|  |  |
| --- | --- |
| Resilient Distributed Datasets | DataFrames |
| RDDs are **low-level abstraction** that represents a **distributed collection of objects**.  They provide **fine-grained control** over the data and the operations performed on it, which **requires explicit programming** in terms of transformations (like ‘map’, ‘filter’, ‘reduce’) and actions (like ‘collect’, ‘count’) | DataFrames are **higher-level abstraction** built on top of RDDs. They represent data in a tabular format, similar to a table in a relational database, with rows and columns.  DataFrames provide a more user-friendly API and allow for declarative programming, where users can specify what they want to do with the data without worrying about how it is done. |
| RDDs provide a low-level API for distributed data processing. This API includes a variety of functions that allow for fine-grained control over data manipulation and transformations. |  |
|  |  |
|  |  |
|  |  |

* Distributed collection of objects
* RDD functions:
  + Transformations:
    - map: applies a function to each element in the RDD and returns a new RDD

A black background with white text

Description automatically generated

* + - flatMap: similar to map but each input element can be mapped to zero or more output elements (flattening the results)
    - filter: returns a new RDD containing only the elements that satisfy a predicate  
      A black background with colorful text

      Description automatically generated
    - Reduce
  + Actions:
    - Collect
    - Count
  + Persistence/Storage functions
  + Partitioning functions

Questions:

1. Can u help me understand flatmap function better?