Day3 notes

Pyspark RDD operations.

Actions/transformations performed on rdd values give us non rdd values..

The non rdd values specify that they are not stored in cluster

- .collect()-----------------------------→Collects info about RDD
- .count()------------------------------→Counts no. of elements in rdd
- .first()--------------------------------→Return the first element in RDD
- .take()--------------------------------→Print upto specific values,take(3) returns first 3 values
- .saveAsTextFile("File Name")---→Storing all the rdd data in local text file.can be viewed in catalog
- .reduce()-----------------------------→Tranformation to reduce rdd according to condition
- .map() -→returns new RDD-→transforms rdd values based on condition provided
- .filter()----→Filter data based on condition
- .FlatMap()
- .Union()-------------------→combine two rdds

Pyspark pair RDD operations

Key value pairs..similar to real world data

Pyspark Transformations in pair RDDS:

- reduceByKey()----------→*#reduce by key*
- *# It performs multiple parallel processes for each key in the data and combines the values for the same keys returns rdd as a result*

- sortByKey() ----

The .sortByKey() transformation sorts the input data by keys from key-value pairs either in ascending or descending order. It returns a unique RDD as a result.

- groupBy() ---*The .groupByKey() transformation groups all the values in the given data with thesame key together. It returns a new RDD as a result.*

Pyspark Actions in pair RDDS

- countByKey()

Selecting renaming columns from rdd:

- This PySpark script creates a Spark DataFrame with sample employee data, renames columns like "DOB" to "date of birth" and "Name" to "personname," and displays the updated DataFrame.

- This PySpark script creates a DataFrame with employee data, then uses selectExpr to rename the "Gender" column as "category," "Name" as "name," and retains other columns, displaying the final DataFrame.
- This PySpark script uses the select function with column aliasing to rename the "salary" column to "Amount" while keeping other columns unchanged, and displays the updated DataFrame.