Pyspark DataType
Spark has lazy evaluation (we have to put .show() for displaying any values)
Transformations and Actions (narrow, wide [shuffle] : join, groupBy)
Cache (Memory) and Persist (Memory or Disk)

In [1]:
```
import findspark
findspark.init()
from pyspark.sql import SparkSession
spark=SparkSession.builder.appName("example").getOrCreate()
```

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/09/22 04:26:48 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-j
ava classes where applicable
```

In [2]:
```
data=[(1,"a",30),(2,"b",32)]
from pyspark.sql.types import *
user_schema=StructType([StructField("id",IntegerType()),
                        StructField("name",StringType()),
                        StructField("age",IntegerType())
])
df=spark.createDataFrame(data,user_schema)
```

In [3]:
```
df.show()
```

```
+---+----+---+
| id|name|age|
+---+----+---+
|  1|   a| 30|
|  2|   b| 32|
```

In [7]:
```
data2=[
    (1,"Alice",["Reading","Hiking"]),
    (2, "Bob", ["Swimming", "Gardening", "Painting"]),
    (3, "Charlie", ["Cooking"]),
    (4, "David", ["Photography", "Skiing", "Cooking"])
]
```

In [8]:
```
user_schema2 = StructType([
    StructField("id", IntegerType()),
    StructField("name", StringType()),
    StructField("hobbies", ArrayType(StringType()))
])
```

In [9]:
```
df2=spark.createDataFrame(data2,user_schema2)
```

In [10]:
```
df2.show()
```

```
+---+-------+--------------------+
| id|   name|             hobbies|
+---+-------+--------------------+
|  1|  Alice|   [Reading, Hiking]|
|  2|    Bob|[Swimming, Garden...|
|  3|Charlie|           [Cooking]|
|  4|  David|[Photography, Ski...|
+---+-------+--------------------+
```

In [11]:
```
from pyspark.sql.functions import *
```

In [12]:
```
df3=df2.select("id", "name", explode("hobbies").alias("hobby")).show()
```

```
+---+-------+-----------+
| id|   name|      hobby|
+---+-------+-----------+
|  1|  Alice|    Reading|
|  1|  Alice|     Hiking|
|  2|    Bob|   Swimming|
|  2|    Bob|  Gardening|
|  2|    Bob|   Painting|
|  3|Charlie|    Cooking|
|  4|  David|Photography|
|  4|  David|     Skiing|
|  4|  David|    Cooking|
+---+-------+-----------+
```

In [16]:
```
df2.withColumn("newhobby",explode("Hobbies")).show()
```

```
+---+-------+--------------------+-----------+
| id|   name|             hobbies|   newhobby|
+---+-------+--------------------+-----------+
|  1|  Alice|   [Reading, Hiking]|    Reading|
|  1|  Alice|   [Reading, Hiking]|     Hiking|
```

In [20]:
```
df3=df2\
.withColumn("Hobbies",explode("hobbies"))\
.withColumn("ingestion_data",current_timestamp())
```

In [22]:
```
df3.show(truncate=False)
```

```
+---+-------+-----------+-----------------------+
|id |name   |Hobbies    |ingestion_data         |
+---+-------+-----------+-----------------------+
|1  |Alice  |Reading    |2023-09-22 04:49:42.035|
|1  |Alice  |Hiking     |2023-09-22 04:49:42.035|
|2  |Bob    |Swimming   |2023-09-22 04:49:42.035|
|2  |Bob    |Gardening  |2023-09-22 04:49:42.035|
|2  |Bob    |Painting   |2023-09-22 04:49:42.035|
|3  |Charlie|Cooking    |2023-09-22 04:49:42.035|
|4  |David  |Photography|2023-09-22 04:49:42.035|
|4  |David  |Skiing     |2023-09-22 04:49:42.035|
|4  |David  |Cooking    |2023-09-22 04:49:42.035|
+---+-------+-----------+-----------------------+
```

In [1]:
```
import findspark
findspark.init()
from pyspark.sql import SparkSession
spark=SparkSession.builder.appName("example").getOrCreate()
```

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/09/22 04:58:03 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-j
ava classes where applicable
23/09/22 04:58:04 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
```

In [11]:
```
from pyspark.sql.functions import *
from pyspark.sql.types import *
```

In [6]:
```
input_files="/home/labuser/1PySpark/Day 2"
```

In [7]:
```
df=spark.read.json(f"{input_files}/constructor.json")
```

In [10]:
```
df.show()
```

```
+-------------+--------------+-----------+-----------+--------------------+
|constructorId|constructorRef|       name|nationality|                 url|
+-------------+--------------+-----------+-----------+--------------------+
|            1|        mclaren|    McLaren|    British|http://en.wikiped...|
```

In [14]:
```
df_final=df\
.withColumn("ingestion_date",current_timestamp())\
.withColumn("path",input_file_name())\
.drop("url")
```

In [15]:
```
df_final.show()
```

```
+-------------+--------------+-----------+-----------+--------------------+--------------------+
|constructorId|constructorRef|       name|nationality|      ingestion_date|                path|
+-------------+--------------+-----------+-----------+--------------------+--------------------+
|            1|       mclaren|    McLaren|    British|2023-09-22 05:35:...|file:///home/labu...|
|            2|    bmw_sauber| BMW Sauber|     German|2023-09-22 05:35:...|file:///home/labu...|
|            3|      williams|   Williams|    British|2023-09-22 05:35:...|file:///home/labu...|
|            4|       renault|    Renault|     French|2023-09-22 05:35:...|file:///home/labu...|
|            5|    toro_rosso| Toro Rosso|    Italian|2023-09-22 05:35:...|file:///home/labu...|
|            6|       ferrari|    Ferrari|    Italian|2023-09-22 05:35:...|file:///home/labu...|
|            7|        toyota|     Toyota|   Japanese|2023-09-22 05:35:...|file:///home/labu...|
|            8|   super_aguri|Super Aguri|   Japanese|2023-09-22 05:35:...|file:///home/labu...|
|            9|      red_bull|   Red Bull|   Austrian|2023-09-22 05:35:...|file:///home/labu...|
|           10|   force_india|Force India|     Indian|2023-09-22 05:35:...|file:///home/labu...|
|           11|         honda|      Honda|   Japanese|2023-09-22 05:35:...|file:///home/labu...|
|           12|        spyker|     Spyker|      Dutch|2023-09-22 05:35:...|file:///home/labu...|
|           13|           mf1|        MF1|    Russian|2023-09-22 05:35:...|file:///home/labu...|
|           14|    spyker_mf1| Spyker MF1|      Dutch|2023-09-22 05:35:...|file:///home/labu...|
|           15|        sauber|     Sauber|      Swiss|2023-09-22 05:35:...|file:///home/labu...|
```

In [19]:
```
output_files="/home/labuser/1PySpark/Day 2/processed_data/constructor_parquet"
```

In [17]:
```
df_final.write.parquet(f"{output_files}")
```

In [20]:
```
df_final=df_final.drop("path")
```

In [21]:
```
df_final.write.mode("overwrite").parquet(f"{output_files}")
```

In [22]:
```
df_final.write.saveAsTable("constructor")
```

In [26]:
```
df_final.write.mode("overwrite").option("path","/home/labuser/1PySpark/Day 2/processed_data/constructor_table").save
```

In [23]:
```
spark.sql("select * from constructor").show()
```

```
+-------------+--------------+-----------+-----------+--------------------+
|constructorId|constructorRef|       name|nationality|      ingestion_date|
+-------------+--------------+-----------+-----------+--------------------+
|            1|       mclaren|    McLaren|    British|2023-09-22 05:53:...|
|            2|    bmw_sauber| BMW Sauber|     German|2023-09-22 05:53:...|
|            3|      williams|   Williams|    British|2023-09-22 05:53:...|
```

In [3]:
```
input_files="/home/labuser/1PySpark/Day 2"
```

In [7]:
```
df = spark.read.option("multiLine",True).json(f"{input_files}/pitstop.json")
```

In [8]:
```
df.show()
```

```
+--------+--------+---+------------+------+----+--------+
|driverId|duration|lap|milliseconds|raceId|stop|    time|
+--------+--------+---+------------+------+----+--------+
|     153|  26.898|  1|       26898|   841|   1|17:05:23|
|      30|  25.021|  1|       25021|   841|   1|17:05:52|
|      17|  23.426| 11|       23426|   841|   1|17:20:48|
|       4|  23.251| 12|       23251|   841|   1|17:22:34|
|      13|  23.842| 13|       23842|   841|   1|17:24:10|
|      22|  23.643| 13|       23643|   841|   1|17:24:29|
|      20|  22.603| 14|       22603|   841|   1|17:25:17|
|     814|  24.863| 14|       24863|   841|   1|17:26:03|
|     816|  25.259| 14|       25259|   841|   1|17:26:50|
|      67|  25.342| 15|       25342|   841|   1|17:27:34|
|       2|  22.994| 15|       22994|   841|   1|17:27:41|
|       1|  23.227| 16|       23227|   841|   1|17:28:24|
|     808|  24.535| 16|       24535|   841|   1|17:28:39|
|       3|  23.716| 16|       23716|   841|   1|17:29:00|
|     155|  24.064| 16|       24064|   841|   1|17:29:06|
```

In [9]:
```
df.sort("driverId").show(5)
```

```
+--------+--------+---+------------+------+----+--------+
|driverId|duration|lap|milliseconds|raceId|stop|    time|
+--------+--------+---+------------+------+----+--------+
|       1|  23.227| 16|       23227|   841|   1|17:28:24|
|       1|  23.199| 36|       23199|   841|   2|17:59:29|
|       2|  25.098| 30|       25098|   841|   2|17:51:32|
|       2|  22.994| 15|       22994|   841|   1|17:27:41|
|       3|  23.716| 16|       23716|   841|   1|17:29:00|
+--------+--------+---+------------+------+----+--------+
only showing top 5 rows
```

In [10]:
```
df.groupby("stop").count().show()
```

```
+----+-----+
|stop|count|
+----+-----+
|   1|   21|
|   3|    3|
|   2|   16|
+----+-----+
```

In [11]:
```
df.count()
```

Out[11]:
```
40
```

In [12]:
```
import matplotlib.pyplot as plt
```

In [14]:
```
df_final=df.sort("driverId")
```

In [15]:
```
df_final.write.mode("overwrite").option("path","/home/labuser/1PySpark/Day 2/processed_data/pitstops").saveAsTable('
```

In [16]:
```
spark.sql("select * from pitstop").show()
```

```
+--------+--------+---+------------+------+----+--------+
|driverId|duration|lap|milliseconds|raceId|stop|    time|
+--------+--------+---+------------+------+----+--------+
|       1|  23.227| 16|       23227|   841|   1|17:28:24|
|       1|  23.199| 36|       23199|   841|   2|17:59:29|
|       2|  22.994| 15|       22994|   841|   1|17:27:41|
|       2|  25.098| 30|       25098|   841|   2|17:51:32|
|       3|  23.716| 16|       23716|   841|   1|17:29:00|
|       4|  23.251| 12|       23251|   841|   1|17:22:34|
|       4|  24.733| 27|       24733|   841|   2|17:46:04|
```

In [4]:
```
df_sales = spark.read.option("header",True).option("inferschema",True).csv('sales.csv')
```

In [5]:
```
df_product = spark.read.option("header",True).option("inferschema",True).csv("product.csv")
```

In [6]:
```
df_sales.show()
df_product.show()
```

```
+--------------+----------+-----------+-------------+-------------------+
|transaction_id|product_id|customer_id|quantity_sold|          timestamp|
+--------------+----------+-----------+-------------+-------------------+
|             1|       101|        201|            5|2023-09-22 10:15:00|
|             2|       102|        202|            3|2023-09-22 11:30:00|
|             3|       101|        203|            2|2023-09-22 12:45:00|
|             4|       103|        204|            1|2023-09-22 14:00:00|
|             5|       102|        205|            4|2023-09-22 15:15:00|
+--------------+----------+-----------+-------------+-------------------+

+----------+------------+-----------+-----+
|product_id|product_name|   category|price|
+----------+------------+-----------+-----+
|       101|      Laptop|Electronics|  800|
|       102|  Smartphone|Electronics|  600|
|       103|        Desk|  Furniture|  250|
|       104|  Headphones|Electronics|  100|
|       105|       Chair|  Furniture|  150|
```

In [7]:
```
df_sales.join(df_product).show()
```

```
+--------------+----------+-----------+-------------+-------------------+----------+------------+-----------+-----
+
|transaction_id|product_id|customer_id|quantity_sold|          timestamp|product_id|product_name|   category|price
|
+--------------+----------+-----------+-------------+-------------------+----------+------------+-----------+-----
+
|             1|       101|        201|            5|2023-09-22 10:15:00|       101|      Laptop|Electronics|  800
|
|             1|       101|        201|            5|2023-09-22 10:15:00|       102|  Smartphone|Electronics|  600
|
|             1|       101|        201|            5|2023-09-22 10:15:00|       103|        Desk|  Furniture|  250
|
|             1|       101|        201|            5|2023-09-22 10:15:00|       104|  Headphones|Electronics|  100
|
|             1|       101|        201|            5|2023-09-22 10:15:00|       105|       Chair|  Furniture|  150
|
|             2|       102|        202|            3|2023-09-22 11:30:00|       101|      Laptop|Electronics|  800
|
|             2|       102|        202|            3|2023-09-22 11:30:00|       102|  Smartphone|Electronics|  600
|
|             2|       102|        202|            3|2023-09-22 11:30:00|       103|        Desk|  Furniture|  250
|
|             2|       102|        202|            3|2023-09-22 11:30:00|       104|  Headphones|Electronics|  100
|
|             2|       102|        202|            3|2023-09-22 11:30:00|       105|       Chair|  Furniture|  150
```

In [8]:
```
df_sales.join(df_product, on="product_id").show()
```

```
+----------+--------------+-----------+-------------+-------------------+------------+-----------+-----+
|product_id|transaction_id|customer_id|quantity_sold|          timestamp|product_name|   category|price|
+----------+--------------+-----------+-------------+-------------------+------------+-----------+-----+
|       101|             1|        201|            5|2023-09-22 10:15:00|      Laptop|Electronics|  800|
|       102|             2|        202|            3|2023-09-22 11:30:00|  Smartphone|Electronics|  600|
|       101|             3|        203|            2|2023-09-22 12:45:00|      Laptop|Electronics|  800|
|       103|             4|        204|            1|2023-09-22 14:00:00|        Desk|  Furniture|  250|
|       102|             5|        205|            4|2023-09-22 15:15:00|  Smartphone|Electronics|  600|
+----------+--------------+-----------+-------------+-------------------+------------+-----------+-----+
```

In [9]:
```
df_sales.join(df_product, df_sales["product_id"]==df_product["product_id"]).show()
```

```
+--------------+----------+-----------+-------------+-------------------+----------+------------+-----------+-----
+
|transaction_id|product_id|customer_id|quantity_sold|          timestamp|product_id|product_name|   category|price
|
+--------------+----------+-----------+-------------+-------------------+----------+------------+-----------+-----
+
|             1|       101|        201|            5|2023-09-22 10:15:00|       101|      Laptop|Electronics|  800
|
|             2|       102|        202|            3|2023-09-22 11:30:00|       102|  Smartphone|Electronics|  600
|
```

In [10]:
```
df_join=df_sales.join(df_product, df_sales["product_id"]==df_product["product_id"],how="left")
```

In [11]:
```
df_join.select(['transaction_id','quantity_sold','category','price']).show()
```

```
+--------------+-------------+-----------+-----+
|transaction_id|quantity_sold|   category|price|
+--------------+-------------+-----------+-----+
|             1|            5|Electronics|  800|
|             2|            3|Electronics|  600|
|             3|            2|Electronics|  800|
|             4|            1|  Furniture|  250|
|             5|            4|Electronics|  600|
+--------------+-------------+-----------+-----+
```

In [12]:
```
df_join.filter("transaction_id=1").show()
```

```
+--------------+----------+-----------+-------------+-------------------+----------+------------+-----------+-----
+
|transaction_id|product_id|customer_id|quantity_sold|          timestamp|product_id|product_name|   category|price
|
+--------------+----------+-----------+-------------+-------------------+----------+------------+-----------+-----
+
|             1|       101|        201|            5|2023-09-22 10:15:00|       101|      Laptop|Electronics|  800
```

In [38]:
```
employeesDF. \
    withColumn("nationality",upper("nationality")).\
    withColumn("last_4_digits", substring(col("ssn"), -4, 4).cast("int")).\
    withColumn("country_code", split("phone_number", " ")[0].cast("int")).\
    withColumn("area_code", split("phone_number", " ")[1].cast("int")).\
    show()
```

```
+-----------+----------+---------+------+-------------+---------------+-----------+-------------+-----------+--
-------+
|employee_id|first_name|last_name|salary|   nationality|    phone_number|        ssn|last_4_digits|country_code|ar
ea_code|
+-----------+----------+---------+------+-------------+---------------+-----------+-------------+-----------+--
-------+
|          1|     Scott|    Tiger|1000.0| UNITED STATES| +1 123 456 7890|123 45 6789|         6789|           1|
123|
|          2|     Henry|     Ford|1250.0|         INDIA|+91 234 567 8901|456 78 9123|         9123|          91|
234|
|          3|      Nick|   Junior| 750.0|UNITED KINGDOM|+44 111 111 1111|222 33 4444|         4444|          44|
111|
|          4|      Bill|    Gomes|1500.0|     AUSTRALIA|+61 987 654 3210|789 12 6118|         6118|          61|
987|
+-----------+----------+---------+------+-------------+---------------+-----------+-------------+-----------+--
-------+
```