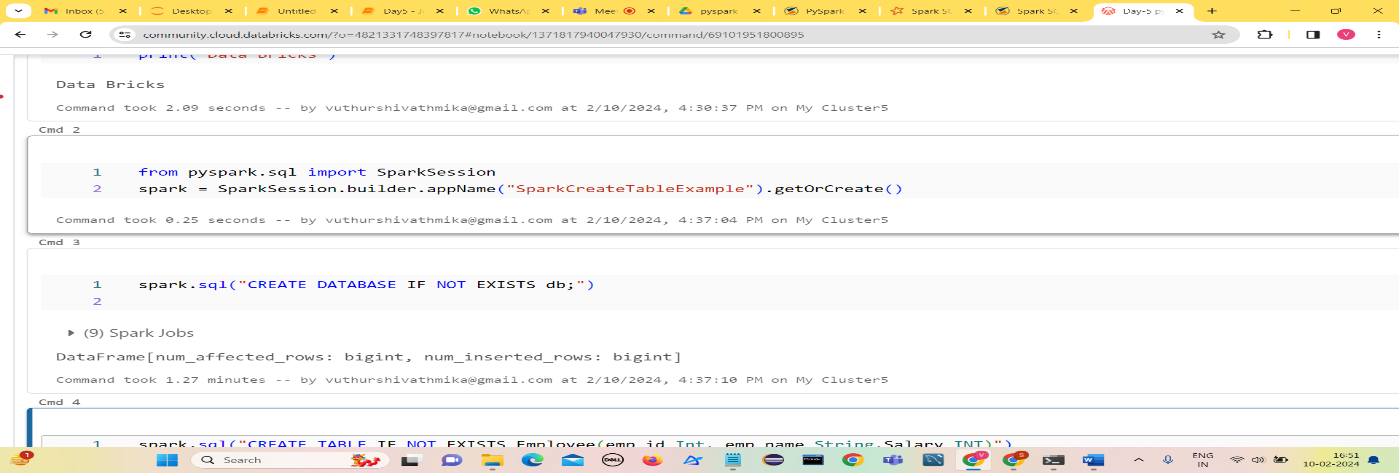
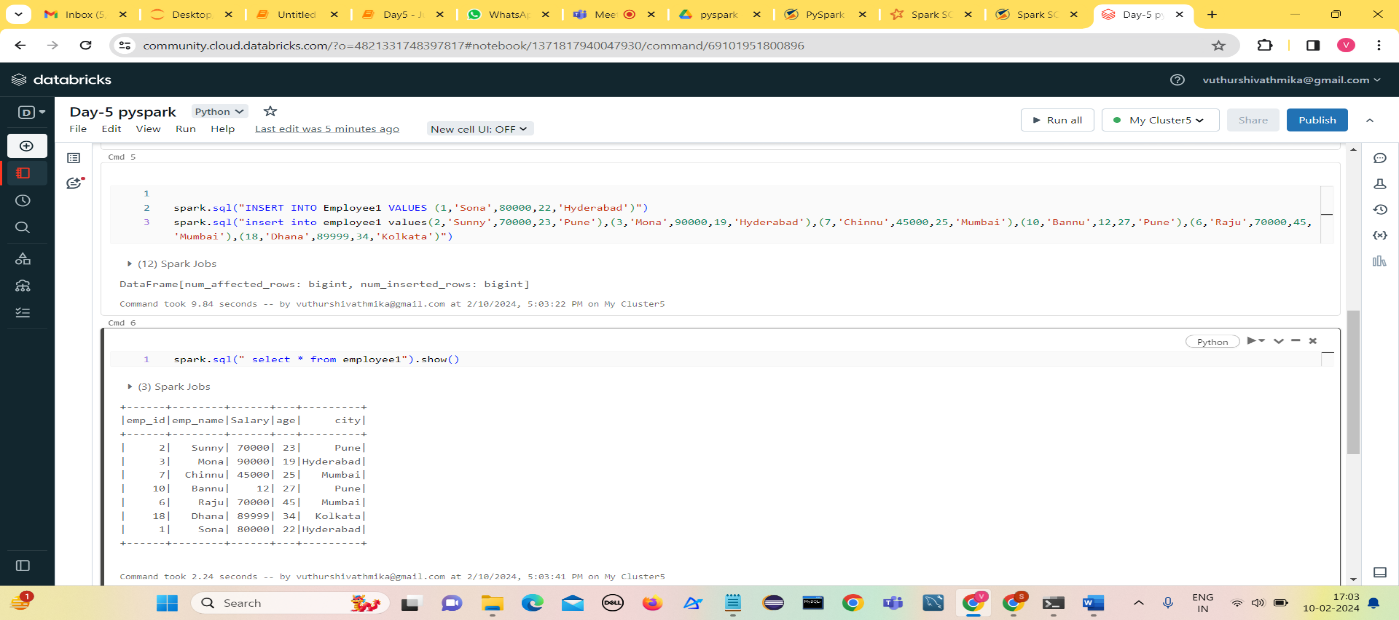
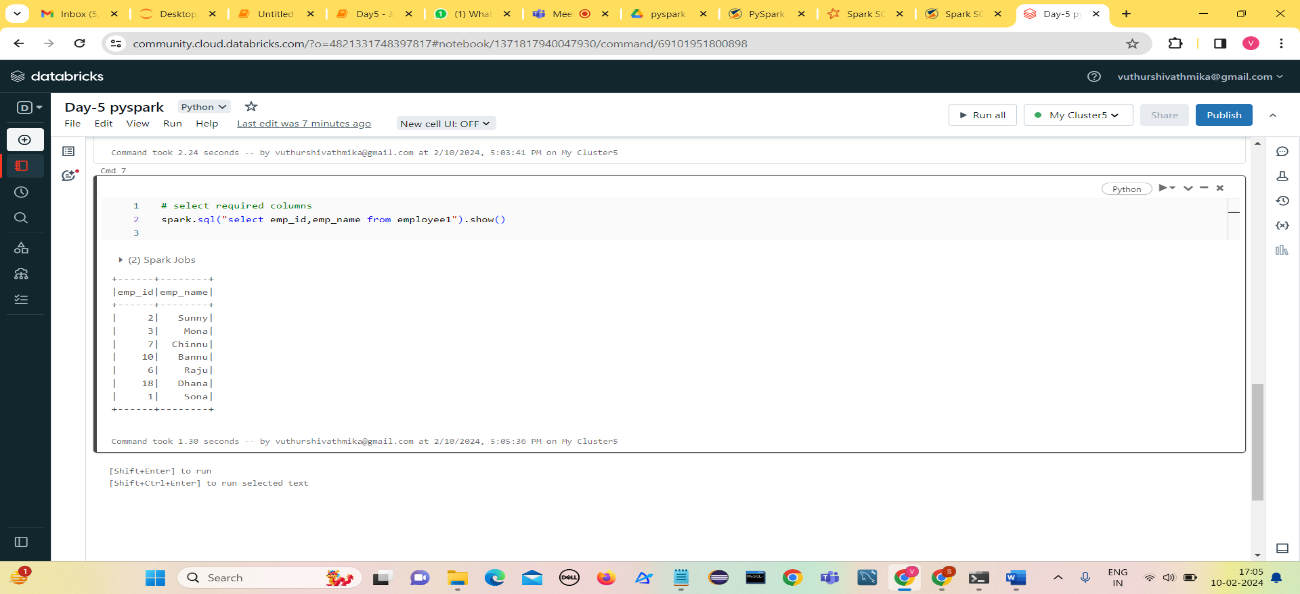
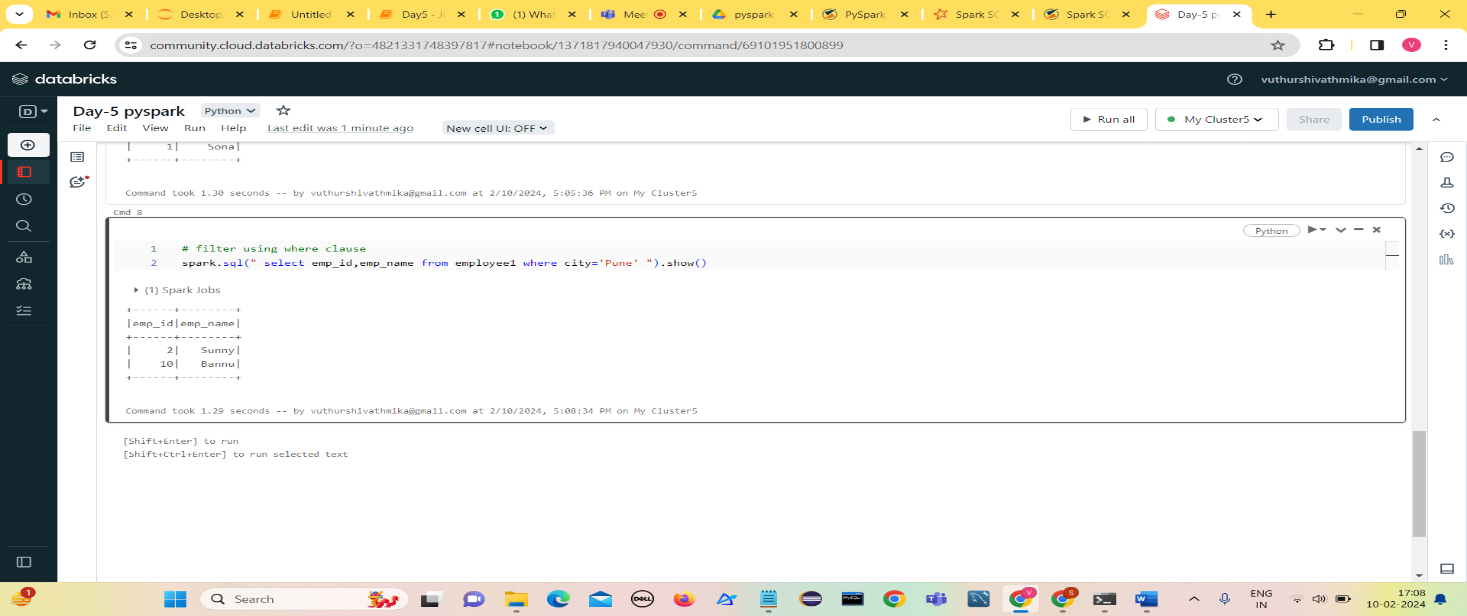
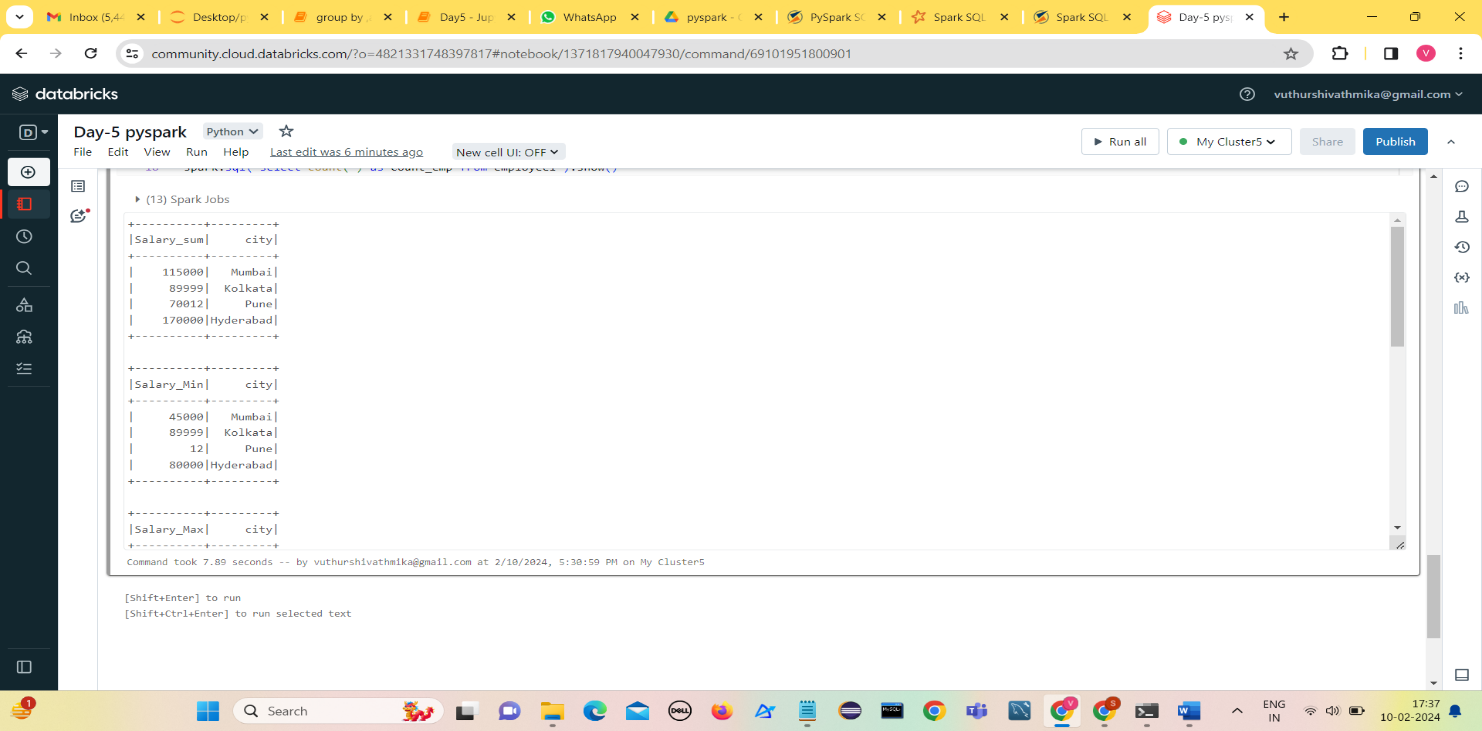
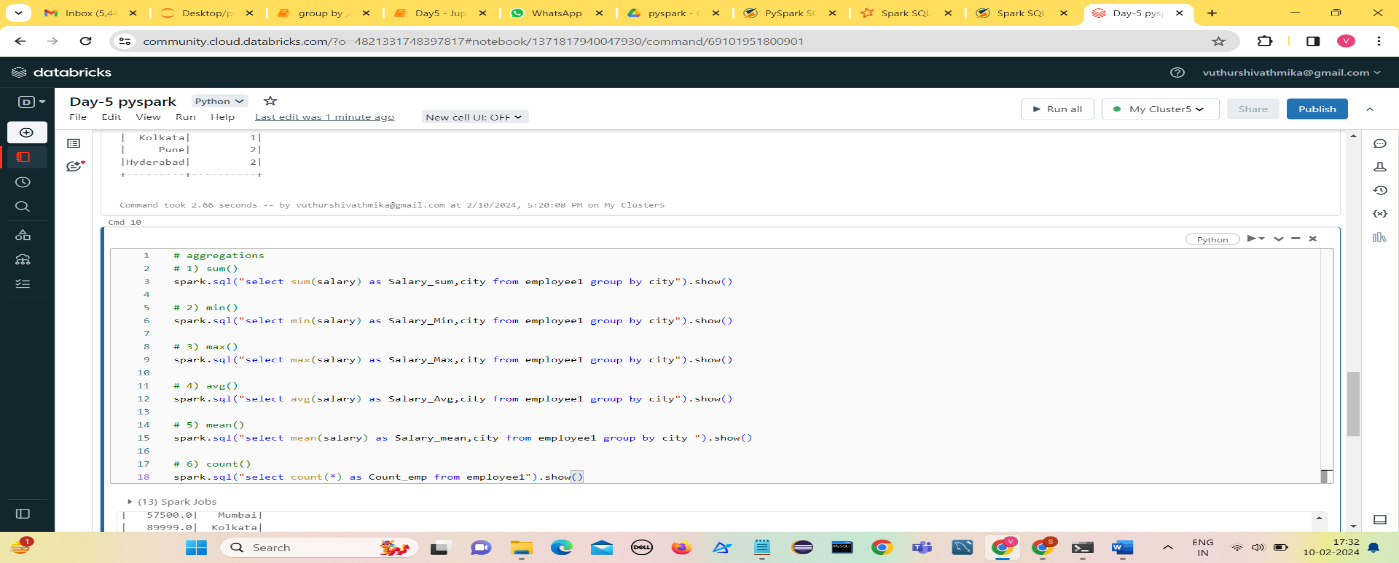
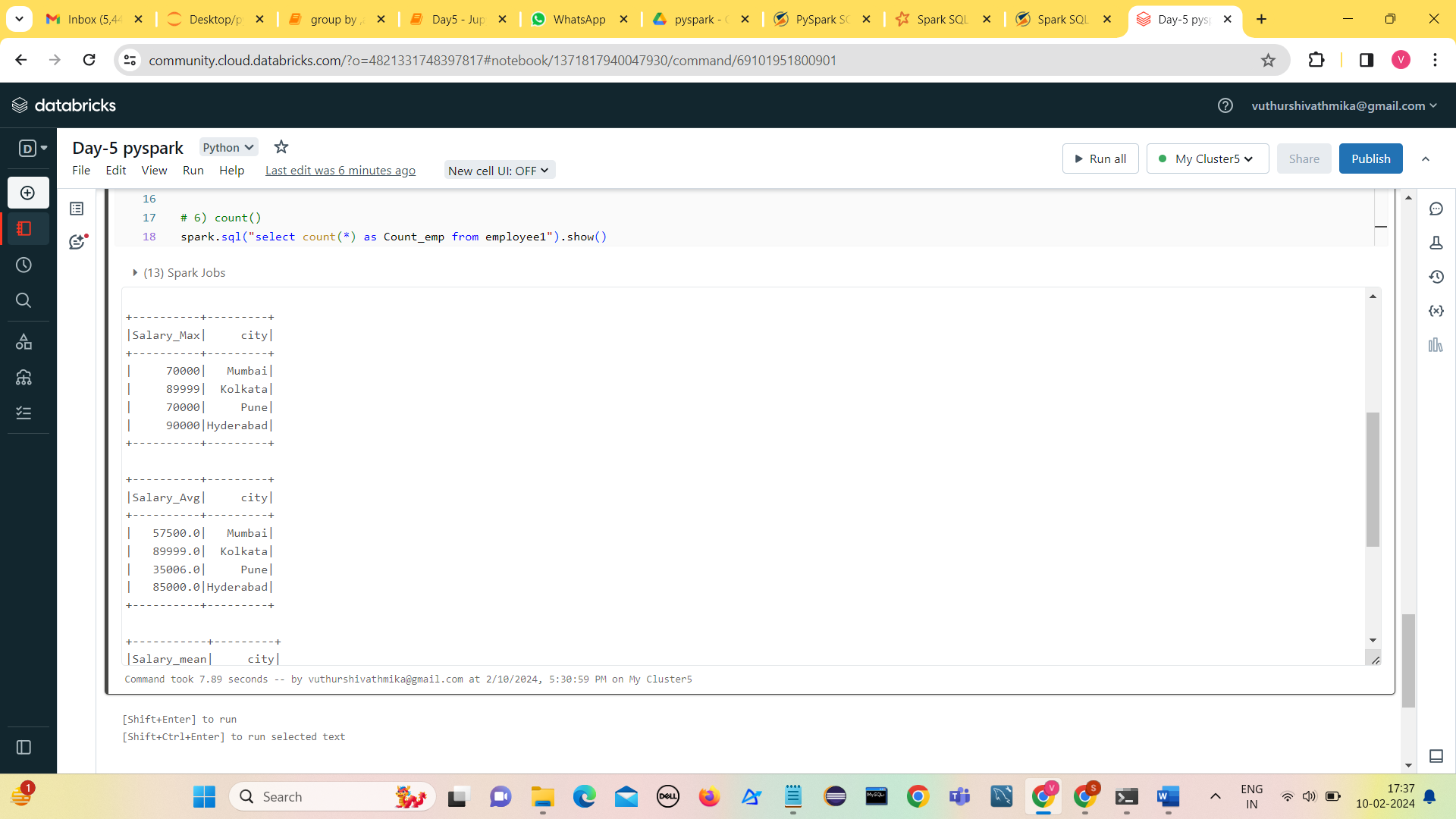
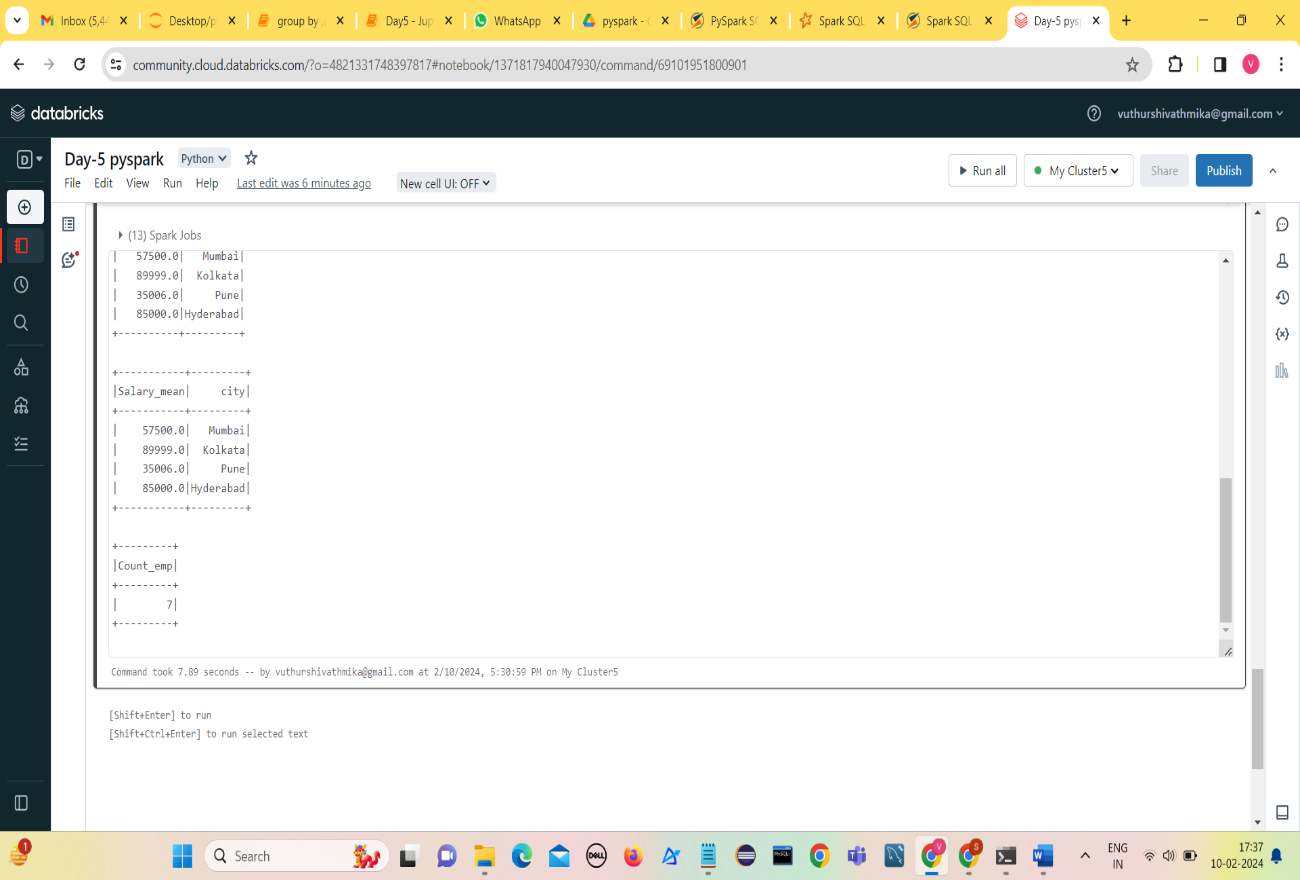
**PySpark Day-5**

**Spark SQL**

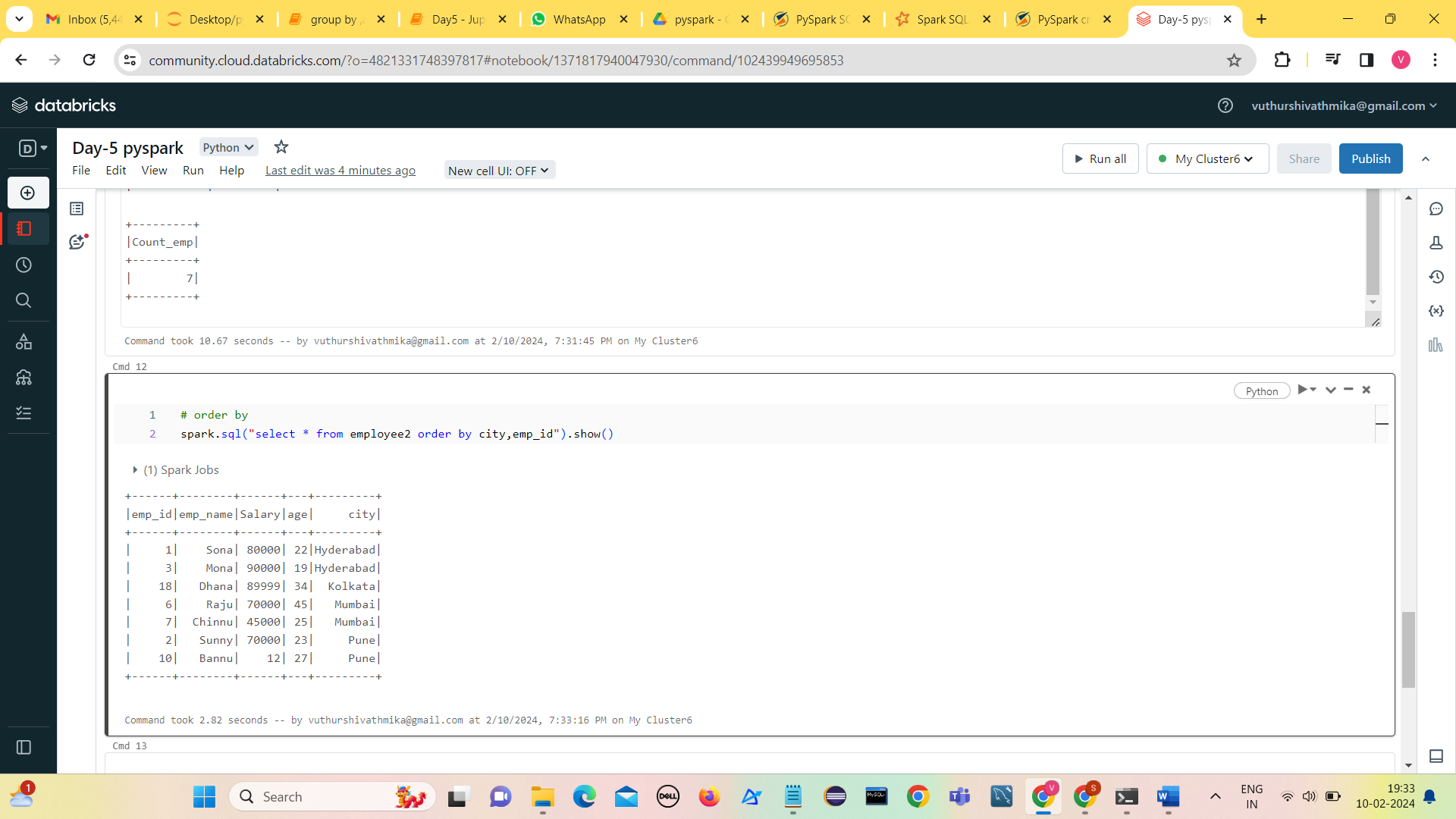
* Spark SQL is a Spark module for structured data processing.
* Spark SQL provide Spark with more information about the structure of both the data and the computation being performed.
* Internally, Spark SQL uses this extra information to perform extra optimizations.
* There are several ways to interact with Spark SQL including SQL and the Dataset API.
* When computing a result, the same execution engine is used, independent of which API/language you are using to express the computation.

1. **Creating a database**
2. **Creating a table Employee1**
3. **Insert values into Employee1 table and display them**
4. **select required columns**
5. **Filter using where clause**
6. **Group by**
7. **Aggregations**

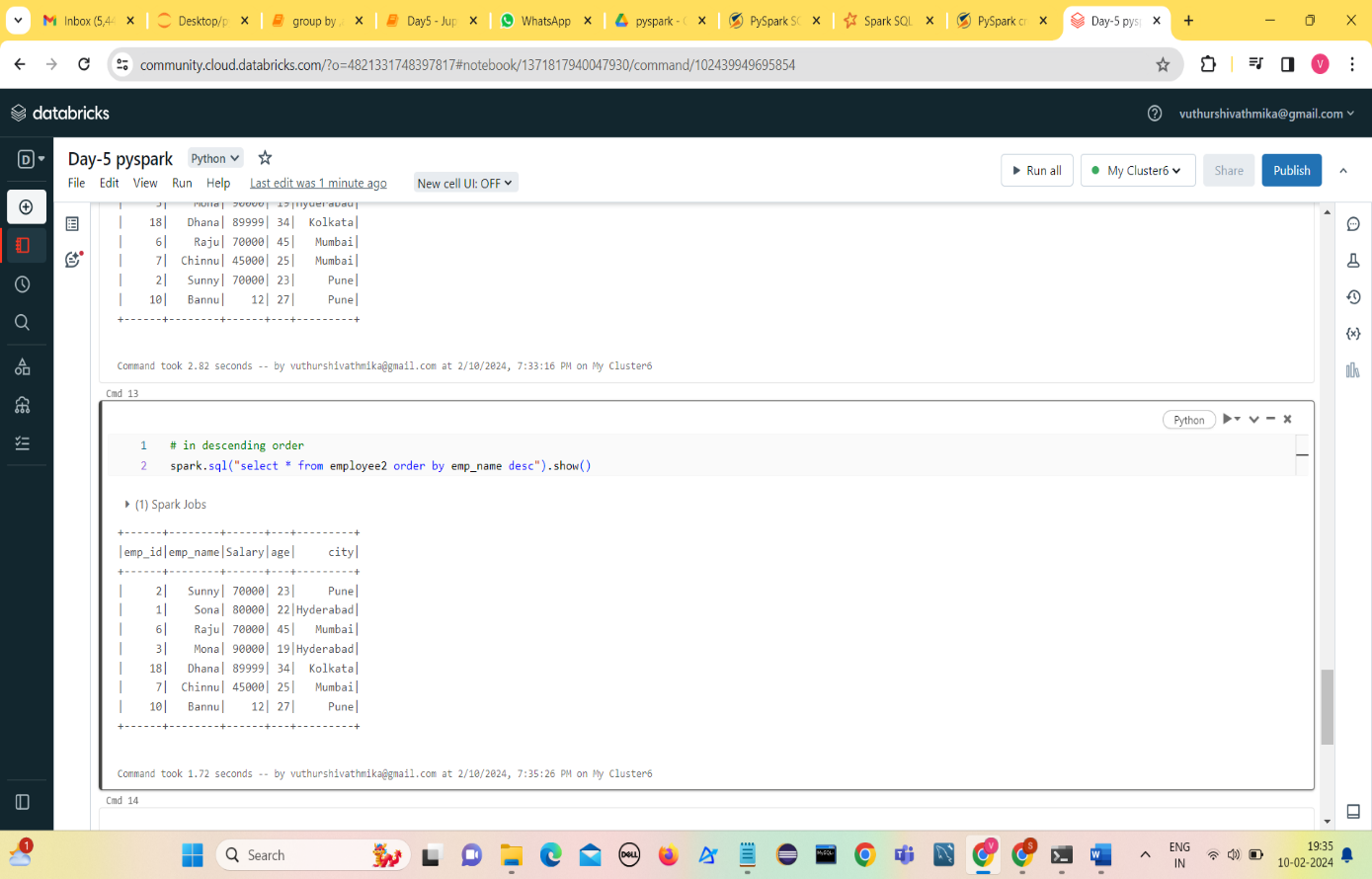




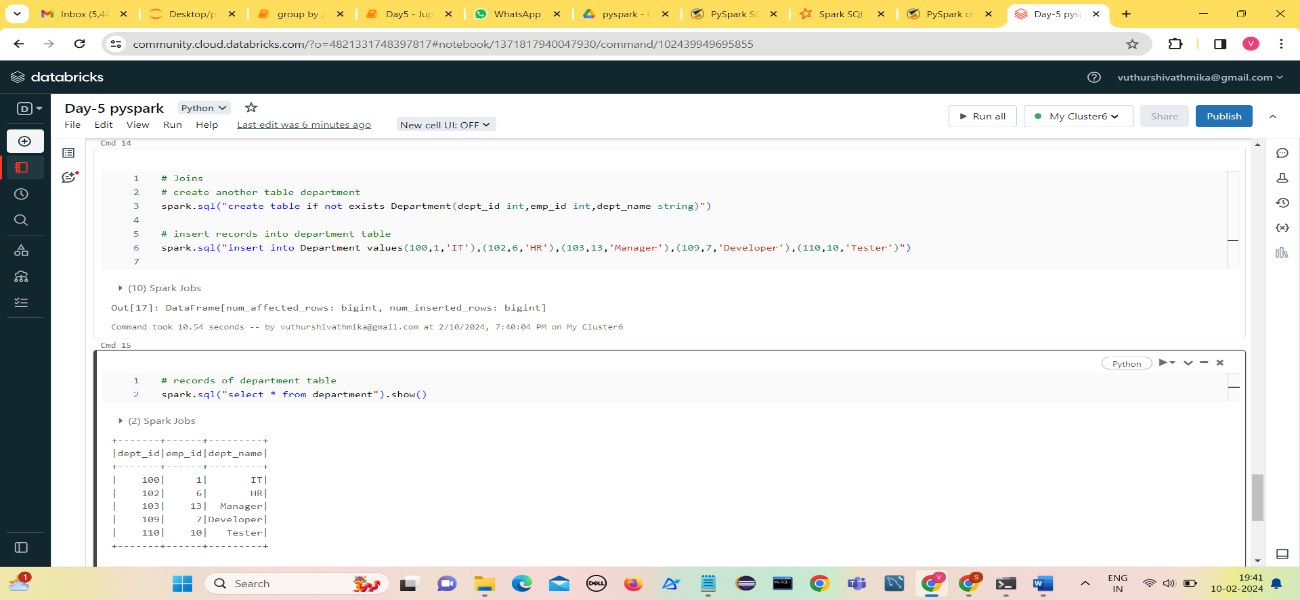
1. **Order by**

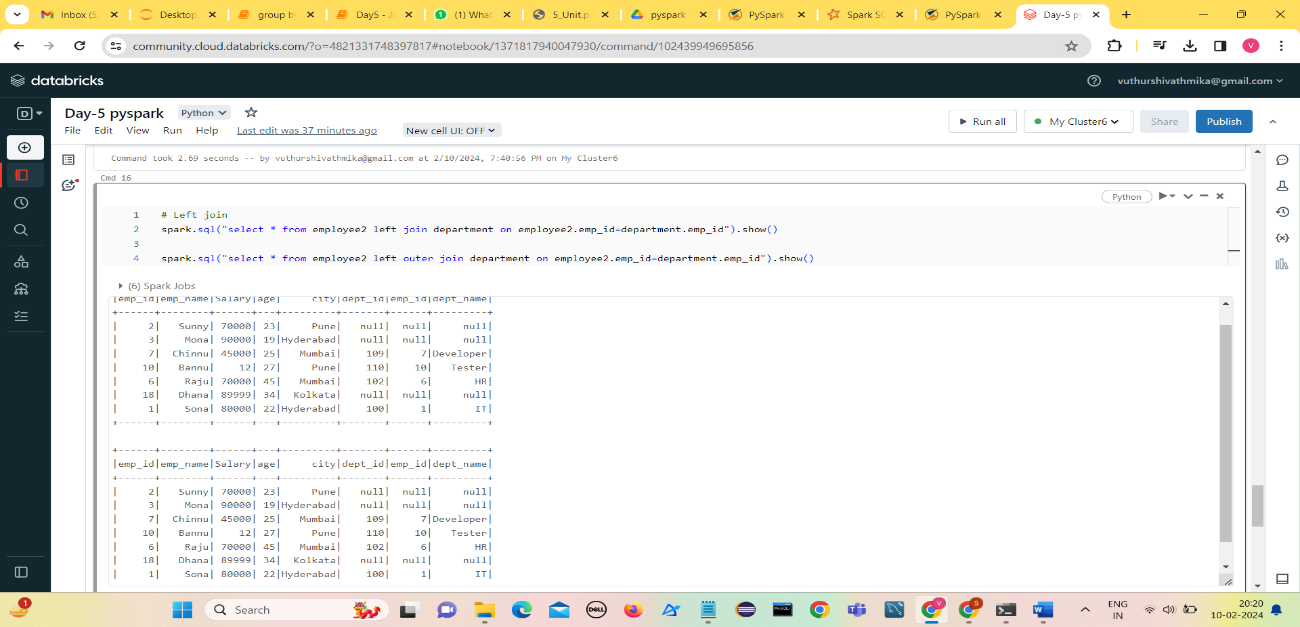
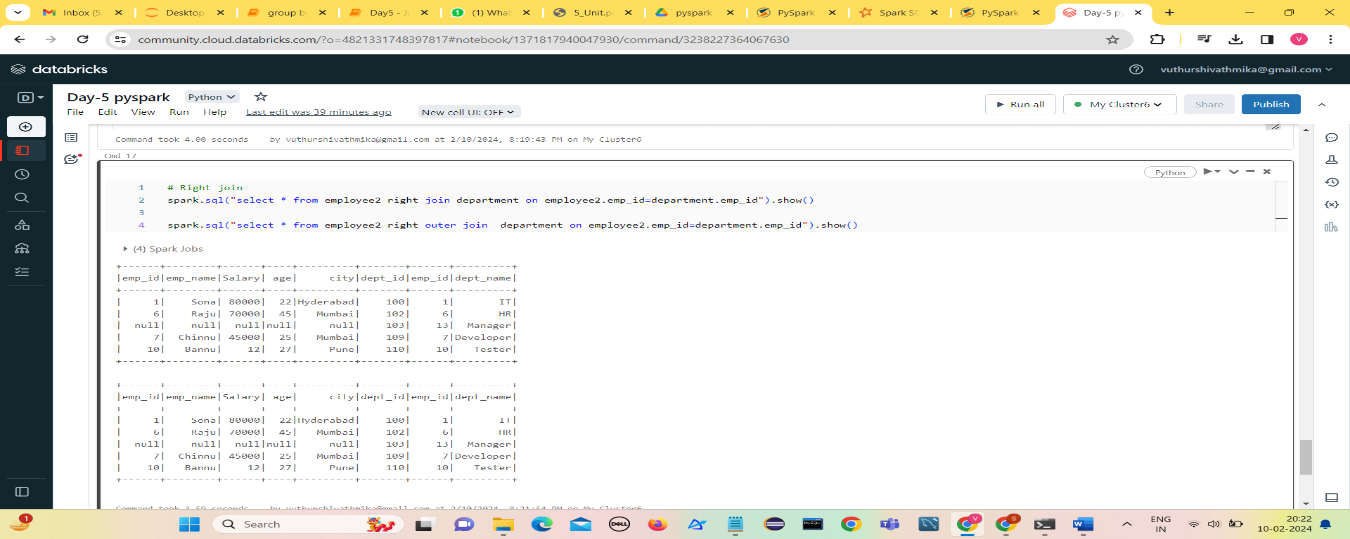
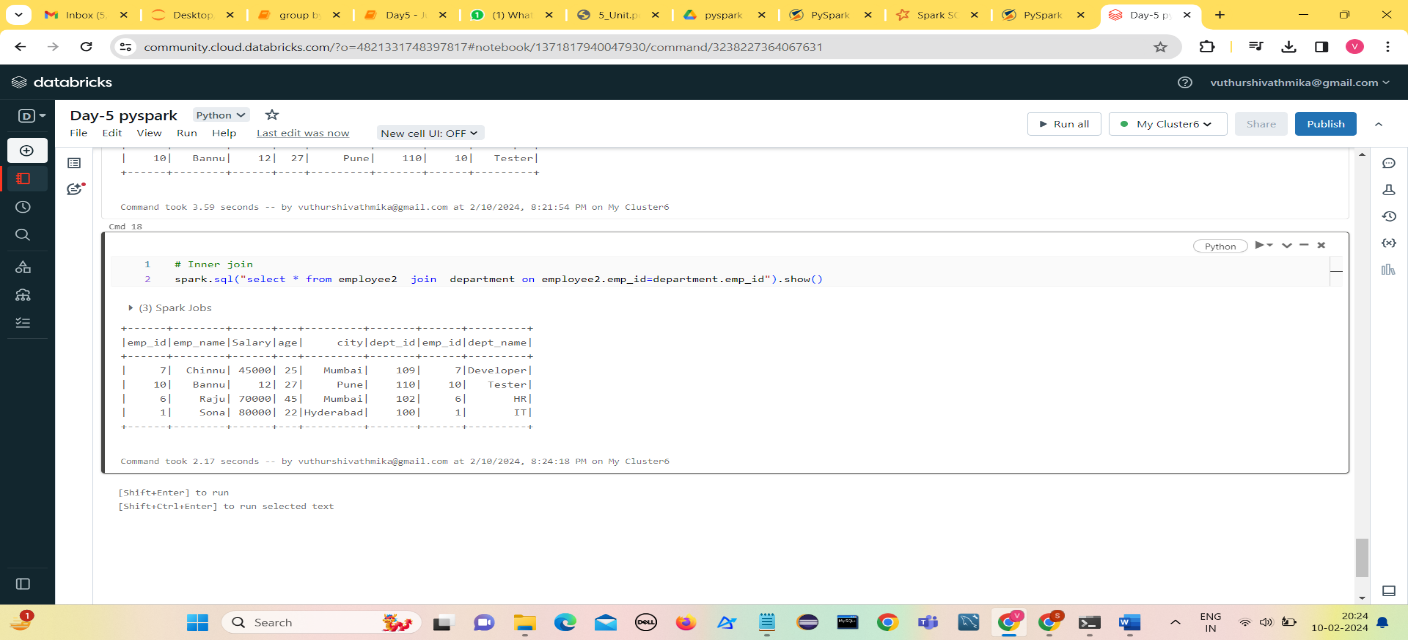
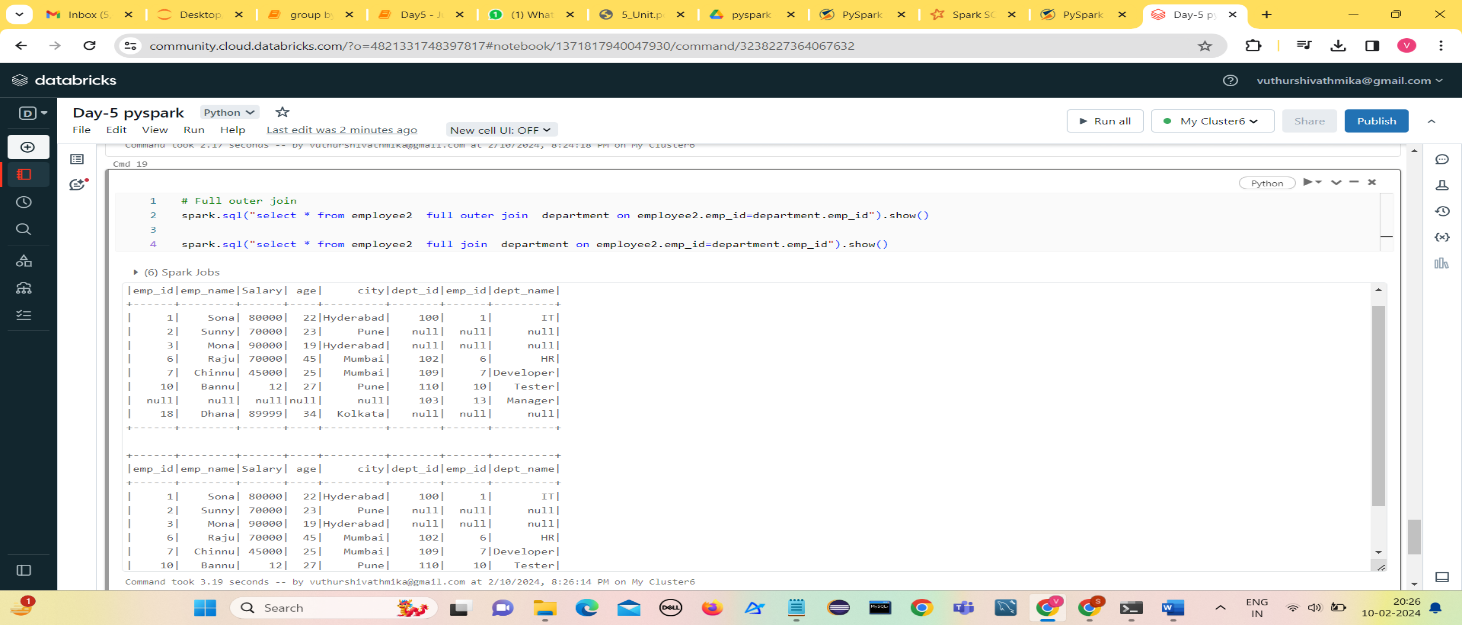
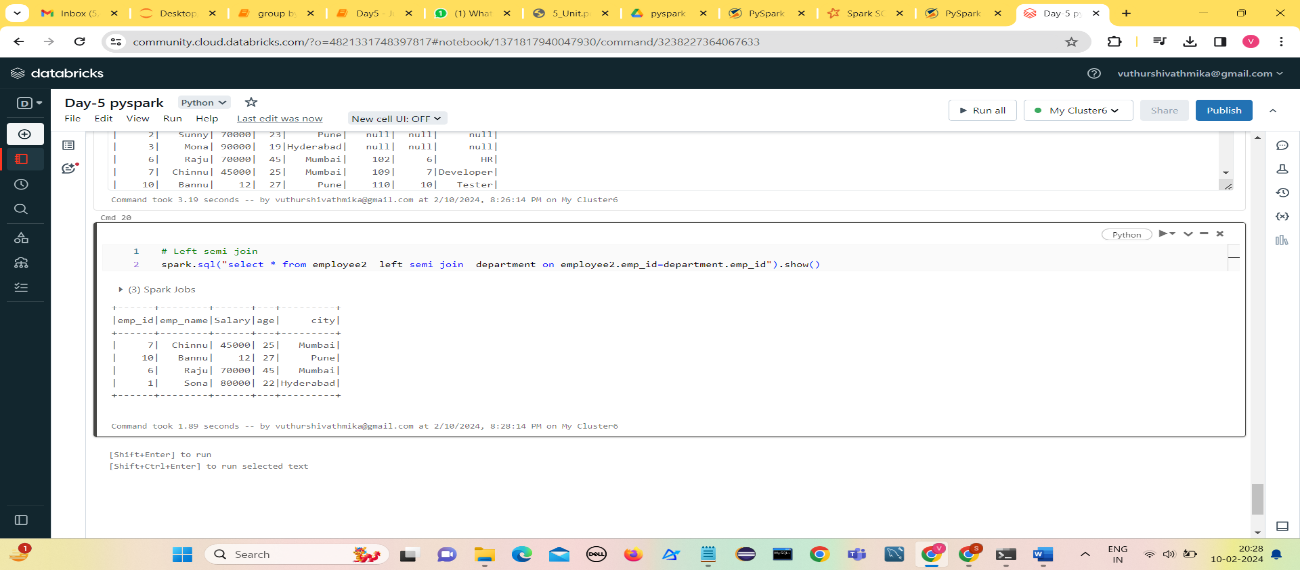
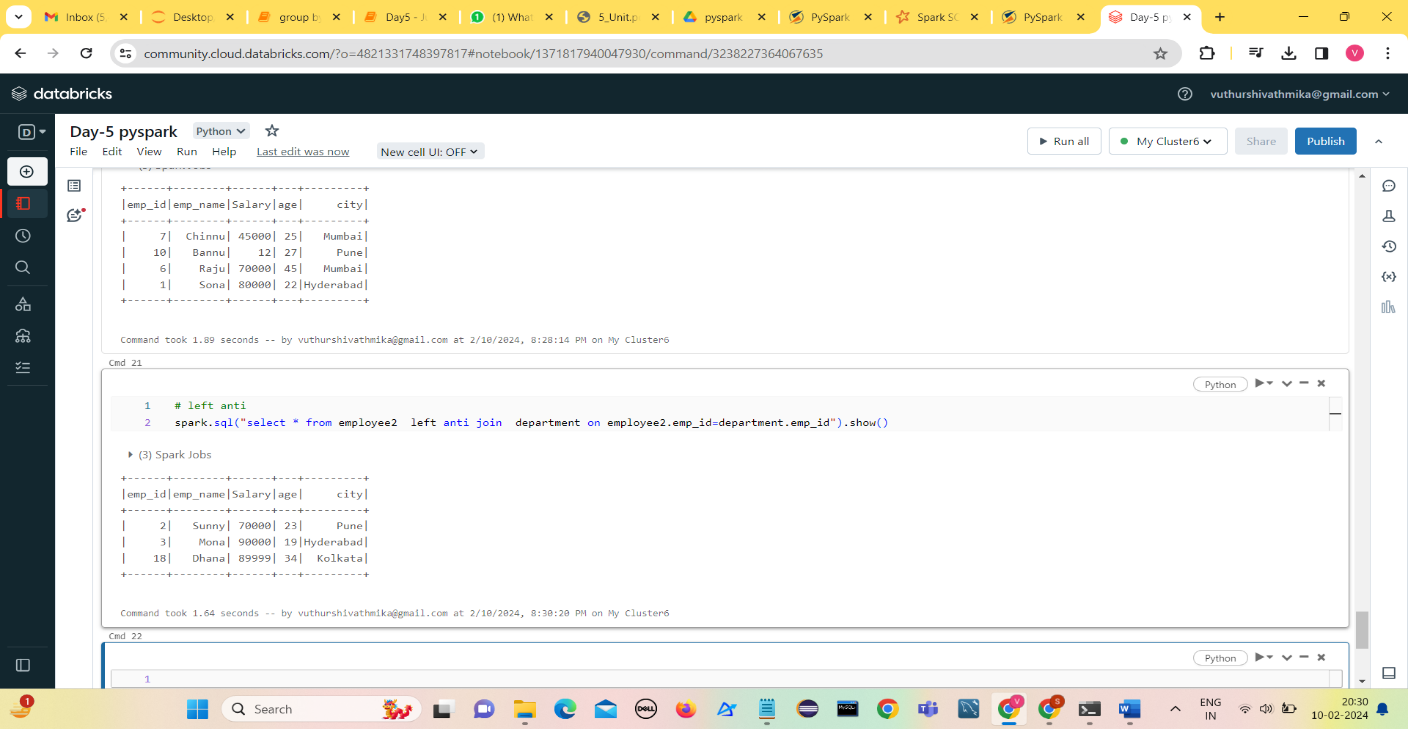


**Order by in descending order**

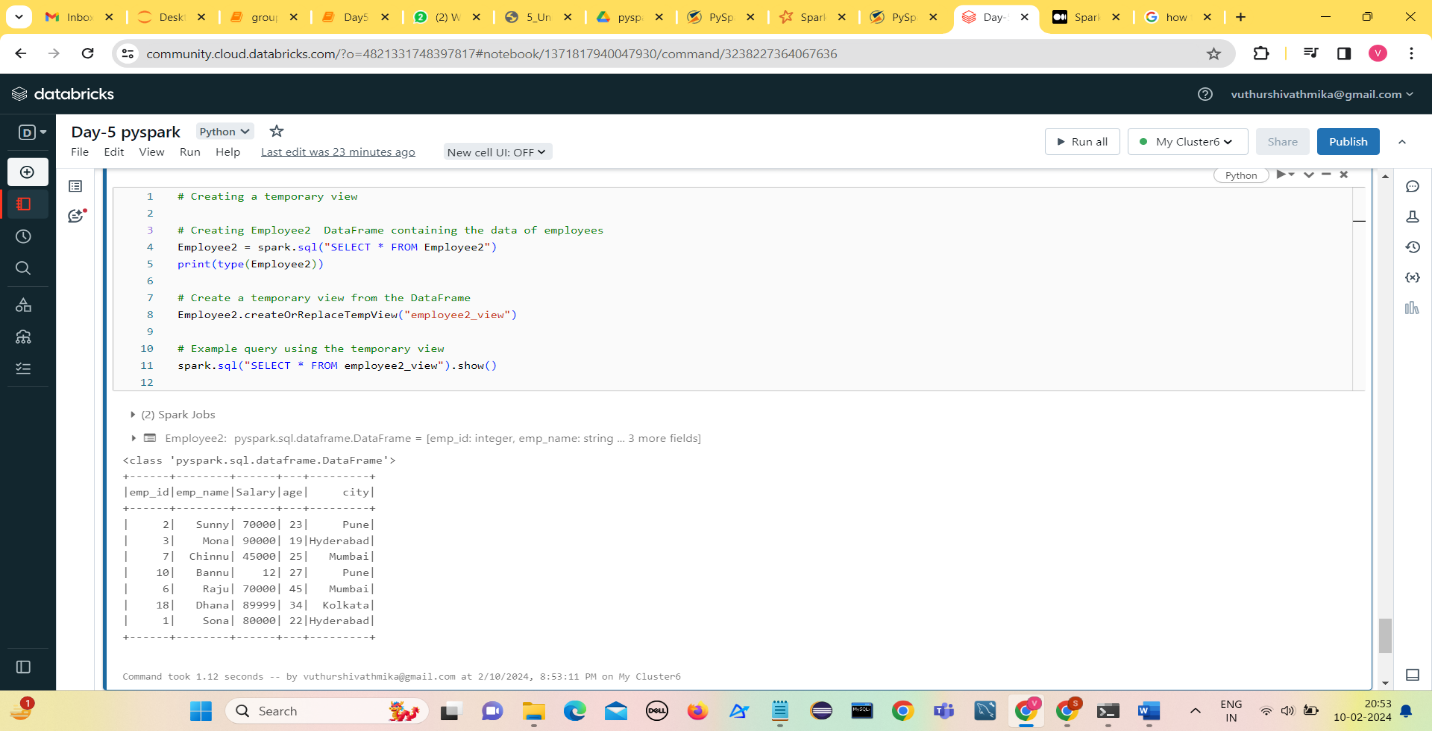


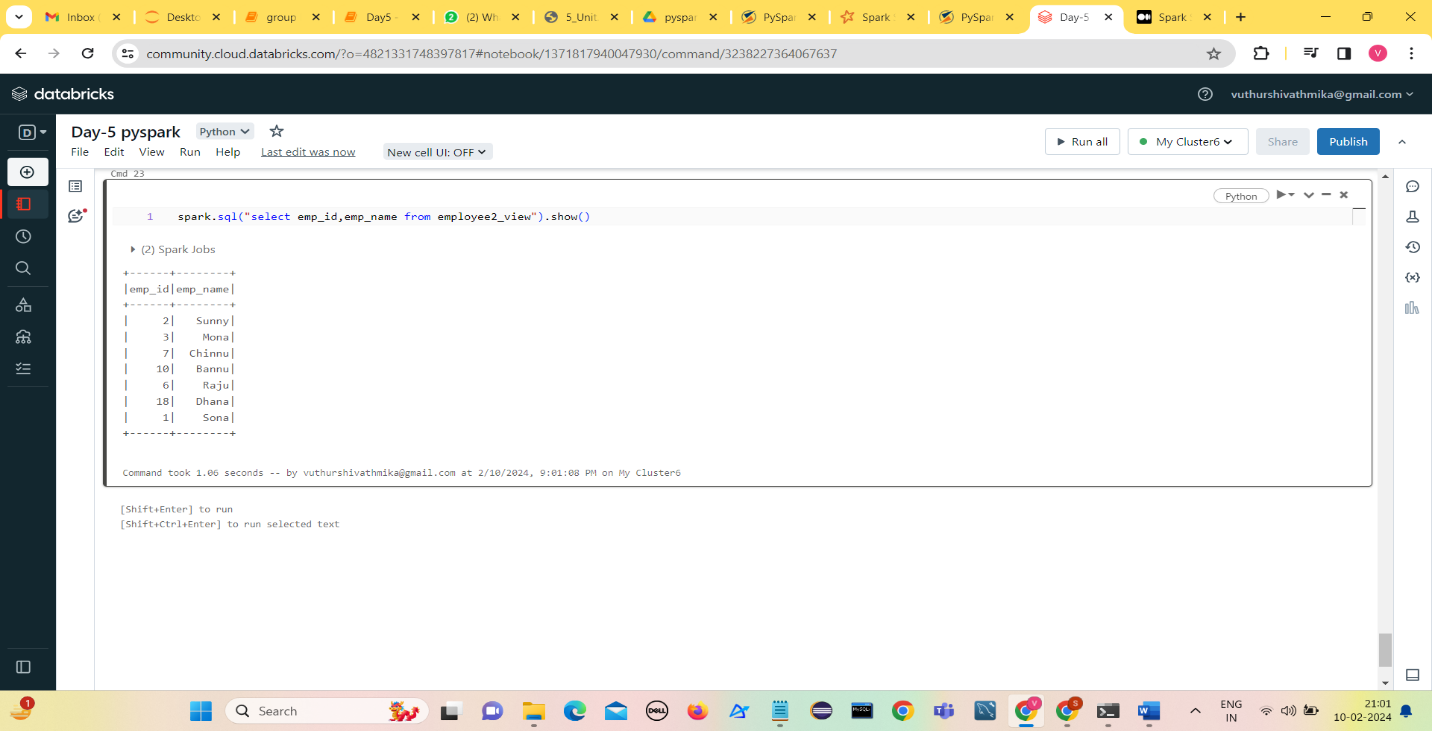
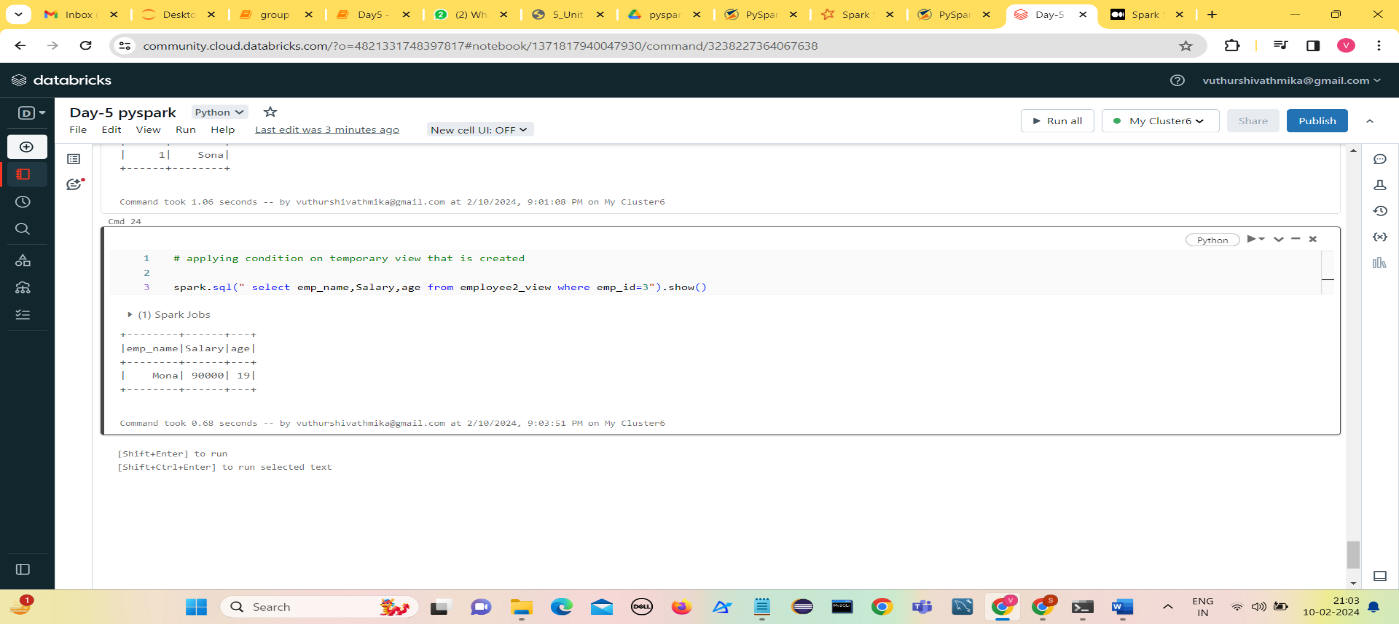
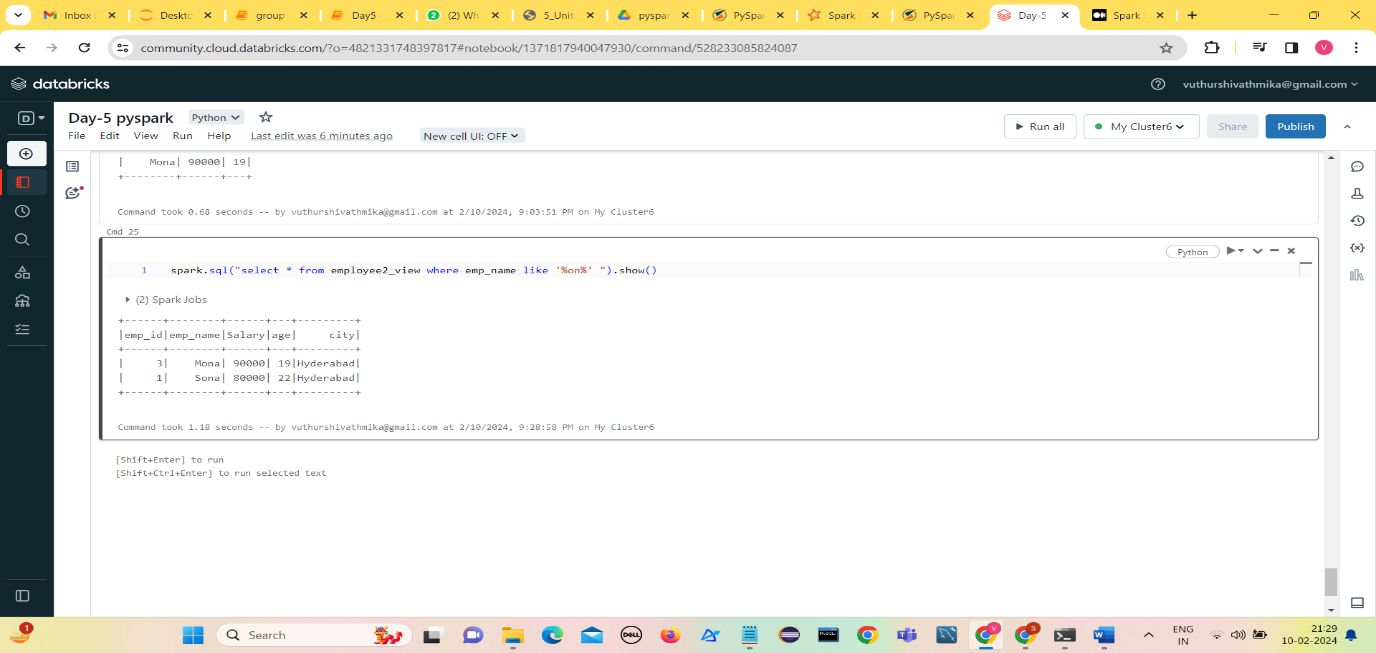
1. **Joins**

Create a table department and insert values into it.

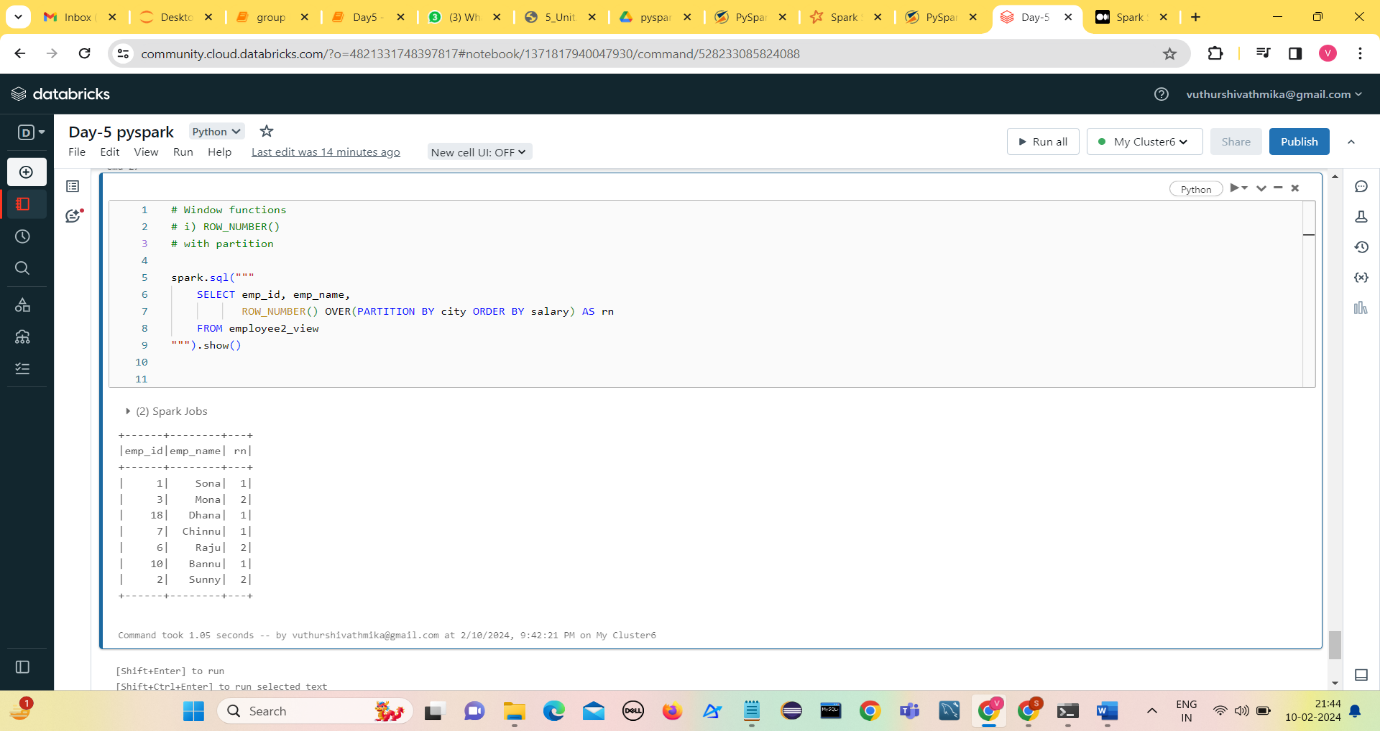
1. **Left join/left outer join**
2. **Right join/right outer join**
3. **Inner Join**
4. **Full join or full outer join**
5. **Left semi join**
6. **Left anti join**
7. **Creating Temporary views**

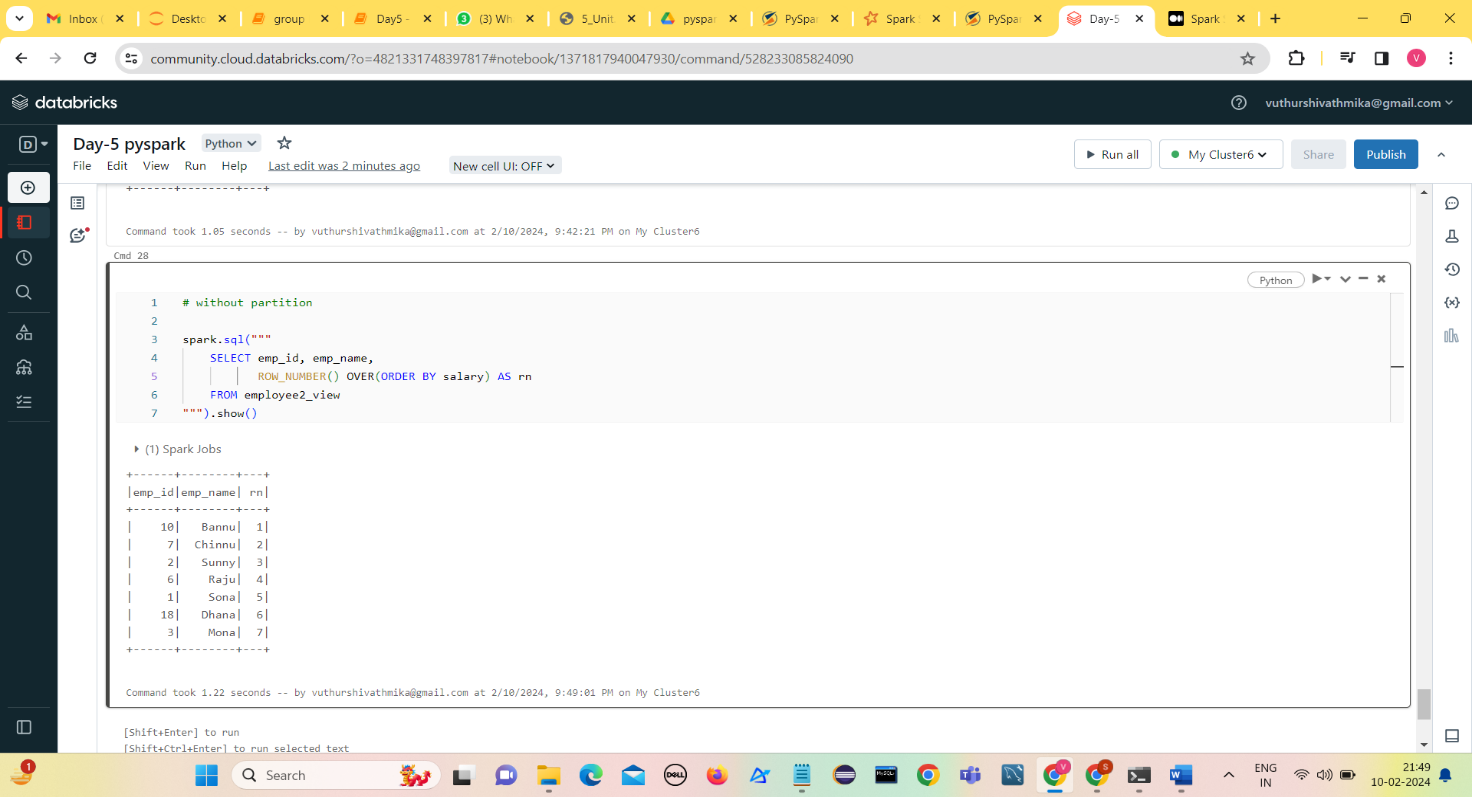
* A Temporary view in PySpark is similar to a real SQL table that contains rows and columns but the view is not materialized into files.
* createOrReplaceTempView() in PySpark creates a view only if not exist, if it exits it replaces the existing view with the new one.
* createOrReplaceTempView() is used when you wanted to store the table for a specific SparkSession.



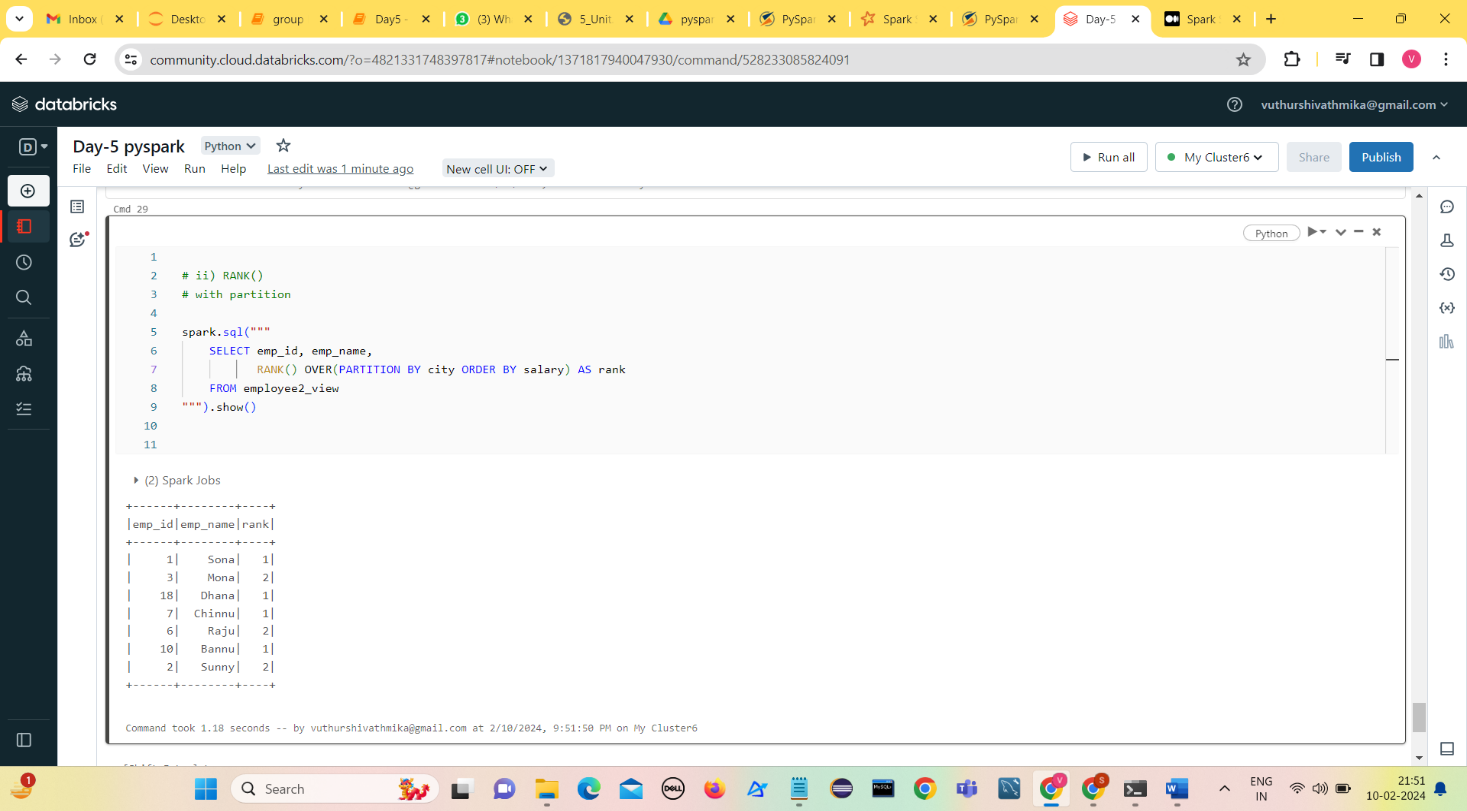


1. **Window functions:**
2. **Row\_number()**

**With partition**

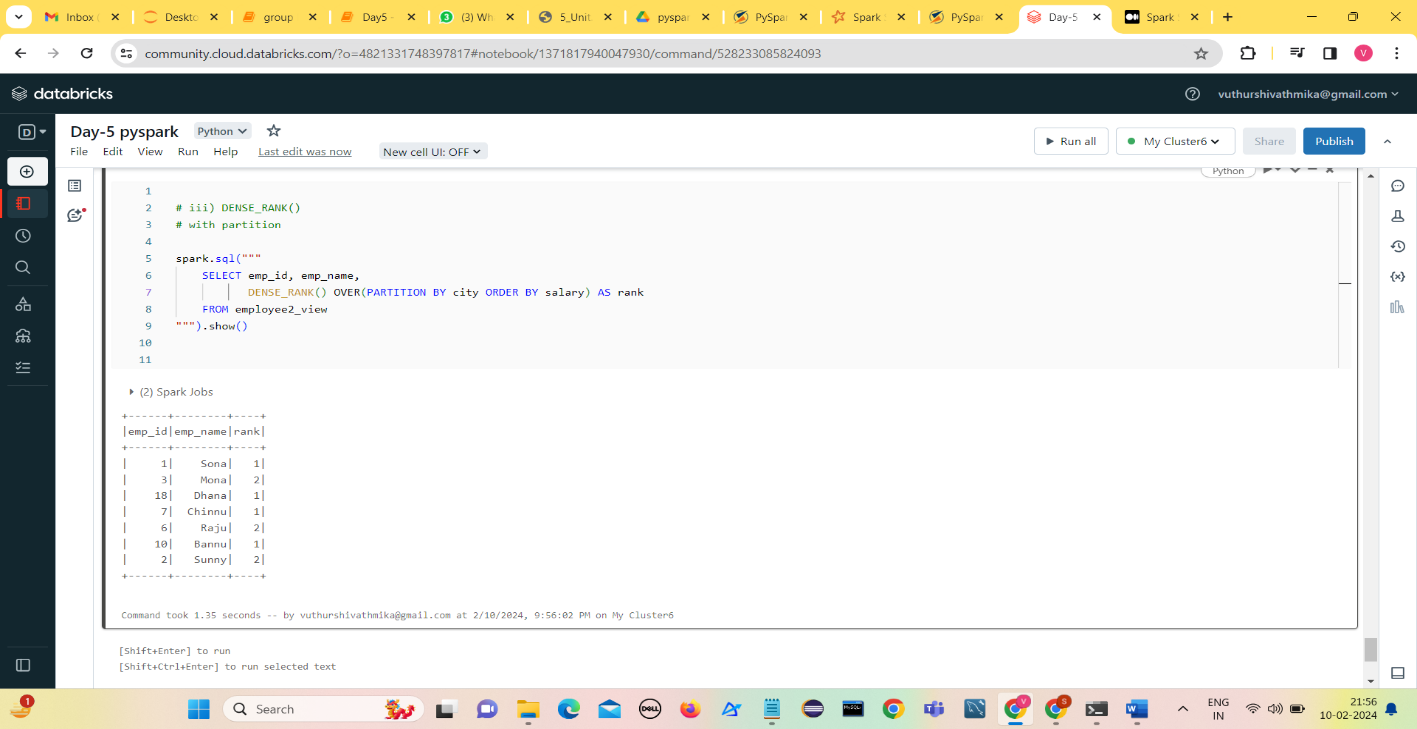
**Without partition**

1. **RANK()**

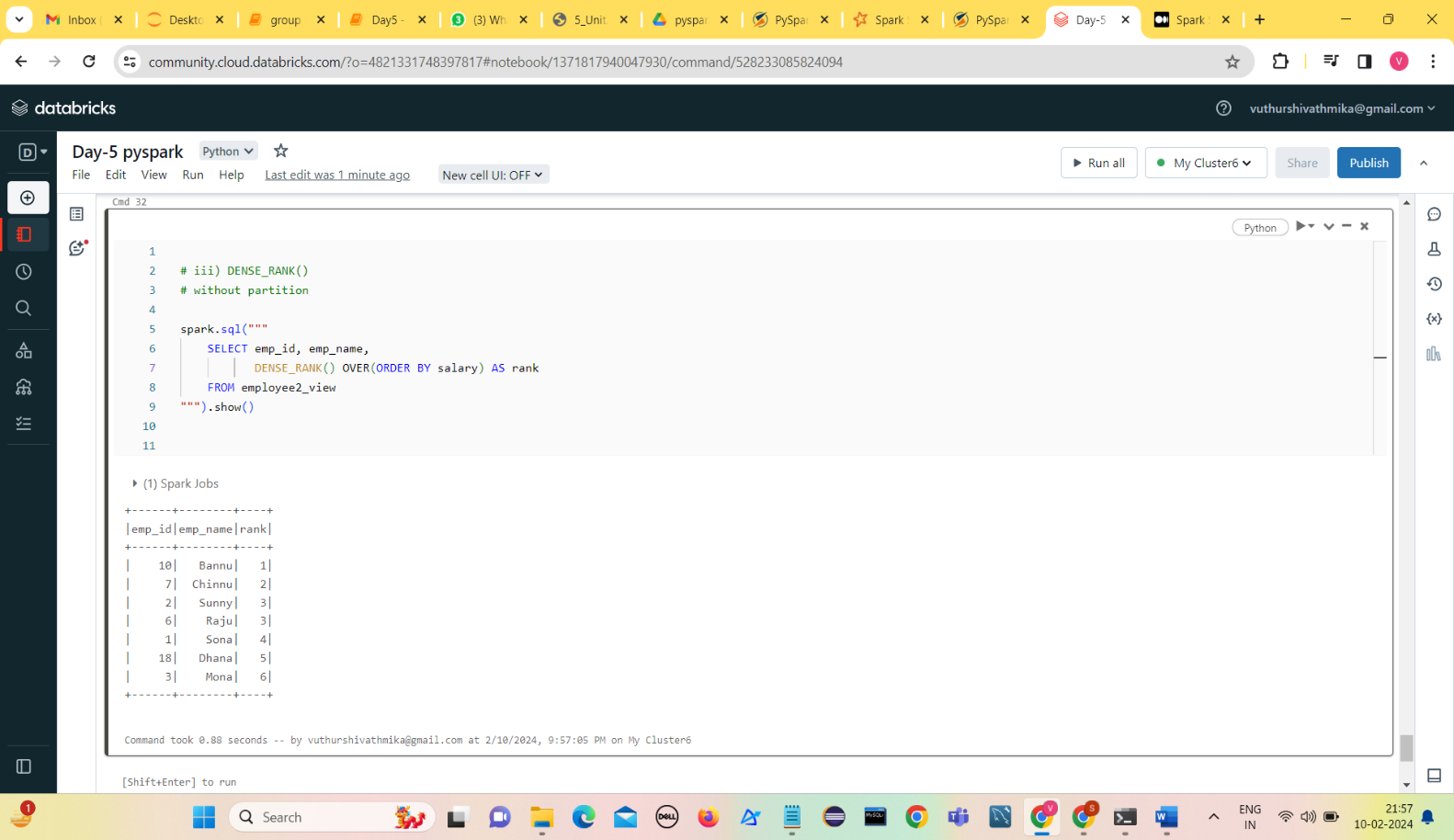
**With partition**

**Without partition**

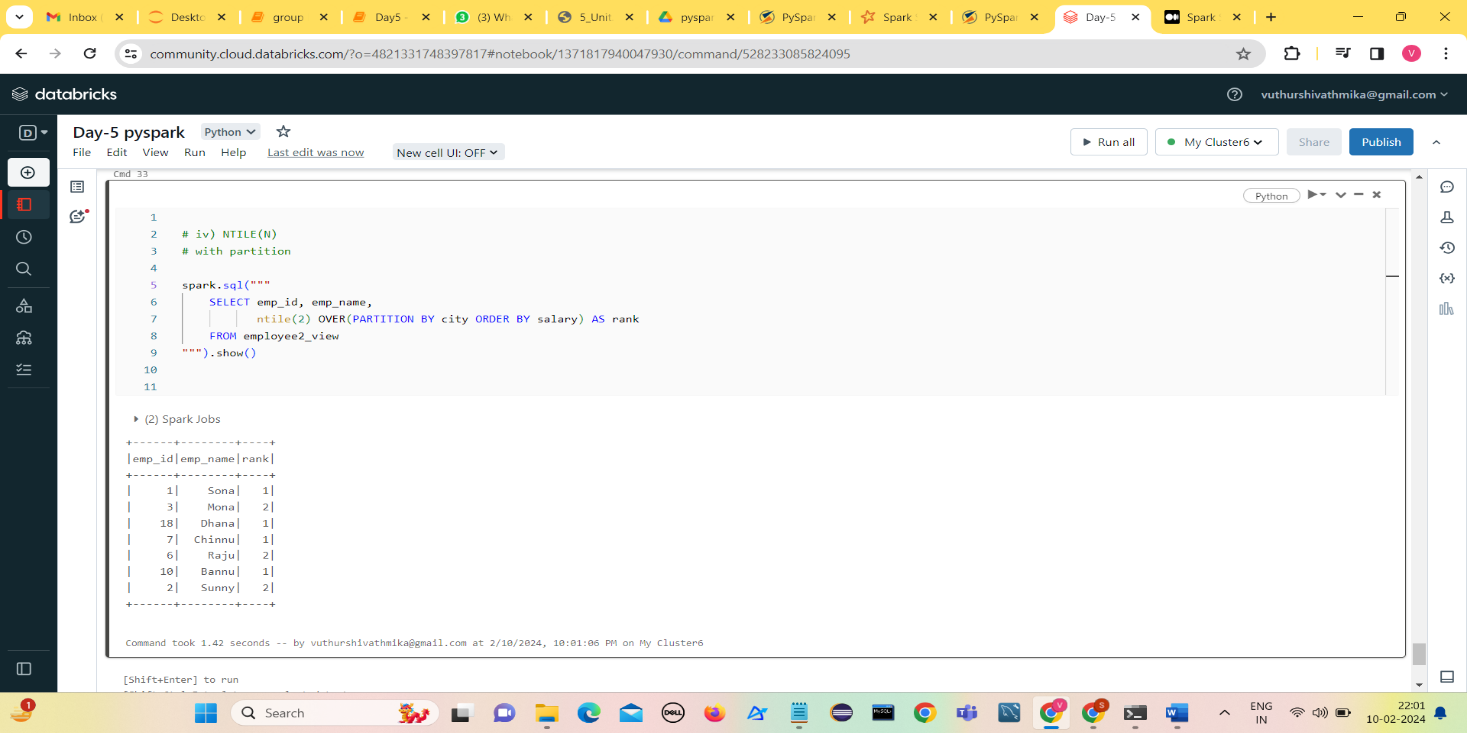
1. **DENSE\_RANK()**

**With partition**

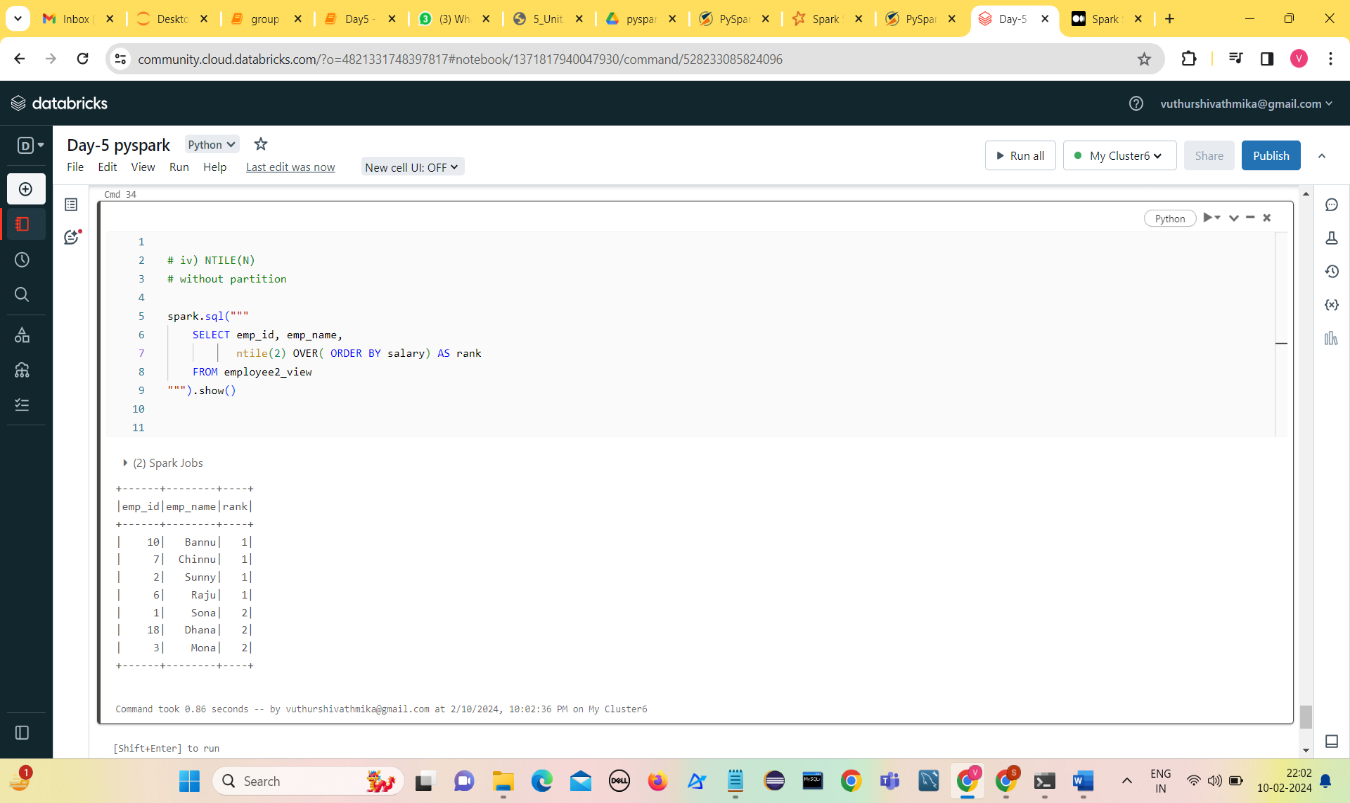
**Without partition**

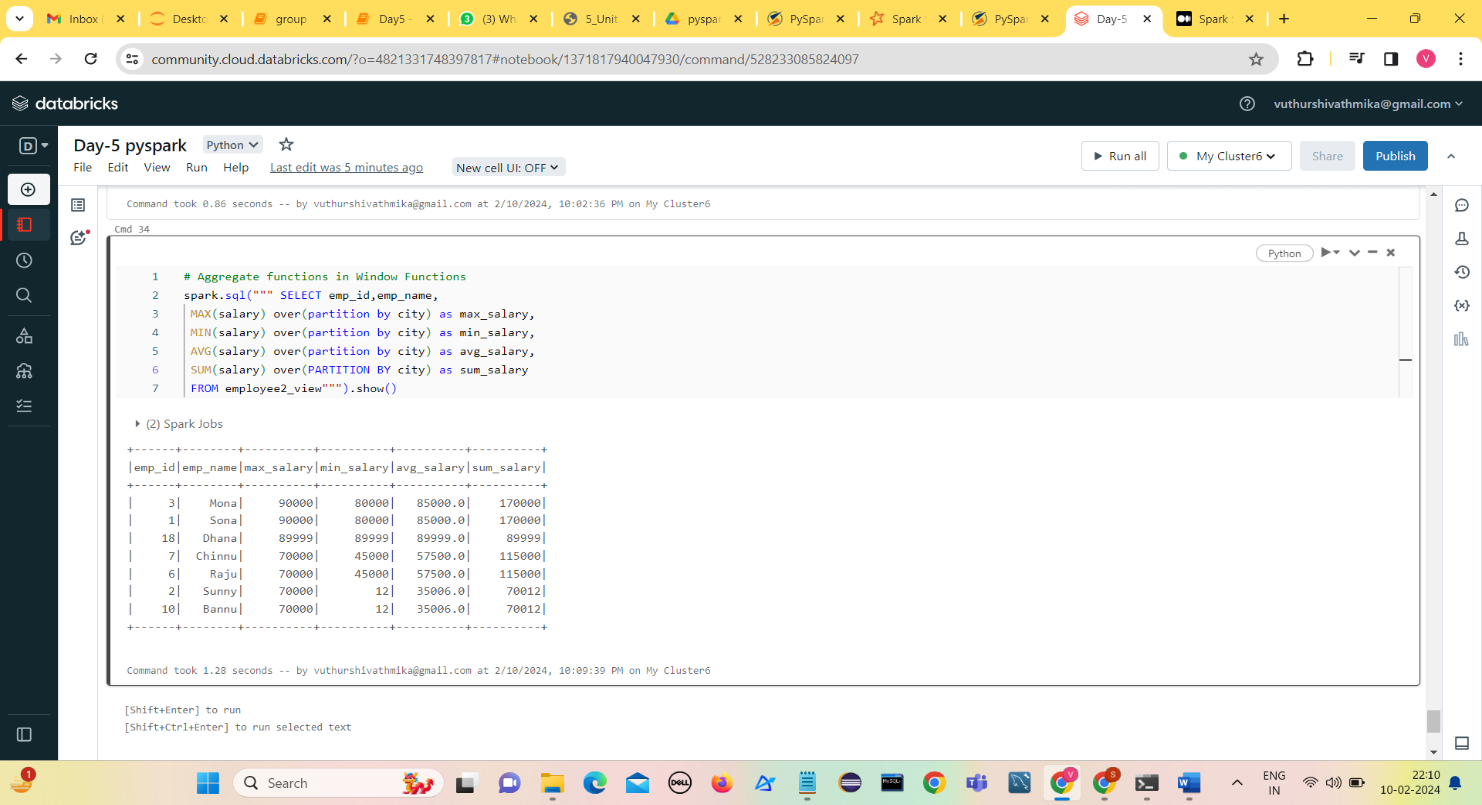


1. **Ntile(n)**

**With partition**

**Without partition**



**Window Functions---Aggregate functions**