Import py spark

In [5]:
```
import findspark
findspark.init()
findspark.find()
```

Out[5]:
```
'D:\\Java Installation\\spark-3.5.0-bin-hadoop3'
```

In [6]:
```
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
sc = SparkContext.getOrCreate()
spark = SparkSession(sc)
```

In [7]:
```
sc
```

Out[7]:

SparkContext

Spark UI

Version	`v3.5.0`
Master	`local[*]`
AppName	`pyspark-shell`

Creating a range of numbers

In [8]:
```
myRange = spark.range(1000).toDF("number")
```

In [9]:
```
myRange
```

Out[9]:
```
DataFrame[number: bigint]
```

Simple transformation to find all even numbers in myRange DF

In [11]:
```
EvenNo = myRange.where("number % 2 = 0")
```

In [12]:
```
EvenNo.count()
```

Out[12]:
```
500
```

Import csv data

In [22]:
```
flightData2015 = spark\
                 .read\
                 .option("inferSchema","true")\
                 .option("header","true")\
                 .csv("C://Users//Admin//Downloads//2015-summary.csv")
```

In [23]:
```
flightData2015.take(3)
```

Out[23]:
```
[Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Romania', count=15),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Croatia', count=1),
 Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Ireland', count=344)]
```

In [24]:
```
flightData2015.sort("count").explain()
```

```
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- Sort [count#29 ASC NULLS FIRST], true, 0
   +- Exchange rangepartitioning(count#29 ASC NULLS FIRST, 200), ENSURE_REQUIREMENTS, [plan_id=71]
      +- FileScan csv [DEST_COUNTRY_NAME#27,ORIGIN_COUNTRY_NAME#28,count#29] Batched: false, DataFilters: [], Format: C
SV, Location: InMemoryFileIndex(1 paths)[file:/C:/Users/Admin/Downloads/2015-summary.csv], PartitionFilters: [], Pushed
Filters: [], ReadSchema: struct<DEST_COUNTRY_NAME:string,ORIGIN_COUNTRY_NAME:string,count:int>
```

In [25]:
```
# Setting number of partitions to 5 instead of taking the default 200 as the partition

spark.conf.set("spark.sql.shuffle.partitions","5")
flightData2015.sort("count").take(2)
```

Out[25]:
```
[Row(DEST_COUNTRY_NAME='United States', ORIGIN_COUNTRY_NAME='Singapore', count=1),
 Row(DEST_COUNTRY_NAME='Moldova', ORIGIN_COUNTRY_NAME='United States', count=1)]
```

In [ ]:
```

```