\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Note that SparkSession 'spark' and SparkContext 'sc' is by default available in PySpark shell.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*to check the version of spark: spark.version

Use python python 3.11.8 instead of 3.12.1 with spaerk 3.4.3

---------------------------------------------------------------------

# Create DataFrame

data = [('James','','Smith','1991-04-01','M',3000),('Michael','Rose','','2000-05-19','M',4000),('Robert','','Williams','1978-09-05','M',4000)]

columns = ["firstname","middlename","lastname","dob","gender","salary"]

df = spark.createDataFrame(data=data, schema = columns)

# Print DataFrame

df.show()

1.Reading data from a text file and displaying the first 4 elements

New\_RDD = sc.textFile("file:/home/cloudera/Desktop/detail.csv")

New\_RDD.collect() //collect() method on RDD returns list of all the elements of the RDD. It’s a great asset for displaying all the contents of our RDD

or print(New\_RDD.collect()) // using print method

New\_RDD.take(4) // to fetch few item from the RDD

...................................................................................................................................

----------------------------------------------------------------------------

@@@ filter() and map() groupByKey()

J. Creating a new RDD with flattened data and filtering out the ‘stopwords’ from the entire RDD

stopwords = ['a','all','the','as','is','am','an','and','be','been','from','had','I','I’d','why','with']

RDD1 = stopwords.filter(lambda x: x not in stopwords)

RDD1.take(4)

Filtering the words starting with ‘b’ // choose any alphabet

filteredRDD = RDD.filter(lambda x: x.startswith('b'))

filteredRDD.distinct().take(50)

....................................................................

Pyspark Dataframes Example 1: FIFA World Cup Dataset

...................................................................

Step1-Let’s load the data from a CSV file by using the method spark.read.format[csv/json].

df = spark.read.csv("file:///C:/Users/CDAC/Desktop/diabt.csv", inferSchema = True, header = True)

df.show()

.....................

Schema of Dataframe- printSchema method

fifa\_df.printSchema()

-------------------------

Column Names and Count (Rows and Column)-When we want to have a look at the names and a count of the number of

Rows and Columns of a particular Dataframe, we use the following methods.

df.columns //Column Names

df.count() // Row Count

len(fifa\_df.columns) //Column Count

...................................................

Describing a Particular Column -If we want to have a look at the summary of any particular column of a Dataframe, we use the describe method

diab\_df.describe('Age').show()

.................................................................

Selecting Multiple Columns -select particular columns from the dataframe, we use the select method.

diab\_df.select('col1','Col2','col3').show()

diab\_df.select('Outcome','Age','BMI').show()

.....................................................................................................

Selecting Distinct Multiple Columns by using Disctinct method

fifa\_df.select('Player Name','Coach Name').distinct().show()

...................................................................................................

Filtering Data -In order to filter the data, according to the condition specified, we use the filter command.

diab\_df.filter(diab\_df.MatchID=='1096').show()

#diab\_df.filter(diab\_df.Age=='31').show()

to count - diab\_df.filter(fifa\_df.Age=='1096').count()

...............................................................................................................

Filtering Data (Multiple Parameters)-We can filter our data based on multiple conditions (AND or OR)

diab\_df.filter((diab\_df.Age=='31') & (diab\_df.Outcome=='1')).show()

...........................................................................................................

Sorting data - Sorting Data (OrderBy)-

diab\_df.orderBy(diab\_df.Age).show()