Big Data Storage Project

Airbnb

Group 3

Victoriya Mokhovikova 20230796

Vladislav Botnev 20230795

Yan Sidoryk 20222004

Christian Deluca 20241264

Nikolaos Kanellopoulos 20242109

Description

This project focused on exploring efficient storage and analysis of large-scale data using MongoDB, based on an extensive Airbnb dataset. Our goal was to understand how data structuring, validation, and querying can support meaningful insights. Through the process, we addressed challenges like data imbalance and storage limitations, and gained experience in organizing semi-structured data for analytical purposes. The project demonstrated how big data tools can help uncover trends in pricing, user behavior, and listing performance—insights that are essential for data-driven decision-making in platforms like Airbnb.

Point A

For this project, we chose to work with a dataset about short-term Airbnb rental properties in Lisbon. There is rich information in the dataset about listings (rented houses or apartments), neighborhoods in the city and reviews from guests who have stayed at the properties.
We chose this topic since Airbnb is very commonly used and contains a great amount of interesting data. There are diverse types of properties, hosts with different styles and a wide diversity in user experiences. This makes the data valuable and shows what could realistically be stored and analyzed in a modern database system.
MongoDB was selected in this project due to its high flexibility when dealing with dynamic and heterogeneous data structures. For example, listings can vary greatly: some include rich amenities and detailed descriptions, others have many reviews or none at all and the hosts may provide different levels of personal information. MongoDB makes it easy to handle this variability because it doesn't require a fixed schema.
Another reason we selected this dataset is that it gives us the opportunity to explore various aspects of data modeling, such as managing user reviews and examining prices, availability and location details.
We can also gain useful insights using queries and aggregations, for example by finding the most popular neighborhoods, highlighting the best-rated listings, and analyzing how well hosts perform.
Even though MongoDB is very flexible, it still requires careful planning. This is especially important when linking data across collections, since it does not support traditional joins like relational databases do. This can be restrictive in some cases, but it’s a fair trade-off considering how well MongoDB handles data that doesn’t follow a strict format.

Additionally, individuals considering hosting on Airbnb can gain valuable insights by analyzing this dataset. What is the most profitable district to buy apartments for further Airbnb hosting, what are the best number of bedrooms, and what is the most popular number of beds? To research this type of data is crucial if someone wants to invest in real estate and rent it out.

Overall, this project offers a great chance to see how MongoDB performs in practice when working with a real-world dataset. It helps us take advantage of its main features, like handling nested data and performing queries efficiently, while also giving us experience in organizing and modeling data in a more adaptable way.

Design decisions

1) We took the dataset from this website. There are long and short options for the dataset. We chose a bigger one to have more data. Because the space in the cluster is not so big, we were forced to cut some data. We cut reviews.
2) The next step was to upload data in the right format. MongoDB provides the data types associated with the column. Based on this information, we chose the most appropriate type.
3) We have an issue with the disproportion of the datasets. The listing has a lot of columns, otherwise others have fewer. We, with our team, decided to make one more table with hosts. Other columns are mostly numerical and describe specific listings.
4) Then we implemented validation rules for the following: date, location (latitude and longitude), minimum and maximum values for certain numeric columns, and the room_type field, which is required and must be one of the predefined enum values.
5) The queries: Top 10 most profitable listings, Top 5 districts by average price of the apartment, Top 5 districts by number og listings, Top 5 districts by average annual revenue, Top 5 districts by average number of bedrooms, The average number of bedrooms in one district, Average revenue for superhost VS average revenue for host, Average availability(for 60 and 90 days) for price chunk, Distribution of the number of beds, First 10 hosts on the platform, Top 5 listings by reviews, The worst 5 listings by reviews. We wrote queries to understand the most attractive place and configuration to start being the host. Availability, revenue, prices help to compare the districts and chose the best for your goals. When we see the reviews we can understand what people like. For Airbnb it is also important to analyze the data. Understanding prices, seasonal trends, and demand fluctuations helps hosts and Airbnb set competitive and dynamic pricing strategies. Analyzing booking patterns and popular amenities helps improve the platform's recommendations and user experience. Statistical anomalies in listings can signal fraudulent activity or quality issues, allowing for proactive intervention. Also, we compare a host and a superhost and showed the oldest and the richest hosts.
6) We wrote 2 queries witt and without indexes. The results are impressive. *Performance I:* Without index: COLLSCAN, 24,264 docs read, **41 ms**; With index: IXSCAN, 5,885 docs read, **20 ms**. *Performance II:* Without index: COLLSCAN, 24,264 docs read, **35 ms**; With index: IXSCAN, 2,518 keys read, **3 ms**
7) The aggregations covered key aspects of Airbnb listings to understand market trends and performance. Neighborhood price analysis looked at prices, total listings, and bedrooms, sorted by average price. Host performance identified the top 20 hosts with multiple listings based on ratings, reviews, and pricing. Neighborhood quality was ranked using cleanliness, location, and value scores. Stay duration analysis grouped listings by minimum nights and calculated average price, occupancy, and ratings. Property type analysis compared pricing and ratings across different property and room types. Finally, amenities impact analysis measured how common each amenity is and its effect on price and ratings.