Deal with Json data

convert json data into pamdas dataframe

In [2]:

```
import pandas as pd
```

In [4]:

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

In [7]:

```
df.head()
```

Out[7]:

	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...

Load json data through API and convert into dataframe

In [11]:

```
df = pd.read_json("https://api.exchangerate-api.com/v4/latest/INR")
```

In [12]:

```
df.head()
```

Out[12]:

	provider	WARNING_UPGRADE_TO_V6	terms	base	date	time_last_updated	r
AED	https://www.exchangerate-api.com	https://www.exchangerate-api.com/docs/free	https://www.exchangerate-api.com/terms	INR	2021-12-11	1639180802	0.(
AFN	https://www.exchangerate-api.com	https://www.exchangerate-api.com/docs/free	https://www.exchangerate-api.com/terms	INR	2021-12-11	1639180802	1.2
ALL	https://www.exchangerate-api.com	https://www.exchangerate-api.com/docs/free	https://www.exchangerate-api.com/terms	INR	2021-12-11	1639180802	1.4
AMD	https://www.exchangerate-api.com	https://www.exchangerate-api.com/docs/free	https://www.exchangerate-api.com/terms	INR	2021-12-11	1639180802	6.5
ANG	https://www.exchangerate-api.com	https://www.exchangerate-api.com/docs/free	https://www.exchangerate-api.com/terms	INR	2021-12-11	1639180802	0.(

In [23]:

```
df.shape
```

Out[23]:

```
(160, 7)
```

Deal with sql data

Step-1 Download XAMPP software ("https://www.apachefriends.org/index.html")# Step-2 Run apache and mysql server and open this url in chrome localhost/phpmyadmin/

Step-3 Create a new database and import that sql file

To communicate with mysql and python we install some library

In [25]:

```
! pip install mysql.connector
```

```
Collecting mysql.connector
  Downloading mysql-connector-2.2.9.tar.gz (11.9 MB)
     |████████████████████████████████| 11.9 MB 742 kB/s eta 0:00:01
Building wheels for collected packages: mysql.connector
  Building wheel for mysql.connector (setup.py) ... done
  Created wheel for mysql.connector: filename=mysql_connector-2.2.9-cp38-cp38-linux_x86_6
4.whl size=247948 sha256=b7783a77d45f86615e62fbaf0b18aadde6e13188012068de72cabb9ef695d6dc
  Stored in directory: /home/shivansh/.cache/pip/wheels/57/e4/98/5feafb5c393dd2540e44b064
a6f95832990d543e5b4f53ea8f
Successfully built mysql.connector
Installing collected packages: mysql.connector
Successfully installed mysql.connector
```

In [26]:

```
import mysql.connector
```

In [30]:

```
conn = mysql.connector.connect(host="localhost",user="shivansh",password="",database="sch
ool")
```

```
---------------------------------------------------------------------------
InterfaceError                            Traceback (most recent call last)
/tmp/ipykernel_93630/4191891603.py in <module>
----> 1 conn = mysql.connector.connect(host="localhost",user="shivansh",password="",datab
ase="school")

~/anaconda3/lib/python3.8/site-packages/mysql/connector/__init__.py in connect(*args, **k
wargs)
    177         return CMySQLConnection(*args, **kwargs)
    178     else:
--> 179         return MySQLConnection(*args, **kwargs)
    180 Connect = connect  # pylint: disable=C0103
    181

~/anaconda3/lib/python3.8/site-packages/mysql/connector/connection.py in __init__(self, *
args, **kwargs)
     93
     94         if len(kwargs) > 0:
---> 95             self.connect(**kwargs)
     96
     97     def _do_handshake(self):

~/anaconda3/lib/python3.8/site-packages/mysql/connector/abstracts.py in connect(self, **k
```

```
wargs)
    714
    715         self.disconnect()
--> 716         self._open_connection()
    717         self._post_connection()
    718

~/anaconda3/lib/python3.8/site-packages/mysql/connector/connection.py in _open_connection
(self)
    206         self._socket.open_connection()
    207         self._do_handshake()
--> 208         self._do_auth(self._user, self._password,
    209                       self._database, self._client_flags, self._charset_id,
    210                       self._ssl)

~/anaconda3/lib/python3.8/site-packages/mysql/connector/connection.py in _do_auth(self, u
sername, password, database, client_flags, charset, ssl_options)
    142             auth_plugin=self._auth_plugin)
    143         self._socket.send(packet)
--> 144         self._auth_switch_request(username, password)
    145
    146         if not (client_flags & ClientFlag.CONNECT_WITH_DB) and database:

~/anaconda3/lib/python3.8/site-packages/mysql/connector/connection.py in _auth_switch_req
uest(self, username, password)
    167             auth = get_auth_plugin(new_auth_plugin)(
    168                 auth_data, password=password, ssl_enabled=self._ssl_active)
--> 169             response = auth.auth_response()
    170             self._socket.send(response)
    171             packet = self._socket.recv()

~/anaconda3/lib/python3.8/site-packages/mysql/connector/authentication.py in auth_respons
e(self)
     76         """
     77         if self.requires_ssl and not self._ssl_enabled:
---> 78             raise errors.InterfaceError("{name} requires SSL".format(
     79                 name=self.plugin_name))
     80         return self.prepare_password()

InterfaceError: sha256_password requires SSL
```

In [ ]:

```
pd.read_sql_query("SELECT * FROM city",conn)
```