⇒ Date functions in my SQL:

Now(): Returns Current date and time
Curdate(): Current date
Date(): Extract the date part of a date (or) date/time Epression.
Extract(): Returns Single part of a date/time.
Date Add(): Adds a specified time interval to a date.
Date Sub(): Subtracts a specified time interval to a date.
Date diff(): Returns the number of day 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(): Extracts the month Component from a date
Day(): Return a day from date

⇒ String function:

Concat: (str1, ...) Concakenates
Substr (str, start, length): Extracts a Substring from a string Starting at the Specified position and with the Specified
Split: (str, delimiter) Splits a string into an array of Substrings based on delimition
Replace (str search, Replace): Replaces all occurences of a specified Substring with another Substring in String.

Regexp - Replace (str, pattern, replace): Replaces all occurences of specified regular expression pattern with a replacement string in a string.

Instr (str, Substr): Returns the position of the first occurence of a substrings with in a string.

Concat - ws (delimites, str1, str2): Concatenates two (or) more strings together with a specified delimeters b/w them.

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

Trim: Removes leading and training white space from a string

Reverse (str): Reverse the characters in string.

Translate (str, from, to): Replaces character in a string based 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 python:

print (): outputs a specified message or value to the Console.

len (): Returns the length (number of length) of an object such as a string list or tuple.

type (): type of an object

range() : generates a sequence of numbers with a specified range.

input() : prompts the user to enter input from the console

int() : Converts a value to an integer data type.

Str() : Converts a value to an string data type.

list() : Converts iterable object into a list

max() : Returns from largest item from a collection of values.

min() : Returns from smallest item from a collection of values.

Sum() : Returns the sum of all items in an iterable

abs() : Returns the absolute value of a number.

round() : Rounds a number to specified number decimals places.

zip() : Combines multiple iterables into a single iterable ~~and returns an iteration~~ of the tuples

map() : Applies a given functions to each item is an iterable and returns an iterator of the results.

filter() : Returns an iteration. ~~Geo~~ it is a function that allows you to filter elements from an iterable based on a specified condition. (built-in function)

open() : it is used to create a file object, which allows you to read, write, or append to the file.

dir : Returns list of names in current namespace.

Enumerate : ~~Returns~~ it allows you to itrate over a sequence (Such as a list, tuple or String) while keeping

track of the index of each item. It provides a convenient way to iterate over both the index and value of each item in a sequence.

⇒ Different function of pandas API:

1. Data loading and inputs:

read_csv() : Read csv (comma separated) files into a Data frame.
read_excel() : Read Excel files into a Dataframe.
read_sql() : Read SQL query or database table into a Dataframe.
read_json() : Read (Json) files into a Dataframe (Javascript object notation)

2. Data Exploration & manipulation:

head() : Returns the first n rows of a Dataframe
tail() : Returns the last n rows of a Dataframe
info() : provide summary of a Dataframe's structure and datatypes.
describe() : Generates description statistics of D.F
columns() : Returns column labels of D.F
unique() : Returns unique values in a column.
sort_values() : Sorts a D.F by one or more columns.

3. Data Selecting and filtering:

loc[]: Accesses rows & columns by labels
iloc[]: Accesses rows & columns by integer based indexing.
isin(): filters rows based on whether a column value is in given list.
query(): filters rows 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(): create spreadsheets-style pivot table based on grouped data.

5. Data cleaning and Transformation:

fillna(): fills missing values in a Dataframe with a specified method.
drop_duplicates(): Removes duplicate rows from D.F.
replace(): Replaces values in D.f with specified values.
apply(): Applies a function to each element / row column of a D.F.

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 boxplot from a. DF or series.

⇒ ~~Df in the numpy~~ API (Spark session):

I. Spark Session.builder(): API is used to create a new Spark session
Eg: Sparksession.builder().appName("App")

II. Spark.catalog() : It is used to interact with catalog in Spark which include databases, tables and function.
Eg: Spark.catalog.listTables().show()

III. Spark.read() : It is used to read data from various source and create a D.f or Dataset.
Eg: spark.read.format("csv")

IV. Spark.write() : used to write data to various formats and series.
Eg: Spark.write.csv(path)

v. Spark.sql → used to allow to execute sql queries on loaded data. we can also run sql on dataframe using spark SQL.

vi. Spark.range → generates a D.f with - single column containing elements from "start" to "end" with a step value.

Eg: Spark.range (1, 10, 2).

vii. Spark.session.getOrCreate() → if a spark session already exits this method returns the existing spark session. otherwise it creates a new one.

Eg: sparksession.getOrCreate().

viii. Spark.udf.register ⇒ registers a user defined function (udf)

ix. Spark.Conf.set → used to set a Configuration property.

x. Spark.stop() → used to stop the spark session and its resources.

Eg: Spark.stop()

⇒ RDD API:

1. map(func) → Applies a transformation function to each element of the RDD and return a new RDD
2. flatMap(func) → Similar to map(), but each input item can be mapped to zero or more output items.
3. filter(func) → Returns a new RDD containing only the elements that satisfy the given predicate function.
4. Union(other RDD) → Return a new RDD that contains the union of element from the RDD and another RDD.
5. intersection(other RDD) → Return a new RDD that contain the intersection of elements b/w the RDD and another RDD.
6. Subtract(other RDD) → Return new RDD the contain elements present in the RDD but not in another RDD.
7. collect() → Return all elements of the RDD as an array.
8. count() → Return the number of elements in the RDD.
9. take(n) → Return the first n elements of the RDD.

—— ∘ ——

⇒ Numpy: import numpy as np.

1. np.array → Create a Numpy array from a python list
   Eg: arr = np.array([1,2,3,4,5])
   print(arr) # [1,2,3,4,5]

2. np.zeros() → Create an array filled with zeros.
3. np.ones() → Create an array filled with ones
   Eg: arr = np.ones((1,2,5))
   print(arr) # [[1. 1. 1.]
   [1. 1. 1.]]
4. np.arange() → Creates an array with a sequence of number.
5. np.linspace() → create an array with evenly spaced values.
6. np.random.rand() → generates an array of random numbers from a uniform distribution.
7. np.max() → Returns the max value in an array.
8. np.min() → Returns the min value in array
9. np.mean() → Compute the mean of array.
10. np.sum() → Sum of all element in array.

— o —

Numpy used to work with arrays

— o —

⇒ String:

1. Concat (string1, string2....) → It Concatenates two or more strings together.
   Eg: Select Concat ('Hello', ' ', 'World') as Concatenated_string;
2. Length (string) → Returns the length of the input string.
   Eg: Select length ('Hive') as string_length;
3. Lower (string) → Convert all characters in the string to lowercase.
   Eg: Select lower ("Hello") as lowercase_string;
4. Upper (string) → Convert all characters in the string to uppercase.
5. Trim/string → Removes leading and trailing white spaces from the string.
6. Substr (string, start, length) → Extracts a substring from the input string, starting at the specified index and with the specified length.
   Eg: Select Substr ('Hive', 2, 4) as Substring;
7. Replace (string, search, replace) → Replaces all occurences of a search string with a replacement string within the input string.

8. locate (search str, string, start) → Returns the starting position of a substring with a string.

eg: select locate ('world', greeting).

—— o ——