Northeastern University

P3: Implementation

Boston Blue Bikes Analysis

Project Group 15

Anjali Kabra

Jhalak Surve

Krishna Kapadia

Shubham Shah

The objective of our database project is to analyze the blue bikes data to get some insights about the most used stations, most frequent routes, trip durations and the overall usage of the bikes. A few of the many questions we would like to answer through our analysis would be:

- What is the average duration for which a rider rents a bike?
- What is the most frequently accessed station?
- What are the rush hours?
- What are the busy blue bike days in the week?
- What is the ride count against the riders' age and gender?

DATA MODEL: ArangoDB Multi-Model (Graph + Document + Key value)

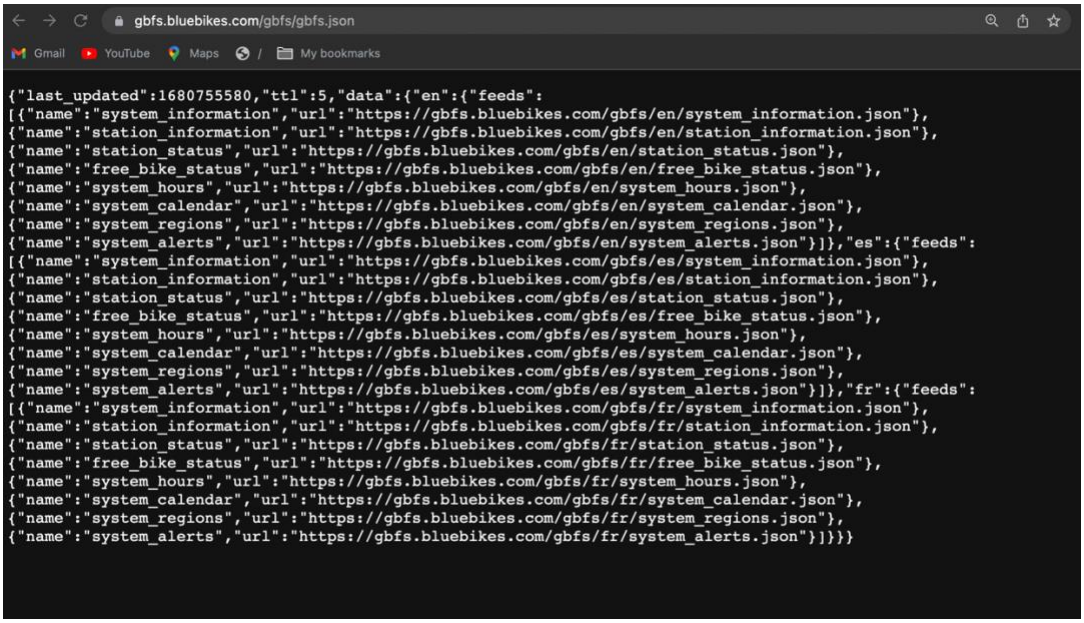
PLATFORM: Local

Brief description of the implementation process:

- For the implementation, we have created a **data pipeline using python** to load the data into our database. (I have also submitted the jupyter notebook file)

- We are using the data from the Boston Blue Bikes official website. https://www.bluebikes.com/system-data

- Bluebikes publishes real-time system data in open General Bikeshare Feed Specification (GBFS) format. https://gbfs.bluebikes.com/gbfs/gbfs.json




```
{"last_updated":1680755580,"ttl":5,"data":{"en":{"feeds":
[{"name":"system_information","url":"https://gbfs.bluebikes.com/gbfs/en/system_information.json"},
{"name":"station_information","url":"https://gbfs.bluebikes.com/gbfs/en/station_information.json"},
{"name":"station_status","url":"https://gbfs.bluebikes.com/gbfs/en/station_status.json"},
{"name":"free_bike_status","url":"https://gbfs.bluebikes.com/gbfs/en/free_bike_status.json"},
{"name":"system_hours","url":"https://gbfs.bluebikes.com/gbfs/en/system_hours.json"},
{"name":"system_calendar","url":"https://gbfs.bluebikes.com/gbfs/en/system_calendar.json"},
{"name":"system_regions","url":"https://gbfs.bluebikes.com/gbfs/en/system_regions.json"},
{"name":"system_alerts","url":"https://gbfs.bluebikes.com/gbfs/en/system_alerts.json"}]},"es":{"feeds":
[{"name":"system_information","url":"https://gbfs.bluebikes.com/gbfs/es/system_information.json"},
{"name":"station_information","url":"https://gbfs.bluebikes.com/gbfs/es/station_information.json"},
{"name":"station_status","url":"https://gbfs.bluebikes.com/gbfs/es/station_status.json"},
{"name":"free_bike_status","url":"https://gbfs.bluebikes.com/gbfs/es/free_bike_status.json"},
{"name":"system_hours","url":"https://gbfs.bluebikes.com/gbfs/es/system_hours.json"},
{"name":"system_calendar","url":"https://gbfs.bluebikes.com/gbfs/es/system_calendar.json"},
{"name":"system_regions","url":"https://gbfs.bluebikes.com/gbfs/es/system_regions.json"},
{"name":"system_alerts","url":"https://gbfs.bluebikes.com/gbfs/es/system_alerts.json"}]},"fr":{"feeds":
[{"name":"system_information","url":"https://gbfs.bluebikes.com/gbfs/fr/system_information.json"},
{"name":"station_information","url":"https://gbfs.bluebikes.com/gbfs/fr/station_information.json"},
{"name":"station_status","url":"https://gbfs.bluebikes.com/gbfs/fr/station_status.json"},
{"name":"free_bike_status","url":"https://gbfs.bluebikes.com/gbfs/fr/free_bike_status.json"},
{"name":"system_hours","url":"https://gbfs.bluebikes.com/gbfs/fr/system_hours.json"},
{"name":"system_calendar","url":"https://gbfs.bluebikes.com/gbfs/fr/system_calendar.json"},
{"name":"system_regions","url":"https://gbfs.bluebikes.com/gbfs/fr/system_regions.json"},
{"name":"system_alerts","url":"https://gbfs.bluebikes.com/gbfs/fr/system_alerts.json"}]}}}
```

- We are using this GBFS feed to fetch data related to the regions, stations and station status. For this purpose, we've used python.

  Firstly, we have used the **python library 'arango'** to connect to the arangodb using python.

  Then, we have defined four collections – **regions, stations, station_status and trips(edge)**

  Then, we are fetching the data using **requests.get()** method and saving the json file locally.

jupyter BostonBlueBikesProject Last Checkpoint: 2 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

Run Code

In [41]:
```
from arango import ArangoClient
```

In [42]:
```
import requests
```

In [43]:
```
#connecting to arango db
client = ArangoClient(hosts='http://localhost:8529')
db = client.db('BlueBikesProject',username='root', password='root')
```

In [ ]:
```
# Defining the collections
region_col = db.create_collection('regions')
station_col = db.create_collection('stations')
status_col = db.create_collection('station_status')
trip_col = db.create_collection('trips', edge=True)
```

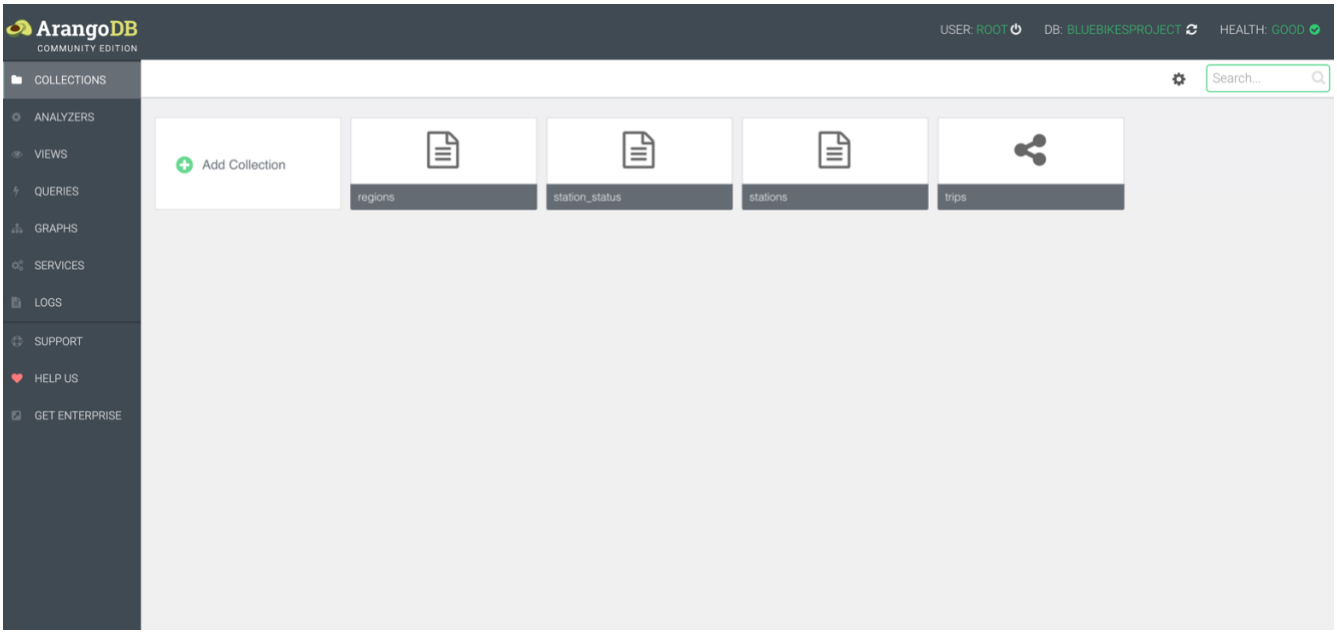
In [45]:
```
# Downloading the data from GBFS
gbfs_url = 'https://gbfs.bluebikes.com/gbfs/gbfs.json'
gbfs_data = requests.get(gbfs_url).json()
```

In [46]:
```
# Import regions data
region_url = None
for feed in gbfs_data['data']['en']['feeds']:
    if feed['name'] == 'system_regions':
        region_url = feed['url']
        break
```

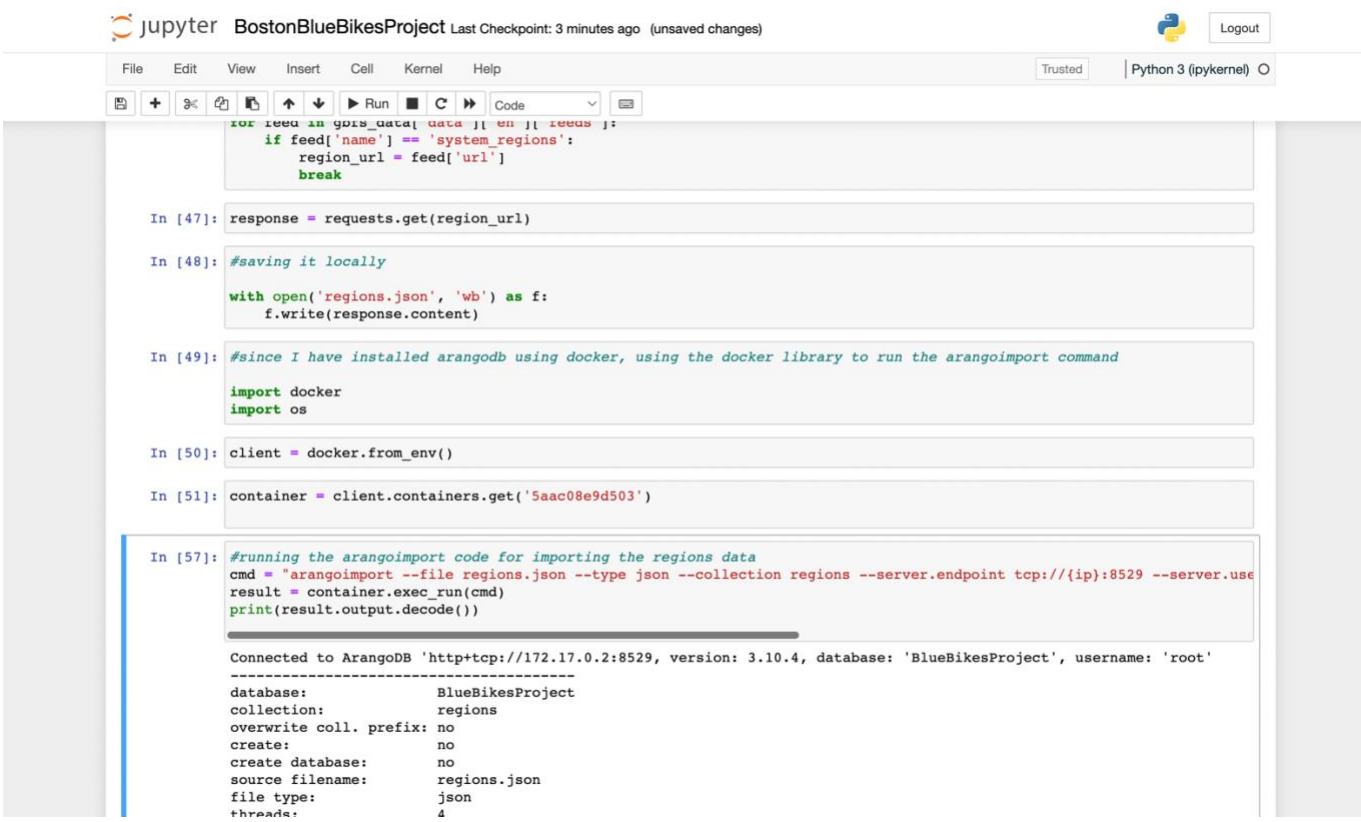
In [47]:
```
response = requests.get(region_url)
```

In [48]:
```
#saving it locally

with open('regions.json', 'wb') as f:
    f.write(response.content)
```

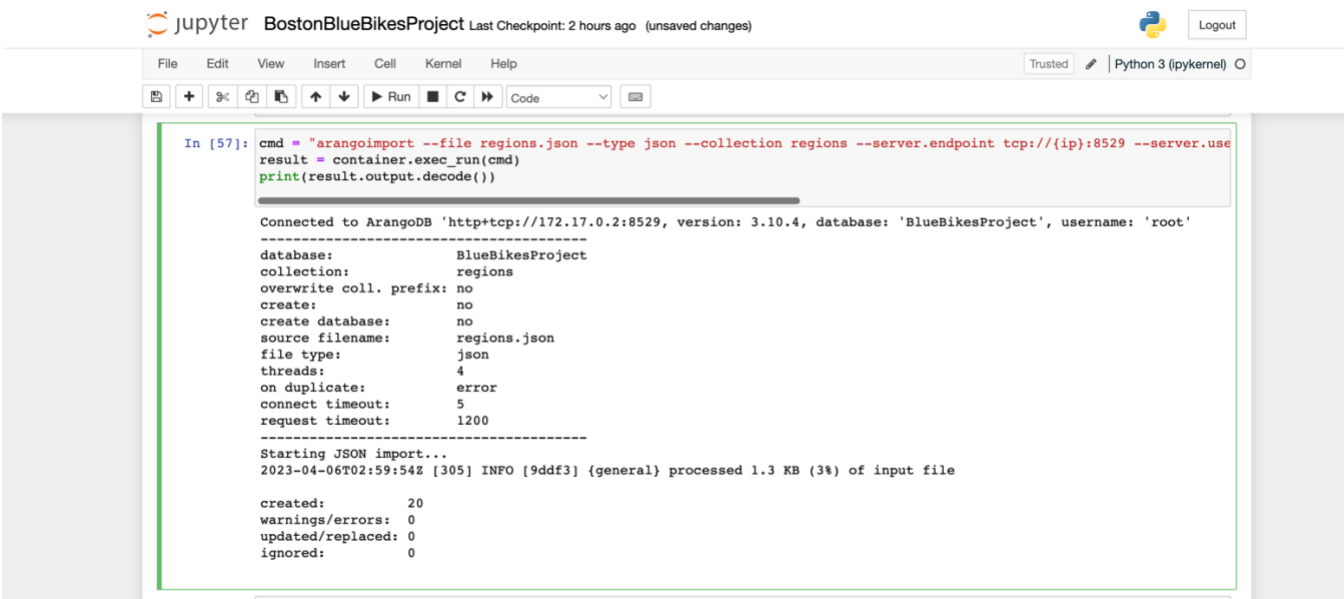
In [49]:
```
#since I have installed arangodb using docker, using the docker library to run the arangoimport command
```

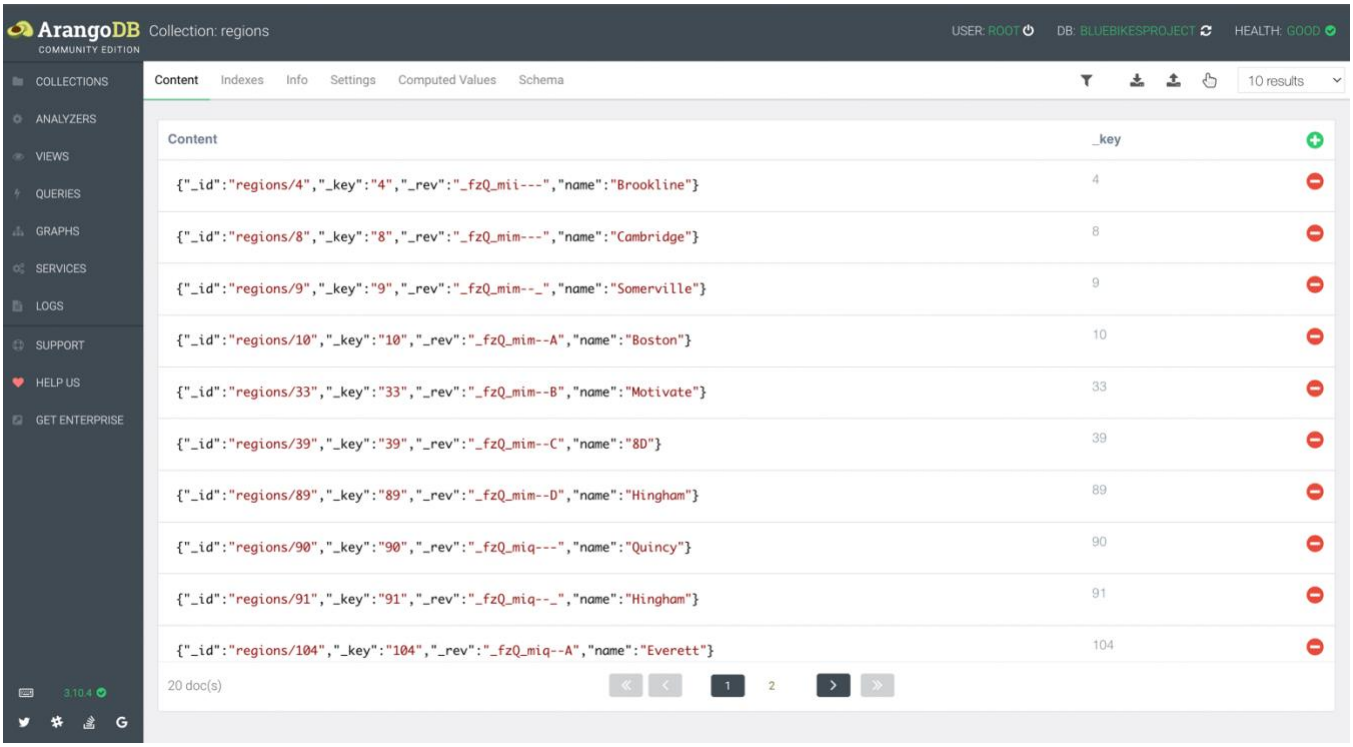


- Then, we have used the **arangoimport** tool to import the json data into collections. We have written python code for the same.



Importing regions data:





Importing stations data:

jupyter BostonBlueBikesProject Last Checkpoint: 5 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel)

Run Code

```
In [59]: response = requests.get(station_url)

In [60]: #saving it locally

         with open('stations.json', 'wb') as f:
             f.write(response.content)

In [61]: #running the arangoimport code for importing the stations data
         cmd = "arangoimport --file stations.json --type json --collection stations --server.endpoint tcp://{ip}:8529 --server.u
         result = container.exec_run(cmd)
         print(result.output.decode())

         Connected to ArangoDB 'http+tcp://172.17.0.2:8529, version: 3.10.4, database: 'BlueBikesProject', username: 'root'
         ----------------------------------------
         database:               BlueBikesProject
         collection:             stations
         overwrite coll. prefix: no
         create:                 no
         create database:        no
         source filename:        stations.json
         file type:              json
         threads:                4
         on duplicate:           error
         connect timeout:        5
         request timeout:        1200
         ----------------------------------------
         Starting JSON import...
         2023-04-06T03:11:31Z [317] INFO [9ddf3] {general} processed 288.8 KB (3%) of input file

         created:          453
         warnings/errors:  0
         updated/replaced: 0
         ignored:          0
```

ArangoDB COMMUNITY EDITION Collection: stations

USER: ROOT DB: BLUEBIKESPROJECT HEALTH: GOOD

COLLECTIONS
ANALYZERS
VIEWS
QUERIES
GRAPHS
SERVICES
LOGS
SUPPORT
HELP US
GET ENTERPRISE

Content Indexes Info Settings Computed Values Schema

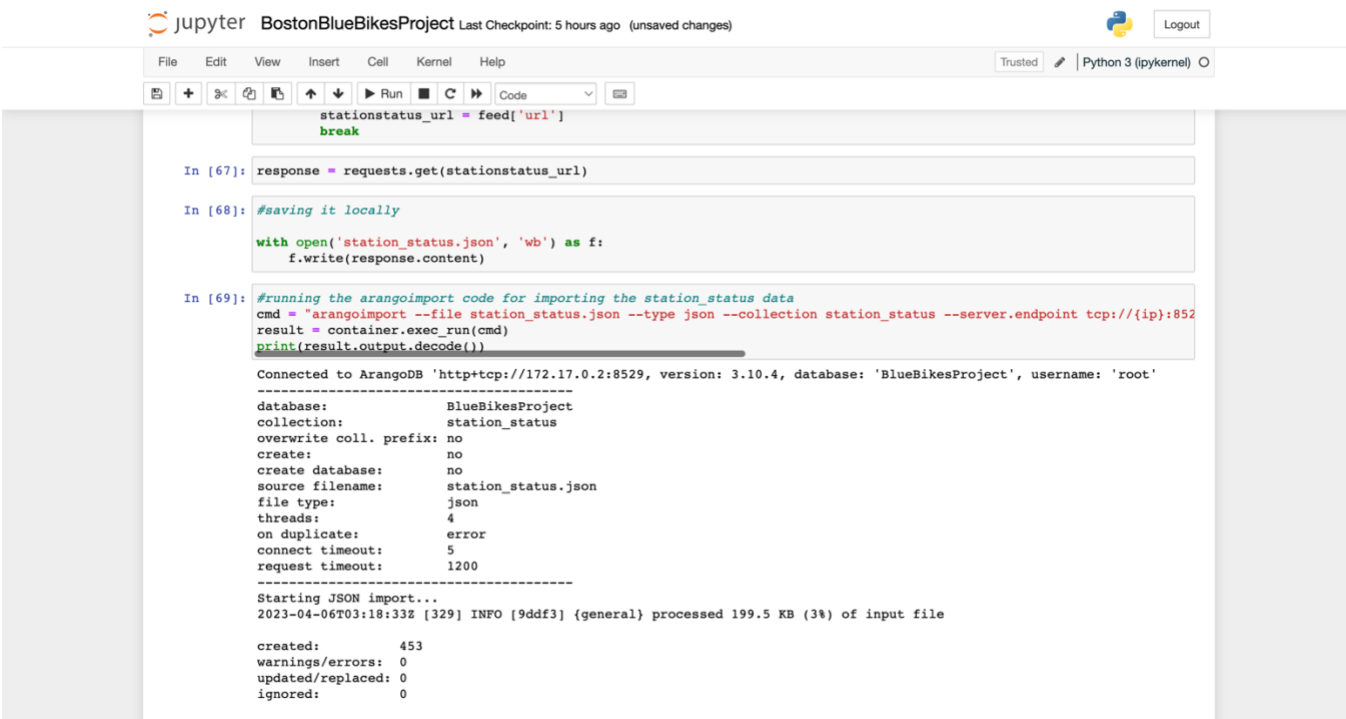
10 results

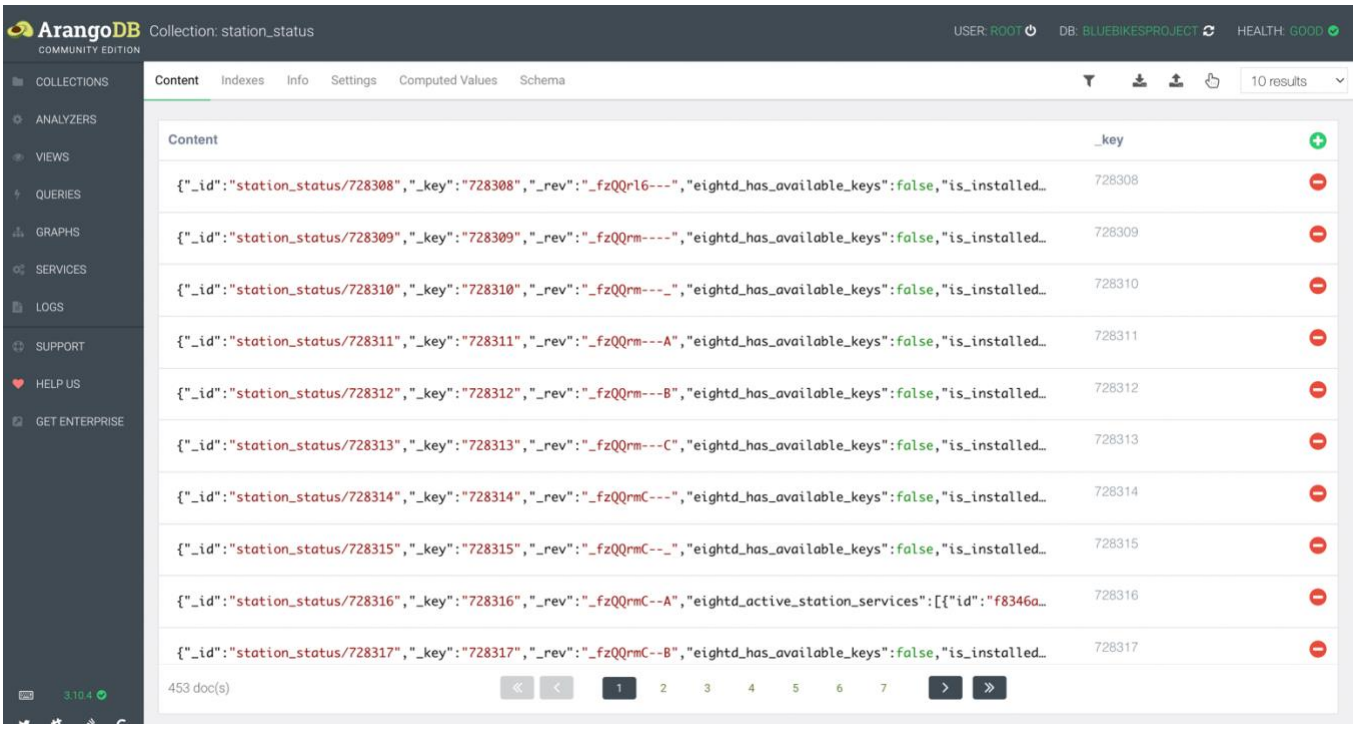
Content	_key
{"_id":"stations/3","_key":"3","_rev":"_fzQKPge---","capacity":15,"eightd_has_key_dispenser":true,"eightd_station_s…	3
{"_id":"stations/4","_key":"4","_rev":"_fzQKPge--_","capacity":19,"eightd_has_key_dispenser":false,"eightd_station_…	4
{"_id":"stations/5","_key":"5","_rev":"_fzQKPgi---","capacity":15,"eightd_has_key_dispenser":false,"eightd_station_…	5
{"_id":"stations/6","_key":"6","_rev":"_fzQKPgi--_","capacity":15,"eightd_has_key_dispenser":false,"eightd_station_…	6
{"_id":"stations/7","_key":"7","_rev":"_fzQKPgi--A","capacity":15,"eightd_has_key_dispenser":false,"eightd_station_…	7
{"_id":"stations/8","_key":"8","_rev":"_fzQKPgi--B","capacity":19,"eightd_has_key_dispenser":false,"eightd_station_…	8
{"_id":"stations/9","_key":"9","_rev":"_fzQKPgi--C","capacity":15,"eightd_has_key_dispenser":false,"eightd_station_…	9
{"_id":"stations/10","_key":"10","_rev":"_fzQKPgm---","capacity":11,"eightd_has_key_dispenser":false,"eightd_statio…	10
{"_id":"stations/11","_key":"11","_rev":"_fzQKPgm--_","capacity":15,"eightd_has_key_dispenser":false,"eightd_statio…	11
{"_id":"stations/12","_key":"12","_rev":"_fzQKPgm--A","capacity":26,"eightd_has_key_dispenser":true,"eightd_station…	12

453 doc(s) 1 2 3 4 5 6 7

3.10.4

Importing station_status data:





- Now, we're not getting the trips data from the GBFS feed, so we have used **existing datasets for trips data** from the blue bikes official website, which is in **csv format**.

- We've **cleaned** this csv data using **python pandas library.**

jupyter BostonBlueBikesProject Last Checkpoint: 5 hours ago (autosaved)

File Edit View Insert Cell Kernel Help

```
In [70]: #cleaning the trip data in the csv form

         import pandas as pd

In [71]: df = pd.read_csv('tripdata022023.csv')

In [72]: #checking for missing values

         df.isnull().sum()

Out[72]: tripduration                 0
         starttime                    0
         stoptime                     0
         start station id             0
         start station name           0
         start station latitude       0
         start station longitude      0
         end station id               0
         end station name             0
         end station latitude         0
         end station longitude        0
         bikeid                       0
         usertype                     0
         postal code               9696
         dtype: int64

In [ ]:
```

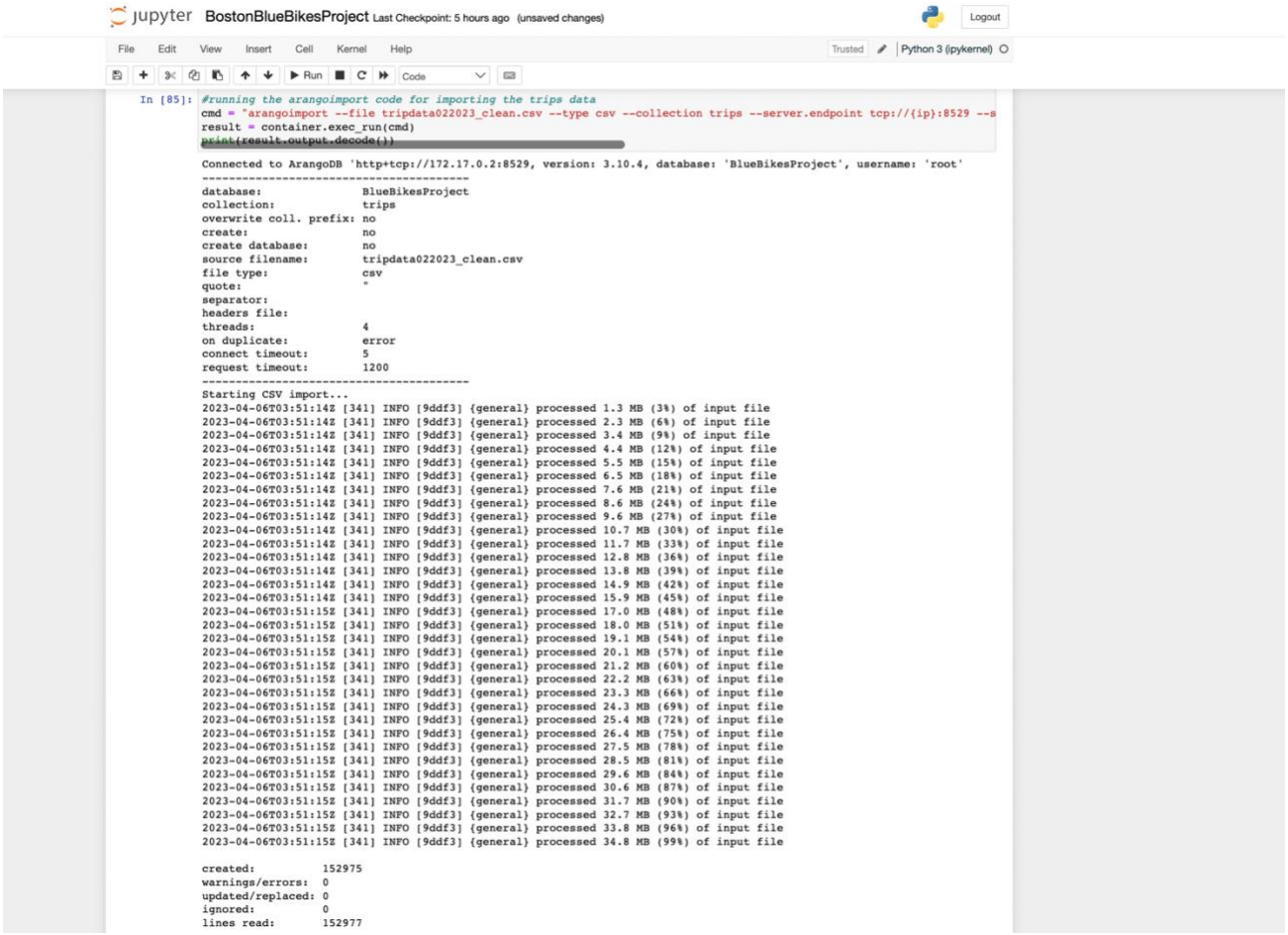
jupyter BostonBlueBikesProject Last Checkpoint: 5 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Help

```
In [78]: #removing null values from postal code column

         df['postal code'] = df['postal code'].fillna(0)

In [79]: #since 'trips' is an edge collection, renaming the start_station_id and end_station_id columns to _from and _to respect

         df = df.rename(columns={'start station id': '_from'})
         df = df.rename(columns={'end station id': '_to'})

In [82]: #appending 'stations/' in front of _from and _to station ids so that they refer to the ids of the station collection

         df['_from'] = df['_from'].astype(str)
         df['_to'] = df['_to'].astype(str)

         df['_from'] = df['_from'].apply(lambda x: 'stations/' + x)
         df['_to'] = df['_to'].apply(lambda x: 'stations/' + x)

In [83]: df

Out[83]:
```

	tripduration	starttime	stoptime	_from	start station name	start station latitude	start station longitude	_to	end station name	end station latitude	end station longitude	bikeid	usc
0	263	2023-02-01 00:00:50.0410	2023-02-01 00:05:13.7090	stations/330	30 Dane St	42.381001	-71.104025	stations/110	Harvard University Gund Hall at Quincy St / Ki...	42.376369	-71.114025	7334	Subs
1	447	2023-02-01 00:02:50.1390	2023-02-01 00:10:17.4090	stations/413	Kennedy-Longfellow School 158 Spring St	42.369553	-71.085790	stations/386	Sennott Park Broadway at Norfolk Street	42.368605	-71.099302	3257	Subs
2	302	2023-02-01 00:09:00.7270	2023-02-01 00:14:03.2560	stations/554	Forsyth St at Huntington Ave	42.339202	-71.090511	stations/27	Roxbury Crossing T Stop - Columbus Ave at	42.331184	-71.095171	7824	Subs

- After cleaning the data, we have imported it using the **arangoimport** tool.



```
In [85]: #running the arangoimport code for importing the trips data
         cmd = "arangoimport --file tripdata022023_clean.csv --type csv --collection trips --server.endpoint tcp://{ip}:8529 --s
         result = container.exec_run(cmd)
         print(result.output.decode())

Connected to ArangoDB 'http+tcp://172.17.0.2:8529, version: 3.10.4, database: 'BlueBikesProject', username: 'root'
----------------------------------------
database:               BlueBikesProject
collection:             trips
overwrite coll. prefix: no
create:                 no
create database:        no
source filename:        tripdata022023_clean.csv
file type:              csv
quote:                  "
separator:
headers file:
threads:                4
on duplicate:           error
connect timeout:        5
request timeout:        1200
----------------------------------------
Starting CSV import...
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 1.3 MB (3%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 2.3 MB (6%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 3.4 MB (9%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 4.4 MB (12%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 5.5 MB (15%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 6.5 MB (18%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 7.6 MB (21%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 8.6 MB (24%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 9.6 MB (27%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 10.7 MB (30%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 11.7 MB (33%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 12.8 MB (36%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 13.8 MB (39%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 14.9 MB (42%) of input file
2023-04-06T03:51:14Z [341] INFO [9ddf3] {general} processed 15.9 MB (45%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 17.0 MB (48%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 18.0 MB (51%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 19.1 MB (54%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 20.1 MB (57%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 21.2 MB (60%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 22.2 MB (63%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 23.3 MB (66%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 24.3 MB (69%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 25.4 MB (72%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 26.4 MB (75%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 27.5 MB (78%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 28.5 MB (81%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 29.6 MB (84%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 30.6 MB (87%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 31.7 MB (90%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 32.7 MB (93%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 33.8 MB (96%) of input file
2023-04-06T03:51:15Z [341] INFO [9ddf3] {general} processed 34.8 MB (99%) of input file

created:          152975
warnings/errors:  0
updated/replaced: 0
ignored:          0
lines read:       152977
```