Word count Scala:

**val** textFile **=** sc.textFile("hdfs://...")

**val** counts **=** textFile.flatMap(line **=>** line.split(" "))

.map(word **=>** (word, 1))

.reduceByKey(**\_** + **\_**)

counts.saveAsTextFile("hdfs://...")

Text Search scala code

**val** textFile **=** sc.textFile("hdfs://...")

*// Creates a DataFrame having a single column named "line"*

**val** df **=** textFile.toDF("line")

**val** errors **=** df.filter(col("line").like("%ERROR%"))

*// Counts all the errors*

errors.count()

*// Counts errors mentioning MySQL*

errors.filter(col("line").like("%MySQL%")).count()

*// Fetches the MySQL errors as an array of strings*

errors.filter(col("line").like("%MySQL%")).collect()

### Simple Data Operations

In this example, we read a table stored in a database and calculate the number of people for every age. Finally, we save the calculated result to S3 in the format of JSON. A simple MySQL table "people" is used in the example and this table has two columns, "name" and "age".

*// Creates a DataFrame based on a table named "people"*

*// stored in a MySQL database.*

**val** url **=**

"jdbc:mysql://yourIP:yourPort/test?user=yourUsername;password=yourPassword"

**val** df **=** sqlContext

.read

.format("jdbc")

.option("url", url)

.option("dbtable", "people")

.load()

*// Looks the schema of this DataFrame.*

df.printSchema()

*// Counts people by age*

**val** countsByAge **=** df.groupBy("age").count()

countsByAge.show()

*// Saves countsByAge to S3 in the JSON format.*

countsByAge.write.format("json").save("s3a://...")