Date functions in MySQL :-

NOW() :- Returns current date and time

CURDATE() :- " current date

(CURTIME() :- current time)

DATE() :- extract the date part of a date (or) date/time expression

EXTRACT() :- Returns single part of a ~~date~~ (or) date/time

DATE ADD() :- Adds a specified ~~part~~ time interval to a date

DATE SUB() :- subtracts a " time interval from date

DATE DIEF() :- Returns the number of days b/w two dates

DATE format() displays date/time data in different formats

TO-DATE :- Converts a character string to a date

Year :- Extracts the year component from a date

Month :- " month " " " "

STRING FUNCTION :-

LENTH
LOWER
UPPER
TRIM
SUBSTR

CONCAT :- (str1,2 --) concatenates

SUBSTR (str, start, length) :- Extracts a substring from a string starting at the specified position and with the specified length

SPLIT :- (str, delimiter) :- splits a string into an array of substrings based on a delimiter.

REPLACE (str, search, Replace) replaces all occurrences of a specified substring with another substring in string.

REGEXP - REPLACE (STR, pattern, replace) Replaces all occurrences of specified regular expression pattern with a replacement string in a string.

INSTR (str, substr) :- Returns the position of the first occurrence of a substrings with in a strang.

CONCAT - WS (delimiter, str1, str2) :- concatenates two (or) more strings together with a specified delimeter b/t them

INITCAP(str) :- Convert the first character of each word in a string to uppercase and the lower case

TRIM :- Removes leading and trailing white space from a string

Reverse(str) :- Reverse the characters in string.

TRANSLATE :- (str, ~~str,str~~ from, to) :- Replaces character in a string based on

on a mapping defined by two strings.
FIND-IN-SET (Str, StrList) :- Returns the position of string with a comma-separated list of strings.

different function in ~~Pyth~~ Python :-

print() :- outputs a specified message or value to the console.
len() :- Returns the length (number of items) of an object, such as a string, list, or tuple.
type() :- Returns the type of an object
range() :- generates a sequence of numbers within a specified range.
input() :- prompts the user to enter input from the console
int() :- Converts a value to an integer data type.
str() :- Converts " " string " "
list() :- " " iterable object into a list
max() :- Returns from largest item from a collection of items
min() :- " " " smallest " " " "
sum() Returns the sum of all items in an iterable
abs() :- Returns absolute value of a number.
round() :- Rounds a number to specified number of decimal places.
zip() :- Combines multiple iterables into a single iterable of tuples.
map() :- Applies a given function to each item in an iterable and returns an iterator of the results.
filter() :- Returns an iterator containing only the items from an iterable that satisfy a given condition.
open() :- opens a file for reading, writing, or both
dir() :- Returns list of names in current namespace
enumerate :- Returns iterator that generates tuples containing index and value of each item in an iterable

different function of PANDAS API :-

1. Data loading and input :-
   read_csv() :- Read csv (comma-separated) files into a Data frame.
   read_excel() :- Read Excel files into a Data frame.
   read_sql() :- " SQL query or database table into a Data frame.
   read_json() :- Read JSON files into a Data frame (JavaScript object notation)

2. Data exploration & manipulation :-
   head() :- Returns the first n rows of a Data frame
   tail() :- " " last " " " " "
   info() provide summary of a Data frame's structure and data types.
   describe() :- generates descriptive statics of DF
   columns() :- Returns column labels of DF
   unique() :- " Unique values in a column
   sort_values() :- Sorts a DF by one or more columns

3. Data selection and filtering :-
   loc[] :- Accesses row & columns by labels
   iloc[] :- " " " " integer based indexing
   isin() :- filters row based on wether a column value is in given list.
   query() :- filters row using a Boolean expression.

4. Data Aggregation and grouping :-
   groupby() :- Groups data based on one or more columns
   agg() :- Applies aggregation functions to grouped data
   pivot_table() :- creat spreadsheet-style pivot table based on grouped data.

5- Data cleaning and Transformation :-

fillna() :- fills missing values in a Data frame with a specified method.

drop_duplicates() :- Removes duplicate row from Data frame.

replace() :- Replaces values in DF with specified values.

apply() :- Applies a function to each element of row column of a Data frome.

6- Data visualization :-

plot() :- generates various types of plots, such as line plots, bar plots, etc.

hist() :- creates a histogram from a Data frame or series.

boxplot :- creates a box plot from a DF or series.

DF in the NUMPY API :-

1. Array creation :-

numpy.array() :- creats an array from a python list or tuple.

numpy.zeros() :- " an array filled with zeros.

numpy.ones() :- creats an array filled with ones.

numpy.empty() :- creates an uninitialized array

numpy.arange() :- creates an array with evenly spaced values.

numpy.linspace() :- creates an array with evenly spaced values a specified range.

numpy.random.rand() :- creates an array of random numbers b/w 0&1.

2. Array manipulation :-
numpy.reshape() :- Reshapes an array into a specified shape.
numpy.concatenate() :- Concatenates arrays along a specified axis.
numpy.split() :- splits an array into multiple sub-arrays
numpy.transpose() :- Transposes an array.

3. Mathematical function :-
numpy.sum() :- computes the sum of array elements
numpy.mean() :- computes the arithmetic mean of array elements
numpy.min() :- finds the minimum value in an array
numpy.max() :- " " maximum " "
numpy.sin(), 'numpy.cos()', 'numpy.tan()', compute trigonometric functions.
numpy.exp() : computes the exponential of all elements in array.
numpy.log() :- computes the natural logarithm of all elements in an array.

4. Array operations :-
numpy.dot() : computes the dot product of 2 arrays
numpy.transpose() :- Transposes an array.
numpy.sort() :- sorts the elements of an array.
numpy.unique() :- finds the unique elements in an array.
numpy.argmax() :- Returns the indices of the maximum value along a specified axis

5. Random Number Generations :-
numpy.random.randint() : Generates random integers
numpy.random.randn() : Generates an array of random numbers from the standard normal distribution.
numpy.random.choice() :- Generates random samples from a given 1-D array.

numpy.random.shuffle() : shuffles the elements of an array randomly.

function of Spark :-

1. Data loading and input :-
   - spark.read.csv() :- Reads csv files into a Data frame
   - spark.read.parquet() :- Reads parquet files into a Data frame
   - spark.read.json() :- " JSON " " " "
   - spark.read.text() :- Read text files into DF
   - spark.read.jdbc() :- Reads data from a JDBC data source into Data frame.
2. Data frame operations :-
   - Data frame.select() :- selects specific columns from a DF
   - Data frame.filter() :- filters rows based on a condition
   - Data frame.withcolumn() :- Adds & replaces a column in DF
   - Data frame.groupBy() :- Groups data based on one & more columns
   - Data frame.join() :- Joins two Data frames based on a column & expression.
   - Data frame.sort() :- sorts the Data frame by one & more columns.
3. Aggregation and window functions :-
   - Data frame.agg() :- Applies aggregation functions to grouped data.
   - Data frame.groupBy().agg() :- Aggregates data based on grouping.
   - PySpark.SQL.functions.sum() :- Computes the sum of a column.
   - PySpark.SQL.functions.count() :- Counts the number of rows & non null values in column
   - PySpark.SQL.functions.avg() :- Computes the average of column
   - PySpark.SQL.functions.rank() :- Computes the rank of a row with in partition.

4. Data Transformation and cleaning :-
Data frame.withColumnRenamed() :- Renames a column in a Data frame.
Data frame.drop() :- Drops specified columns from a dataframe
Data frame.fillna() :- fills missing values in Data frame with specified value.
Dataframe.na.drop() :- Drops rows with missing values from a Data frame.
PYSpark.SQL.functions.when() : Applies conditional transformations to Column.

5. SQL queries :-
Spark.sql() :- executes SQL queries on registered tables or Data frames.
Data frame.create or Replace Temp View() - Registers a Data frames as temporary table.
Spark.catalog.list tables() :- lists the table available in the spark catalog.
Spark.Catalog.refresh table() :- Refreshes the metadata of a table in spark catalog.
spark.catalog.refresh table() : Refreshes the meta data of a table in the Spark catalog.

6. output and storage :-
Data frame.write.csv() :- writes a Dataframe to CSV files.
Data frame.write.parquet() :- writes a Data frame to Parquet files.
Data frame.JDBC() :- writes a Data frame to a JDBC Datasource
Data frame.write.JSON() :- " " " " JSON files

* functions in the RDD API :-

1. Transformation functions :-

map() : Applies a function to each element and returns a new RDD

filter() :- filter the elements based on a condition

flatmap() :- Applies a function to each element and and Returns a new RDD by flattening the results

union() :- Returns the union of two RDDs.

distinct() : Removes duplicates and returns a new RDD

sortBY() :- sorts the RDD elements based on key

groupBYKey() :- Groups the elements by key.

reduceBYKey() :- Aggregation the values of each key

Join() :- Join two RDDs based on a common key

coalesce() :- Reduces the number of partitions scheme of the RDD

Action functions :-

collect() :- Returns all the elements of the RDD as an array.

count() :- Returns the number of elements in the RDD

first() : Returns the first elements of the RDD

take() :- Returns the first n elements of the RDD

reduce() :- Reduces the elements of the RDD using a binary operator.

for each :- Applies a function to each element of the RDD

Pair RDD functions :-

Key() :- Returns an RDD of the keys

values() :- " " " values

SortBYKey() :- sorts the RDD elements by key

ReduceBYKey() :- Aggregates the values of each key

groupBYkey() :- groups the elements by key.

persistence and caching functions:

cache() :- caches the RDD in memory for faster access

persist() persists the RDD in memory or disk storage.

unpersist() : Removes the RDD from memory (or) disk storage.