**Step 1: Connect to the Snowflake SQL**

- Download SNOWSQL : https://developers.snowflake.com/snowsql/
- Open cmd
  - ➢ snowsql -a xsymtcm-em56590 -u amritaneogi
  - ➢ Password: **************

Here, account_name = xsymtcm-em56590

login_name = amritaneogi

**Step 2: Create a virtual Schema or Database**

➢ CREATE DATABASE HOUSE_PRICE;

Default schema available for each database

➢ amritaneogi#COMPUTE_WH@HOUSE_PRICE.PUBLIC >

Connected to the new database

**Step 3: Create a virtual Data Warehouse**; COMPUTE_WH is the default warehouse provided.

Having a data warehouse in Snowflake is crucial, unlike other databases. The warehouse serves as the essential resource allocator for various computations, including running queries and performing SQL operations. This necessity arises from Snowflake's nature as a cloud-based data warehouse, where every operation consumes resources. The resources are used from the warehouse select, and based on this the cost is determined.

➢ CREATE WAREHOUSE PRICE_WH;
➢ amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC >

**** To reconnect-> Use Warehouse PRICE_WH**

**Use Database HOUSE_PRICE;**

**Step 4: Create a 'Stage'.**

In other databases we load the file directly into the table. However, in Snowflake, we have to load it into a stage. A stage is nothing but a location which stores data files.

While creating stage we also need to mention the mention the format of the data files that it will store.

➢
```
CREATE OR REPLACE FILE FORMAT JSON_FORMAT
TYPE = 'JSON'
STRIP_OUTER_ARRAY = TRUE;
```

This is important to mention. This strips or eliminates the '[' and ']' at the beginning and end of the JSON file. If we skip this command and try to export a file, it will throw an error since it will treat the entire file as one single record.

➢
```
CREATE OR REPLACE FILE FORMAT CSV_FORMAT
  TYPE = csv
  FIELD_DELIMITER = ','
  FIELD_OPTIONALLY_ENCLOSED_BY='"';
```

Creating a new stage since we are uploading the data directly into stage. The data is in .csv file but the format is JSON.

-- creating stage

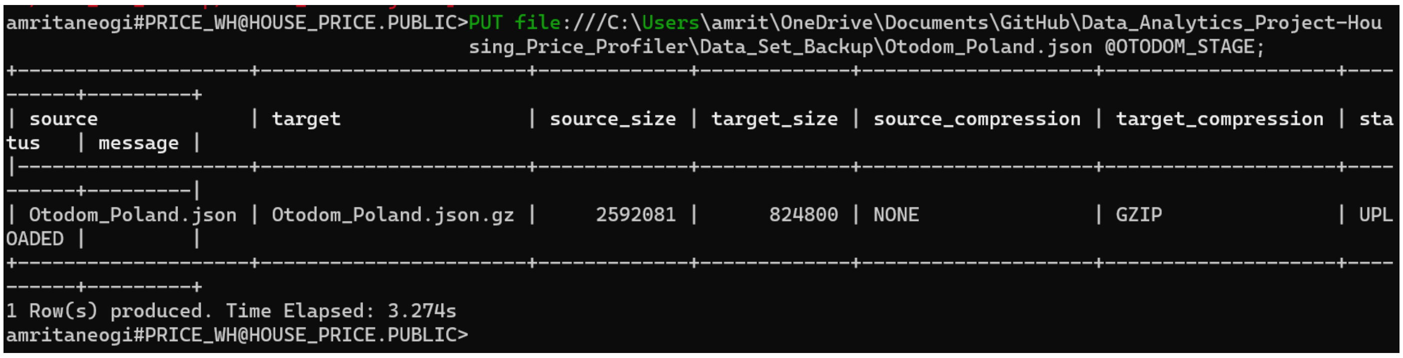
➢
```
CREATE OR REPLACE STAGE OTODOM_STAGE
FILE_FORMAT = CSV_FORMAT;
```

**Step 5: Import file in the Stage.**

We can directly export the file (either in .csv or JASON format) from the BrightData; but we need to purchase the file for that.

Instead, we will directly upload the file in the Snowflake 'Stage' that we created earlier.

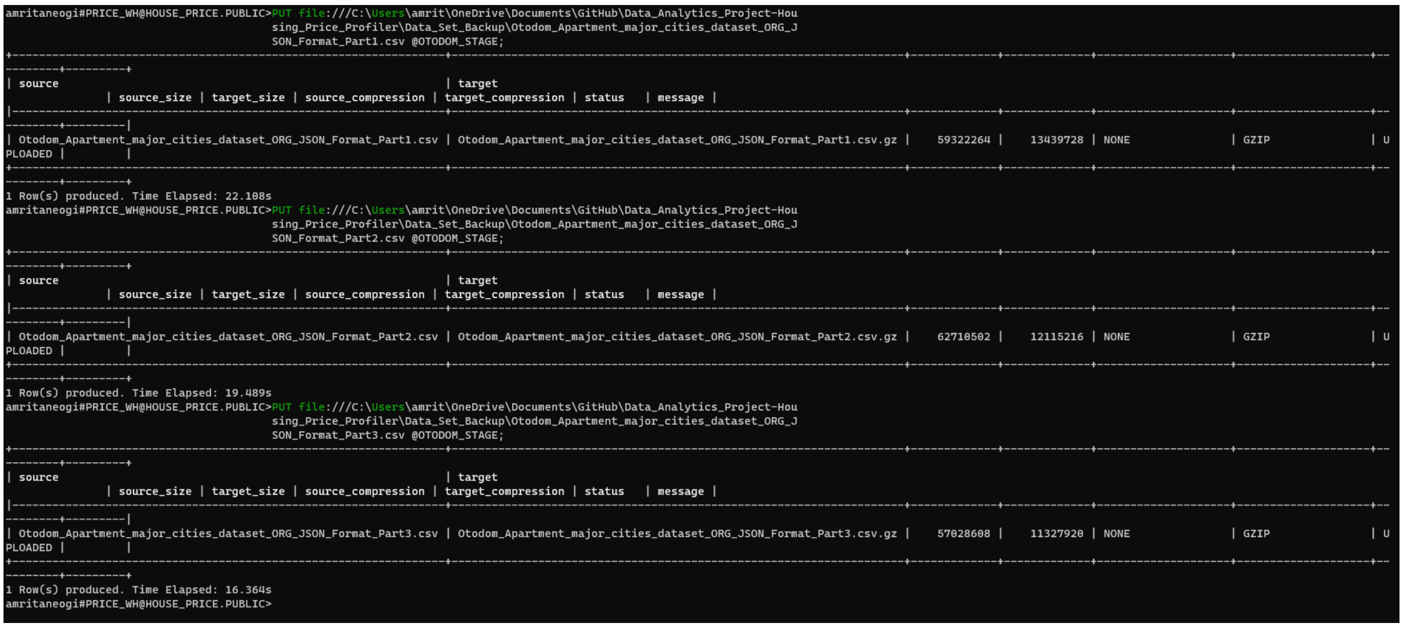
- PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-Hou
  sing_Price_Profiler\Data_Set_Backup\Otodom_Poland.json @OTODOM_STAGE;



```
amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC>PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-Hou
                                        sing_Price_Profiler\Data_Set_Backup\Otodom_Poland.json @OTODOM_STAGE;
+---------------------+-----------------------+-------------+-------------+--------------------+--------------------+----
------+----------+
| source              | target                | source_size | target_size | source_compression | target_compression | sta
tus     | message |
|---------------------+-----------------------+-------------+-------------+--------------------+--------------------+----
------+----------|
| Otodom_Poland.json  | Otodom_Poland.json.gz |     2592081 |      824800 | NONE               | GZIP               | UPL
OADED |          |
+---------------------+-----------------------+-------------+-------------+--------------------+--------------------+----
------+----------+
1 Row(s) produced. Time Elapsed: 3.274s
amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC>
```

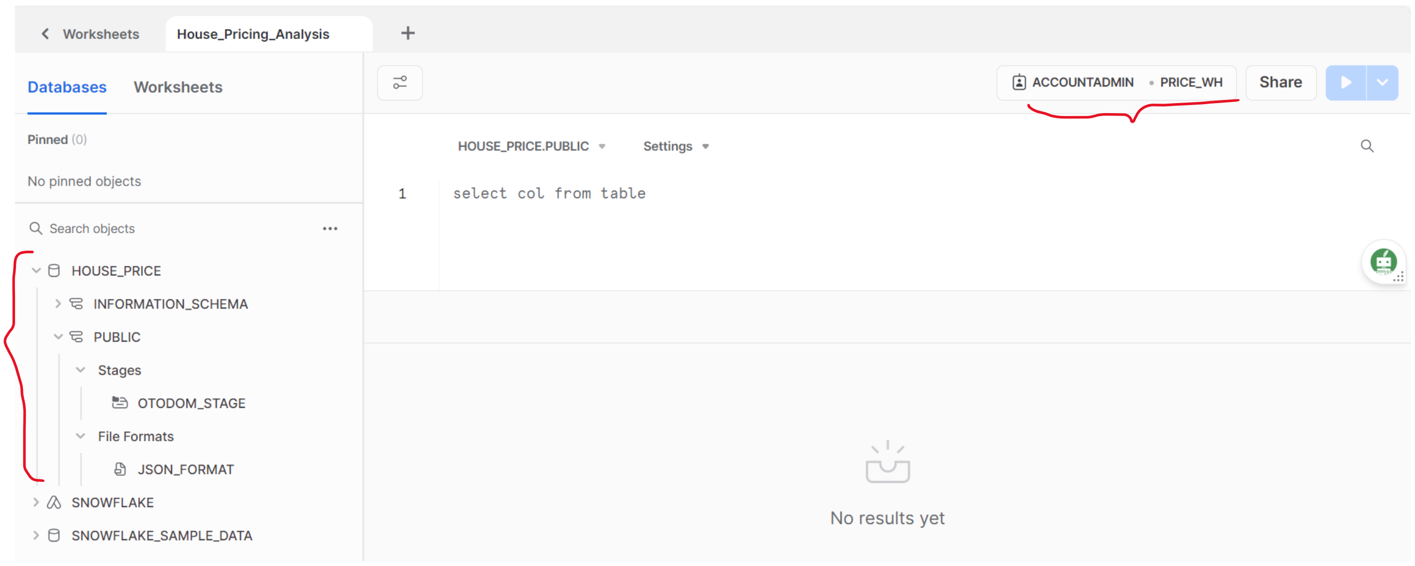
While using the CSV_FORMAT we will do it the following way:

- PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-Housing_Price_Profiler\Data_Set_Backup\Otodom_Apartment_major_cities_dataset_ORG_JSON_Format_Part1.csv @OTODOM_STAGE;
- PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-Housing_Price_Profiler\Data_Set_Backup\Otodom_Apartment_major_cities_dataset_ORG_JSON_Format_Part2.csv @OTODOM_STAGE;
- PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-Housing_Price_Profiler\Data_Set_Backup\Otodom_Apartment_major_cities_dataset_ORG_JSON_Format_Part3.csv @OTODOM_STAGE;

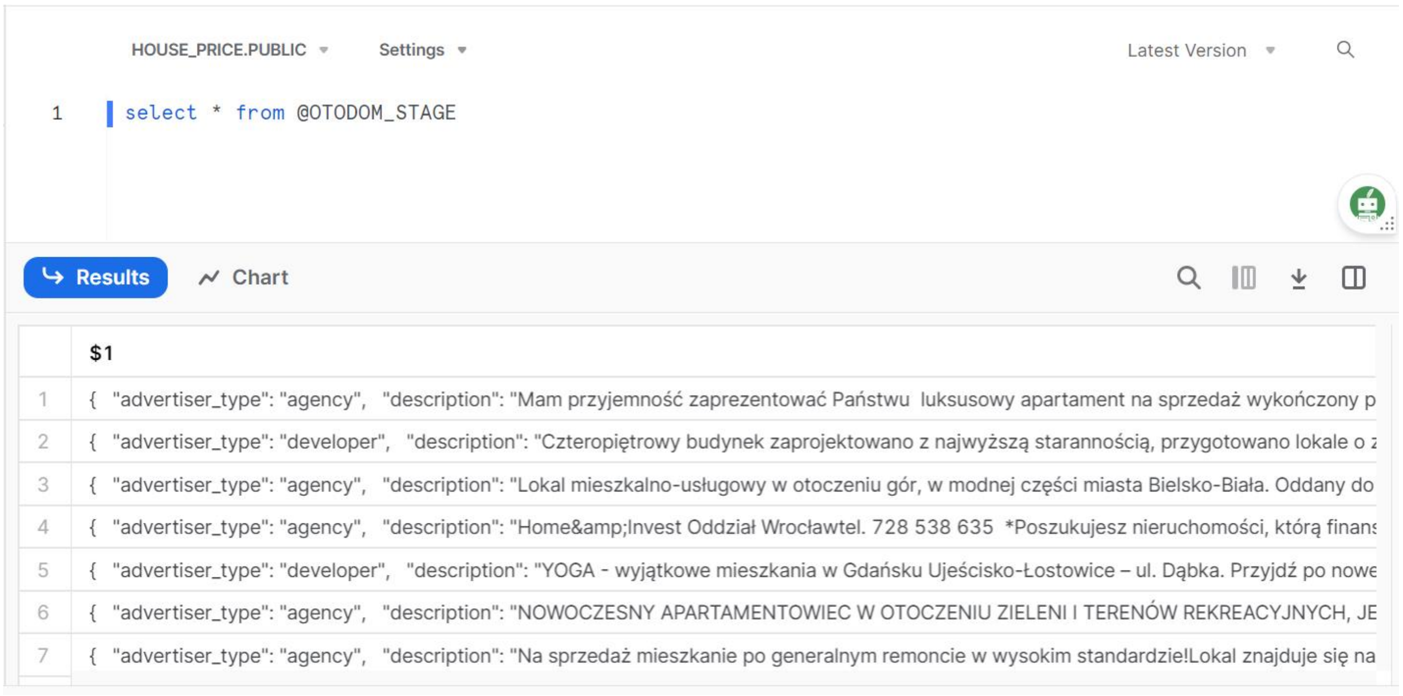


```
amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC>PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-Hou
                                        sing_Price_Profiler\Data_Set_Backup\Otodom_Apartment_major_cities_dataset_ORG_J
                                        SON_Format_Part1.csv @OTODOM_STAGE;
+----------------------------------------------------------+-------------------------------------------------------------+-------------+-------------+--------------------+--------------------+--
--------+---------+
| source                                                   | target
           | source_size | target_size | source_compression | target_compression | status   | message |
|----------------------------------------------------------+-------------------------------------------------------------+-------------+-------------+--------------------+--------------------+--
--------+---------|
| Otodom_Apartment_major_cities_dataset_ORG_JSON_Format_Part1.csv | Otodom_Apartment_major_cities_dataset_ORG_JSON_Format_Part1.csv.gz |    59322264 |    13439728 | NONE               | GZIP               | U
PLOADED |         |
+----------------------------------------------------------+-------------------------------------------------------------+-------------+-------------+--------------------+--------------------+--
--------+---------+
1 Row(s) produced. Time Elapsed: 22.108s
amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC>PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-Hou
                                        sing_Price_Profiler\Data_Set_Backup\Otodom_Apartment_major_cities_dataset_ORG_J
                                        SON_Format_Part2.csv @OTODOM_STAGE;
+----------------------------------------------------------+-------------------------------------------------------------+-------------+-------------+--------------------+--------------------+--
--------+---------+
| source                                                   | target
           | source_size | target_size | source_compression | target_compression | status   | message |
|----------------------------------------------------------+-------------------------------------------------------------+-------------+-------------+--------------------+--------------------+--
--------+---------|
| Otodom_Apartment_major_cities_dataset_ORG_JSON_Format_Part2.csv | Otodom_Apartment_major_cities_dataset_ORG_JSON_Format_Part2.csv.gz |    62710502 |    12115216 | NONE               | GZIP               | U
PLOADED |         |
+----------------------------------------------------------+-------------------------------------------------------------+-------------+-------------+--------------------+--------------------+--
--------+---------+
1 Row(s) produced. Time Elapsed: 19.489s
amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC>PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-Hou
                                        sing_Price_Profiler\Data_Set_Backup\Otodom_Apartment_major_cities_dataset_ORG_J
                                        SON_Format_Part3.csv @OTODOM_STAGE;
+----------------------------------------------------------+-------------------------------------------------------------+-------------+-------------+--------------------+--------------------+--
--------+---------+
| source                                                   | target
           | source_size | target_size | source_compression | target_compression | status   | message |
|----------------------------------------------------------+-------------------------------------------------------------+-------------+-------------+--------------------+--------------------+--
--------+---------|
| Otodom_Apartment_major_cities_dataset_ORG_JSON_Format_Part3.csv | Otodom_Apartment_major_cities_dataset_ORG_JSON_Format_Part3.csv.gz |    57028608 |    11327920 | NONE               | GZIP               | U
PLOADED |         |
+----------------------------------------------------------+-------------------------------------------------------------+-------------+-------------+--------------------+--------------------+--
--------+---------+
1 Row(s) produced. Time Elapsed: 16.364s
amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC>
```

This is now the Snowflake server look once all the changes are made:



```
select col from table
```



```
select * from @OTODOM_STAGE
```

|   | $1 |
|---|---|
| 1 | {  "advertiser_type": "agency",   "description": "Mam przyjemność zaprezentować Państwu  luksusowy apartament na sprzedaż wykończony p |
| 2 | {  "advertiser_type": "developer",   "description": "Czteropiętrowy budynek zaprojektowano z najwyższą starannością, przygotowano lokale o z |
| 3 | {  "advertiser_type": "agency",   "description": "Lokal mieszkalno-usługowy w otoczeniu gór, w modnej części miasta Bielsko-Biała. Oddany do |
| 4 | {  "advertiser_type": "agency",   "description": "Home&amp;Invest Oddział Wrocławtel. 728 538 635  *Poszukujesz nieruchomości, którą finans |
| 5 | {  "advertiser_type": "developer",   "description": "YOGA - wyjątkowe mieszkania w Gdańsku Ujeścisko-Łostowice – ul. Dąbka. Przyjdź po nowe |
| 6 | {  "advertiser_type": "agency",   "description": "NOWOCZESNY APARTAMENTOWIEC W OTOCZENIU ZIELENI I TERENÓW REKREACYJNYCH, JE |
| 7 | {  "advertiser_type": "agency",   "description": "Na sprzedaż mieszkanie po generalnym remoncie w wysokim standardzie!Lokal znajduje się na |

** for the data we are using, it’s too large to display

**Step 6: Once the data is loaded into the stage, create a table.**

We are creating a table with only one column here ‘json_data’.

*# writing codes on the snowflake console directly and not using cmd.*

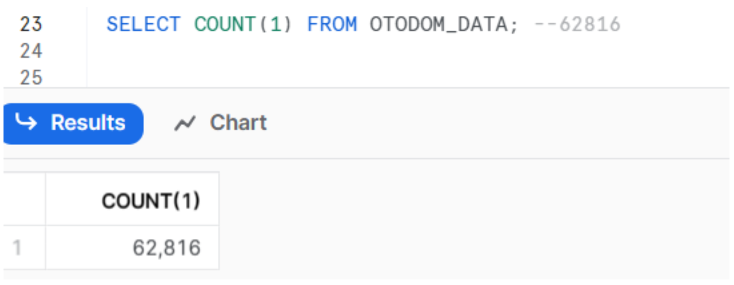
```
Create table OTODOM_DATA
(
json_data VARIANT  -- creating table with one column
);
```

‘VARIANT’ → this is a special kind of data type. It is used to store semi-structured data from JSON file.

**Step 7: Load data from stage to table.**

```
--copying data from stage to the table
copy into OTODOM_DATA
from @OTODOM_STAGE
on_error= 'skip_file'; -- skip any kind of error that could abandon the entire file.
```

***NOTE***: *If the data is exported directly from BrightData, then it will have 2 extra files of one record each. This are the files that BrightData sends for the purpose of testing the connection.*



```
SELECT COUNT(1) FROM OTODOM_DATA; --62816
```

|   | COUNT(1) |
|---|---|
| 1 | 62,816 |

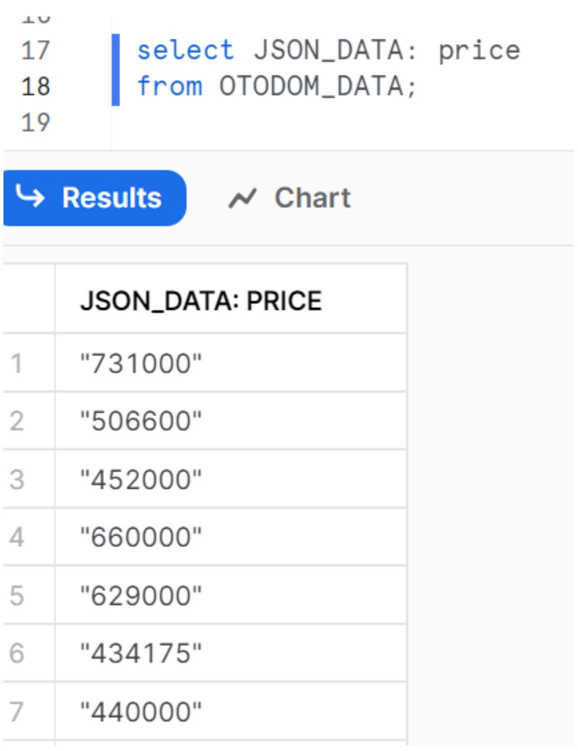
**Step 8: Flattening of file.**

The data we loaded is in JSON format, it is semi structured and not apt for any kind of analysis. So, we need to transform this into appropriate table columns. This process is called flattening of files.

Actual record (semi-structured):

{ "advertiser_type": "agency", "description": "Chcesz mieszkać w ścisłym centrum Głogowa? A może poszukujesz nieruchomości, pod kątem inwestycyjnym? Świetnie! Mam dla Ciebie idealne rozwiązanie. Mieszkanie zlokalizowane jest na Placu Jana z Głogowa, nieopodal skweru z fontanną oraz Parku Słowiańskiego i Parku Sapera, zatem w otoczeniu zieleni. W pobliżu odnajdziesz wszystko co jest niezbędne do funkcjonowania – sklepy spożywcze, piekarnię, cukiernię, punkty usługowe oraz gastronomiczne. Blok posiada dodatkową bramę wejściową zabezpieczoną domofonem. Mieszkanie charakteryzuje się 36,5 m2 powierzchni użytkowej. Nieruchomość posadowiona jest na parterze pięcio-kondygnacyjnego budynku i składa się z: ● Salonu z bardzo słonecznym, dużym oknem oraz balkonem, ● Aneksu kuchennego, który jest oddzielony ścianką lecz bez problemu można go otworzyć na salon uzyskując większą przestrzeń, ● Sypialni ● Łazienki, wyposażonej w wannę, umywalkę oraz WC, ● Przedpokoju. Mieszkanie jest utrzymane w bardzo dobrej kondycji, stolarka okienna pcv biała. Ściany zostały odświeżone, a nieruchomość nadaje się do zamieszkania. Jest to również nieruchomość gotowa pod inwestycję, bowiem całe wyposażenie pozostaje do dyspozycji nowego nabywcy. Ogrzewanie miejskie, mieszkanie z zasobów Spółdzielni Mieszkaniowej. Do nieruchomości przynależy również piwnica. Chcesz dowiedzieć się więcej? Zadzwoń i umów się na bezpłatną prezentację nieruchomości.Istnieje możliwość negocjacji ceny. Biuro Nieruchomości CUPRUM HOUSE Anna Wernik – Kaniewska Specjalista ds. nieruchomości tel. 723 411 712 \r\n\r\nOferta wysłana z programu dla biur nieruchomości ASARI CRM ()\r\n", " ": "pełna własność", "is_for_sale": true, "lighting": null, "location": "Głogów, głogowski, dolnośląskie", "market": [ "market", "secondary" ], "no_of_rooms": 2, "posting_id": "4mSyJ", "price": "210000", "remote_support": null, "surface": "36.5", "timestamp": "2023-09-20", "title": "Dwa pokoje w ścisłym centrum do zamieszkania", "url": "https://otodom.pl/pl/oferta/dwa-pokoje-w-scislym-centrum-do-zamieszkania-ID4mSyJ" }

For example:



```
select JSON_DATA: price
from OTODOM_DATA;
```

| | JSON_DATA: PRICE |
|---|---|
| 1 | "731000" |
| 2 | "506600" |
| 3 | "452000" |
| 4 | "660000" |
| 5 | "629000" |
| 6 | "434175" |
| 7 | "440000" |

SELECT JSON_DATA:price

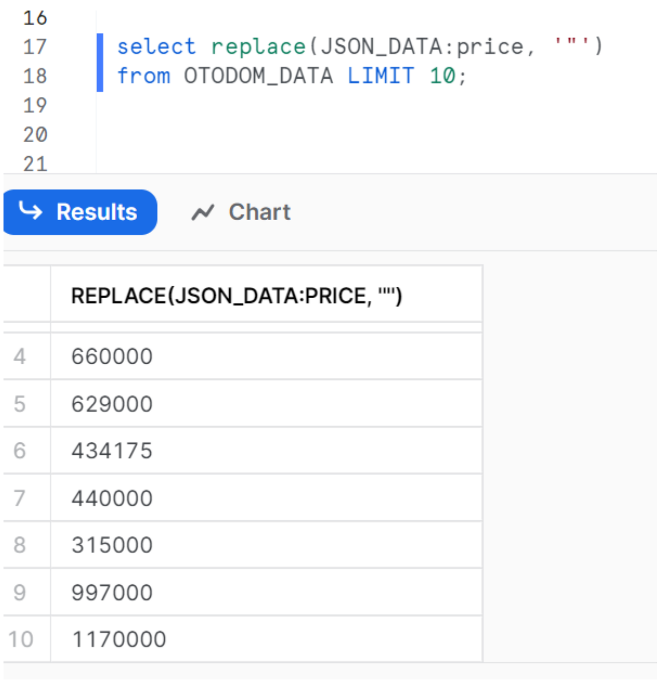
FROM OTODOM_DATAt limit 5;

This might throw error since the field is text but has JSON format data hence need to use PARSE_JSON function to parse the JSON data in this field as shown in above query:

SELECT PARSE_JSON(json_data):price

FROM OTODOM_DATA limit 5;

Remove the double quotes:



```
select replace(JSON_DATA:price, '"')
from OTODOM_DATA LIMIT 10;
```

| | REPLACE(JSON_DATA:PRICE, '"') |
|---|---|
| 4 | 660000 |
| 5 | 629000 |
| 6 | 434175 |
| 7 | 440000 |
| 8 | 315000 |
| 9 | 997000 |
| 10 | 1170000 |

We will do the same for all other columns:

```
CREATE OR REPLACE table OTODOM_DATA_FLATTEN
as
select row_number() over(order by title) as rn
, x.*
from (
select replace(parse_json(json_data):advertiser_type,'"')::string as advertiser_type
, replace(parse_json(json_data):balcony_garden_terrace,'"')::string as balcony_garden_terrace
, regexp_replace(replace(parse_json(json_data):description,'"'), '<[^>]+>')::string as description
, replace(parse_json(json_data):heating,'"')::string as heating
, replace(parse_json(json_data):is_for_sale,'"')::string as is_for_sale
, replace(parse_json(json_data):lighting,'"')::string as lighting
, replace(parse_json(json_data):location,'"')::string as location
, replace(parse_json(json_data):price,'"')::string as price
, replace(parse_json(json_data):remote_support,'"')::string as remote_support
, replace(parse_json(json_data):rent_sale,'"')::string as rent_sale
, replace(parse_json(json_data):surface,'"')::string as surface
, replace(parse_json(json_data):timestamp,'"')::date as timestamp
, replace(parse_json(json_data):title,'"')::string as title
, replace(parse_json(json_data):url,'"')::string as url
, replace(parse_json(json_data):form_of_property,'"')::string as form_of_property
, replace(parse_json(json_data):no_of_rooms,'"')::string as no_of_rooms
, replace(parse_json(json_data):parking_space,'"')::string as parking_space
from OTODOM_DATA
) x;
```

***Note***:

*The code uses regexp_replace to remove HTML tags from the "description" column. This is done by replacing anything that matches the regular expression '<[^>]+>' (which matches HTML tags) with an empty string.*

**Step 9: Transformation of data for further analysis**

1. Locations contains longitudinal and latitudinal values
2. Change the language from Polish to English

We will use Python and Google sheet for the same.

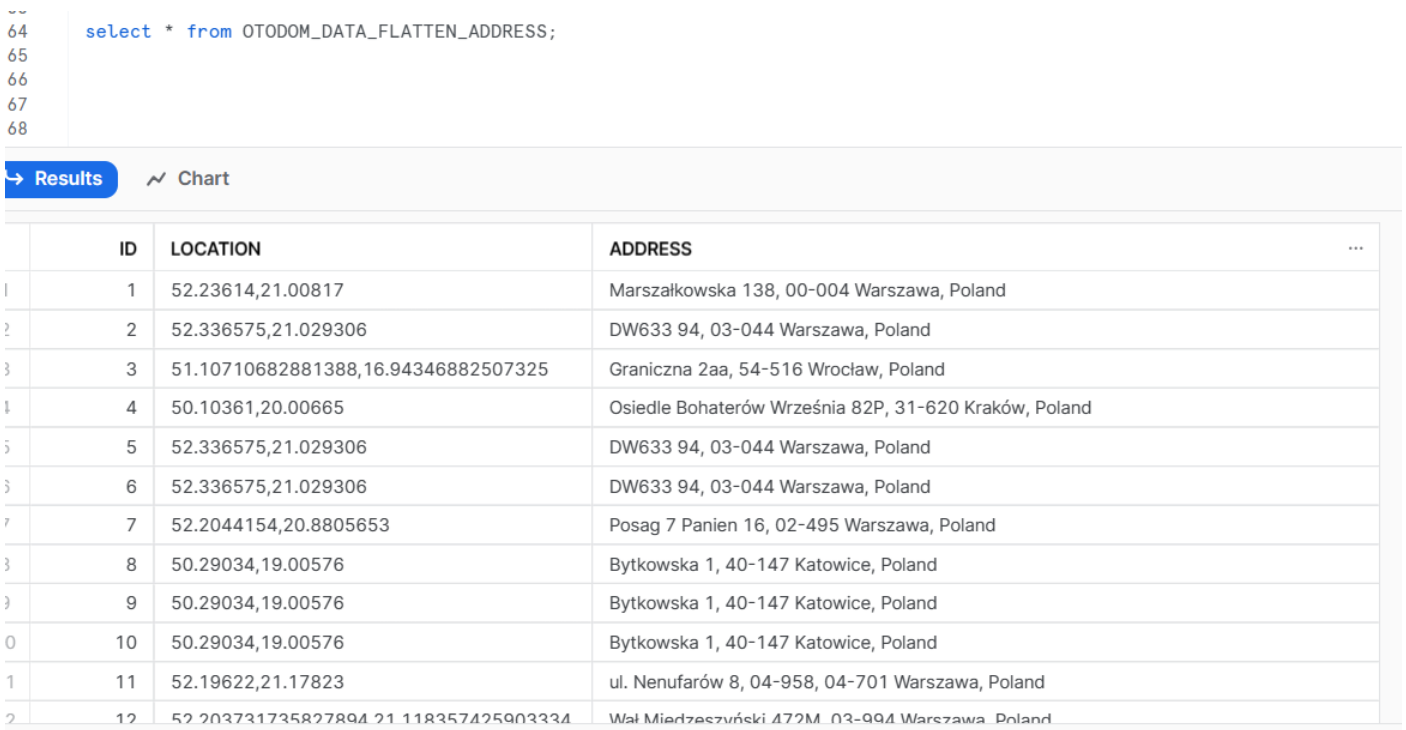
**Step 9.1: Create Python Virtual Environment**

Open Terminal from Jupyter Notebook

- ➢ conda create --name data_analytics_project python=3.11
  All the required packages are installed with it at the same time

  Changes done specific to my code ( ran code on Jupyter Notebook):

- ➢ conda active data_analytics_project
- ➢ conda install -n data_analytics_project jupyter
- ➢ jupyter notebook ⟶ Open a new Jupyter notebook
- ➢ conda install -n data_analytics_project -c conda-forge snowflake-sqlalchemy ⟶ From the newly created Jupyter notebook
- ➢ !pip install snowflake-sqlalchemy
- ➢ # import all the packages and libraries (refer the .ipynb file)
- ➢ Execute the Python code → Transform into proper Longitude and Latitude coordinates → Convert between addresses and geographic coordinates using GOOGLE GEOCODE API → Add new ADDRESS column to the original dataset with proper address → Create a new Table "OTODOM_DATA_FLATTEN_AADRESS" and insert proper address to their corresponding geographic coordinates.



```
select * from OTODOM_DATA_FLATTEN_ADDRESS;
```

Results    Chart

| ID | LOCATION | ADDRESS |
|---|---|---|
| 1 | 52.23614,21.00817 | Marszałkowska 138, 00-004 Warszawa, Poland |
| 2 | 52.336575,21.029306 | DW633 94, 03-044 Warszawa, Poland |
| 3 | 51.10710682881388,16.94346882507325 | Graniczna 2aa, 54-516 Wrocław, Poland |
| 4 | 50.10361,20.00665 | Osiedle Bohaterów Września 82P, 31-620 Kraków, Poland |
| 5 | 52.336575,21.029306 | DW633 94, 03-044 Warszawa, Poland |
| 6 | 52.336575,21.029306 | DW633 94, 03-044 Warszawa, Poland |
| 7 | 52.2044154,20.8805653 | Posag 7 Panien 16, 02-495 Warszawa, Poland |
| 8 | 50.29034,19.00576 | Bytkowska 1, 40-147 Katowice, Poland |
| 9 | 50.29034,19.00576 | Bytkowska 1, 40-147 Katowice, Poland |
| 10 | 50.29034,19.00576 | Bytkowska 1, 40-147 Katowice, Poland |
| 11 | 52.19622,21.17823 | ul. Nenufarów 8, 04-958, 04-701 Warszawa, Poland |
| 12 | 52.203731735827894,21.118357425903334 | Wał Miedzeszyński 472M, 03-994 Warszawa, Poland |

We have more than 300K records, so it was not possible to compute the code for that many rows. So ran codes in different chunks and uploaded the result in the form of csv to a new table OTODOM_DATA_FLATTEN_ADDRESS_FULL with the same columns as in OTODOM_DATA_FLATTEN_ADDRESS.

- Create table

```
CREATE TABLE OTODOM_DATA_FLATTEN_ADDRESS_FULL

(

ID int

LOCATION text

ADDRESS text  )

)
```

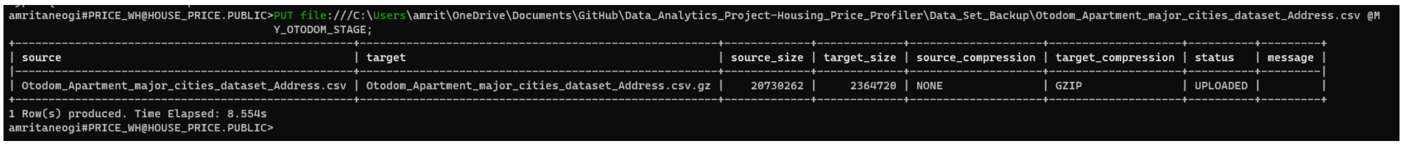
- Create the File Format and stage

```
CREATE OR REPLACE FILE FORMAT CSV_FORMAT
TYPE = csv
FIELD_DELIMITER = ','
FIELD_OPTIONALLY_ENCLOSED_BY='"';

CREATE OR REPLACE STAGE MY_OTODOM_STAGE
FILE_FORMAT = CSV_FORMAT;
```

- Upload data (.csv file) from path to stage

```
PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-
Housing_Price_Profiler\Data_Set_Backup\Otodom_Apartment_major_cities_dataset_Address.csv @M
Y_OTODOM_STAGE;
```



```
amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC>PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-Housing_Price_Profiler\Data_Set_Backup\Otodom_Apartment_major_cities_dataset_Address.csv @M
                                        Y_OTODOM_STAGE;
+------------------------------------------------+-----------------------------------------------------+-------------+-------------+--------------------+--------------------+----------+---------+
| source                                         | target                                              | source_size | target_size | source_compression | target_compression | status   | message |
|------------------------------------------------+-----------------------------------------------------+-------------+-------------+--------------------+--------------------+----------+---------|
| Otodom_Apartment_major_cities_dataset_Address.csv | Otodom_Apartment_major_cities_dataset_Address.csv.gz |    20730262 |     2364720 | NONE               | GZIP               | UPLOADED |         |
+------------------------------------------------+-----------------------------------------------------+-------------+-------------+--------------------+--------------------+----------+---------+
1 Row(s) produced. Time Elapsed: 8.554s
amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC>
```

- Copy data from stage to table

```
COPY INTO OTODOM_DATA_FLATTEN_ADDRESS_FULL
FROM @MY_OTODOM_STAGE;
```

**Step 9.2: Translate the address to English**

-- Translatet the Address title from Polish to English using Google Sheet

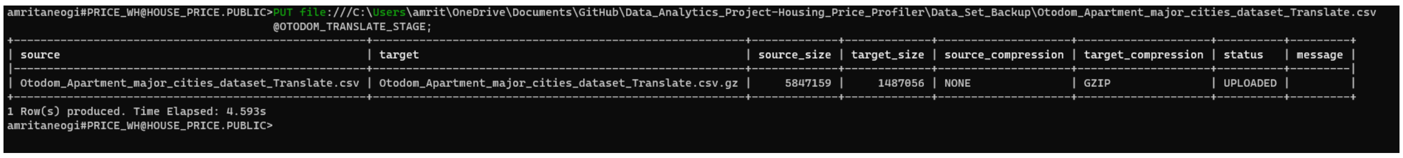
-- Refer Python scripts for the entire translation

```
select * from otodom_data_flatten_translate;

select count(1) from otodom_data_flatten_translate; -- sample 300 records
```

To translate the entire dataset we will directly upload .csv into the snowflake table

➔ Follow the same process of creating stage and copy data from stage to table



```
amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC>PUT file:///C:\Users\amrit\OneDrive\Documents\GitHub\Data_Analytics_Project-Housing_Price_Profiler\Data_Set_Backup\Otodom_Apartment_major_cities_dataset_Translate.csv
                                        @OTODOM_TRANSLATE_STAGE;
+--------------------------------------------------+------------------------------------------------------+-------------+-------------+--------------------+--------------------+----------+---------+
| source                                           | target                                               | source_size | target_size | source_compression | target_compression | status   | message |
|--------------------------------------------------+------------------------------------------------------+-------------+-------------+--------------------+--------------------+----------+---------|
| Otodom_Apartment_major_cities_dataset_Translate.csv | Otodom_Apartment_major_cities_dataset_Translate.csv.gz |     5847159 |     1487056 | NONE               | GZIP               | UPLOADED |         |
+--------------------------------------------------+------------------------------------------------------+-------------+-------------+--------------------+--------------------+----------+---------+
1 Row(s) produced. Time Elapsed: 4.593s
amritaneogi#PRICE_WH@HOUSE_PRICE.PUBLIC>
```

- **Summary**

  select * from OTODOM_DATA_FLATTEN; --All data

  select * from OTODOM_DATA_FLATTEN_ADDRESS_FULL; -- All geocode converted to address/locations

  select * from OTODOM_DATA_FLATTEN_TRANSLATE_FULL; --All translated Address data

## ANALYSIS AND REPORT BUILDING

**Step 10: Join all three tables**

Do basic transformation to:

- extract ‘suburbs’, ‘city’ and ‘country’ from the address
- remove ‘PLN’ and ‘€’ from the Price
- convert Euros (€) to Polish Złoty by multiplying with 4.43
- remove $m^2$ and $M^2$ from the Surface column
- OTODOM site has a number of for flats, garage , etc., however, we are only interested in flats. So, create a new flag for ‘apartment’ and ‘non apartment’

```
CREATE OR REPLACE TABLE OTODOM_DATA_TRANSFORMED

as

with cte as

  (select ot.*

  , case when price like 'PLN%' then try_to_number(replace(price,'PLN ',''),'999,999,999.99')

       when price like '€%' then try_to_number(replace(price,'€',''),'999,999,999.99') * 4.43

   end as price_new

  , try_to_double(replace(replace(replace(replace(surface,'m²',''),'м²',''),' ',''),',','.'),'9999.99') as surface_new

  , replace(parse_json(addr.address):suburb,'""', '') as suburb

  , replace(parse_json(addr.address):city,'""', '') as city

  , replace(parse_json(addr.address):country,'""', '') as country

  , trans.title_eng as title_eng

  from otodom_data_flatten ot

  left join otodom_data_flatten_address_full addr on ot.ID=addr.ID

  left join otodom_data_flatten_translate_full trans on ot.ID=trans.ID)

select *

, case when lower(title_eng) like '%commercial%' or lower(title_eng) like '%office%' or lower(title_eng) like
'%shop%' then 'non apartment'

     when is_for_sale = 'false' and surface_new <=330 and price_new <=55000 then 'apartment'

     when is_for_sale = 'false' then 'non apartment'

     when is_for_sale = 'true'  and surface_new <=600 and price_new <=20000000 then 'apartment'
```

```
       when is_for_sale = 'true'  then 'non apartment'
  end as apartment_flag
from cte;
```

**Refer the ‘Analysis.txt’ file for the problems and solutions.**