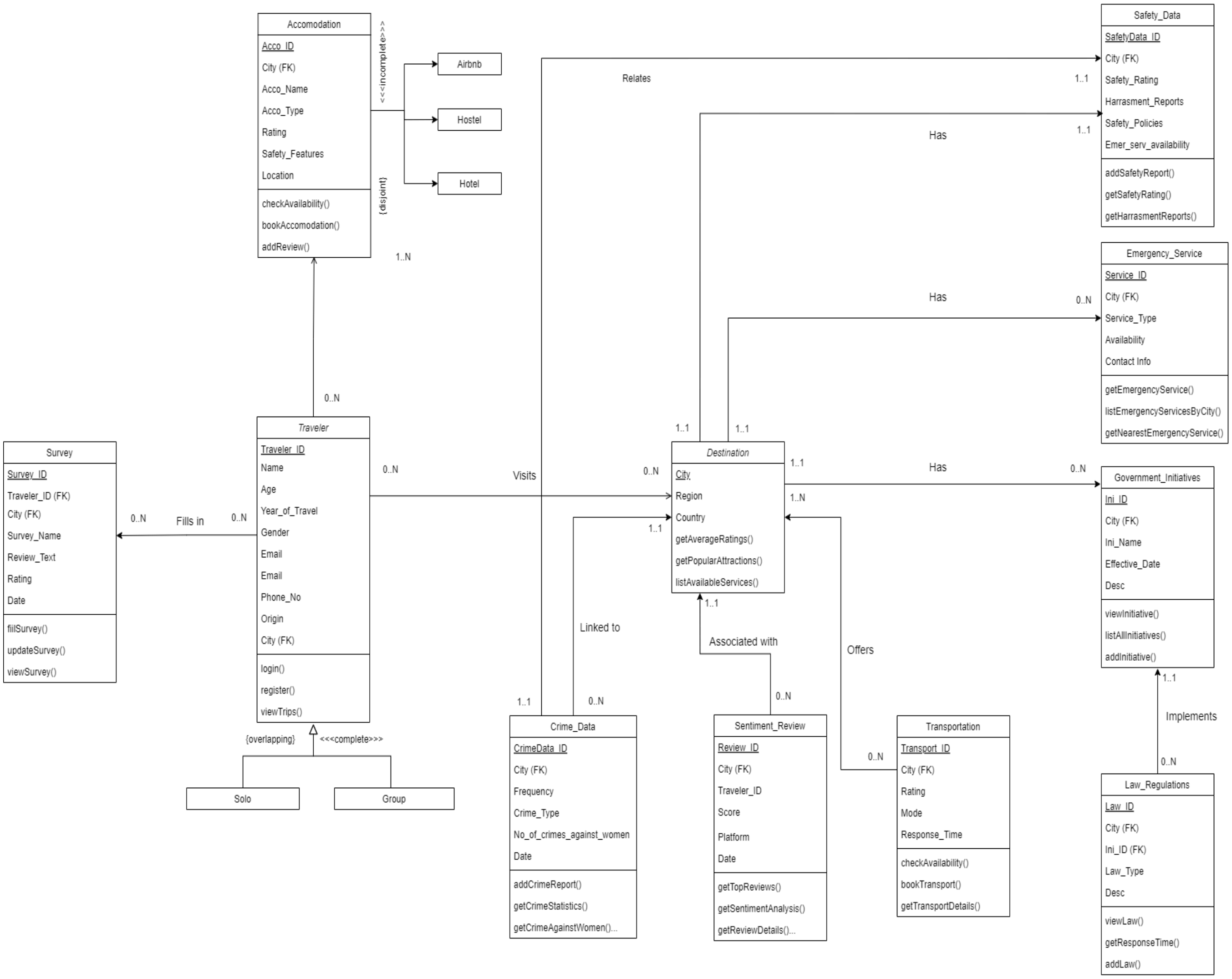
By combining these factors, the project aims to deliver not just a data-driven tool but a comprehensive safety platform that evolves with user needs and global travel trends.

II. Conceptual Data Modeling

1. EER Diagram:



2. UML Diagram:



III. Mapping Conceptual Model to Relational Model

Note: Primary Key is underlined, Foreign Key is in Italics.

Traveler (Traveler_ID, Name, Age, YearOfTravel, Gender, Email, Phone_No, Origin, *City*)

- Foreign key City refers to City from Destination; NULL NOT ALLOWED

Solo (*Traveler_ID*)

- Foreign key Traveler_ID refers to Traveler_ID from Traveler; NULL NOT ALLOWED

Group (*Traveler_ID*)

- Foreign key Traveler_ID refers to Traveler_ID from Traveler; NULL NOT ALLOWED

Survey (Survey_ID, Survey_Name, *Traveler_ID*, Review_Text, Rating, Date, *City*)

- Foreign key Traveler_ID refers to Traveler_ID from Traveler; NULL NOT ALLOWED
- Foreign key City refers to City from Destination; NULL NOT ALLOWED

Accommodation (Acco_ID, *City*, Acco-Name, Acco_Type, Rating, Safety_Features, Location)

- Foreign key City refers to City from Destination; NULL NOT ALLOWED

Hostel (*Acco_ID,* NoOfBeds)

- Foreign key Acco_ID refers to Acco_ID from Accomodation; NULL NOT ALLOWED

Hotel (*Acco_ID*)

- Foreign key Acco_ID refers to Acco_ID from Accomodation; NULL NOT ALLOWED

Airbnb (*Acco_ID*)

- Foreign key Acco_ID refers to Acco_ID from Accomodation; NULL NOT ALLOWED

SafetyData (SafetyData_ID, *City*, Safety_Rating, Harassment_Reports, Safety_Policies, Emer_Serv_Availability)

- Foreign key City refers to City from Destination; NULL NOT ALLOWED

Emergency Service (Service_ID, Service_Type, Availability, Contact_Info, *City*)

- Foreign key City refers to City from Destination; NULL NOT ALLOWED

Government_Initiatives (Ini_ID, Effective_Date, Desc, Ini_Name, *City*)

- Foreign key City refers to City from Destination; NULL NOT ALLOWED

Transportation (Transport_ID, *City*, Mode, Response_Time, Rating)

- Foreign key City refers to City from Destination; NULL NOT ALLOWED

Sentiment_Review (Review_ID, *City*, *Traveler_ID*, Platform, Date, Score)

- Foreign key City refers to City from Destination; NULL NOT ALLOWED
- Foreign key Traveler_ID refers to Traveler_ID from Traveler; NULL NOT ALLOWED

Crime_Data (CrimeData_ID, *City*, Frequency, Crime_Type, NoOfCrimesAgainstWomen)

- Foreign key City refers to City from Destination; NULL NOT ALLOWED

Destination (City, Region, Country)

TakesSurvey (*Traveler_ID, Survey_ID*)

- Foreign key Traveler_ID refers to Traveler_ID from Traveler; NULL NOT ALLOWED
- Foreign key Survey_ID refers to Survey_ID from Survey; NULL NOT ALLOWED

StaysIn (*Traveler_ID, Acco_ID*)

- Foreign key Traveler_ID refers to Traveler_ID from Traveler; NULL NOT ALLOWED
- Foreign key Acco_ID refers to Acco_ID from Accomodation; NULL NOT ALLOWED

Transport_Offered (*City, Transport_ID*)

- Foreign key City refers to City from Destination; NULL NOT ALLOWED
- Foreign key Transport_ID refers to Transport_ID from Transportation; NULL NOT ALLOWED

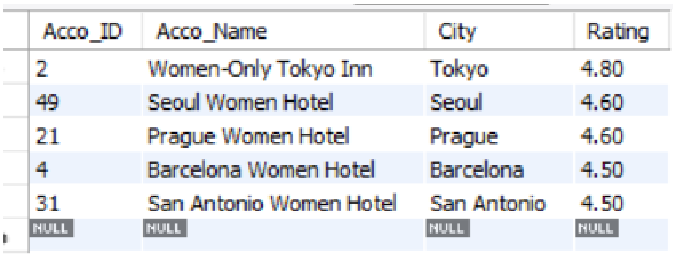
IV. Implementation of Relation Model via MySQL and NoSQL

MySQL Implementation:

The database was created in MySQL and the following queries were performed:

Query 1: Simple Query - Retrieve the top 5 highest-rated accommodations of type "Hotel" across all cities, ordered by rating

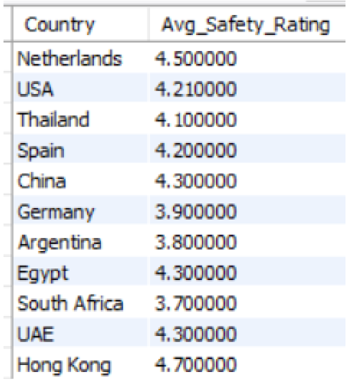
```
SELECT Acco_ID, Acco_Name, City, Rating
FROM accommodation
WHERE Acco_Type = 'Hotel'
ORDER BY Rating DESC
LIMIT 5;
```



Acco_ID	Acco_Name	City	Rating
2	Women-Only Tokyo Inn	Tokyo	4.80
49	Seoul Women Hotel	Seoul	4.60
21	Prague Women Hotel	Prague	4.60
4	Barcelona Women Hotel	Barcelona	4.50
31	San Antonio Women Hotel	San Antonio	4.50
NULL	NULL	NULL	NULL

Query 2: Aggregate Query - Find the average safety rating of all cities grouped by country

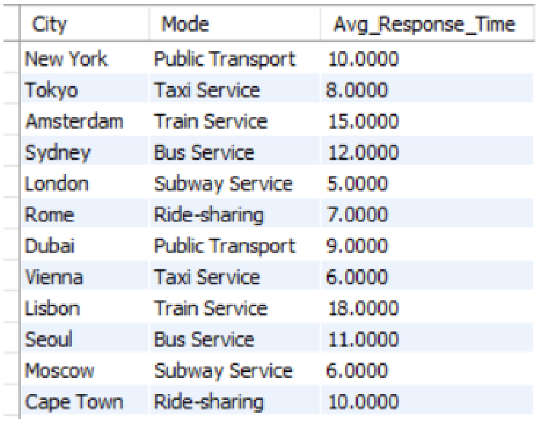
```
SELECT d.Country, AVG(sd.Safety_Rating) AS
Avg_Safety_Rating
FROM Destination d
JOIN SafetyData sd ON d.City = sd.City
GROUP BY d.Country;
```



Country	Avg_Safety_Rating
Netherlands	4.500000
USA	4.210000
Thailand	4.100000
Spain	4.200000
China	4.300000
Germany	3.900000
Argentina	3.800000
Egypt	4.300000
South Africa	3.700000
UAE	4.300000
Hong Kong	4.700000

Query 3: Inner Join with Aggregates - List all cities along with their transportation modes and the average response time for each mode

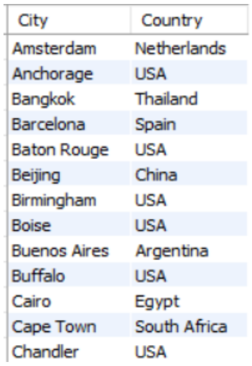
```
SELECT t.City, t.Mode, AVG(t.Response_Time) AS
Avg_Response_Time
FROM Transportation t
JOIN Destination d ON t.City = d.City
GROUP BY t.City, t.Mode;
```



City	Mode	Avg_Response_Time
New York	Public Transport	10.0000
Tokyo	Taxi Service	8.0000
Amsterdam	Train Service	15.0000
Sydney	Bus Service	12.0000
London	Subway Service	5.0000
Rome	Ride-sharing	7.0000
Dubai	Public Transport	9.0000
Vienna	Taxi Service	6.0000
Lisbon	Train Service	18.0000
Seoul	Bus Service	11.0000
Moscow	Subway Service	6.0000
Cape Town	Ride-sharing	10.0000

Query 4: Outer Join with Conditions - Find cities without any accommodations rated above 4.5

SELECT d.City, d.Country
FROM destination d
LEFT OUTER JOIN accommodation a ON d.City = a.City AND a.Rating
4.5
WHERE a.Acco_ID IS NULL;

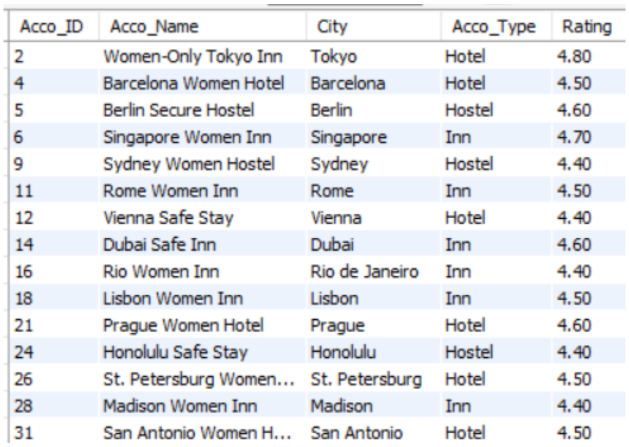


City	Country
Amsterdam	Netherlands
Anchorage	USA
Bangkok	Thailand
Barcelona	Spain
Baton Rouge	USA
Beijing	China
Birmingham	USA
Boise	USA
Buenos Aires	Argentina
Buffalo	USA
Cairo	Egypt
Cape Town	South Africa
Chandler	USA

>

Query 5: Nested Query with Aggregates - Fetch accommodations whose rating is above the average rating of their type (e.g., Hostel, Hotel)

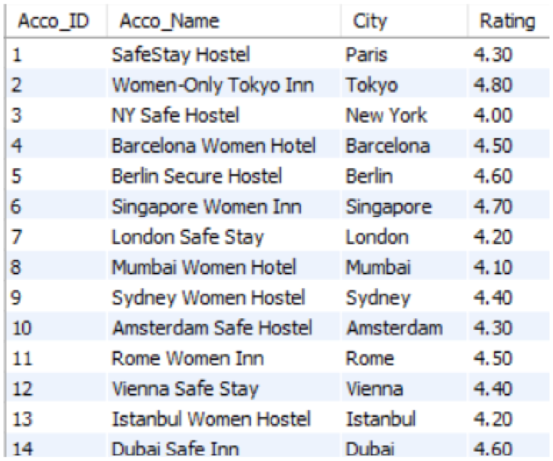
```
SELECT Acco_ID, Acco_Name, City,
Acco_Type, Rating
FROM accommodation
WHERE Rating > (SELECT AVG(Rating)
        FROM accommodation
        WHERE Acco_Type =
accommodation.Acco_Type);
```



Acco_ID	Acco_Name	City	Acco_Type	Rating
2	Women-Only Tokyo Inn	Tokyo	Hotel	4.80
4	Barcelona Women Hotel	Barcelona	Hotel	4.50
5	Berlin Secure Hostel	Berlin	Hostel	4.60
6	Singapore Women Inn	Singapore	Inn	4.70
9	Sydney Women Hostel	Sydney	Hostel	4.40
11	Rome Women Inn	Rome	Inn	4.50
12	Vienna Safe Stay	Vienna	Hotel	4.40
14	Dubai Safe Inn	Dubai	Inn	4.60
16	Rio Women Inn	Rio de Janeiro	Inn	4.40
18	Lisbon Women Inn	Lisbon	Inn	4.50
21	Prague Women Hotel	Prague	Hotel	4.60
24	Honolulu Safe Stay	Honolulu	Hostel	4.40
26	St. Petersburg Women...	St. Petersburg	Hotel	4.50
28	Madison Women Inn	Madison	Inn	4.40
31	San Antonio Women H...	San Antonio	Hotel	4.50

Query 6: Correlated Query with Ranking - Retrieve accommodations that have the highest rating in their respective cities

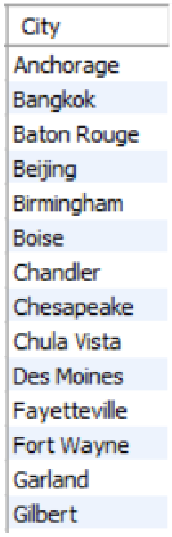
```
SELECT a1.Acco_ID, a1.Acco_Name, a1.City,
a1.Rating
FROM accommodation a1
WHERE a1.Rating = (SELECT MAX(a2.Rating)
          FROM accommodation a2
          WHERE a2.City = a1.City);
```



Acco_ID	Acco_Name	City	Rating
1	SafeStay Hostel	Paris	4.30
2	Women-Only Tokyo Inn	Tokyo	4.80
3	NY Safe Hostel	New York	4.00
4	Barcelona Women Hotel	Barcelona	4.50
5	Berlin Secure Hostel	Berlin	4.60
6	Singapore Women Inn	Singapore	4.70
7	London Safe Stay	London	4.20
8	Mumbai Women Hotel	Mumbai	4.10
9	Sydney Women Hostel	Sydney	4.40
10	Amsterdam Safe Hostel	Amsterdam	4.30
11	Rome Women Inn	Rome	4.50
12	Vienna Safe Stay	Vienna	4.40
13	Istanbul Women Hostel	Istanbul	4.20
14	Dubai Safe Inn	Dubai	4.60

Query 7: Comparison with EXISTS and NOT EXISTS - List cities where no crimes against women were reported

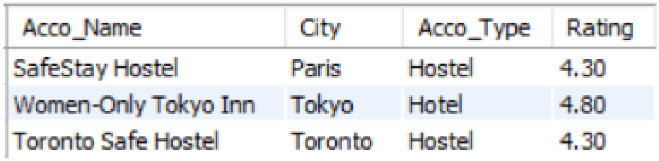
```
SELECT d.City
FROM Destination d
WHERE NOT EXISTS (
   SELECT 1
   FROM Crime_Data cd
   WHERE cd.City = d.City AND
cd.NoOfCrimesAgainstWomen > 0
);
```



City
Anchorage
Bangkok
Baton Rouge
Beijing
Birmingham
Boise
Chandler
Chesapeake
Chula Vista
Des Moines
Fayetteville
Fort Wayne
Garland
Gilbert

Query 8: Set Operation (Union with Additional Filters) - List all accommodations in cities where the country is "France" or the city starts with "T"

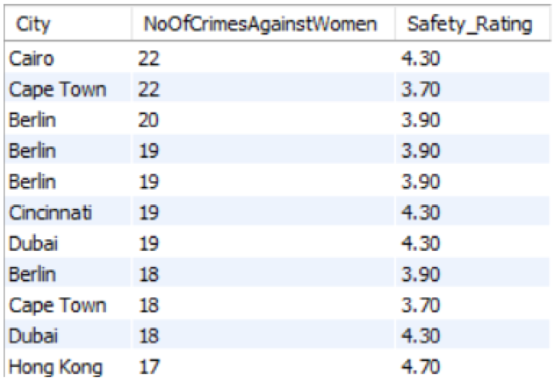
```
SELECT Acco_Name, City, Acco_Type,
Rating
FROM accommodation
WHERE City IN (SELECT City FROM
destination WHERE Country = 'France')
UNION
SELECT Acco_Name, City, Acco_Type, Rating
FROM accommodation
WHERE City LIKE 'T%';
```



Acco_Name	City	Acco_Type	Rating
SafeStay Hostel	Paris	Hostel	4.30
Women-Only Tokyo Inn	Tokyo	Hotel	4.80
Toronto Safe Hostel	Toronto	Hostel	4.30

Query 9: Subqueries in SELECT and FROM - Rank cities by the frequency of crimes against women, along with the safety rating

```
SELECT cd.City, cd.NoOfCrimesAgainstWomen,
sd.Safety_Rating
FROM Crime_Data cd
JOIN SafetyData sd ON cd.City = sd.City
WHERE cd.City IN (
   SELECT City
   FROM Crime_Data
   WHERE NoOfCrimesAgainstWomen > 10
)
ORDER BY cd.NoOfCrimesAgainstWomen DESC;
```



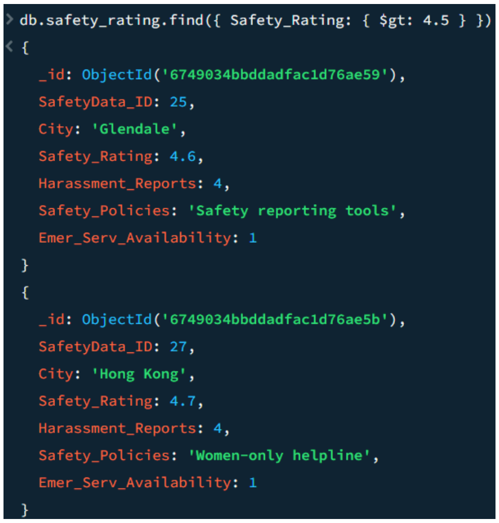
City	NoOfCrimesAgainstWomen	Safety_Rating
Cairo	22	4.30
Cape Town	22	3.70
Berlin	20	3.90
Berlin	19	3.90
Berlin	19	3.90
Cincinnati	19	4.30
Dubai	19	4.30
Berlin	18	3.90
Cape Town	18	3.70
Dubai	18	4.30
Hong Kong	17	4.70

NoSQL Implementation:

The database was imported and created in MongoDB, and collections were created for all the tables in this project. We executed simple queries on both MongoDB Shell and Compass.

Query 1: To find the cities where the safety rating is greater than 4.5

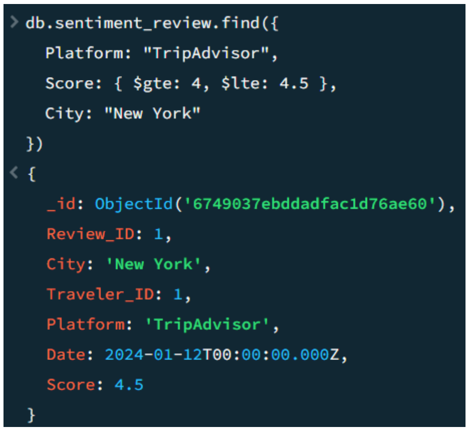
db.safety_rating.find({ Safety_Rating: { $gt: 4.5 } })



```
> db.safety_rating.find({ Safety_Rating: { $gt: 4.5 } })
< {
    _id: ObjectId('6749034bbddadfac1d76ae59'),
    SafetyData_ID: 25,
    City: 'Glendale',
    Safety_Rating: 4.6,
    Harassment_Reports: 4,
    Safety_Policies: 'Safety reporting tools',
    Emer_Serv_Availability: 1
  }
  {
    _id: ObjectId('6749034bbddadfac1d76ae5b'),
    SafetyData_ID: 27,
    City: 'Hong Kong',
    Safety_Rating: 4.7,
    Harassment_Reports: 4,
    Safety_Policies: 'Women-only helpline',
    Emer_Serv_Availability: 1
  }
```

Query 2: To find all reviews from platform TripAdvisor where the score is between 4 and 4.5 and the review was written about city New York, and display the Platform, Score, and City fields

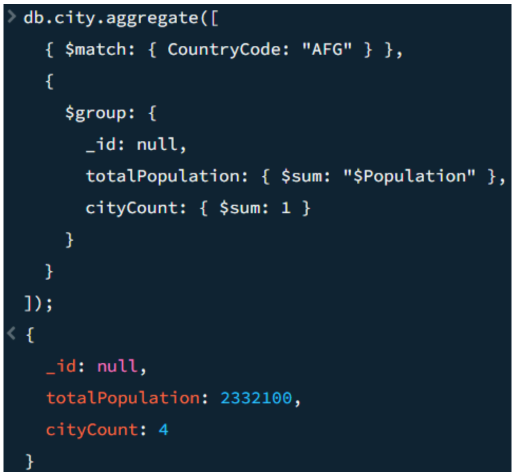
```
db.sentiment_review.find({
  Platform: "TripAdvisor",
  Score: { $gte: 4, $lte: 4.5 },
  City: "New York"
})
```



```
> db.sentiment_review.find({
    Platform: "TripAdvisor",
    Score: { $gte: 4, $lte: 4.5 },
    City: "New York"
  })
< {
    _id: ObjectId('6749037ebddadfac1d76ae60'),
    Review_ID: 1,
    City: 'New York',
    Traveler_ID: 1,
    Platform: 'TripAdvisor',
    Date: 2024-01-12T00:00:00.000Z,
    Score: 4.5
  }
```

Query 3: To find the total population of all cities in the country with CountryCode "AFG", and return the total population and the number of cities in that country

```
db.city.aggregate([
  { $match: { CountryCode: "AFG" } },
  {
    $group: {
      _id: null,
      totalPopulation: { $sum: "$Population" },
      cityCount: { $sum: 1 }
    }
  }
]);
```

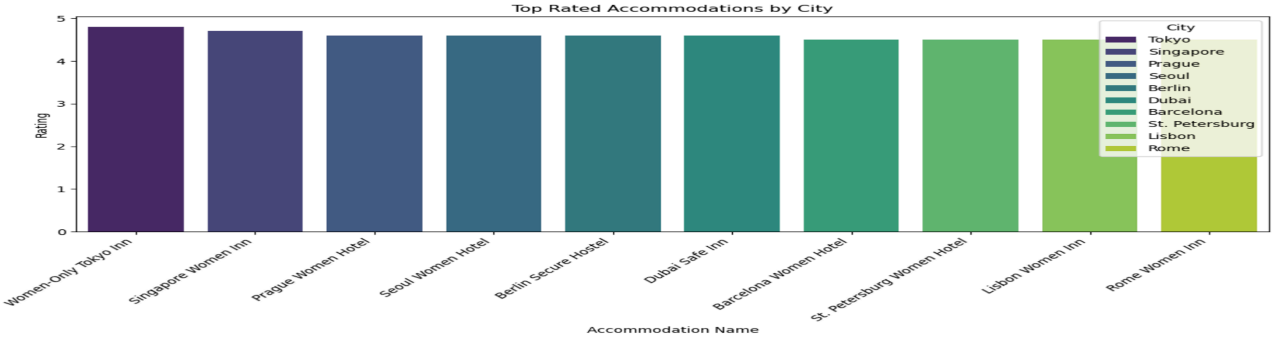


```
> db.city.aggregate([
    { $match: { CountryCode: "AFG" } },
    {
      $group: {
        _id: null,
        totalPopulation: { $sum: "$Population" },
        cityCount: { $sum: 1 }
      }
    }
  ]);
< {
    _id: null,
    totalPopulation: 2332100,
    cityCount: 4
  }
```

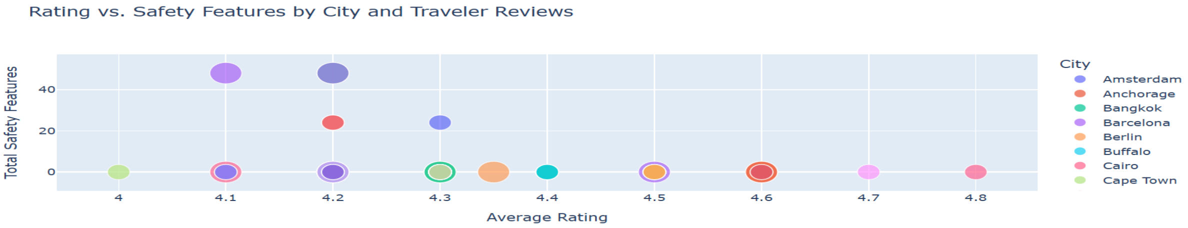
V. Database Access via Python

The database is accessed using Python and visualization of analyzed data is shown below, The database connection is established leveraging **`pymysql`**, **`SQLAlchemy`**, and **SQL Magic**. SQL Magic is used within Jupyter to execute queries, with **`SQLAlchemy`** managing database connections. Retrieved data is converted into a DataFrame using **`pandas`** and visualized using both **`matplotlib`** and **`plotly`** for dynamic and interactive analytical insights.

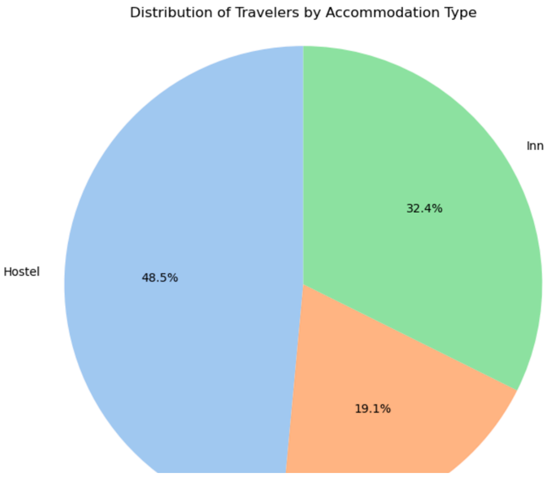
1. Top Rated Accommodations by City



2. Rating vs Safety Features by City and Traveler Reviews



3. Distribution of travelers by accommodation Type



VII. Summary and Recommendation

The **Women’s Travel Safety database**, designed on MySQL, is a robust, industry-ready relational database that can be implemented to address critical safety challenges in the tourism industry, especially for female travelers. This project emphasizes the need for structured and reliable data to better understand the safety landscape, making it invaluable for travel companies, policymakers, and individual travelers.

The database provides an extensive foundation for **data-driven insights** through demographic, safety, sentiment, and transportation data, a small subset of which is demonstrated in this report. We used Python for preprocessing and libraries like Plotly, Pandas and Matplotlib for visualization to analyze traveler safety trends ensuring the database can integrate seamlessly with practical applications for end-users.

Key Recommendations:

1. **Automated Data Validation for Quality Assurance**:
   Implement automated validation mechanisms for traveler data, such as email format verification and phone number standardization, to minimize entry errors. Utilizing government APIs for real-time verification of traveler details, such as passport numbers or national IDs, can improve the database's accuracy and reliability.
2. **Enhance Sentiment Analysis Integration**:
   Integrate advanced Natural Language Processing (NLP) tools to analyze traveler reviews and feedback. This enhancement can provide deeper insights into the sentiment of female travelers regarding safety and highlight specific areas of concern, such as transportation, accommodation, or public spaces.
3. **Integration with Machine Learning Models**: Use safety ratings, traveler reviews, and crime statistics as features for predictive modeling to assess travel risks.
4. **Future Enhancements**: Incorporate real-time data feeds, such as local crime alerts or updated government safety initiatives, to provide dynamic recommendations for travelers.

The **Women’s Travel Safety database** offers substantial advantages in understanding and addressing safety concerns, ensuring women travelers are better informed, and equipping stakeholders with actionable insights. While exploring NoSQL alternatives adds value in certain contexts, the relational model remains optimal for current applications in women’s safety during travel.