In [1]:
```
pip install findspark
```

```
Requirement already satisfied: findspark in c:\users\user\anaconda3\lib\site-packages (2.0.0)Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
pip install pyspark
```

```
Requirement already satisfied: pyspark in c:\users\user\anaconda3\lib\site-packages (3.2.1)
Requirement already satisfied: py4j==0.10.9.3 in c:\users\user\anaconda3\lib\site-packages (from pyspark) (0.10.9.3)
Note: you may need to restart the kernel to use updated packages.
```

In [20]:
```
import findspark
findspark.init()
```

In [21]:
```
import pyspark
from pyspark.sql import SparkSession
spark=SparkSession.builder.getOrCreate()
SparkSession('''select "Spark" as hello''')
df.show()
```

```
+-----+
|hello|
+-----+
|Spark|
+-----+
```

In [22]:
```
#
spark
```

Out[22]:
SparkSession - in-memory

SparkContext

Spark UI

Version	v3.2.1
Master	local[*]
AppName	pyspark-shell

In [23]:
```
import findspark
findspark.init()
```

In [24]:
```
import pyspark
from pyspark.sql import SparkSession
spark=SparkSession.builder.appName("StudyApp").getOrCreate()
```

In [29]:
```
data=[(101,"Trupti",50000),(102,"Srushti",60000),(103,"Kajal",40000)]
df=spark.createDataFrame(data,["EmpId","EmpName","Salary"])
```

In [30]:
```
df
```

Out[30]:
```
DataFrame[EmpId: bigint, EmpName: string, Salary: bigint]
```

In [31]:
```
type(df)
```

Out[31]:
```
pyspark.sql.dataframe.DataFrame
```

In [32]:
```
df.show()
```

```
+-----+-------+------+
|EmpId|EmpName|Salary|
+-----+-------+------+
|  101| Trupti| 50000|
|  102|Srushti| 60000|
|  103|  Kajal| 40000|
+-----+-------+------+
```

In [35]:
```
df1=df.toPandas()
df1
```

Out[35]:

	EmpId	EmpName	Salary
0	101	Trupti	50000
1	102	Srushti	60000
2	103	Kajal	40000

In [36]:
```
type(df1)
```

Out[36]:
```
pandas.core.frame.DataFrame
```

In [37]:
```
df.columns
```

Out[37]:
```
['EmpId', 'EmpName', 'Salary']
```

In [38]:
```
df.count()
```

Out[38]:
```
3
```

In [67]:
```
df=spark.read.csv("D:\\items.csv",header=True)
df
```

Out[67]:
```
DataFrame[ItemId: string, ItemName: string, ItemCost: string, Supplier: string, Grade: string]
```

In [68]:
```
df.show()
```

```
+------+--------+--------+--------+-----+
|ItemId|ItemName|ItemCost|Supplier|Grade|
+------+--------+--------+--------+-----+
|     4|   Chock|   65.76|       X|    A|
|     5|  Pencil|   45.65|       Y|    B|
|     6|     Pen|   76.87|       X|    A|
|     7|  Duster|      54|       Y|    C|
|     8|    Book|   54.23|    null| null|
|     9|   Scale|   23.01|    null|    B|
|    10|    Tape|   43.09|       Z| null|
+------+--------+--------+--------+-----+
```

In [69]:
```
df.head()
```

Out[69]:
```
Row(ItemId='4', ItemName='Chock', ItemCost='65.76', Supplier='X', Grade='A')
```

In [70]:
```
pdf=df.head()
type(pdf)
```

Out[70]:
```
pyspark.sql.types.Row
```

In [71]:
```
df1=df.toPandas()
df1
```

Out[71]:

	ItemId	ItemName	ItemCost	Supplier	Grade
0	4	Chock	65.76	X	A
1	5	Pencil	45.65	Y	B
2	6	Pen	76.87	X	A
3	7	Duster	54	Y	C
4	8	Book	54.23	None	None
5	9	Scale	23.01	None	B
6	10	Tape	43.09	Z	None

In [72]:
```
# Aggregation Tasks
df.groupBy("Grade").agg({"Grade": 'count'}).show()
```

```
+-----+-------------+
|Grade|count(Grade)|
+-----+-------------+
| null|            0|
|    B|            2|
|    C|            1|
|    A|            2|
+-----+-------------+
```

In [58]:
```
df=spark.read.csv("D:\\loan prediction dataset in banking.csv",header=True)
df
```

Out[58]:
```
DataFrame[Loan_ID: string, Gender: string, Married: string, Dependents: string, Education: string, Self_Employed: string, ApplicantIncome: string, CoapplicantIncome: string, LoanAmount: string, Loan_Amount_Term: string, Credit_History: string, Property_Area: string, Loan_Status: string]
```

In [59]:
```
df.show()
```

```
+-------+------+-------+----------+--------------+-------------+----------------+-----------------+----------+----------------+--------------+-------------+-----------+
| Loan_ID|Gender|Married|Dependents|     Education|Self_Employed|ApplicantIncome|CoapplicantIncome|LoanAmount|Loan_Amount_Term|Credit_History|Property_Area|Loan_Status|
+-------+------+-------+----------+--------------+-------------+----------------+-----------------+----------+----------------+--------------+-------------+-----------+
|LP001002|  Male|     No|         0|      Graduate|           No|            5849|              0.0|      null|           360.0|           1.0|        Urban|          Y|
|LP001003|  Male|    Yes|         1|      Graduate|           No|            4583|           1508.0|       128|           360.0|           1.0|        Rural|          N|
|LP001005|  Male|    Yes|         0|      Graduate|          Yes|            3000|              0.0|        66|           360.0|           1.0|        Urban|          Y|
|LP001006|  Male|    Yes|         0|Not Graduate|           No|            2583|           2358.0|       120|           360.0|           1.0|        Urban|          Y|
|LP001008|  Male|     No|         0|      Graduate|           No|            6000|              0.0|       141|           360.0|           1.0|        Urban|          Y|
|LP001011|  Male|    Yes|         2|      Graduate|          Yes|            5417|           4196.0|       267|           360.0|           1.0|        Urban|          Y|
|LP001013|  Male|    Yes|         0|Not Graduate|           No|            2333|           1516.0|        95|           360.0|           1.0|        Urban|          Y|
|LP001014|  Male|    Yes|        3+|      Graduate|           No|            3036|           2504.0|       158|           360.0|           0.0|    Semiurban|          N|
|LP001018|  Male|    Yes|         2|      Graduate|           No|            4006|           1526.0|       168|           360.0|           1.0|        Urban|          Y|
|LP001020|  Male|    Yes|         1|      Graduate|           No|            12841|          10968.0|       349|           360.0|           1.0|    Semiurban|          N|
|LP001024|  Male|    Yes|         2|      Graduate|           No|            3200|           700.0|        70|           360.0|           1.0|        Urban|          Y|
|LP001027|  Male|    Yes|         2|      Graduate|          null|            2500|          1840.0|       109|           360.0|           1.0|        Urban|          Y|
|LP001028|  Male|    Yes|         2|      Graduate|           No|            3073|          8106.0|       200|           360.0|           1.0|        Urban|          Y|
|LP001029|  Male|     No|         0|      Graduate|           No|            1853|          2840.0|       114|           360.0|           1.0|        Rural|          N|
|LP001030|  Male|    Yes|         2|      Graduate|           No|            1299|          1086.0|        17|           120.0|           1.0|        Urban|          Y|
|LP001032|  Male|     No|         0|      Graduate|           No|            4950|              0.0|       125|           360.0|           1.0|        Urban|          Y|
|LP001034|  Male|     No|         1|Not Graduate|           No|            3596|              0.0|       100|           240.0|          null|        Urban|          Y|
|LP001036|Female|     No|         0|      Graduate|           No|            3510|              0.0|        76|           360.0|           0.0|        Urban|          N|
|LP001038|  Male|    Yes|         0|Not Graduate|           No|            4887|              0.0|       133|           360.0|           1.0|        Rural|          N|
|LP001041|  Male|    Yes|         0|      Graduate|          null|            2600|           3500.0|       115|           null|           1.0|        Urban|          Y|
+-------+------+-------+----------+--------------+-------------+----------------+-----------------+----------+----------------+--------------+-------------+-----------+
only showing top 20 rows
```

In [60]:
```
df.head()
```

Out[60]:
```
Row(Loan_ID='LP001002', Gender='Male', Married='No', Dependents='0', Education='Graduate', Self_Employed='No', ApplicantIncome='5849', CoapplicantIncome='0', LoanAmount=None, Loan_Amount_Term='360', Credit_History='1', Property_Area='Urban', Loan_Status='Y')
```

In [62]:
```
df1=df.toPandas()
df1
```

Out[62]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0	None	360	1	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	None	1508	128	360	1	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	None	0	66	360	1	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	None	2358	120	360	1	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
...	...	...	...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Female	No	0	Graduate	No	2900	0	71	360	1	Rural	Y
610	LP002979	Male	Yes	3+	Graduate	No	4106	0	40	180	1	Rural	Y
611	LP002983	Male	Yes	1	Graduate	No	8072	240	253	360	1	Urban	Y
612	LP002984	Male	Yes	2	Graduate	No	7583	0	187	360	1	Urban	Y
613	LP002990	Female	No	0	Graduate	Yes	4583	0	133	360	0	Semiurban	N

614 rows × 13 columns

In [64]:
```
df.groupBy("Married").agg({"Married": 'count'}).show()
```

```
+-------+--------------+
|Married|count(Married)|
+-------+--------------+
|   null|             8|
|     No|           213|
|    Yes|           398|
+-------+--------------+
```

In [77]:
```
df1=spark.read.csv("D:\\items.csv")
df1.show()
```

```
+-----+--------+--------+--------+-----+
|  _c0|     _c1|     _c2|     _c3|  _c4|
+-----+--------+--------+--------+-----+
|ItemId|ItemName|ItemCost|Supplier|Grade|
|    4|   Chock|   65.76|       X|    A|
|    5|  Pencil|   45.65|       Y|    B|
|    6|     Pen|   76.87|       X|    A|
|    7|  Duster|      54|       Y|    C|
|    8|    Book|   54.23|    null| null|
|    9|   Scale|   23.01|    null|    B|
|   10|    Tape|   43.09|       Z| null|
+-----+--------+--------+--------+-----+
```

In [84]:
```
df1.groupBy("_c3").agg({"_c3": 'count'}).show()
```

```
+--------+----------+
|     _c3|count(_c3)|
+--------+----------+
|Supplier|         1|
|    null|         2|
|       X|         3|
|       Z|         1|
|       Y|         2|
+--------+----------+
```

In [83]:
```
df.groupBy("Supplier").agg({"ItemCost":'min'}).show()
df.groupBy("Supplier").agg({"ItemCost":'max'}).show()
```

```
+--------+-------------+
|Supplier|min(ItemCost)|
+--------+-------------+
|    null|        23.01|
|       X|        65.76|
|       Y|        45.65|
|       Z|        43.09|
+--------+-------------+

+--------+-------------+
|Supplier|max(ItemCost)|
+--------+-------------+
|    null|        54.23|
|       X|        76.87|
|       Y|           54|
|       Z|        43.09|
+--------+-------------+
```

In [92]:
```
from pyspark.sql import functions as f
df.groupBy("ItemId").agg(f.max("ItemCost"),f.min("ItemCost"),
                         f.avg("ItemCost"),f.sum("ItemCost")).show()
```

```
+------+--------------+--------------+------------------+--------------+
|ItemId|max(ItemCost)|min(ItemCost)|avg(ItemCost)|sum(ItemCost)|
+------+--------------+--------------+------------------+--------------+
|    10|         43.09|         43.09|             43.09|         43.09|
|     4|         65.76|         65.76|             65.76|         65.76|
|     5|         45.65|         45.65|             45.65|         45.65|
|     6|         76.87|         76.87|             76.87|         76.87|
|     7|            54|            54|              54.0|          54.0|
|     8|         54.23|         54.23|             54.23|         54.23|
|     9|         23.01|         23.01|             23.01|         23.01|
+------+--------------+--------------+------------------+--------------+
```

In [93]:
```
df.rdd.id()
```

Out[93]:
```
291
```

In [94]:
```
df1=df
df1.rdd.id()
```

Out[94]:
```
291
```

In [95]:
```
df2=df.withColumn("Total",df["ItemCost"]+df["Supplier"])
df2.rdd.id()
```

Out[95]:
```
297
```

In [96]:
```
df2=df.withColumn("Amount",df["ItemCost"]+2000)
```

In [100]:
```
rowlist=df2.collect()
print(rowlist)
```

```
[Row(ItemId='4', ItemName='Chock', ItemCost='65.76', Supplier='X', Grade='A', Amount=2065.76), Row(ItemId='5', ItemName='Pencil', ItemCost='45.65', Supplier='Y', Grade='B', Amount=2045.65), Row(ItemId='6', ItemName='Pen', ItemCost='76.87', Supplier='X', Grade='A', Amount=2076.87), Row(ItemId='7', ItemName='Duster', ItemCost='54', Supplier='Y', Grade='C', Amount=2054.0), Row(ItemId='8', ItemName='Book', ItemCost='54.23', Supplier=None, Grade=None, Amount=2054.23), Row(ItemId='9', ItemName='Scale', ItemCost='23.01', Supplier=None, Grade='B', Amount=2023.01), Row(ItemId='10', ItemName='Tape', ItemCost='43.09', Supplier='Z', Grade=None, Amount=2043.09)]
```

In [ ]:
```

```