* Date functions in SQL

1) NOW() → Returns the current date & time
2) CURDATE() → Returns the current date
3) DATE() → Extracts the date part of a date or date/time expression
4) DATE_ADD() → Adds a specific time interval to a date
5) DATE_SUB() → Subtracts "
6) DATEDIFF() → Returns the number of days between two dates
7) DATE_FORMAT() → Displays date/time data in different format
8) EXTRACT() → Returns a single part of a date/time
9) TO_DATE() → Converts a character string to a data
10) YEAR → Extracts year component from a date
11) MONTH → " month "

* String functions in SQL:-

1) CONCAT(str1, str, ...) → Concatenates two or more strings together
2) LENGTH(str), LOWER(str), UPPER(str), REVERSE(str).
3) TRIM(str) → Removes leading & trailing whitespaces from a string

4) SUBSTR (str, start, length) → Extracts a substring from a string, starting at the specified position & with the specified length

5) SPLIT (str, delimiter) → Splits a string into an array of substrings based on a delimiter.

6) REPLACE (str, search, replace) → Replaces all occurences of a specified regular expression pattern with a replacement string in a str

7) INITCAP (str) → Converts first character of each word in a string to uppercase & the rest to lowercase.

8) TRANSLATE (str, from, to) → Replaces characters in a string based on a mapping defined by two strings.

9) STRPOS (str, substring) → Returns the starting position of a substring within a string.

* 10 different functions from python:-

1) print(), type(), min(), max(), sum(), round(), open() (open file for reading writing both)
2) len() → Returns the length of an object, such as string, list or tuple
3) range() → Generates a sequence of numbers within a specified range
4) input() → Prompts the user to enter input from the console.
5) int() → Converts a value to an integer data type
6) str() → " " " " " string " "
7) list() → Converts an iterable object into a list.
8) abs() → Returns the absolute value of a number (neglects +ve -ve) x=10, abs_x=10
9) filter() → Returns iterator containing items from iterable that satisfies a specific condition.
10) dir() → Returns list of names in current namespace or specified object
11) enumerate() → Returns iterator that generates tuples containing index & value of each item in an iterable.
12) zip() → Combines multiple iterables into a single iterable of tuples
13) sorted() → Returns a sorted version of given iterable

* functions & methods in Pandas API's

1) Data Loading & I/P :-
- 'read_csv()' → Read csv (comma separated values) files into a Dataframe
- 'read_excel()' → Read excel files into a Dataframe
- 'read_sql()' → Read SQL query or database table into a Dataframe
- 'read_json()' → Read JSON (Javascript Object Notation) files into a DF

2) Data Exploration & Manipulation :-
- 'head()' → Returns first n rows of a Df
- 'tail()' → " last " " "
- 'info()' → provides a summary of a Dataframe's structure & data types
- 'describe()' → Generates descriptive statistics of a Df.
- 'shape' → Returns dimensions (rows, columns) of a Df
- columns → Returns column labels of a Df.
- unique() → Returns unique values in a column
- 'sort_values()' → Sorts a Df by one or more columns.

3) Data Selection & filtering :-
- loc[] → Accesses a group of rows & columns by labels.
- iloc[] → " " " " " " 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 & Grouping :-
- 'groupby()' :- Groups data based on one or more columns
- 'agg()' :- Applies aggregation functions to grouped data
- 'pivot_table()' → Creates a spreadsheet-style pivot table based on grouped data

5) Data Cleaning & Transformation :-
- 'fillna()' → fills missing values in a Df with specified method
- 'drop_duplicates()' → Removes duplicate rows from a Df
- 'replace()' → Replaces values in Df with specified values
- 'apply()' → Applies function to each element or row/column of a Df.

6) Data Visualization :- 'plot()' → Generates various types of plots
'hist()' → Creates a histogram from Df
'boxplot()' → Creates boxplot from a Df

* functions in Numpy API's :-

1) Array Creation :-

numpy.array() → Creates array from python list or tuple
numpy.zeros() → " " filled with zeros
numpy.ones() → " " " " ones
numpy.empty() → Creates uninitialized array
numpy.arange() → Creates an array with evenly spaced values
numpy.linspace() → " " " " over a specified range
numpy.random.rand() → " " " of random numbers b/w 0 & 1.

2) Array Manipulation :-

numpy.reshape() → Reshapes an array into specified shape.
numpy.concatenate() → concatenates arrays along a specified axis
numpy.split() → Splits an array into multiple sub-arrays
numpy.transpose() → Transposes an array.
numpy.flatten() → flattens a multi-dimensional array into 1D array
numpy.resize() → Resizes an array to a specified shape
numpy.append() → Appends values to the end of an array

3) Mathematical functions :-

numpy.sum() → computes sum of array elements
numpy.mean() → " arithmetic mean of array elements.
numpy.min() → finds min value in array
numpy.max() → " max " " "
numpy.sin(), numpy.cos(), numpy.tan() → Compute trigonometric funcs
numpy.exp() → computes the exponential of all elements in an array
numpy.log() → computes the natural logarithm of all elements in array

4) Array Operations :-

numpy.dot() → computes dot product of two arrays.
numpy.transpose() → transposes an array
numpy.sort() → sorts elements of an array.
numpy.unique() → finds unique elements in an array.
numpy.argmax() → Returns indices of max values along specified axis

5) Random Number Generation :-

numpy.random.randint() → Generates random integers.
numpy.random.randn() → Generates an array of random numbers from standard normal distribution
numpy.random.choice() → Generates random samples from given 1D array
numpy.random.shuffle() → Shuffles elements of an array randomly.

* functions in Spark APIs :-

1) Data Loading & IO :-
- spark.read.csv() → Reads CSV files into a DF
- spark.read.parquet() → " Parquet " " "
- spark.read.json() → " JSON " " "
- spark.read.text() → " text files into " "
- spark.read.jdbc() → " data from JDBC data source into a DF.

2) Dataframe Operations :-
- Dataframe.select() → Selects specific columns from a DF
- Dataframe.filter() → filters rows based on a condition
- " .withColumn() → Adds or replaces a column in a DF.
- " .groupBy() → Groups data based on one or more columns
- " .join() → Joins two Dataframes based on a column or expression
- " .sort() → Sorts the " by one or more columns

3) Aggregation & Window functions :-
- Dataframe.agg() → Applies aggregation functions to grouped data
- Dataframe.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 a column.
- pyspark.sql.functions.rank() → " rank of a row within a partition.

4) Data Transformation & Cleaning :-
- Dataframe.withColumnRenamed() → Renames column in a DF.
- " .drop() → Drops specified columns from a DF.
- " .fillna() → fills missing values in a DF with a specified value.
- " .na.drop() → Drops rows with missing values from a DF.
- " sql.functions.when() → Applies conditional transformations to a column

5) SQL Queries :-
- spark.sql() → Executes SQL queries on registered tables of DF.
- Dataframe.createOrReplaceTempView() → Registers a DF as a temporary table.
- spark.catalog.listTables() → Lists the tables available in Spark Catalog
- spark.catalog.refreshTable() → Refreshes metadata of a table in the Spark Catalog

6) Output & Storage :-
- Dataframe.write.csv() → Writes a DF to CSV files
- " " .parquet() → " " " Parquet files
- " " .jdbc() → " " to JDBC data source
- " " .json() → " " " JSON files

* functions in RDD APIs :-

1) Transformation functions :-

- map() → Applies a transformation function to each element of RDD & returns a new RDD.
- filter() → filters elements based on a given condition & return a new RDD containing filtered elements
- flatMap() → Applies a transformation function that returns an iterator for each element & flattens results into a new RDD.
- distinct() → Returns a new RDD containing distinct elements from original RDD.
- sortBy() → sorts the elements of RDD based on a specified criterion & returns a new RDD
- union(), intersection(), subtract() → Performs set operations b/w two RDDs.

2) Action functions :-

- collect() → Returns all the elements of RDD as an array to driver Program
- count() → Returns the number 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 a function to each element of RDD.

3) Pair RDD functions :-

- reduceByKey() → performs a reduction operation on the values of a Pair RDD based on the key
- groupByKey() → Groups values of a pair RDD based on key
- sortByKey() → Sorts elements " " " "
- join() → performs an inner join between two pair RDDs based on key.
- cogroup() → Groups the values of multiple pair RDDs sharing same key.

4) Persistence functions :-

- cache() → persists the RDD in memory for faster future access.
- persist() → Allows specifying different storage levels for persisting RDD.
- unpersist() → Removes the RDD from memory / disk storage.

5) Input & Output functions :-

- textFile() → Reads a text file & converts it into an RDD of strings
- saveAsTextFile() → writes the contents of an RDD to a text file.