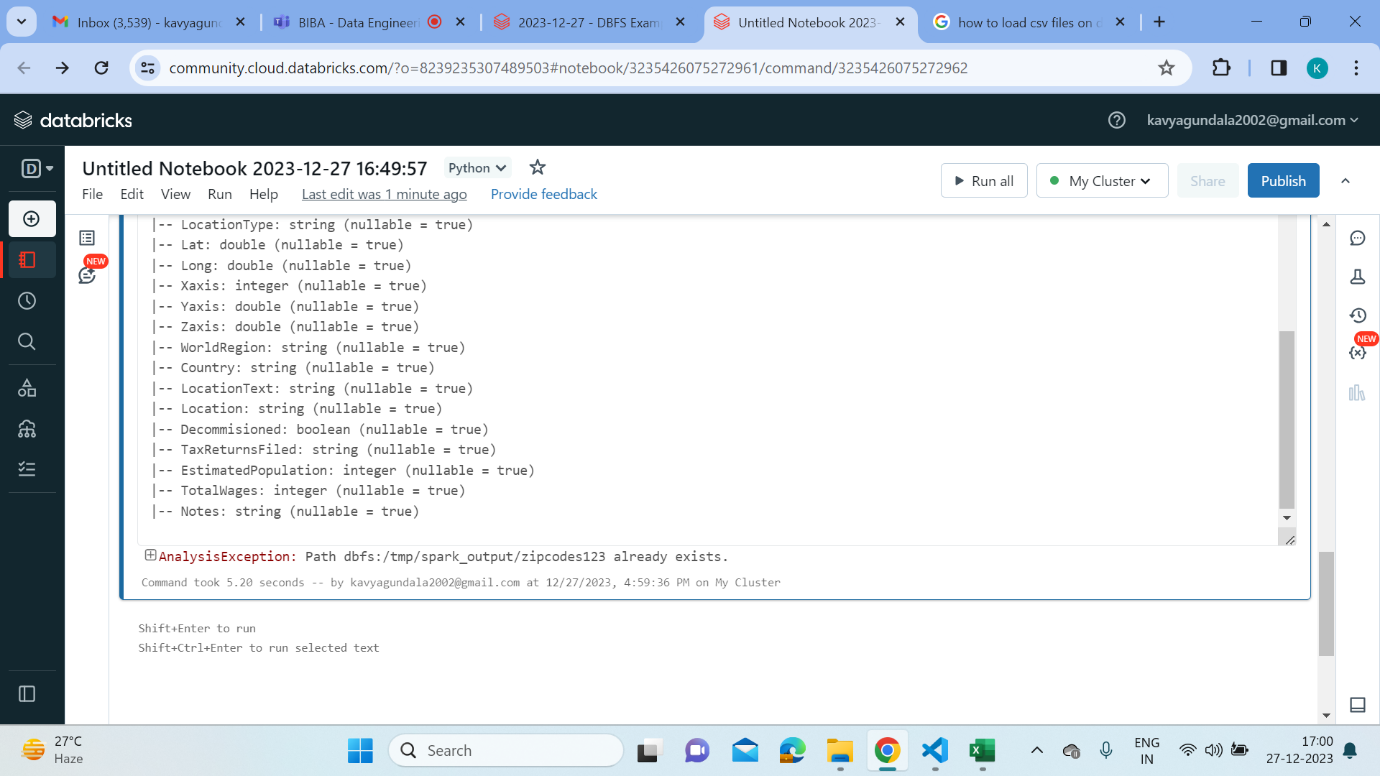
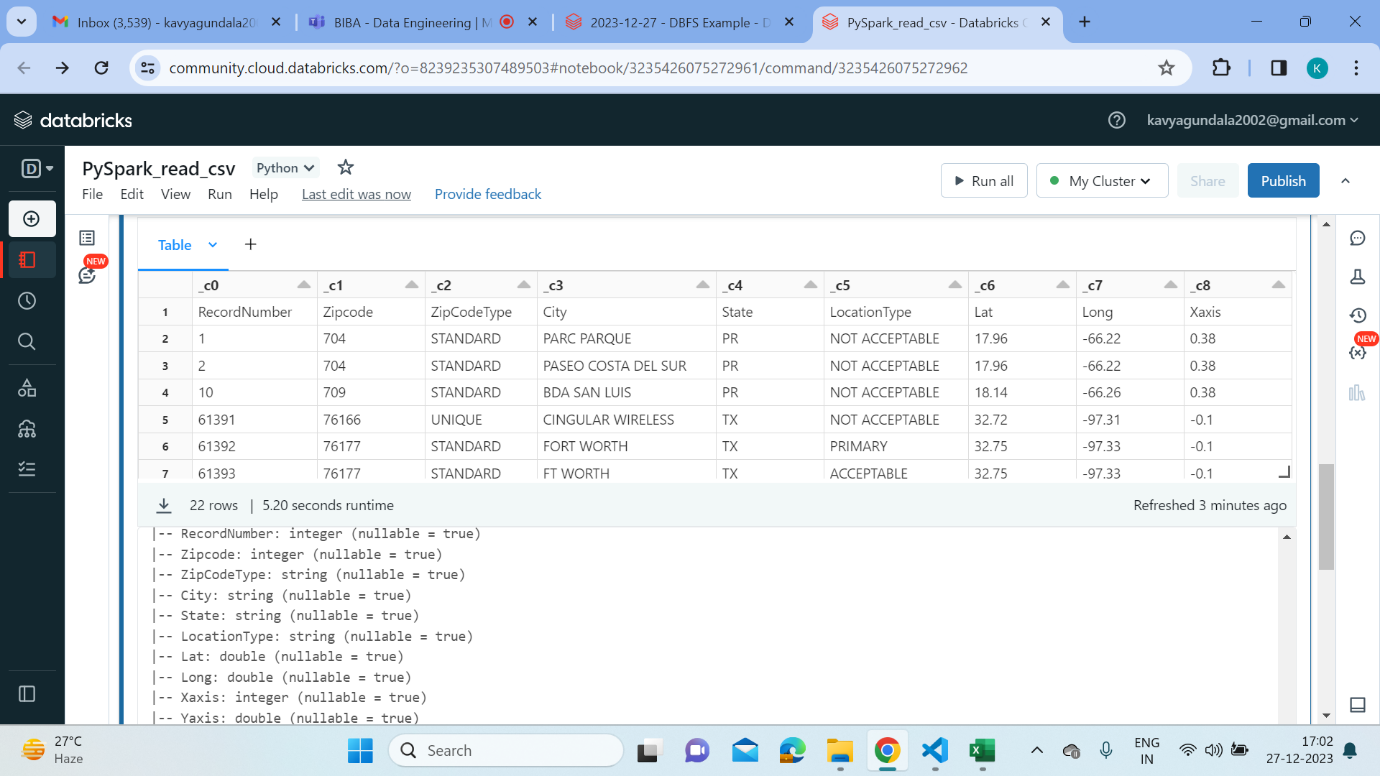
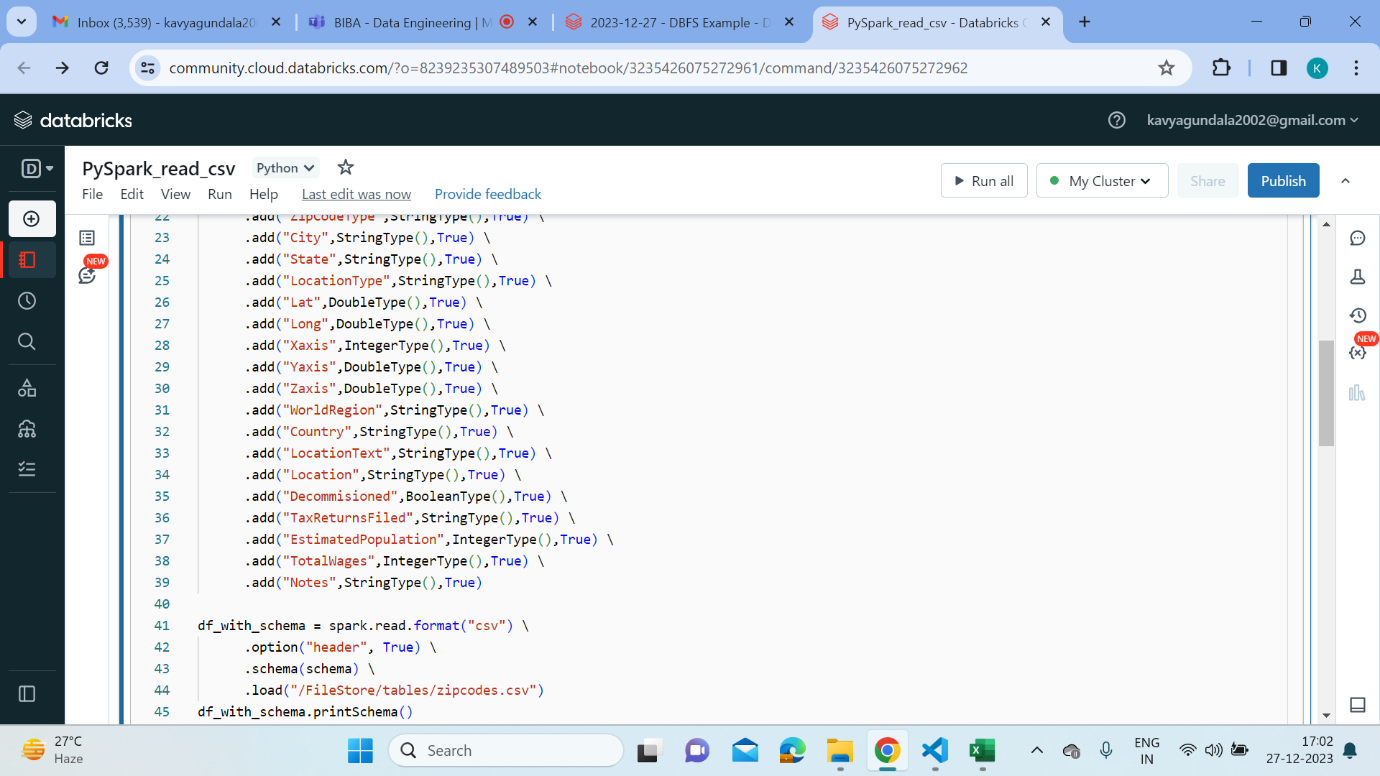
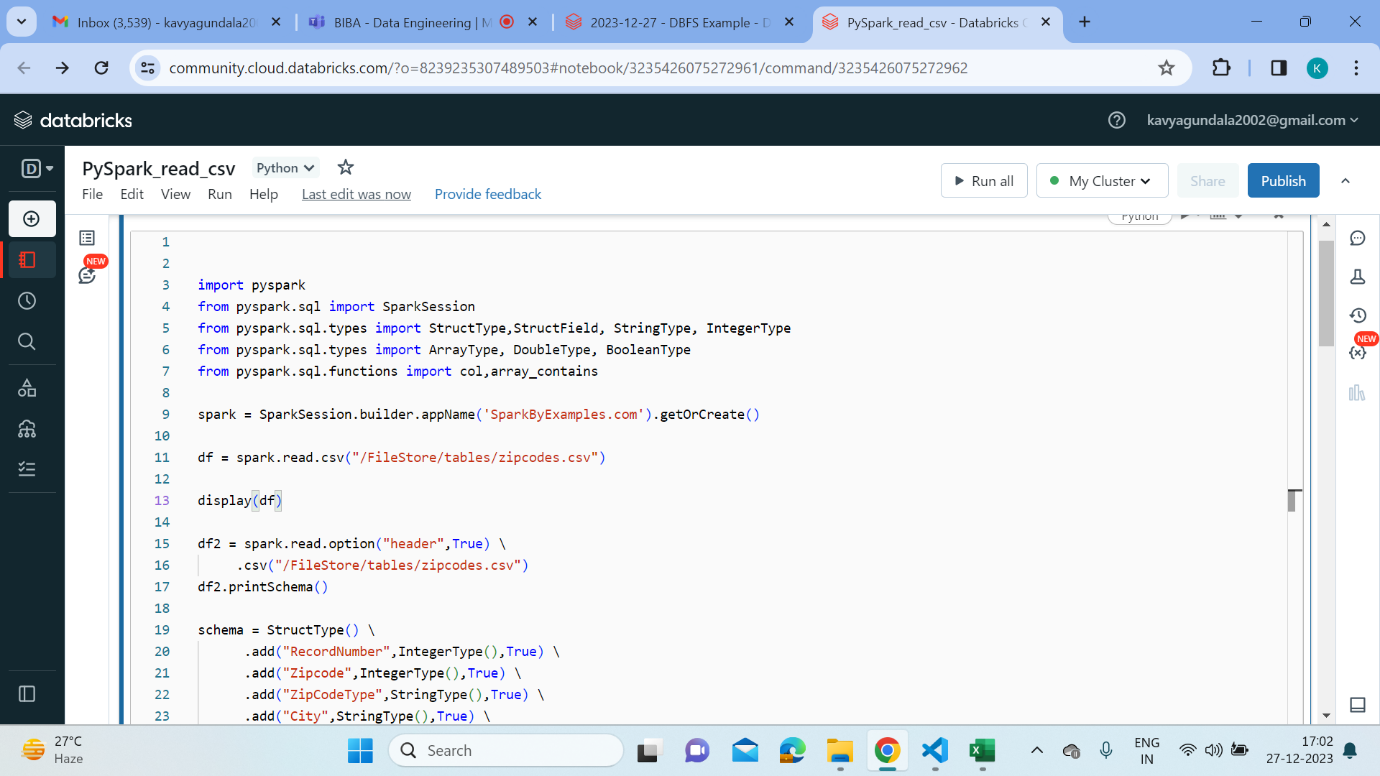
Kavya Gundala

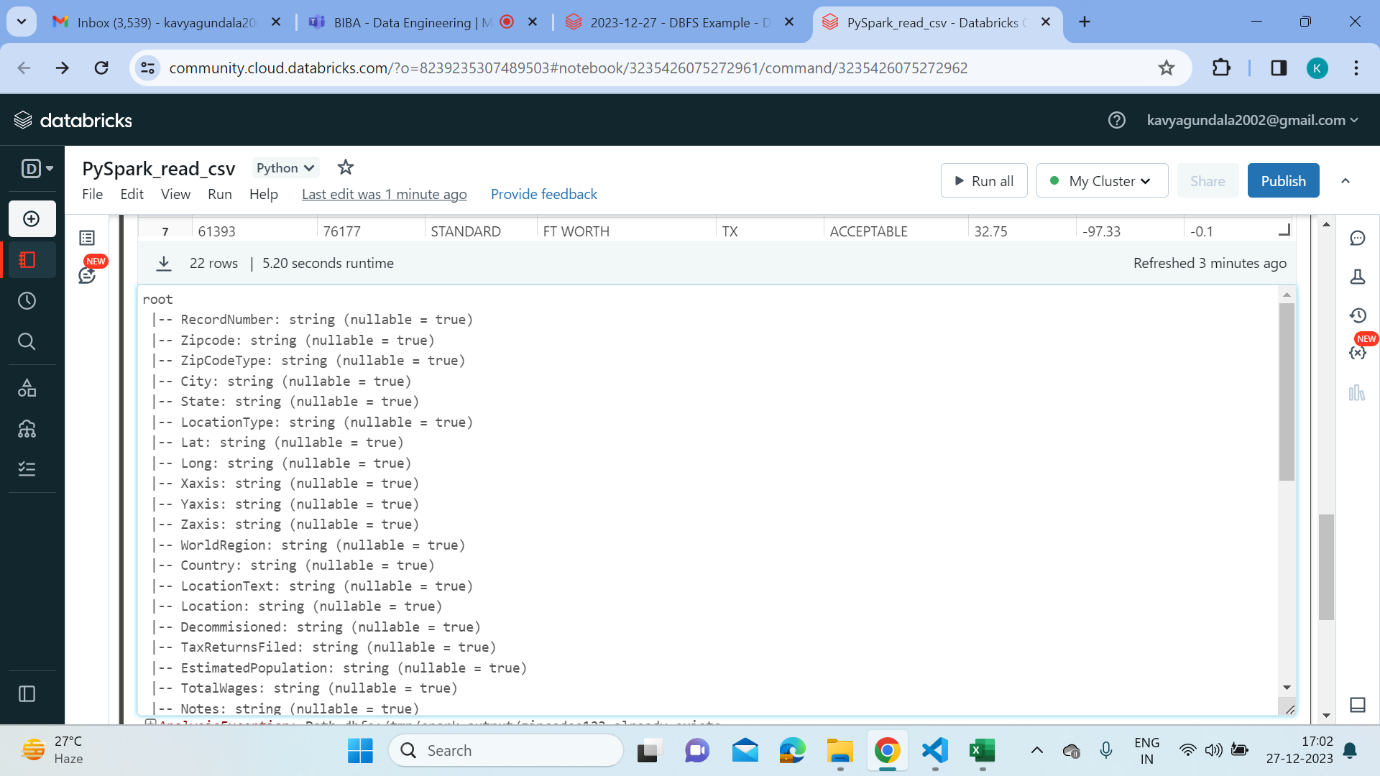
**27-12-2023**

1. **Implement Processing JSON and CSV data with PySpark**

PySpark implementation with csv files in Databricks:

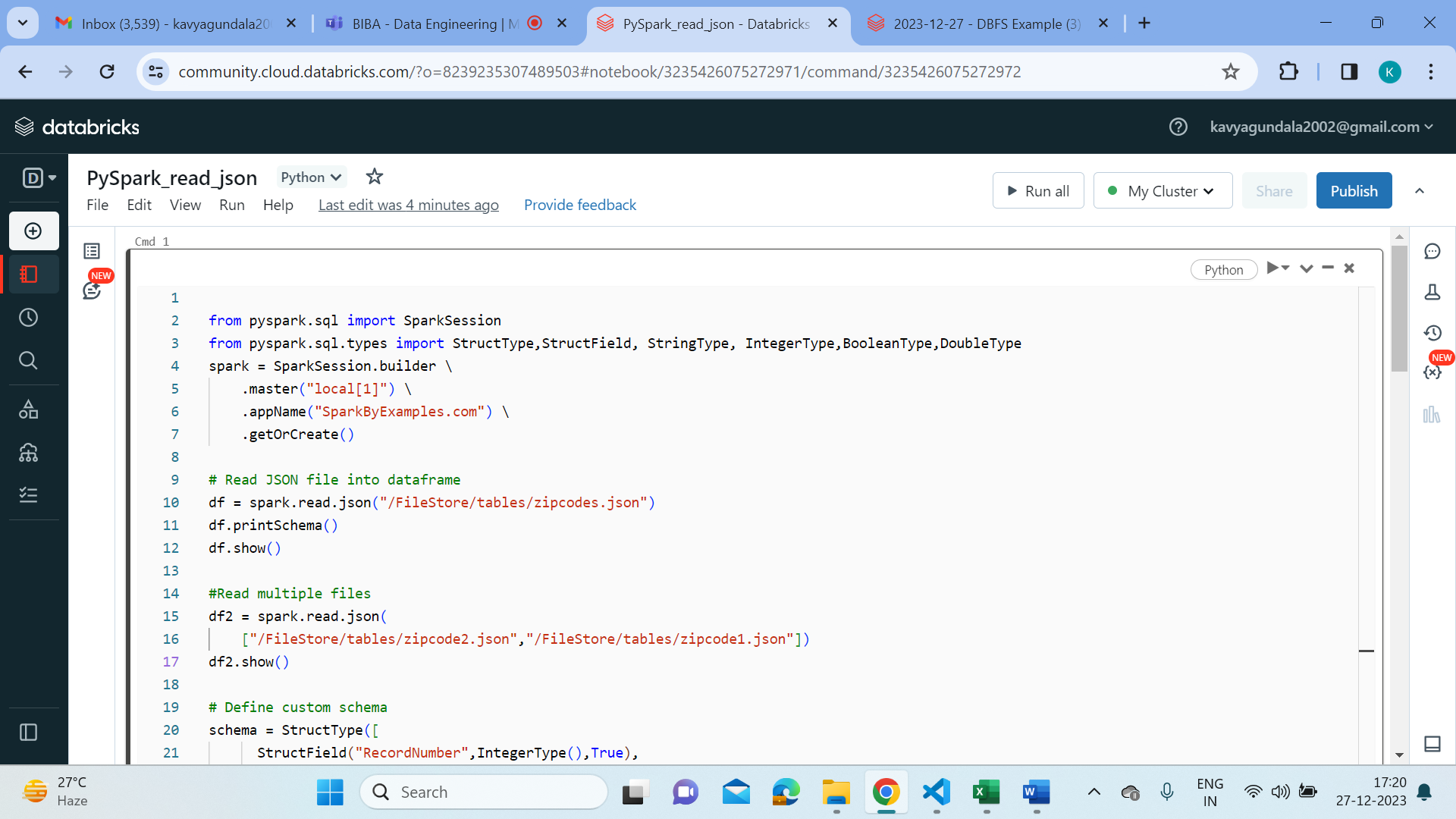
* Import all the required packages from pyspark
* Initialize SparkSession
* Upload csv files into databricks and read the csv file in PySpark
* Display the csv table and its schema using below commands.

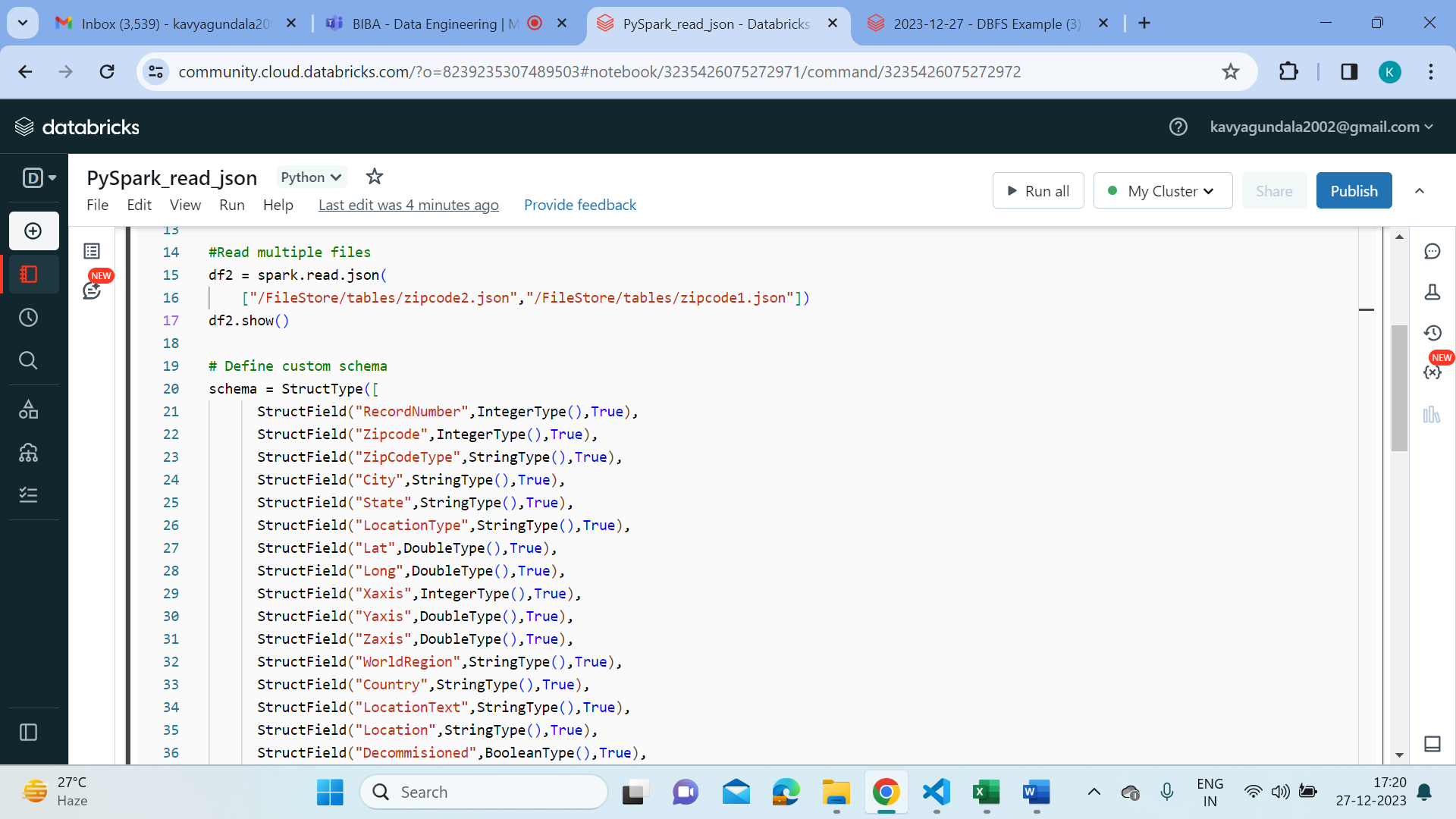


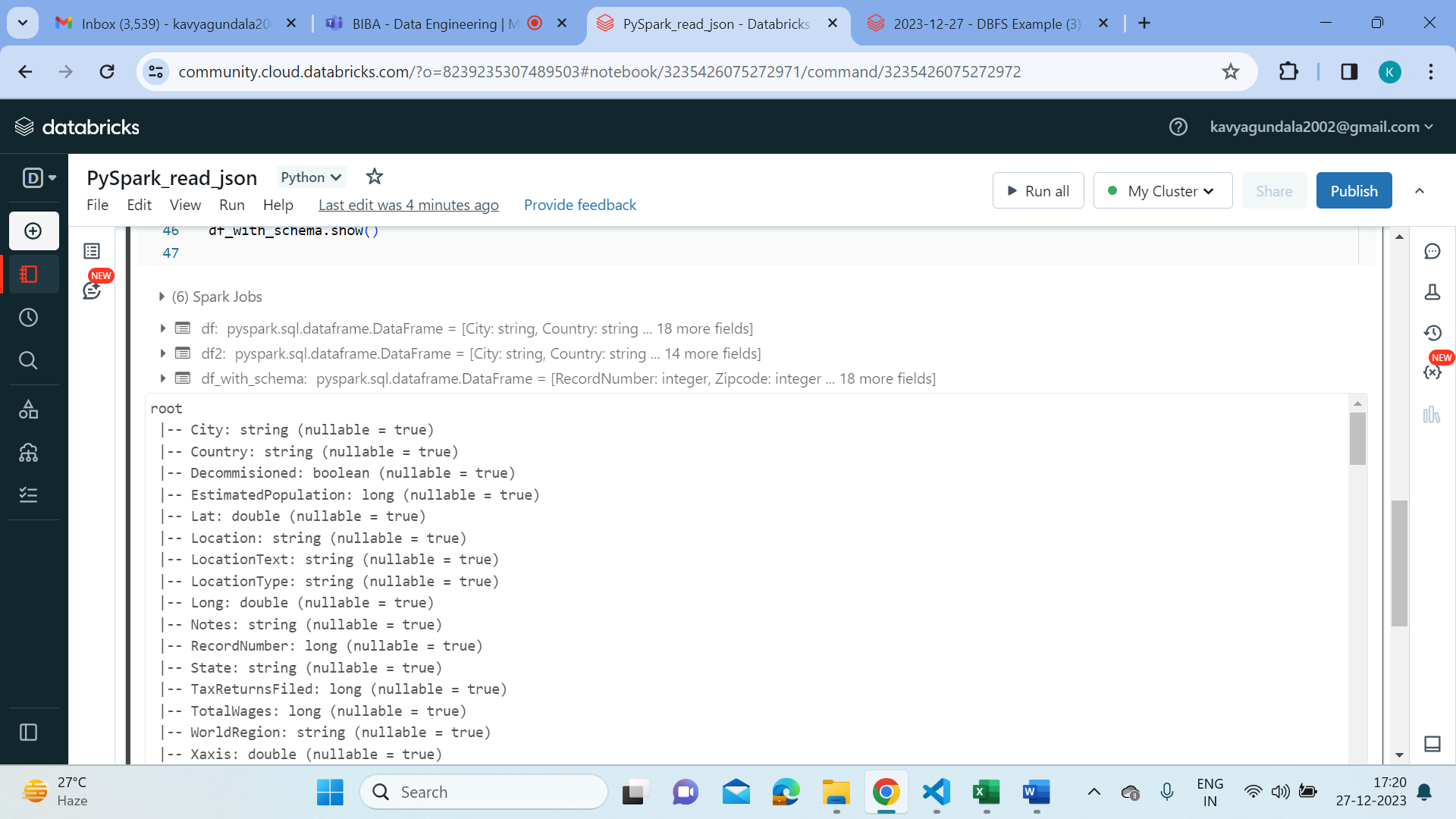


PySpark implementation with json files in Databricks:

* Import all the required packages from pyspark
* Initialize SparkSession
* Upload json files into databricks and read the json file in PySpark
* Display the json table and its schema using below commands.







1. **Explain ETL with PySpark?**

* PySpark, the Python API for Apache Spark, has revolutionized the way we handle

Big Data. It’s an ETL powerhouse that combines the simplicity of Python with the

scalability and performance of Spark.

ETL with PySpark is Advantageous for:

* Performance: PySpark leverages in-memory computing, making ETL processes faster than ever.
* Ease of Use: Python developers can seamlessly transition to PySpark due to its

Pythonic syntax.

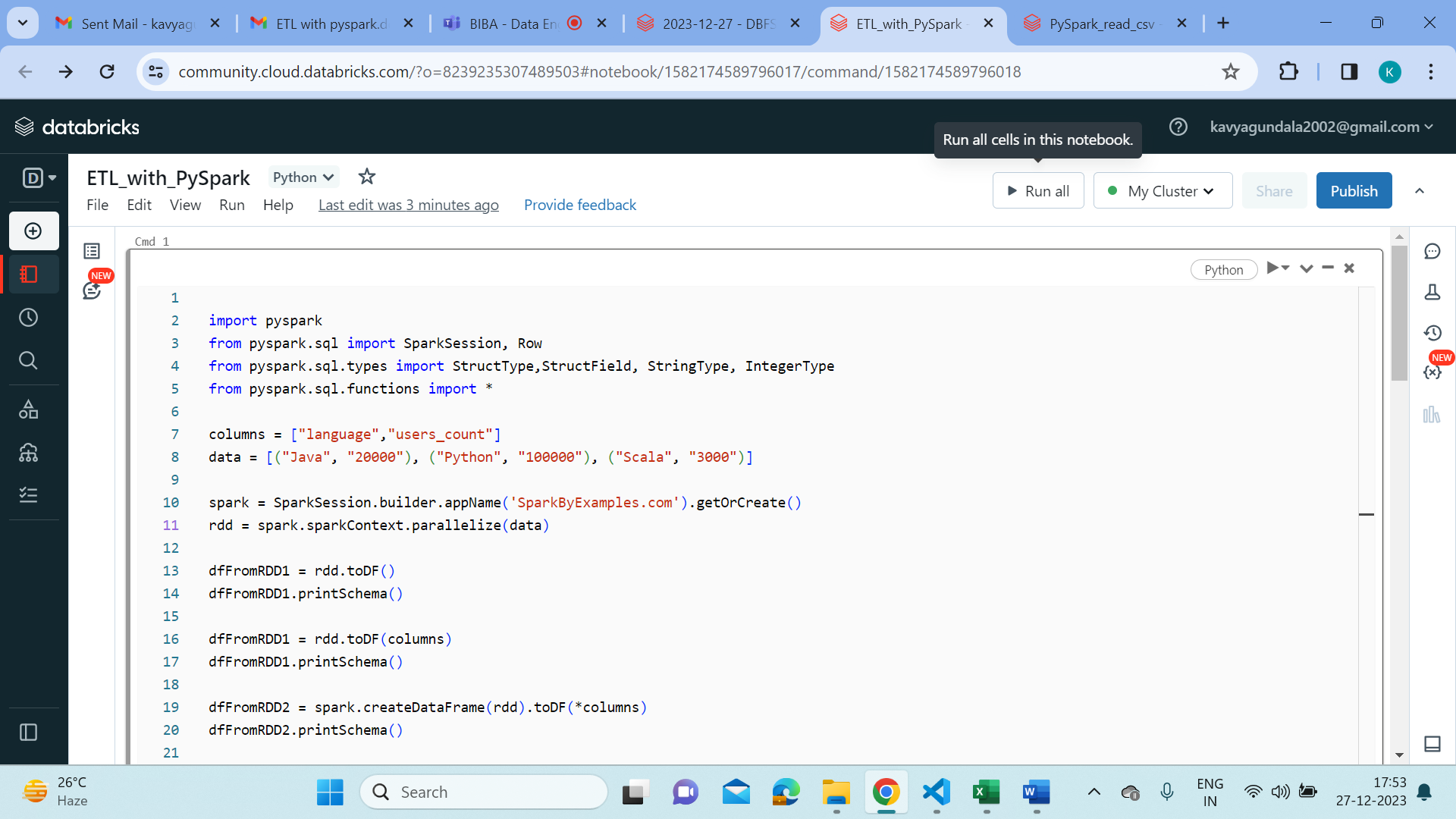
* Scalability: Handle massive datasets with ease, thanks to Spark’s distributed processing.
* Rich Ecosystem: PySpark integrates with popular tools and libraries, making it

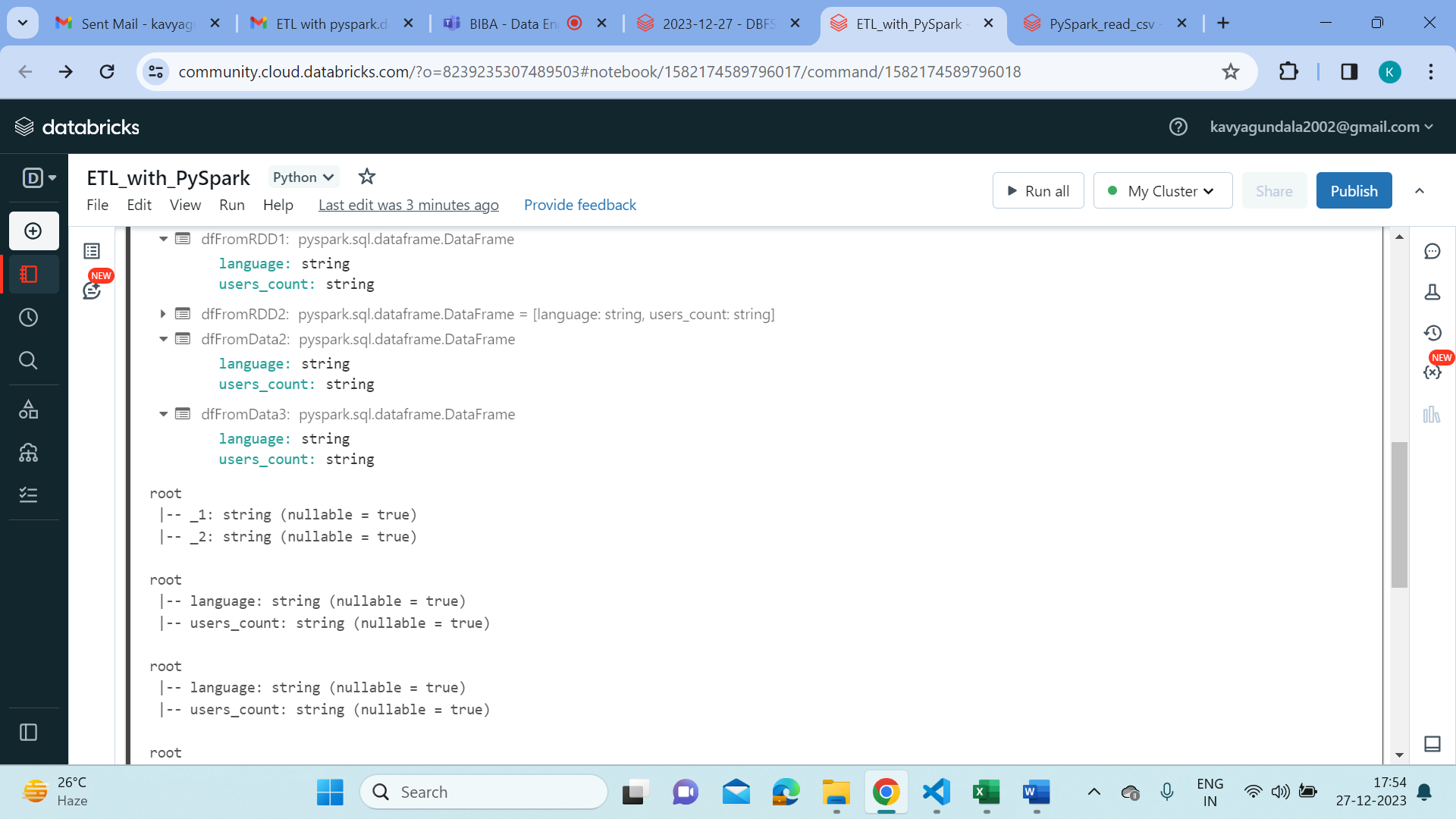
versatile for various data tasks.

**The PySpark ETL Workflow:**

* Extract: Retrieve data from various sources like databases, files, or APIs.
* Transform: Clean, aggregate, and manipulate data to fit your analysis needs.
* Load: Store the transformed data into a database or data warehouse for analysis.

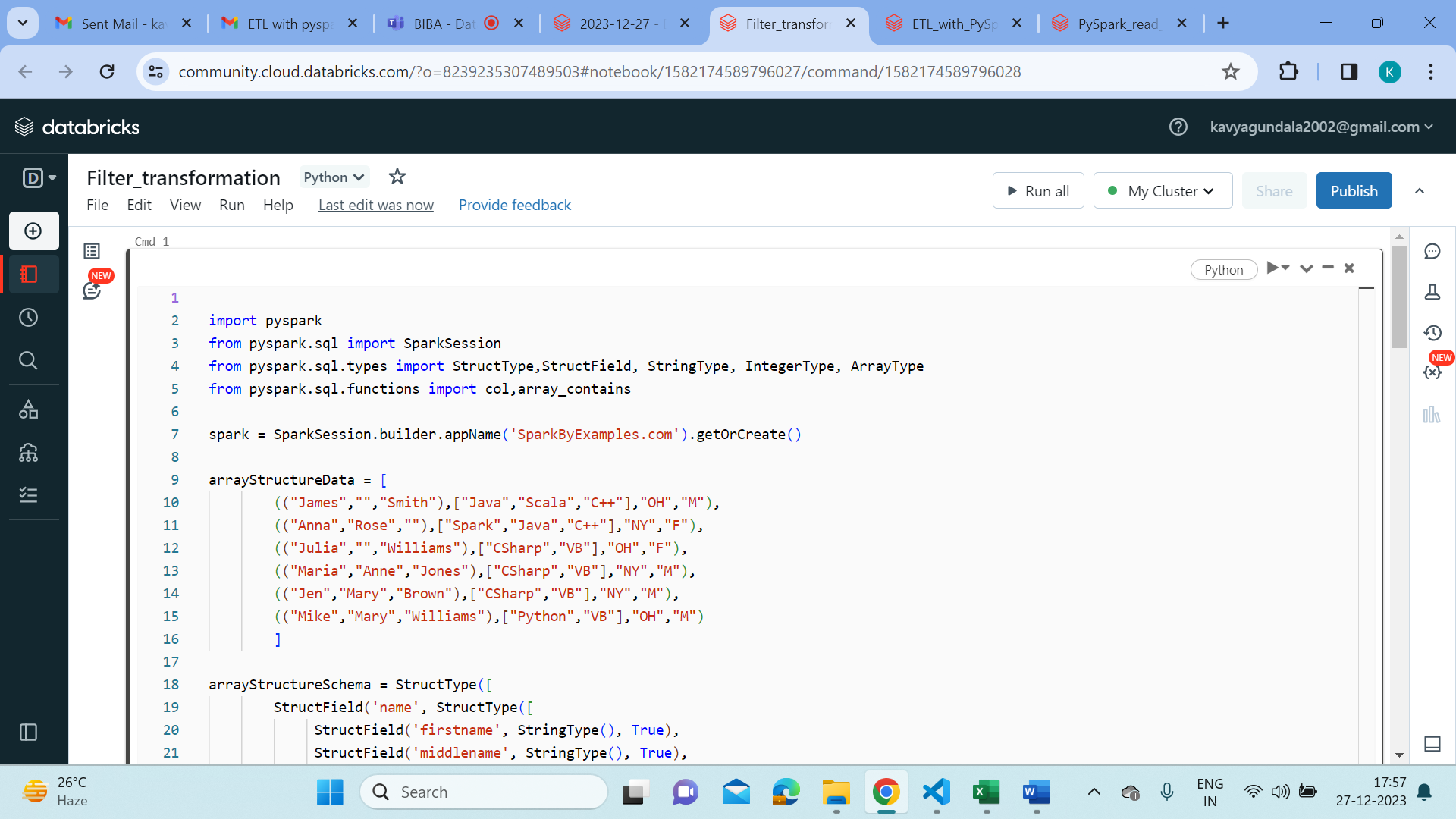
1. **Using Spark SQL - Creating databases, tables**

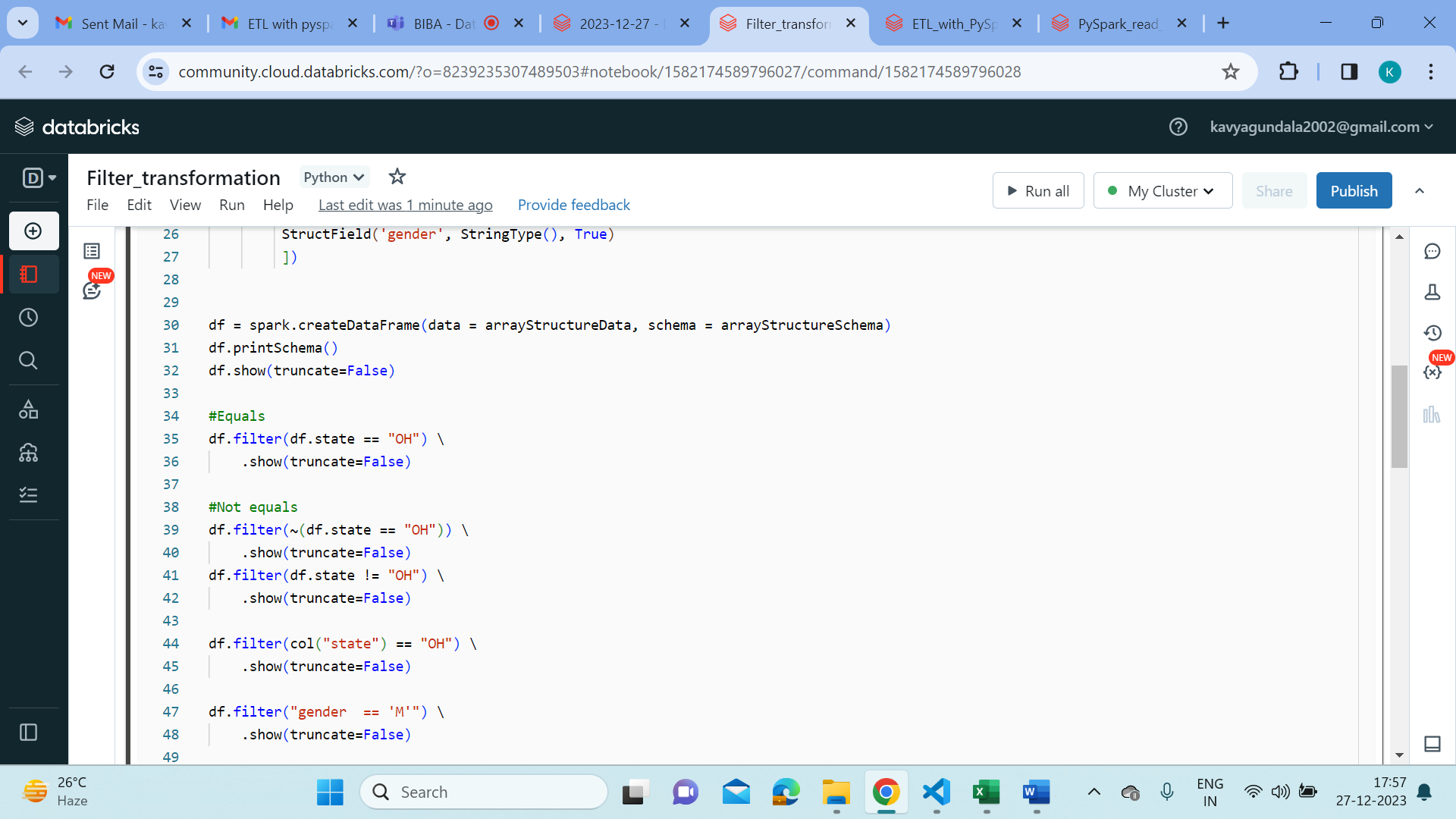


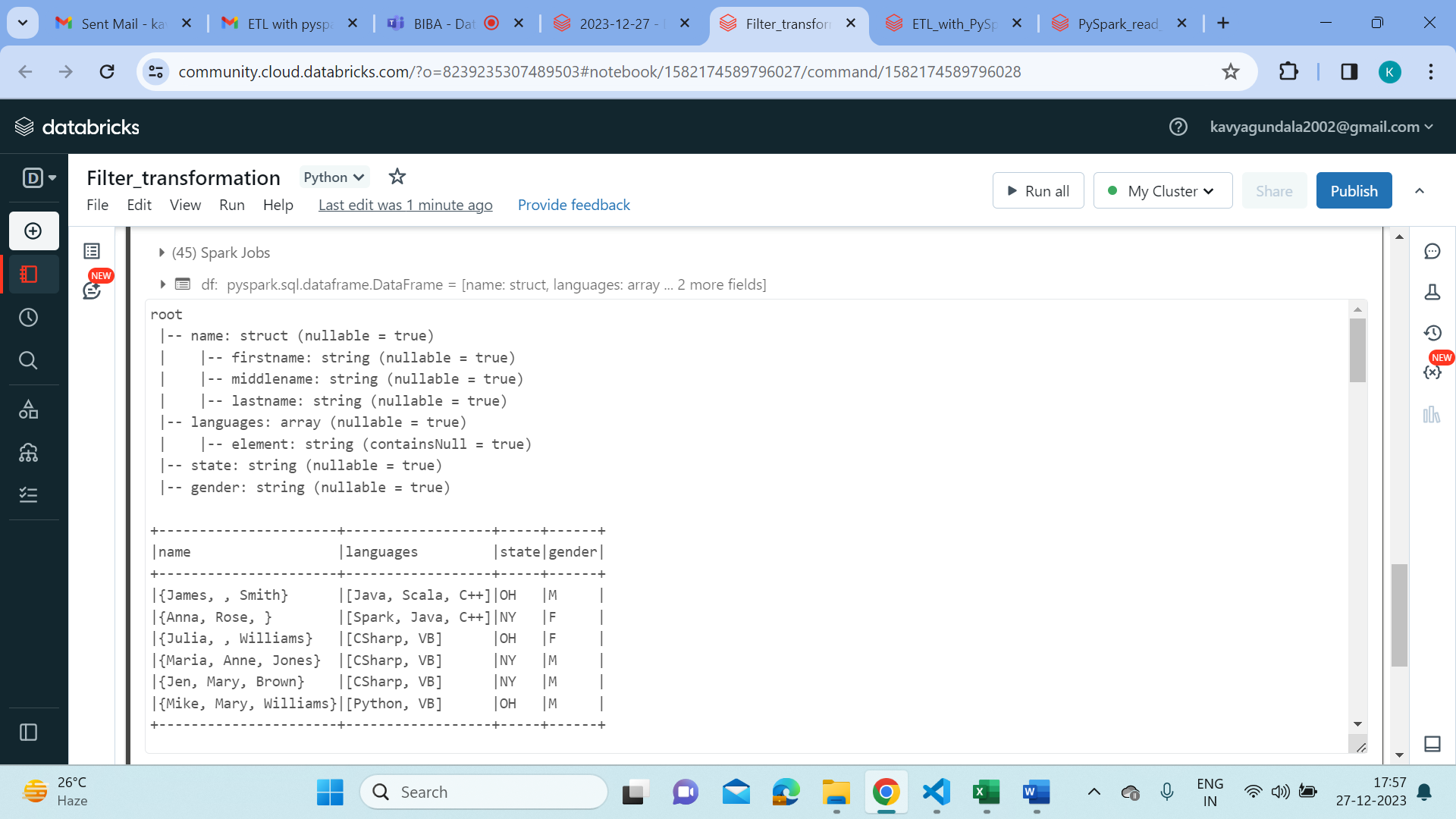


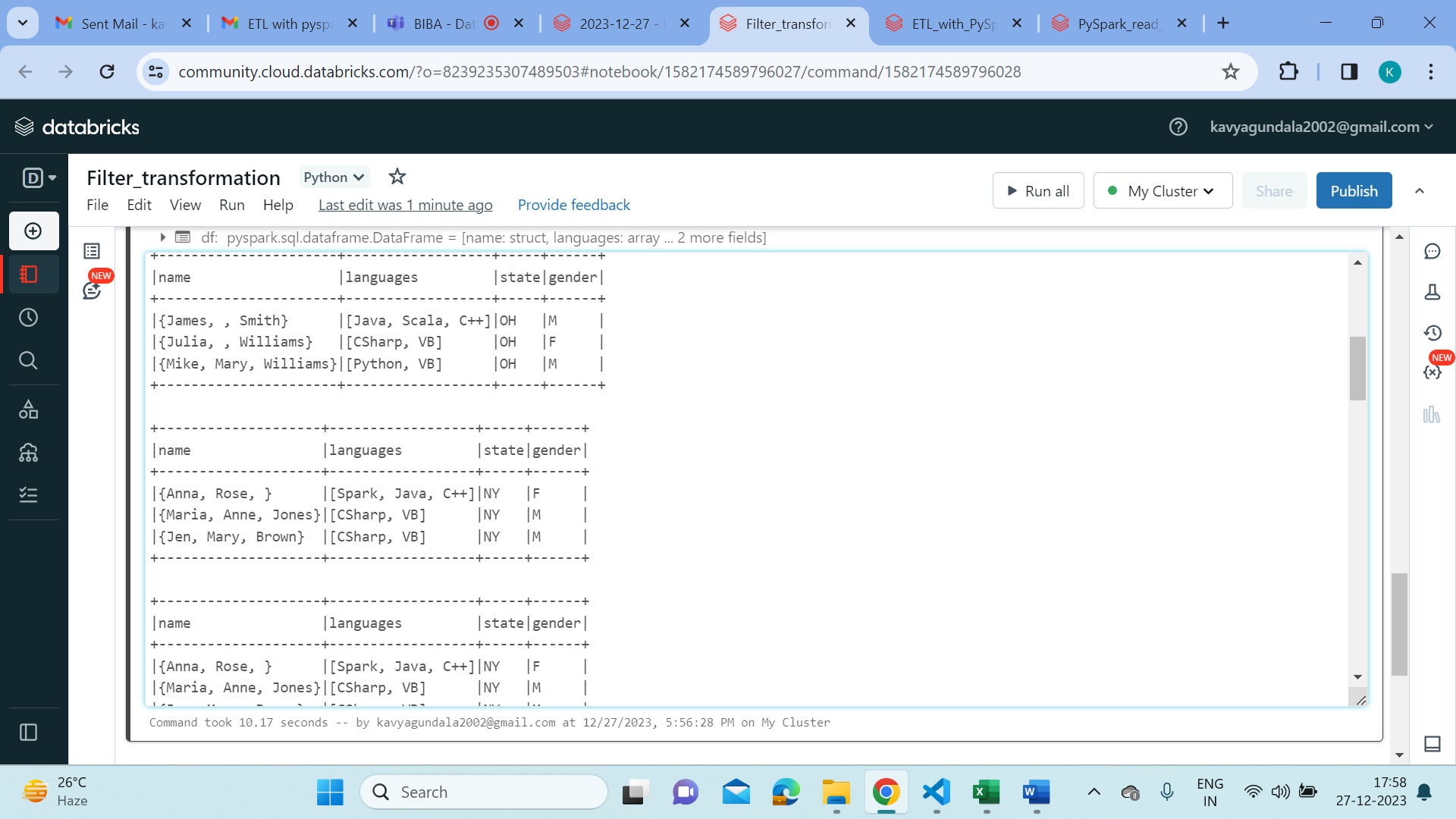
1. **Transformations such as Filter, Join, Simple Aggregations, GroupBy.**

* Filter:

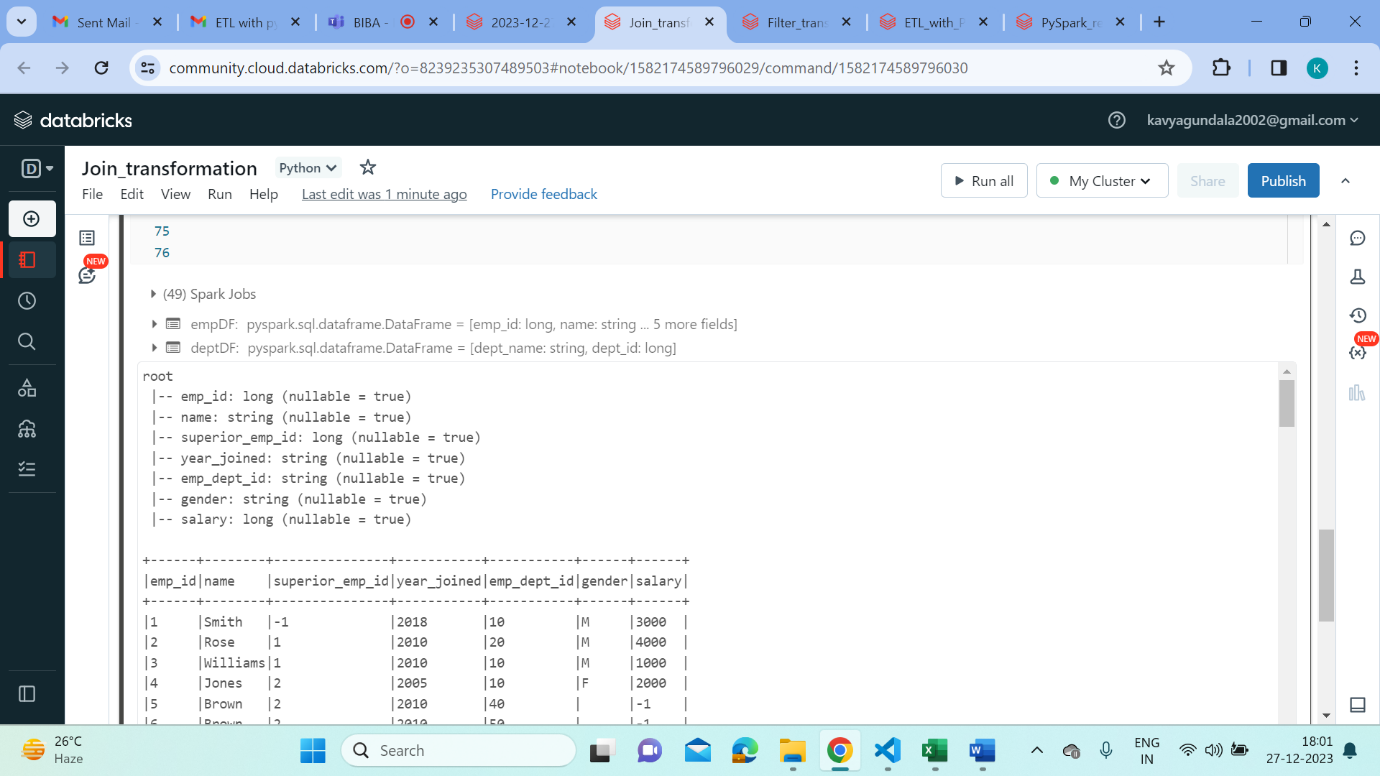
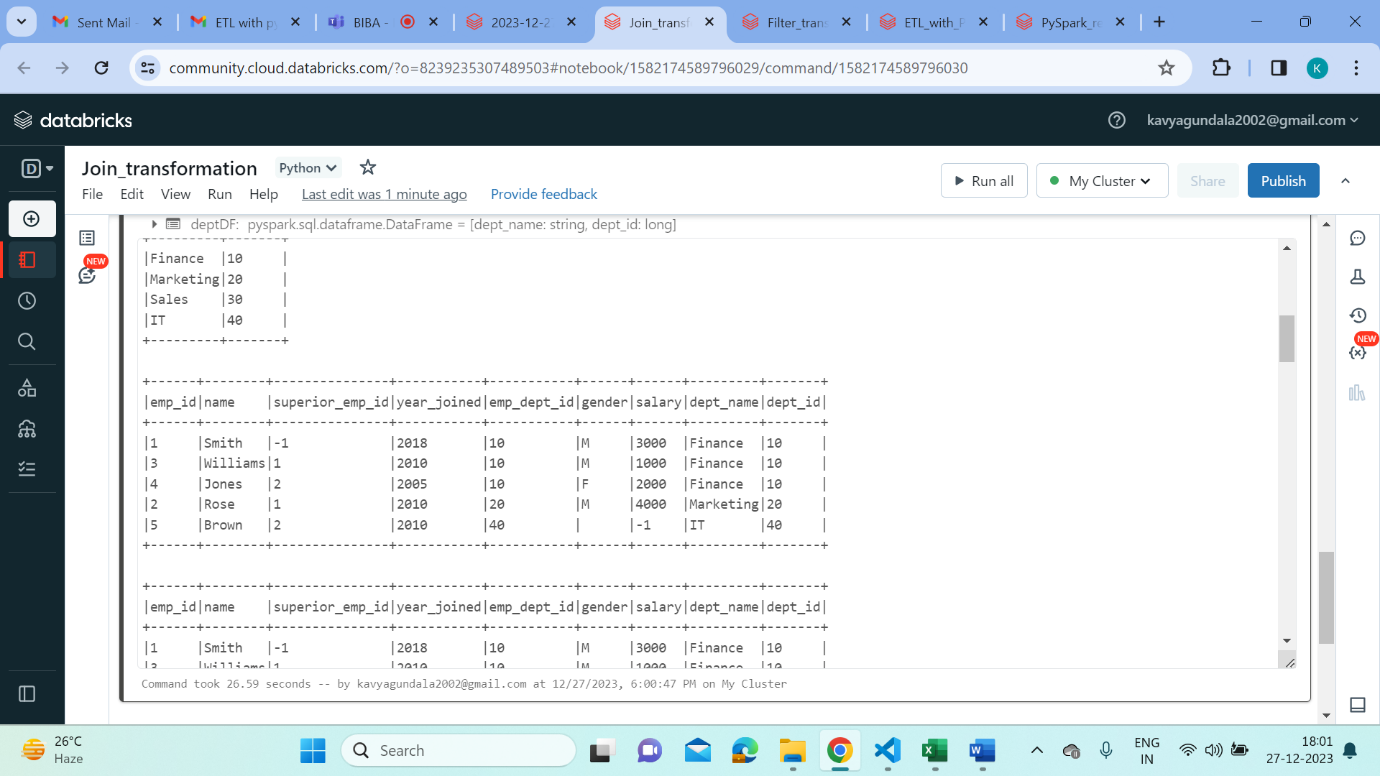
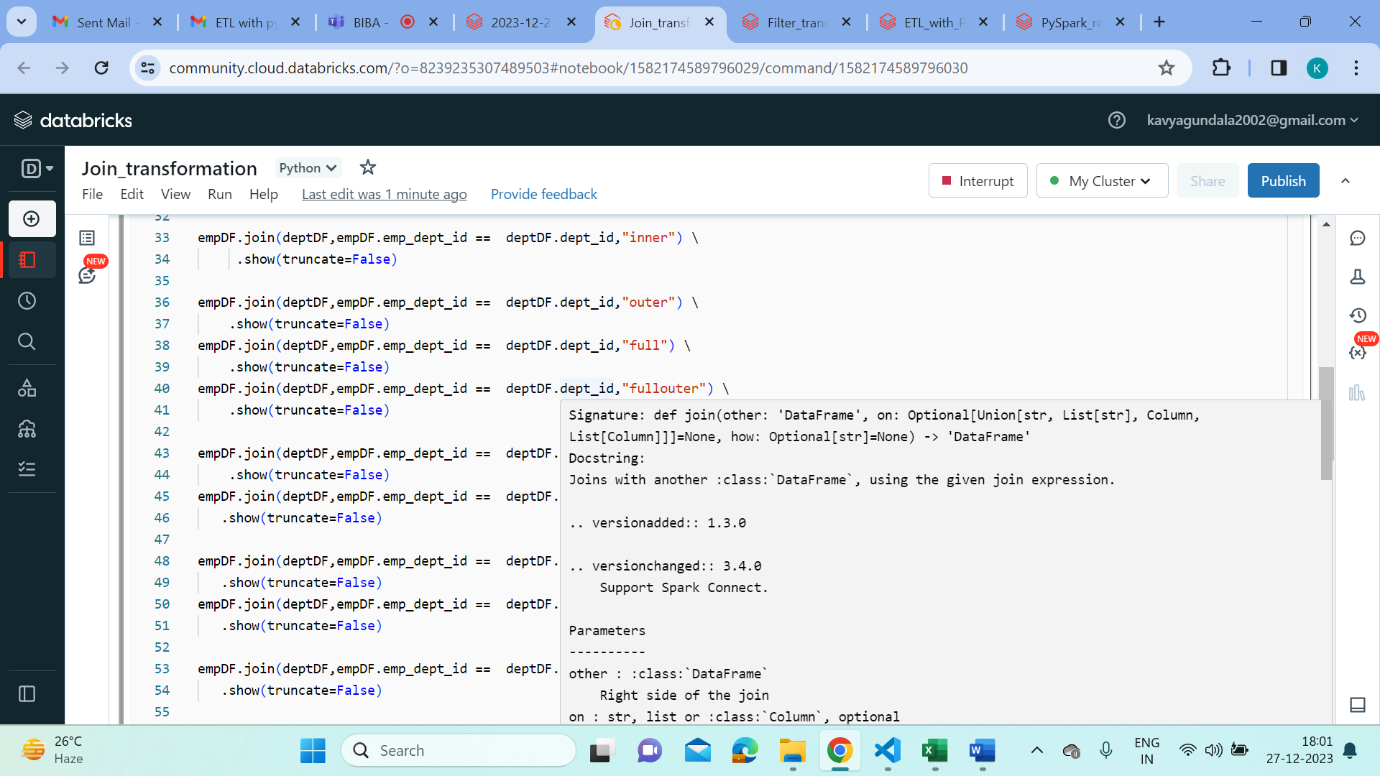
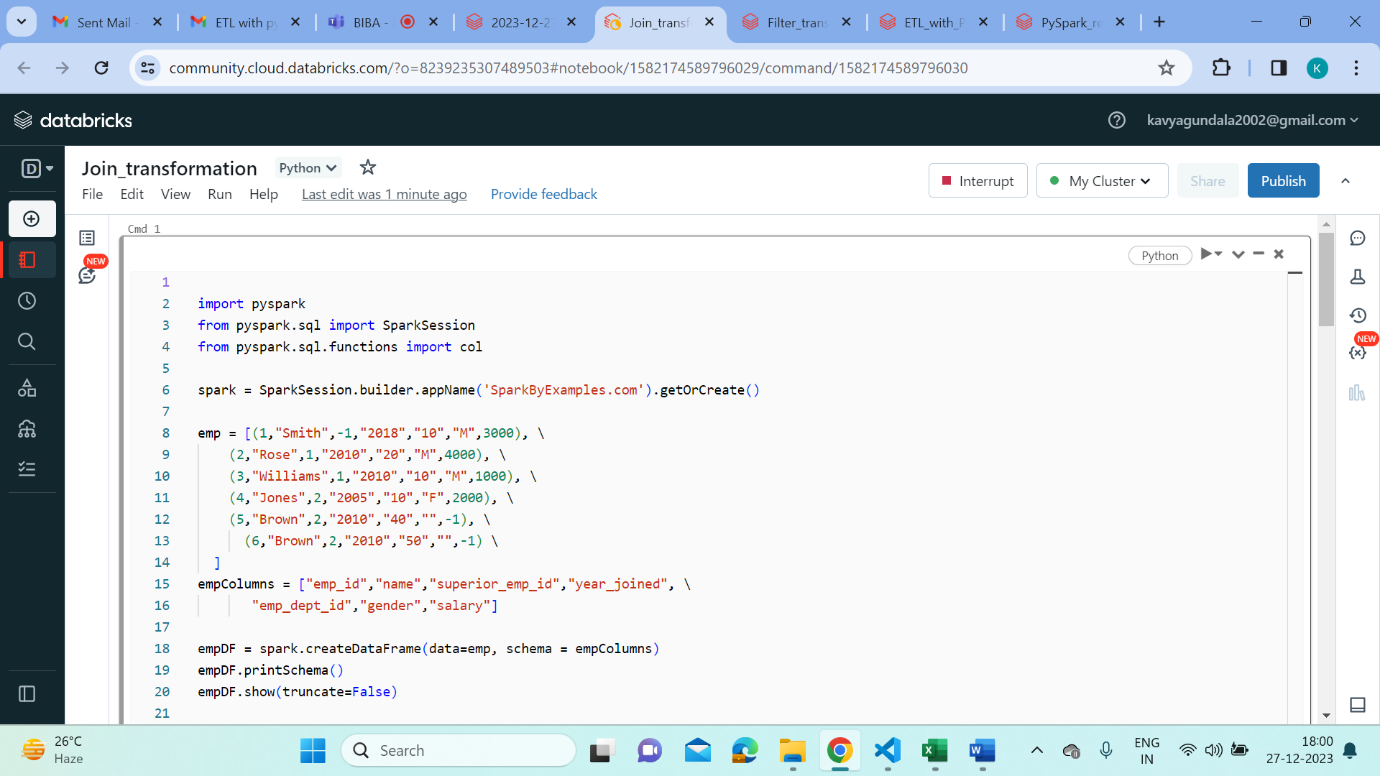




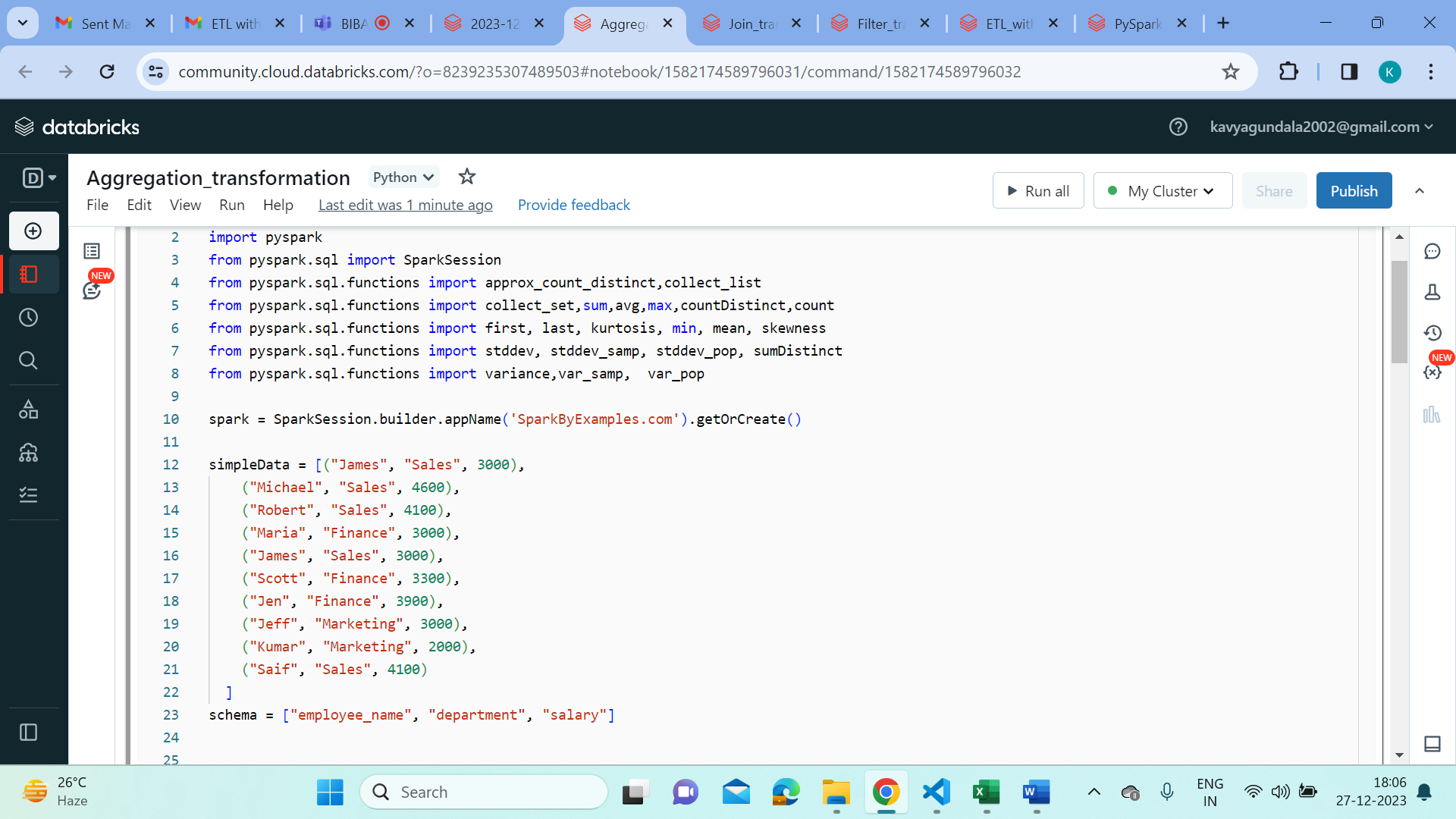


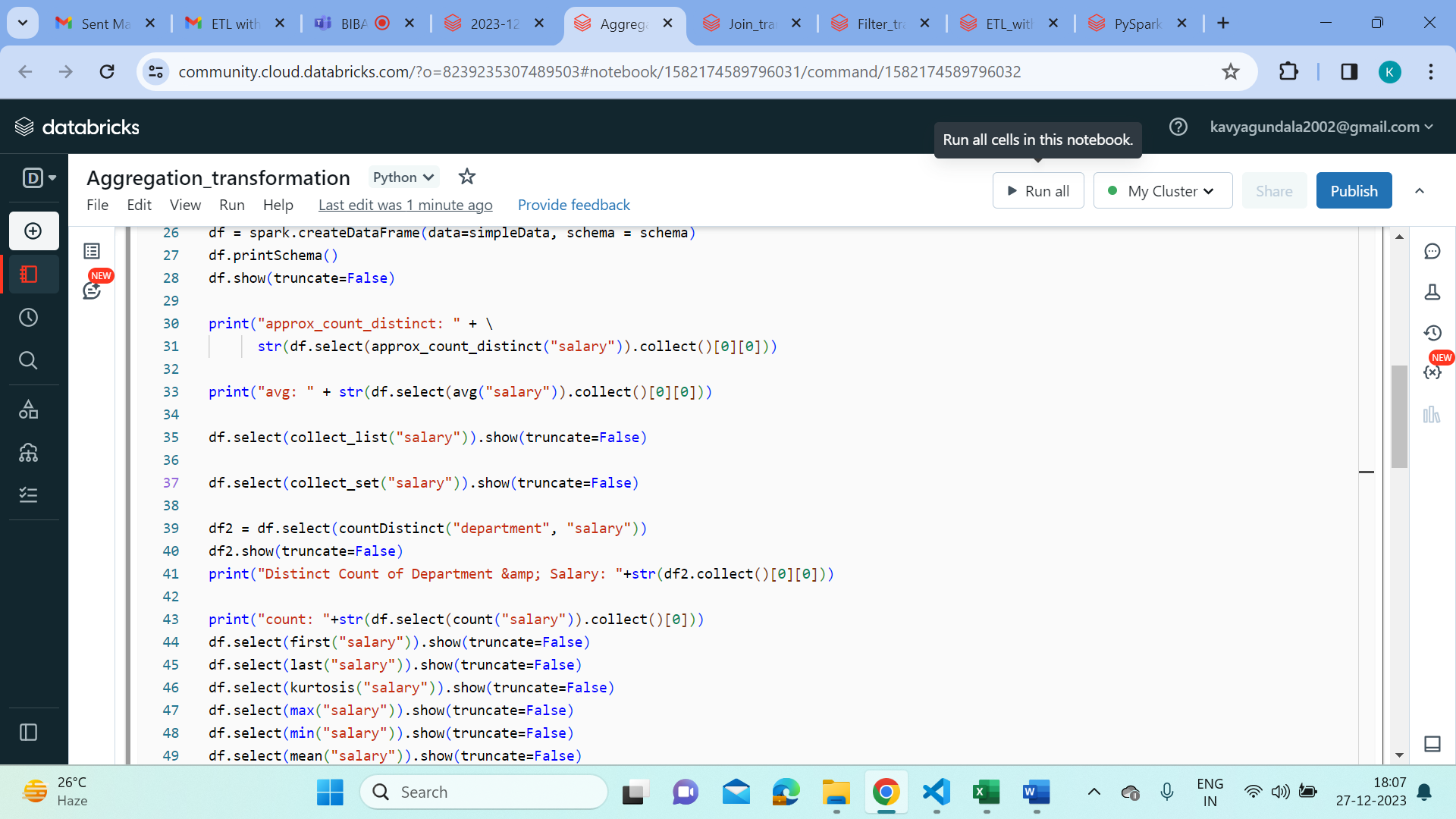


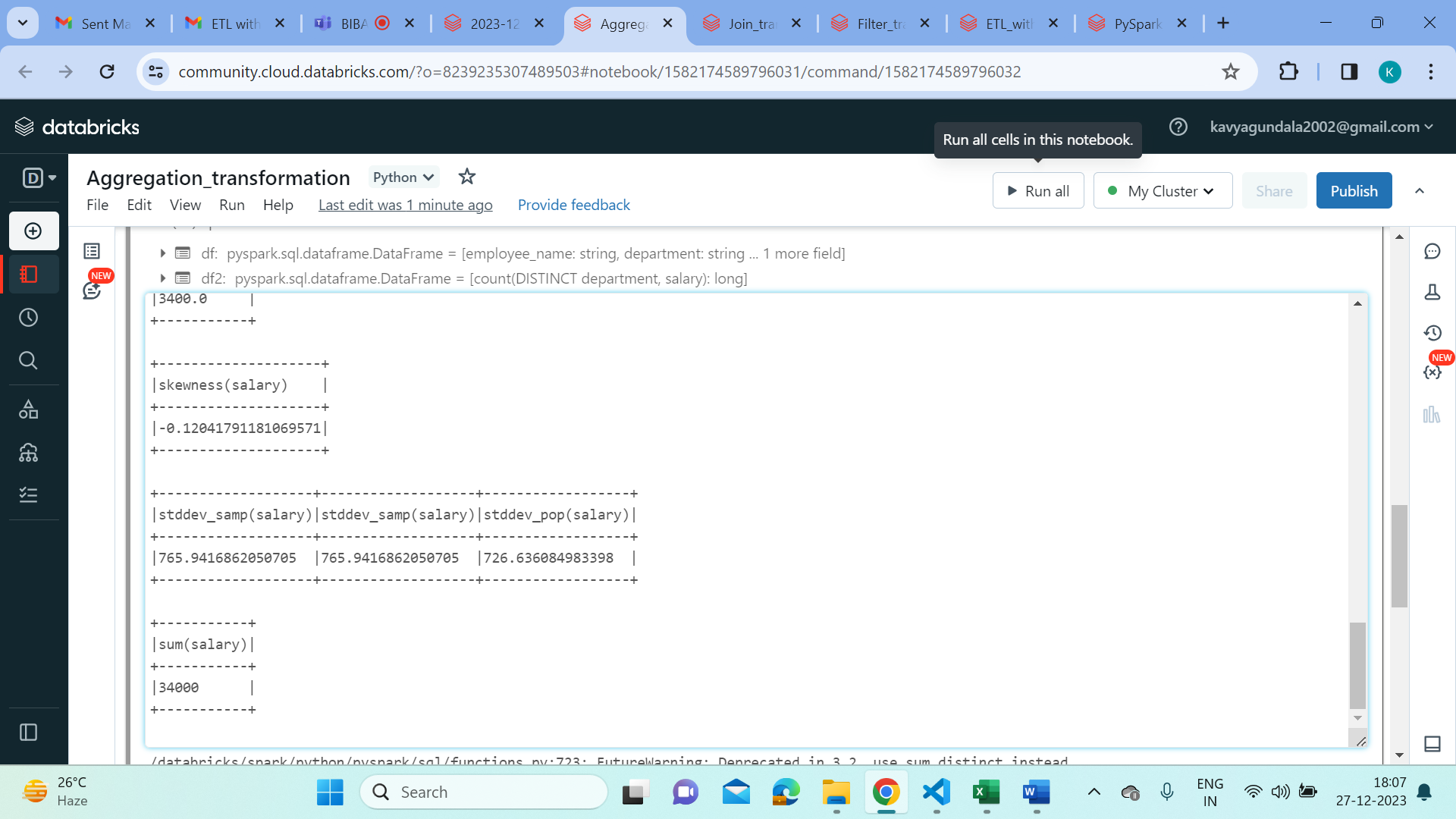
* Join:



* Aggregation:







* GroupBy:

