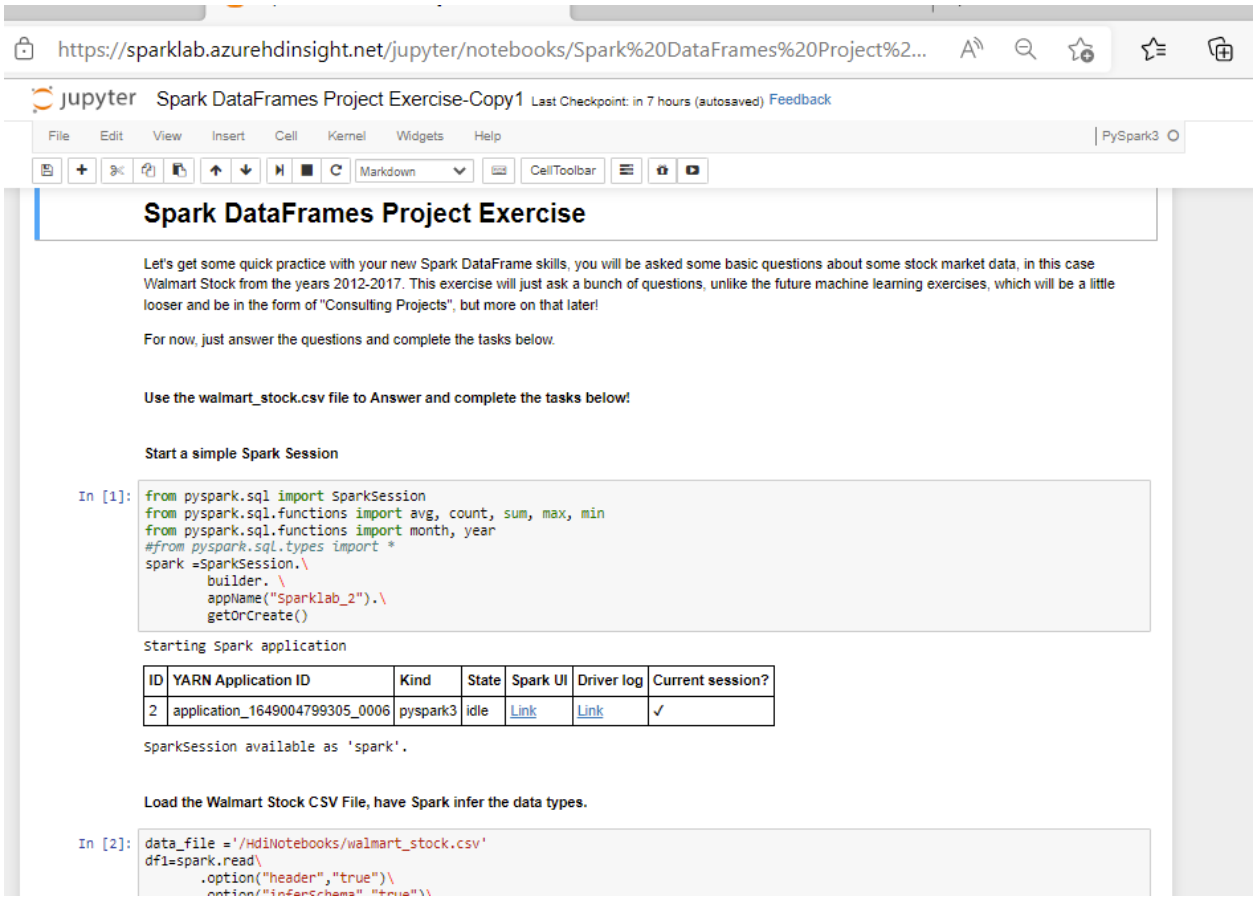
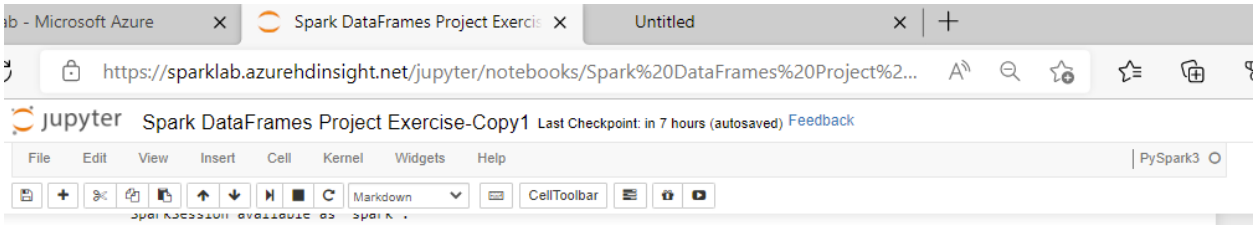

https://sparklab.azurehdinsight.net/jupyter/notebooks/Spark%20DataFrames%20Project%2...
jupyter Spark DataFrames Project Exercise-Copy1 Last Checkpoint: in 7 hours (autosaved) Feedback
File Edit View Insert Cell Kernel Widgets Help
PySpark3
Markdown
CellToolbar
Spark DataFrames Project Exercise
Let's get some quick practice with your new Spark DataFrame skills, you will be asked some basic questions about some stock market data, in this case Walmart Stock from the years 2012-2017. This exercise will just ask a bunch of questions, unlike the future machine learning exercises, which will be a little looser and be in the form of "Consulting Projects", but more on that later!
For now, just answer the questions and complete the tasks below.
Use the walmart_stock.csv file to Answer and complete the tasks below!
Start a simple Spark Session
In [1]:
from pyspark.sql import SparkSession
from pyspark.sql.functions import avg, count, sum, max, min
from pyspark.sql.functions import month, year
#from pyspark.sql.types import *
spark =SparkSession.\
builder. \
appName("Sparklab_2").\
getOrCreate()
Starting Spark application
ID YARN Application ID Kind State Spark UI Driver log Current session?
2 application_1649004799305_0006 pyspark3 idle Link Link ✓
SparkSession available as 'spark'.
Load the Walmart Stock CSV File, have Spark infer the data types.
In [2]:
data_file ='/HdiNotebooks/walmart_stock.csv'
df1=spark.read\
.option("header","true")\



Load the Walmart Stock CSV File, have Spark infer the data types.

In [2]:
```
data_file ='/HdiNotebooks/walmart_stock.csv'
df1=spark.read\
       .option("header","true")\
       .option("inferSchema","true")\
       .csv(data_file)
```

What are the column names?

In [3]:
```
df1.columns
```

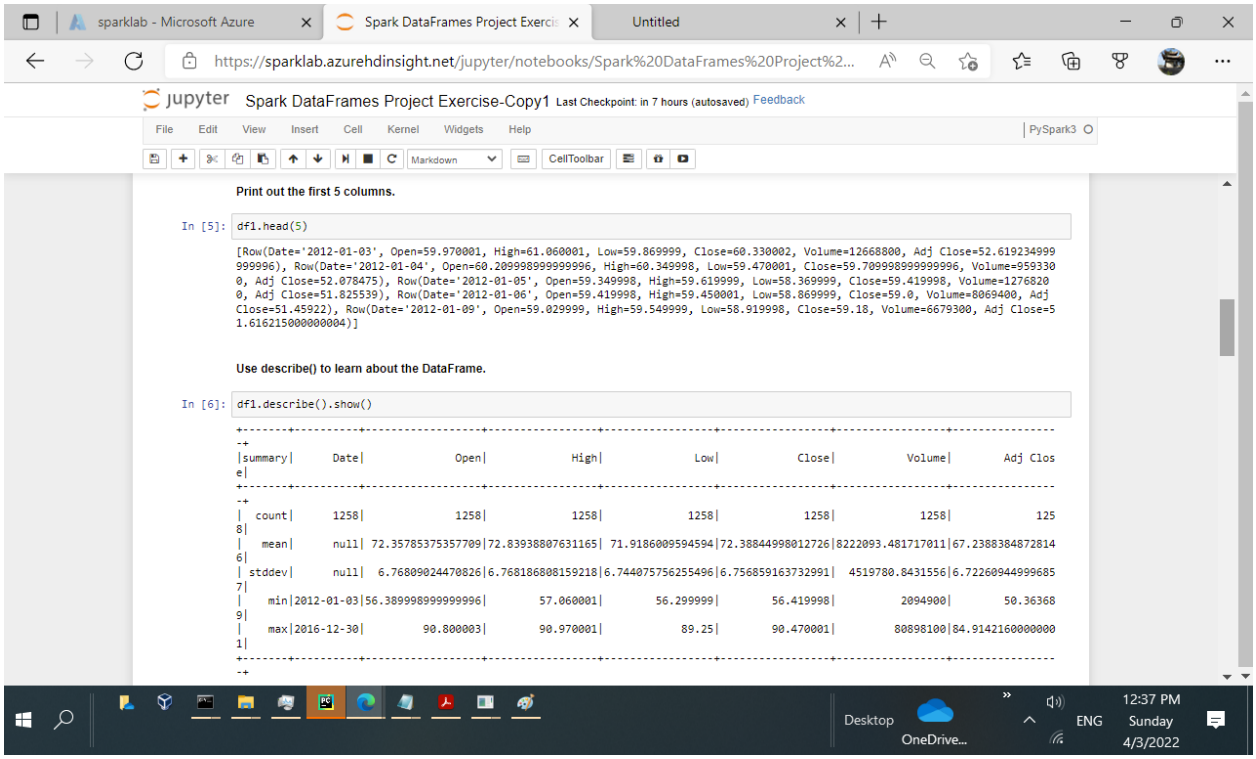
```
['Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Adj Close']
```

What does the Schema look like?

In [4]:
```
df1.printSchema()
```

```
root
 |-- Date: string (nullable = true)
 |-- Open: double (nullable = true)
 |-- High: double (nullable = true)
 |-- Low: double (nullable = true)
 |-- Close: double (nullable = true)
 |-- Volume: integer (nullable = true)
 |-- Adj Close: double (nullable = true)
```

Print out the first 5 columns.



https://sparklab.azurehdinsight.net/jupyter/notebooks/Spark%20DataFrames%20Project%2...

File Edit View Insert Cell Kernel Widgets Help PySpark3

Markdown CellToolbar

Bonus Question!

There are too many decimal places for mean and stddev in the describe() dataframe. Format the numbers to just show up to two decimal places. Pay careful attention to the datatypes that .describe() returns, we didn't cover how to do this exact formatting, but we covered something very similar.
Check this link for a hint

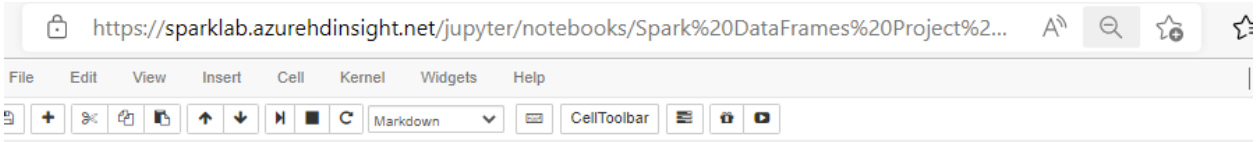
If you get stuck on this, don't worry, just view the solutions.

In [7]:
```
from pyspark.sql.functions  import col

from pyspark.sql.types import DecimalType

df1_1 = df1.describe()
df1_1.printSchema()
```

```
root
 |-- summary: string (nullable = true)
 |-- Date: string (nullable = true)
 |-- Open: string (nullable = true)
 |-- High: string (nullable = true)
 |-- Low: string (nullable = true)
 |-- Close: string (nullable = true)
 |-- Volume: string (nullable = true)
 |-- Adj Close: string (nullable = true)
```

In [9]:
```
df1_1.select(col("summary"),
             col("Open").cast(DecimalType(10,2)),
             col("High").cast(DecimalType(10,2)),
             col("low").cast(DecimalType(10,2)),
             col("close").cast(DecimalType(10,2)),
             col("Volume").cast(DecimalType(10))
            ).show()
```

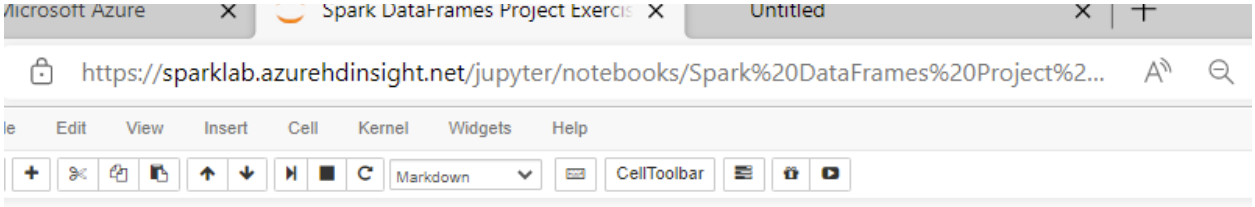
```
+-------+-------+-------+-------+-------+--------+
|summary|   Open|   High|    low|  close|  Volume|
+-------+-------+-------+-------+-------+--------+
|  count|1258.00|1258.00|1258.00|1258.00|    1258|
|   mean|  72.36|  72.84|  71.92|  72.39| 8222093|
| stddev|   6.77|   6.77|   6.74|   6.76| 4519781|
|    min|  56.39|  57.06|  56.30|  56.42| 2094900|
|    max|  90.80|  90.97|  89.25|  90.47|80898100|
+-------+-------+-------+-------+-------+--------+
```



Create a new dataframe with a column called HV Ratio that is the ratio of the High Price versus volume of stock traded for a day.

In [10]:
```
df2 = df1.withColumn("HV Ratio",df1['High']/df1['Volume'])
df2.select("HV Ratio").show()
```

```
+--------------------+
|            HV Ratio|
+--------------------+
|4.819714653321546E-6|
|6.290848613094555E-6|
|4.669412994783916E-6|
|7.367338463826307E-6|
|8.915604778943901E-6|
|8.644477436914568E-6|
|9.351828421515645E-6|
| 8.29141562102703E-6|
|7.712212102001476E-6|
|7.071764823529412E-6|
|1.015495466386981E-5|
|6.576354146362592...|
| 5.90145296180676E-6|
|8.547679455011844E-6|
|8.420709512685392E-6|
|1.041448341728929...|
|8.316075414862431E-6|
|9.721183814992126E-6|
|8.029436027707578E-6|
|6.307432259386365E-6|
+--------------------+
only showing top 20 rows
```



What day had the Peak High in Price?

In [19]:
```
from pyspark.sql.window import Window
from pyspark.sql.functions import  col, row_number ,count,rank, desc
windowSpec  = Window.partitionBy().orderBy(desc("High"))
top_rec=df1.withColumn("rank",rank().over(windowSpec)).head(1)

from datetime import datetime
top_date= top_rec[0]['Date']
dt_obj = datetime.strptime(top_date, '%Y-%S-%d')
dt_obj
```

```
datetime.datetime(2015, 1, 13, 0, 0, 1)
```

What is the mean of the Close column?

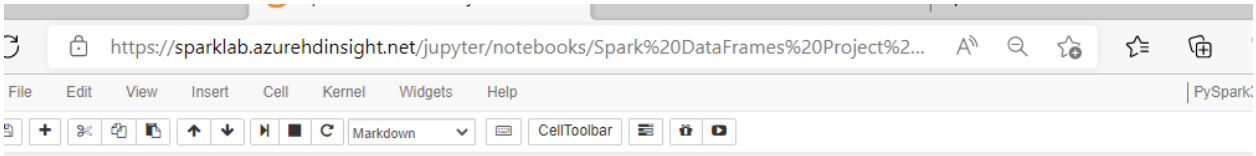
In [11]:
```
df1.select(avg("Close")).show()
```

```
+-----------------+
|       avg(Close)|
+-----------------+
|72.38844998012726|
+-----------------+
```

What is the max and min of the Volume column?

In [12]:
```
df1.select(max(col("Volume")),min(col("Volume"))).show()
```

```
+-----------+-----------+
|max(Volume)|min(Volume)|
+-----------+-----------+
|   80898100|    2094900|
+-----------+-----------+
```



How many days was the Close lower than 60 dollars?

In [13]:
```
df1.filter(df1["Close"]<60).count()
```

```
81
```

What percentage of the time was the High greater than 80 dollars ?

In other words, (Number of Days High>80)/(Total Days in the dataset)

In [14]:
```
df1.filter(df1["High"]>80).count()/df1.count()*100
```

```
9.141494435612083
```

What is the Pearson correlation between High and Volume?

Hint

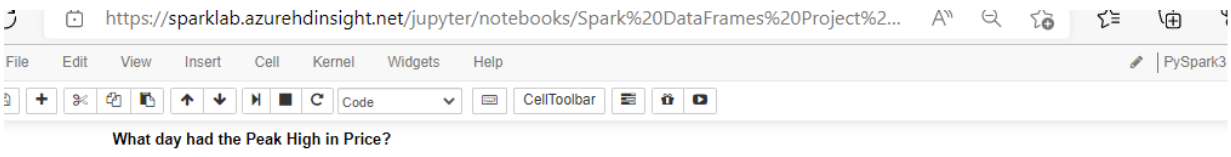
In [15]:
```
df1.stat.corr("High", "Volume")
```

```
-0.3384326061737161
```

What is the max High per year?

In [16]:
```
df2 = df1.withColumn("Year",year(df1['Date']))
df2.groupBy("Year").agg(max("High")).show()
```

```
+----+---------+
|Year|max(High)|
+----+---------+
```



What day had the Peak High in Price?

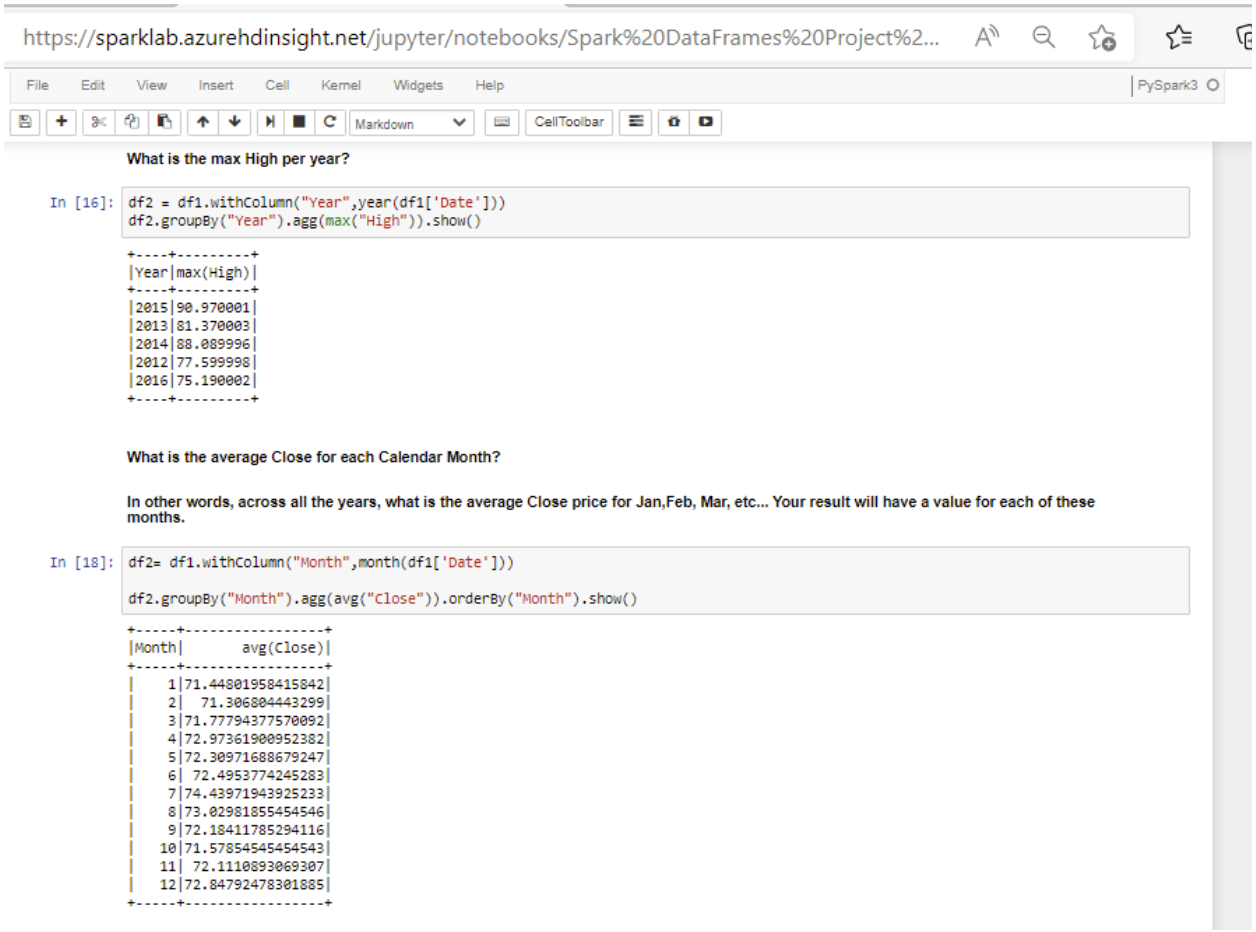
In [23]:
```
from pyspark.sql.window import Window
from pyspark.sql.functions import  col, row_number ,count,rank, desc
windowSpec  = Window.partitionBy().orderBy(desc("High"))
top_rec=df1.withColumn("rank",rank().over(windowSpec)).head(1)

from datetime import datetime
top_date= top_rec[0]['Date']
dt_obj = datetime.strptime(top_date, '%Y-%S-%d')
dt_obj
```

Command finished at 04-03-2022 12:46:26.933 -07:00, execution took 3s 307ms

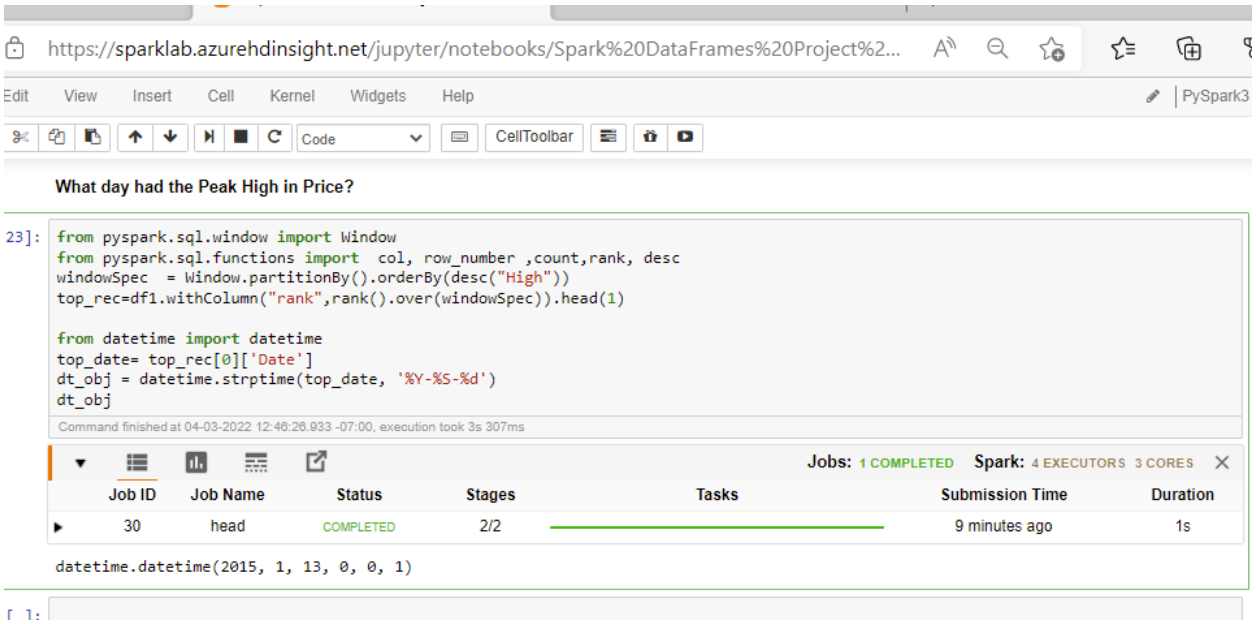
Jobs: 1 COMPLETED Spark: 4 EXECUTORS 3 CORES

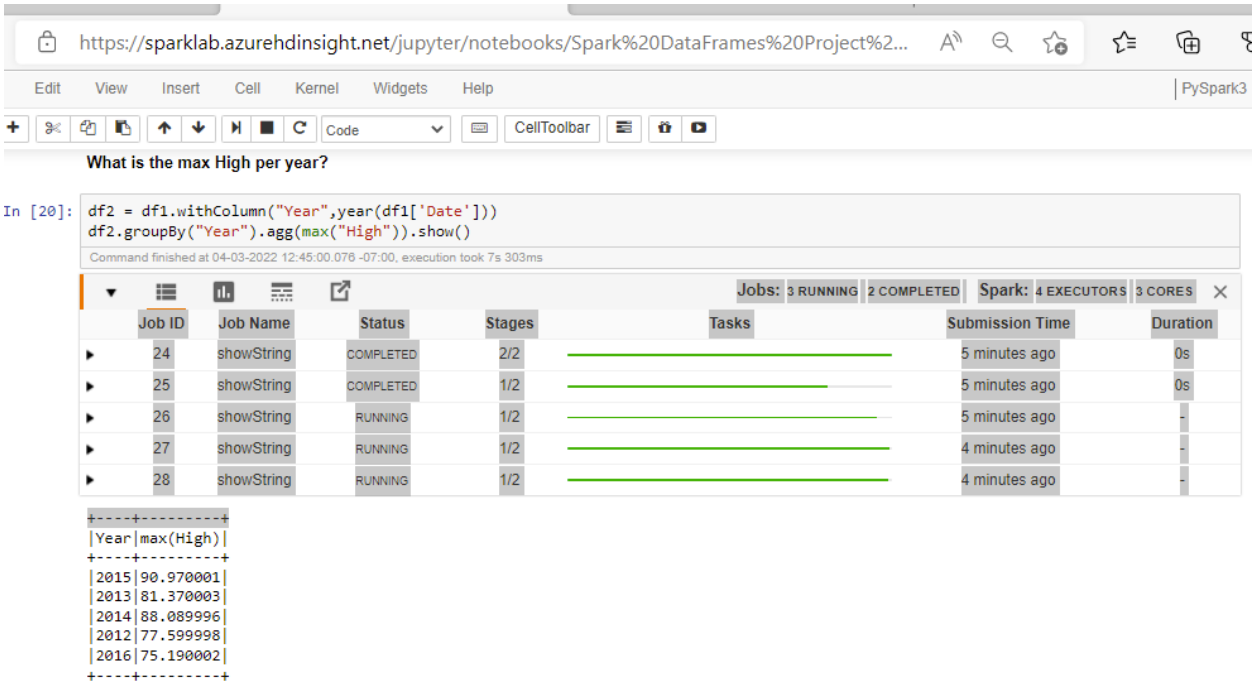
Job ID	Job Name	Status	Stages	Tasks	Submission Time	Duration
30	head	COMPLETED	2/2		a minute ago	1s

```
datetime.datetime(2015, 1, 13, 0, 0, 1)
```

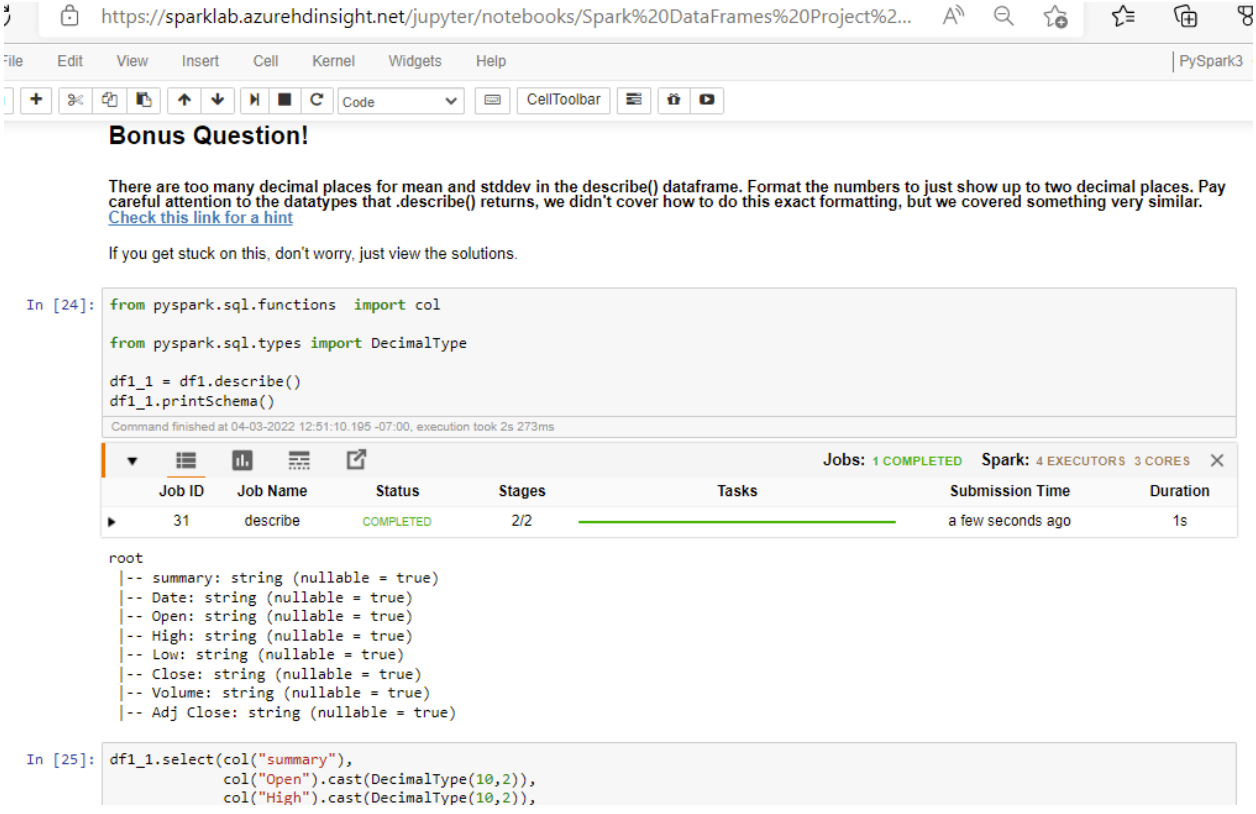


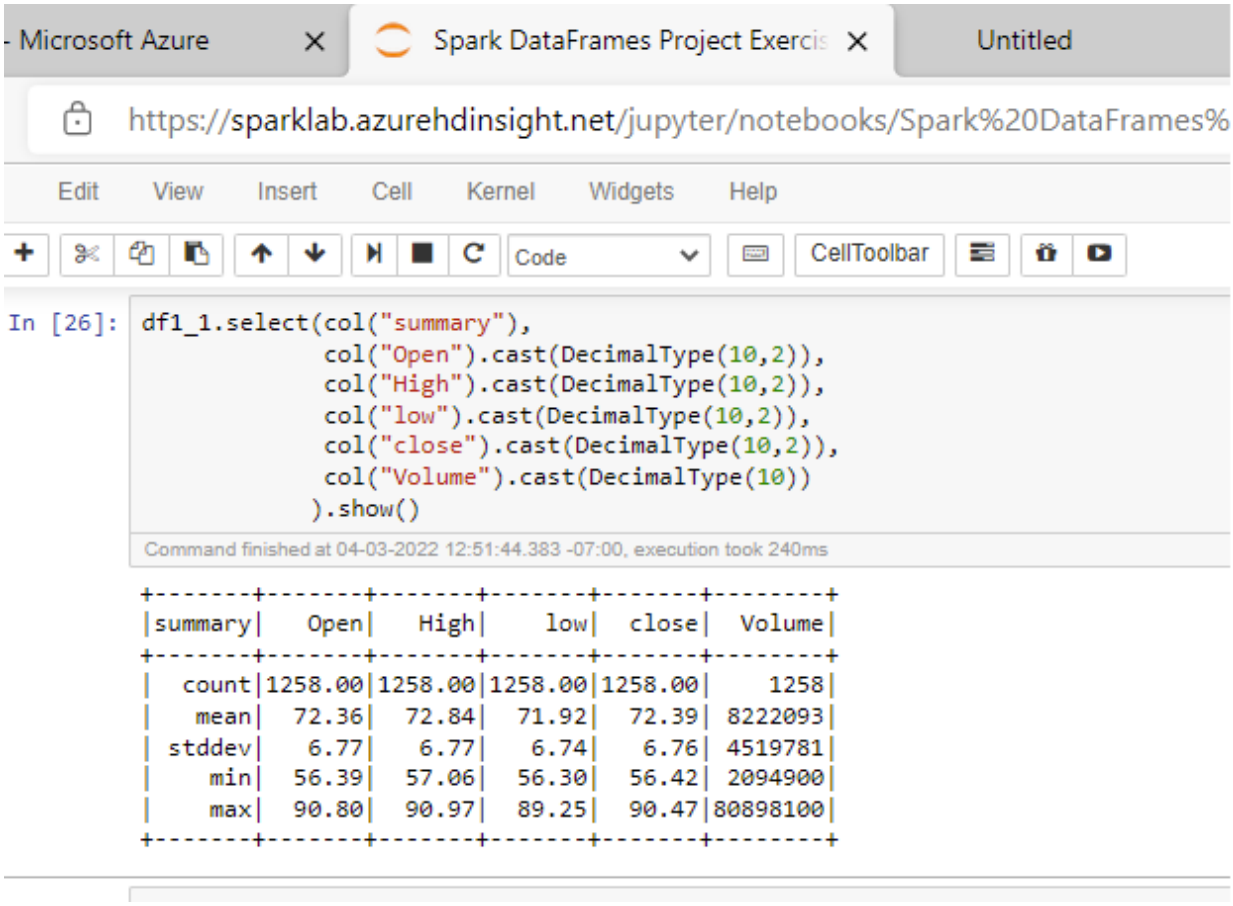
—---------- Additional screenshot for some questions with running information—----------





Bonus question




- Microsoft Azure
Spark DataFrames Project Exercis
Untitled
https://sparklab.azurehdinsight.net/jupyter/notebooks/Spark%20DataFrames%
Edit View Insert Cell Kernel Widgets Help
Code
CellToolbar
In [26]:
df1_1.select(col("summary"),
col("Open").cast(DecimalType(10,2)),
col("High").cast(DecimalType(10,2)),
col("low").cast(DecimalType(10,2)),
col("close").cast(DecimalType(10,2)),
col("Volume").cast(DecimalType(10))
).show()
Command finished at 04-03-2022 12:51:44.383 -07:00, execution took 240ms
+-------+-------+-------+-------+-------+--------+
|summary| Open| High| low| close| Volume|
+-------+-------+-------+-------+-------+--------+
count	1258.00	1258.00	1258.00	1258.00	1258
mean	72.36	72.84	71.92	72.39	8222093
stddev	6.77	6.77	6.74	6.76	4519781
min	56.39	57.06	56.30	56.42	2094900
max	90.80	90.97	89.25	90.47	80898100
+-------+-------+-------+-------+-------+--------+