DADS 6002 / CI 7301
Big Data Analytics
Spark and Data Frame Lab

Creating a DataFrame from RDD

```
# pyspark
>>> person =
    [('Anna',25,'CA'),('Jack',22,'TX'),('Tom',20,'FL'),('Bob',2
    6,'NY'),('Frank', 29, 'CA')]
>>> rdd = sc.parallelize(person)
>>> df = sqlContext.createDataFrame(rdd,
    ['name','age','state'])
# Check type of the created DataFrame and print the
    content of the DataFrame and its schema
>>> type(df)
>>> df.show()
>>> df.printSchema()
```

DataFrame Basic Operations

- show(n) – print the first n rows of a DataFrame.

  >>> df.show(5)

- count() – return the number of rows in the DataFrame.

  >>> df.count()

- select(cols) – return a new DataFrame from the list of specified columns cols.

  >>> df.select([‘name’,’age’]).show()

DataFrame Basic Operations

- orderBy(cols, ascending) – creates a new DataFrame ordered by the columns specified in cols, ascending is a boolean argument which determines the sort order, default is ascending.

  >>> df.orderBy([‘age’], ascending=False).show()

DataFrame Basic Operations

- groupBy(cols) – creates a new DataFrame containing the columns specified in cols and grouped by the columns, usually followed by the aggregate operations e.g count(), avg(), sum()

  >>> df.groupBy([‘state’]).avg(‘age’).show()

DataFrame Basic Operations

- filter(cond) – apply the given filter condition (cond) to a given DataFrame and return a result DataFrame of the filtering

  >>> df.filter('age > 25').show()

DataFrame Basic Operations

- registerAsTable(table_name) – register a given DataFrame as a table with the given table name. So we can apply SQL queries on the table.

  >>> df.registerAsTable(‘person_table’)

  >>> sqlContext.sql(‘select name,state from person_table’).show(5)

Creating a DataFrame from a text file

- Download a text file test.data from MS Teams into the shared folder and copy it to the working directory then execute the following commands.

hadoop fs –put test.data /user/cloudera

pyspark

Creating a DataFrame from a text file

```
>>> from pyspark.sql.types import *
>>> rdd =
    sc.textFile('/user/cloudera/test.data').map(lambda line:
    line.split(",")).map(lambda
    line:[int(line[0]),int(line[1]),float(line[2])])
>>> schema = StructType([ StructField( "x", IntegerType(),
    True), StructField( "y", IntegerType(), True), StructField(
    "z", FloatType(), True) ])
>>> df = sqlContext.createDataFrame(rdd, schema)
>>> df.show()
```