Explore how the collect() operation works in PySpark using a dataset with basic RDD operations.

In [18]:

```
sc
```

Out[18]: **SparkContext**

Spark UI

Version	`v4.0.0`
Master	`local[*]`
AppName	`PySparkShell`

In [19]:

```
# from pyspark import SparkContext

# # Initialize SparkContext
# sc = SparkContext("local", "CSV_RDD_Example")
```

In [20]:

```
# Load CSV file (assuming students.csv is in working directory)
data = sc.textFile("students.csv")
```

In [21]:

```
# Step 1: Remove header
header = data.first()
rows = data.filter(lambda line: line != header)
```

In [22]:

```
# Step 2: Split by comma
split_rdd = rows.map(lambda line: line.split(","))
```

In [23]:

```
print("=== Student Dataset (first 10 rows) ===")
for row in split_rdd.take(10):   # you can change 10 → 20, 50 etc.
    print(row)
```

```
=== Student Dataset (first 10 rows) ===
['1', 'Alice', '20', 'F', '66', '92', '44']
['2', 'Bob', '20', 'M', '82', '52', '77']
['3', 'Charlie', '22', 'F', '43', '57', '76']
['4', 'David', '19', 'M', '95', '69', '46']
['5', 'Eva', '19', 'F', '62', '44', '96']
['6', 'Frank', '22', 'F', '70', '78', '94']
['7', 'Grace', '24', 'F', '67', '66', '93']
['8', 'Henry', '21', 'F', '53', '82', '60']
['9', 'Ivy', '19', 'M', '64', '52', '46']
['10', 'Jack', '19', 'F', '44', '59', '60']
```

In [24]:

```python
# Step 3: Convert fields into structured format
# (id, name, age, gender, math, science, english)
students_rdd = split_rdd.map(lambda x: (int(x[0]), x[1], int(x[2]), x[3], int(x[4]), int(x[5]), int(x[6])))
```

In [25]:

```python
# Step 4: Calculate average marks for each student
avg_marks_rdd = students_rdd.map(lambda x: (x[1], (x[4] + x[5] + x[6]) / 3))
```

In [26]:

```python
# Step 5: Filter students who scored avg >= 75
passed_rdd = avg_marks_rdd.filter(lambda x: x[1] >= 75)
```

In [27]:

```python
# Step 6: Sort students by avg marks (descending)
sorted_passed_rdd = passed_rdd.sortBy(lambda x: x[1], ascending=False)
```

In [28]:

```python
# Step 7: Collect results to driver
results = sorted_passed_rdd.collect()
```

In [29]:

```python
# Print results
print("=== Students with Average >= 75 ===")
for student in results:
    print(f"Name: {student[0]}, Avg Marks: {student[1]:.2f}")
```

```
=== Students with Average >= 75 ===
Name: Leo, Avg Marks: 88.00
Name: Olivia, Avg Marks: 88.00
Name: Rita, Avg Marks: 86.67
Name: Kathy, Avg Marks: 81.67
Name: George, Avg Marks: 81.67
Name: Frank, Avg Marks: 80.67
Name: Oscar, Avg Marks: 80.00
Name: Uma, Avg Marks: 78.33
Name: Kyle, Avg Marks: 78.33
Name: Matt, Avg Marks: 78.33
Name: Tina, Avg Marks: 76.00
Name: Victor, Avg Marks: 75.67
Name: Grace, Avg Marks: 75.33
Name: Mona, Avg Marks: 75.00
Name: Will, Avg Marks: 75.00
```

In [30]:

```python
# Step 8: Some extra RDD operations for practice
# (a) Count how many students passed
count_passed = passed_rdd.count()
print("\nNumber of students who passed:", count_passed)
```

```
Number of students who passed: 15
```

In [31]:

```python
# (b) Find max average scorer
topper = passed_rdd.reduce(lambda a, b: a if a[1] > b[1] else b)
print("Topper:", topper)
```

```
Topper: ('Olivia', 88.0)
```

In [32]:

```python
# (c) Show first 5 passed students
print("\nFirst 5 Passed Students (via take):")
print(passed_rdd.take(5))
```

```
First 5 Passed Students (via take):
[('Frank', 80.66666666666667), ('Grace', 75.33333333333333), ('Kathy', 81.66666666666667), ('Leo', 88.0), ('Mona', 75.0)]
```

In [33]:

```python
# Stop SparkContext
# sc.stop()
```