SQL

1. What are window functions in SQL?
2. What are the different ranking functions? Explain each one of them using examples.

SQOOP

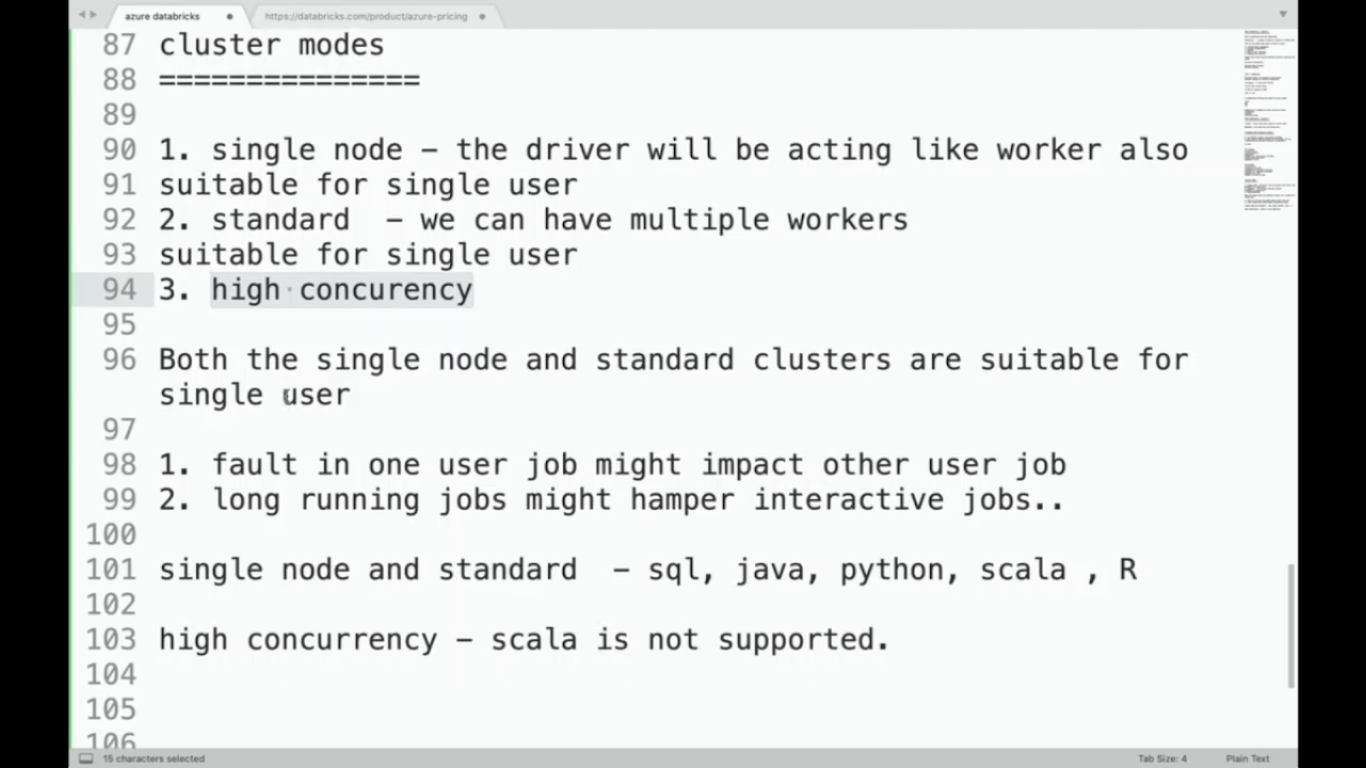
1. What is sqoop?
2. What are different sqoop commands?
3. What is the difference between target directory and warehouse directory arguments in sqoop import?
4. How to redirect output and log messages from console to files?
5. What are the different file formats and compression techniques in sqoop import?
6. Explain the arguments --columns, --where, --table in sqoop import?
7. What are free form query imports?
8. Is sqoop import a map-reduce job? Pls explain number of mappers argument in sqoop import.
   1. number of mappers
   2. split by
   3. controlling parallelism
   4. bounding vals query
9. How to format output in sqoop import?
10. How to append to existing data in sqoop import?
11. How to delete existing directory during sqoop import?
12. How to deal with nulls during sqoop import?
13. Explain incremental import in sqoop import.
14. What are the different ways of handling password in sqoop.
15. Explain sqoop import execution flow.
16. What is the concept of staging table in sqoop export and why it is needed?
17. Explain sqoop job.
18. How you schedule sqoop job in your project?

HIVE

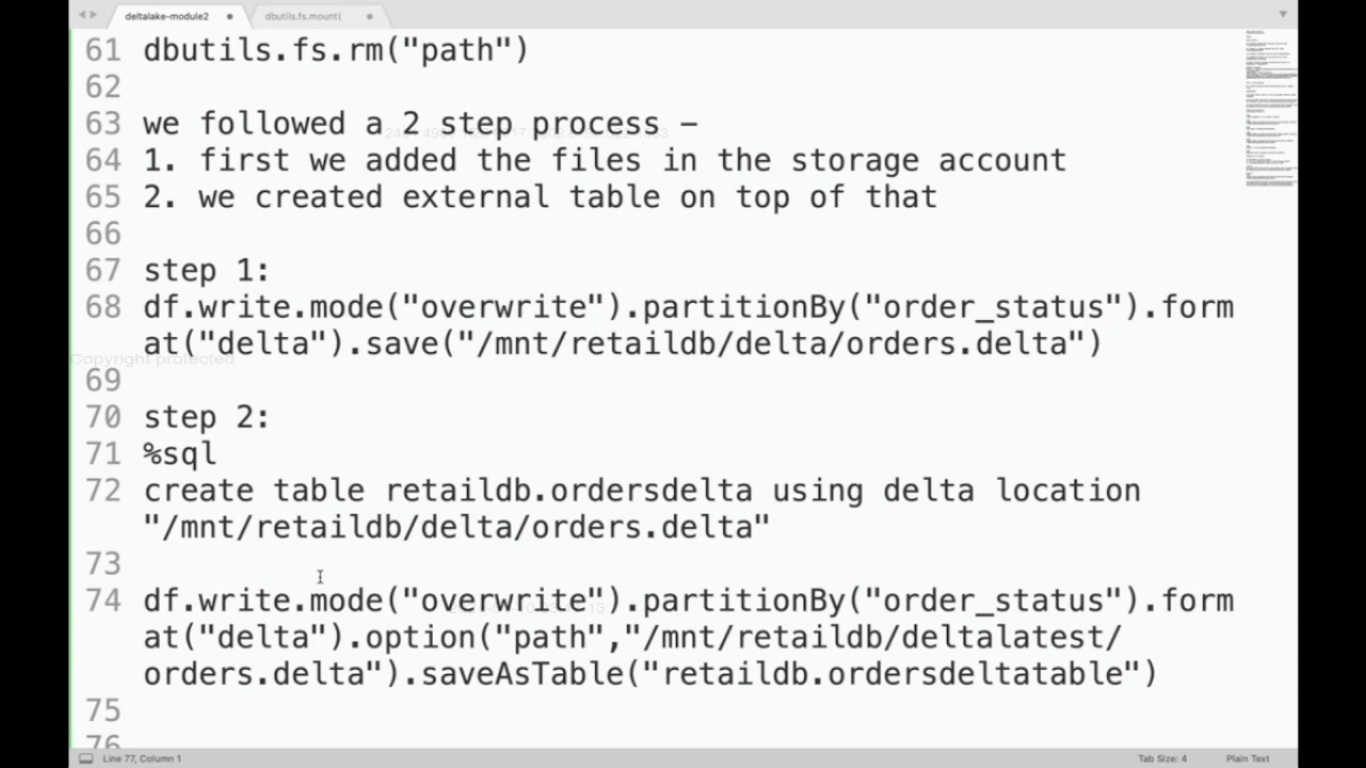
1. What is hive? How it places itself in Hadoop ecosystem?
2. Why we use hive in place of MapReduce?
3. Can you explain hive internal architecture in detail?
   1. https://cwiki.apache.org/confluence/display/Hive/Design
4. What is thrift server / hive server?
5. Difference between hive and rdbms.
6. Difference between hql and sql.
7. What is the concept of a table in hive?
8. Why metadata is stored in a rdbms?
9. What are the different types of tables in hive? And what is the difference between them?
   1. managed
   2. external
   3. temporary
10. What are the 3 different ways to insert data into a hive table?
    1. insert command using values
    2. load command (from local (copy paste), from hdfs (cut paste))
    3. insert command using queries (one table to another table)
11. What are the different ways to connect to hive?
    1. hive shell
    2. hue ui
    3. beeline
12. How you run hive queries in production?
    1. beeline with hive.hql file containing hive queries
13. What are different data types in hive?
    1. https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Types
14. What are hive subqueries?
    1. https://cwiki.apache.org/confluence/display/Hive/LanguageManual+SubQueries
15. What are hive views? Why we use views?
16. What are different set operations in hive?
    1. https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Union
17. What are the different types of functions in hive?
    1. https://cwiki.apache.org/confluence/display/Hive/LanguageManual%20UDF
18. What is lateral view and how we use it?
    1. <https://spark.apache.org/docs/latest/sql-ref-syntax-qry-select-lateral-view.html>
    2. <https://cwiki.apache.org/confluence/display/Hive/LanguageManual+LateralView>
19. How to create temporary as well as permanent custom UDFs in hive?
    1. <https://cwiki.apache.org/confluence/display/Hive/HivePlugins>
20. What is the difference between order by, sort by, distribute by, cluster by?
21. What is the difference between normalization and de-normalization and what is preferred in hive?
22. What is msck repair command and why do we use it?
23. How will you skip header rows in hive?
24. How and why we make a table immutable?
25. Difference between drop, truncate, purge, delete.
26. What is the use of the configuration serilization.null.format?
    1. <https://stackoverflow.com/questions/43263038/empty-string-is-not-treated-as-null-in-hive>
27. What are hive variables? Why do we use it?
28. How can we print headers along with data in hive?
29. What if while inserting data, data types in table definition and of actual data didn’t match?
30. --------------------------------------------------------------------------------------------------------
31. How and why do you integrate hive with Spark?
32. Why we need ACID properties in hive if hive is used for analytical processing?
33. What are some of the important points that you need to remember while dealing with ACID properties in hive? What properties need to be set to enable ACID in hive?
    1. <https://medium.com/@randy-huang/abc-b7cbaf37e4f7>
    2. <https://sparkbyexamples.com/apache-hive/hive-enable-and-use-acid-transactions/>
    3. <https://blog.clairvoyantsoft.com/hive-acid-transactions-part-i-f08da70b591b>
    4. https://blog.clairvoyantsoft.com/hive-acid-transactions-part-ii-6f5e478bbff
34. How ACID property is implemented in hive?
35. What is compaction? Why it is needed?
36. Describe some of the properties of insert-only transactional tables.
37. Is it possible to convert a non-acid table to acid table & vice-versa? How will you do that?

PYSPARK, DATABRICKS

1. What is databricks?
2. What are the challenges with open-source version of spark? Why we use databricks?
3. What is the pricing structure in databricks?
4. Is databricks available only on azure? Why databricks with azure?
5. What is a cluster?
6. What is a notebook?
7. What are the different types of cluster options?
8. What are the different types of cluster modes?



1. What are the different types of nodes in databricks?
2. What is the difference between azure databricks and azure synapse?
3. What is the use of magic command?
   1. To write various types of codes (sql, python, fs etc.) in same notebook.
4. What is DBFS?
5. Explain the architecture of databricks.
   1. <https://medium.com/@melissa.milligan/getting-started-with-data-engineering-on-databricks-88b4ebaf7a94>
6. What is dbutils?
7. Explain the functionality of fs utility in dbutils.
8. Explain the functionality of mount in fs utility in dbutils.
   1. mount(), unmount(), mounts(), updateMount()
9. Explain the functionality of data utility in dbutils.
   1. summarize
10. Explain the functionality of notebook utility in dbutils.
    1. exit
    2. run
11. Explain the functionality of widgets utility in dbutils.
    1. combobox, dropdown, multiselect, text
    2. get, remove, removeAll
12. How to pass parameters from one notebook to another?
    1. Using an argument in run command and widgets
13. What is databricks cli? How to install it?
14. How to configure databricks cli to connect to workspace?
    1. databricks configure --token
15. What is a datalake?
16. What are some of the advantages of datalake?
17. What are the challenges of using a datalake?
18. What are some of the scenarios which shows ACID properties violation and other problems with datalake?
19. What is deltalake?
20. How deltalake solves the problems which are there in datalake?
    1. crud with ACID compliance
    2. constraints check
    3. schema management
    4. time travel
21. How crud happens in deltalake?
    1. create, insert (append, overwrite)
    2. read
    3. update
    4. delete
22. How to create tables on top of delta data?



1. How to insert new data into a delta table / delta location?
   1. insert command
   2. copy command
   3. dataframe write with different modes



1. How to evolve schema in delta?
   1. .option(“mergeSchema”, “true”)
2. How to update some existing data in delta table?
   1. update command
3. How to delete some existing data from delta table?
   1. delete command
4. How to apply constraints on delta table?
   1. alter command with check
5. How will you time travel across different versions in delta table?
   1. describe history tablename
   2. version as of, timestamp as of
   3. restore table tablename to version as of x
6. What is change data feed feature in databricks? Why we need it?
7. What are the different ways to enable change data feed feature?
8. How the change data feed feature works in databricks?
   1. https://docs.databricks.com/en/delta/delta-change-data-feed.html
9. What is data governance? How unity catalog comes into play here?
10. What are some key features of unity catalog?
11. What are the limitations of implementing data governance without unity catalog?

OPTIMIZATIONS

1. What are the different optimization techniques you have used?
2. What is partitioning?
3. How do you select proper columns for partitioning?
4. What is the difference between static and dynamic partitioning?
5. What is bucketing?
6. How do you select proper columns for bucketing?
7. How to calculate the number of buckets needed?
8. Can we have partitions inside buckets? Can we have buckets inside partitions?
9. Tell some differences between partitioning and bucketing.
10. Given following operations - filter, joins, group by - which of them get optimized with the help of partitioning, which of them get optimized with the help of bucketing?
11. What are the configurations that need to be set for partitioning and bucketing?
    1. hive> SET hive.exec.dynamic.partition=true;
    2. hive> SET hive.exec.dynamic.partition.mode=non-strict;
    3. hive> SET hive.enforce.bucketing =true;​
    4. <https://medium.com/@sujathamudadla1213/how-can-you-configure-hive-for-dynamic-partitioning-and-specify-the-number-of-reducers-for-a-job-1a6464da4143>

I'd be glad to explain hive.exec.dynamic.partition.mode in Hive and provide an example:

**What is** hive.exec.dynamic.partition.mode**?**

In Apache Hive, hive.exec.dynamic.partition.mode is a configuration property that controls how Hive handles partitioning during data insertion into partitioned tables using dynamic partitioning. Dynamic partitioning allows you to insert data without explicitly specifying the partition values in your INSERT statement. Hive infers the partition values from the data itself based on the specified partition columns.

**Dynamic Partition Modes:**

This property has two main modes:

* **STRICT (default):** (Value: STRICT)
  + In this mode (the default), Hive strictly enforces that every row in the data being inserted must have values for all the partition columns. If any row is missing a partition value, the entire insertion operation fails. This mode is more secure as it prevents accidental creation of empty partitions.
* **NONSTRICT:** (Value: NONSTRICT)
  + With NONSTRICT mode, Hive inserts rows that have valid partition values into their corresponding partitions. However, rows missing partition values are simply skipped and not inserted into any partition. This mode offers more flexibility but requires caution to avoid unintended data loss.

**Choosing the Right Mode:**

* Use STRICT mode when data quality and completeness are crucial, and you want to ensure every row has all necessary partition values.
* Consider NONSTRICT mode if you anticipate some rows might lack partition values, but you still want to insert the rest of the valid data. However, be mindful of the potential for skipped rows and unintended consequences.

**Example:**

Suppose you have a partitioned table sales\_data with partition columns year and month. You're inserting data from a non-partitioned table temp\_sales.

**Scenario 1: STRICT Mode (Default)**

SQL

-- Assuming temp\_sales has all columns, including year and month

SET hive.exec.dynamic.partition=true;

INSERT OVERWRITE TABLE sales\_data PARTITION (year, month)

SELECT \* FROM temp\_sales;

Use code [with caution.](https://gemini.google.com/faq#coding)

content\_copy

In this case, with STRICT mode (the default), Hive will:

* Read each row from temp\_sales.
* Attempt to infer year and month values from the corresponding columns in the row.
* If all rows have valid year and month values, they'll be inserted into their respective partitions in sales\_data.
* If even one row is missing a year or month value, the entire insertion will fail.

**Scenario 2: NONSTRICT Mode**

SQL

SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=NONSTRICT;

INSERT OVERWRITE TABLE sales\_data PARTITION (year, month)

SELECT \* FROM temp\_sales;

Use code [with caution.](https://gemini.google.com/faq#coding)

content\_copy

Here, with NONSTRICT mode:

* Hive will process each row from temp\_sales.
* Rows with valid year and month values will be inserted into their corresponding partitions.
* Rows missing year or month values will be **skipped** and not inserted into any partition.

**Key Points:**

* Dynamic partitioning can improve performance and data organization, but it's essential to choose the appropriate mode based on your data quality and desired behavior.
* Consider using STRICT mode for stricter data validation.
* Use NONSTRICT mode cautiously, understanding that rows might be skipped.
* Always test your queries thoroughly in a non-production environment before running them on critical data.

1. How do you usually perform partitioning?
   1. static - load command
   2. dynamic - transfer data from non-partitioned table to partitioned table
2. How do you usually perform bucketing?
   1. transfer data from non-bucketed table to bucketed table
3. Which among mappers and reducers are invoked during partitioning, bucketing?
   1. partitioning - only mappers
   2. bucketing - both mappers and reducers (number of buckets ~ number of reducers)
4. What are the different types of join optimizations that you have used?
5. How a normal join works?
6. When can we perform a map side join?
7. Explain working of map side join in detail.
8. Among inner, left, right, full outer joins, which one of them can be performed using a map side join if the left table is small and right table is big?
9. When can we perform a bucket map join?
10. Explain working of bucket map join in detail.
11. When can we perform a sort merge bucket join?
12. Explain working of sort merge bucket join in detail.
13. What are some of the configurations needed to be set for performing different joins?
14. What is broadcast join and when to use it?
15. What are window functions and how they help in optimizing our queries?
16. What is the difference between row and columnar file formats? Why columnar file formats are preferred?
17. Why we need to think about file formats and what are the different factors based on which we decide a suitable file format?
18. Explain text file format in detail.
19. Explain avro file format in detail.
20. Explain orc file format in detail.
21. Explain parquet file format in detail.
22. What are the different tradeoffs when it comes to compression? What are the different factors based on which we decide a suitable compression technique?
23. Explain different compression techniques in detail.
    1. https://stackoverflow.com/questions/32382352/is-snappy-splittable-or-not-splittable
24. What is vectorization? What is the configuration to enable it?
25. What are the different supported engines for hive? Which configuration is used to change the hive engine?
26. How to deal with UDFs in a proper way?
27. What is CBO in hive?
    1. <https://teepika-r-m.medium.com/cost-based-optimization-in-hive-ea0c6296894c#:~:text=CBO%20is%20one%20of%20the%20optimization%20techniques%20used%20to%20boost,they%20are%20expensive%20to%20compute>.
28. -----------------------------------------------------------------------------------------------------------
29. What is performance tuning?
30. What are some of the performance tuning challenges?
31. What is the end goal of performance tuning?
32. Explain benchmarking in detail.
33. How do you perform instrumentation and pinpointing?
34. How do you optimize reading schemas in spark?
35. What is disk caching and how it helps optimize queries?
36. How can you optimize read using cache()?
37. How can you optimize the response time of the query that performs caching? Will column elimination help?
38. How can we push filter condition down to storage layer?
39. What are crippled predicates and how can we avoid them?
40. What is directory scan overhead and how can we resolve it?
41. What is small file problem and how can we resolve it?
42. What is haystack query? How z-ordering helps us with haystack queries?
43. How can we estimate the initial number of partitions in a dataframe?
44. Can we tune spark.sql.files.maxPartitionBytes value? What is the best value for it?
45. Can we tune spark.sql.files.openCostInBytes value? What is the best value for it?
46. Explain Spark memory management in Databricks.
47. How to check the amount of memory mapped per core?
48. What is data spill? Why does it happen?
49. What is data explosion? What are some scenarios that may cause data explosion?
50. Why data spill is bad?
51. How will you detect data spill?
52. What options may not solve data spill?
53. What are different ways you can resolve data spill issues (before and after shuffle spills)?
54. What are the implications of tuning spark.memory.fraction value?
55. Can you resolve data spill by using larger VMs?
56. How will you tune shuffle partitions to resolve data spill?
57. You are doing an explode() and its causing data explosion before shuffle. How will you resolve it?
58. You are doing an union() with orderBy and its causing data explosion after shuffle. How will you resolve it?
59. What is skew and how it is created?
60. Why skew is bad?
61. How to detect skew?
62. What options may not work to resolve skew?
63. What are some of the different skew tuning approaches?
64. How salting works?
65. Can you write code to implement salting in a join?
    1. <https://medium.com/curious-data-catalog/sparks-salting-a-step-towards-mitigating-skew-problem-5b2e66791620>
66. Can you write code to implement salting in groupBy?
67. What are some disadvantages of salting?
68. What are skew hints in databricks? Do they resolve data skew?
69. How AQE resolves data skew?
70. How shuffle works?
71. How shuffle works in different wide transformations like repartition, orderBy, groupBy etc?
72. How will you tune shuffling?
    1. Avoid shuffle.
    2. If cant avoid shuffle, tune shuffle bottlenecks.
73. Explain working of shuffle sort merge join.
74. Explain working of broadcast hash join.
75. Explain working of shuffle hash join.
76. How broadcast performs with small datasets?
77. How broadcast performs with large datasets?
78. How will you tune a large-to-large table join?
79. How bucketing works?
80. Explain different bucket joins.
    1. bucket map join
    2. sort merge bucket join
81. What are some of the limitations of using bucketing in joins?
82. How will you perform a large-to-large table join with the help of bucketing?
83. What are the different techniques of saving the results of intermediate joins to be used later?
84. ---------------------------------------------------------------------------------------------------------
85. What is data skipping and how delta helps us to skip data?
    1. https://towardsdatascience.com/delta-lake-enables-effective-caching-mechanism-and-query-optimization-in-addition-to-acid-96c216b95134
86. What is disk caching and how it helps optimize queries?
87. What is the problem with small files in delta? How do they get created?
88. How can we resolve small files problem?
89. What is z-ordering and how does it help in data skipping?
90. Why we use vacuum command?
    1. vacuum tablename retain x hours
91. What is auto optimize in databricks?
    1. The term auto optimize is sometimes used to describe functionality controlled by the settings delta.autoOptimize.autoCompact and delta.autoOptimize.optimizeWrite.
    2. https://docs.databricks.com/en/delta/tune-file-size.html#auto-compact
92. What is photon engine and why we use it?
93. --------------------------------------------------------------------------------------------------------

ADF

1. -------------------------------------------------------------------------------------------------------
2. What are the different kinds of triggers in ADF? Explain the working of each one of them.

SYSTEM DESIGN

1. Difference between transactional vs analytical processing.
2. Difference between database, data warehouse, datalake.
3. Difference between normalization and de-normalization.
4. What are the different normal forms?
5. What is SCD? What are the different types of SCD?
6. Can you write code for SCD1 in PySpark, SQL?
7. Can you write code for SCD2 in PySpark, SQL?
8. What do you mean by ACID properties?
9. What is modern data architecture / 2-tier architecture? What are the challenges with 2-tier architecture?
10. What is lakehouse architecture? How does it resolve the challenges of 2-tier architecture?
11. How databricks has implemented its lakehouse architecture?
    1. storage layer (data lake) -> transactional layer (delta lake) -> delta engine
12. What is medallion architecture?
13. Can u implement a sample pipeline having medallion architecture using change data feed feature?

PROJECT

1. Explain your project in detail.
2. How much amount of data that you are processing per day?
3. What you mean by incremental data? And how you process incremental data in your project?
4. What are the challenges/bottlenecks you faced while developing your pipeline and how did you resolve them?