PySpark for **ETL**:

* Performance
* Ease of Use
* Scalability
* Rich Ecosystem

The PySpark ETL **Workflow:**  
**Extract**: Retrieve data from various sources like databases, files, or APIs.

* Initialize a Spark session.
* Define the external source and target paths.
* Extract: Read data from an CSV file  
   df = spark.read.csv(source\_path, header=True,schema = ‘column names’).

**Transform**: Clean, aggregate, and manipulate data to fit your analysis needs.

* Using concat(),filter(),adding a column,group by ,avg(),orderby(),floor()

**Load:** Store the transformed data into a database or data warehouse for analysis.

* Save the transformed data to an CSV file  
   df.write.csv(target\_path, mode=”overwrite”, header=True)

Why spark:

**1.Speed**

Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.

**2. Ease of Use**

Write applications quickly in Java, Scala, Python, R.

**3. Generality**

Spark powers a stack of libraries including SQL and DataFrames, MLlib for machine learning,

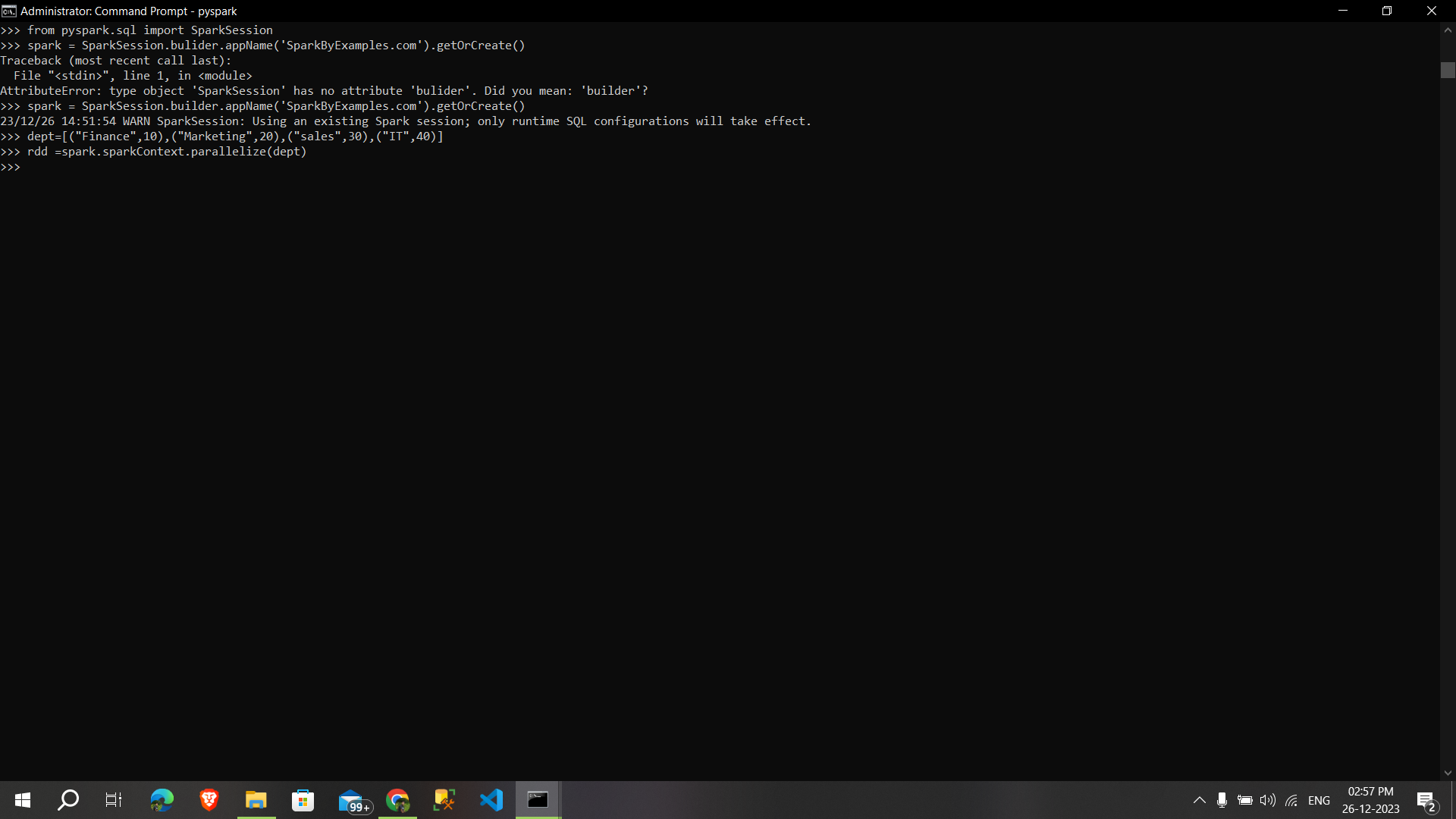
GraphX, and Spark Streaming.

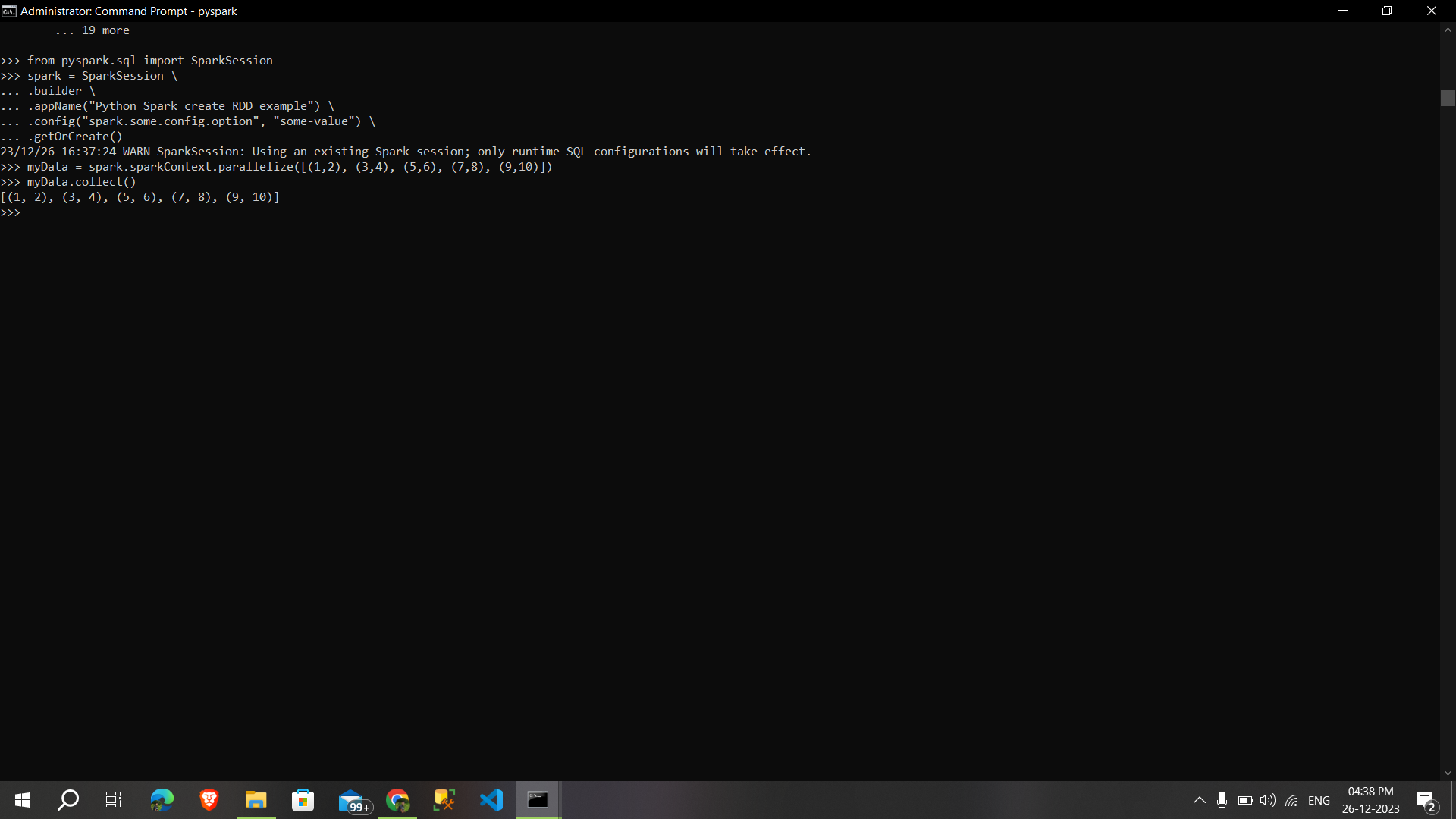
**4. Runs Everywhere**

Spark runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, and S3.

Create **RDD:**

by using **parallelize()** function





By using **createDataFrame( )** function:

By using **read and load** functions:

1. Read dataset from **.csv file**

2. Read dataset from **DataBase**

3. Read dataset from **HDFS**

**Spark Components:**

1. Spark Driver

• separate process to execute user applications

• creates SparkContext to schedule jobs execution and negotiate with cluster manager

2. Executors

• run tasks scheduled by driver

• store computation results in memory, on disk or off-heap

• interact with storage systems

3. Cluster Manager

• Mesos

• YARN

• Spark Standalone

The following code is **not working** in my local system mam,i’ve checked a lot and still can't find the error.(pysparkruntimeerror)

