Webscraping Project for the Restaurants in a Public Website

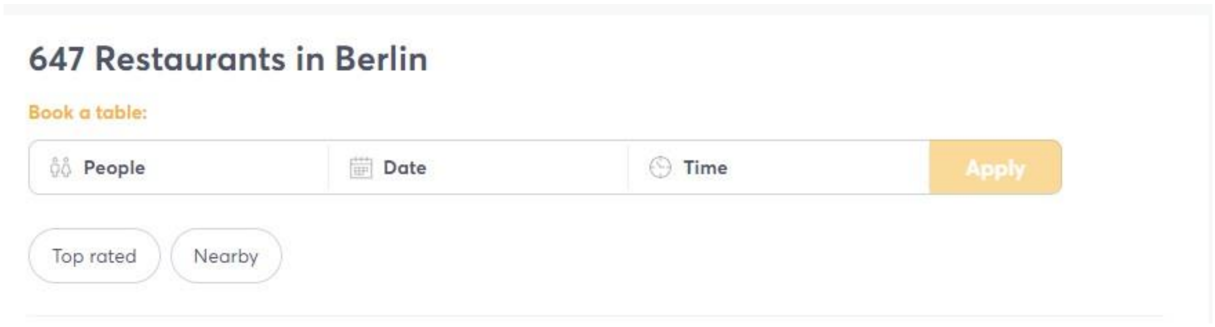
Data Source:

I tried webscraping the Quandoo Restaurants Booking Webpage and have documented the process.

I tried to handle different edge cases:

a. Best Case Scenario – Berlin – where there are 33 pages of results.
b. Best Case Scenario – Frankfurt - where there are 5 pages of results.
c. Edge Case Scenario – Rostock – where there is just 1 page result.
d. Worst Case Scenario - Paris – no result.

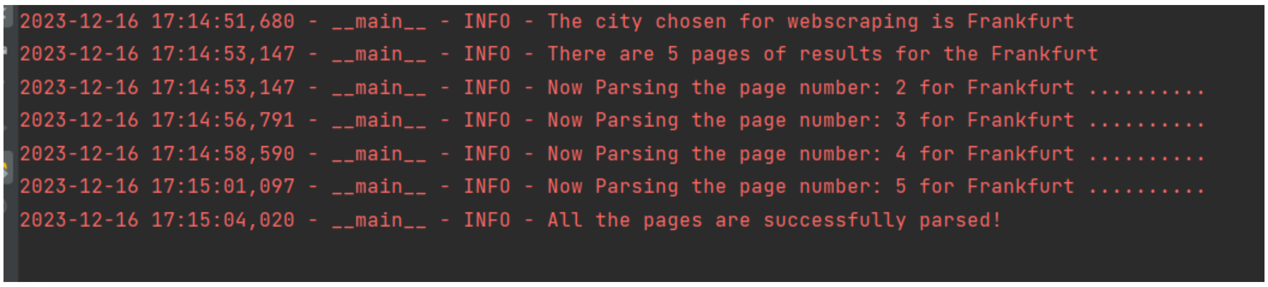
As of December 16th 2023, Quandoo had 647 restaurants enlisted for Berlin. The webscraping also generated the same number of results.



647 Restaurants in Berlin

Fig 1: Berlin Restaurant Webpage (https://www.quandoo.de/en/result?destination=berlin) taken on 16.11.23

I added logging to know the progress of the scraping task and helped me to navigate to the right page in case of failure in the scraping process due to unexpected behavior (missing data).



```
2023-12-16 17:14:51,680 - __main__ - INFO - The city chosen for webscraping is Frankfurt
2023-12-16 17:14:53,147 - __main__ - INFO - There are 5 pages of results for the Frankfurt
2023-12-16 17:14:53,147 - __main__ - INFO - Now Parsing the page number: 2 for Frankfurt ..........
2023-12-16 17:14:56,791 - __main__ - INFO - Now Parsing the page number: 3 for Frankfurt ..........
2023-12-16 17:14:58,590 - __main__ - INFO - Now Parsing the page number: 4 for Frankfurt ..........
2023-12-16 17:15:01,097 - __main__ - INFO - Now Parsing the page number: 5 for Frankfurt ..........
2023-12-16 17:15:04,020 - __main__ - INFO - All the pages are successfully parsed!
```

Fig 2: Logging records generated while scraping the restaurant data for Frankfurt

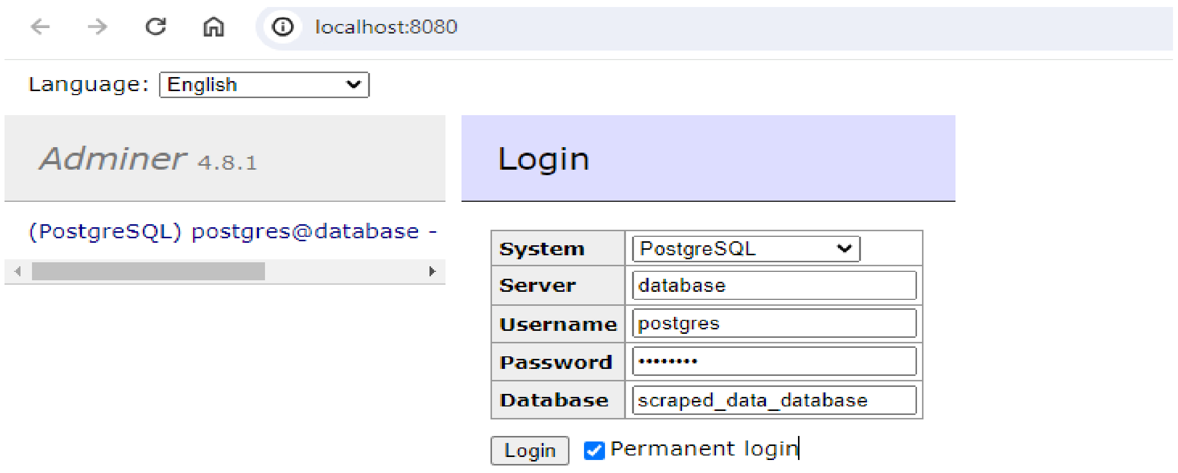
To run the webscraping script:

a. Please navigate to the directory ‘1.quandoo_restaurants_scraper’
b. It contains ‘quandoo_webscraper_app.py’ script, Dockerfile, docker-compose file, ‘scraped_data_results’ directory to store the result files generated after scraping.
c. Docker commands to run:
   1. docker-compose build
   2. docker-compose up -d
   3. The script can be run independently too, no input required.
d. The generated files are placed automatically inside the folder called “scraped_data_results”.

Postgres Database:

To spin up a postgres database:

a. Please navigate to ‘2.database_setup’ directory.
b. It contains docker-compose file, postgres_connection.py script and quandoo_berlin_results.csv file.
c. Docker commands to run:
   1. docker-compose build
   2. docker-compose up -d
   3. Please log on to localhost:8080 to see the database homepage.
d. The database login page appears like this:


localhost:8080

Language: English

Adminer 4.8.1

(PostgreSQL) postgres@database -

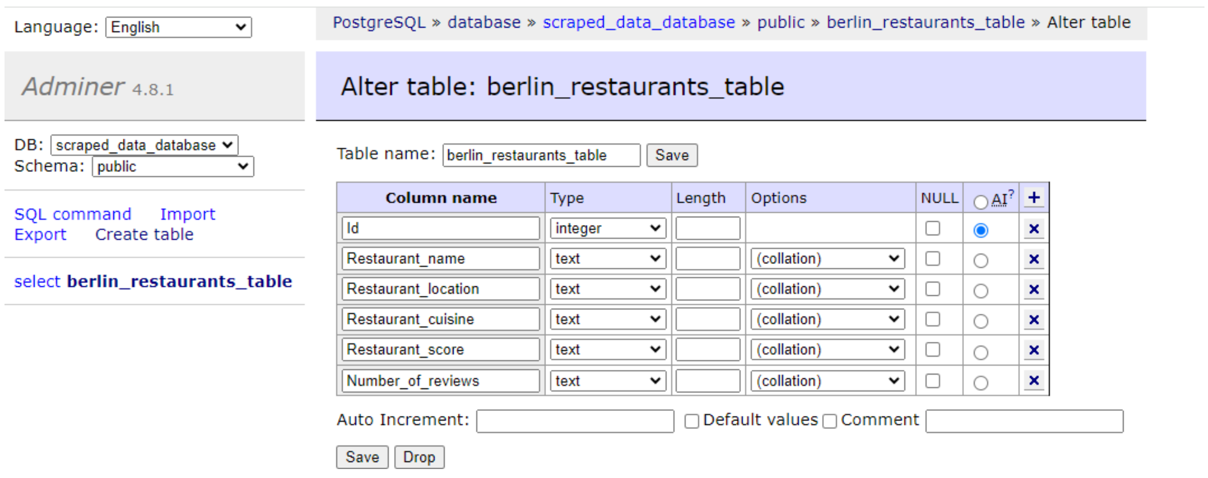
Login

System	PostgreSQL
Server	database
Username	postgres
Password	••••••••
Database	scraped_data_database

Login [✓ Permanent login]

Fig 3: Loginpage of the database

e. After login, I created a table called “berlin_restaurants_table” in the database.



PostgreSQL » database » scraped_data_database » public » berlin_restaurants_table » Alter table

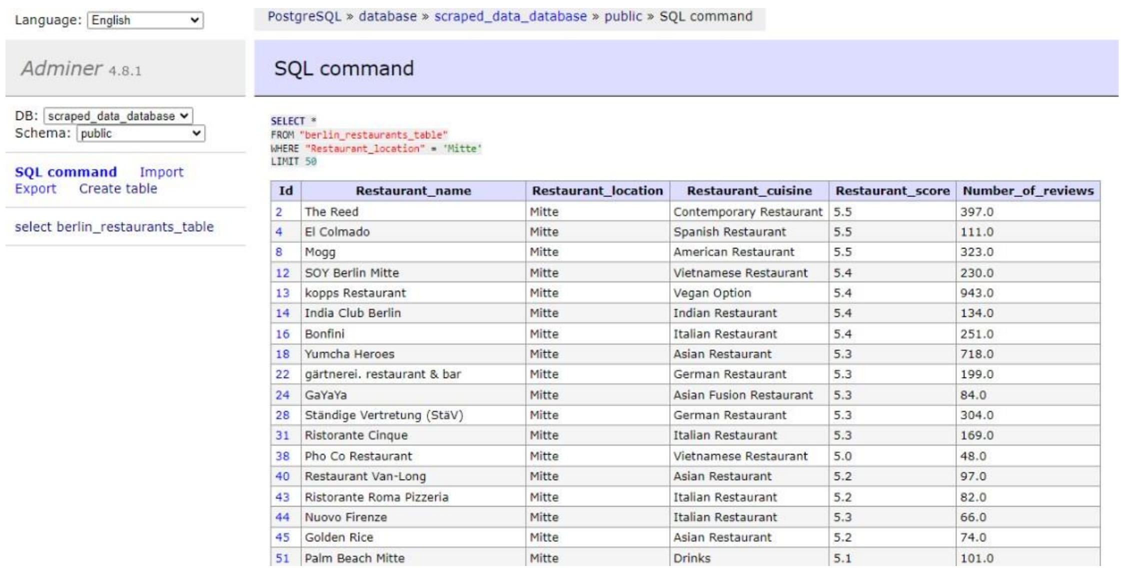
Alter table: berlin_restaurants_table

Table name: berlin_restaurants_table

Column name	Type	Length	Options	NULL	AI
Id	integer			☐	●
Restaurant_name	text		(collation)	☐	○
Restaurant_location	text		(collation)	☐	○
Restaurant_cuisine	text		(collation)	☐	○
Restaurant_score	text		(collation)	☐	○
Number_of_reviews	text		(collation)	☐	○

Fig 4: berlin_restaurants_table in the database

f. After populating the table with the data from the csv file, it looked like:



PostgreSQL » database » scraped_data_database » public » SQL command

SQL command

```
SELECT *
FROM "berlin_restaurants_table"
WHERE "Restaurant_location" = 'Mitte'
LIMIT 50
```

Id	Restaurant_name	Restaurant_location	Restaurant_cuisine	Restaurant_score	Number_of_reviews
2	The Reed	Mitte	Contemporary Restaurant	5.5	397.0
4	El Colmado	Mitte	Spanish Restaurant	5.5	111.0
8	Mogg	Mitte	American Restaurant	5.5	323.0
12	SOY Berlin Mitte	Mitte	Vietnamese Restaurant	5.4	230.0
13	kopps Restaurant	Mitte	Vegan Option	5.4	943.0
14	India Club Berlin	Mitte	Indian Restaurant	5.4	134.0
16	Bonfini	Mitte	Italian Restaurant	5.4	251.0
18	Yumcha Heroes	Mitte	Asian Restaurant	5.3	718.0
22	gärtnerei. restaurant & bar	Mitte	German Restaurant	5.3	199.0
24	GaYaYa	Mitte	Asian Fusion Restaurant	5.3	84.0
28	Ständige Vertretung (StäV)	Mitte	German Restaurant	5.3	304.0
31	Ristorante Cinque	Mitte	Italian Restaurant	5.3	169.0
38	Pho Co Restaurant	Mitte	Vietnamese Restaurant	5.0	48.0
40	Restaurant Van-Long	Mitte	Asian Restaurant	5.2	97.0
43	Ristorante Roma Pizzeria	Mitte	Italian Restaurant	5.2	82.0
44	Nuovo Firenze	Mitte	Italian Restaurant	5.3	66.0
45	Golden Rice	Mitte	Asian Restaurant	5.2	74.0
51	Palm Beach Mitte	Mitte	Drinks	5.1	101.0

Fig 5: Simple querying in the database

g. The script called 'postgres_connection.py' can be used to add data and fetch the data.

I started with the data modeling (ER Diagram) that is one of the starting point for the "Transformation" task of DBT.

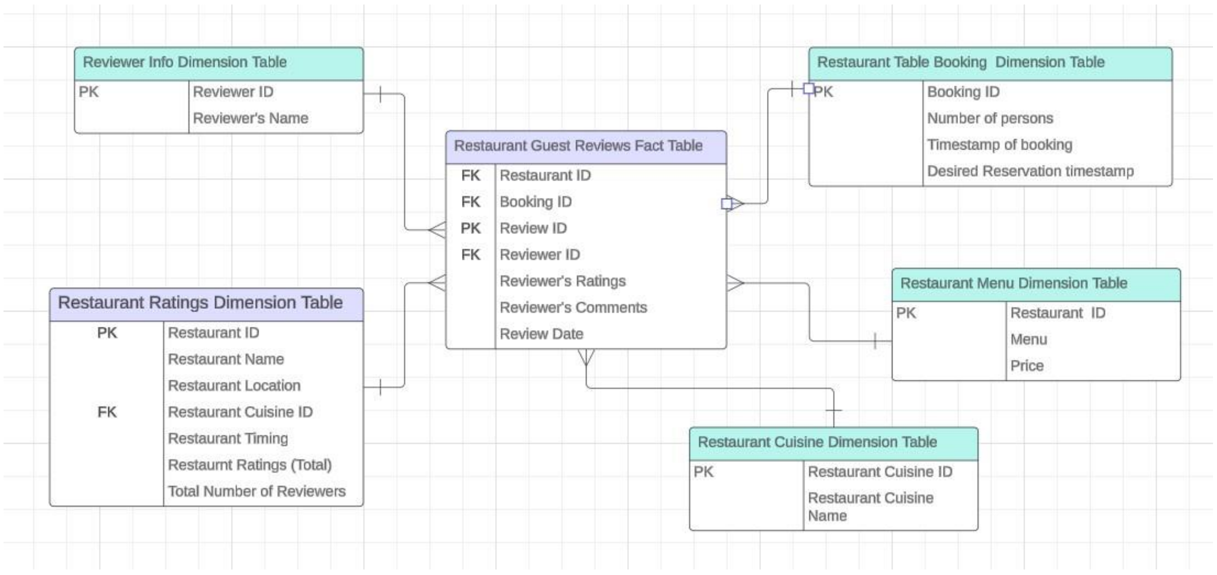


Fig 6: ERD of Restaurants Data in Data warehouse (My assumption)

Visualization of the results:

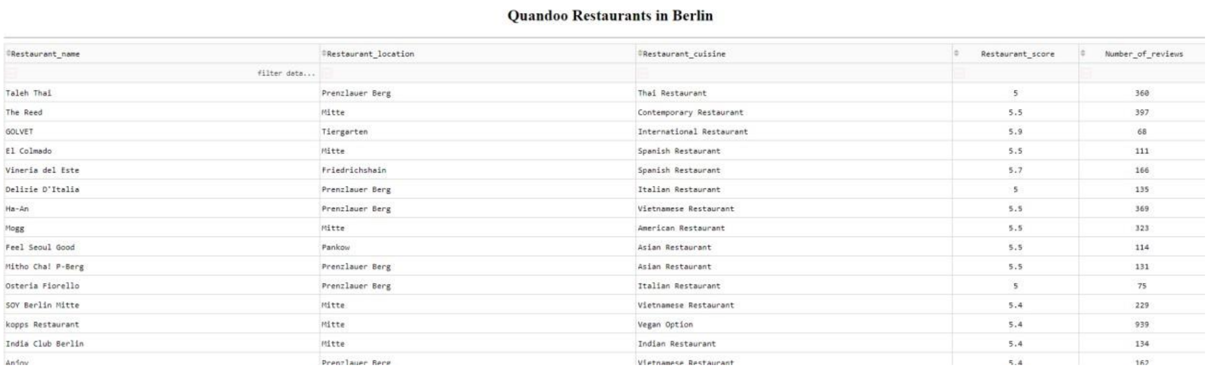
For this task, I tried to create a simple dashboard using Plotly dash with the scraped data of Berlin.

The dashboard has:

1. Data table to see/filter the scraped results.
2. Scatter plot that represents the Berliner's top 10 highest review for each kind of cuisine.
3. Scatter plot that represents the Berliner's top 10 highest review for each locality.
4. Bar plot that represents the highest number of restaurants in each locality.

To run the visualization script:

a. Please navigate to '3.quandoo_dashboard'.
b. It contains 'berlin_restaurants_viz.py' script, docker-compose file, Dockerfile and the directory 'scraped_data' containing the source file.
c. Docker commands to run:
    1. docker-compose build
    2. docker-compose up -d
    3. Please log on to http://127.0.0.1:9002/ see the dashboard live.
    4. The script can also be run independently.



Quandoo Restaurants in Berlin

Restaurant_name	Restaurant_location	Restaurant_cuisine	Restaurant_score	Number_of_reviews
filter data...				
Taleh Thai	Prenzlauer Berg	Thai Restaurant	5	360
The Reed	Mitte	Contemporary Restaurant	5.5	397
GOLVET	Tiergarten	International Restaurant	5.9	68
El Colmado	Mitte	Spanish Restaurant	5.5	111
Vineria del Este	Friedrichshain	Spanish Restaurant	5.7	166
Delizie D'Italia	Prenzlauer Berg	Italian Restaurant	5	135
Ha-An	Prenzlauer Berg	Vietnamese Restaurant	5.5	369
Mogg	Mitte	American Restaurant	5.5	323
Feel Seoul Good	Pankow	Asian Restaurant	5.5	114
Mitho Cha! P-Berg	Prenzlauer Berg	Asian Restaurant	5.5	131
Osteria Fiorello	Prenzlauer Berg	Italian Restaurant	5	75
SOY Berlin Mitte	Mitte	Vietnamese Restaurant	5.4	229
kopps Restaurant	Mitte	Vegan Option	5.4	939
India Club Berlin	Mitte	Indian Restaurant	5.4	134
Anjoy	Prenzlauer Berg	Vietnamese Restaurant	5.4	162

Fig 7: Data table in the dashboard

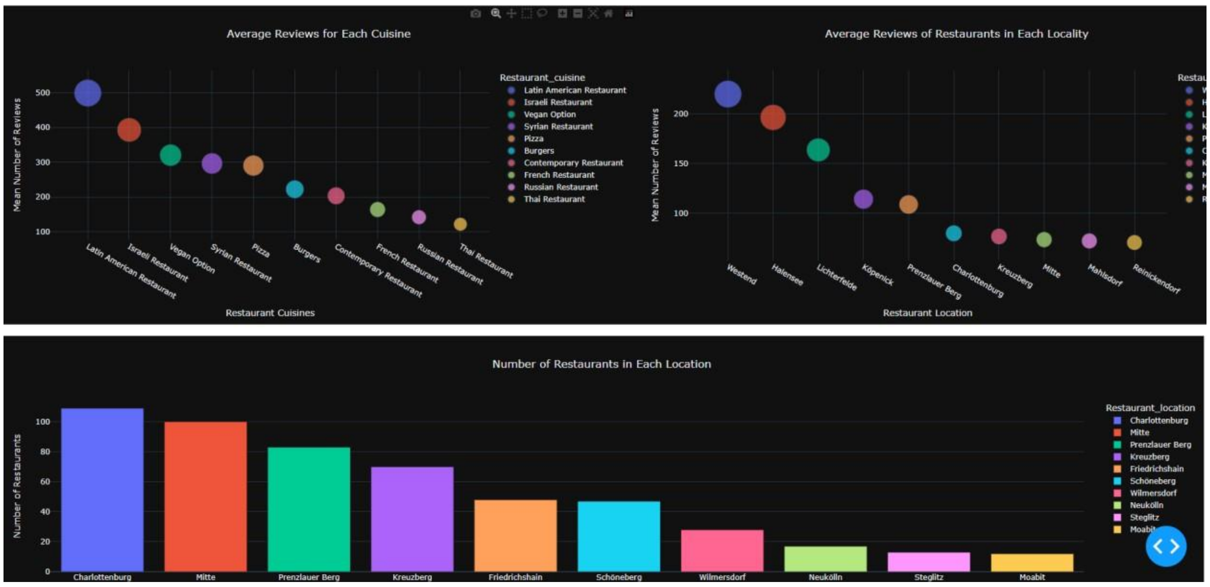


Fig 8: Plots in the dashboard

References:

1. https://python.plainenglish.io/how-to-run-a-python-file-inside-a-docker-container-e17668891c85
2. https://python.plainenglish.io/dockerizing-plotly-dash-5c23009fc10b
3. https://www.youtube.com/watch?v=nlk0QlPdbcY
4. https://docs.getdbt.com/terms/dimensional-modeling
5. https://docs.getdbt.com/blog/kimball-dimensional-model