Day2

3 Vs: Volume, Variety, Velocity

Limitation of Row Based
Column Based File Format

functional programming

lambda function

```
double = lambda x: x * 2
```

map function

list -> list

```
# map(function, list)

items = [...]
list(map(lambda x: x ** 10, items))
```

inline function

filter function

```
# filter(bool function, list)

items = [...]
list(filter(lambda x: (x%2 != 0), items))
```

RDD

Resilient Distributed Datasets

```
hdfs dfsadmin -report
hdfs dfs -D dfs.blocksize=1048570 -put stocks.csv
hdfs dfs -ls

hdfs fsck /user/root/stocks.csv -files -blocks -locations
```

```
[root@namenode ~]# hdfs fsck /user/root/stocks.csv -files -blocks -locations

Connecting to namenode via http://namenode.example.com:9870
FSCK started by root (user=root) from /192.168.1.100 for path
/user/root/stocks.csv at Fri Jun 27 15:33:45 IST 2025

/user/root/stocks.csv <length: 157286400> <replication: 3> <blocksize:
134217728> <Complete>
   0. blk_1073741825_1001 len=134217728 repl=3 [datanode1.example.com:9866,
datanode2.example.com:9866, datanode3.example.com:9866]
   1. blk_1073741826_1002 len=23068672 repl=3 [datanode2.example.com:9866,
datanode3.example.com:9866, datanode1.example.com:9866]

The filesystem under path '/user/root/stocks.csv' is HEALTHY

------------------------------------------------
FSCK ended at Fri Jun 27 15:33:45 IST 2025 in 0 milliseconds


The filesystem under path '/user/root/stocks.csv' is HEALTHY
```

repl factor
replication factor

patitions -> logical division of a large distributed data set

```
fileRDD.getNumPartitions() # gives numner of file partitions
# 2
fileRDD.glom().collect()
```

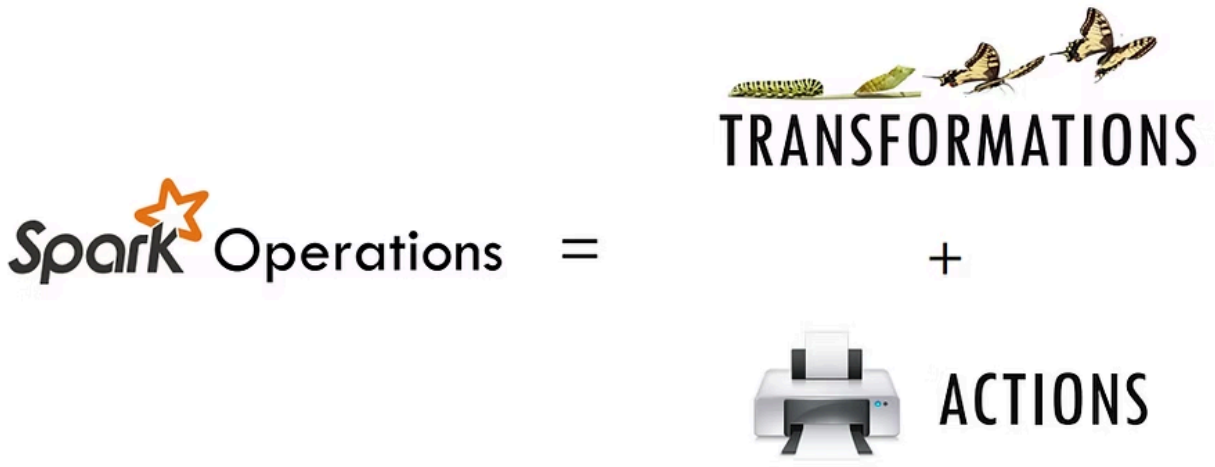
RDD collect

collect complications
partition
nodes
block locations

```
rdd = sc.parallelize(range(1, 11), numSlices = 4)
print(rdd.getNumPartitions())
print(rdd.glom().collect())
rdd.collect()
```

Operations

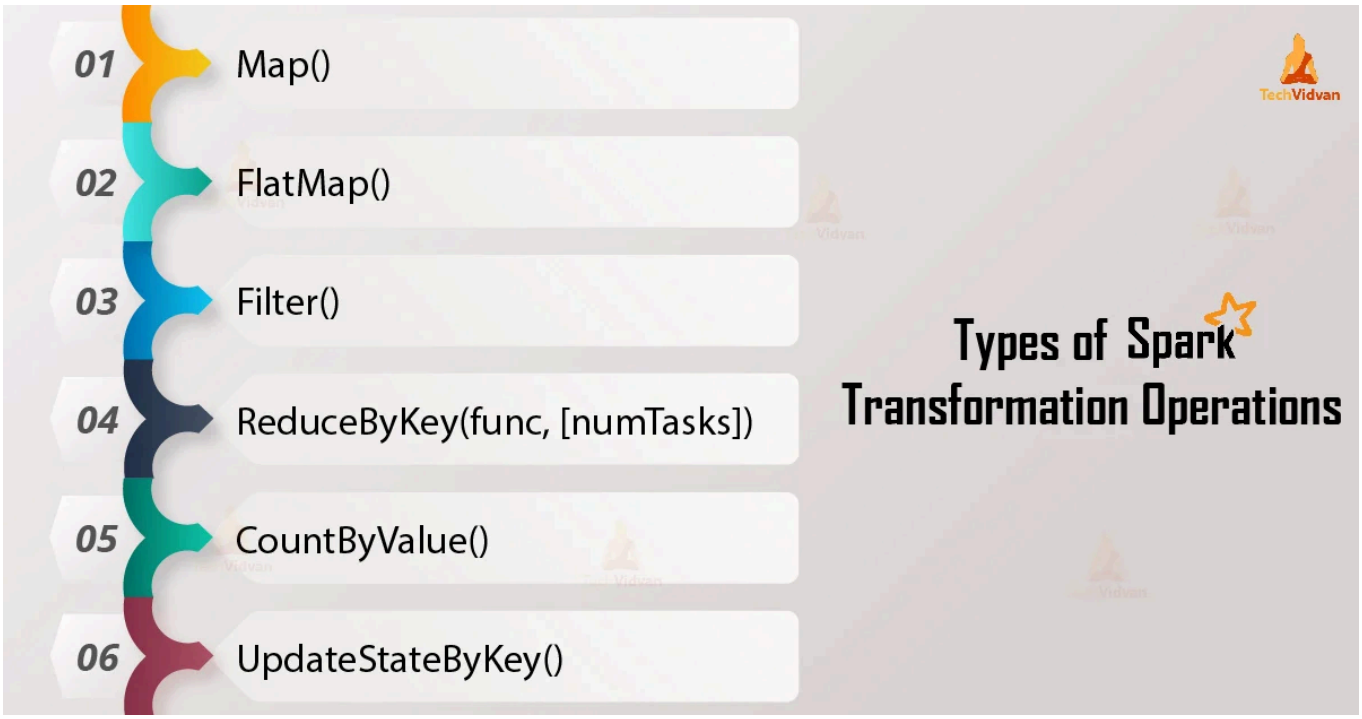
transformations, action



transformations create new RDD
actions perform

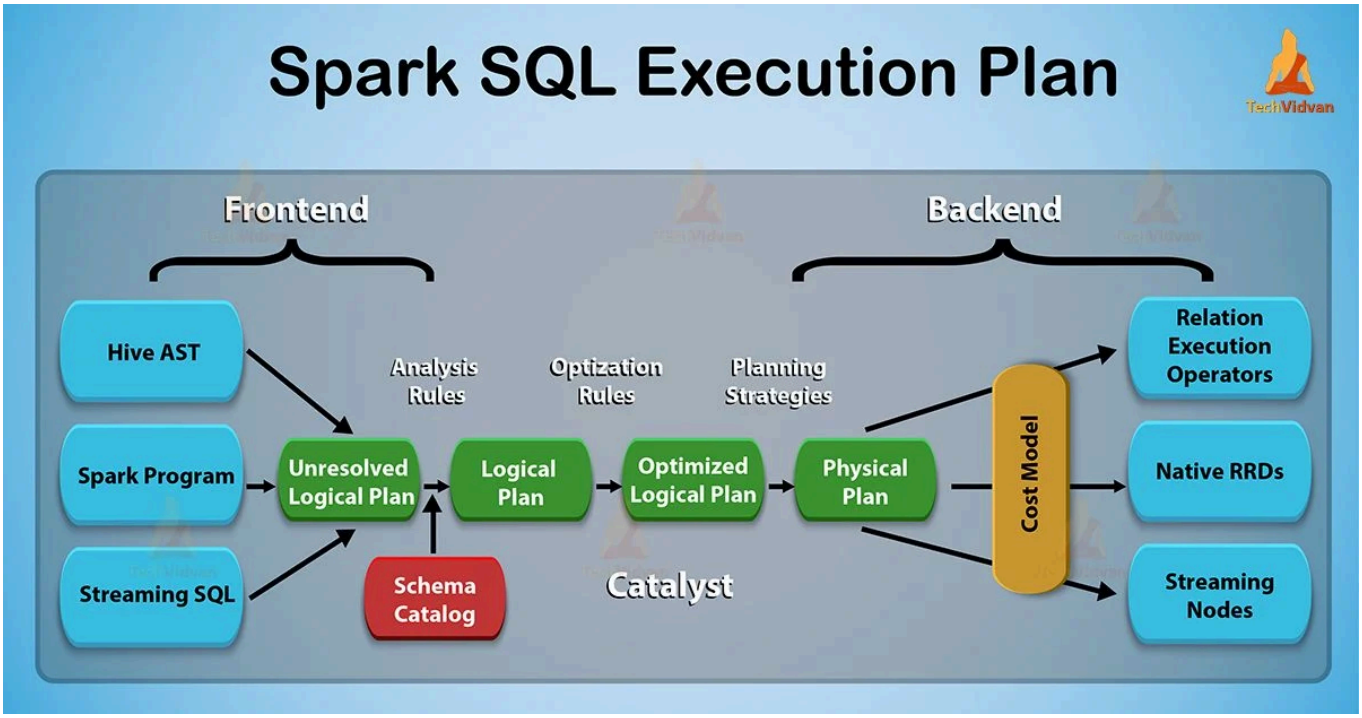
Transformations

follow Lazy evaluations
every transformations return new RDD



map, filter, flatmap, reduce, union

Execution Plan




```
>>> RDD = sc.parallelize(["hello world", "how are you"])

>>> RDD_map = RDD.map(lambda x: x.split(" "))
>>> RDD_flatmap = RDD.flatMap(lambda x: x.split(" "))

>>> RDD_map.collect()
```

```
[['hello', 'world'], ['how', 'are', 'you']]
>>> RDD_flatmap.collect()
['hello', 'world', 'how', 'are', 'you']
```

```
>>> RDD2 = sc.textFile("file:///home/talentum/log.txt")

>>> errorsRDD2 = RDD2.filter(lambda x: "ERROR" in x)
>>> warningsRDD2 = RDD2.filter(lambda x: "WARNING" in x)
>>> combineRDD2 = errorsRDD2.union(warningsRDD2)

>>> errorsRDD2.take(2)
['2025-06-28 17:35:03,421 - ERROR - db.connector - Database connection pool
exhausted. Failed to acquire connection for transaction ID: TXN_987654.',
"2025-06-28 17:35:04,701 - ERROR - mail.sender - SMTP server unreachable:
smtp.example.com:587. Email notification failed for user 'user@example.com'."]
>>> warningsRDD2.take(2)
["2025-06-28 17:35:02,123 - WARNING - auth.service - User 'guest' attempted
login from 192.168.1.55 with invalid credentials.", '2025-06-28 17:35:03,889 -
WARNING - storage.disk - Disk usage on /var/log is at 85%. Consider archiving
old logs.']
>>> combineRDD2.take(4)
['2025-06-28 17:35:03,421 - ERROR - db.connector - Database connection pool
exhausted. Failed to acquire connection for transaction ID: TXN_987654.',
"2025-06-28 17:35:04,701 - ERROR - mail.sender - SMTP server unreachable:
smtp.example.com:587. Email notification failed for user 'user@example.com'.",
"2025-06-28 17:35:06,311 - ERROR - system.health - Unresponsive service
detected: 'metrics_aggregator'. Restarting process ID 12345.", "2025-06-28
17:35:02,123 - WARNING - auth.service - User 'guest' attempted login from
192.168.1.55 with invalid credentials."]
```

Actions

collect, take, first, count

every actions return python local object

Classwork

```
cp ~/shared_folder/2_BasicRDDTransformationsandActions/* ./
```

```
### Path
D:\VM_ubuntu\pyspark\shared_folder
```