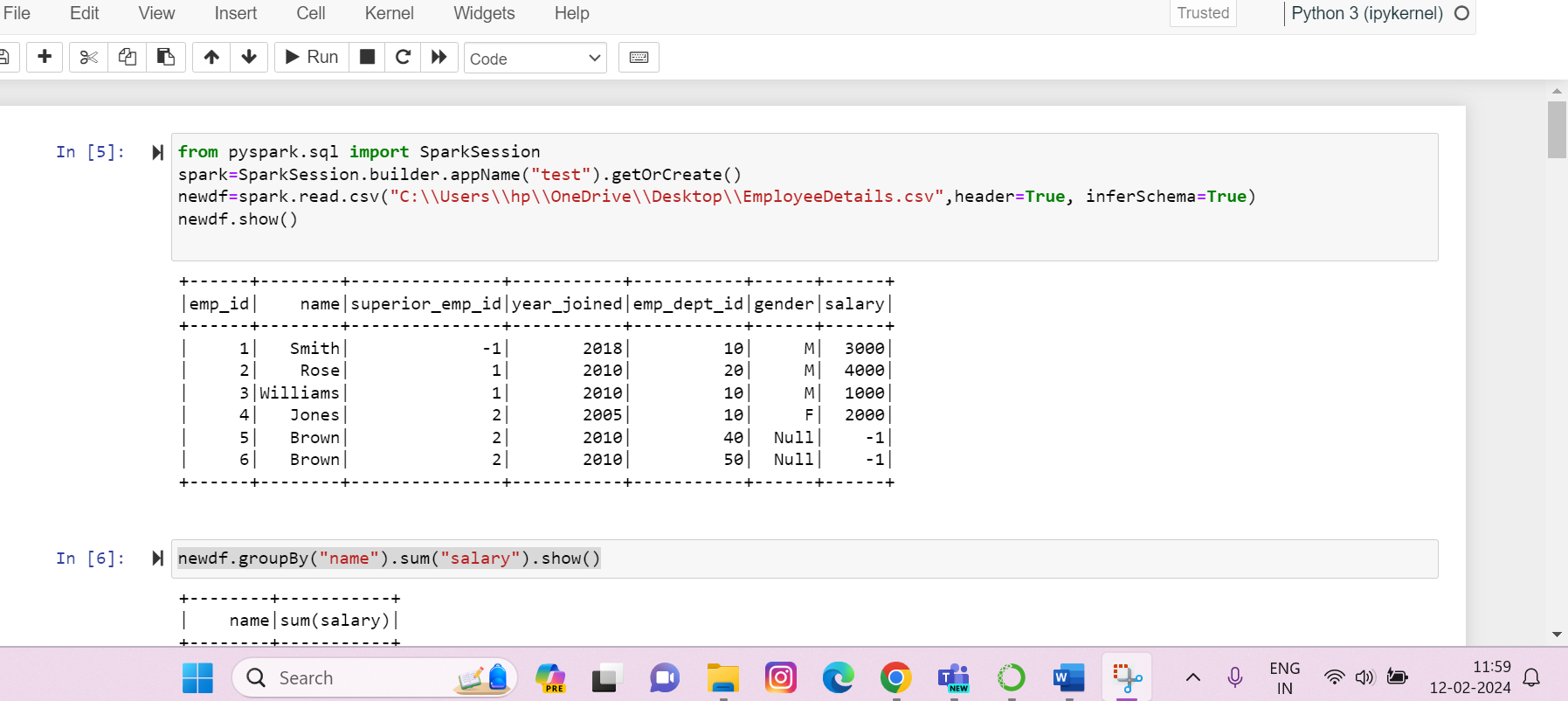
**Coding Challenge**

**Question1:**

Execute Manipulating, Droping, Sorting, Aggregations, GroupBy, Joining in DataFrames

**Schema that I will be using for all the functions:**

****

**GroupBy and Aggregations()**

**GroupBy and SUM():**

**A screenshot of a computer

Description automatically generated**

**GroupBy and COUNT():**

**A screenshot of a computer

Description automatically generated**

**GroupBy and AVG ():**

**A screenshot of a computer

Description automatically generated**

**GroupBy and MIN():**

**A screenshot of a computer

Description automatically generated**

**GroupBy and MAX():**

**A screenshot of a computer

Description automatically generated**

**GroupBy and MEAN ():**

**A screenshot of a computer

Description automatically generated**

**GroupBy and AGG ():**

**A screenshot of a computer

Description automatically generated**

**Droping**

**Using na.drop(): It drops all the rows with null values.**

**A screenshot of a computer

Description automatically generated**

**The three parameters of dropping are:**

**1)How**

**2)Thresh**

**3)Subset**

**Using na.drop with how=”all”: -it drops rows if all values in rows are null**

**A screenshot of a computer

Description automatically generated**

**Using na.drop with how=” any” and thresh: thresh specifies the number of non-null values that should be present:**

**In this case atleast 2 non-null values should be present**

**A screenshot of a computer

Description automatically generated**

**Using na.drop with how=”any” and subset: subset specifies the column that only in that the rows get deleted:**

**A screenshot of a computer

Description automatically generated**

**Sorting**

**By default the sort() function is set to ascending order:**

**A screenshot of a computer

Description automatically generated**

**Sort in descending order:**

**A screenshot of a computer

Description automatically generated**

**Sorting with two columns:**

**A screenshot of a computer

Description automatically generated**

**We can also use Orderby() to arrange the data:  
A screenshot of a computer

Description automatically generated**

**-----Continued Below------**

**Joining**

**Two datafrmaes on which we will perform joins:**

**A screenshot of a computer

Description automatically generated**

**Inner Join:**

**A screenshot of a computer

Description automatically generated**

**Outer Join:**

**A screenshot of a computer

Description automatically generated**

**Left Join:**

**A screenshot of a computer

Description automatically generated**

**Right Join:**

**A screenshot of a computer

Description automatically generated**

**LeftSemi Join:**

**A screenshot of a computer

Description automatically generated**

**Leftanti Join:**

**A screenshot of a computer

Description automatically generated**