**PySpark Coding Assessment(27-12-2023)**

Coding Assesment:

1) Implement Processing JSON and CSV data with PySpark.

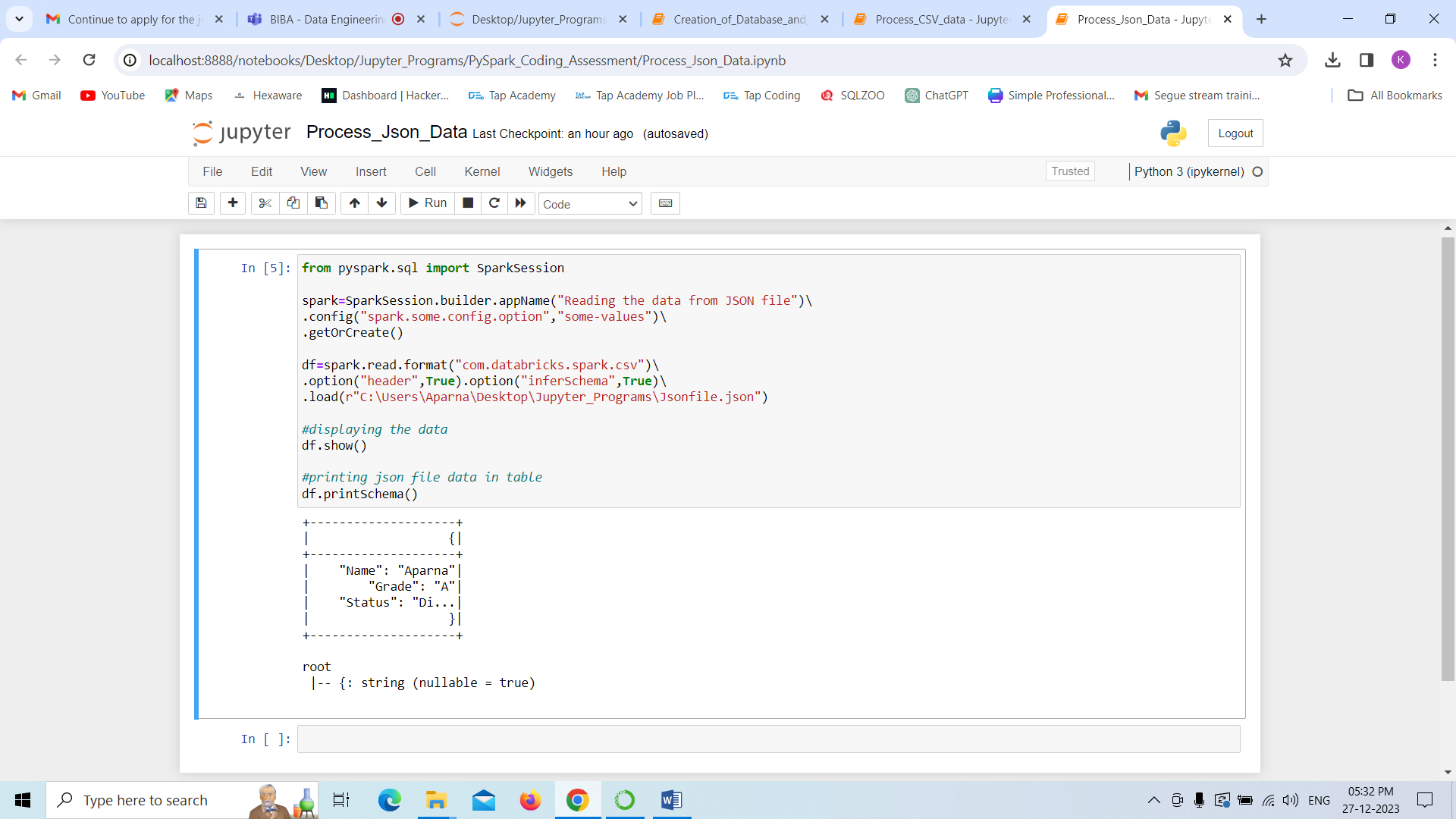
2)Explain ETL (Extract, Transform, Load) with PySpark

3)Using Spark SQL - Creating databases, tables

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

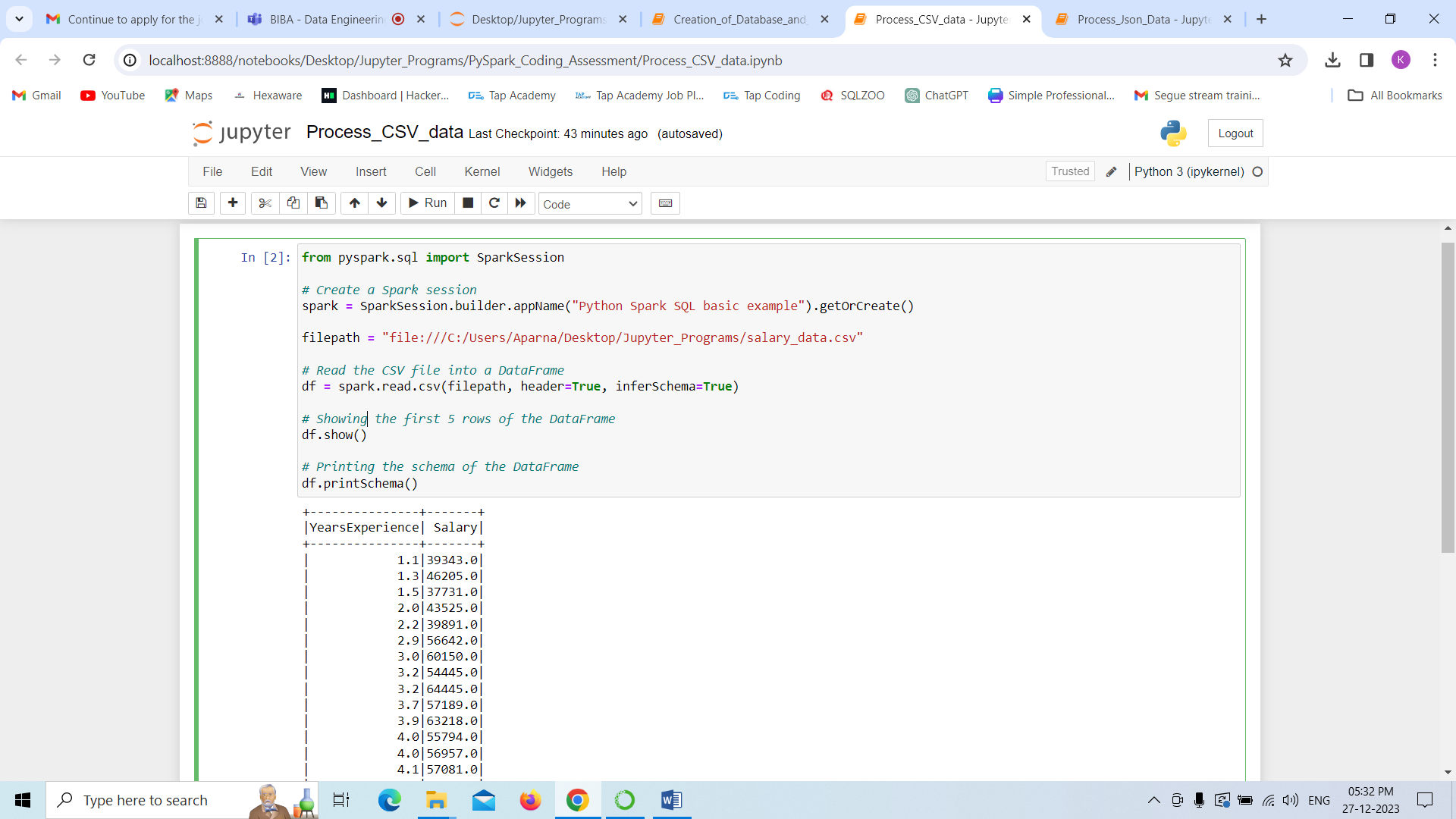
**1)Implement Processing JSON data with PySpark.**

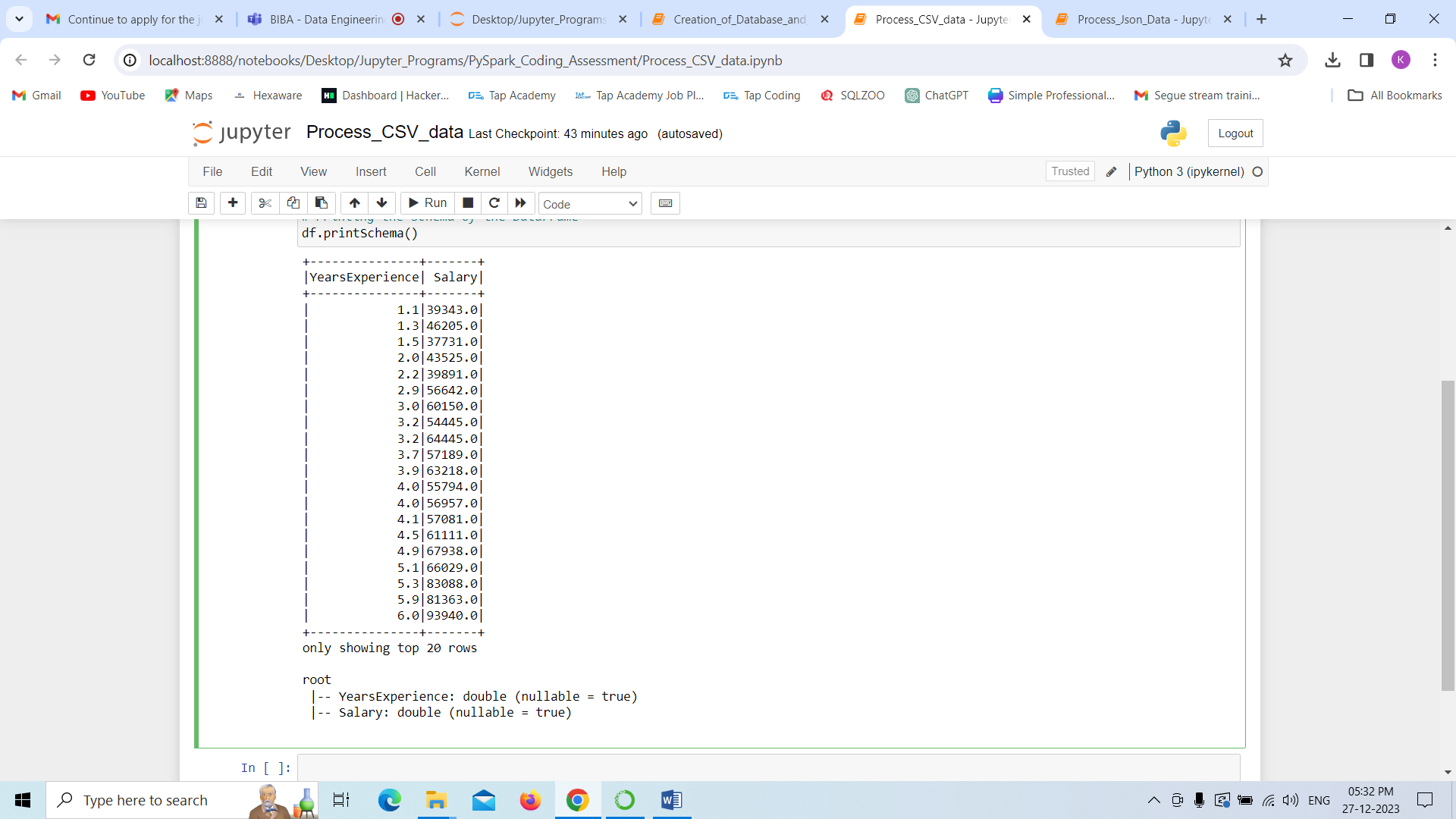
* Here,I am importing the SparkSession.
* Reading the data in JSON file with the help of spark.read.format function.
* Finally,I am displaying the data with show() in table format with printSchema function.



**Implement Processing CSV data with PySpark.**

* Here,I am importing the SparkSession.
* Reading the data in CSV file with the help of spark.read.format function.
* Finally,I am displaying the data with show() in table format with printSchema function.





**2)Explain ETL (Extract, Transform, Load) with PySpark**

**ETL :**Basically ETL stands for Extract ,Tranform ,Load

ETL (Extract, Transform, Load) is a process commonly used in data warehousing and data integration to collect, clean, transform, and store data from various sources into a data repository or data warehouse.

Let’s see what each phase will do :

**Extract:**

In this phase, we will extract the data is gathered from multiple sources, which can include databases, applications, files, APIs, and more.

**Transform:**

In this transform phase, the extracted data is processed to meet the requirements of the target system or data warehouse. Transformations include cleaning, filtering, aggregating, and converting data into a standardized format.

**Load:**

In the load phase, the transformed data is loaded into the target data warehouse or database, making it available for reporting, analytics, and business intelligence.

Advantages:

**Data Integration:**

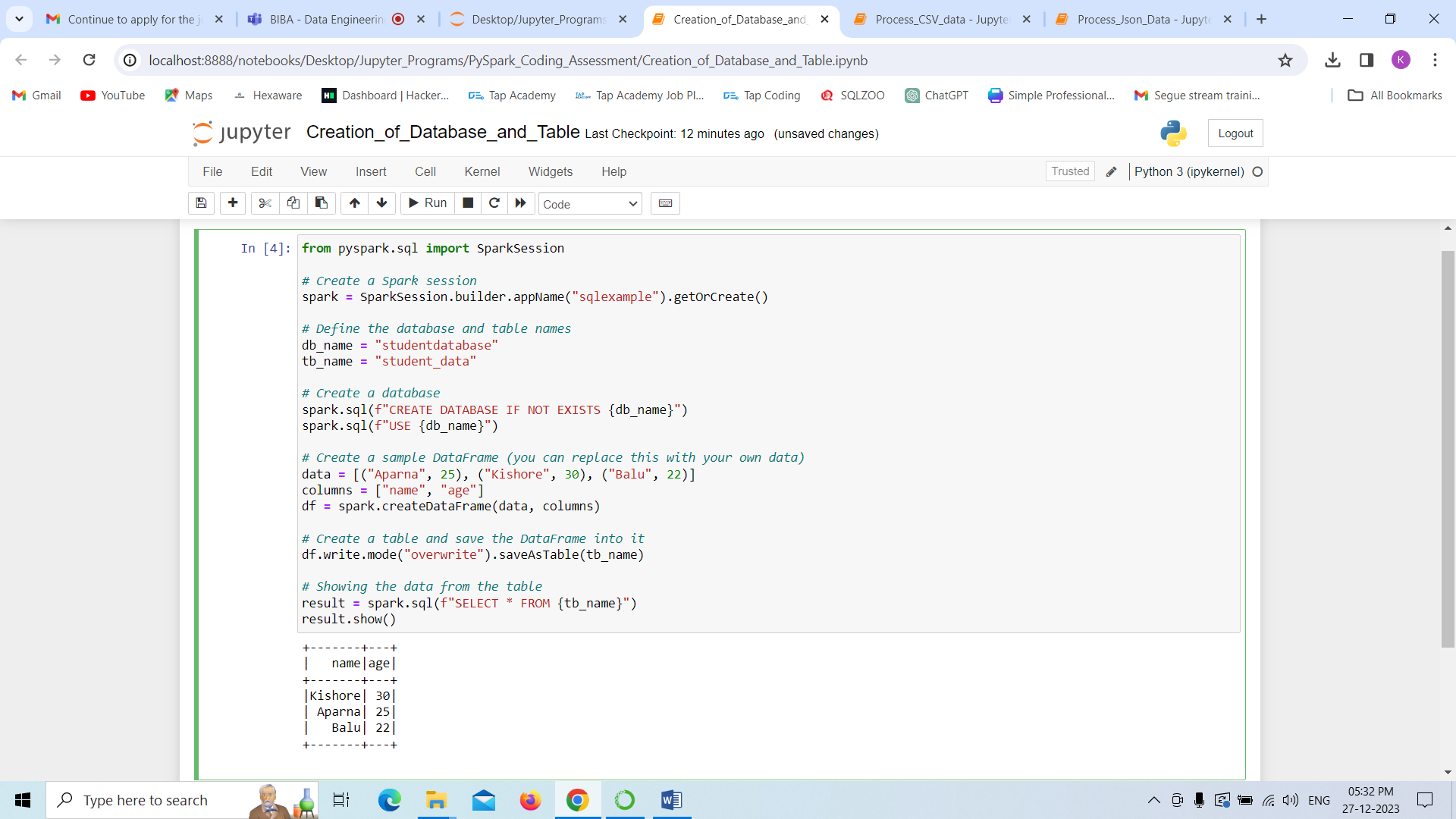
ETL allows integration of data from various sources into a centralized repository, we can see unified view of the data.

**Data Quality and Consistency:**

This tool processes often include data cleaning and transformation, improving data quality and ensuring the consistency across the organization where we are using.

**3)Using Spark SQL - Creating databases, tables :**

* Here we are creating database as studentdatabase
* a table that is named as Student\_data.
* Along with that I have inserted records into it.
* Finally,I am displaying the data with the help of show() function.



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

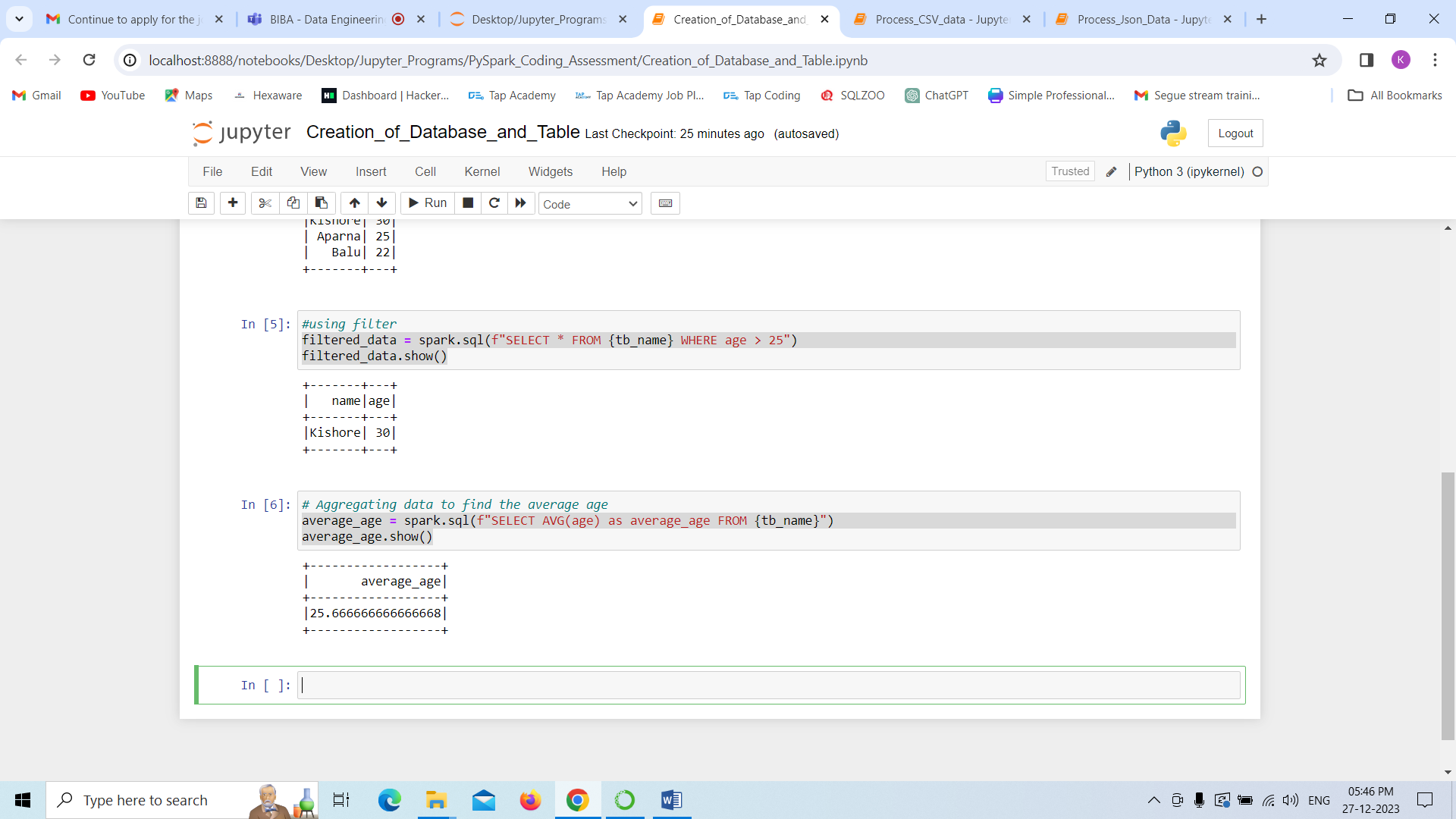
**Using Filter and aggregating function that is displaying average data**

**Filter-**

* The filter transformation is used to filter rows of a DataFrame based on a specified condition.
* Here, 25 is the condition, indicating that we want to keep only those rows where the "Age" column has a value greater than 25.

**Aggregation(Avg function):**

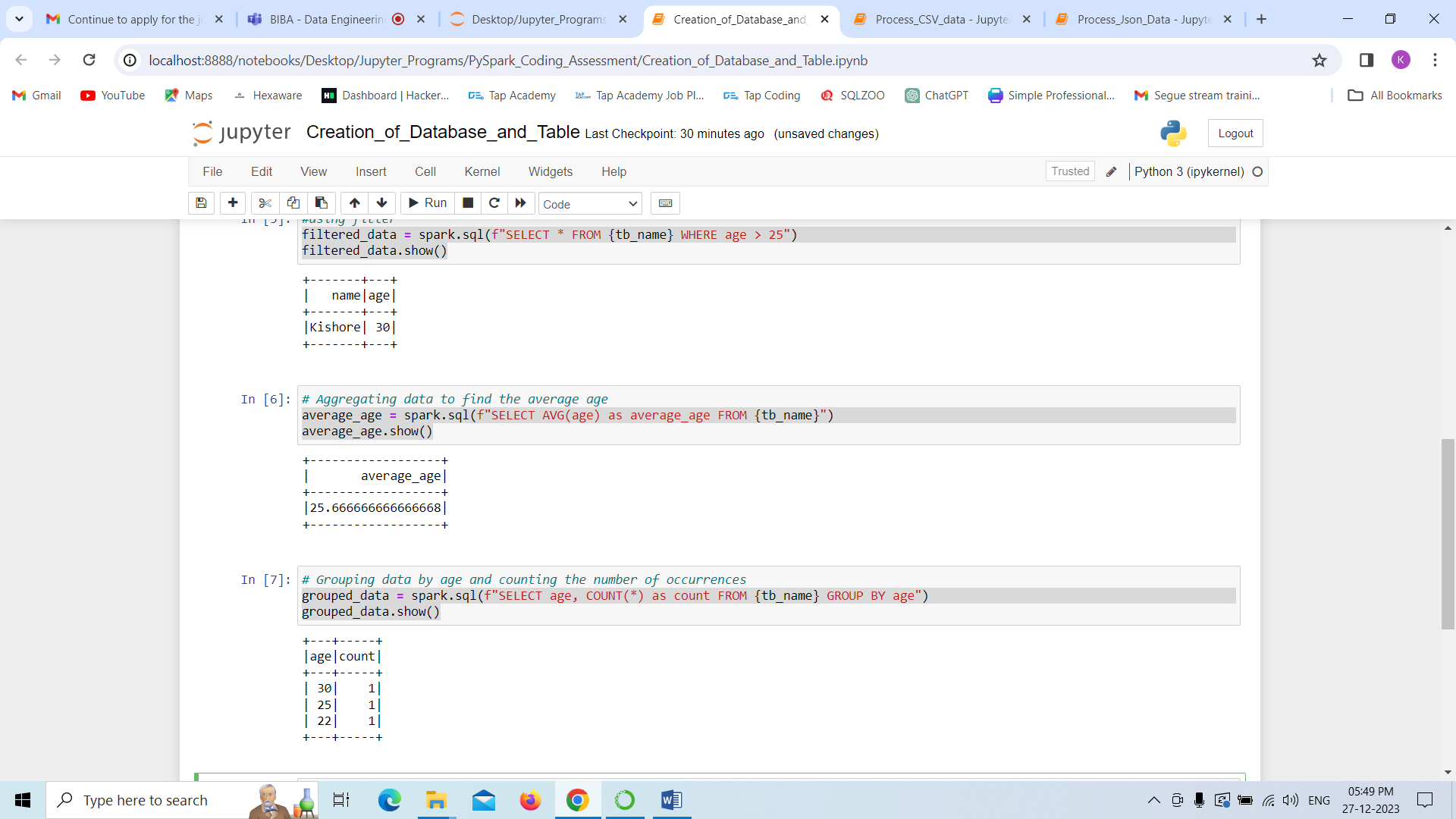
* The avg function is used for calculating the average (mean) of a numeric column in a DataFrame.
* Here we find the average of all ages in the particular table.



**Perfoming groupby**

**GROUP BY**

* The GROUP BY operation is used to group the rows of a DataFrame based on one or more columns. It is a transformation operation that creates a grouped DataFrame.
* This SQL query is selecting two columns, "age" and the count of records for each unique age value (aliased as "count"). The data is obtained from a table with the name specified by the Python variable {tb\_name}.
* The GROUP BY clause groups the data based on the "age" column.



**Perfoming Join Operations by creating one more table that is course table.**

**Joins:**

* Basically if we want to combine the data and perform operations we need Joins.
* Here,performs a JOIN operation between two tables ({tb\_name} and course\_data) based on the "name" column, creating a new DataFrame (joined\_data) that contains information about students, their ages, and the courses they are enrolled.

