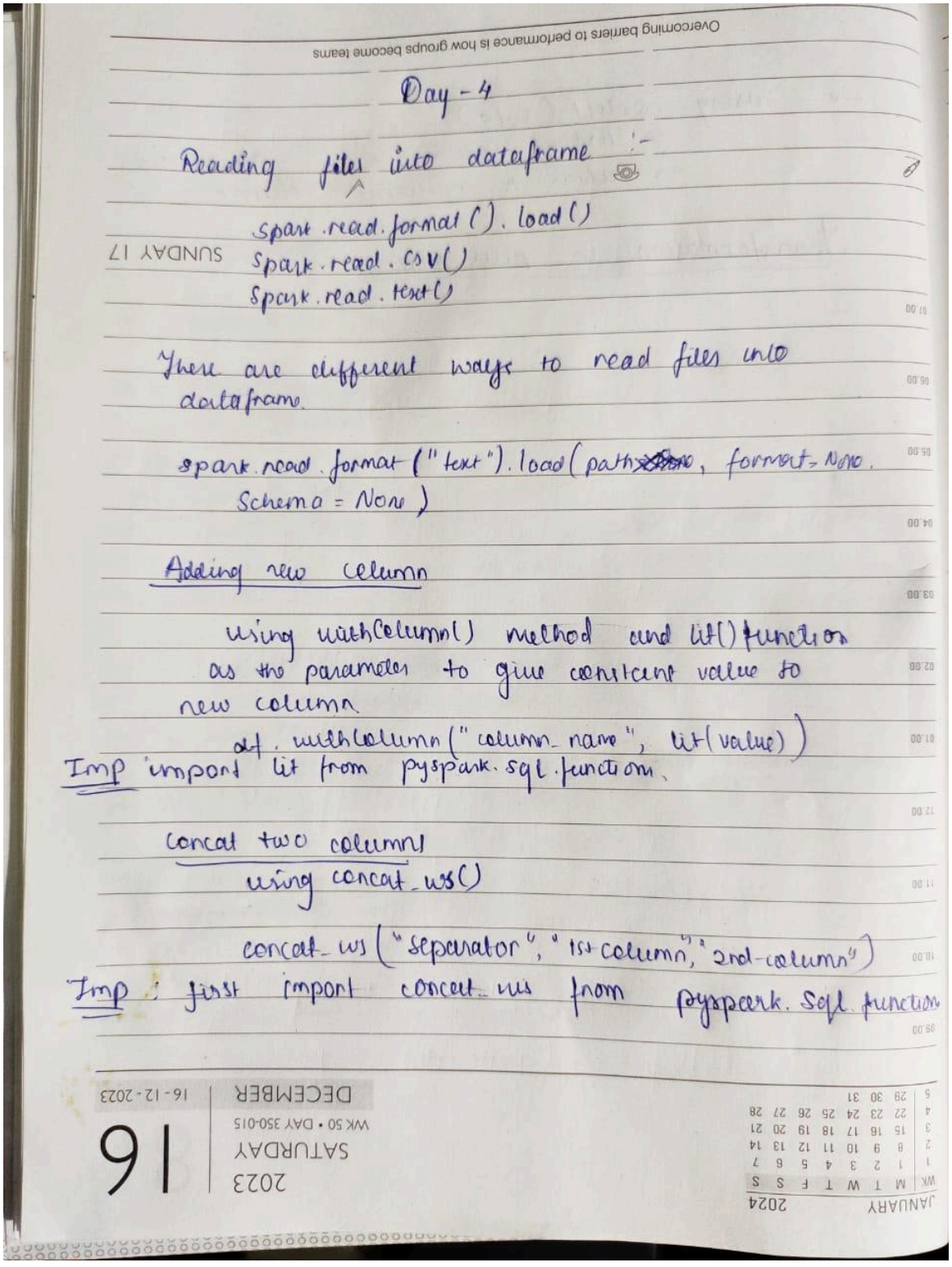
DAY-14 (Spark Assignment - 4)
Mitushi Vishwakarma



Overcoming barriers to performance is how groups become teams

Day - 4

Reading files into dataframe :-

spark.read.format().load()
spark.read.csv()
spark.read.text()

There are different ways to read files into dataframe.

spark.read.format("text").load(path, format=None, schema = None)

Adding new column

using withColumn() method and lit() function as the parameter to give constant value to new column.

df.withColumn("column_name", lit(value))

Imp import lit from pyspark.sql.functions.

Concat two columns
using concat_ws()

concat_ws("separator", "1st column", "2nd column")

Imp : first import concat_ws from pyspark.sql.functions

SUNDAY 17

DECEMBER 16-12-2023
SATURDAY 2023
WK 50 • DAY 350-015
16

JANUARY 2024

WK	M	T	W	T	F	S	S
1	1	2	3	4	5	6	7
2	8	9	10	11	12	13	14
3	15	16	17	18	19	20	21
4	22	23	24	25	26	27	28
5	29	30	31				

Group By and Aggregation

Similar to SQL.

→ reading CSV file as dataframe. by first creating spark session ~~then~~

groupBy()

→ groups the data on the basis of column name passed into it.

→ Then we can use aggregate functions on grouped data.

df.groupBy('Department').sum("salary").show()

Spark.read.csv(path, inferSchema = True, header = True)

By default inferSchema and header attributes are false.

If inferSchema = True,
Dataframe infers the data according to different data types.

header = True
first line of file will be taken as column names.

Strategy is not the consequence of planning, but the opposite: its the starting point

aggregate functions

→ min, max, mean, avg
→ count
→ .agg ({'column_name': 'agg-fun'})

- pivot :- it can rotate the data from one column into multiple columns

Handling missing values

if any null values are present in the data then we can drop those using
dataframe.na.drop()

attributes / parameters of drop() :

1) how = 'all'
: if all values of row are null then drop
how - 'any' (default)
thresh = 2 atleast 2 non-null values should be present.
subset = ["column name"] values that are null in subset only that column.

filling na values

using .fill('string') for string values
.fill(0) for integers values

Unless commitment is made, there are only promises and hopes; but no plans

sort. → ~~parameters.~~

df.sort(df["column_name"].desc()).show()

default is asc.

multiple columns in sort.

df.sort("col1","col2")

OrderBy :-

order column values. (single column).

Joins :-

Similar to SQL.

Left semi :- returns columns from left dataset for only matched records.

Left Anti → columns from left dataset for non-matched records.

09.00
10.00
11.00
12.00
01.00
02.00
03.00
04.00
05.00
06.00
07.00

13 | 2023 WEDNESDAY DECEMBER
Wk 50 • DAY 347-018
13-12-2023

Wk	M	T	W	T	F	S	S
48					1	2	3
49	4	5	6	7	8	9	10
50	11	12	13	14	15	16	17
51	18	19	20	21	22	23	24
52	25	26	27	28	29	30	31

Read Text file into PySpark Dataframe:

There are three methods to read files into pyspark datafram. They are read.text(),

Using read.csv() :

```
]: csv_file = spark.read.csv('/FileStore/tables/table1/jobs_in_data.csv', sep = ',', inferSchema = True, header = True)
csv_file.show()
```

```
+---------+--------------------+--------------------+--------------+------+-------------+-----------------+---------------+---------------+----------
--+---------------+------------+
|work_year|           job_title|        job_category|salary_currency|salary|salary_in_usd|employee_residence|experience_level|employment_type|work_setti
ng|company_location|company_size|
+---------+--------------------+--------------------+--------------+------+-------------+-----------------+---------------+---------------+----------
--+---------------+------------+
|     2023|Data DevOps Engineer|    Data Engineering|           EUR| 88000|        95012|          Germany|      Mid-level|      Full-time|      Hybr
id|        Germany|           L|
|     2023|      Data Architect|Data Architecture...|           USD|186000|       186000|    United States|         Senior|      Full-time|   In-pers
on|  United States|           M|
|     2023|      Data Architect|Data Architecture...|           USD| 81800|        81800|    United States|         Senior|      Full-time|   In-pers
on|  United States|           M|
|     2023|      Data Scientist|Data Science and ...|           USD|212000|       212000|    United States|         Senior|      Full-time|   In-pers
on|  United States|           M|
|     2023|      Data Scientist|Data Science and ...|           USD| 93300|        93300|    United States|         Senior|      Full-time|   In-pers
on|  United States|           M|
|     2023|      Data Scientist|Data Science and ...|           USD|130000|       130000|    United States|         Senior|      Full-time|      Remo
te|  United States|           M|
|     2023|      Data Scientist|Data Science and ...|           USD|100000|       100000|    United States|         Senior|      Full-time|      Remo
te|  United States|           M|
|     2023|Machine Learning ...|Machine Learning ...|           USD|224400|       224400|    United States|      Mid-level|      Full-time|   In-pers
on|  United States|           M|
```

and

Using read.format(format_type).load(path) : The .format() specifies the input data source format as “text”. The .load() loads data from a data source and returns DataFrame.

```
[ ]: import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('DataFrame_operations').getOrCreate()
```

```
[ ]: df_read_format = spark.read.format("text").load("/FileStore/tables/Sample-1.txt")
df_read_format.selectExpr("split(value,' ') as Text_data_in_rows_using_format").show()
```

```
+------------------------------+
|Text_data_in_rows_using_format|
+------------------------------+
|          [work_year,job_ti...|
|          [2023,Data, DevOp...|
|          [2023,Data, Archi...|
|          [2023,Data, Archi...|
|          [2023,Data, Scien...|
|          [2023,Data, Scien...|
|          [2023,Data, Scien...|
|          [2023,Data, Scien...|
|          [2023,Machine, Le...|
|          [2023,Machine, Le...|
|          [2023,Data, Engin...|
|          [2023,Data, Engin...|
|          [2023,Machine, Le...|
|          [2023,Machine, Le...|
|          [2023,Data, Scien...|
```

Adding New Column to the dataframe :

There are various methods to add new column to pyspark dataframe :

First we create dataframe as given below :

```
]: # Create data in dataframe
data = [(('Ram'), '1991-04-01', 'M', 3000),
        (('Mike'), '2000-05-19', 'M', 4000),
        (('Rohini'), '1978-09-05', 'M', 4000),
        (('Maria'), '1967-12-01', 'F', 4000),
        (('Jenis'), '1980-02-17', 'F', 1200)]

# Column names in dataframe
columns = ["Name", "DOB", "Gender", "salary"]

# Create the spark dataframe
df = spark.createDataFrame(data=data ,schema=columns)

# Print the dataframe
df.show()
```

```
+------+----------+------+------+
|  Name|       DOB|Gender|salary|
+------+----------+------+------+
|   Ram|1991-04-01|     M|  3000|
|  Mike|2000-05-19|     M|  4000|
|Rohini|1978-09-05|     M|  4000|
| Maria|1967-12-01|     F|  4000|
| Jenis|1980-02-17|     F|  1200|
+------+----------+------+------+
```

Using withColumn() with Constant Value :

Here we use the lit() function parameter of the withColumn() function to give the constant value to the column. Importing lit() from pyspark.sql.functions module.

Syntax : `dataframe.withColumn("column_name", lit(value))`

```
from pyspark.sql.functions import lit
df_addColumn = df.withColumn('Country',lit('India'))
df_addColumn.show()
```

```
+------+----------+------+------+-------+
|  Name|       DOB|Gender|salary|Country|
+------+----------+------+------+-------+
|   Ram|1991-04-01|     M|  3000|  India|
|  Mike|2000-05-19|     M|  4000|  India|
|Rohini|1978-09-05|     M|  4000|  India|
| Maria|1967-12-01|     F|  4000|  India|
| Jenis|1980-02-17|     F|  1200|  India|
+------+----------+------+------+-------+
```

Using concat_ws() : concat the two existing columns and make them as a new column by importing this method from pyspark.sql.functions module.

Syntax:

dataframe.withColumn("column_name",concat_ws("Separator","existing_column1",'existing_column2'))

```
from pyspark.sql.functions import concat_ws
df_concat = df_addColumn.withColumn("Name-Country",concat_ws("-",'Name','Country'))
df_concat.show()
```

```
+------+----------+------+------+-------+------------+
|  Name|       DOB|Gender|salary|Country|Name-Country|
+------+----------+------+------+-------+------------+
|   Ram|1991-04-01|     M|  3000|  India|   Ram-India|
|  Mike|2000-05-19|     M|  4000|  India|  Mike-India|
|Rohini|1978-09-05|     M|  4000|  India|Rohini-India|
| Maria|1967-12-01|     F|  4000|  India| Maria-India|
| Jenis|1980-02-17|     F|  1200|  India| Jenis-India|
+------+----------+------+------+-------+------------+
```

Add Column When not Exists on DataFrame :

```
if 'Country' not in df.columns:
    df.withColumn("Country", lit('India')).show()
```

```
+------+----------+------+------+-------+
|  Name|       DOB|Gender|salary|Country|
+------+----------+------+------+-------+
|   Ram|1991-04-01|     M|  3000|  India|
|  Mike|2000-05-19|     M|  4000|  India|
|Rohini|1978-09-05|     M|  4000|  India|
| Maria|1967-12-01|     F|  4000|  India|
| Jenis|1980-02-17|     F|  1200|  India|
+------+----------+------+------+-------+
```

GroupBy and Aggregate :

Here we are grouping the data on Gender and calculating the sum of salary for grouped data.

```
from pyspark.sql.functions import col
df.show()
# rename grouped column using select
df.groupBy("Gender").sum('salary').select(col('Gender'),col("sum(salary)").alias('Group_sum_Salary')).show()
```

```
+------+----------+------+------+
|  Name|       DOB|Gender|salary|
+------+----------+------+------+
|   Ram|1991-04-01|     M|  3000|
|  Mike|2000-05-19|     M|  4000|
|Rohini|1978-09-05|     M|  4000|
| Maria|1967-12-01|     F|  4000|
| Jenis|1980-02-17|     F|  1200|
+------+----------+------+------+

+------+----------------+
|Gender|Group_sum_Salary|
+------+----------------+
|     M|           11000|
|     F|            5200|
+------+----------------+
```

Following aggregations are performed on different dataset :

sum()
min()
max()
mean()
count()
avg()

sum()

```
[ ]: csv_file.groupBy('job_title').sum('salary').show()
```

```
+--------------------+-----------+
|           job_title|sum(salary)|
+--------------------+-----------+
|Machine Learning ...|     363100|
|      Data Scientist|    1134300|
|        Data Analyst|     170000|
|Data DevOps Engineer|      88000|
|      Data Architect|     267800|
|Machine Learning ...|     363100|
|       Data Engineer|     378000|
+--------------------+-----------+
```

min()

```
[ ]: csv_file.groupBy('job_title').min('salary').show()
```

```
+--------------------+-----------+
|           job_title|min(salary)|
+--------------------+-----------+
|Machine Learning ...|     138700|
|      Data Scientist|      30000|
|        Data Analyst|      75000|
|Data DevOps Engineer|      88000|
|      Data Architect|      81800|
|Machine Learning ...|     138700|
|       Data Engineer|     168000|
+--------------------+-----------+
```

max()

```
[ ]: csv_file.groupBy('job_title').max('salary').show()
```

```
+--------------------+-----------+
|           job_title|max(salary)|
+--------------------+-----------+
|Machine Learning ...|     224400|
|      Data Scientist|     300000|
|        Data Analyst|      95000|
|Data DevOps Engineer|      88000|
|      Data Architect|     186000|
|Machine Learning ...|     224400|
|       Data Engineer|     210000|
+--------------------+-----------+
```

avg()

```
[ ]: csv_file.groupBy('job_title').avg('salary').show()
```

```
+--------------------+-----------+
|           job_title|avg(salary)|
+--------------------+-----------+
|Machine Learning ...|   181550.0|
|      Data Scientist|   141787.5|
|        Data Analyst|    85000.0|
|Data DevOps Engineer|    88000.0|
|      Data Architect|   133900.0|
|Machine Learning ...|   181550.0|
|       Data Engineer|   189000.0|
+--------------------+-----------+
```

mean()

```
[ ]: csv_file.groupBy('job_title').mean('salary').show()
```

```
+--------------------+-----------+
|           job_title|avg(salary)|
+--------------------+-----------+
|Machine Learning ...|   181550.0|
|      Data Scientist|   141787.5|
|        Data Analyst|    85000.0|
|Data DevOps Engineer|    88000.0|
|      Data Architect|   133900.0|
|Machine Learning ...|   181550.0|
|       Data Engineer|   189000.0|
+--------------------+-----------+
```

count() : gives the count of rows in grouped data

```
]: csv_file.groupBy('job_title').count().show()
```

```
+--------------------+-----+
|           job_title|count|
+--------------------+-----+
|Machine Learning ...|    2|
|      Data Scientist|    8|
|        Data Analyst|    2|
|Data DevOps Engineer|    1|
|      Data Architect|    2|
|Machine Learning ...|    2|
|       Data Engineer|    2|
+--------------------+-----+
```

Grouping on two columns :

```
[ ]: csv_file.groupBy("employee_residence","job_title").sum("salary").show()
```

```
+-----------------+--------------------+-----------+
|employee_residence|           job_title|sum(salary)|
+-----------------+--------------------+-----------+
|    United States|       Data Engineer|     378000|
|    United States|Machine Learning ...|     363100|
|    United States|Machine Learning ...|     363100|
|   United Kingdom|      Data Scientist|      65000|
|    United States|        Data Analyst|     170000|
|    United States|      Data Scientist|    1069300|
|    United States|      Data Architect|     267800|
|          Germany|Data DevOps Engineer|      88000|
+-----------------+--------------------+-----------+
```

Using .agg() with groupBy :

```
[ ]: csv_file.agg({'salary':'sum'}).show()
```

```
+-----------+
|sum(salary)|
+-----------+
|    2764300|
+-----------+
```

Pivot() : It rotates one column into multiple columns. Here we pivoted employee_residence column.

```
[ ]: csv_file.groupBy("job_title").pivot("employee_residence").count().show()
```

```
+--------------------+-------+--------------+-------------+
|           job_title|Germany|United Kingdom|United States|
+--------------------+-------+--------------+-------------+
|Machine Learning ...|   null|          null|            2|
|      Data Scientist|   null|             2|            6|
|        Data Analyst|   null|          null|            2|
|Data DevOps Engineer|      1|          null|         null|
|      Data Architect|   null|          null|            2|
|Machine Learning ...|   null|          null|            2|
|       Data Engineer|   null|          null|            2|
+--------------------+-------+--------------+-------------+
```

Handling Missing Values Pyspark :

Created a dataset with null values.

```
df_null = spark.read.csv('/FileStore/tables/jobs_in_data___Copy.csv',header=True,inferSchema=True)
df_null.show()
```

```
+---------+--------------------+------+-----------------+------------+----------------+
|work_year|           job_title|salary|employee_residence|work_setting|company_location|
+---------+--------------------+------+-----------------+------------+----------------+
|     2023|Data DevOps Engineer| 88000|          Germany|      Hybrid|         Germany|
|     2023|      Data Architect|186000|    United States|   In-person|   United States|
|     2023|      Data Architect| 81800|             null|   In-person|   United States|
|     2023|      Data Scientist|212000|    United States|   In-person|   United States|
|     2023|                null| 93300|    United States|   In-person|   United States|
|     2023|      Data Scientist|130000|    United States|      Remote|   United States|
|     2023|      Data Scientist|100000|    United States|      Remote|            null|
|     null|Machine Learning ...|  null|    United States|   In-person|   United States|
|     2023|Machine Learning ...|138700|    United States|        null|   United States|
|     2023|       Data Engineer|210000|    United States|      Remote|   United States|
|     2023|                null|168000|    United States|      Remote|   United States|
|     2023|Machine Learning ...|224400|    United States|   In-person|   United States|
|     2023|Machine Learning ...|138700|    United States|   In-person|   United States|
|     2023|      Data Scientist| 35000|   United Kingdom|   In-person|  United Kingdom|
|     2023|      Data Scientist| 30000|   United Kingdom|   In-person|  United Kingdom|
|     2023|        Data Analyst| 95000|             null|   In-person|   United States|
|     2023|        Data Analyst| 75000|    United States|   In-person|   United States|
|     2023|      Data Scientist|300000|    United States|   In-person|   United States|
|     2023|                null|234000|    United States|   In-person|   United States|
+---------+--------------------+------+-----------------+------------+----------------+
```

Drop rows with any null values using .na.drop() :

```
df_null.na.drop().show()
```

```
+---------+--------------------+------+------------------+------------+----------------+
|work_year|           job_title|salary|employee_residence|work_setting|company_location|
+---------+--------------------+------+------------------+------------+----------------+
|     2023|Data DevOps Engineer| 88000|           Germany|      Hybrid|         Germany|
|     2023|      Data Architect|186000|     United States|   In-person|   United States|
|     2023|      Data Scientist|212000|     United States|   In-person|   United States|
|     2023|      Data Scientist|130000|     United States|      Remote|   United States|
|     2023|       Data Engineer|210000|     United States|      Remote|   United States|
|     2023|Machine Learning ...|224400|     United States|   In-person|   United States|
|     2023|Machine Learning ...|138700|     United States|   In-person|   United States|
|     2023|      Data Scientist| 35000|    United Kingdom|   In-person|  United Kingdom|
|     2023|      Data Scientist| 30000|    United Kingdom|   In-person|  United Kingdom|
|     2023|        Data Analyst| 75000|     United States|   In-person|   United States|
|     2023|      Data Scientist|300000|     United States|   In-person|   United States|
+---------+--------------------+------+------------------+------------+----------------+
```

Parameters of drop(how=" all/any ")

```
df_null.na.drop(how='all').show() # all values should be null to get removed
```

```
+---------+--------------------+------+------------------+------------+----------------+
|work_year|           job_title|salary|employee_residence|work_setting|company_location|
+---------+--------------------+------+------------------+------------+----------------+
|     2023|Data DevOps Engineer| 88000|           Germany|      Hybrid|         Germany|
|     2023|      Data Architect|186000|     United States|   In-person|   United States|
|     2023|      Data Architect| 81800|              null|   In-person|   United States|
|     2023|      Data Scientist|212000|     United States|   In-person|   United States|
|     2023|                null| 93300|     United States|   In-person|   United States|
|     2023|      Data Scientist|130000|     United States|      Remote|   United States|
|     2023|      Data Scientist|100000|     United States|      Remote|            null|
|     null|Machine Learning ...|  null|     United States|   In-person|   United States|
|     2023|Machine Learning ...|138700|     United States|        null|   United States|
|     2023|       Data Engineer|210000|     United States|      Remote|   United States|
|     2023|                null|168000|     United States|      Remote|   United States|
|     2023|Machine Learning ...|224400|     United States|   In-person|   United States|
|     2023|Machine Learning ...|138700|     United States|   In-person|   United States|
|     2023|      Data Scientist| 35000|    United Kingdom|   In-person|  United Kingdom|
|     2023|      Data Scientist| 30000|    United Kingdom|   In-person|  United Kingdom|
|     2023|        Data Analyst| 95000|              null|   In-person|   United States|
|     2023|        Data Analyst| 75000|     United States|   In-person|   United States|
|     2023|      Data Scientist|300000|     United States|   In-person|   United States|
|     2023|                null|234000|     United States|   In-person|   United States|
+---------+--------------------+------+------------------+------------+----------------+
```

Parameters of drop(thresh=)

```
df_null.na.drop(thresh=5).show() # atleast 5 non-null values should be there in row to be in result
```

```
+---------+--------------------+------+------------------+------------+----------------+
|work_year|           job_title|salary|employee_residence|work_setting|company_location|
+---------+--------------------+------+------------------+------------+----------------+
|     2023|Data DevOps Engineer| 88000|           Germany|      Hybrid|         Germany|
|     2023|      Data Architect|186000|     United States|   In-person|   United States|
|     2023|      Data Architect| 81800|              null|   In-person|   United States|
|     2023|      Data Scientist|212000|     United States|   In-person|   United States|
|     2023|                null| 93300|     United States|   In-person|   United States|
|     2023|      Data Scientist|130000|     United States|      Remote|   United States|
|     2023|      Data Scientist|100000|     United States|      Remote|            null|
|     2023|Machine Learning ...|138700|     United States|        null|   United States|
|     2023|       Data Engineer|210000|     United States|      Remote|   United States|
|     2023|                null|168000|     United States|      Remote|   United States|
|     2023|Machine Learning ...|224400|     United States|   In-person|   United States|
|     2023|Machine Learning ...|138700|     United States|   In-person|   United States|
|     2023|      Data Scientist| 35000|    United Kingdom|   In-person|  United Kingdom|
|     2023|      Data Scientist| 30000|    United Kingdom|   In-person|  United Kingdom|
|     2023|        Data Analyst| 95000|              null|   In-person|   United States|
|     2023|        Data Analyst| 75000|     United States|   In-person|   United States|
|     2023|      Data Scientist|300000|     United States|   In-person|   United States|
|     2023|                null|234000|     United States|   In-person|   United States|
+---------+--------------------+------+------------------+------------+----------------+
```

Parameters of drop(subset=” ”)

```
df_null.na.drop(subset='job_title').show()
```

```
+---------+--------------------+------+------------------+------------+----------------+
|work_year|           job_title|salary|employee_residence|work_setting|company_location|
+---------+--------------------+------+------------------+------------+----------------+
|     2023|Data DevOps Engineer| 88000|           Germany|      Hybrid|         Germany|
|     2023|      Data Architect|186000|     United States|   In-person|   United States|
|     2023|      Data Architect| 81800|              null|   In-person|   United States|
|     2023|      Data Scientist|212000|     United States|   In-person|   United States|
|     2023|      Data Scientist|130000|     United States|      Remote|   United States|
|     2023|      Data Scientist|100000|     United States|      Remote|            null|
|     null|Machine Learning ...|  null|     United States|   In-person|   United States|
|     2023|Machine Learning ...|138700|     United States|        null|   United States|
|     2023|       Data Engineer|210000|     United States|      Remote|   United States|
|     2023|Machine Learning ...|224400|     United States|   In-person|   United States|
|     2023|Machine Learning ...|138700|     United States|   In-person|   United States|
|     2023|      Data Scientist| 35000|    United Kingdom|   In-person|  United Kingdom|
|     2023|      Data Scientist| 30000|    United Kingdom|   In-person|  United Kingdom|
|     2023|        Data Analyst| 95000|              null|   In-person|   United States|
|     2023|        Data Analyst| 75000|     United States|   In-person|   United States|
|     2023|      Data Scientist|300000|     United States|   In-person|   United States|
+---------+--------------------+------+------------------+------------+----------------+
```

Filling null string values using .na.fill("string"):

```
df_null.na.fill('Missing Values').show()
```

```
+---------+--------------------+------+-----------------+-------------+----------------+
|work_year|           job_title|salary|employee_residence|  work_setting|company_location|
+---------+--------------------+------+-----------------+-------------+----------------+
|     2023|Data DevOps Engineer| 88000|          Germany|       Hybrid|         Germany|
|     2023|      Data Architect|186000|    United States|    In-person|   United States|
|     2023|      Data Architect| 81800|   Missing Values|    In-person|   United States|
|     2023|      Data Scientist|212000|    United States|    In-person|   United States|
|     2023|      Missing Values| 93300|    United States|    In-person|   United States|
|     2023|      Data Scientist|130000|    United States|       Remote|   United States|
|     2023|      Data Scientist|100000|    United States|       Remote|  Missing Values|
|     null|Machine Learning ...|  null|    United States|    In-person|   United States|
|     2023|Machine Learning ...|138700|    United States|Missing Values|   United States|
|     2023|       Data Engineer|210000|    United States|       Remote|   United States|
|     2023|      Missing Values|168000|    United States|       Remote|   United States|
|     2023|Machine Learning ...|224400|    United States|    In-person|   United States|
|     2023|Machine Learning ...|138700|    United States|    In-person|   United States|
|     2023|      Data Scientist| 35000|   United Kingdom|    In-person|  United Kingdom|
|     2023|      Data Scientist| 30000|   United Kingdom|    In-person|  United Kingdom|
|     2023|        Data Analyst| 95000|   Missing Values|    In-person|   United States|
|     2023|        Data Analyst| 75000|    United States|    In-person|   United States|
|     2023|      Data Scientist|300000|    United States|    In-person|   United States|
|     2023|      Missing Values|234000|    United States|    In-person|   United States|
+---------+--------------------+------+-----------------+-------------+----------------+
```

Filling null integer values using .na.fill(0):

```
df_null.na.fill(0).show()
```

```
+---------+--------------------+------+-----------------+------------+----------------+
|work_year|           job_title|salary|employee_residence|work_setting|company_location|
+---------+--------------------+------+-----------------+------------+----------------+
|     2023|Data DevOps Engineer| 88000|          Germany|      Hybrid|         Germany|
|     2023|      Data Architect|186000|    United States|   In-person|   United States|
|     2023|      Data Architect| 81800|             null|   In-person|   United States|
|     2023|      Data Scientist|212000|    United States|   In-person|   United States|
|     2023|                null| 93300|    United States|   In-person|   United States|
|     2023|      Data Scientist|130000|    United States|      Remote|   United States|
|     2023|      Data Scientist|100000|    United States|      Remote|            null|
|        0|Machine Learning ...|     0|    United States|   In-person|   United States|
|     2023|Machine Learning ...|138700|    United States|        null|   United States|
|     2023|       Data Engineer|210000|    United States|      Remote|   United States|
|     2023|                null|168000|    United States|      Remote|   United States|
|     2023|Machine Learning ...|224400|    United States|   In-person|   United States|
|     2023|Machine Learning ...|138700|    United States|   In-person|   United States|
|     2023|      Data Scientist| 35000|   United Kingdom|   In-person|  United Kingdom|
|     2023|      Data Scientist| 30000|   United Kingdom|   In-person|  United Kingdom|
|     2023|        Data Analyst| 95000|             null|   In-person|   United States|
|     2023|        Data Analyst| 75000|    United States|   In-person|   United States|
|     2023|      Data Scientist|300000|    United States|   In-person|   United States|
|     2023|                null|234000|    United States|   In-person|   United States|
+---------+--------------------+------+-----------------+------------+----------------+
```

Sort data using sort() :

```
df_null.sort('salary').show()
```

```
+---------+--------------------+------+------------------+------------+----------------+
|work_year|           job_title|salary|employee_residence|work_setting|company_location|
+---------+--------------------+------+------------------+------------+----------------+
|     null|Machine Learning ...|  null|     United States|   In-person|   United States|
|     2023|      Data Scientist| 30000|    United Kingdom|   In-person|  United Kingdom|
|     2023|      Data Scientist| 35000|    United Kingdom|   In-person|  United Kingdom|
|     2023|        Data Analyst| 75000|     United States|   In-person|   United States|
|     2023|      Data Architect| 81800|              null|   In-person|   United States|
|     2023|Data DevOps Engineer| 88000|           Germany|      Hybrid|         Germany|
|     2023|                null| 93300|     United States|   In-person|   United States|
|     2023|        Data Analyst| 95000|              null|   In-person|   United States|
|     2023|      Data Scientist|100000|     United States|      Remote|            null|
|     2023|      Data Scientist|130000|     United States|      Remote|   United States|
|     2023|Machine Learning ...|138700|     United States|        null|   United States|
|     2023|Machine Learning ...|138700|     United States|   In-person|   United States|
|     2023|                null|168000|     United States|      Remote|   United States|
|     2023|      Data Architect|186000|     United States|   In-person|   United States|
|     2023|       Data Engineer|210000|     United States|      Remote|   United States|
|     2023|      Data Scientist|212000|     United States|   In-person|   United States|
|     2023|Machine Learning ...|224400|     United States|   In-person|   United States|
|     2023|                null|234000|     United States|   In-person|   United States|
|     2023|      Data Scientist|300000|     United States|   In-person|   United States|
+---------+--------------------+------+------------------+------------+----------------+
```

Joins on pyspark dataframe :

```
emp = [(1,"Smith",-1,"2018","10","M",3000),
       (2, "Rose",1 , "2010", "20","M", 4000),
       (3,"Williams",1,"2010","10","M",1000),
       (4, "Jones",2 ,"2005","10","F",2000),
       (5,"Brown",2,"2010","40","",-1),
       (6, "Brown", 2, "2010","50","",-1)]
empColumns = ["emp_id","name","superior_emp_id","year_joined", "emp_dept_id","gender","salary"]

empDF = spark.createDataFrame(data=emp, schema = empColumns)
empDF.printSchema()
empDF.show()

dept = [("Finance",10),("Marketing",20),("Sales",30),("IT",40)]
deptColumns = ["dept_name","dept_id"]
deptDF = spark.createDataFrame(data=dept, schema = deptColumns)
deptDF.printSchema()
deptDF.show()
```

```
root
 |-- emp_id: long (nullable = true)
 |-- name: string (nullable = true)
 |-- superior_emp_id: long (nullable = true)
 |-- year_joined: string (nullable = true)
 |-- emp_dept_id: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- salary: long (nullable = true)
```

emp_id	name	superior_emp_id	year_joined	emp_dept_id	gender	salary
1	Smith	-1	2018	10	M	3000
2	Rose	1	2010	20	M	4000
3	Williams	1	2010	10	M	1000
4	Jones	2	2005	10	F	2000
5	Brown	2	2010	40		-1
6	Brown	2	2010	50		-1

```
root
 |-- dept_name: string (nullable = true)
 |-- dept_id: long (nullable = true)
```

dept_name	dept_id
Finance	10
Marketing	20
Sales	30
IT	40

Inner join :

```
empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"inner") .show()
```

```
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|     1|   Smith|             -1|       2018|         10|     M|  3000|  Finance|     10|
|     3|Williams|              1|       2010|         10|     M|  1000|  Finance|     10|
|     4|   Jones|              2|       2005|         10|     F|  2000|  Finance|     10|
|     2|    Rose|              1|       2010|         20|     M|  4000|Marketing|     20|
|     5|   Brown|              2|       2010|         40|      |    -1|       IT|     40|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
```

Outer join :

```
empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"outer").show()
#Or instead of outer we can give full Or fullouter
```

```
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|     1|   Smith|             -1|       2018|         10|     M|  3000|  Finance|     10|
|     3|Williams|              1|       2010|         10|     M|  1000|  Finance|     10|
|     4|   Jones|              2|       2005|         10|     F|  2000|  Finance|     10|
|     2|    Rose|              1|       2010|         20|     M|  4000|Marketing|     20|
|  null|    null|           null|       null|       null|  null|  null|    Sales|     30|
|     5|   Brown|              2|       2010|         40|      |    -1|       IT|     40|
|     6|   Brown|              2|       2010|         50|      |    -1|     null|   null|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
```

Left join :

```
empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"left").show()
# Or leftouter
```

```
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|     1|   Smith|             -1|       2018|         10|     M|  3000|  Finance|     10|
|     2|    Rose|              1|       2010|         20|     M|  4000|Marketing|     20|
|     3|Williams|              1|       2010|         10|     M|  1000|  Finance|     10|
|     4|   Jones|              2|       2005|         10|     F|  2000|  Finance|     10|
|     5|   Brown|              2|       2010|         40|      |    -1|       IT|     40|
|     6|   Brown|              2|       2010|         50|      |    -1|     null|   null|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
```

Right join :

```
empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"right").show()
# Or rightouter
```

```
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|dept_name|dept_id|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
|     4|   Jones|              2|       2005|         10|     F|  2000|  Finance|     10|
|     3|Williams|              1|       2010|         10|     M|  1000|  Finance|     10|
|     1|   Smith|             -1|       2018|         10|     M|  3000|  Finance|     10|
|     2|    Rose|              1|       2010|         20|     M|  4000|Marketing|     20|
|  null|    null|           null|       null|       null|  null|  null|    Sales|     30|
|     5|   Brown|              2|       2010|         40|      |    -1|       IT|     40|
+------+--------+---------------+-----------+-----------+------+------+---------+-------+
```

Leftsemi join :

```
]: empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"leftsemi").show()
```

```
+------+--------+---------------+-----------+-----------+------+------+
|emp_id|    name|superior_emp_id|year_joined|emp_dept_id|gender|salary|
+------+--------+---------------+-----------+-----------+------+------+
|     1|   Smith|             -1|       2018|         10|     M|  3000|
|     3|Williams|              1|       2010|         10|     M|  1000|
|     4|   Jones|              2|       2005|         10|     F|  2000|
|     2|    Rose|              1|       2010|         20|     M|  4000|
|     5|   Brown|              2|       2010|         40|      |    -1|
+------+--------+---------------+-----------+-----------+------+------+
```

Leftanti join :

```
empDF.join(deptDF,empDF.emp_dept_id ==  deptDF.dept_id,"leftanti").show()
```

```
+------+-----+---------------+-----------+-----------+------+------+
|emp_id| name|superior_emp_id|year_joined|emp_dept_id|gender|salary|
+------+-----+---------------+-----------+-----------+------+------+
|     6|Brown|              2|       2010|         50|      |    -1|
+------+-----+---------------+-----------+-----------+------+------+
```