1.what is the process for loading a dataset from an external source?

ANS:

When you load data from an external source, you **load it into a suspense table**. You can then review the data in the suspense table and modify it. To load data into the suspense table, position the source file or tape, specify the location of the source, and run the appropriate load external data process.

2.How can we use pandas to read JSON files?

ANS:

Reading JSON files using pandas:-

To read the files, we
use **read_json()** function and through it, we pass the path to the JSON file we want to read. Once we do that, it returns a “DataFrame”(A table of rows and columns) that stores data. If we want to read a file that is located on remote servers then we pass the link to its location instead of a local path.

Example 1: Reading JSON file

Python3

```
import pandas as pd




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

df.head()

Output:

One Two 0 60 110 1 60 117 2 60 103 3 45 109 4 45 117 5 60 102

Example 2: Creating JSON data and reading in dataframe

Here we will create JSON data and then create a dataframe through it using **pd.Dataframe() methods.**

Python3

import pandas as pd

data = {

```
"One": {
    "0": 60,
    "1": 60,
    "2": 60,
    "3": 45,
    "4": 45,
    "5": 60
},
"Two": {
    "0": 110,
    "1": 117,
```

```
        "2": 103,
        "3": 109,
        "4": 117,
        "5": 102
    }
}

df = pd.DataFrame(data)

print(df)
```

Output:

One Two 0 60 110 1 60 117 2 60 103 3 45 109 4 45 117 5 60 102

3.Describe the significance of DASK.

ANS:

Dask is a free and open-source library for parallel computing in Python. Dask helps you scale your data science and machine learning workflows. Dask makes it easy to work with Numpy, pandas, and Scikit-Learn, but that’s just the beginning. Dask is a framework to build distributed applications that has since been used with dozens of other systems like XGBoost, PyTorch, Prefect, Airflow, RAPIDS, and more. It’s a full distributed computing toolbox that fits comfortably in your hand.

If you have larger-than-memory data, you can use Dask to *scale up* your workflow to leverage all the cores of your local workstation, or even *scale out* to the cloud.

Dask is convenient on a laptop. It installs trivially with conda or pip and extends the size of convenient datasets from “fits in memory” to “fits on disk”. Dask can scale to a cluster of 100s of machines. It is **resilient, elastic, data local, and low latency**.

4.Describe the functions of DASK.
ANS:

Dask is a free and open-source library for parallel computing in Python.
Dask **helps you scale your data science**

and machine learning workflows. Dask makes it easy to work with Numpy, pandas, and Scikit-Learn, but that's just the beginning.

In simple words, **Dask arrays are distributed numpy arrays**! Every operation on a Dask array triggers operations on the smaller numpy arrays, each using a core on the machine. Thus all available cores are used simultaneously enabling computations on arrays which are larger than the memory size.

5.Describe Cassandra's features.
ANS:

Features of Cassandra

Apache Cassandra is an open source, user-

available, distributed, NoSQL DBMS which is designed to handle large amounts of data across many servers. It provides zero point of failure. Cassandra offers massive support for clusters spanning multiple datacentres.

There are some massive features of Cassandra. Here are some of the features described below:

- **Distributed:**

Each node in the cluster has has same role. There's no question of failure & the data set is distributed across the cluster but one issue is there that is the master isn't present in each node to support request for service.

- **Supports replication & Multi data center replication:**

Replication factor comes with best configurations in cassandra. Cassandra is

designed to have a distributed system, for the deployment of large number of nodes for across multiple data centers and other key features too.

- **Scalability:**

It is designed to r/w throughput, Increase gradually as new machines are added without interrupting other applications.

- **Fault-tolerance:**

Data is automatically stored & replicated for fault-tolerance. If a node Fails, then it is replaced within no time.

- **MapReduce Support:**

It supports Hadoop integration with MapReduce support.Apache Hive & Apache Pig is also supported.

- **Query Language:**

Cassandra has introduced the CQL (Cassandra Query Language). Its a simple

interface for accessing the Cassandra.

Cassandra Query Language (CQL) :
CQL has simple interface for accessing the Cassandra, also an alternative for the traditional SQL. CQL adds an abstraction layer to hide the implementation of structure & also provides the native syntax for collections.

For example please follow the given sample which shows how to create a keyspace including column family in CQL 3.0-

CREATE KEYSPACE MyKeySpace WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 3 }; USE MyKeySpace; CREATE COLUMNFAMILY MyColumns (id text, Last text, First text, PRIMARY KEY(id)); INSERT INTO MyColumns (id, Last, First) VALUES ('1', 'Doe', 'John');

Query:

SELECT * FROM MyColumns;

Which gives:

id | First | Last ----+-------+------ 1 | Ratul | Sarkar

(1 rows)

Some facts regarding Cassandra are as follows:

- Before the updates of versions of Cassandra, upto Cassandra 1.0, Cassandra wasn’t row level consistent, which means inserting & updating the table. It may affect the same row that are processed at approximately the same time may affect the non-key columns in a inconsistent manner.

Cassandra 1.1 solved this using row level isolation.

- Deletion of markers called the Tombstones (source Internet) are also known to causes performance degradation upto severe consequence levels.

- Cassandra, essentially a hybrid between a key-value & a organised tabular DBMS.Tables can be created, dropped and altered at run time without blocking updates & queries.

- A column family called table represents a RDBMS. Each row is specifically identified by a row & key, name, value, timestamp etc. A table in Cassandra is a disturbed multi dimensional map monitored by a key. Further more applications are specified by a super column family.