SPARK SQL TECHNICAL SCENARIOS

1.How can one Create a dataframe of employees with employee name, department and salary and remove the duplicate elements from the dataframe.

CREATING A DATAFRAME

df = spark.read.options(header = True, infer_schema = True).csv("/FileStore/tables/*")

REMOVING THE DUPLICATE ELEMENTS

df.dropDuplicates()

2.How can one Create a dataframe of employee with employee name, department and salary and display the sum of salaries of all the departments.

CREATING A DATAFRAME

df = spark.read.options(header = True, infer_schema = True).csv("/FileStore/tables/*")

DISPLAYING THE SUM OF SALARIES OF ALL THE DEPARTMENTS

df7 = df6.withColumn("total_salary" , col("SAL") + col("COMM"))

df7.groupBy("DEPTNO").agg(sum(df7.total_salary)).show()

3.How can one Create a dataframe of employee with employee name, department and salary and display the minimum, maximum and average salary of all the department.

MAX SALARY

df7.groupBy("DEPTNO").agg(max(df7.total_salary)).show()

MIN SALARY

df7.groupBy("DEPTNO").agg(min(df7.total_salary)).show()

AVERAGE SALARY

df7.groupBy("DEPTNO").agg(avg(df7.total_salary)).show()

4.How can one Create a dataframe of employee with employee name, department and salary and sort the data based on salary and department in ascending and descending order respectively.

df7.sort(df7.total_salary.asc(), df7.DEPTNO.desc()).show()

5.How can one Create a dataframe of employee with id, name and manager_id and dataframe of manager with manager_id and manager_name and perform left outer join operation.

df8.alias("e").join(df8.alias("m"), col("e.MGR") == col("m.EMPNO"),
"leftOuter").select(col("e.EMPNO").alias("Employee_id"),
col("e.ENAME").alias("Employee_name"), col("e.MGR").alias("Manager_id"),
col("m.ENAME").alias("Manager_Name")).show()

6.What Are The Various Data Sources Available In Spark SQL ?
JSON, .csv, text files, orc files, hive tables, parquet file

7.What Is The Advantage Of A Parquet File?
COLUMNAR DATA - Minimizes Latency and thus takes less time in reading
SECURE - As not Human-readable
Storage - Low storage consumption

8.How Spark SQL Is Different From HQL and SQL?
Spark SQL brings native support of SQL and HIVE QL to spark rdds and dataframes
Allows developers to intermix SQL commands with complex analytics like machine learning and thus blurring the lines between rdds and relational tables

9.How can one Do Real-time Processing Using Spark SQL?
Spark streaming api of spark core enables us to perform real time processing using Spark SQL

10.How can one List The Functions Of Spark SQL.
SHOW [USER | SYSTEM |ALL] FUNCTIONS ([LIKE] regex pattern)

11.How can one .select a name whose name starts with letter 'S'
df7.filter(upper(col("ENAME")).like('S%')).show()
or
df7.where(upper(col("ENAME")).like('S%')).show()

12.How can one select the employee who has max and min age
df.filter(age == max(age) and age == min(age))

13.How can one display location, highest salary in that location.
df.select(col(“LOCATION”), col(“SAL”)).groupBy(“LOCATION”).agg(max(“SAL”))

14.How can one display department name, grade, max salary offered to each grade at each department.
df.select(col(“DNAME”), col(“GRADE”), col(“SAL”)).groupBy(“GRADE”, “DNAME”).agg(“max(SAL)”)

15.How can one write a command to see the structure of the dataframe?
df7.summary()
df.dtypes
df.columns

16.How can one select the employees who are all in the same age group
df.groupBy(col(“AGE_GRP”))

17.How can one select the department name where employee earning more than 2000
df7.select("DNAME", "total_salary").where(col("total_salary") > 2000).show()

18.How can one List the department name and the employee name who are staying in Chicago with higher grade
df7.select("ENAME", "DNAME").where((col("LOCATION") == "CHICAGO”) & (col("GRADE") == “HIGHER”).show()

19.How can one list different locations from where employees are reporting to the boss
df.select(col("LOCATION")).distinct().show()

20.How can one Create a dataframe and change datatype using cast function
df_new = df.withColumn(“col_name”, col(“col_name”).cast(DataType))

21.How can one Create a dataframe and remove the duplicate rows. Print the total number of non-duplicate records
df_new = df.dropDuplicates()
df_new.count()

22.How can one Create two dataframe and perform union method to combine them regardless of duplicate data.
df_union_All = df.unionAll(df1)

23.How can one Create two dataframe and retrieve all the records using inner join
df_inner_join = df1.alias(“a”).join(df2.alias(“b”), a.key = b.key, “inner”).select(“a.*”, “b.*”)

24.How can one Create two dataframe and retrieve all the records from the right dataset and matching records from the left dataset?
df_inner_join = df1.alias(“a”).join(df2.alias(“b”), a.key = b.key, “inner”).select(“a.*”, “b.*”)

25.How can one Create two dataframe and retrieve all the records from the left dataset and matching records from the right dataset
df_inner_join = df1.alias(“a”).join(df2.alias(“b”), a.key = b.key, “leftOuter”).select(“a.*”, “b.*”)

26.How can one Create a dataframe and find the total number of records, maximum value, minimum value, average value for each group.
df7.select(col("DEPTNO"),col("total_salary")).groupBy(col("DEPTNO")).agg(max(col("total_salary")).alias("Maximum sal"), min(col("total_salary")).alias("Minimum sal"), avg(col("total_salary")).alias("Average sal"), count(col("total_salary")).alias("No of employees")).show()

27.How can one Create a dataframe and add new column using lit function
df_new = df.withColumn(“col_name”, value = lit(“value”))

28.How can one Create the schema represented by a Struct Type matching the structure of Rows in the RDD

```
schema = StructType([
                StructField(col1, Datatype, nullable),
    .
            .])
df = spark.read.options(schema = schema).csv()
```

29.How can we get the aggregated values based on specific column values, which will be turned to multiple columns used in SELECT clause?
df.groupBy(col1).pivot(col2).agg(agg_fn(col_3))

30.List Few Commonly Used Spark Ecosystems.
SPARK CORE, SPARK SQL, SPARK STREAMING, MLIB, SPARK GRAPHX, SPARK R

31.List the Various Levels Of Persistence In Apache Spark?
Need of persistence?:As spark follows lazy evaluation, it will require to recompute all rdds from the beginning and this may require lots of time for computation. Hence we cache some rdds which are to be used frequently.

Methods to persist rdd/df
cache() - Uses memory only as default storage level
persist() - We can specify the level of persistence

Different Level of Persistence(works with persist only)

1. **MEMORY_ONLY: rdd/Dataframe is stored as deserialized JVM objects. If the size of RDD/DF is greater than memory it will not cache some partition and recompute them next time whenever needed**
   SPACE REQD: HIGH
   COMPUTATION TIME: LOW
2. **MEMORY_AND_DISK: Stores as deserialized JVM objects. Diff from above case is it uses disk if size of rdd > memory**
   SPACE REQD: HIGH
   COMPUTATION TIME: MEDIUM
3. **MEMORY_ONLY_SER: Stored as serialized JVM object(1 byte per partition). It is more space efficient compared to deserialized objects but at the expense of computational time.**
   SPACE REQD: LOW
   COMPUTATION TIME: HIGH
4. **MEMORY_AND_DISK_SER: Makes use of disk if needed and stores as serialized objects.**
   SPACE REQD: LOW
   COMPUTATION TIME: HIGH
5. **DISK_ONLY: Makes use of disk only**
   SPACE REQD: LOW
   COMPUTATION TIME: HIGH

35.What Do You Understand By Lazy Evaluation in Spark?
Lazy evaluation means only action commands lead to execution. Until then nothing is executed and rdds are just pointed to the previous rdds

36.What Are the Benefits Of Spark Over Map reduce?

Spark is 100X faster than MapReduce in in-memory computation and 10X faster than MapReduce in disks

37.Is It Necessary To Start Hadoop To Run Any Apache Spark Application?
Apache spark doesn't need a Hadoop cluster to work

38.How Hadoop Uses Replication To Achieve Fault Tolerance. How Is This Achieved In Apache Spark?
Hadoop uses replication to achieve fault tolerance. Hadoop strips the file into smaller units(blocks) and creates replicas by storing them into different worker nodes which are continuously sending heartbeats to name node. In case of a failure of any worker node, the replicas can be used to restore the file. Hadoop makes sure it maintains a certain replica factor (default is 3)

In case of Apache Spark the basis of fault tolerance is the immutable property of rdd. Since rdd's are immutable, the lineage graph in DAG saves the dependencies of every rdd involved in the operation. Thus in case of a failure, spark refers to these lineage graph to achieve fault tolerance and recover the rdds

39.How can one , With the help of Spark SQL command calculate a return value for each row based on the group of rows.
using the window function
from pyspark.sql.window import Window
window_spec = Window.partionBy(partn_col_name).orderBy(order_col_name)

df.withColumn(col(), dense_rank().over(window_spec)).show()

WINDOW FUNCTION USAGE & SYNTAX	WINDOW FUNCTION DESCRIPTION
row_number(): Column	Returns a sequential number starting from 1 within a window partition
rank(): Column	Returns the rank of rows within a window partition, with gaps.
percent_rank(): Column	Returns the percentile rank of rows within a window partition.
dense_rank(): Column	Returns the rank of rows within a window partition without any gaps. Where as Rank() returns rank with gaps.
ntile(n: Int): Column	Returns the ntile id in a window partition
cume_dist(): Column	Returns the cumulative distribution of values within a window partition

40.How can we generate a virtual table containing one or more rows in Spark SQL.
df.createOrReplaceTempView("table_name")

41.How to Create temporary result set that a user can reference possibly multiple times within the scope of a SQL statement.

42.How can we repartition the data based on the input expressions and make sure the data is not sorted within each partition.
DISTRIBUTE BY <key>

43.How can we repartition the data based on the input expressions and then sort the data within each partition.
CLUSTER BY <key>