Data Analytics

Tourism in Europe’s Capitals

Beds & Budgets: Mapping Airbnb Supply and Urban Travel Demand

Maïlys JAFFRET

September, 2025

Table of content

1- Introduction

Tourism is a key driver of growth for European cities, but it is also very competitive. To stay attractive, destinations need to balance affordability, accessibility, and quality of services. In recent years, platforms like Airbnb have reshaped the accommodation market. They provide new opportunities for travelers but also raise challenges for local communities, from housing pressure to regulation. At the same time, factors such as the cost of living and the number of international visitors play an important role in shaping travel demand and a city's global appeal. Looking at these dimensions together offers a more complete view of how tourism systems function today.

Project Goal:

The goal of this project is to analyze tourism patterns using multiple open data sources. Eurostat and Wikipedia are used to track international arrivals, Numbeo provides data on cost of living, and Airbnb listings give insights into the short-term rental supply. All datasets are integrated into a relational database to connect demand (arrivals), affordability (cost of living), and supply (Airbnb). Using SQL, I extract key insights, build a small Flask API to serve the data, and design interactive dashboards to explore and visualize the results.

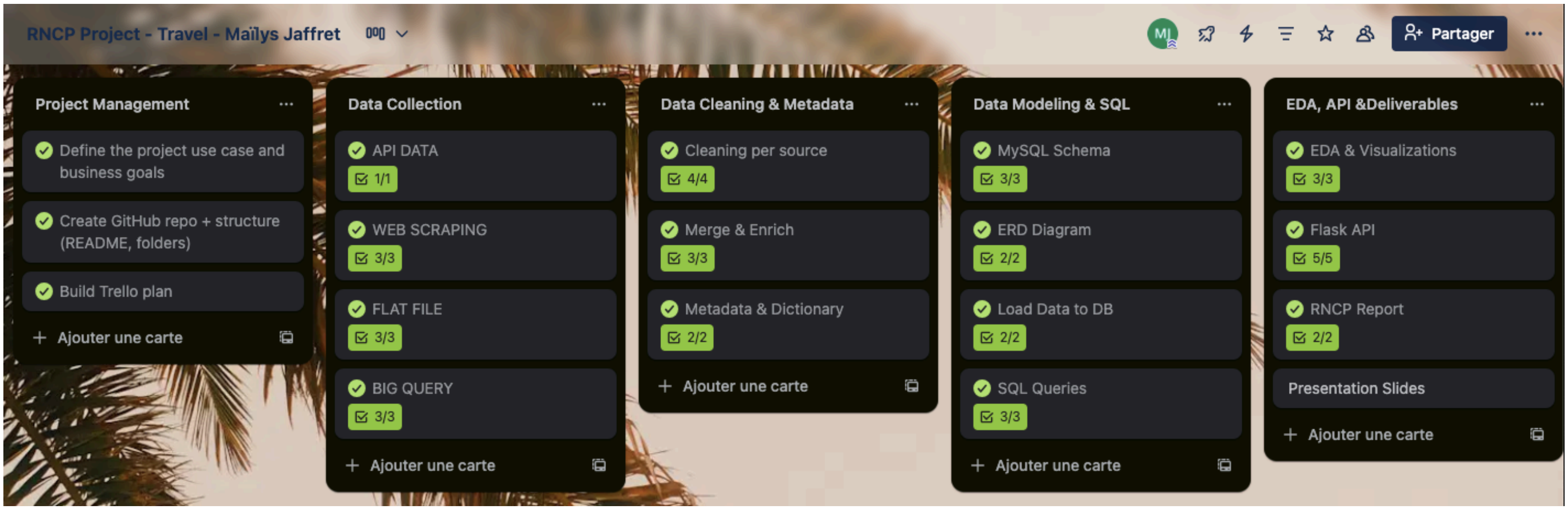
High-level plan:

- Research about the project topic (tourism and accommodation markets)
- Data collection from multiple sources:
  - API (Eurostat)
  - Web scraping (Wikipedia for arrivals, Numbeo for cost of living)
  - Flat file (Airbnb listings)
  - Relational database (MySQL for integration)
  - Big Data system (BigQuery for scalability)
- Project scope and planning in Trello
- Exploratory data analysis in Python (data wrangling, cleaning, visualization)
- Selection and creation of a relational database schema with MySQL
- Adding data to the database and designing an Entity Relationship Diagram (ERD)
- Data manipulation in SQL (5 queries to extract insights)
- Exposing data via a Flask API (with multiple endpoints)
- Visualization of insights in Tableau

2- Project Management

Trello:

Kanban board to manage project tasks



Project Management
- Define the project use case and business goals
- Create GitHub repo + structure (README, folders)
- Build Trello plan

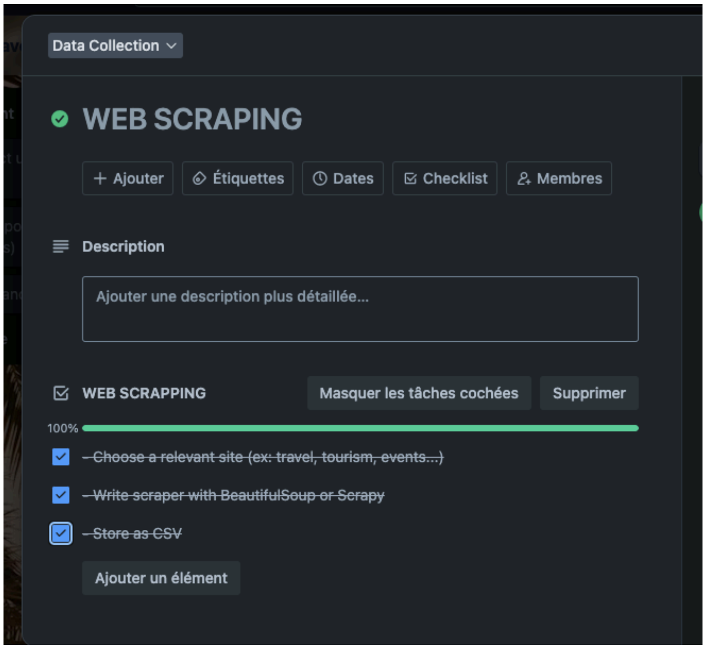
Data Collection
- API DATA 1/1
- WEB SCRAPING 3/3
- FLAT FILE 3/3
- BIG QUERY 3/3

Data Cleaning & Metadata
- Cleaning per source 4/4
- Merge & Enrich 3/3
- Metadata & Dictionary 2/2

Data Modeling & SQL
- MySQL Schema 3/3
- ERD Diagram 2/2
- Load Data to DB 2/2
- SQL Queries 3/3

EDA, API &Deliverables
- EDA & Visualizations 3/3
- Flask API 5/5
- RNCP Report 2/2
- Presentation Slides

Most of the cards on the Kanban board included a checklist to track specific sub-tasks. Once all checklist items were completed, the card was updated and marked as « Done ».



WEB SCRAPING

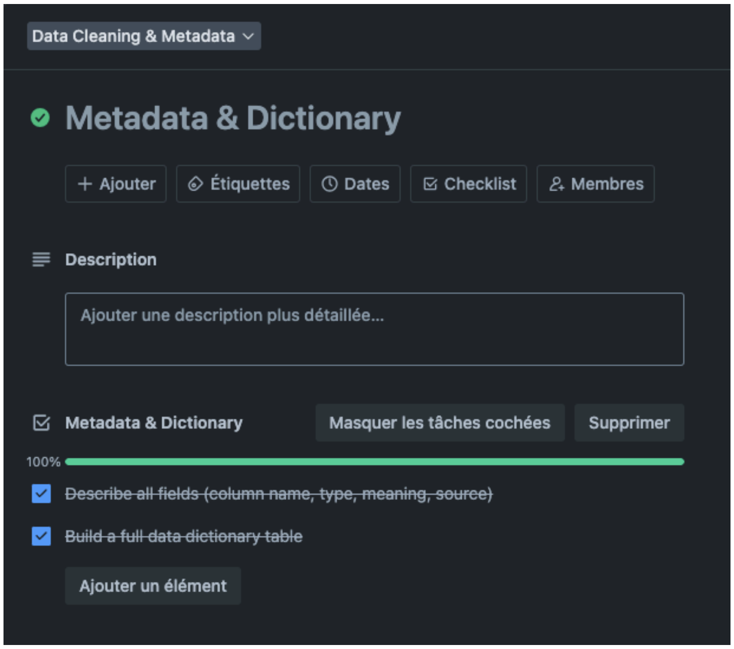
Description

Ajouter une description plus détaillée...

WEB SCRAPPING

100%

- ☑ ~~Choose a relevant site (ex: travel, tourism, events...)~~
- ☑ ~~Write scraper with BeautifulSoup or Scrapy~~
- ☑ ~~Store as CSV~~



Metadata & Dictionary

Description

Ajouter une description plus détaillée...

Metadata & Dictionary

100%

- ☑ ~~Describe all fields (column name, type, meaning, source)~~
- ☑ ~~Build a full data dictionary table~~

3- Data Collection and Data Dictionary

To better address the research objectives, the data collection process was organized around three main dimensions: international tourist arrivals, cost of living, and short-term rental markets. Each of these dimensions was explored using different types of sources and collection methods, ensuring diversity and complementarity of the information gathered. In total, five distinct approaches were applied: web scraping, APIs, flat files, structured databases, and big data integration.

3.1. API : Eurostat

I accessed official EU tourism statistics through the Eurostat API to size demand and provide context for my city analysis. Eurostat offers harmonized, well-documented datasets and a stable API.

1-1- Trips by main mode of Accommodation

https://ec.europa.eu/eurostat/databrowser/view/tour_dem_ttac/default/table?lang=en

Number of trips with at least one overnight stay made by EU residents broken down by the main accommodation mode used on the trip.
This dataset provides insights into traveler preferences regarding accommodation types, which is crucial to understand patterns and compare them with Airbnb's short-term rental offer. Modes of accommodation :

- *Non-rented accommodation:*
    - Accommodation provided without charge by relatives or friends
    - Own holiday home
    - Other

- *Rented accommodation:*
    - Apartment, house, villa or room in a dwelling
    - Campsites, carve, or trailer park
    - Hotels or similar establishments
    - Other

1-2- Trips by main mode of Transport

https://ec.europa.eu/eurostat/databrowser/view/tour_dem_tttr/default/table?lang=en

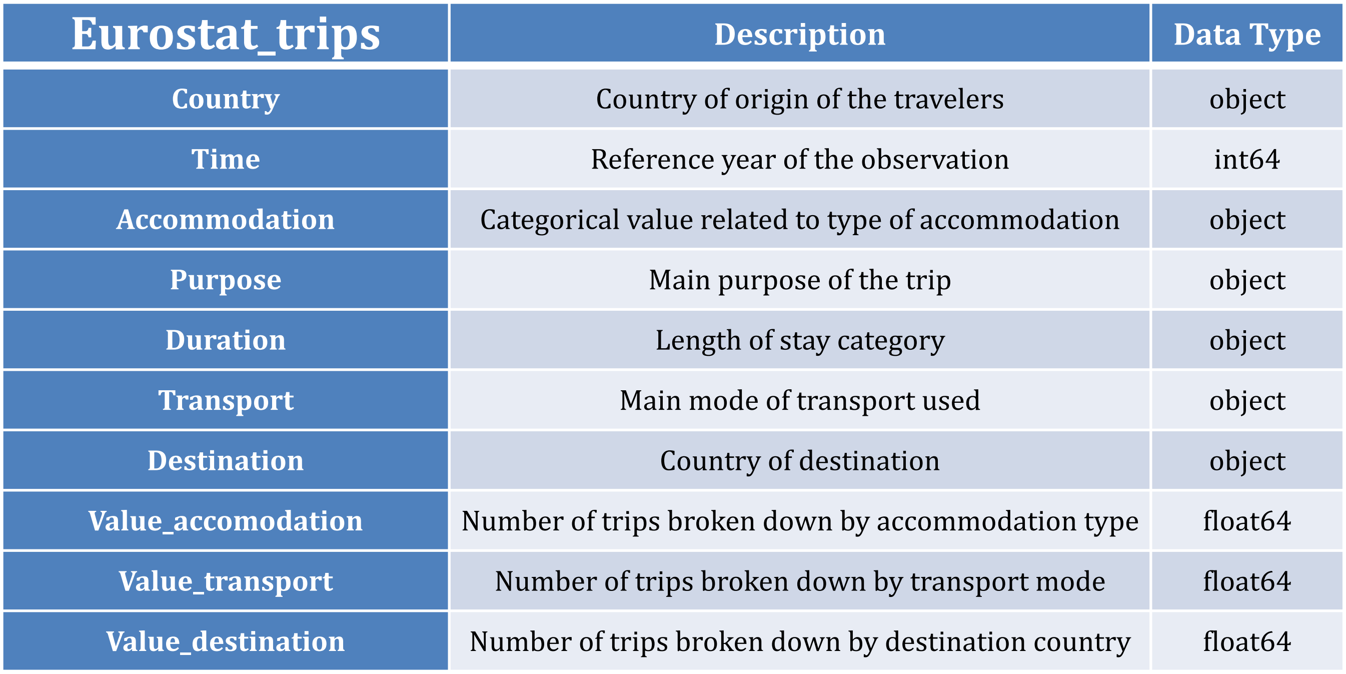
Number of trips with at least one overnight stay by EU residents broken down by main transport mode used on the trip. Transport accessibility is a key factor in tourism flows.
This dataset helps explain visitor mobility patterns and complements the city-level analysis by showing how tourists travel to their destinations.
Modes of transport:
- Air / Buses, coaches / Passenger road motor vehicles excluding buses and coaches / Railways /Waterway / Other land

1-3- Trips by detail net country/world region of main destination

https://ec.europa.eu/eurostat/databrowser/view/tin00190/default/table?lang=en

Number of trips with at least one overnight stay made by EU residents during the year, broken down by destination country or world region (main destination of the trip).
This dataset allows identifying top EU destinations for 2023 and narrowing the scope to the capitals of France, Germany, Spain, and Italy (Paris, Berlin, Madrid, and Rome), ensuring a clear and comparable focus for the analysis.

Data Dictionary for eurostat_trips:



Eurostat_trips	Description	Data Type
Country	Country of origin of the travelers	object
Time	Reference year of the observation	int64
Accommodation	Categorical value related to type of accommodation	object
Purpose	Main purpose of the trip	object
Duration	Length of stay category	object
Transport	Main mode of transport used	object
Destination	Country of destination	object
Value_accomodation	Number of trips broken down by accommodation type	float64
Value_transport	Number of trips broken down by transport mode	float64
Value_destination	Number of trips broken down by destination country	float64

3.2. Web scraping

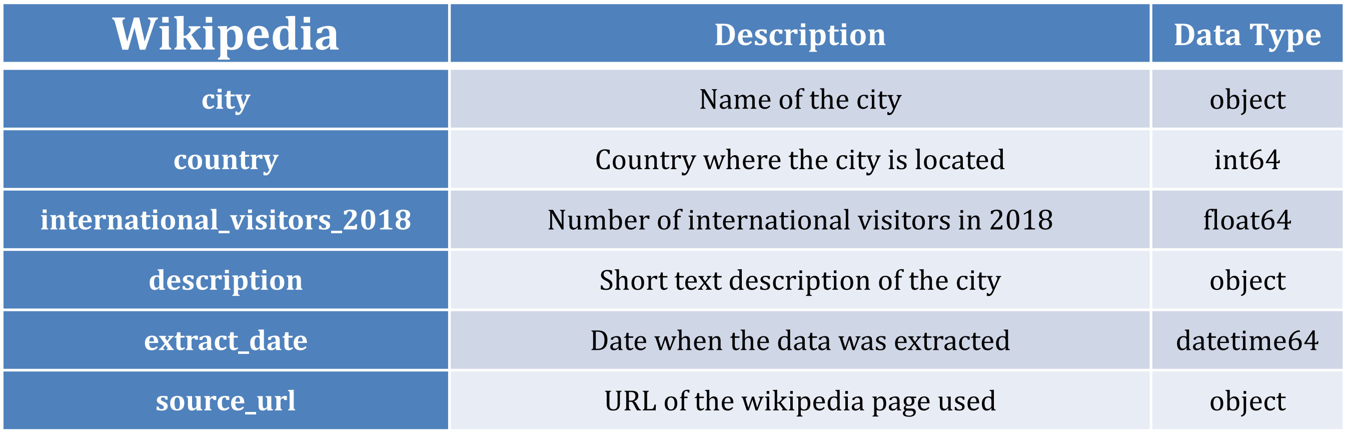
I scraped three public sites to add city and country context to the Eurostat data.

2-1- Wikipedia: international visitors (city arrivals)

https://en.wikipedia.org/wiki/List_of_cities_by_international_visitors?

The last common year available is 2018. This dataset focuses on the number of international tourists in the capitals (Paris, Madrid, Rome, Berlin) and provides city descriptions for context.

Data Dictionary for wikipedia:



Wikipedia	Description	Data Type
city	Name of the city	object
country	Country where the city is located	int64
international_visitors_2018	Number of international visitors in 2018	float64
description	Short text description of the city	object
extract_date	Date when the data was extracted	datetime64
source_url	URL of the wikipedia page used	object

2-2- WorldData: country arrivals (10 years trend)

France : **https://www.worlddata.info/europe/france/tourism.php**
Germany : **https://www.worlddata.info/europe/germany/tourism.php**
Spain : **https://www.worlddata.info/europe/spain/tourism.php**
Italy : **https://www.worlddata.info/europe/italy/tourism.php**

WorldData provides a 10-year trend of tourist arrivals at the country level. It allows comparing pre- and post-COVID tourism flows for France, Germany, Spain, and Italy.

Data Dictionary for world_data:

World_data	Description	Data Type
Country	Name of the country	object
Year	Reference year of the observation	int64
Arrivals (millions)	Number of international tourist arrivals, expressed in millions of visitors.	float64

2-3- Numbeo: cost of living

Paris : **https://www.numbeo.com/cost-of-living/in/Paris**
Berlin : **https://www.numbeo.com/cost-of-living/in/Berlin**
Madrid : **https://www.numbeo.com/cost-of-living/in/Madrid**
Rome : **https://www.numbeo.com/cost-of-living/in/Rome**

Numbeo provides city-level prices in EUR for Cappuccino, Cinema, Fitness monthly, Gasoline 1l, Mcdonalds, Meal inexpensive restaurant, Monthly pass (transport), One way ticket (transport), Taxi 1km. It helps compare the affordability of each capital for tourists.

Data Dictionary for numbeo:

Numbeo	Description	Data Type
City	Name of the city where the item price was collected	object
Item_label	Name of the good or service	object
Price_eur	Price of the item in euros	float64

3.3. Flat files: Airbnb

https://insideairbnb.com/fr/get-the-data/?

I downloaded Airbnb listings data for the four capitals to capture the structure of short-term rental markets. The files provide detailed information on supply (types of accommodation, availability, number of bedrooms, beds, bathrooms) as well as demand indicators (price, number of reviews, review scores).

Files used:

- listings_paris.csv
- listings_berlin.csv
- listings_madrid.csv
- listings_rome.csv

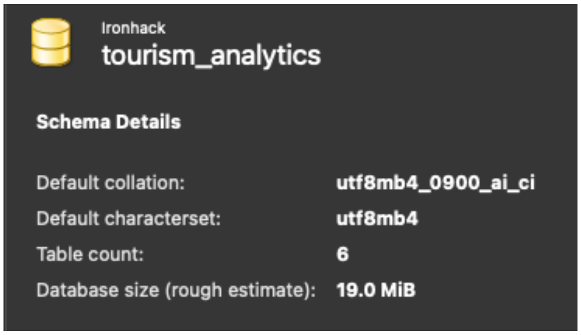
Data Dictionary for airbnb:

Airbnb	Description	Data Type
listing_name	Name of the listing	object
neighbourhood_cleansed	Standardized neighborhood name where the listing is located	object
latitude	Latitude coordinate of the listing	float64
longitude	Longitude coordinate of the listing	float64
room_type	Type of room : entire, private, shared…	object
accommodates	Maximum number of guests the listing can host	object
bedrooms	Number of bedrooms available in the listing	float64
beds	Number of beds provided in the listing	float64
bathrooms	Number of bathrooms	float64
bathrooms_text	Text description of bathrooms	object
price	Price of the listing per night	int64
minimum_nights	Minimum number of nights allowed per booking	int64
maximum_nights	Maximum number of nights allowed per booking	int64
availability_365	Number of days the listing is available for booking within a year (0 - 365)	int64
number_of_reviews	Total number of reviews received by the listing	int64
review_scores_rating	Average review rating (0 - 5)	float64
city	City where the listing is located	object

3.4. Database Creation - MySQL Workbench

The « tourism_analytics » database was created in MySQL Workbench to store 6 tables related to the tourism in Europe.

The schema contains 6 interrelated tables: airbnb_listing, city, country, eurostat_fact, numbeo_cost, and worlddata_fact. These tables bring together data from multiple sources, covering aspects such as tourist flows, cost of living, country and city attributes, and short-term rental markets. By using a relational database, it is possible to manage structured data with clear relationships, reduce redundancy, ensure integrity, and efficiently perform SQL queries involving joins across multiple tables.

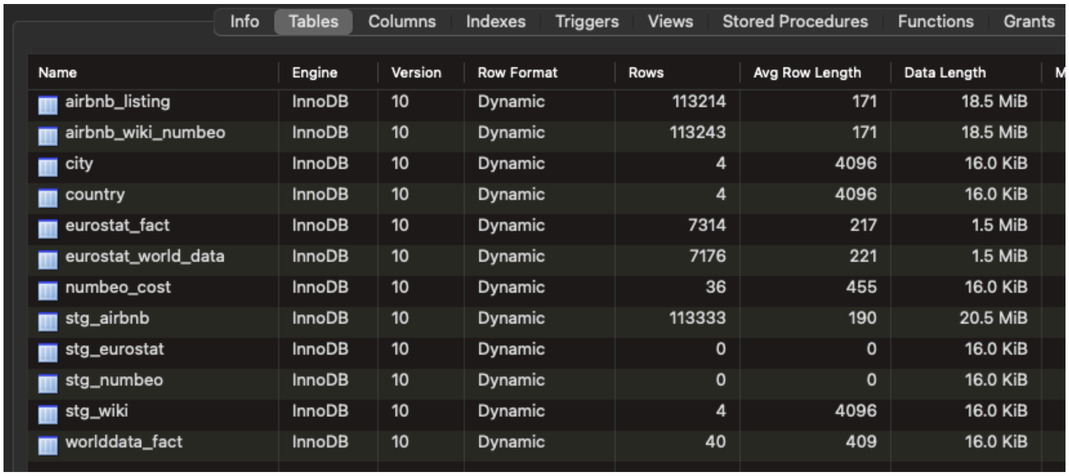


Ironhack
tourism_analytics

Schema Details

Default collation: utf8mb4_0900_ai_ci
Default characterset: utf8mb4
Table count: 6
Database size (rough estimate): 19.0 MiB

Database tables:



Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	M
airbnb_listing	InnoDB	10	Dynamic	113214	171	18.5 MiB	
airbnb_wiki_numbeo	InnoDB	10	Dynamic	113243	171	18.5 MiB	
city	InnoDB	10	Dynamic	4	4096	16.0 KiB	
country	InnoDB	10	Dynamic	4	4096	16.0 KiB	
eurostat_fact	InnoDB	10	Dynamic	7314	217	1.5 MiB	
eurostat_world_data	InnoDB	10	Dynamic	7176	221	1.5 MiB	
numbeo_cost	InnoDB	10	Dynamic	36	455	16.0 KiB	
stg_airbnb	InnoDB	10	Dynamic	113333	190	20.5 MiB	
stg_eurostat	InnoDB	10	Dynamic	0	0	16.0 KiB	
stg_numbeo	InnoDB	10	Dynamic	0	0	16.0 KiB	
stg_wiki	InnoDB	10	Dynamic	4	4096	16.0 KiB	
worlddata_fact	InnoDB	10	Dynamic	40	409	16.0 KiB	

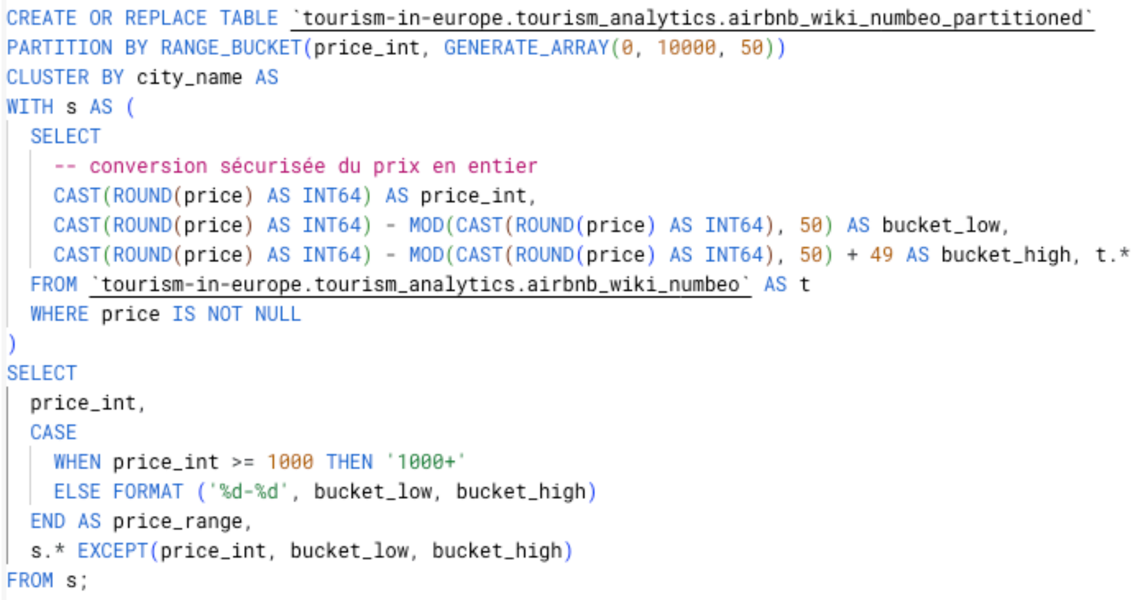
3.5. Big Data System : BigQuery

To complement the SQL database in MySQL, I chose to use Google BigQuery for the Big Data part of the project. BigQuery is a cloud-based data warehouse designed for very large datasets. It provides scalability, fast queries on millions of rows, and native features such as partitioning and clustering.

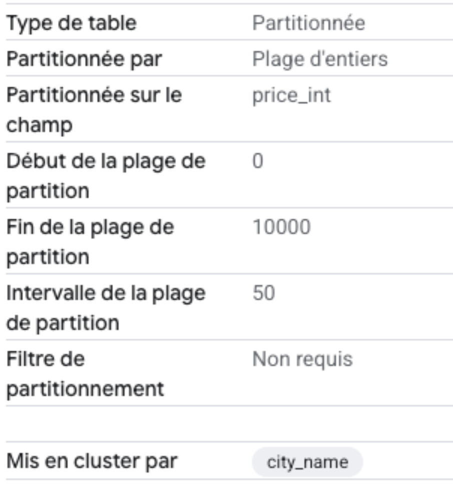
The Airbnb dataset was denormalized for BigQuery to perform operation of comparing each city. I created a partitioned and clustered version of my Airbnb dataset in BigQuery.

- Partitioning: based on price_int, using RANGE_BUCKET to group listings into price intervals of 50 EUR.
- Clustering: by city_name, so queries filtered by city are more efficient.
- Price ranges: I added a derived field price_range to categorize listings (e.g., 0–49, 50–99, …, 1000+).

This setup makes the dataset easier to analyze by price category while keeping queries optimized through partitioning and clustering.

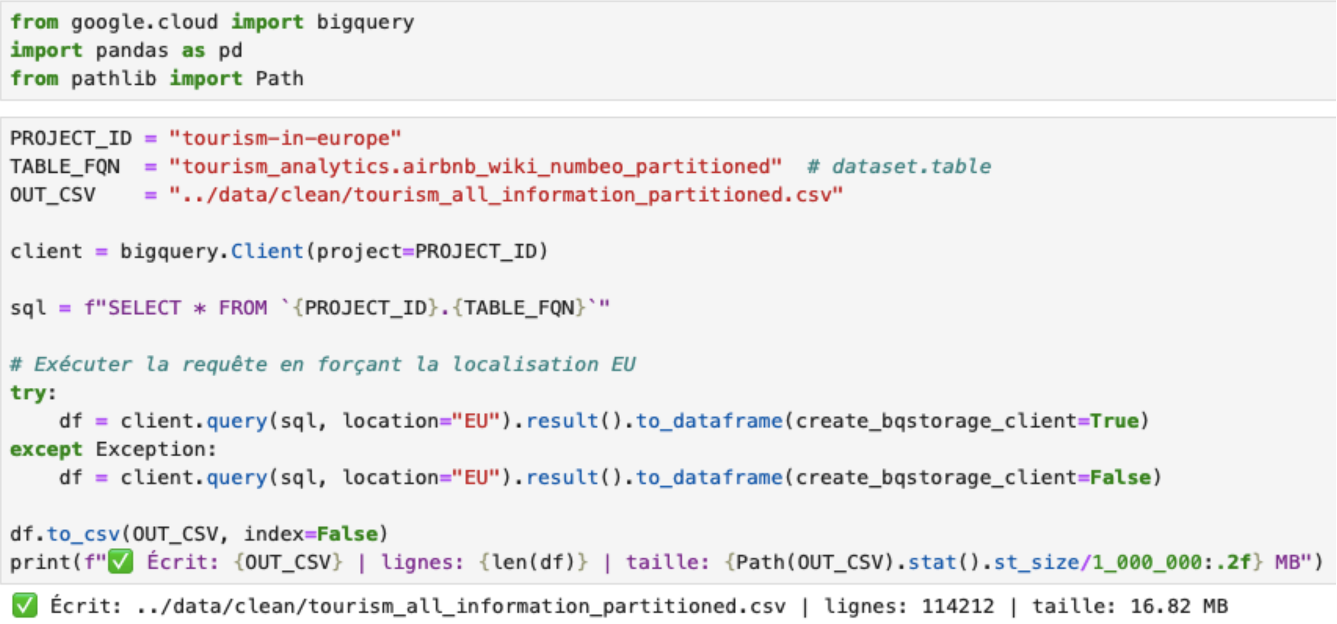


```
CREATE OR REPLACE TABLE `tourism-in-europe.tourism_analytics.airbnb_wiki_numbeo_partitioned`
PARTITION BY RANGE_BUCKET(price_int, GENERATE_ARRAY(0, 10000, 50))
CLUSTER BY city_name AS
WITH s AS (
  SELECT
    -- conversion sécurisée du prix en entier
    CAST(ROUND(price) AS INT64) AS price_int,
    CAST(ROUND(price) AS INT64) - MOD(CAST(ROUND(price) AS INT64), 50) AS bucket_low,
    CAST(ROUND(price) AS INT64) - MOD(CAST(ROUND(price) AS INT64), 50) + 49 AS bucket_high, t.*
  FROM `tourism-in-europe.tourism_analytics.airbnb_wiki_numbeo` AS t
  WHERE price IS NOT NULL
)
SELECT
  price_int,
  CASE
    WHEN price_int >= 1000 THEN '1000+'
    ELSE FORMAT ('%d-%d', bucket_low, bucket_high)
  END AS price_range,
  s.* EXCEPT(price_int, bucket_low, bucket_high)
FROM s;
```



Type de table	Partitionnée
Partitionnée par	Plage d'entiers
Partitionnée sur le champ	price_int
Début de la plage de partition	0
Fin de la plage de partition	10000
Intervalle de la plage de partition	50
Filtre de partitionnement	Non requis
Mis en cluster par	city_name

Finally, I connected a Python notebook to BigQuery to query and export the results into CSV files. This workflow gives flexibility: I can refresh the dataset, run new queries, or update the export at any time. With this approach, I was able to prepare a clean and efficient dataset for dashboards and storytelling.



```python
from google.cloud import bigquery
import pandas as pd
from pathlib import Path
```

```python
PROJECT_ID = "tourism-in-europe"
TABLE_FQN  = "tourism_analytics.airbnb_wiki_numbeo_partitioned"  # dataset.table
OUT_CSV    = "../data/clean/tourism_all_information_partitioned.csv"

client = bigquery.Client(project=PROJECT_ID)

sql = f"SELECT * FROM `{PROJECT_ID}.{TABLE_FQN}`"

# Exécuter la requête en forçant la localisation EU
try:
    df = client.query(sql, location="EU").result().to_dataframe(create_bqstorage_client=True)
except Exception:
    df = client.query(sql, location="EU").result().to_dataframe(create_bqstorage_client=False)

df.to_csv(OUT_CSV, index=False)
print(f"✅ Écrit: {OUT_CSV} | lignes: {len(df)} | taille: {Path(OUT_CSV).stat().st_size/1_000_000:.2f} MB")
```

```
✅ Écrit: ../data/clean/tourism_all_information_partitioned.csv | lignes: 114212 | taille: 16.82 MB
```

4. Data cleaning

Airbnb Data

- Merged multiple city CSV files : *pd.concat*
- Reindexed missing columns for schema alignment : *df.reindex(columns=all_cols)*
- Selected useful columns : *df[useful_cols]*
- Dropped missing values : *df.dropna(subset=[« price"])*
- Filtered invalid rows : *df[df[« price"] != 0]*

Eurostat Data

- Downloaded JSON data : *requests.get*
- Decoded sparse cube indices : *unravel()* custom function
- Converted values to numeric : *pd.to_numeric(df[« value"], errors="coerce")*
- Aggregated by country/year : *df.groupby([…]).sum()*
- Removed aggregates using conditions : *df[~df[« Country »].isin(exclude_geo)]*

Numbeo

- Scraped cost-of-living tables : *requests.get* + *BeautifulSoup*
- Indexed rows by label : *by_label[label] = value*
- Parsed numeric values with regex in : *parse_price()* (handled ranges, decimals, currencies)
- Normalized labels : *unicodedata.normalize("NFKC", s).lower().strip()*
- Extracted items : meals, McDonald's, cappuccino, gasoline, transport, cinema, fitness

Wikipedia Data

- Scraped tables : *pd.read_html*
- Parsed numbers with regex *re.sub* inside *parse_number()* → handled formats like "24.5 million" or "24,500,000".
- Normalized city names : *.str.replace(…, regex=True).str.strip()*
- Converted columns to numeric : *pd.to_numeric(df[« year"], errors="coerce")*
- Dropped duplicates : *df.drop_duplicates(subset=[« city"])*

WorldData Information

- Scraped HTML pages : *requests.get* + *BeautifulSoup*
- Extracted relevant tables : *pd.read_html*
- Cleaned arrival numbers with regex in : *parse_arrivals_cell()*
- Standardized decimals : *replace(« ,", ".")*
- Converted to integers : *int(round(val))*
- Filtered last 10 years : *.sort_values(« year »).tail(10)*

General Cleaning Practices Across Sources

- **Handling missing data** → dropna, conditional filtering.
- **String normalization** → .str.strip(), .str.lower(), unicodedata.normalize.
- **Regex transformations** → re.sub for removing unwanted symbols/units.
- **Schema harmonization** → renaming columns (df.rename(columns=...)) + reindexing.
- **Outlier/aggregate removal** → explicit exclusion lists (exclude_geo, exclude_transport, etc.).

5. EDA and visualizations

5.1. H1 : Room type influence price

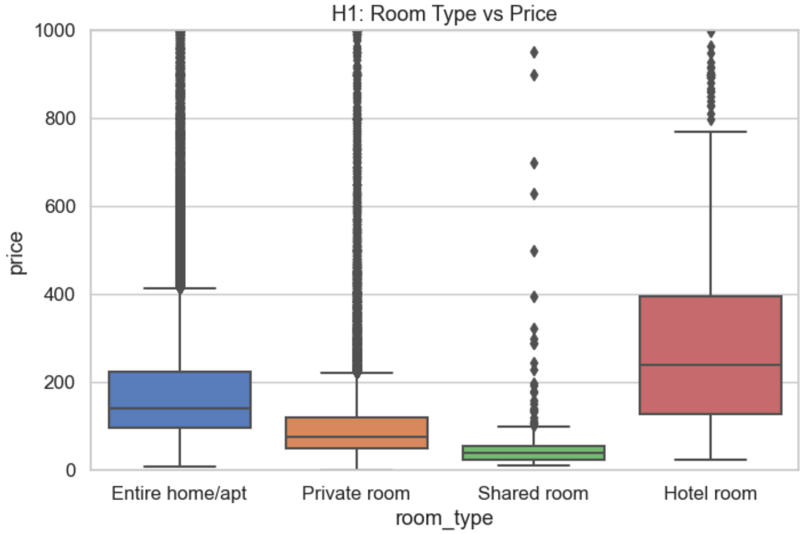
The analysis of raw Airbnb prices shows a highly skewed distribution: most listings are priced below 300 €, but a few extreme values reach up to 1000 €.
To reduce the impact of these outliers, we applied a log transformation (log(1+price)), which resulted in a distribution much closer to normal.

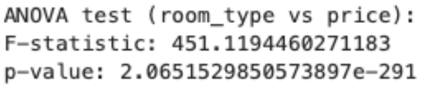
The distribution of prices shows that *entire homes/apartments* and *hotel rooms* are much more expensive than *private* or *shared rooms*.

The boxplot confirms that the median price is:

- Entire home/apt: around 120-150 €
- Hotel room: around 200 €
- Private room: around 60-70 €
- Shared room: around 30-40 €

An ANOVA test gave a p-value close to 0, which means the differences between room types are statistically significant.





```
ANOVA test (room_type vs price):
F-statistic: 451.1194460271183
p-value: 2.0651529850573897e-291
```

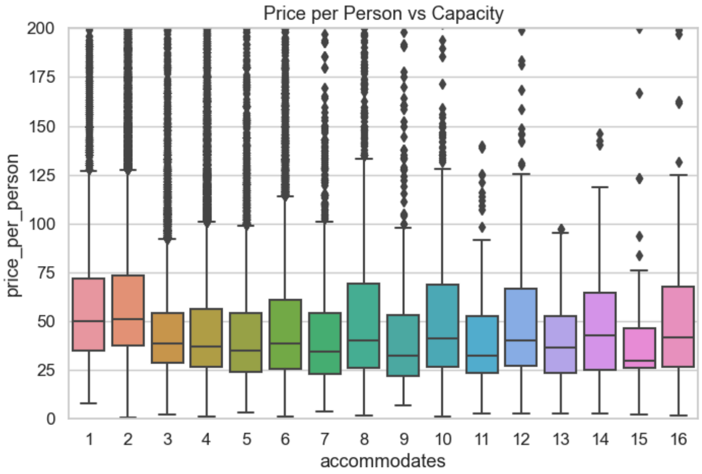
Conclusion: Room type is a key factor that explains the variation in Airbnb prices. Entire homes and hotel rooms are, on average, 2 to 3 times more expensive than private or shared rooms.

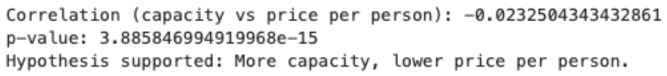
5.2. H2 : Capacity vs Price per person

I tested the hypothesis that larger listings are cheaper per person.

- The correlation between capacity and price per person is -0.023. This shows a very weak negative relationship.
- The p-value is extremely small (< 0.001), which means the result is statistically significant.
- The boxplot also confirms the trend: price per person tends to decrease slightly as capacity increases.

Conclusion: The hypothesis is supported, but the effect is weak. Larger listings are only slightly cheaper per person on average, but the trend is consistent and statistically significant.



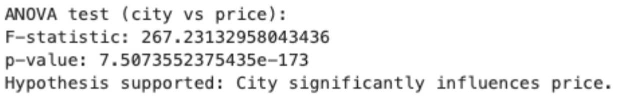


```
Correlation (capacity vs price per person): -0.0232504343432861
p-value: 3.885846994919968e-15
Hypothesis supported: More capacity, lower price per person.
```

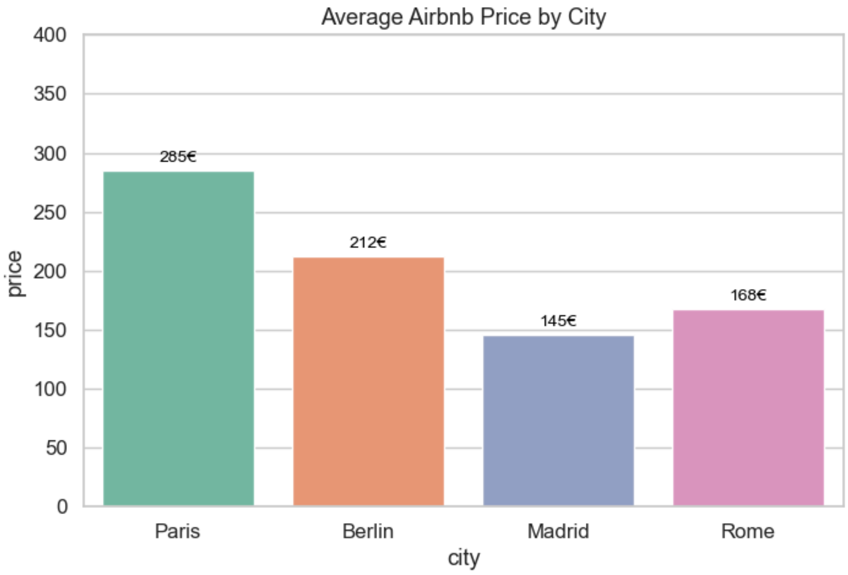
5.3. H3: The city has a significant influence on Airbnb prices

I tested the hypothesis that the city influences Airbnb prices.

- The boxplot shows clear differences across cities.
- The average price per city is:
  - Paris: 285 €
  - Berlin: 212 €
  - Rome: 168 €
  - Madrid: 145 €
- The ANOVA test confirms this with an F-statistic of 267.2 and a p-value close to 0.



```
ANOVA test (city vs price):
F-statistic: 267.23132958043436
p-value: 7.5073552375435e-173
Hypothesis supported: City significantly influences price.
```



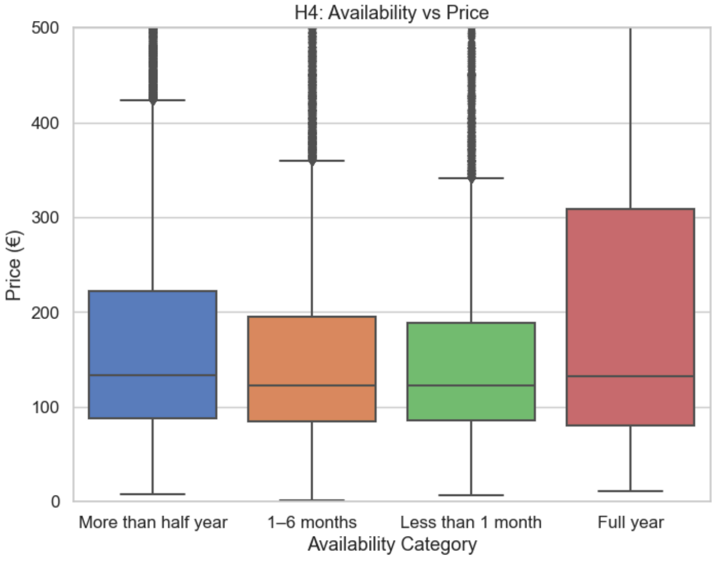
Conclusion: The hypothesis is supported. City is a strong determinant of Airbnb pricing. Paris is the most expensive destination in our dataset, while Madrid is the cheapest.

5.4. Does the rating score influence the price?

I tested the relationship between the review score rating and the price of Airbnb listings. The correlation analysis showed a coefficient of 0.0 with a p-value of 0.90. This means that there is no significant correlation between rating scores and prices. In other words, better-rated listings do not necessarily charge higher prices.

5.5. Availability (days listed per year) influence price

The boxplot shows that listings available all year tend to have higher prices compared to those available for shorter periods (less than 1 month, 1–6 months, or more than half year). The median price is visibly higher for full-year listings, and the distribution is wider, suggesting that professional or commercial rentals might dominate this category.



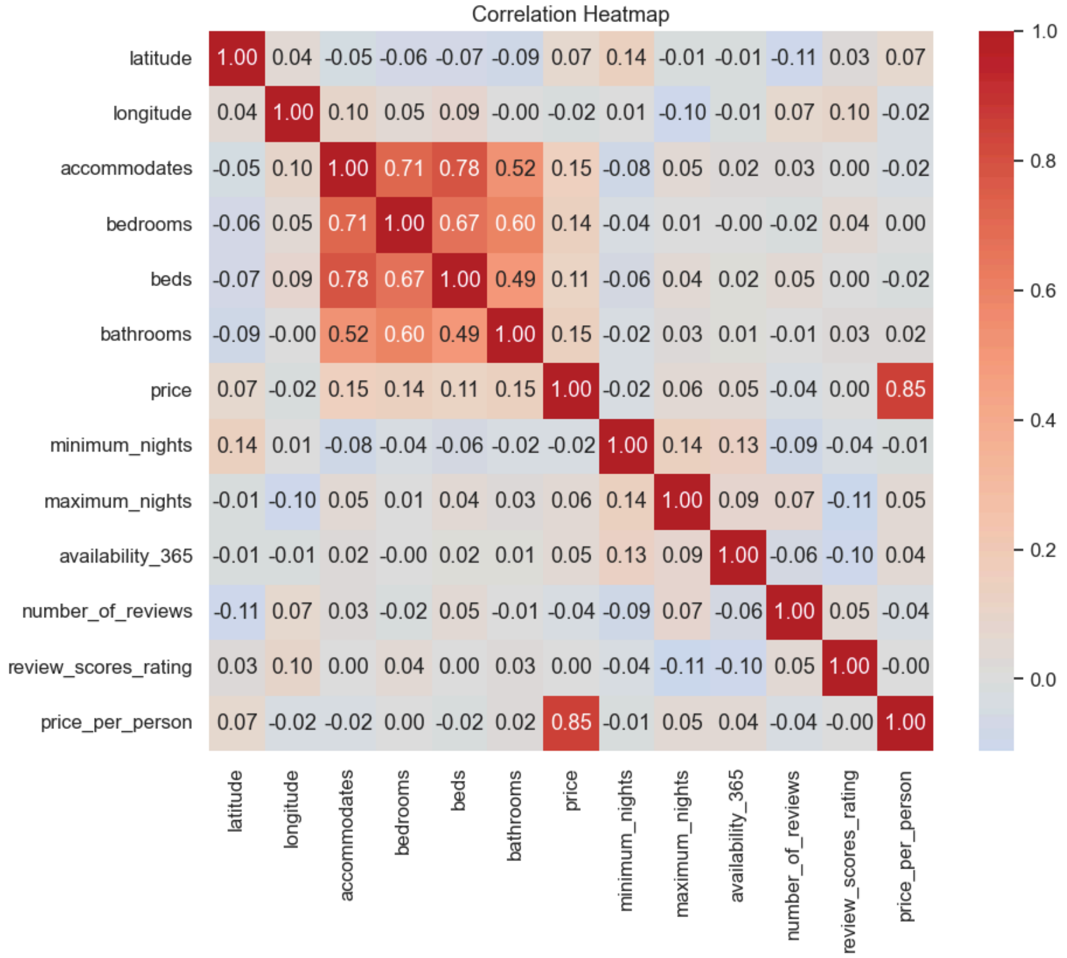
To test this pattern, I used two approaches:

1. **ANOVA test** (categorical availability)
I divided availability into four categories: *Less than 1 month*, *1–6 months*, *More than half year*, and *Full year*. The ANOVA test shows that there are significant differences in prices across these categories (F = 146.66, $p < 0.001$). Listings available all year tend to have higher prices.

2. **Correlation test** (numeric availability in days)
I also tested the exact number of days available (0–365) against the price. The correlation coefficient was $r = 0.05$, with a very low p-value, indicating that although the relationship is statistically significant, the effect size is very weak.

Conclusion: Availability influences price mainly at the categorical level (full-year listings vs. others), but the linear relationship with the exact number of days is weak.

5.6. Correlation Heatmap - Numerical values



5.7. Conclusion EDA

The correlation analysis and statistical tests provide a clear view of what drives Airbnb prices. Capacity variables (beds, bedrooms, accommodates) are strongly correlated with each other but only weakly linked to price, confirming that capacity has little influence on pricing. Review scores also show no meaningful effect, which means that ratings do not shape how listings are priced.

By contrast, availability has a significant impact when treated as categories: listings open all year form a separate group with higher prices, often run by professional hosts. This effect does not appear as a linear correlation but as a clear difference between groups. The strong correlation between price and price_per_person is expected since one is derived from the other.

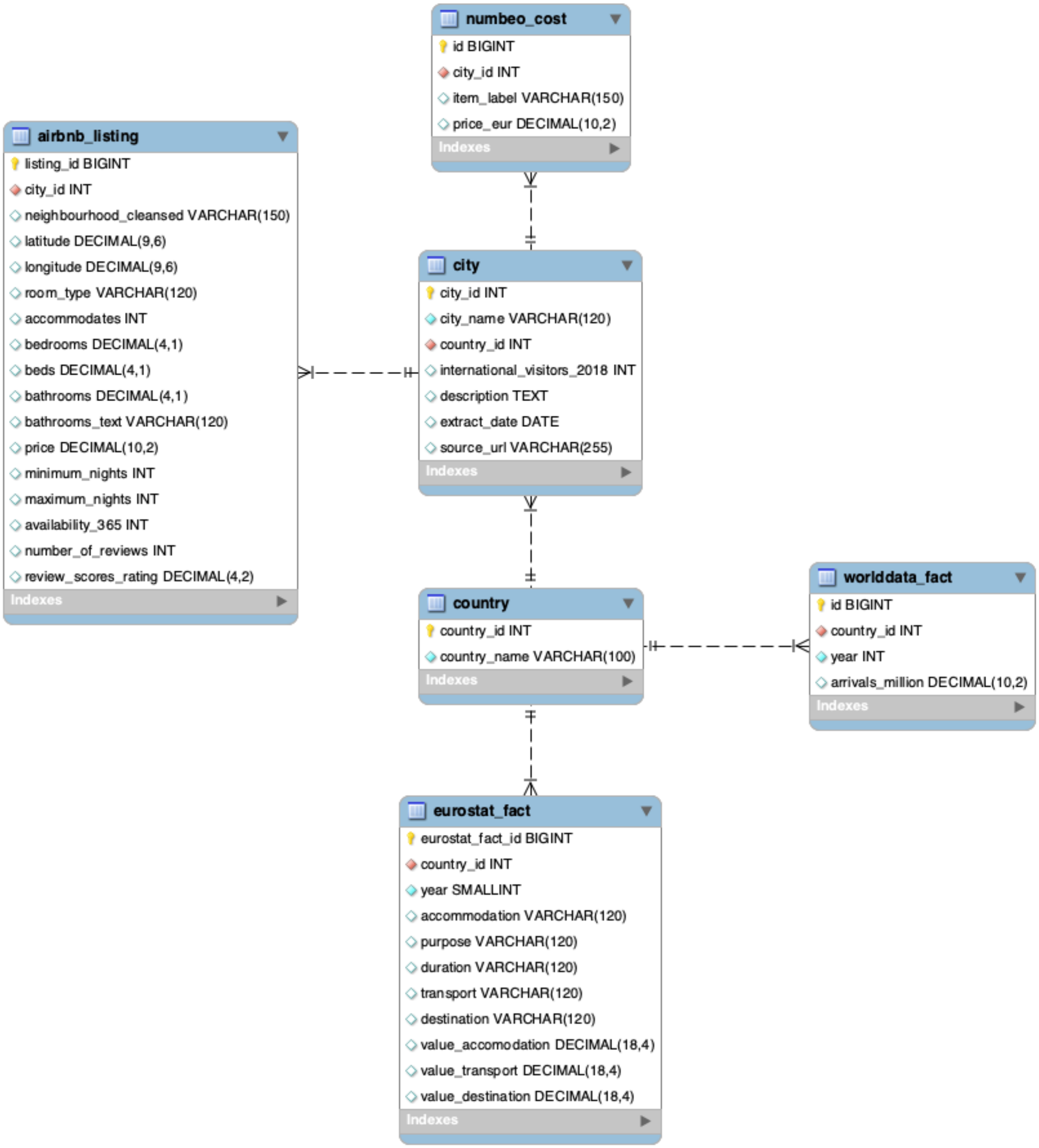
Beyond the numeric correlations, the main price drivers are categorical: room type, city, and availability. Entire homes and hotel rooms are the most expensive, Paris is the costliest destination, Madrid the most affordable, and year-round availability is linked to professionalized rentals. These factors define clear market segments: premium offers for high-budget travelers, affordable southern cities for price-sensitive visitors, and professional hosts dominating the top end of the market.

From a business perspective, these findings suggest several actions:

- Hosts and Airbnb can adapt strategies by targeting different segments — budget travelers with private rooms, groups with larger listings, and premium clients with entire homes or hotel-style options.
- Tourism boards can promote the competitiveness of affordable cities such as Madrid and Rome to attract travelers priced out of Paris.

Overall, the analysis confirms that short-term rental prices in Europe follow clear structural patterns. Data-driven insights like these help businesses improve their offers and allow policymakers to design fairer regulations that balance tourism growth with urban housing needs.

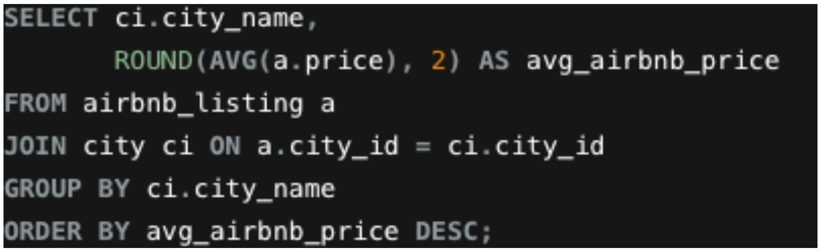
6. Entity-relationship diagram



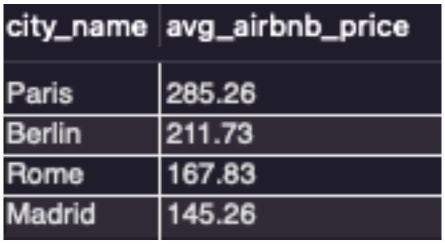
The Entity-relationship Diagram shows the structure of my tourism analytics database. Two reference tables were created: country and city. Each country has a unique primary key **country_id**, and each city has its own primary key **city_id** linked to the country through a foreign key. The fact tables are then connected to these references. For example, Airbnb listings and Numbeo cost of living use the *city_id foreign key*, while Eurostat statistics and Worlddata arrivals use the *country_id foreign key*. This structure ensures data consistency, avoids duplicates, and makes it possible to analyze information at both the city and country levels.

7. 5 SQL Scripts

7.1. Average Airbnb Price per City



```sql
SELECT ci.city_name,
       ROUND(AVG(a.price), 2) AS avg_airbnb_price
FROM airbnb_listing a
JOIN city ci ON a.city_id = ci.city_id
GROUP BY ci.city_name
ORDER BY avg_airbnb_price DESC;
```

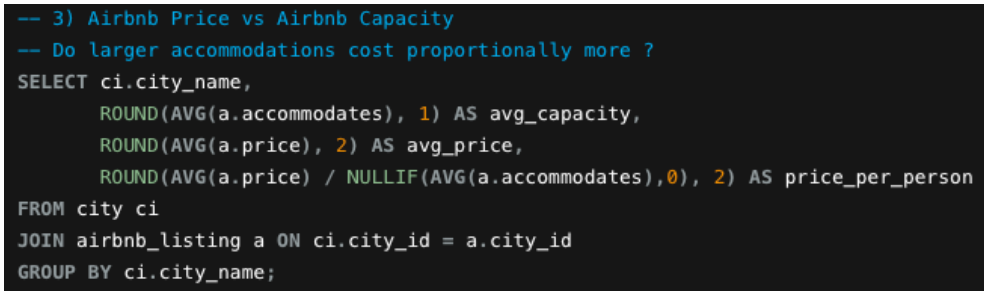


city_name	avg_airbnb_price
Paris	285.26
Berlin	211.73
Rome	167.83
Madrid	145.26

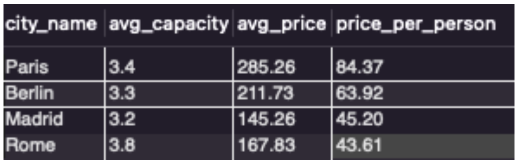
Paris has by far the most expensive Airbnb prices, while Madrid offers the most affordable stays among the four cities.

7.2. Airbnb price vs Airbnb Capacity

Do larger accommodations cost proportionally more ?



```sql
-- 3) Airbnb Price vs Airbnb Capacity
-- Do larger accommodations cost proportionally more ?
SELECT ci.city_name,
       ROUND(AVG(a.accommodates), 1) AS avg_capacity,
       ROUND(AVG(a.price), 2) AS avg_price,
       ROUND(AVG(a.price) / NULLIF(AVG(a.accommodates),0), 2) AS price_per_person
FROM city ci
JOIN airbnb_listing a ON ci.city_id = a.city_id
GROUP BY ci.city_name;
```



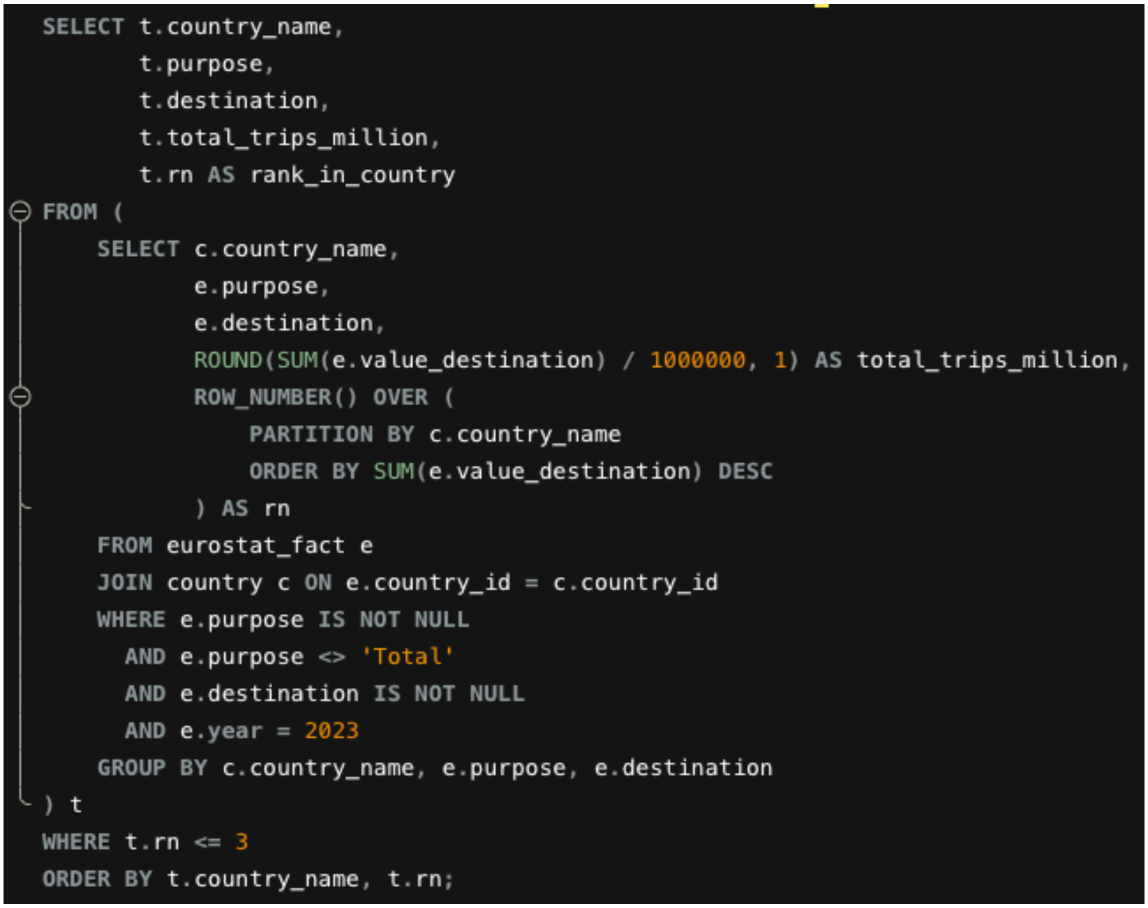
city_name	avg_capacity	avg_price	price_per_person
Paris	3.4	285.26	84.37
Berlin	3.3	211.73	63.92
Madrid	3.2	145.26	45.20
Rome	3.8	167.83	43.61

The analysis shows that larger Airbnb accommodations do not cost proportionally more.

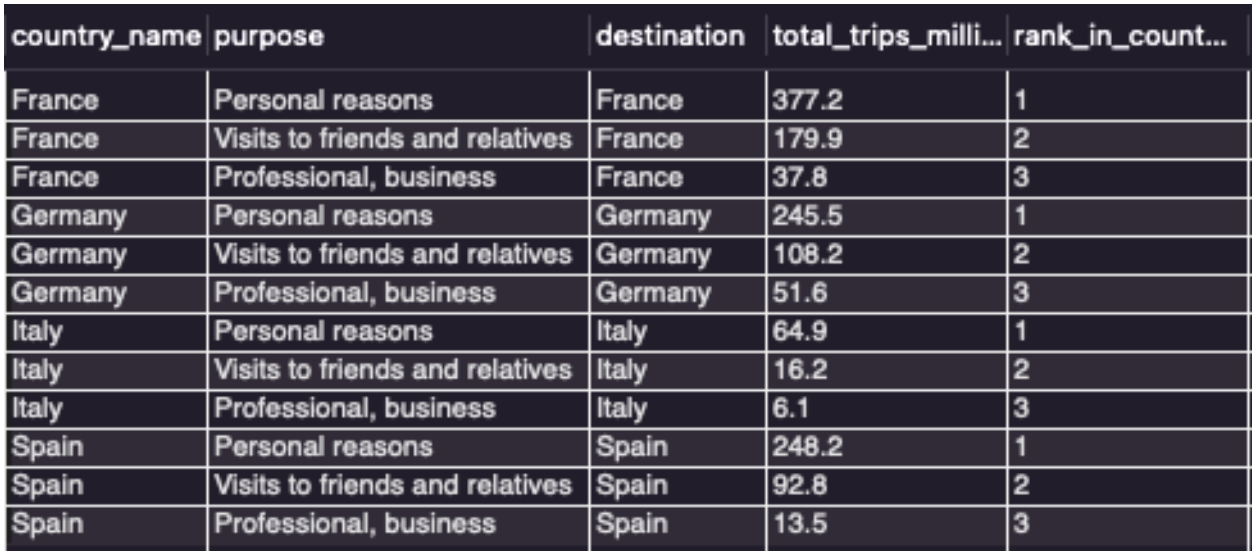
- Paris has the highest average nightly price (285€) and also the highest price per person (84€).
- Berlin is cheaper overall (212€ per night, 64€ per person).
- Madrid and Rome offer the lowest price per person (around 44€), even though Rome has the largest average capacity (3.8 guests).

Conclusion: While bigger listings are more expensive in total, the cost per person decreases as capacity increases. Larger accommodations actually provide a better value per traveler compared to smaller ones.

7.3. TOP 3 travel purposes and destinations by country (2023)



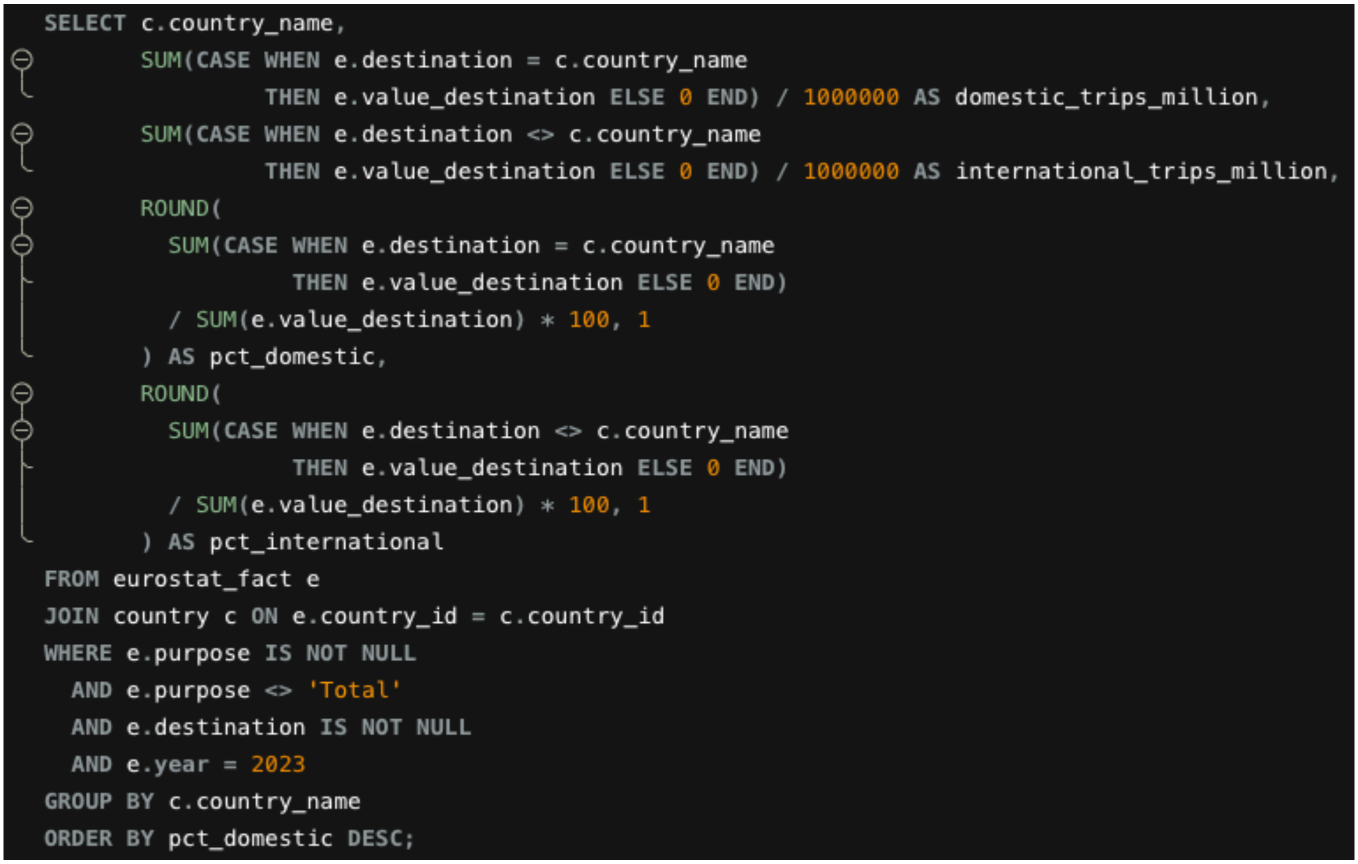
```
SELECT t.country_name,
       t.purpose,
       t.destination,
       t.total_trips_million,
       t.rn AS rank_in_country
FROM (
    SELECT c.country_name,
           e.purpose,
           e.destination,
           ROUND(SUM(e.value_destination) / 1000000, 1) AS total_trips_million,
           ROW_NUMBER() OVER (
               PARTITION BY c.country_name
               ORDER BY SUM(e.value_destination) DESC
           ) AS rn
    FROM eurostat_fact e
    JOIN country c ON e.country_id = c.country_id
    WHERE e.purpose IS NOT NULL
      AND e.purpose <> 'Total'
      AND e.destination IS NOT NULL
      AND e.year = 2023
    GROUP BY c.country_name, e.purpose, e.destination
) t
WHERE t.rn <= 3
ORDER BY t.country_name, t.rn;
```



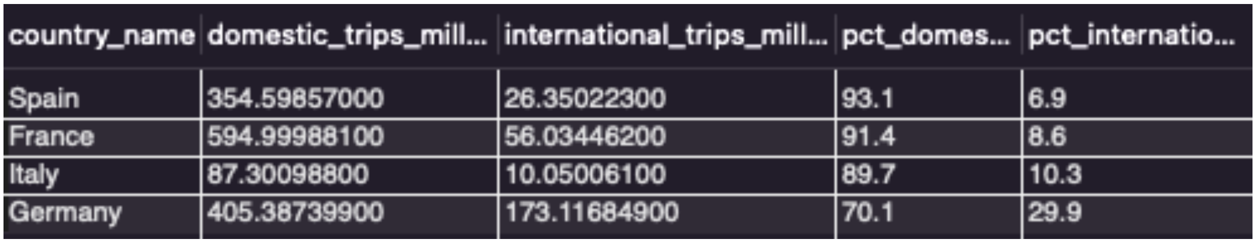
country_name	purpose	destination	total_trips_milli...	rank_in_count...
France	Personal reasons	France	377.2	1
France	Visits to friends and relatives	France	179.9	2
France	Professional, business	France	37.8	3
Germany	Personal reasons	Germany	245.5	1
Germany	Visits to friends and relatives	Germany	108.2	2
Germany	Professional, business	Germany	51.6	3
Italy	Personal reasons	Italy	64.9	1
Italy	Visits to friends and relatives	Italy	16.2	2
Italy	Professional, business	Italy	6.1	3
Spain	Personal reasons	Spain	248.2	1
Spain	Visits to friends and relatives	Spain	92.8	2
Spain	Professional, business	Spain	13.5	3

The pattern is similar for all the countries: the main destination is the country of origin itself, and the ranking of purposes (personal, family visits, business) remains the same across all countries.

7.4. Domestic vs International Trips by Country (2023)



```sql
SELECT c.country_name,
       SUM(CASE WHEN e.destination = c.country_name
                THEN e.value_destination ELSE 0 END) / 1000000 AS domestic_trips_million,
       SUM(CASE WHEN e.destination <> c.country_name
                THEN e.value_destination ELSE 0 END) / 1000000 AS international_trips_million,
       ROUND(
         SUM(CASE WHEN e.destination = c.country_name
                  THEN e.value_destination ELSE 0 END)
         / SUM(e.value_destination) * 100, 1
       ) AS pct_domestic,
       ROUND(
         SUM(CASE WHEN e.destination <> c.country_name
                  THEN e.value_destination ELSE 0 END)
         / SUM(e.value_destination) * 100, 1
       ) AS pct_international
FROM eurostat_fact e
JOIN country c ON e.country_id = c.country_id
WHERE e.purpose IS NOT NULL
  AND e.purpose <> 'Total'
  AND e.destination IS NOT NULL
  AND e.year = 2023
GROUP BY c.country_name
ORDER BY pct_domestic DESC;
```

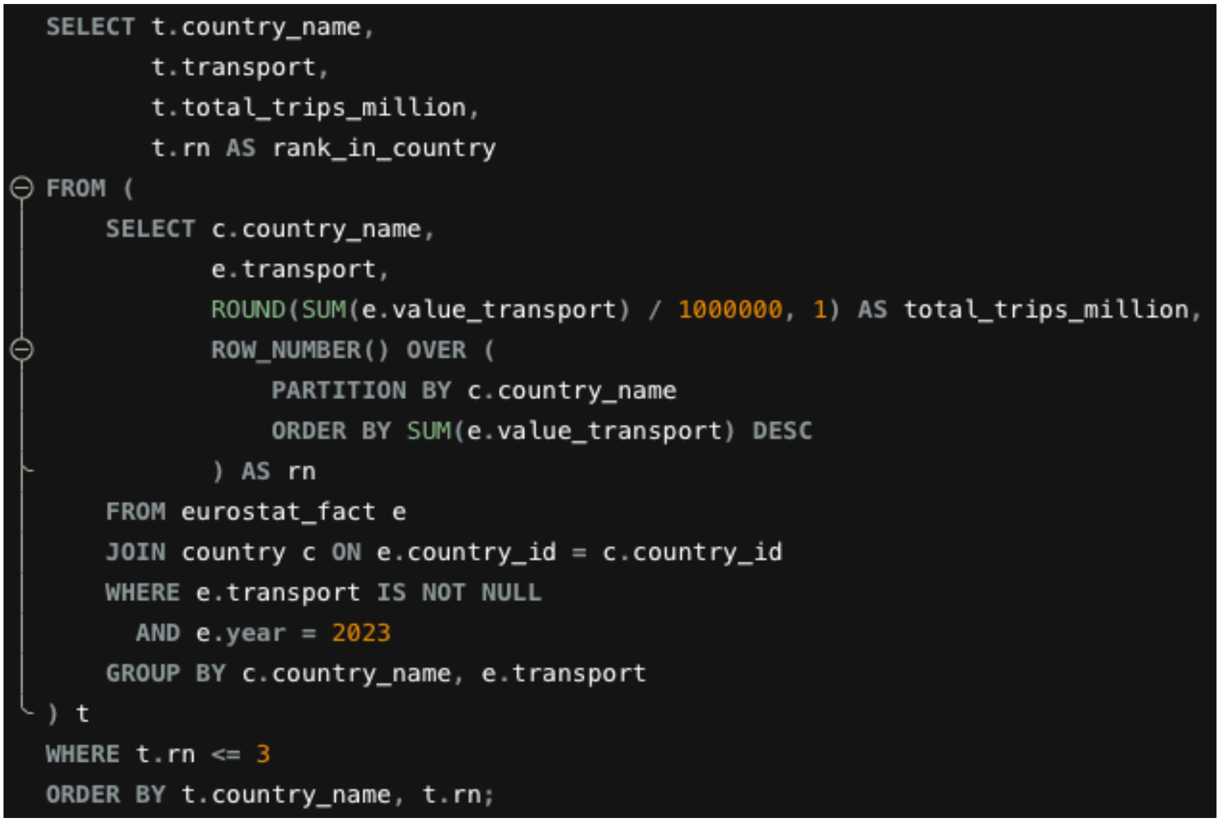


country_name	domestic_trips_mill...	international_trips_mill...	pct_domes...	pct_internatio...
Spain	354.59857000	26.35022300	93.1	6.9
France	594.99988100	56.03446200	91.4	8.6
Italy	87.30098800	10.05006100	89.7	10.3
Germany	405.38739900	173.11684900	70.1	29.9

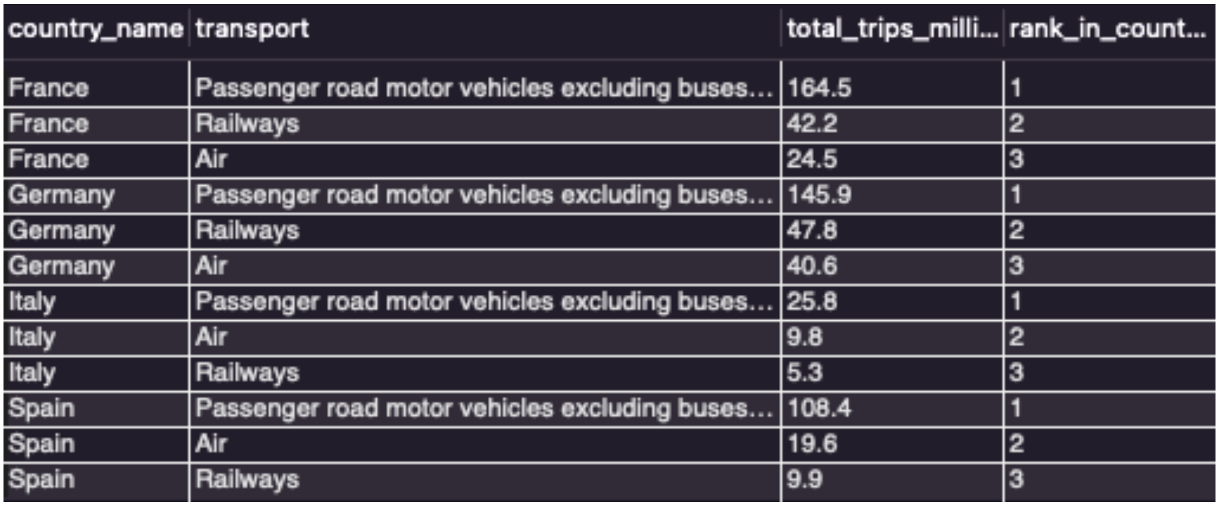
In 2023, most trips were inside each country. Spain, France, and Italy depended a lot on domestic travel. Germany was different, with more trips abroad than the others.

Conclusion: Domestic travel is the main driver, but Germany is more international than the Southern European countries.

7.5. Main transport Modes by Country



```
SELECT t.country_name,
       t.transport,
       t.total_trips_million,
       t.rn AS rank_in_country
FROM (
    SELECT c.country_name,
           e.transport,
           ROUND(SUM(e.value_transport) / 1000000, 1) AS total_trips_million,
           ROW_NUMBER() OVER (
               PARTITION BY c.country_name
               ORDER BY SUM(e.value_transport) DESC
           ) AS rn
    FROM eurostat_fact e
    JOIN country c ON e.country_id = c.country_id
    WHERE e.transport IS NOT NULL
      AND e.year = 2023
    GROUP BY c.country_name, e.transport
) t
WHERE t.rn <= 3
ORDER BY t.country_name, t.rn;
```



country_name	transport	total_trips_milli...	rank_in_count...
France	Passenger road motor vehicles excluding buses...	164.5	1
France	Railways	42.2	2
France	Air	24.5	3
Germany	Passenger road motor vehicles excluding buses...	145.9	1
Germany	Railways	47.8	2
Germany	Air	40.6	3
Italy	Passenger road motor vehicles excluding buses...	25.8	1
Italy	Air	9.8	2
Italy	Railways	5.3	3
Spain	Passenger road motor vehicles excluding buses...	108.4	1
Spain	Air	19.6	2
Spain	Railways	9.9	3

In 2023, cars had the highest number of trips in all four countries.

- France and Germany also show a strong use of trains compared to Spain and Italy.
- Spain depends much more on cars, with trains playing only a small role.
- Italy has fewer car trips than the other countries but a high number of trips by air.

Conclusion: Road travel is the main transport mode everywhere, but France and Germany rely more on trains, Spain is very car-focused, and Italy uses air travel more than the others.

7.6. Conclusion SQL scripts

The SQL analysis provides a broad picture of tourism and travel behavior in Europe. Prices vary strongly between capitals: Paris is clearly the most expensive destination, while Madrid is the most affordable. This confirms that accommodation costs play a major role in how travelers choose where to stay. Larger properties do not cost proportionally more per person, making them especially attractive for families and groups.

Travel purposes remain stable across countries, with personal trips dominating, followed by family visits and business travel. This stability shows that motivations are consistent and predictable, even after the Covid-19 pandemic. Looking at mobility patterns, most trips remain domestic in Spain, France, and Italy, while German residents stand out with a more international profile. Transport habits also differ: cars dominate everywhere, but France and Germany show strong rail usage, while Italy relies more heavily on air travel.

From a business and policy perspective, these results have clear implications. Airbnb and hosts can market larger properties as “best value for groups” and highlight affordable capitals such as Madrid and Rome to attract price-sensitive travelers. Tourism boards can adapt their focus: strengthening domestic markets in Southern Europe while recognizing the more international travel profile of Germany. Cities can also align infrastructure planning with traveler behavior, for example by investing in rail in France and Germany, or by managing air travel capacity in Italy.

In summary, the SQL results show that European tourism is shaped by stable motivations, strong domestic demand in Southern Europe, and a more international orientation in Germany. These findings help explain differences in competitiveness between cities and can support better decisions for businesses, policymakers, and urban planners.

8. Data via API : Flask

As part of my *Tourism Analytics Project*, I developed a Flask API to expose data stored in a MySQL database. I documented the API using Swagger (Flasgger), which provides an interactive interface to test the endpoints.

An API provides:

- Controlled access to the data (only exposing what I decide).
- Flexibility for integration with dashboards, machine learning models, or third-party tools.
- Scalability since queries can be filtered, paginated, and adapted to different use cases.

Technical stacks used:

- Flask: lightweight Python framework for building APIs.
- PyMySQL: connector to my MySQL database.
- Flasgger: automatically generates Swagger documentation for the API.
- .env file: keeps database credentials safe (instead of writing passwords in the code).

I implemented two resources with a total of four endpoints. Available endpoints:

1. Airbnb Listings

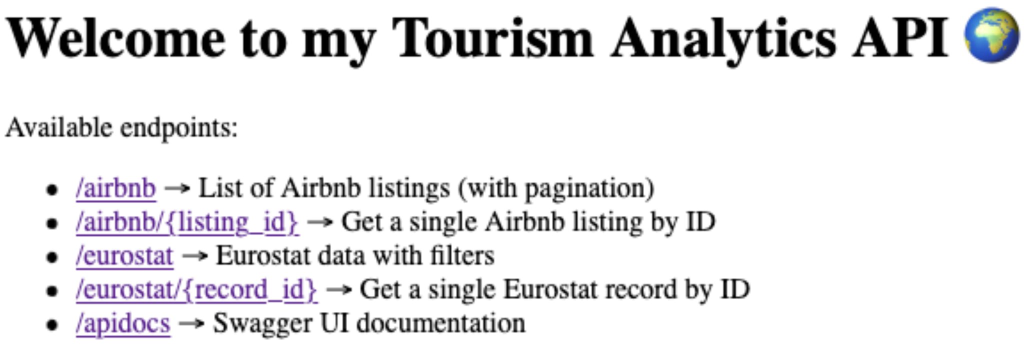
- GET /airbnb → Returns a list of Airbnb listings, with pagination (?limit=10).
  Example: http://127.0.0.1:5000/airbnb?limit=5
- GET /airbnb/{listing_id} → Returns the details of a single Airbnb listing.
  Example: http://127.0.0.1:5000/airbnb/262141

2. Eurostat Data

- GET /eurostat → Returns Eurostat tourism data, with optional filters on country and year.
  Example: http://127.0.0.1:5000/eurostat?country=France&year=2018
- GET /eurostat/{record_id} → Returns the details of a single Eurostat record.
  Example: http://127.0.0.1:5000/eurostat/1

Homepage

I also created a homepage (http://127.0.0.1:5000/) that lists all available endpoints and links directly to them.



Welcome to my Tourism Analytics API 🌍

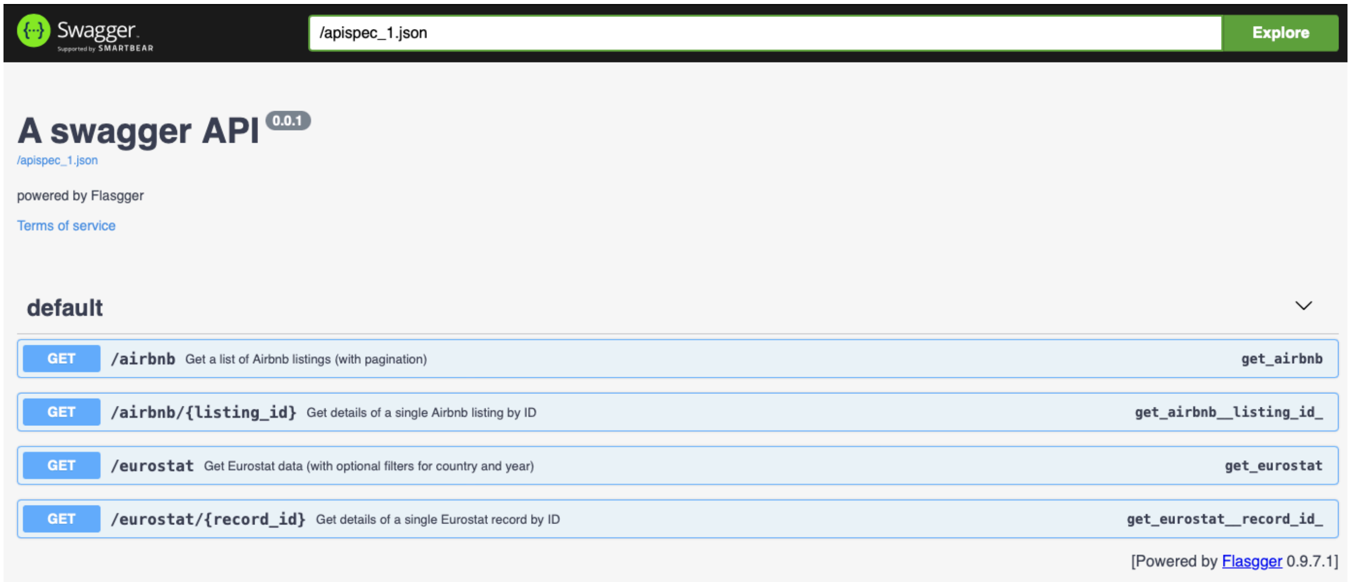
Available endpoints:

- /airbnb → List of Airbnb listings (with pagination)
- /airbnb/{listing_id} → Get a single Airbnb listing by ID
- /eurostat → Eurostat data with filters
- /eurostat/{record_id} → Get a single Eurostat record by ID
- /apidocs → Swagger UI documentation

Documentation (Swagger UI)

To make the API user-friendly, I added Swagger UI using the flasgger package.

- Swagger provides a visual interface where users can test the endpoints directly in their browser.
- Available at: http://127.0.0.1:5000/apidocs.



A swagger API 0.0.1
/apispec_1.json

powered by Flasgger

Terms of service

default

GET /airbnb Get a list of Airbnb listings (with pagination) get_airbnb

GET /airbnb/{listing_id} Get details of a single Airbnb listing by ID get_airbnb__listing_id_

GET /eurostat Get Eurostat data (with optional filters for country and year) get_eurostat

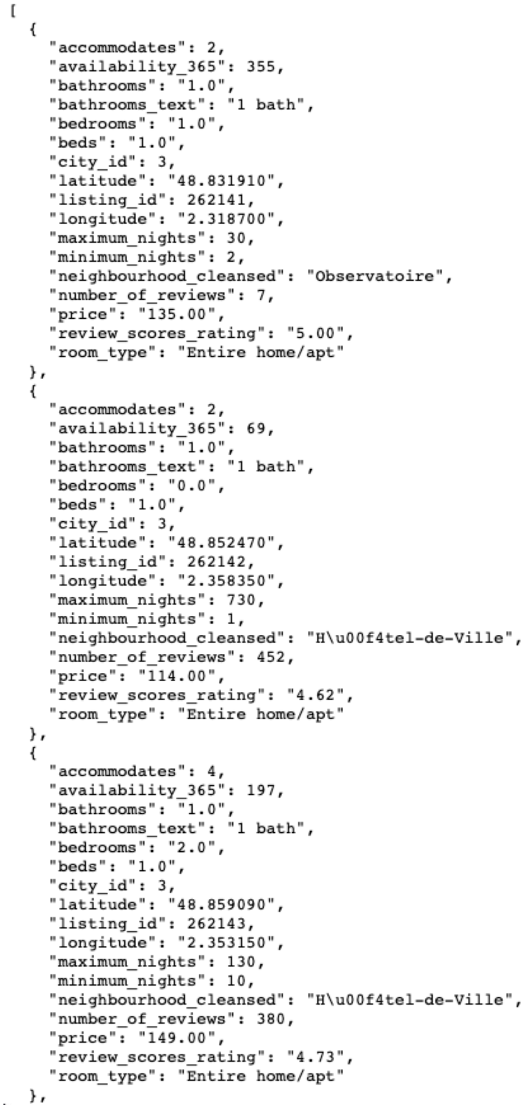
GET /eurostat/{record_id} Get details of a single Eurostat record by ID get_eurostat__record_id_

[Powered by Flasgger 0.9.7.1]

Example

When I access : http://127.0.0.1:5000/airbnb?limit=5

The API returns a JSON response like this :



```
[
  {
    "accommodates": 2,
    "availability_365": 355,
    "bathrooms": "1.0",
    "bathrooms_text": "1 bath",
    "bedrooms": "1.0",
    "beds": "1.0",
    "city_id": 3,
    "latitude": "48.831910",
    "listing_id": 262141,
    "longitude": "2.318700",
    "maximum_nights": 30,
    "minimum_nights": 2,
    "neighbourhood_cleansed": "Observatoire",
    "number_of_reviews": 7,
    "price": "135.00",
    "review_scores_rating": "5.00",
    "room_type": "Entire home/apt"
  },
  {
    "accommodates": 2,
    "availability_365": 69,
    "bathrooms": "1.0",
    "bathrooms_text": "1 bath",
    "bedrooms": "0.0",
    "beds": "1.0",
    "city_id": 3,
    "latitude": "48.852470",
    "listing_id": 262142,
    "longitude": "2.358350",
    "maximum_nights": 730,
    "minimum_nights": 1,
    "neighbourhood_cleansed": "H\u00f4tel-de-Ville",
    "number_of_reviews": 452,
    "price": "114.00",
    "review_scores_rating": "4.62",
    "room_type": "Entire home/apt"
  },
  {
    "accommodates": 4,
    "availability_365": 197,
    "bathrooms": "1.0",
    "bathrooms_text": "1 bath",
    "bedrooms": "2.0",
    "beds": "1.0",
    "city_id": 3,
    "latitude": "48.859090",
    "listing_id": 262143,
    "longitude": "2.353150",
    "maximum_nights": 130,
    "minimum_nights": 10,
    "neighbourhood_cleansed": "H\u00f4tel-de-Ville",
    "number_of_reviews": 380,
    "price": "149.00",
    "review_scores_rating": "4.73",
    "room_type": "Entire home/apt"
  },
```

This approach shows the value of APIs in data projects:

- Make the data accessible and reusable for different applications (dashboards, machine learning models, web apps, etc.),
- Provide flexibility, since users can request only the information they need,
- And ensure scalability, because the same API could later be deployed to a server or cloud platform to support multiple users.

While my API currently runs locally on 127.0.0.1:5000, it demonstrates the key concepts of modern data engineering: data integration, exposure through services, and documentation with Swagger to improve usability.

9. Machine Learning

9.1. Clustering

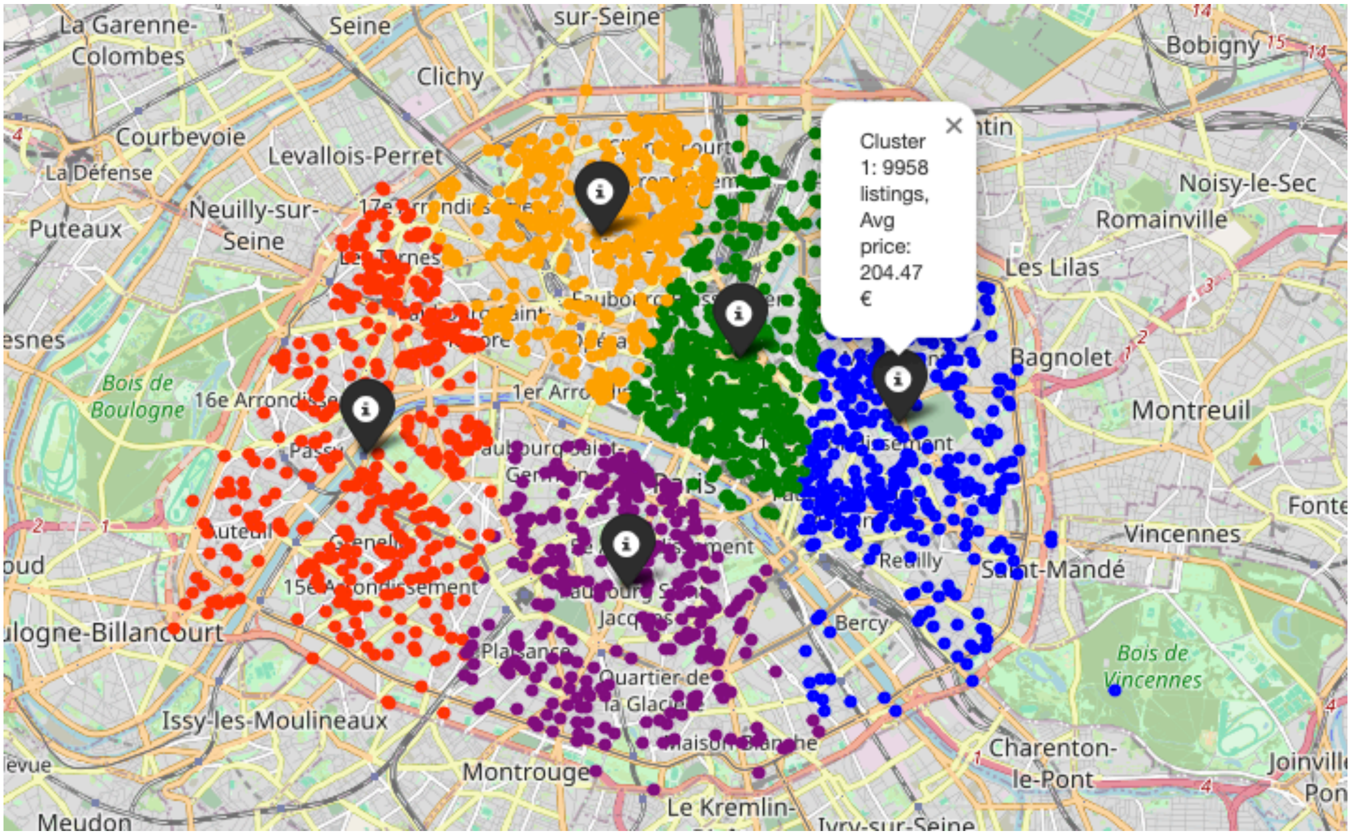
For the machine learning part, I applied a clustering model using the K-means algorithm on Airbnb listings in Paris. The goal was to identify natural groups of listings based on their geographical location and price.

In unsupervised learning, hyperparameter tuning is limited compared to supervised models. For K-means, the most important hyperparameter is the number of clusters (k). To determine the best value, I tested different values of k and used the silhouette score to evaluate the quality of clustering. This method showed that five clusters provide the best separation of the data.

The results revealed clear segmentation in the Paris Airbnb market:

- One premium cluster, with the highest prices (around 370 € on average).
- One budget cluster, with the lowest prices (around 200 €).
- Three mid-range clusters, representing the majority of listings with prices between 270 € and 285 €.

Conclusion: Hyperparameter tuning confirmed that 5 is the optimal number of clusters for this dataset. The clustering highlights different market segments (luxury, budget, mid-range), which can be useful for travelers to choose accommodation according to their budget, and for hosts to position their listings strategically



10. GDPR

Since my project involves tourism-related data (Airbnb listings, Eurostat, Numbeo, Worlddata, Wikipedia), I needed to make sure that the analysis respected the principles of the General Data Protection Regulation (GDPR), which applies in the European Union.

1. No personal data

- The datasets I used are aggregated (Eurostat, Worlddata) or anonymized (Airbnb public listings, Wikipedia, Numbeo).
- There are no names, emails, phone numbers, or other identifiers that could be linked to an individual.

2. Anonymization and pseudonymization

- Airbnb listings are identified by a numeric listing_id, but this cannot directly identify a natural person.
- This means the data is considered sufficiently anonymized for analytical purposes.

3. Data minimization

- I only extracted and stored the columns relevant for my analysis (e.g., price, accommodates, reviews, international arrivals, costs of living).
- This reduces the risk of holding unnecessary or sensitive information.

4. Transparency and sources

- All data sources are **open data** or publicly available (Eurostat, Worlddata, Wikipedia, Numbeo, Airbnb public data).
- In my report and database, I reference the original sources, which aligns with GDPR's requirement for transparency.

5. If this API were deployed

- Access control would be needed to ensure only authorized users can query the data.
- Logging and monitoring should be implemented to track usage.
- A privacy policy would be necessary to inform users how data is stored and processed.

11. Conclusion

This project combined several open data sources to study tourism in Europe. By linking Airbnb listings with Eurostat, WorldData, Wikipedia, and Numbeo, I built a complete picture of both supply and demand.

The results show that Airbnb prices are mostly shaped by room type, city, and availability. Entire homes and hotel rooms are the most expensive, Paris is the priciest city, and listings open all year can be managed by professional hosts. Review scores and capacity matter less, which means travelers focus more on location and type of accommodation than on ratings.

On the demand side, most trips are domestic and for personal reasons. German travelers are more international, while transport habits differ: trains are stronger in France and Germany, cars dominate in Spain, and Italy relies more on air travel.

The clustering of Paris listings revealed three clear markets: budget, mid-range, and premium. This helps Airbnb and hosts adjust their pricing, travelers compare options, and cities monitor the pressure of rentals on housing.

One limitation is the variable availability_365. It only shows how many days the host keeps the listing open, not how many days it is actually rented. And in some cities like Paris, being open all year is often against the rules. This makes the variable useful to spot professional or full-time hosts, but it must be interpreted with care.

In conclusion, these findings provide real opportunities:

- Airbnb and hosts can better target travelers by price sensitivity and property type.
- Tourism boards can adapt strategies based on domestic vs international demand.
- Cities can design smarter rules for short-term rentals and plan infrastructure in line with real travel habits.

This project shows that open data, when combined and analyzed, can turn scattered information into clear insights that support better business and policy decisions.