In [1]:

```
from pyspark.sql import SparkSession
spark=SparkSession.builder.appName("DfApp").getOrCreate()
```

In [2]:

```
pdf=spark.read.option("header","true").csv('D:\employees.csv',inferSchema=True)
pdf.show()
```

```
+----------+------+----------+---------------+------+-------+-----------------+--------------------+
|First Name|Gender|Start Date|Last Login Time|Salary|Bonus %|Senior Management|                Team|
+----------+------+----------+---------------+------+-------+-----------------+--------------------+
|   Douglas|  Male|  8/6/1993|       12:42 PM| 97308|  6.945|             true|           Marketing|
|    Thomas|  Male| 3/31/1996|        6:53 AM| 61933|   4.17|             true|                null|
|     Maria|Female| 4/23/1993|       11:17 AM|  null| 11.858|            false|             Finance|
|     Jerry|  Male|  3/4/2005|        1:00 PM|138705|   null|             true|             Finance|
|     Larry|  Male| 1/24/1998|        4:47 PM|  null|  1.389|             true|     Client Services|
|    Dennis|  Male| 4/18/1987|        1:35 AM|115163| 10.125|            false|               Legal|
|      Ruby|Female|      null|           null| 65476| 10.012|             true|             Product|
|      null|Female| 7/20/2015|       10:43 AM|  null|   null|             null|             Finance|
|    Angela|Female|      null|        6:29 AM| 95570|   null|             true|         Engineering|
|   Frances|Female|      null|        6:51 AM|139852|  7.524|             true|Business Development|
|    Louise|Female| 8/12/1980|        9:01 AM| 63241| 15.132|             true|                null|
|     Julie|Female|10/26/1997|        3:19 PM|102508| 12.637|             true|               Legal|
|   Brandon|  Male| 12/1/1980|        1:08 AM|112807| 17.492|             true|     Human Resources|
|      Gary|  Male|      null|       11:40 PM|109831|  5.831|            false|               Sales|
|  Kimberly|Female| 1/14/1999|        7:13 AM| 41426| 14.543|             true|             Finance|
|   Lillian|Female|  6/5/2016|        6:09 AM| 59414|  1.256|            false|             Product|
|    Jeremy|  Male| 9/21/2010|        5:56 AM| 90370|  7.369|            false|     Human Resources|
|     Shawn|  Male| 12/7/1986|        7:45 PM|111737|  6.414|            false|             Product|
|     Diana|Female|      null|       10:27 AM|132940| 19.082|            false|     Client Services|
|     Donna|Female| 7/22/2010|        3:48 AM| 81014|  1.894|            false|             Product|
+----------+------+----------+---------------+------+-------+-----------------+--------------------+
only showing top 20 rows
```

In [5]:

```
pdf.printSchema()
```

```
root
 |-- First Name: string (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Start Date: string (nullable = true)
 |-- Last Login Time: string (nullable = true)
 |-- Salary: integer (nullable = true)
 |-- Bonus %: double (nullable = true)
 |-- Senior Management: boolean (nullable = true)
 |-- Team: string (nullable = true)
```

In [6]:

```
from pyspark.sql.functions import col
pdf1=pdf.select(col('First Name').alias('FirstName'),
                col('Gender').alias('Gen'),
                col('Start Date').alias('StartDate'),
                col('Last Login Time').alias('LastLoginTime'),
                col('Salary').alias('Salary'),
                col('Bonus %').alias('BonusPer'),
                col('Senior Management').alias('SM'),
                col('Team').alias('Team'))
```

In [7]:

```
pdf1.show()
```

```
+---------+------+----------+-------------+------+--------+-----+--------------------+
|FirstName|   Gen| StartDate|LastLoginTime|Salary|BonusPer|   SM|                Team|
+---------+------+----------+-------------+------+--------+-----+--------------------+
|  Douglas|  Male|  8/6/1993|     12:42 PM| 97308|   6.945| true|           Marketing|
|   Thomas|  Male| 3/31/1996|      6:53 AM| 61933|    4.17| true|                null|
|    Maria|Female| 4/23/1993|     11:17 AM|  null|  11.858|false|             Finance|
|    Jerry|  Male|  3/4/2005|      1:00 PM|138705|    null| true|             Finance|
|    Larry|  Male| 1/24/1998|      4:47 PM|  null|   1.389| true|     Client Services|
|   Dennis|  Male| 4/18/1987|      1:35 AM|115163|  10.125|false|               Legal|
|     Ruby|Female|      null|         null| 65476|  10.012| true|             Product|
|     null|Female| 7/20/2015|     10:43 AM|  null|    null| null|             Finance|
|   Angela|Female|      null|      6:29 AM| 95570|    null| true|         Engineering|
|  Frances|Female|      null|      6:51 AM|139852|   7.524| true|Business Development|
|   Louise|Female| 8/12/1980|      9:01 AM| 63241|  15.132| true|                null|
|    Julie|Female|10/26/1997|      3:19 PM|102508|  12.637| true|               Legal|
|  Brandon|  Male| 12/1/1980|      1:08 AM|112807|  17.492| true|     Human Resources|
|     Gary|  Male|      null|     11:40 PM|109831|   5.831|false|               Sales|
| Kimberly|Female| 1/14/1999|      7:13 AM| 41426|  14.543| true|             Finance|
|  Lillian|Female|  6/5/2016|      6:09 AM| 59414|   1.256|false|             Product|
|   Jeremy|  Male| 9/21/2010|      5:56 AM| 90370|   7.369|false|     Human Resources|
|    Shawn|  Male| 12/7/1986|      7:45 PM|111737|   6.414|false|             Product|
|    Diana|Female|      null|     10:27 AM|132940|  19.082|false|     Client Services|
|    Donna|Female| 7/22/2010|      3:48 AM| 81014|   1.894|false|             Product|
+---------+------+----------+-------------+------+--------+-----+--------------------+
only showing top 20 rows
```

In [8]:

```
pdf1.createOrReplaceTempView('tempview')
```

In [9]:

```
spark.sql("select * from tempview ").toPandas()
```

Out[9]:

	FirstName	Gen	StartDate	LastLoginTime	Salary	BonusPer	SM	Team
0	Douglas	Male	8/6/1993	12:42 PM	97308.0	6.945	True	Marketing
1	Thomas	Male	3/31/1996	6:53 AM	61933.0	4.170	True	None
2	Maria	Female	4/23/1993	11:17 AM	NaN	11.858	False	Finance
3	Jerry	Male	3/4/2005	1:00 PM	138705.0	NaN	True	Finance
4	Larry	Male	1/24/1998	4:47 PM	NaN	1.389	True	Client Services
...	...	...	...	...	...	...	...	...
995	Henry	None	11/23/2014	6:09 AM	132483.0	16.655	False	Distribution
996	Phillip	Male	1/31/1984	6:30 AM	42392.0	19.675	False	Finance
997	Russell	Male	5/20/2013	12:39 PM	96914.0	1.421	False	Product
998	Larry	Male	4/20/2013	4:45 PM	60500.0	11.985	False	Business Development
999	Albert	Male	5/15/2012	6:24 PM	129949.0	10.169	True	Sales

1000 rows × 8 columns

In [10]:

```
res=spark.sql("select Team,sum(Salary) as Total_Sal from tempview GROUP BY Team")
res.toPandas()
```

Out[10]:

	Team	Total_Sal
0	Sales	8664303
1	Engineering	8366157
2	None	4063842
3	Business Development	9169238
4	Finance	8840590
5	Client Services	9250785
6	Distribution	7965042
7	Legal	7620468
8	Marketing	8862688
9	Product	8423223
10	Human Resources	8275952

In [11]:

```
pdf1.selectExpr("Team","Salary").filter("length(Team)==7").groupBy('Team').sum('Salary').toPandas()
```

Out[11]:

	Team	sum(Salary)
0	Finance	8840590
1	Product	8423223

In [12]:

```
spark.sql("select Team,sum(Salary) from tempview where length(Team)==7 GROUP BY Team").show()
```

```
+-------+-----------+
|   Team|sum(Salary)|
+-------+-----------+
|Finance|    8840590|
|Product|    8423223|
+-------+-----------+
```

In [85]:

```
pdf1.printSchema()
```

```
root
 |-- FirstName: string (nullable = true)
 |-- Gen: string (nullable = true)
 |-- StartDate: string (nullable = true)
 |-- LastLoginTime: string (nullable = true)
 |-- Salary: integer (nullable = true)
 |-- BonusPer: double (nullable = true)
 |-- SM: boolean (nullable = true)
 |-- Team: string (nullable = true)
```

In [24]:

```
from pyspark.ml.feature import SQLTransformer
```

In [25]:

```
sqltrans=SQLTransformer(statement="select FirstName,Team,Salary,BonusPer from __THIS__")
sqltrans.transform(pdf1).show(5)
```

```
+---------+---------------+------+--------+
|FirstName|           Team|Salary|BonusPer|
+---------+---------------+------+--------+
|  Douglas|      Marketing| 97308|   6.945|
|   Thomas|           null| 61933|    4.17|
|    Maria|        Finance|  null|  11.858|
|    Jerry|        Finance|138705|    null|
|    Larry|Client Services|  null|   1.389|
+---------+---------------+------+--------+
only showing top 5 rows
```

In [38]:

```
sqltrans=SQLTransformer(statement="select FirstName,Team,Salary,BonusPer from __THIS__ where FirstName=='Jerry'")
sqltrans.transform(pdf1).show()
```

```
+---------+--------------------+------+--------+
|FirstName|                Team|Salary|BonusPer|
+---------+--------------------+------+--------+
|    Jerry|             Finance|138705|    null|
|    Jerry|     Client Services| 95734|  19.096|
|    Jerry|     Client Services|140810|   9.177|
|    Jerry|Business Development|121357|  18.845|
|    Jerry|     Client Services| 98393|  11.393|
|    Jerry|             Finance|140850|  18.855|
+---------+--------------------+------+--------+
```

In [26]:

```
type(sqltrans)
```

Out[26]: pyspark.ml.feature.SQLTransformer

In [56]:

```
pdf1.withColumn("Gen",expr("CASE WHEN Gen='Male' THEN 'M'"+
                           "WHEN Gen='Female' THEN 'F' END")).show()
```

```
+---------+---+----------+-------------+------+--------+-----+--------------------+
|FirstName|Gen| StartDate|LastLoginTime|Salary|BonusPer|   SM|                Team|
+---------+---+----------+-------------+------+--------+-----+--------------------+
|  Douglas|  M|  8/6/1993|     12:42 PM| 97308|   6.945| true|           Marketing|
|   Thomas|  M| 3/31/1996|      6:53 AM| 61933|    4.17| true|                null|
|    Maria|  F| 4/23/1993|     11:17 AM|  null|  11.858|false|             Finance|
|    Jerry|  M|  3/4/2005|      1:00 PM|138705|    null| true|             Finance|
|    Larry|  M| 1/24/1998|      4:47 PM|  null|   1.389| true|     Client Services|
|   Dennis|  M| 4/18/1987|      1:35 AM|115163|  10.125|false|               Legal|
|     Ruby|  F|      null|         null| 65476|  10.012| true|             Product|
|     null|  F| 7/20/2015|     10:43 AM|  null|    null| null|             Finance|
|   Angela|  F|      null|      6:29 AM| 95570|    null| true|         Engineering|
|  Frances|  F|      null|      6:51 AM|139852|   7.524| true|Business Development|
|   Louise|  F| 8/12/1980|      9:01 AM| 63241|  15.132| true|                null|
|    Julie|  F|10/26/1997|      3:19 PM|102508|  12.637| true|               Legal|
|  Brandon|  M| 12/1/1980|      1:08 AM|112807|  17.492| true|     Human Resources|
|     Gary|  M|      null|     11:40 PM|109831|   5.831|false|               Sales|
| Kimberly|  F| 1/14/1999|      7:13 AM| 41426|  14.543| true|             Finance|
|  Lillian|  F|  6/5/2016|      6:09 AM| 59414|   1.256|false|             Product|
|   Jeremy|  M| 9/21/2010|      5:56 AM| 90370|   7.369|false|     Human Resources|
|    Shawn|  M| 12/7/1986|      7:45 PM|111737|   6.414|false|             Product|
|    Diana|  F|      null|     10:27 AM|132940|  19.082|false|     Client Services|
|    Donna|  F| 7/22/2010|      3:48 AM| 81014|   1.894|false|             Product|
+---------+---+----------+-------------+------+--------+-----+--------------------+
only showing top 20 rows
```

In [74]:

```
pdf1.select('FirstName','Salary','BonusPer',expr('round(BonusPer,1) as roundedup')).show()
```

```
+---------+------+--------+---------+
|FirstName|Salary|BonusPer|roundedup|
+---------+------+--------+---------+
|  Douglas| 97308|   6.945|      6.9|
|   Thomas| 61933|    4.17|      4.2|
|    Maria|  null|  11.858|     11.9|
|    Jerry|138705|    null|     null|
|    Larry|  null|   1.389|      1.4|
|   Dennis|115163|  10.125|     10.1|
|     Ruby| 65476|  10.012|     10.0|
|     null|  null|    null|     null|
|   Angela| 95570|    null|     null|
|  Frances|139852|   7.524|      7.5|
|   Louise| 63241|  15.132|     15.1|
|    Julie|102508|  12.637|     12.6|
|  Brandon|112807|  17.492|     17.5|
|     Gary|109831|   5.831|      5.8|
| Kimberly| 41426|  14.543|     14.5|
|  Lillian| 59414|   1.256|      1.3|
|   Jeremy| 90370|   7.369|      7.4|
|    Shawn|111737|   6.414|      6.4|
|    Diana|132940|  19.082|     19.1|
|    Donna| 81014|   1.894|      1.9|
+---------+------+--------+---------+
only showing top 20 rows
```

In [77]:

```
pdf1.selectExpr("FirstName","round(BonusPer,2)").filter('FirstName=="Jerry"').toPandas()
```

Out[77]:

	FirstName	round(BonusPer, 2)
0	Jerry	NaN
1	Jerry	19.10
2	Jerry	9.18
3	Jerry	18.85
4	Jerry	11.39
5	Jerry	18.86

In [28]:

```
pdf=spark.read.option("header","true").csv('D:\items.csv',inferSchema=True)
pdf.show()
```

```
+------+--------+--------+-------+----------+
|ItemId|ItemName|ItemCost|ItemQty|SupplierId|
+------+--------+--------+-------+----------+
|     4|   Chock|   65.76|     23|         X|
|     5|  Pencil|   45.65|     34|         Y|
|     6|     Pen|   76.87|     32|         X|
|     7|  Duster|    54.0|     10|         Z|
|     8|    null|    23.0|     45|         Y|
|     9|    book|    53.0|     25|      null|
+------+--------+--------+-------+----------+
```

In [50]:

```
pdf.printSchema()
```

```
root
 |-- ItemId: integer (nullable = true)
 |-- ItemName: string (nullable = true)
 |-- ItemCost: double (nullable = true)
 |-- ItemQty: integer (nullable = true)
 |-- SupplierId: string (nullable = true)
```

In [51]:

```
pdf.createOrReplaceTempView('tempview')
```

In [52]:

```
spark.sql("select * from tempview ").toPandas()
```

Out[52]:

	ItemId	ItemName	ItemCost	ItemQty	SupplierId
0	4	Chock	65.76	23	X
1	5	Pencil	45.65	34	Y
2	6	Pen	76.87	32	X
3	7	Duster	54.00	10	Z
4	8	None	23.00	45	Y
5	9	book	53.00	25	None

In [29]:

```
from pyspark.ml.feature import SQLTransformer
```

In [35]:

```
sqltrans=SQLTransformer(statement="select ItemName, ItemCost, ItemQty from __THIS__")
sqltrans.transform(pdf).show()
```

```
+--------+--------+-------+
|ItemName|ItemCost|ItemQty|
+--------+--------+-------+
|   Chock|   65.76|     23|
|  Pencil|   45.65|     34|
|     Pen|   76.87|     32|
|  Duster|    54.0|     10|
|    null|    23.0|     45|
|    book|    53.0|     25|
+--------+--------+-------+
```

In [54]:

```
sqltrans=SQLTransformer(statement="select max(ItemCost), min(ItemCost), sum(ItemCost) from __THIS__ where ItemName='Duster'")
sqltrans.transform(pdf).show()
```

```
+-------------+-------------+-------------+
|max(ItemCost)|min(ItemCost)|sum(ItemCost)|
+-------------+-------------+-------------+
|         54.0|         54.0|         54.0|
+-------------+-------------+-------------+
```

In [36]:

```
# exp()
from pyspark.sql.functions import expr
```

In [40]:

```
pdf.withColumn("Increment",expr("ItemCost+50")).show()
```

```
+------+--------+--------+-------+----------+---------+
|ItemId|ItemName|ItemCost|ItemQty|SupplierId|Increment|
+------+--------+--------+-------+----------+---------+
|     4|   Chock|   65.76|     23|         X|   115.76|
|     5|  Pencil|   45.65|     34|         Y|    95.65|
|     6|     Pen|   76.87|     32|         X|   126.87|
|     7|  Duster|    54.0|     10|         Z|    104.0|
|     8|    null|    23.0|     45|         Y|     73.0|
|     9|    book|    53.0|     25|      null|    103.0|
+------+--------+--------+-------+----------+---------+
```

In [45]:

```
pdf.selectExpr('ItemName','ItemQty','ItemQty+100').show()
```

```
+--------+-------+---------------+
|ItemName|ItemQty|(ItemQty + 100)|
+--------+-------+---------------+
|   Chock|     23|            123|
|  Pencil|     34|            134|
|     Pen|     32|            132|
|  Duster|     10|            110|
|    null|     45|            145|
|    book|     25|            125|
+--------+-------+---------------+
```

In [61]:

```
pdf.withColumn("ItemCost",expr("CASE WHEN ItemCost>60 THEN 'A'" +
                               "WHEN ItemCost>50 THEN 'B'" +
                               "WHEN ItemCost>40 THEN 'C' END")).show()
```

```
+------+--------+--------+-------+----------+
|ItemId|ItemName|ItemCost|ItemQty|SupplierId|
+------+--------+--------+-------+----------+
|     4|   Chock|       A|     23|         X|
|     5|  Pencil|       C|     34|         Y|
|     6|     Pen|       A|     32|         X|
|     7|  Duster|       B|     10|         Z|
|     8|    null|    null|     45|         Y|
|     9|    book|       B|     25|      null|
+------+--------+--------+-------+----------+
```

In [64]:

```
pdf.withColumn("ItemCost",expr("CASE WHEN ItemCost>=60 THEN 'A'" +
                               "WHEN ItemCost>=50 THEN 'B' Else 'C' END")).show()
```

```
+------+--------+--------+-------+----------+
|ItemId|ItemName|ItemCost|ItemQty|SupplierId|
+------+--------+--------+-------+----------+
|     4|   Chock|       A|     23|         X|
|     5|  Pencil|       C|     34|         Y|
|     6|     Pen|       A|     32|         X|
|     7|  Duster|       B|     10|         Z|
|     8|    null|       C|     45|         Y|
|     9|    book|       B|     25|      null|
+------+--------+--------+-------+----------+
```

In [67]:

```
pdf.select('ItemName','ItemCost',expr('round(ItemCost,1) as roundedup')).show()
```

```
+--------+--------+---------+
|ItemName|ItemCost|roundedup|
+--------+--------+---------+
|   Chock|   65.76|     65.8|
|  Pencil|   45.65|     45.7|
|     Pen|   76.87|     76.9|
|  Duster|    54.0|     54.0|
|    null|    23.0|     23.0|
|    book|    53.0|     53.0|
+--------+--------+---------+
```

In [80]:

```
pdf.selectExpr("ItemName","round(ItemCost,1)").filter('ItemName=="Pen"').toPandas()
```

Out[80]:

	ItemName	round(ItemCost, 1)
0	Pen	76.9

In [ ]: