In [1]:
```
import pandas as pd
```

how to read json file in pandas

In [5]:
```
# this is the method how we can read the json file

df = pd.read_json('train.json')
```

In [6]:
```
df
```

Out[6]:

	id	cuisine	ingredients
0	10259	greek	[romaine lettuce, black olives, grape tomatoes...
1	25693	southern_us	[plain flour, ground pepper, salt, tomatoes, g...
2	20130	filipino	[eggs, pepper, salt, mayonaise, cooking oil, g...
3	22213	indian	[water, vegetable oil, wheat, salt]
4	13162	indian	[black pepper, shallots, cornflour, cayenne pe...
...	...	...	...
39769	29109	irish	[light brown sugar, granulated sugar, butter, ...
39770	11462	italian	[KRAFT Zesty Italian Dressing, purple onion, b...
39771	2238	irish	[eggs, citrus fruit, raisins, sourdough starte...
39772	41882	chinese	[boneless chicken skinless thigh, minced garli...
39773	2362	mexican	[green chile, jalapeno chilies, onions, ground...

39774 rows × 3 columns

In [7]:
```
!pip install mysql.connector
```

```
Collecting mysql.connector
  Downloading mysql-connector-2.2.9.tar.gz (11.9 MB)
     ---------------------------------------- 11.9/11.9 MB 727.0 kB/s eta 0:00:00
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Building wheels for collected packages: mysql.connector
  Building wheel for mysql.connector (setup.py): started
  Building wheel for mysql.connector (setup.py): finished with status 'done'
  Created wheel for mysql.connector: filename=mysql_connector-2.2.9-cp39-cp39-win_amd
64.whl size=247946 sha256=15687f27692e39be4844aa1c8b8285990a58066cc4381d9f1359c17beed
823c2
  Stored in directory: c:\users\pmyls\appdata\local\pip\cache\wheels\7b\14\39\5aad423
666e827dfe9a1fbcd111ac17171e7c9865d570780ce
Successfully built mysql.connector
Installing collected packages: mysql.connector
Successfully installed mysql.connector-2.2.9
```

how to make a connection with mysql to get data

In [8]:
```
import mysql.connector
```

In [12]:
```
# to make a connection with mysql and get data from it and convert into pandas datafra

conn = mysql.connector.connect(host='localhost',user='root',password='',database='wor
```

In [15]:
```
# now to read the sql data we have a function in pandas 'pd.read_sql_quary()'

country = pd.read_sql_query('SELECT * FROM country',conn)
```

```
C:\Users\PMYLS\anaconda3\lib\site-packages\pandas\io\sql.py:762: UserWarning: pandas
only support SQLAlchemy connectable(engine/connection) ordatabase string URI or sqlit
e3 DBAPI2 connectionother DBAPI2 objects are not tested, please consider using SQLAlc
hemy
  warnings.warn(
```

In [16]:
```
country
```

Out[16]:

	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	
0	ABW	Aruba	North America	Caribbean	193.0	NaN	103000	78.4	
1	AFG	Afghanistan	Asia	Southern and Central Asia	652090.0	1919.0	22720000	45.9	
2	AGO	Angola	Africa	Central Africa	1246700.0	1975.0	12878000	38.3	
3	AIA	Anguilla	North America	Caribbean	96.0	NaN	8000	76.1	
4	ALB	Albania	Europe	Southern Europe	28748.0	1912.0	3401200	71.6	
...	...	...	...	...	...	...	...	...	
234	YEM	Yemen	Asia	Middle East	527968.0	1918.0	18112000	59.8	
235	YUG	Yugoslavia	Europe	Southern Europe	102173.0	1918.0	10640000	72.4	1
236	ZAF	South Africa	Africa	Southern Africa	1221037.0	1910.0	40377000	51.1	11
237	ZMB	Zambia	Africa	Eastern Africa	752618.0	1964.0	9169000	37.2	
238	ZWE	Zimbabwe	Africa	Eastern Africa	390757.0	1980.0	11669000	37.8	

239 rows × 15 columns

In [17]:
```
# if we want to get some specific data from mysql not the whole then we do this

pd.read_sql_query('SELECT * FROM city WHERE ID > 50',conn)
```

```
C:\Users\PMYLS\anaconda3\lib\site-packages\pandas\io\sql.py:762: UserWarning: pandas only support SQLAlchemy connectable(engine/connection) ordatabase string URI or sqlite3 DBAPI2 connectionother DBAPI2 objects are not tested, please consider using SQLAlchemy
  warnings.warn(
```

Out[17]:

	ID	Name	CountryCode	District	Population
0	51	Ech-Chleff (el-Asnam)	DZA	Chlef	96794
1	52	Ghardaïa	DZA	Ghardaïa	89415
2	53	Tafuna	ASM	Tutuila	5200
3	54	Fagatogo	ASM	Tutuila	2323
4	55	Andorra la Vella	AND	Andorra la Vella	21189
...	...	...	...	...	...
4024	4075	Khan Yunis	PSE	Khan Yunis	123175
4025	4076	Hebron	PSE	Hebron	119401
4026	4077	Jabaliya	PSE	North Gaza	113901
4027	4078	Nablus	PSE	Nablus	100231
4028	4079	Rafah	PSE	Rafah	92020

4029 rows × 5 columns

the above you see that using pd.read_sql_query() it provide a warning

In [ ]:

NOW WE DO THE SAME THINGS USING sqlalchemy library

In [18]:
```
from sqlalchemy import create_engine
```

In [ ]:

In [25]:
```
pip install pymysql
```

```
Collecting pymysql
  Downloading PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
Downloading PyMySQL-1.1.1-py3-none-any.whl (44 kB)
   ---------------------------------------- 45.0/45.0 kB 369.5 kB/s eta 0:00:00
Installing collected packages: pymysql
Successfully installed pymysql-1.1.1
Note: you may need to restart the kernel to use updated packages.
```

In [27]:
```
engine = create_engine('mysql+pymysql://root:''@localhost/world')

# Using pandas with SQLAlchemy engine to read from the 'city' table
df = pd.read_sql('SELECT * FROM city', con=engine)
```

In [28]:
```
df
```

Out[28]:

	ID	Name	CountryCode	District	Population
0	1	Kabul	AFG	Kabol	1780000
1	2	Qandahar	AFG	Qandahar	237500
2	3	Herat	AFG	Herat	186800
3	4	Mazar-e-Sharif	AFG	Balkh	127800
4	5	Amsterdam	NLD	Noord-Holland	731200
...	...	...	...	...	...
4074	4075	Khan Yunis	PSE	Khan Yunis	123175
4075	4076	Hebron	PSE	Hebron	119401
4076	4077	Jabaliya	PSE	North Gaza	113901
4077	4078	Nablus	PSE	Nablus	100231
4078	4079	Rafah	PSE	Rafah	92020

4079 rows × 5 columns

In [60]:
```
# some markdowns are as under
# df.to_markdown()
```

REVISION

NOW WE HAVE MADE TWO CONNECTIONS OF PYTHON WITH MYSQL DATABASE

As a revision we will discuss this again

Method 1

In [32]:
```
import mysql.connector
```

In [33]:
```
conn=mysql.connector.connect(host='localhost',user='root',password='',database='entry_
```

In [38]:
```
student_info = pd.read_sql_query('SELECT * FROM student_info',conn)
```

```
C:\Users\PMYLS\anaconda3\lib\site-packages\pandas\io\sql.py:762: UserWarning: pandas
only support SQLAlchemy connectable(engine/connection) ordatabase string URI or sqlit
e3 DBAPI2 connectionother DBAPI2 objects are not tested, please consider using SQLAlc
hemy
  warnings.warn(
```

In [39]:
```
student_info
```

Out[39]:

	ID	Name	FName	Gender
0	25	amir	shahid	Male
1	26	kazim	munir	Male
2	27	suhail	amaad	Male
3	28	ihtisham	intiaz	Male
4	29	Afaq	Ishtiaq	Male
5	30	adifakd	fahro	Female
6	31	shoaibm	zarm	Male

In [45]:
```
print('this the first method to make connection with mysql database using pandas pd.re
print('|*|'*60)
```

```
this the first method to make connection with mysql database using pandas pd.read_sql
_query()
|*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||
*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*||*
||*||*||*|
```

Method 2

In [47]:
```
from sqlalchemy import create_engine
import pymysql
```

In [57]:
```
conn = create_engine("mysql+pymysql://root@localhost/world")
```

In [58]:
```
pd.read_sql('SELECT * FROM city',con=conn)
```

Out[58]:

	ID	Name	CountryCode	District	Population
0	1	Kabul	AFG	Kabol	1780000
1	2	Qandahar	AFG	Qandahar	237500
2	3	Herat	AFG	Herat	186800
3	4	Mazar-e-Sharif	AFG	Balkh	127800
4	5	Amsterdam	NLD	Noord-Holland	731200
...	...	...	...	...	...
4074	4075	Khan Yunis	PSE	Khan Yunis	123175
4075	4076	Hebron	PSE	Hebron	119401
4076	4077	Jabaliya	PSE	North Gaza	113901
4077	4078	Nablus	PSE	Nablus	100231
4078	4079	Rafah	PSE	Rafah	92020

4079 rows × 5 columns

In [59]:
```
print('this is the sencond method to make a connection')
print('*'*70)
```

```
this is the sencond method to make a connection
**********************************************************************
```

In [61]:
```
print('The advantage of method 2 is that it even not provide warning')
```

```
The advantage of method 2 is that it even not provide warning
```

In [ ]:
```

```