PYSPARK

ASSESSMENT

S.R.TAANUSRI

27.12.2023

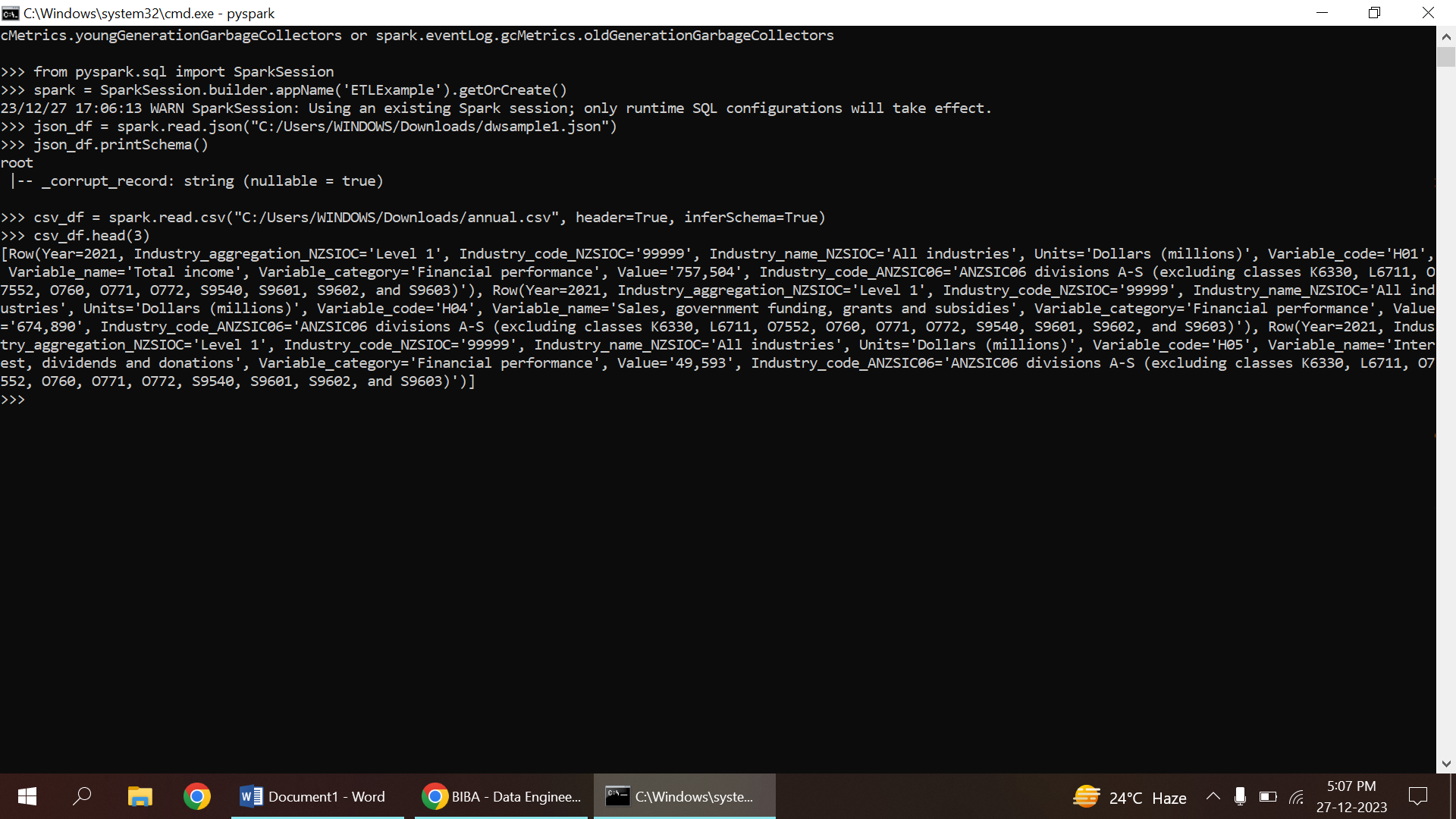
1. Implement Processing JSON and CSV data with PySpark

Explain ETL (Extract, Transform, Load) with PySpark

Using Spark SQL - Creating databases, tables

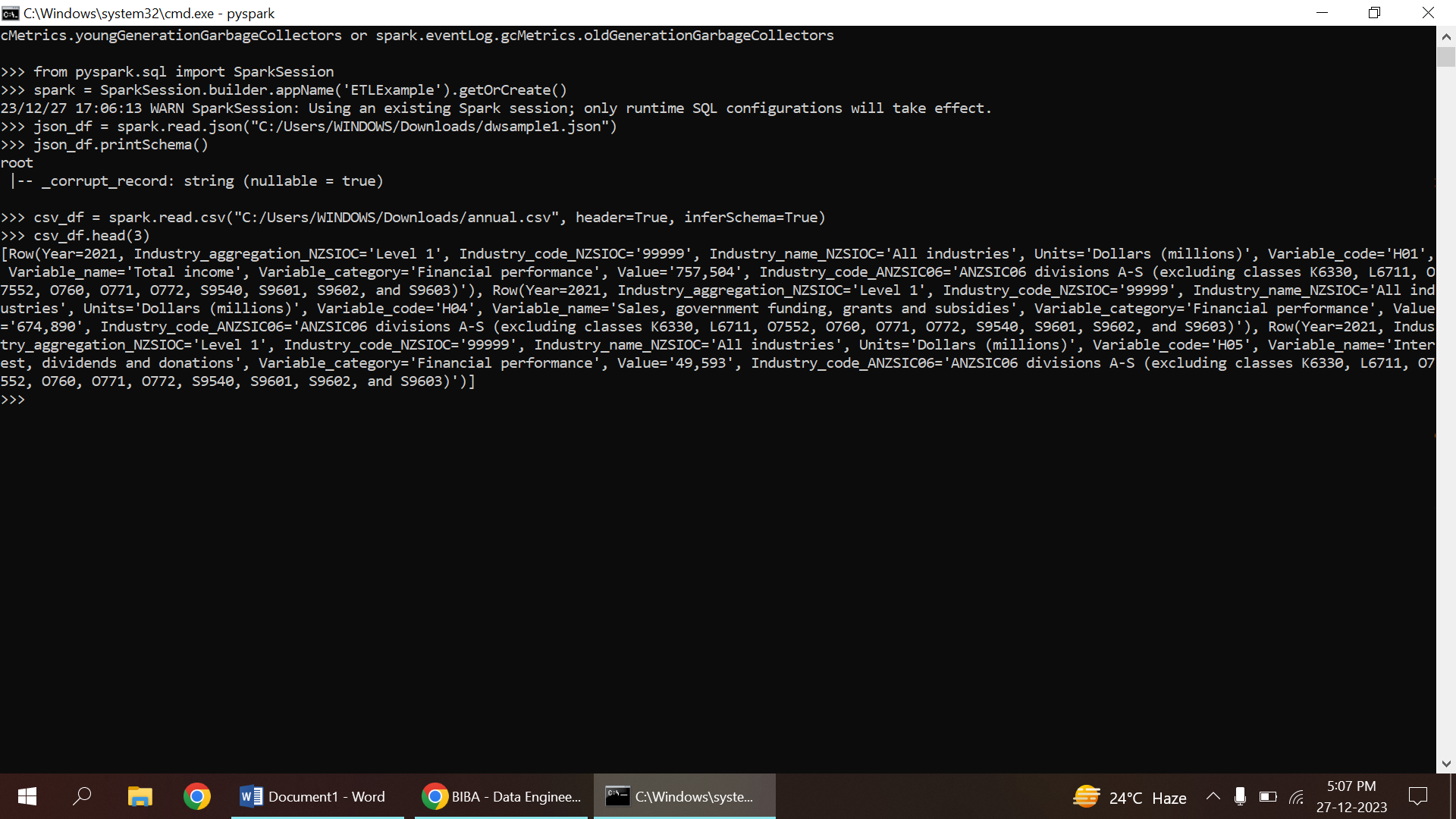
Using Spark SQL - Transformations such as Filter, Join, Simple Aggregations, GroupBy.

1. Implement Processing JSON and CSV data



2.ETL

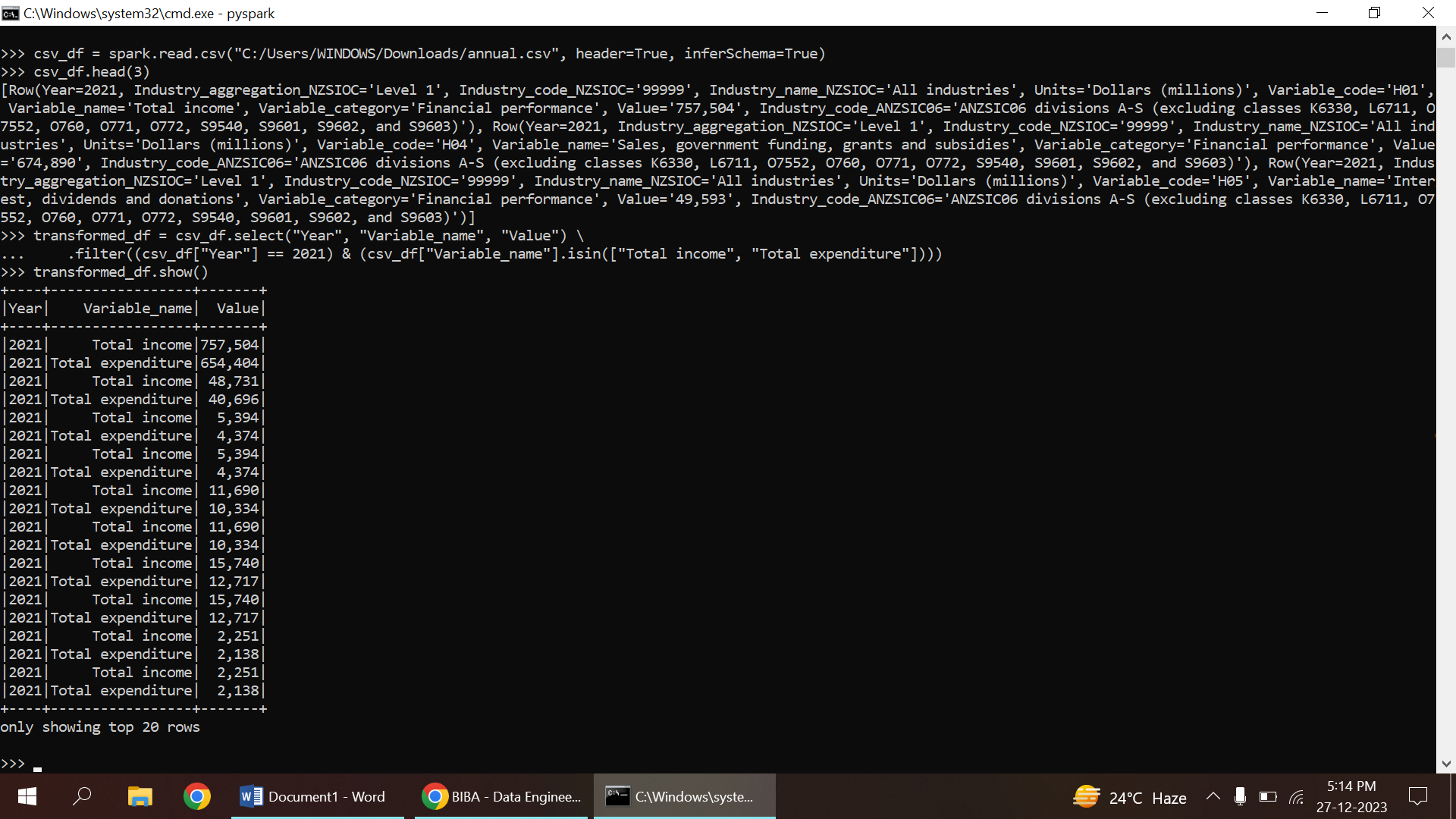
a.EXTRACT(data extraction)



STEPS:

1. Initialize SparkSession
2. Read JSON data into a DataFrame
3. Read CSV data into a DataFrame

b.TRANSFORMATION

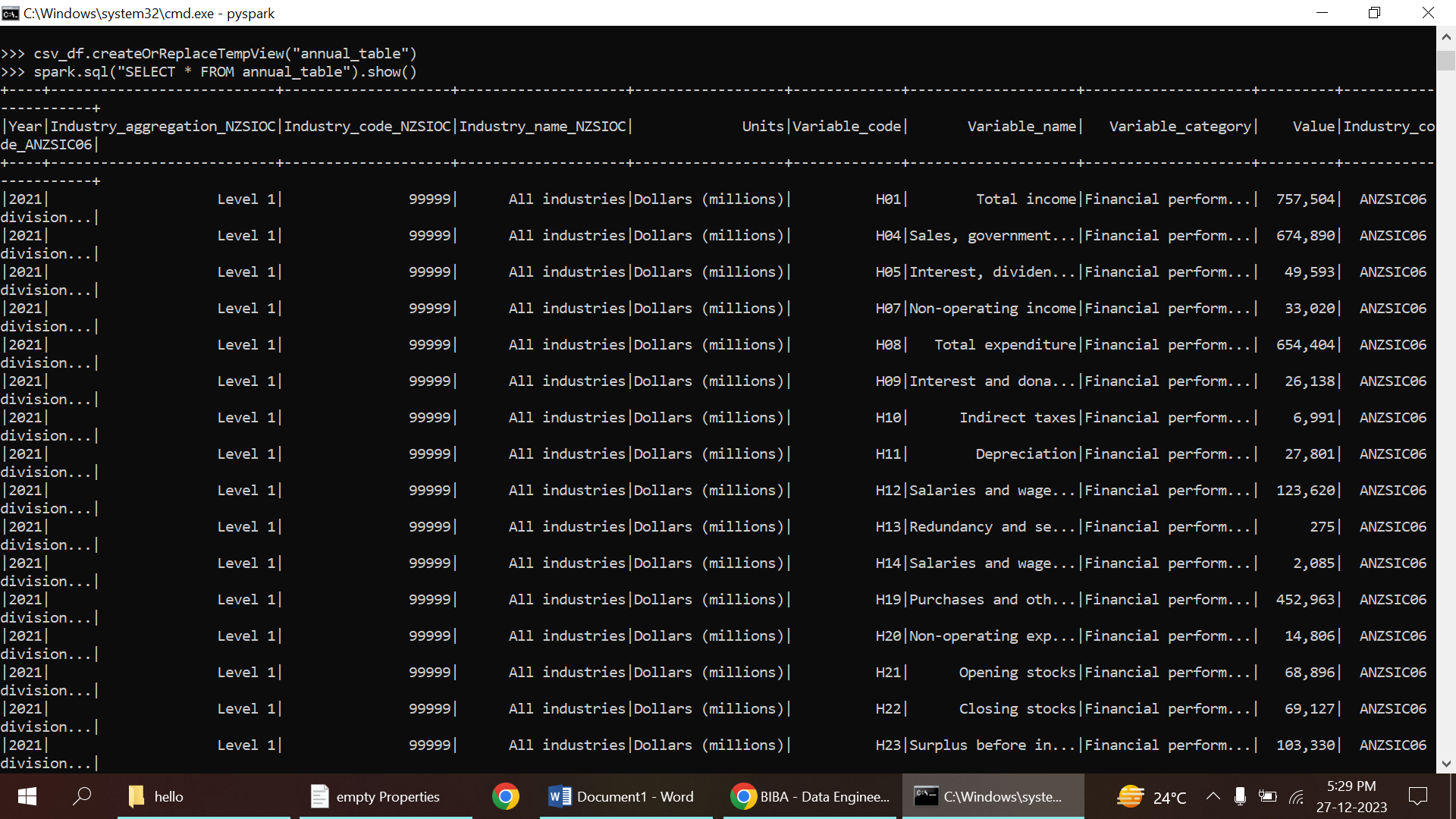


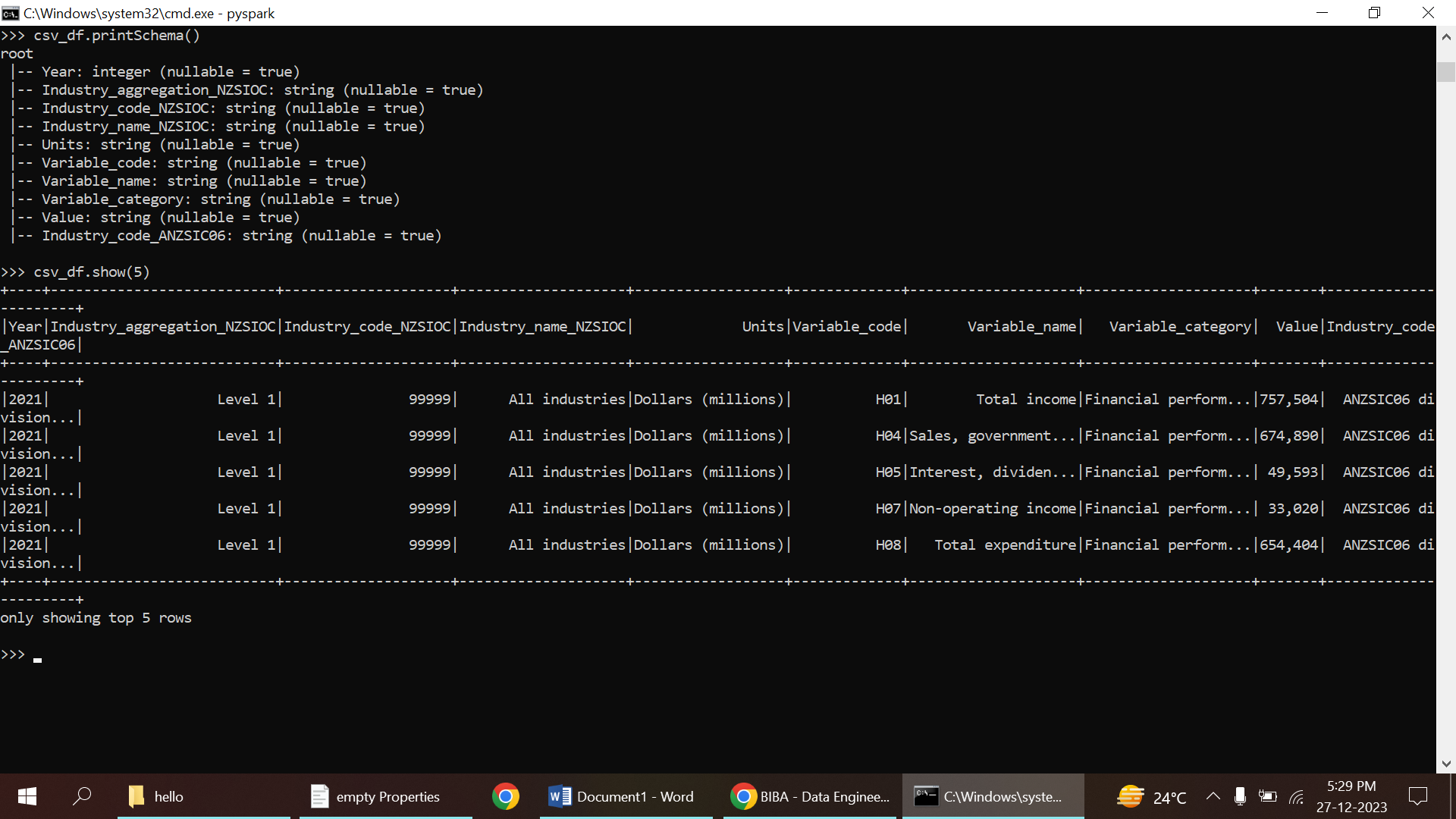
STEPS

1. Selecting required columns and filtering specific rows
2. Showing the transformed DataFrame

* Selects specific columns ("Year", "Variable\_name", "Value") from the CSV DataFrame and filters rows where the "Year" column is 2021 and the "Variable\_name" is either "Total income" or "Total expenditure".
* Displays the transformed DataFrame.

c.LOAD

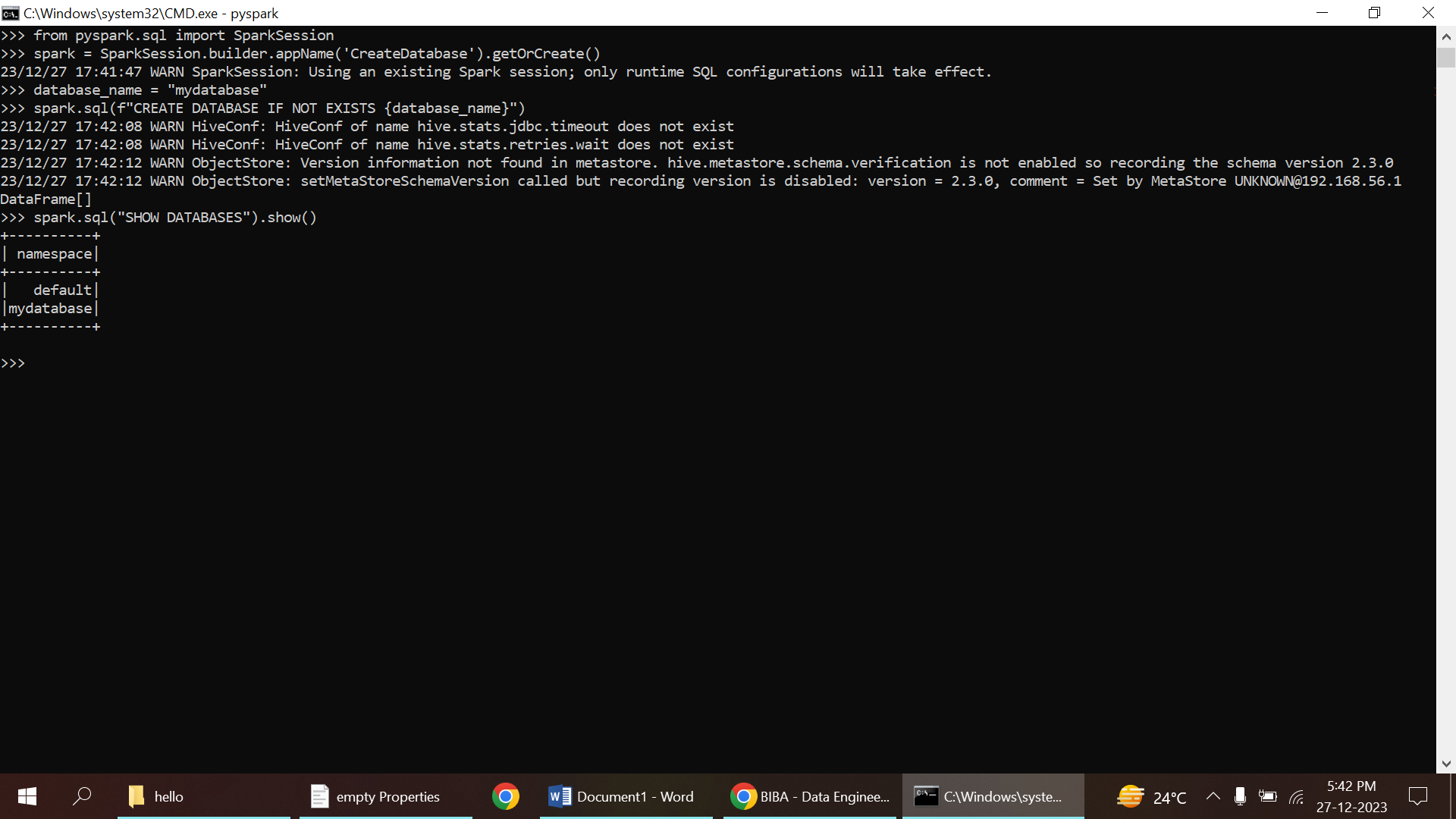




STEPS:

1. Create a temporary table from the DataFrame for SQL operations
2. can perform Spark SQL operations on this temporary table
3. Show the schema and some sample data

3.CREATE DATA BASE

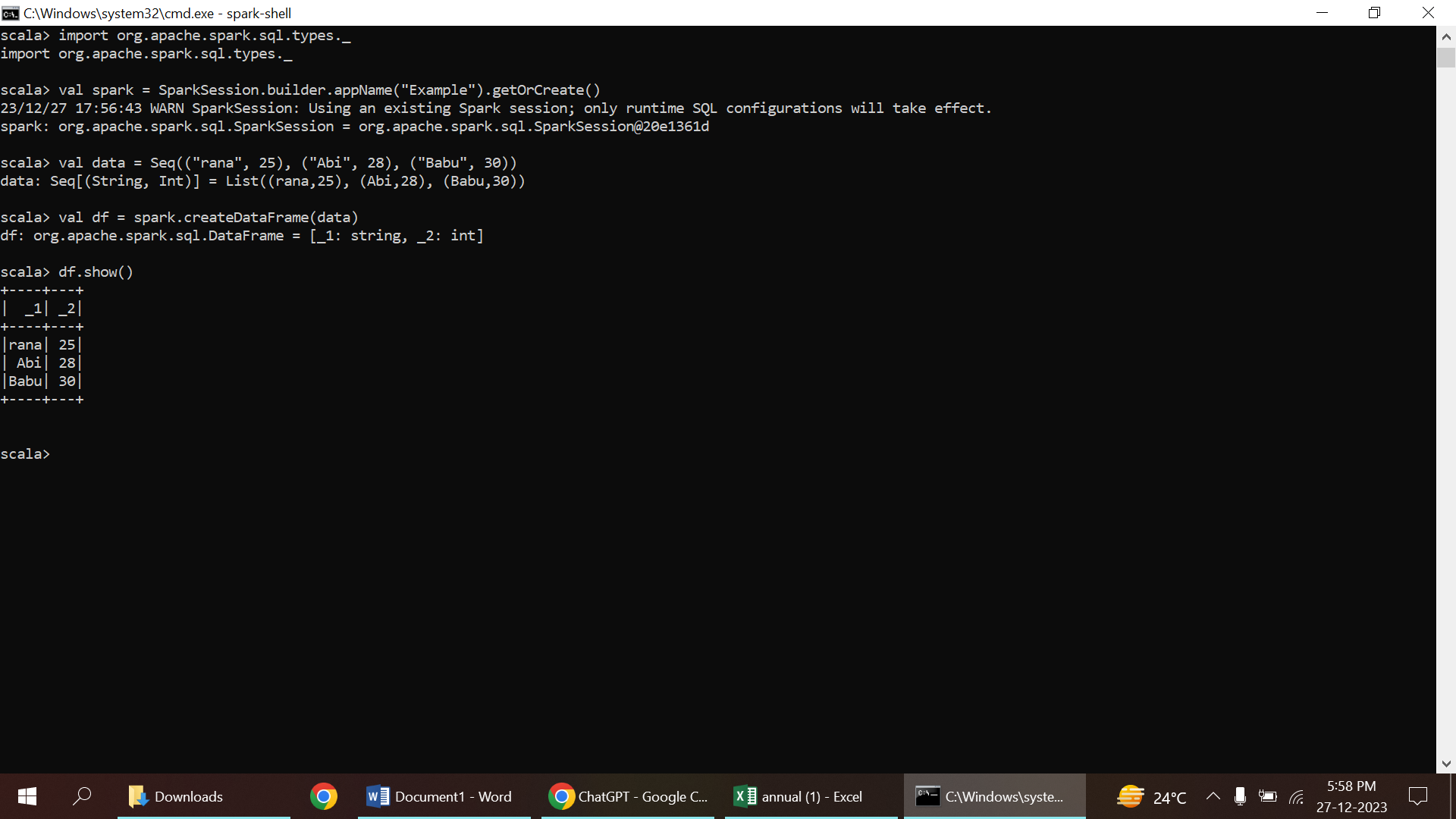


STEPS

1. Creating a Database
2. Verifying Database Creation

* a database named "mydatabase" using the spark.sql() method.
* The IF NOT EXISTS clause ensures that the database is created only if it does not already exist.
* The .show() function displays the result of the executed SQL command.

CREATE TABLE IN PYSPARK

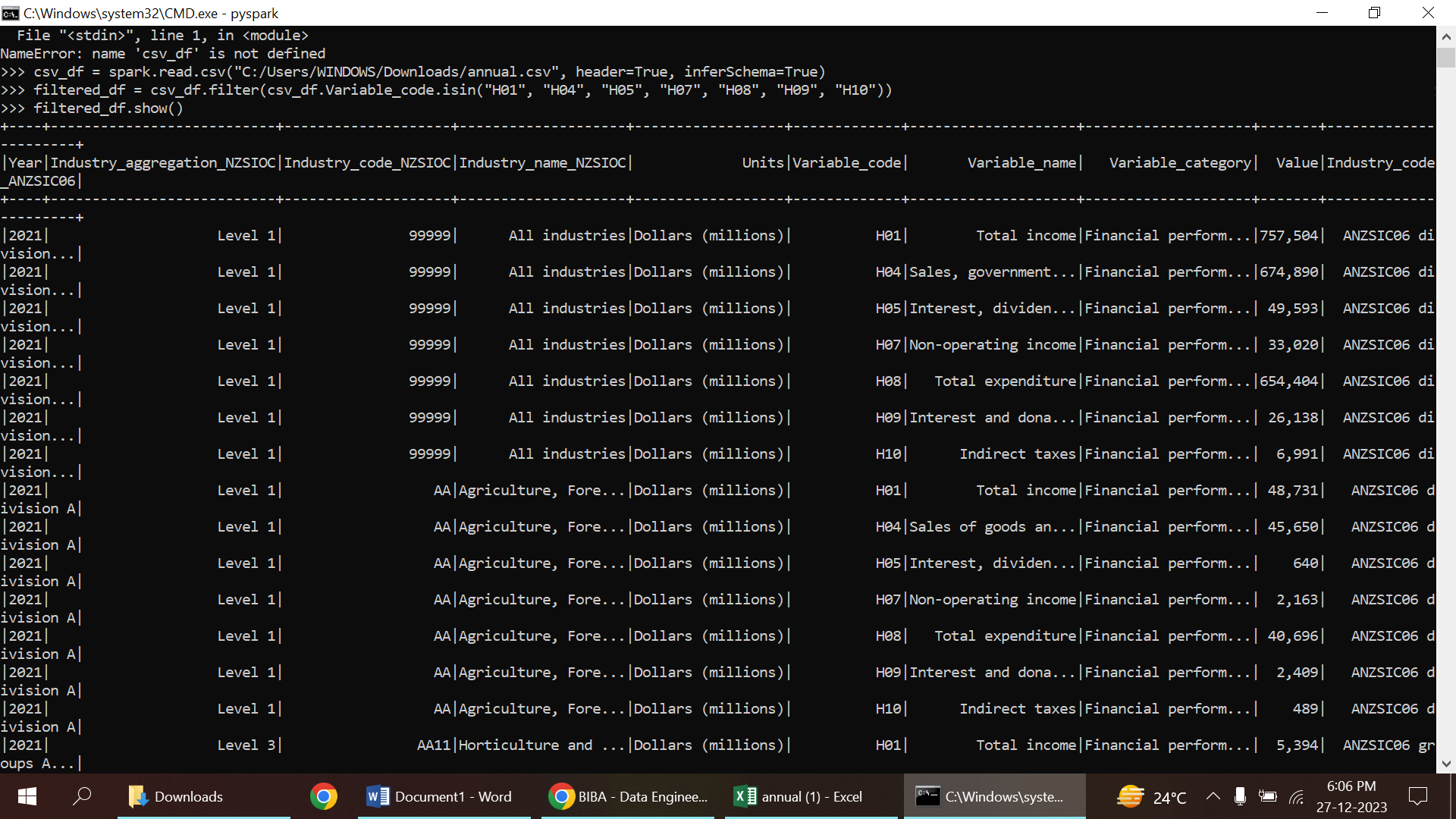


STEPS:

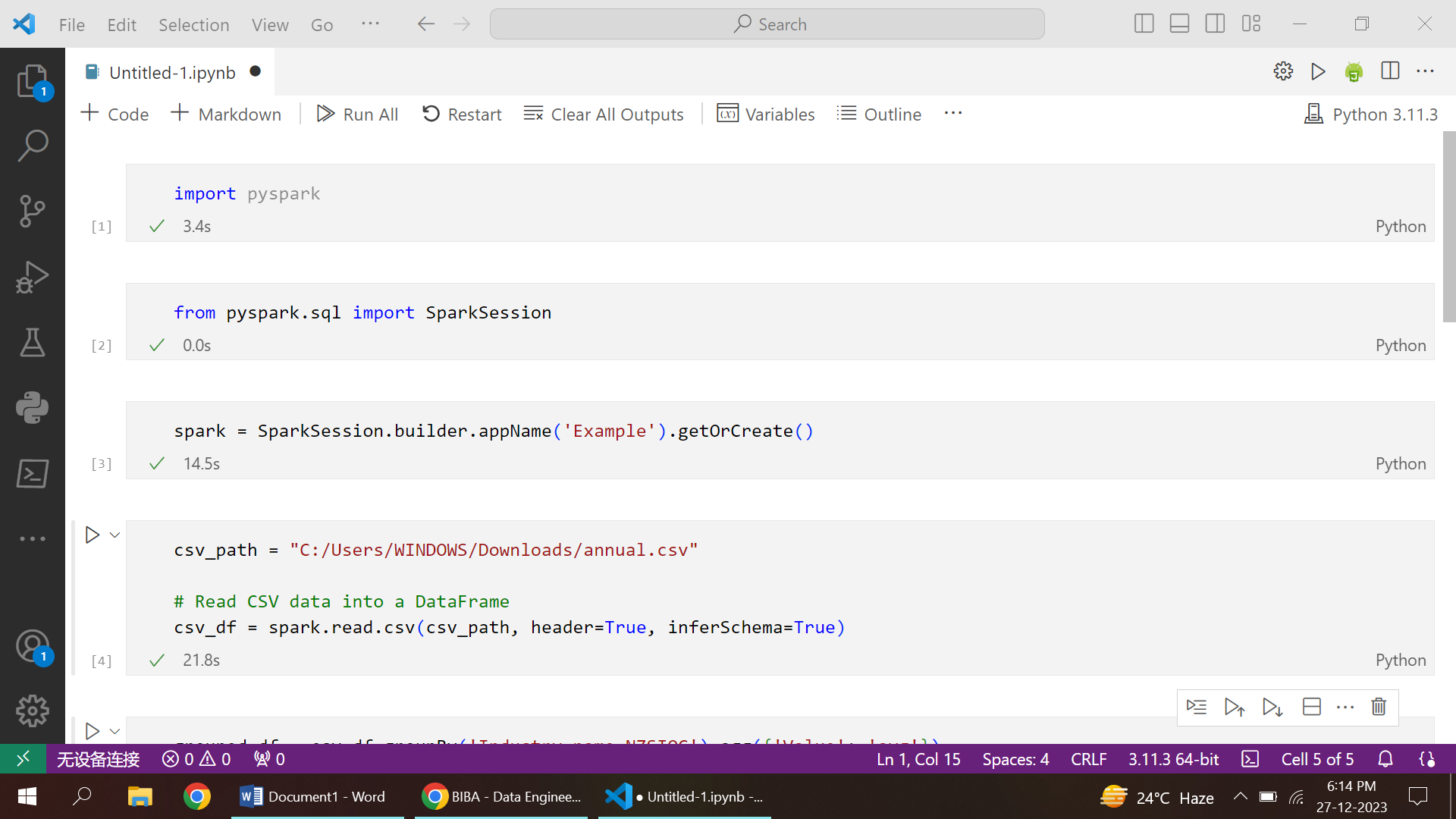
1. Sample data
2. Create DataFrame
3. Show the contents of the DataFrame

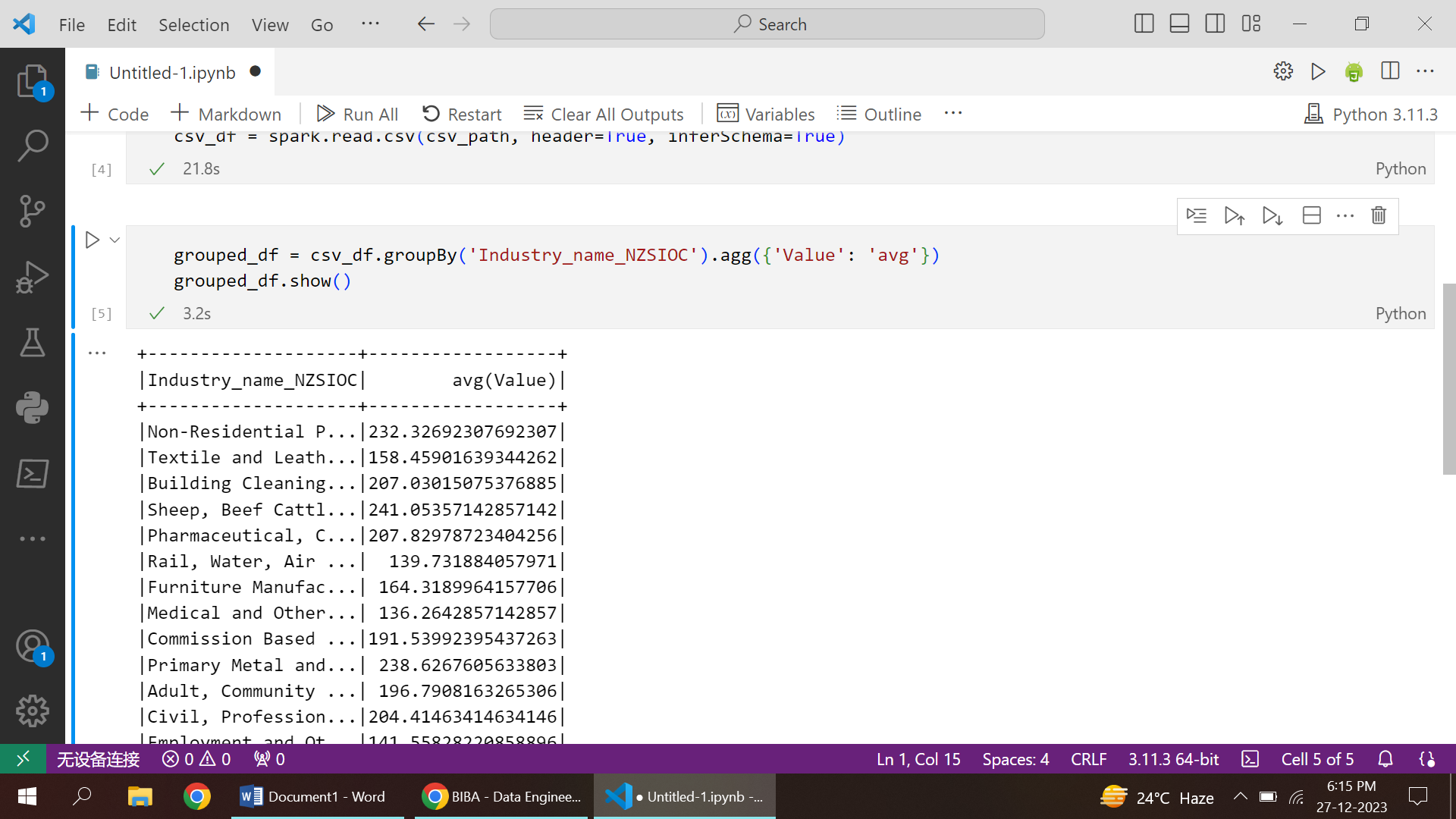
**4.Filter, Simple Aggregations, GroupBy.**

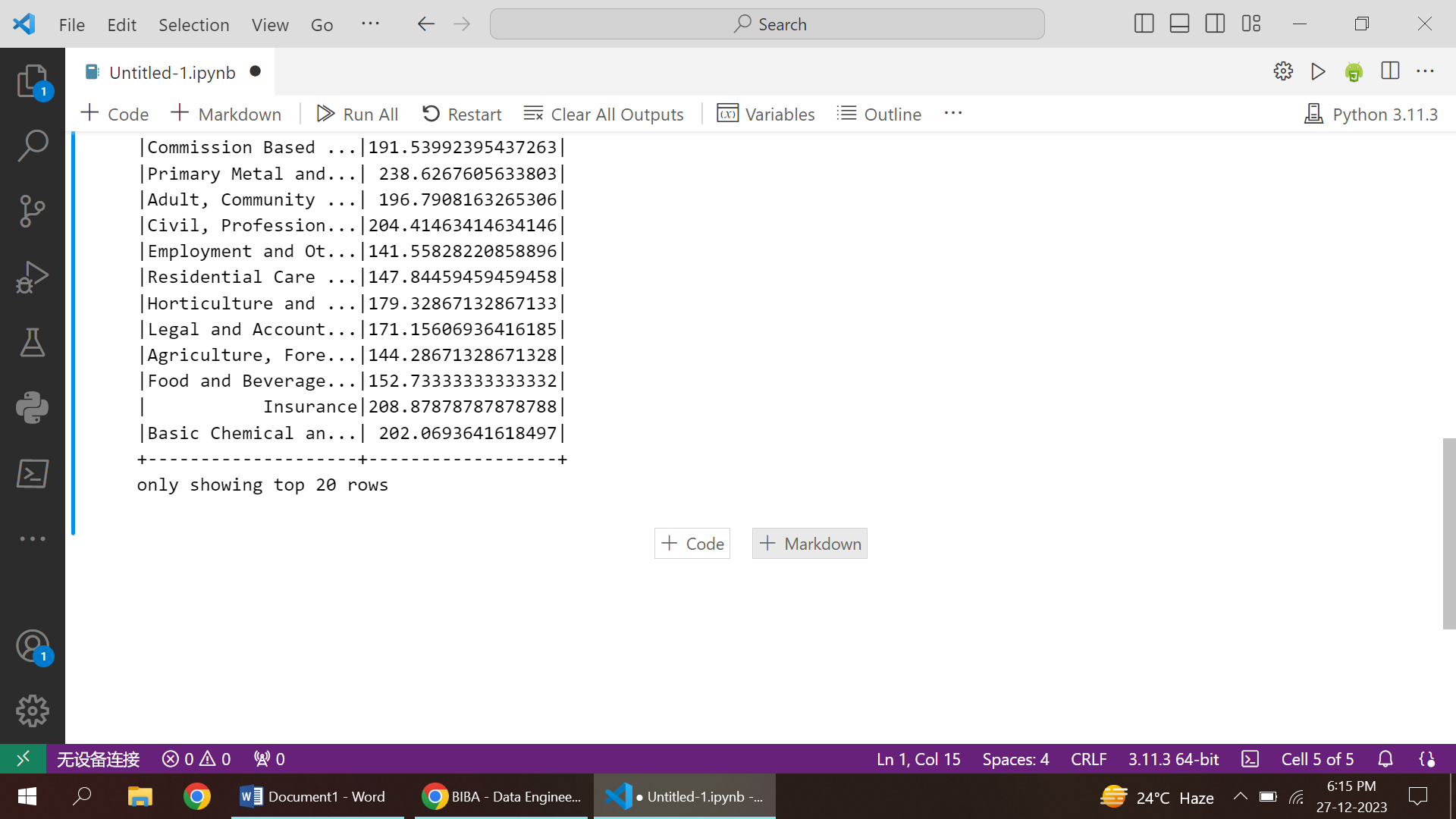
**FILTER**



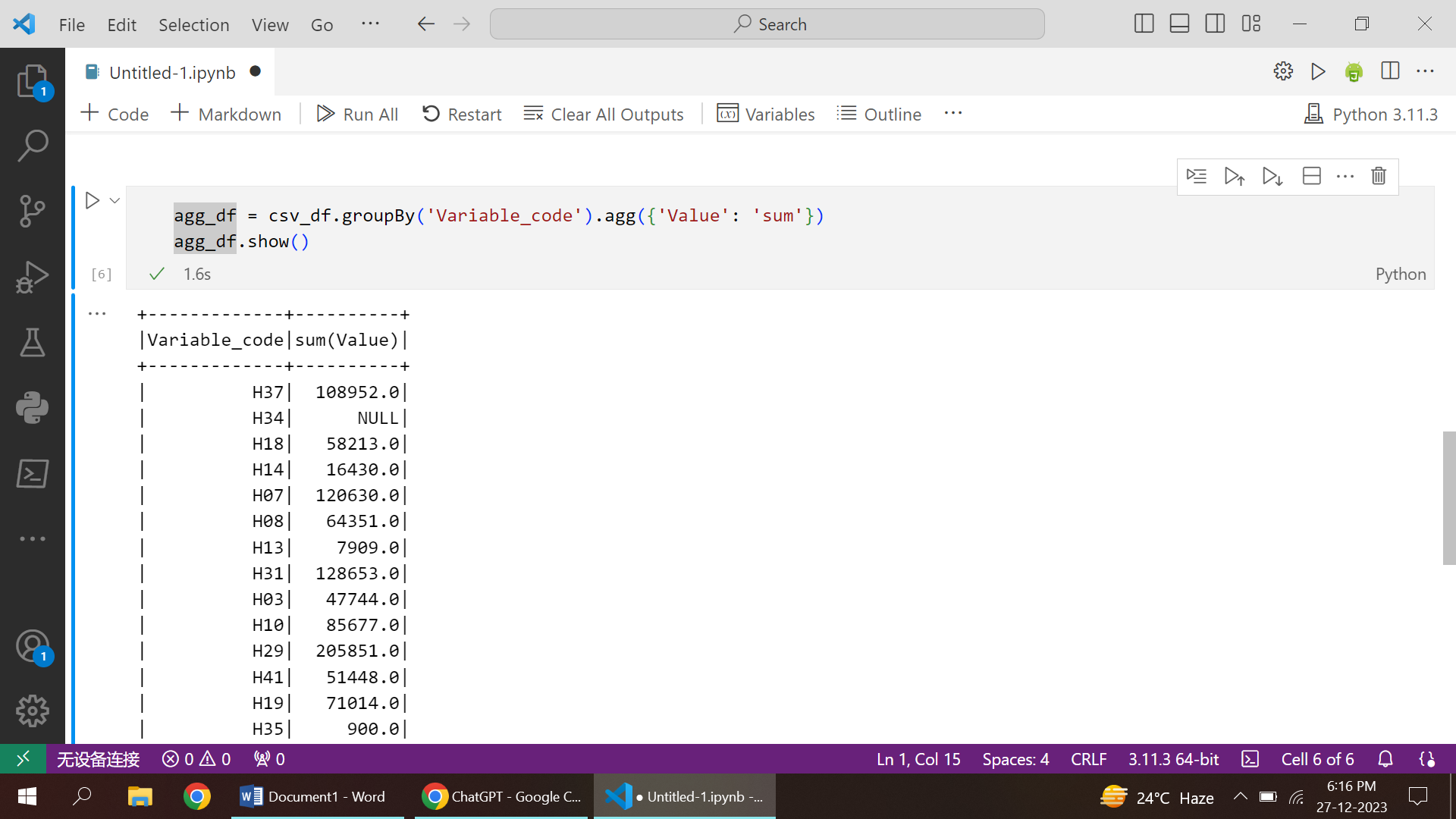
**GROUP BY**







**AGGREGATE**



* group the data in a DataFrame based on one or more columns
* compute summary statistics or aggregations on DataFrame columns
* function is used to extract rows from a DataFrame that satisfy specific conditions.