Date functions in SQL

1) GetDate(): Returns current date and time.
2) current_timestamp: Returns current date and time
3) Datepart(part, date): extracts specific part of date or time value
4) Year(date): Returns year from date value.
5) Month(date): Returns month from a date value.
6) Day(date): Returns day from a date value.
7) DateAdd(interval, num, date): ~~Returns day from a date value.~~ Add specific no. of intervals to a date.
8) Datediff(interval, start date, end date): ~~Adds a spe~~ Calculate the diff between two dates in specified interval.
9) Convert(data_type, expression, style): Convert a date value to a specified data type.
10) Format(date, format): formats a date value using a specified format.

String fun:

1) len(str): length of str.
2) upper(str): upper
3) lower(str): lower
4) left(str, len): Return specified num of characters from left side of string.
5) Right(str, len): " " " " Right side of str
6) substring(str, start, len): Returns a substr from a str starting at a specified position with a specified length.

7) Replace(str, old, new) : Replaces all occurances of substr with new substr in the str.

8) Concat (str1, str2) : Concates.

9) Trim (str) : Removes leading and trailing spaces from a str.

10) Charindex(substr, str) : Returns the starting position of a substring within a string.

Pandas API fun

1) Pd.read_csv() : Reads data from csv file and creates dataframe

2) df.head() : Returns the first n rows of a dataframe (default=5)

3) df.tail() : Returns last n rows of dataframe

4) df.shape : Returns the dimensions (row, col) of a dataframe.

5) df.info() : Provides a concise summary of dataframe.

6) df.describe() : Gerelate descriptive statistics of a dataframe include count, min, max etc

7) df.columns : Returns the column labels of a dataframe.

8) df.loc[] : Access a group of rows and columns by label(s) or a boolean array.

9) df.iloc[] : Access a group of rows and cols by integer position

10) df.groupby() : Groups data in a dataframe based on specific column

11) df.merge() : Combines 2 dataframes based on common columns

12) df.sort_values() : Sorts dataframe by one or more columns.

13) df.dropna() : Drop rows or columns with missing vals from dataframe.

14) df.fillna() : fills missing values in a dataframe with specified value.

15) df.to_csv() : Writes a dataframe to a csv file.

NumPy API fns:

1. np.array(): Creates a numpy array from a python list or tuple
2. np.zeros(): Creates an array filled with zeros.
3. np.ones(): " " " " ones
4. np.arange(): Returns an array with evenly spaced values within a specified range.
5. np.linspace(): Creates an array with specified number of evenly spaced values within a specified range.
6. np.random.rand(): Generates an array of random numbers between 0 and 1.
7. np.random.randint(): Generates an array of random integers within a specified range.
8. np.shape(): Returns dimensions of an array.
9. np.reshape(): Reshapes an array to specified range.
10. np.mean(): Computes mean of array.
11. np.sum(): " sum " " elements
12. np.max(): Max value in a array.
13. np.min(): min " " " "
14. np.dot(): Computes dot product of two arrays.
15. np.transpose(): Transposes a array.

RDD API fns

1) Creation function:

* SC.parallelize(): creates an RDD from a local collection
* SC.textFile(): Reads a text file and creates an RDD where each line repeats an element.

2) Transformation function:

* map(): Applies a transformation function to each element of RDD and retuen a new RDD.
* filter(): filters the elements of RDD based on a given Predicate function.
* flatmap(): Applies a transformation function that returns an iterator for each element and flattens the results.
* union(): combines two RDD's in a single RDD
* distinct(): Returns a new RDD with distinct elements
* SortBy(): Sorts the RDD elements based on a given key.
* groupByKey(): Groups the values of each key in the RDD
* reduceByKey(): Reduces the values of each key using a Specified reduction function.
* join(): Performs a inner join between two RDD's based on a common key.

3) Action function:

* Count(): Returns the num of elements in RDD
* collect(): Returns all elements of RDD as an array to driver program
* first(): Returns the first element of RDD.
* take(): Returns the first n elements of RDD.

* reduce(): Applies a reduction function to aggregate the elements of RDD.
* foreach(): Applies a function to each element of RDD.

4. Persistence functions:-
* Cache(): persists the RDD in memory.
* persist(): persists the RDD with a specified storage level.
* unpersist(): Removes the RDD from memory.

5. Partitioning functions:-
* repartition(): Repartitions the RDD into a specified number of partitions.
* coalesce(): Reduces the number of partitions in RDD.

6. Other functions:-
* sample(): Returns a random sample of RDD.
* foreachPartition(): Applies a function to each partition of RDD.
* glom(): Returns RDD where each element is an array of elements from the original RDD partition.

Spark Session object:-

It is an entry point and main interface for interacting with Apache Spark. It provides unified API to creating and working with Spark functionality.

important methods and properties of SparkSession object:-

Creating a SparkSession:

* SparkSession.Builder: creates a SparkSessionBuilder instance to configure and Build a SparkSession.
* appName(name: String): Sets the name of Spark application
* master(master: String): sets the master URL for connecting to the spark cluster.
* Config(key: String, value: String): Sets a configuration property for Spark.
* getOrCreate(): Returns an existing SparkSession instance or create a new one if it doesn't exist.

Accessing properties:-

* SparkContext: Returns the underlying 'SparkContext' object
* Catalog: provides access to catalog metadata, such as tables and databases.
* Conf: Returns the 'SparkConf' object, which holds Spark configuration properties.
* SessionState: Returns the 'SessionState' object, which manages the state of 'SparkSession'.

Working with dataframes and datasets:

* CreateDataframe(data: RDD[Row], schema: StructType): Create a Dataframe from an RDD of 'Row' objects and schema.
* CreateDataframe(data: java.util.List[_], beanClass: Class[_]): creates a Dataframe from a Java list of objects and corresponding Javabean class.
* read(): Returns a 'DataframeReader' to read data from various sources.
* Sql(sqlText: String): Executes a Sql query and returns the result as a Dataframe.
* table(tablename: String): Returns a Dataframe representing a table registered in the catalog.

Executing Actions:

* 'Dataframe' and 'Dataset' objects provide methods like show(), Count(), Collect() etc. to perform actions on data.

Managing Resources:

* stop(): stops the 'SparkSession' and releases the allocated resources.
* close(): Closes the SparkSession.