Functions of Hive

String:

length( ) – returns the length of string

reverse ( ) – reverse input string

Concat ( ) – returns string or bytes after concatenating all strings

Space ( ) – returns a string with spaces

~~Split~~ ←→

upper( ) / lower( ) – converts to upper case / lower case

Date:

current_date( ) – current date is returned

current_timestamp( ) – current system time & date in format

date_diff ( ) – diff b/w specified dates

Year, Quarter, Month, Day

last_day( )

date_sub( ) – date & no of days diff

functions of list:

append() - adds element at end of list

POP() - remove at specified position

sort() - Sorts the list

reverse() - reverses the list

remove() - first item with specified value is removed

insert() - adds element at specified position

index() - returns index of first element

extend() - add elements of list to end of current list

count() - returns no. of elements with value

copy() - returns a copy of list

clear() - removes all elements from list

Namenode: has all the information such as which file is saved in which location, access-time of file, which user access a file on current time.

Secondary: stores data when namenode fails it is used to restart the namenode.
It runs on different machines for memory management.

Functions in Python pandas

1) read-csv() - helps read a comma seperated values file into pandas dataframe. Path of file to be read is mentioned.
delimiters other than comma, like or tab.

2) head() - used to return first n rows of a dataset.
df.head() - returns first 5 rows of dataframe

3) Describe() - used to generate descriptive statistics of data in pandas Df
Quick overview of dataset

4) memory-usage() - returns a pandas series having memory usage of each column in a pandas

5) astype() – cast a python object to a particular data type.

6) loc [:] – helps to access a group of rows and columns in a dataset

7) to_datetime() – convert python object to datetime format

8) value_counts() – returns Pandas series containing counts of unique values.

9) drop_duplicates() – returns Pandas dataframe removing duplicate rows

10) fillna() – helps to replace all NaN (not a number) values in a dataframe by imputing these missing values with more appropriate values.

Numpy functions:

1) np.array() – create an array in Numpy

2) np.mean() – find mean value of array

3) np.dot() – find dot products of two arrays

4) np.random.randint() – create array with random integer values blw specified range

5) np.random.rand() - create an array with random values blw 0 and 1

6) np.linspace() - create an array with a specified number of evenly spaced values.

7) np.ones() - Create array filled with ones

8) np.max() - max value in array

9) np.zeros() - Create array filled with zeros

10) np.median() - median value of array

Spark session object

1) sparksession.builder() - To create a new spark session

2) stop() - stop/end spark session

3) range() - returns single column with long type

4) conf - returns runtime config object

5) create dataframe - dataframe from collections RDD

6) createDataset() – creates dataset from collection, DataFrame & RDD

7) emptyDataFrame() – creates empty data frame

8) catalog – returns catalog object to access metadata

9) getDefaultSession() – returns default spark session returned by builder

10) getOrCreate() – creates or returns a Spark Context

Functions in RDD API's

1) Transformation functions:

Map() – Applies transformation function to each element of RDD & returns a new RDD

filter() – filters elements based on a given condition and return a new RDD containing filtered elements

flatmap() – applies a transformation function that returns an iterator for each element & flattens results into new RDD

distinct() – returns a new RDD containing distinct elements from original RDD

SortBy() – Sorts elements of RDD based on specified Criteria & returns a new RDD

2) Action functions:

Collect() - returns all elements of RDD as an array to driver program

count() - returns no. of elements in RDD

reduce() - Aggregates elements of RDD using a specified function.

take() - Returns the first N elements from RDD as an array

foreach() - applies as function to each element of RDD

3) Pair RDD functions:

reduceByKey() - Performs a reduction operation on values of a RDD

groupByKey() - groups values of a pair RDD based on key based on key

sortByKey() - sorts elements "

join() - Performs an inner join b/w two pair RDD's based on

Cogroup() - groups the values of multiple pair RDD's sharing same key

4) Persistence functions:

cache() - Persists RDD in memory for faster future access

persist() - Allows specifying different storage levels for persisting RDD

unpersist() - Removes RDD from memory/ disk storage

5) Input/output functions:

textfile() - Reads a textfile and converts it into an RDD of strings

saveAsTextFile() - writes the contents of RDD to text file

Functions in Spark APIs

1) Data Loading & I/P:

spark.read.csv() - read csv files into DF
spark.read.parquet() - read parquet files into DF
spark.read.json() - read ~~text files~~ json files into DF
spark.read.text() - read text files into DF
spark.read.jdbc() - read data from JDBC data to use into DF

2) Data-frame Operations:

Data frame.select() - selects specific columns from DF

Data frame.filter() - filters row based on a condition

Dataframe.withcolumn() - Adds or replaces a column in DF

Data frame.groupBy() - groups data based one or more columns

3) Aggregation & window functions:

Dataframe.agg() - applies aggregation functions to grouped data

Data frame.groupBy().agg() - Aggregates data based on grouping

Pyspark.sql.functions.sum() - computes sum of a column

Pyspark.sql.functions.count() - Counts no. of rows or non-null values in column

Pyspark.sql.functions.avg() - computes average of column

Pyspark.sql.functions.rank() - computes rank of a row within a partition

4) Data Transformation & cleaning:

Dataframe.withColumnRenamed() – renames column in Df

Dataframe.drop() – drops specified column from Df

Dataframe.fillna() – fills missing values in a Df with a specified value

Dataframe.na.drop() – drops rows with missing value from Df

Dataframe.sql.functions.when() – Applies conditions transformations to a column

5) SQL Queries:

~~Dataframe.withcolumnRenamed() – Renames column in Df~~

Spark.sql() – executes SQL queries on registered tables or DF

Dataframe.CreateOrReplaceTempView() – Registers a Df as temporary table

spark.catalog.listTables() – lists the tables available in spark catalog

spark.catalog.refreshTable() – Refreshes metadata of a table in spark catalog