Alberta A Place to stay: ETL Process

Algonquin College
Course Number: CST_0022_10
Database Systems Admin and Mgmt

Professor: Andreas Gausrab
Date: 13th August 2023

Submitted By:
Dhruv Gadhiya (Team Lead)
Het Patel
Vidhi Tripathi
Saiyam Shah

Content

1. Abstract:

To perform effective analysis on the address data it should be well-structured, and it became easy to perform meaningful analysis using normalized street and city name, leading to more accurate insights for the future use. Normalized street names make precise geospatial analysis like plotting location on maps and calculate the distances and analyze the patterns based on the geographic location in the city. To fulfill these requirements here performed ETL process make huge difference for future analysis.

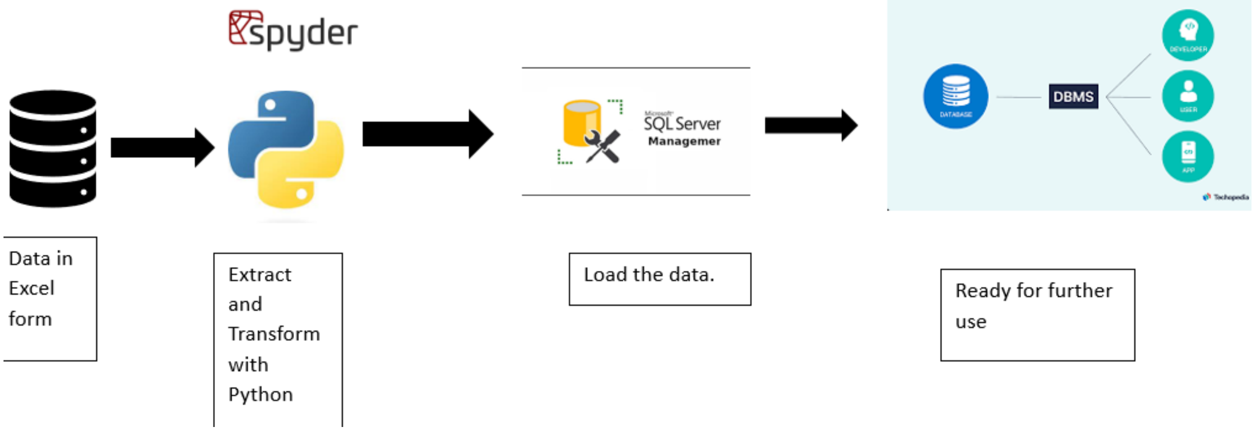
2. Introduction:

Alberta is the place to stay:

Alberta is the home to six UNESCO designated world Heritage site: the Canadian Rocky Mountain parks, dinosaur Provincial Park, and many more places to explore. [1] It is fourth largest province by area at 661,848 square kilometers, and the fourth most populous place, being home to 4,262,635 people. Having capital as a Edmonton, while Calgary is it's the largest city. The two are Albert's largest census metropolitan areas. [2] Edmonton, the provincial capital of Alberta, is situated roughly in the geographic middle of the province. It is the most northerly metropolitan city in Canada and acts as a crossing point and the hub for the development of resources in the north. The area possesses the majority of West Canada's crude oil processing capacity due to its proximity to the country's main oil reserves. Calgary is surrounded by an extensive grazing region and lies about 280 km (170 mi) south of Edmonton and 240 km (150 mi) north of Montana. The Calgary-Edmonton Corridor is house to about 75% of the province's residents. In the early years of the province, population growth had been encouraged by the land grant policies to the railways. [1] Thus, oil industry and fertile land for the farming of the Albert's state make it distinct stat of the Canada and attractive place to leave within Canada after Ontario. There are several places where humankind is living in huge amount having different areas, street and various types of housing system. Here in this data analysis, we would like to explore home address data of Alberta's residents.

3. Tools and techniques:

In this project, first we analyze the data from the excel sheet and explore the data with the help of python to find the dimension, statistical analysis, and the type of the data. From Python script we extract the normalized data from the original data source. Finally, SQL server management studio to load the data in the database with the help of Spyder script. Then load the data which is ready to use for the visualization purpose and further analysis.



Data in Excel form

Extract and Transform with Python

Load the data.

Ready for further use

4. About Data:

This data is extracted from the statistics Canada open database of address (ODA) is a collection of open address point data and available by government of the Canada under the Open Government License-Canada. The inputs for the ODA are datasets provided by municipal, regional, or provincial sources available to the general public through open government portals under various types of open data licenses. The current version of the database (version 1.0) contains approximately 10 million records and includes provinces and territories where open address data were found during the collection period (January to April 2021). Individual datasets sourced from their respective open data portals were processed and harmonized into the ODA. The data sources used do not deploy a uniform standardization system. The ODA harmonizes addresses by converting them to a uniform standard. The bellow's variables included in the ODA are as:

Source Index
ODA Index
Group Index
Civic Number
Street with street type and direction
Street name
Street type
Street direction
Unit
Full address
Postal code
City
Processed City
Standardized street name, type, and direction
Provide
Census subdivision Name and Unique Identifier
Province or Territory Unique Identifier
Latitude and Longitude

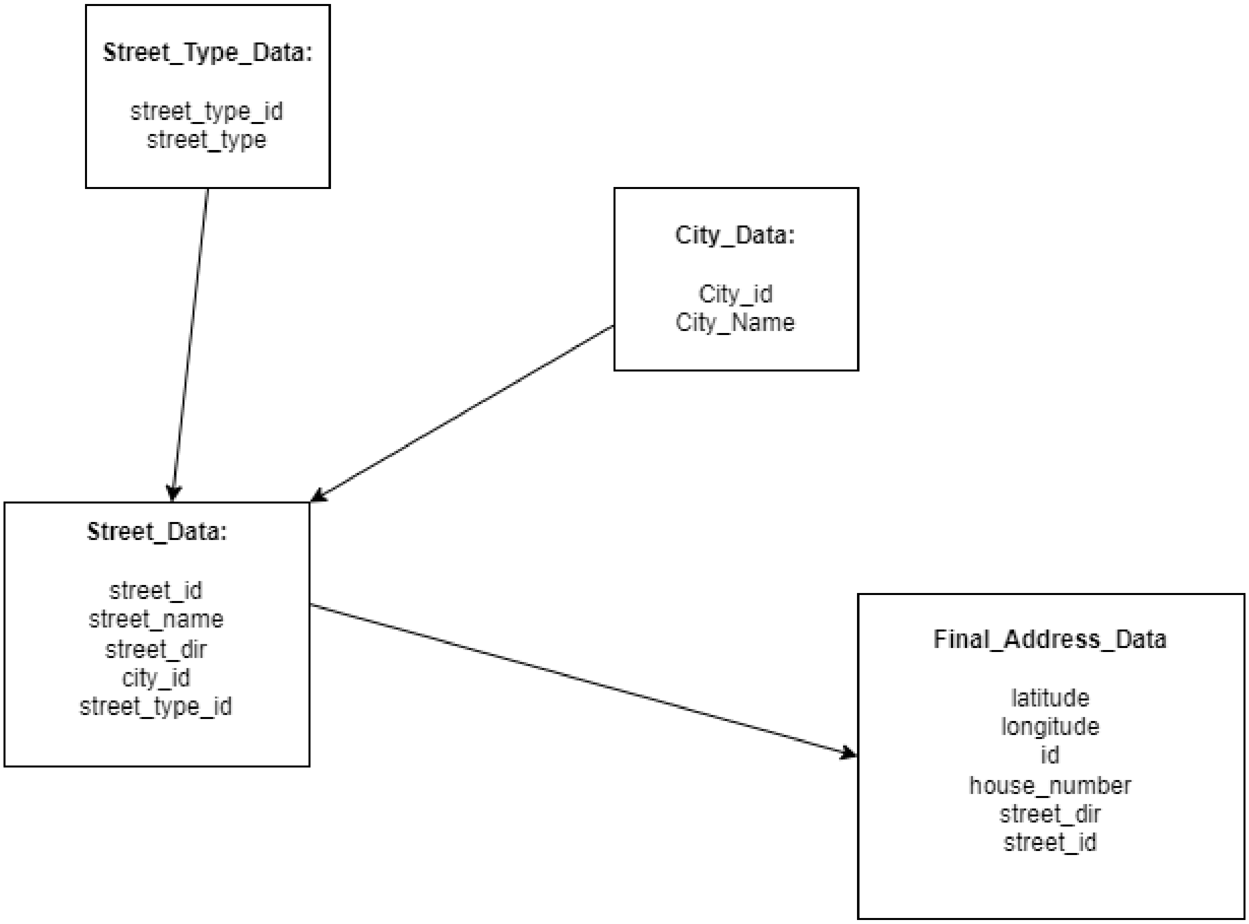
From this data source here, focus on only Alberta province where the data file is in the form of excel and it contain 1777428 rows and 22 columns as below. Thus, data is in 2 dimensions and having size of 39103416.Also, the dataset having all object datatype, where, latitude, longitude and postalcode in float whereas group_id, csdname and street id are integer format.

The main objective is to find the street id with normalized column with normalized city name according to that street.

5. Performed Task:

Extract the Data:

Here, first we created the relational schema from the original data source to create the needed tables for the further tasks.

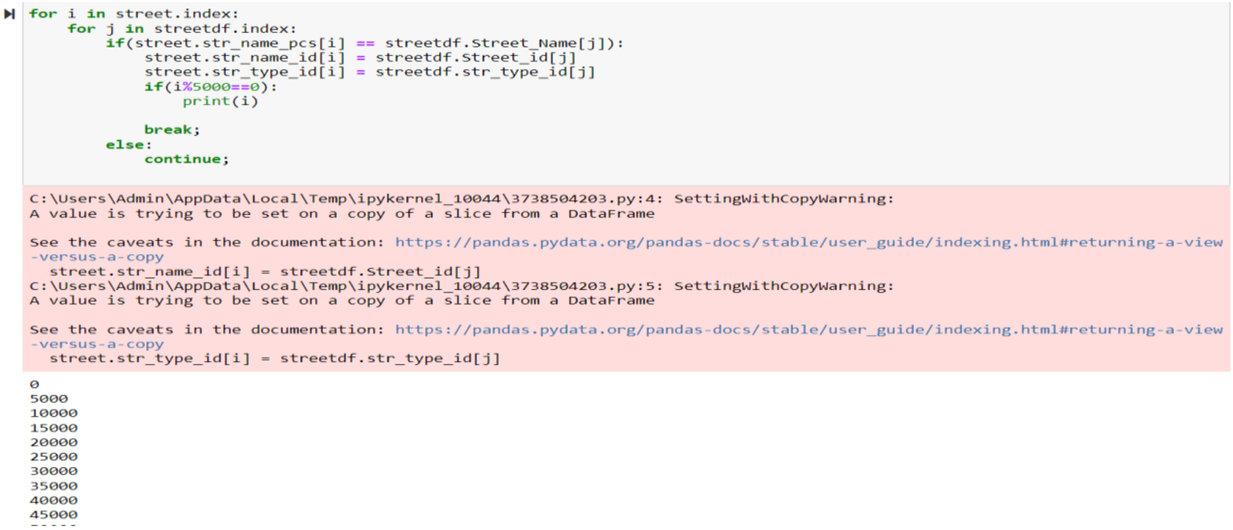


As per the below script tables are generated in the SQL server management studio.
We initiated the process by focusing on the unique city entries with the Unique city entries with the dataset. With the help of Unique () function, we extract the distinct city values in the different excel sheet. This function allowed to isolate each city for the further normalizing process.

Normalization of the street names:

To enhance the consistency and accuracy of the street name, we employed the normalization process, and it involves transforming the varying street names ensured that data quality and the accuracy of the analysis were upheld.

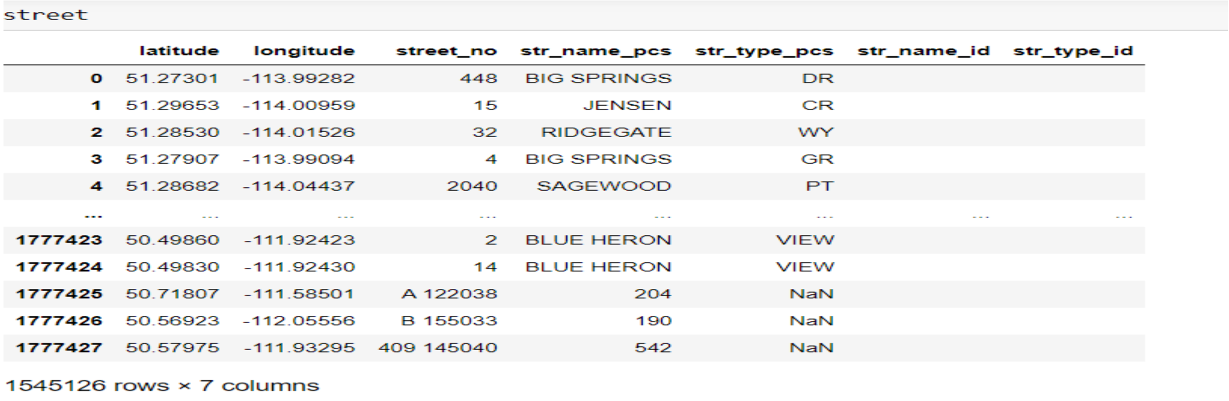
Transform the Data (Data Preparation):



```
for i in street.index:
    for j in streetdf.index:
        if(street.str_name_pcs[i] == streetdf.Street_Name[j]):
            street.str_name_id[i] = streetdf.Street_id[j]
            street.str_type_id[i] = streetdf.str_type_id[j]
            if(i%5000==0):
                print(i)

            break;
        else:
            continue;
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_10044\3738504203.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy
  street.str_name_id[i] = streetdf.Street_id[j]
C:\Users\Admin\AppData\Local\Temp\ipykernel_10044\3738504203.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy
  street.str_type_id[i] = streetdf.str_type_id[j]
```

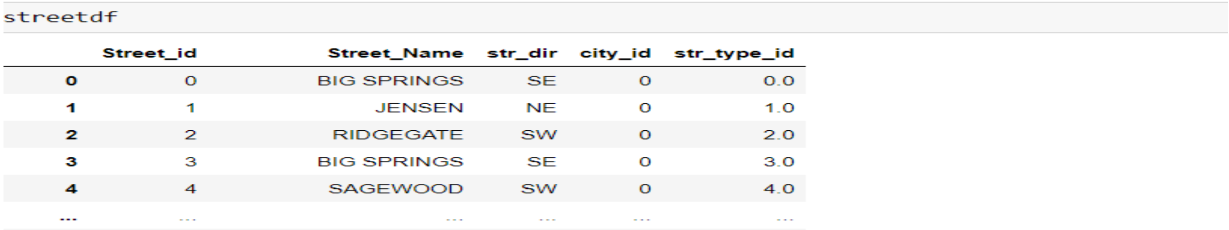
```
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
```



```
street
```

	latitude	longitude	street_no	str_name_pcs	str_type_pcs	str_name_id	str_type_id
0	51.27301	-113.99282	448	BIG SPRINGS	DR		
1	51.29653	-114.00959	15	JENSEN	CR		
2	51.28530	-114.01526	32	RIDGEGATE	WY		
3	51.27907	-113.99094	4	BIG SPRINGS	GR		
4	51.28682	-114.04437	2040	SAGEWOOD	PT		
...	...	...	...	...	...	...	...
1777423	50.49860	-111.92423	2	BLUE HERON	VIEW		
1777424	50.49830	-111.92430	14	BLUE HERON	VIEW		
1777425	50.71807	-111.58501	A 122038	204	NaN		
1777426	50.56923	-112.05556	B 155033	190	NaN		
1777427	50.57975	-111.93295	409 145040	542	NaN		

1545126 rows × 7 columns



```
streetdf
```

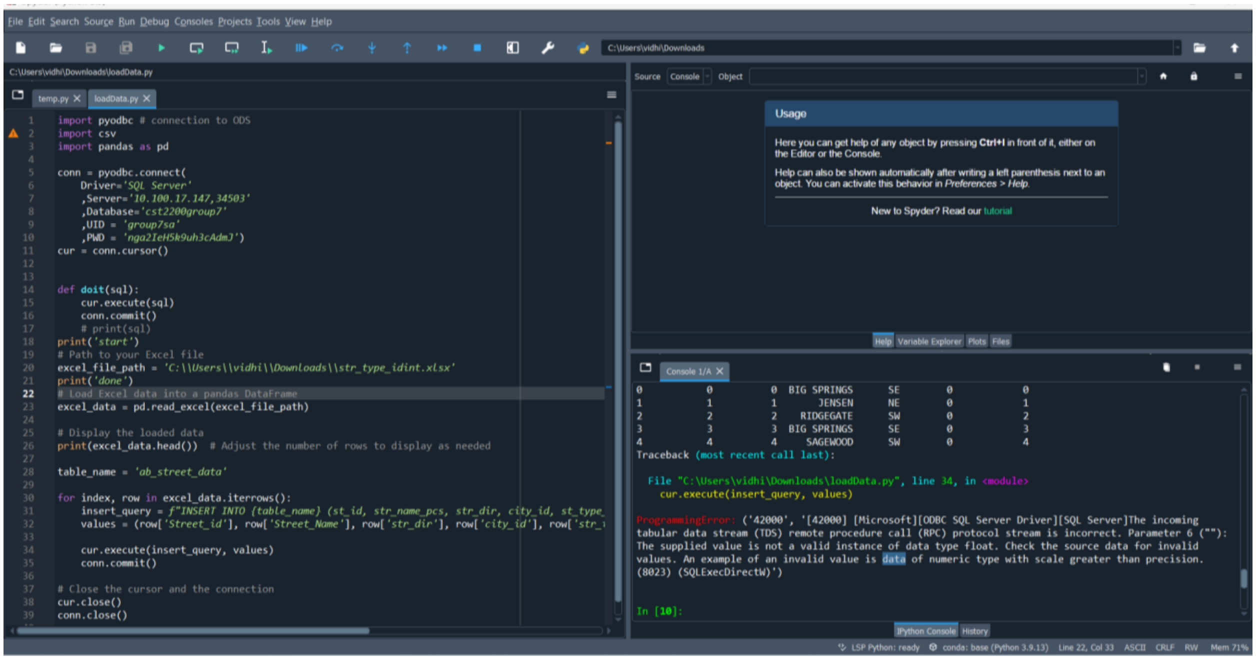
	Street_id	Street_Name	str_dir	city_id	str_type_id
0	0	BIG SPRINGS	SE	0	0.0
1	1	JENSEN	NE	0	1.0
2	2	RIDGEGATE	SW	0	2.0
3	3	BIG SPRINGS	SE	0	3.0
4	4	SAGEWOOD	SW	0	4.0
...	...	...	...	...	...

Once the city and street name data were successfully extracted but we need to transform the type of the data into one format. To create columns for street and city with one id need to be in one datatype but both have different dimension and the different datatype. For the purpose of normalization, we need unique city id and street type id which was achieved by using python. Thus, we extract the whole data.

After successfully extracting and normalizing the city and street name data, we combined the data into a comprehensive dataset. Future analysis, reporting, and decision-making will be built upon the data in this collection.

Our method of normalizing street names along with the use of the city_pcs. unique() function to extract different city values ensures that our dataset is ready for in-depth examination. This initial phase creates the foundation for insightful observations and well-informed choices based on precise and structured facts.

Load the data:

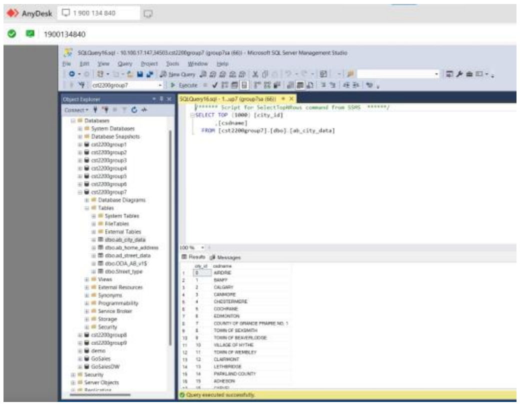


File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\vidhi\Downloads
C:\Users\vidhi\Downloads\loadData.py
temp.py
loadData.py
Source
Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Help
Variable Explorer
Plots
Files
Console 1/A
IPython Console
History
LSP Python: ready
conda: base (Python 3.9.13)
Line 22, Col 33
ASCII
CRLF
RW
Mem 71%

```python
import pyodbc # connection to ODS
import csv
import pandas as pd

conn = pyodbc.connect(
    Driver='SQL Server'
    ,Server='10.100.17.147,34503'
    ,Database='cst2200group7'
    ,UID = 'group7sa'
    ,PWD = 'nga2IeH5k9uh3cAdmJ')
cur = conn.cursor()


def doit(sql):
    cur.execute(sql)
    conn.commit()
    # print(sql)
print('start')
# Path to your Excel file
excel_file_path = 'C:\\Users\\vidhi\\Downloads\\str_type_idint.xlsx'
print('done')
# Load Excel data into a pandas DataFrame
excel_data = pd.read_excel(excel_file_path)

# Display the loaded data
print(excel_data.head())  # Adjust the number of rows to display as needed

table_name = 'ab_street_data'

for index, row in excel_data.iterrows():
    insert_query = f"INSERT INTO {table_name} (st_id, str_name_pcs, str_dir, city_id, st_type_
    values = (row['Street_id'], row['Street_Name'], row['str_dir'], row['city_id'], row['str_

    cur.execute(insert_query, values)
    conn.commit()

# Close the cursor and the connection
cur.close()
conn.close()
```

```
0          0          0  BIG SPRINGS    SE       0           0
1          1          1       JENSEN    NE       0           1
2          2          2    RIDGEGATE    SW       0           2
3          3          3  BIG SPRINGS    SE       0           3
4          4          4     SAGEWOOD    SW       0           4
Traceback (most recent call last):

  File "C:\Users\vidhi\Downloads\loadData.py", line 34, in <module>
    cur.execute(insert_query, values)

ProgrammingError: ('42000', '[42000] [Microsoft][ODBC SQL Server Driver][SQL Server]The incoming
tabular data stream (TDS) remote procedure call (RPC) protocol stream is incorrect. Parameter 6 (""):
The supplied value is not a valid instance of data type float. Check the source data for invalid
values. An example of an invalid value is data of numeric type with scale greater than precision.
(8023) (SQLExecDirectW)')


In [10]:
```

The main SQL file is now ready to do further -process, to clean the data with the help of python. As per the table requirements must need to the normalized data and that would be achieved through python. Now the data is ready to load in the database. Here, Spyder tool is used to load the data into the database.

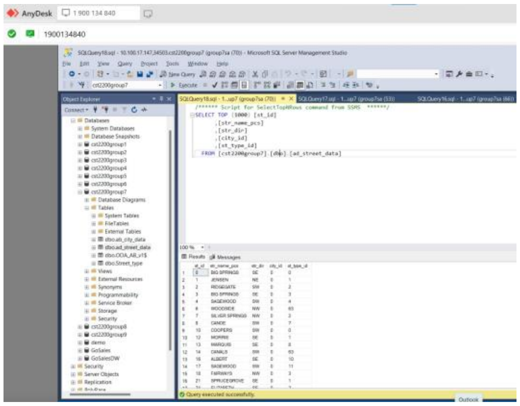


AnyDesk
1 900 134 840
1900134840
SQLQuery16.sql - 10.100.17.147,34503.cst2200group7 (group7sa (66)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
cst2200group7
Object Explorer
Connect
SQLQuery16.sql - 1...up7 (group7sa (66))
100 %
Results
Messages

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [city_id]
      ,[csdname]
  FROM [cst2200group7].[dbo].[ab_city_data]
```

	city_id	csdname
1	0	AIRDRIE
2	1	BANFF
3	2	CALGARY
4	3	CANMORE
5	4	CHESTERMERE
6	5	COCHRANE
7	6	EDMONTON
8	7	COUNTY OF GRANDE PRAIRIE NO. 1
9	8	TOWN OF SEXSMITH
10	9	TOWN OF BEAVERLODGE
11	10	VILLAGE OF HYTHE
12	11	TOWN OF WEMBLEY
13	12	CLAIRMONT
14	13	LETHBRIDGE
15	14	PARKLAND COUNTY
16	15	ACHESON

Query executed successfully.

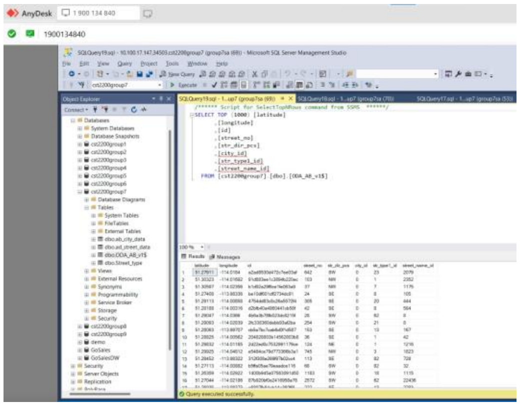


AnyDesk
1 900 134 840
1900134840
SQLQuery18.sql - 10.100.17.147,34503.cst2200group7 (group7sa (70)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
cst2200group7
Object Explorer
Connect
SQLQuery18.sql - 1...up7 (group7sa (70))
SQLQuery17.sql - 1...up7 (group7sa (53))
SQLQuery16.sql - 1...up7 (group7sa (66))
100 %
Results
Messages
Outlook

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [st_id]
      ,[str_name_pcs]
      ,[str_dir]
      ,[city_id]
      ,[st_type_id]
  FROM [cst2200group7].[dbo].[ad_street_data]
```

	st_id	str_name_pcs	str_dir	city_id	st_type_id
1	0	BIG SPRINGS	SE	0	0
2	1	JENSEN	NE	0	1
3	2	RIDGEGATE	SW	0	2
4	3	BIG SPRINGS	SE	0	3
5	4	SAGEWOOD	SW	0	4
6	6	WOODSIDE	NW	0	63
7	7	SILVER SPRINGS	NW	0	2
8	8	CANOE	SW	0	7
9	10	COOPERS	SW	0	0
10	12	MORRIS	SE	0	1
11	13	MARQUIS	SE	0	8
12	14	CANALS	SW	0	63
13	16	ALBERT	SE	0	10
14	17	SAGEWOOD	SW	0	11
15	18	FAIRWAYS	NW	0	3
16	21	SPRUCEGROVE	SE	0	1

Query executed successfully.

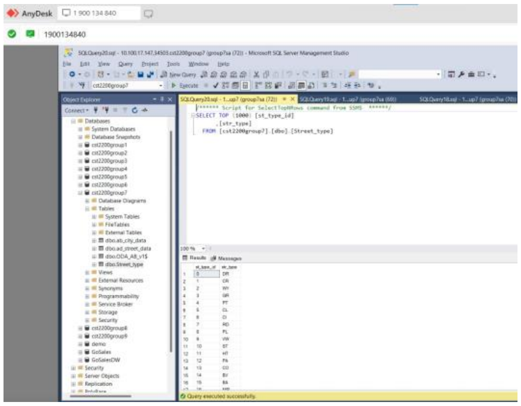


AnyDesk
1 900 134 840
1900134840
SQLQuery19.sql - 10.100.17.147,34503.cst2200group7 (group7sa (69)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
cst2200group7
Object Explorer
Connect
SQLQuery19.sql - 1...up7 (group7sa (69))
SQLQuery18.sql - 1...up7 (group7sa (70))
SQLQuery17.sql - 1...up7 (group7sa (53))
100 %
Results
Messages

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [latitude]
      ,[longitude]
      ,[id]
      ,[street_no]
      ,[str_dir_pcs]
      ,[city_id]
      ,[str_type1_id]
      ,[street_name_id]
  FROM [cst2200group7].[dbo].[ODA_AB_v1$]
```

	latitude	longitude	id	street_no	str_dir_pcs	city_id	str_type1_id	street_name_id
1	51.27911	-114.0184	[illegible]	642	SW	0	23	2079
2	51.30323	-114.01682	[illegible]	103	NW	0	1	2352
3	51.30587	-114.02398	[illegible]	37	NW	0	7	1176
4	51.27408	-113.98335	[illegible]	24	SE	0	8	105
5	51.29113	-114.00088	[illegible]	305	SE	0	20	444
6	51.28188	-114.00316	[illegible]	42	SE	0	8	564
7	51.29047	-114.0388	[illegible]	28	SW	0	82	8
8	51.29063	-114.02039	[illegible]	254	SW	0	21	8
9	51.28083	-113.99707	[illegible]	153	SE	0	13	167
10	51.28025	-114.00562	[illegible]	36	SE	0	1	42
11	51.28032	-114.01185	[illegible]	124	NE	0	1	1216
12	51.29025	-114.04512	[illegible]	745	NW	0	3	1823
13	51.28452	-113.98322	[illegible]	113	SE	0	82	728
14	51.27113	-114.00882	[illegible]	68	SW	0	82	32
15	51.26359	-114.02922	[illegible]	1183	SW	0	18	1115
16	51.27044	-114.02188	[illegible]	2572	SW	0	62	22436

Query executed successfully.



AnyDesk
1 900 134 840
1900134840
SQLQuery20.sql - 10.100.17.147,34503.cst2200group7 (group7sa (72)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
cst2200group7
Object Explorer
Connect
SQLQuery20.sql - 1...up7 (group7sa (72))
SQLQuery19.sql - 1...up7 (group7sa (69))
SQLQuery18.sql - 1...up7 (group7sa (70))
100 %
Results
Messages

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [st_type_id]
      ,[str_type]
  FROM [cst2200group7].[dbo].[Street_type]
```

	st_type_id	str_type
1	0	DR
2	1	CR
3	2	WY
4	3	GR
5	4	PT
6	5	CL
7	6	CI
8	7	RD
9	8	PL
10	9	VW
11	10	ST
12	11	HT
13	12	PA
14	13	CO
15	14	BY
16	15	BA

Query executed successfully.

6. Future Work and applications:

After loading the normalized data, one can visualize the data with the help of power BI or Tableau for more visual insight. It can be helpful in city planning, decision making and understanding societal trends. Demographic analysis helps to understand the patterns fro density of the population and the characteristics of different regions. Urban Planning for the government which can identify the high population area and the housing needs for the Alberta region. It can be also useful in the real estate business to understand the housing demand, property values and rental market. Moreover, businesses can use census data to analyze the consumer behavior, tailor market strategies and choose optimal locations for new business based on the population demographics and income level. Thua, overall, after fetching the normalized data it can be useful for the analysis purpose with data visualization to get more insights for the future use.

7. Lesson Learned:

After performing these ETL projects we get that if the data is too large as we have in this project then we need more time to load the data. As, here we have 1.5 million rows it took almost more than 28 hours to extract the data from python. So, time management is the key in this kind of project as it is a very time-consuming process. Apart from this, as a new learner, this is not a solo task, group work is very important in such a kind of project where everyone performed their role to complete the project on time and scope as we achieved in this project as a group.

8. Conclusion:

In a nutshell, after performing ETL process on conscious data raw data is extracted from the various source and transformed into structured and usable format and finally load by python into the SQL server management studio one can utilized data for the future analysis.

The ETL process's importance is highlighted by its ability to increase data quality, simplify data workflows, and offer enterprises valuable information.

9. References:

1. https://en.wikipedia.org/wiki/Alberta#:~:text=Alberta%20is%20home%20to%20six,%2Don%2DStone%20Provincial%20Park.
2. *"Population and dwelling counts, for Canada, provinces and territories, census metropolitan areas and census agglomerations, 2016 and 2011 censuses – 100% data (Alberta)". Statistics Canada. February 7, 2018. Archived from the original on February 23, 2020. Retrieved December 29, 2020.*