Certification Questions Practice Set (Performance Optimization)

1. Select the incorrect statement while working with a virtual warehouse ?

   A) Compute resources waiting to shut down are considered to be in “quiesce” mode.

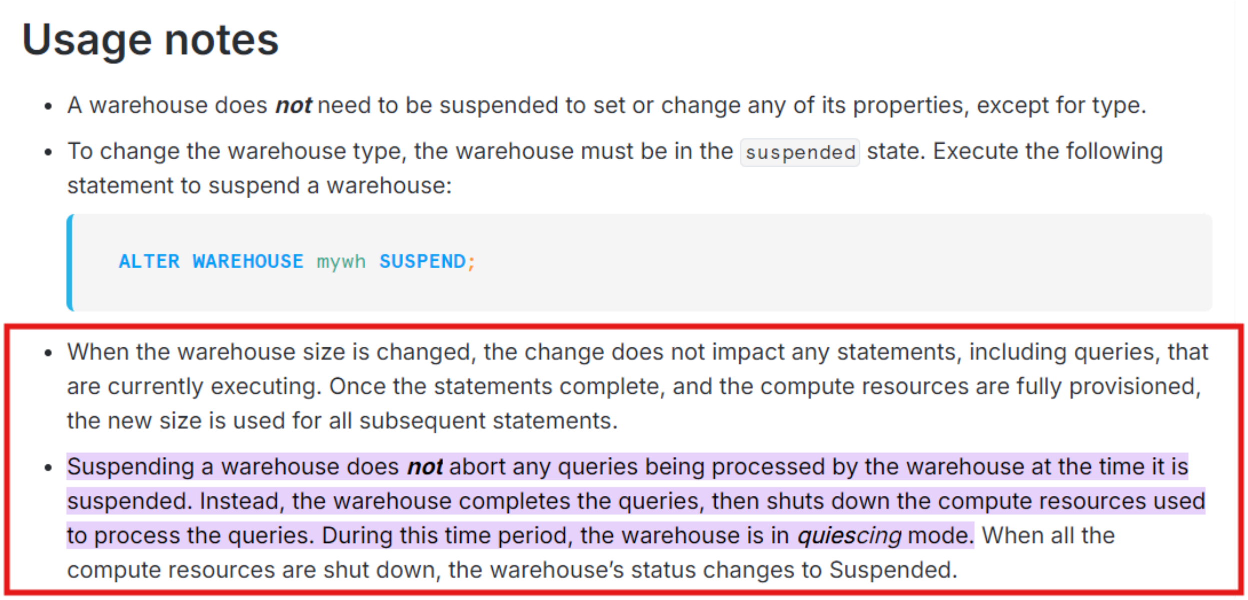
   B) Resizing a warehouse to a larger size is useful while loading & unloading significant amounts of data.

   C) Resizing a warehouse would have an immediate impact on statements that are currently running or being executed by the warehouse.

   D) Resizing a suspended warehouse does not provision new compute resources for the warehouses.

Answer : C

Explanation : https://docs.snowflake.com/en/sql-reference/sql/alter-warehouse



Usage notes

- A warehouse does ***not*** need to be suspended to set or change any of its properties, except for type.
- To change the warehouse type, the warehouse must be in the `suspended` state. Execute the following statement to suspend a warehouse:

```
ALTER WAREHOUSE mywh SUSPEND;
```

- When the warehouse size is changed, the change does not impact any statements, including queries, that are currently executing. Once the statements complete, and the compute resources are fully provisioned, the new size is used for all subsequent statements.
- Suspending a warehouse does ***not*** abort any queries being processed by the warehouse at the time it is suspended. Instead, the warehouse completes the queries, then shuts down the compute resources used to process the queries. During this time period, the warehouse is in *quiescing* mode. When all the compute resources are shut down, the warehouse’s status changes to Suspended.

2. A company is reporting performance issues when querying an `ORDER` fact table in Snowflake that gets updated every hour. The query is complex with multiple `JOIN` and `GROUP BY` statements. The fact table is 1.5 TB, and the company is using a single cluster with a size Small virtual warehouse.

What is the FIRST action to take to improve the performance of the query?

1. Manually re-cluster the `ORDER` fact table using appropriate filters.
2. Create a materialized view on top of the `ORDER` fact table.
3. Change the virtual warehouse configuration to multi-cluster.
4. Increase the size of the virtual warehouse.

Answer : 4

Explanation : Look at the hints provided

The query is complex. So, it's a complexity issue and not scale issue

1. Re-clustering or creating materialized view on a 1.5 TB sized table will take a lot of time, So may not qualify for the 1st action taken
2. Increasing size of warehouse brings immediate relief by offering a higher horse power to execute the complex query faster

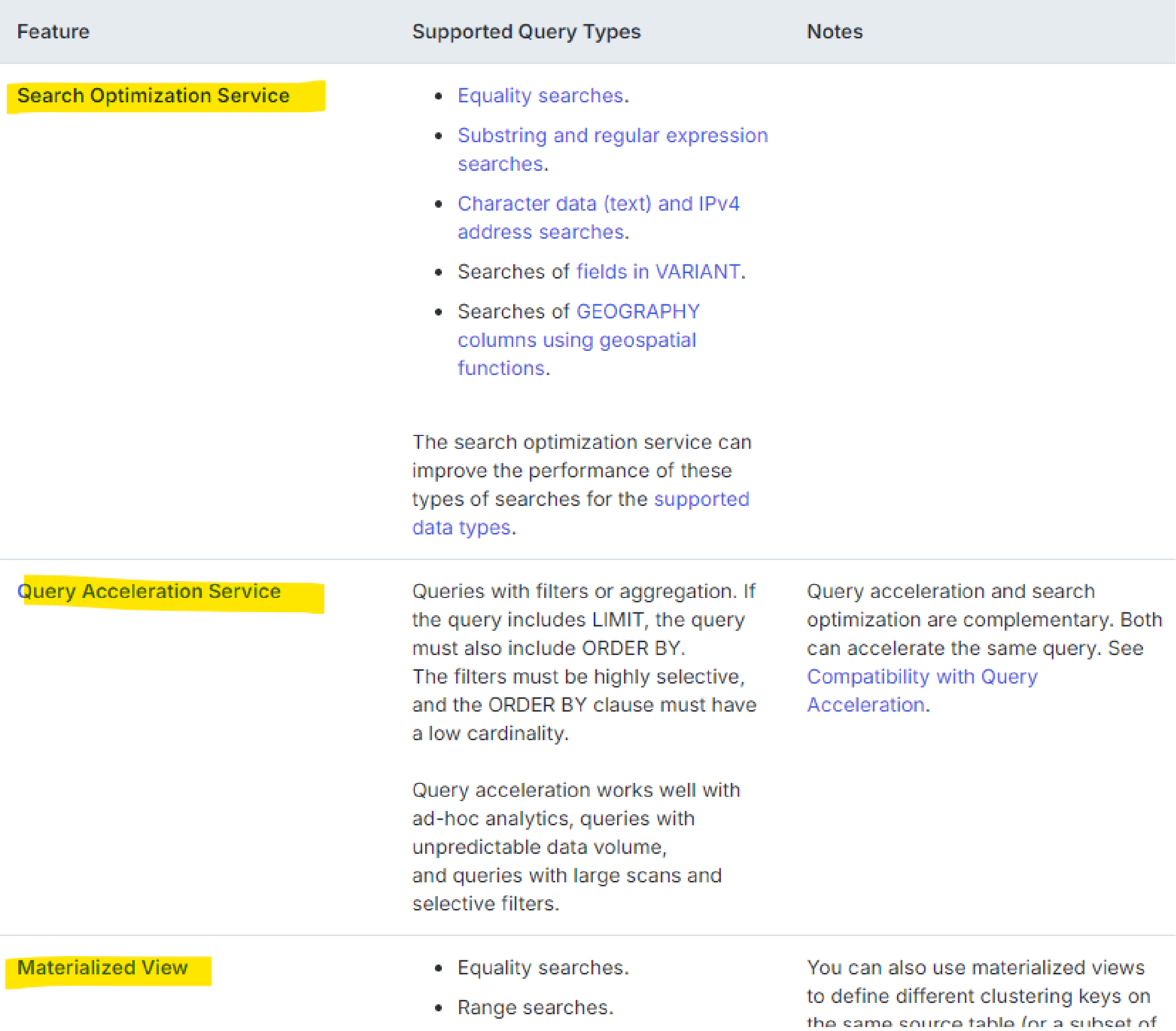
3. Melissa, Senior Data Engineer, is looking to optimize query performance for one of the Critical Control Dashboards. She found that most of the searches by the users on the control dashboards are based on equality search on all the underlying columns mostly.

Which best techniques should she consider here?

A. She can go for clustering on underlying tables which can speed up equality searches.
B. A materialized view speeds both equality searches and range searches.
C. The search optimization service would best fit here as it can be applied to all underlying columns and speeds up equality searches.
D. Melissa can create indexes and hints on the searchable columns to speed up equality search.

Answer : C

Explanation : https://docs.snowflake.com/en/user-guide/search-optimization-service



Feature	Supported Query Types	Notes
Search Optimization Service	- Equality searches. - Substring and regular expression searches. - Character data (text) and IPv4 address searches. - Searches of fields in VARIANT. - Searches of GEOGRAPHY columns using geospatial functions. The search optimization service can improve the performance of these types of searches for the supported data types.	
Query Acceleration Service	Queries with filters or aggregation. If the query includes LIMIT, the query must also include ORDER BY. The filters must be highly selective, and the ORDER BY clause must have a low cardinality. Query acceleration works well with ad-hoc analytics, queries with unpredictable data volume, and queries with large scans and selective filters.	Query acceleration and search optimization are complementary. Both can accelerate the same query. See Compatibility with Query Acceleration.
Materialized View	- Equality searches. - Range searches.	You can also use materialized views to define different clustering keys on the same source table (or a subset of

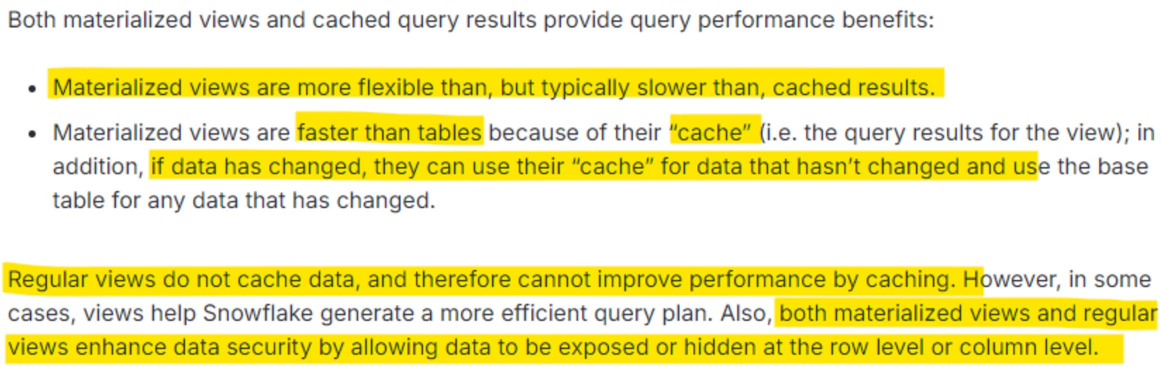
3. David, a Lead Data Engineer with XYZ company, is looking to improve query performance and other benefits while working with tables, regular views, materialized views (MVs), and cached results.

Which one of the following does not show key similarities and differences between tables, regular views, cached query results, and materialized views while choosing any of them by David?

A. Regular views do not cache data, and therefore cannot improve performance by caching.
B. As with non-materialized views, a materialized view automatically inherits the privileges of its base table.
C. Cached query results: Used only if data has not changed and if the query only uses deterministic functions (e.g., not CURRENT_DATE).
D. Materialized views are faster than tables because of their "cache" (i.e., the query results for the view); in addition, if data has changed, they can use their "cache" for data that hasn’t changed and use the base table for any data that has changed.
E. Both materialized views and regular views enhance data security by allowing data to be exposed or hidden at the row level or column level.

Answers : B

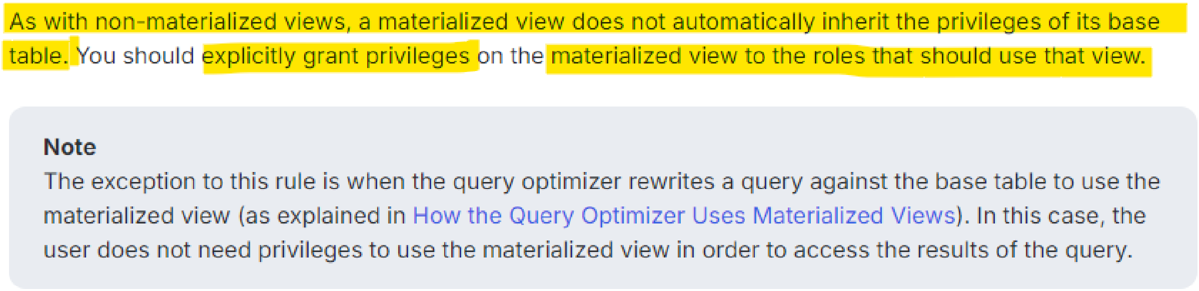
Explanation : https://docs.snowflake.com/en/user-guide/views-materialized



Both materialized views and cached query results provide query performance benefits:

- Materialized views are more flexible than, but typically slower than, cached results.
- Materialized views are faster than tables because of their “cache” (i.e. the query results for the view); in addition, if data has changed, they can use their “cache” for data that hasn’t changed and use the base table for any data that has changed.

Regular views do not cache data, and therefore cannot improve performance by caching. However, in some cases, views help Snowflake generate a more efficient query plan. Also, both materialized views and regular views enhance data security by allowing data to be exposed or hidden at the row level or column level.



As with non-materialized views, a materialized view does not automatically inherit the privileges of its base table. You should explicitly grant privileges on the materialized view to the roles that should use that view.

Note

The exception to this rule is when the query optimizer rewrites a query against the base table to use the materialized view (as explained in How the Query Optimizer Uses Materialized Views). In this case, the user does not need privileges to use the materialized view in order to access the results of the query.

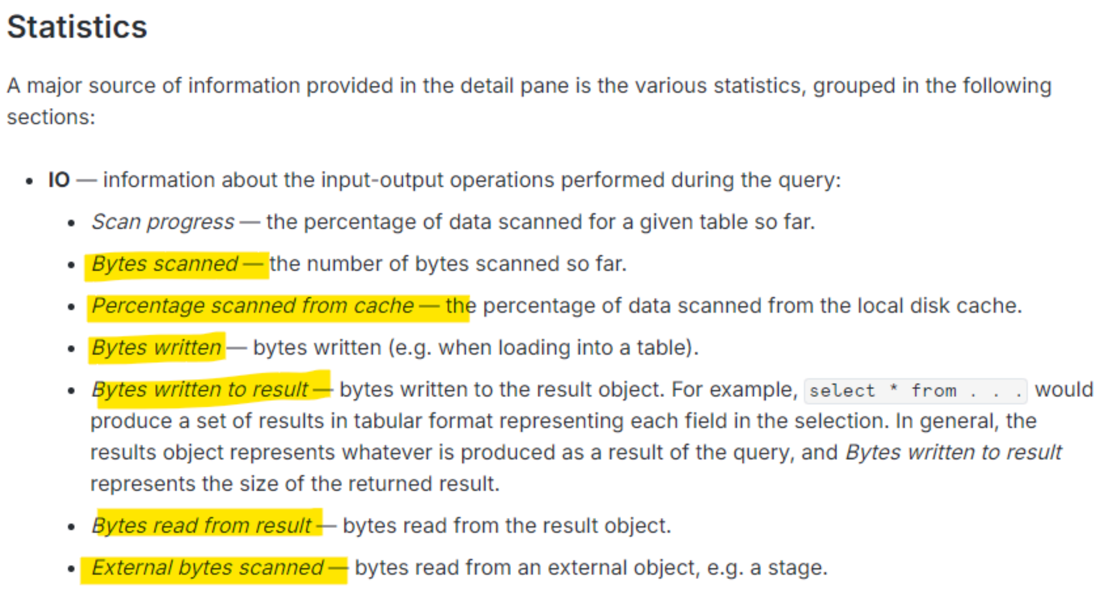
4. A Data Engineer wants to analyze query performance and is looking for profiling information. He went to Query/Operator Details, also called Profile Overview of Query Profile Interface, and is searching for statistics attributes around I/O.

Which of the following information can't he get from there?

A. Percentage scanned from cache — the percentage of data scanned from the local disk cache.
B. Bytes written — bytes written (e.g., when loading into a table).
C. External bytes scanned — bytes read from an external object, e.g., a stage.
D. Bytes sent over the wireframe — the amount of data sent over the wireframe.
E. Bytes read from result — bytes read from the result object.

Answer : D

Explanation - https://docs.snowflake.com/en/user-guide/ui-snowsight-activity



Statistics

A major source of information provided in the detail pane is the various statistics, grouped in the following sections:

- **IO** — information about the input-output operations performed during the query:
  - *Scan progress* — the percentage of data scanned for a given table so far.
  - *Bytes scanned* — the number of bytes scanned so far.
  - *Percentage scanned from cache* — the percentage of data scanned from the local disk cache.
  - *Bytes written* — bytes written (e.g. when loading into a table).
  - *Bytes written to result* — bytes written to the result object. For example, `select * from . . .` would produce a set of results in tabular format representing each field in the selection. In general, the results object represents whatever is produced as a result of the query, and *Bytes written to result* represents the size of the returned result.
  - *Bytes read from result* — bytes read from the result object.
  - *External bytes scanned* — bytes read from an external object, e.g. a stage.

5. Jonas, a Lead Performance Engineer, identified that some operations of his query, which functionally remove duplicates from a huge data set, are spilling the data to remote disk.

How can he alleviate spilling to a remote disk for better query performance?

A. Jonas can recommend using a large warehouse, which effectively increases the available memory/local disk space for the operations.

B. He can process data in smaller batches to manage workload.

C. Spilling does not have a profound effect on query performance (especially if a remote disk is used for spilling).

D. Data sharing can be helpful to improve query performance

Answers : A, B

---

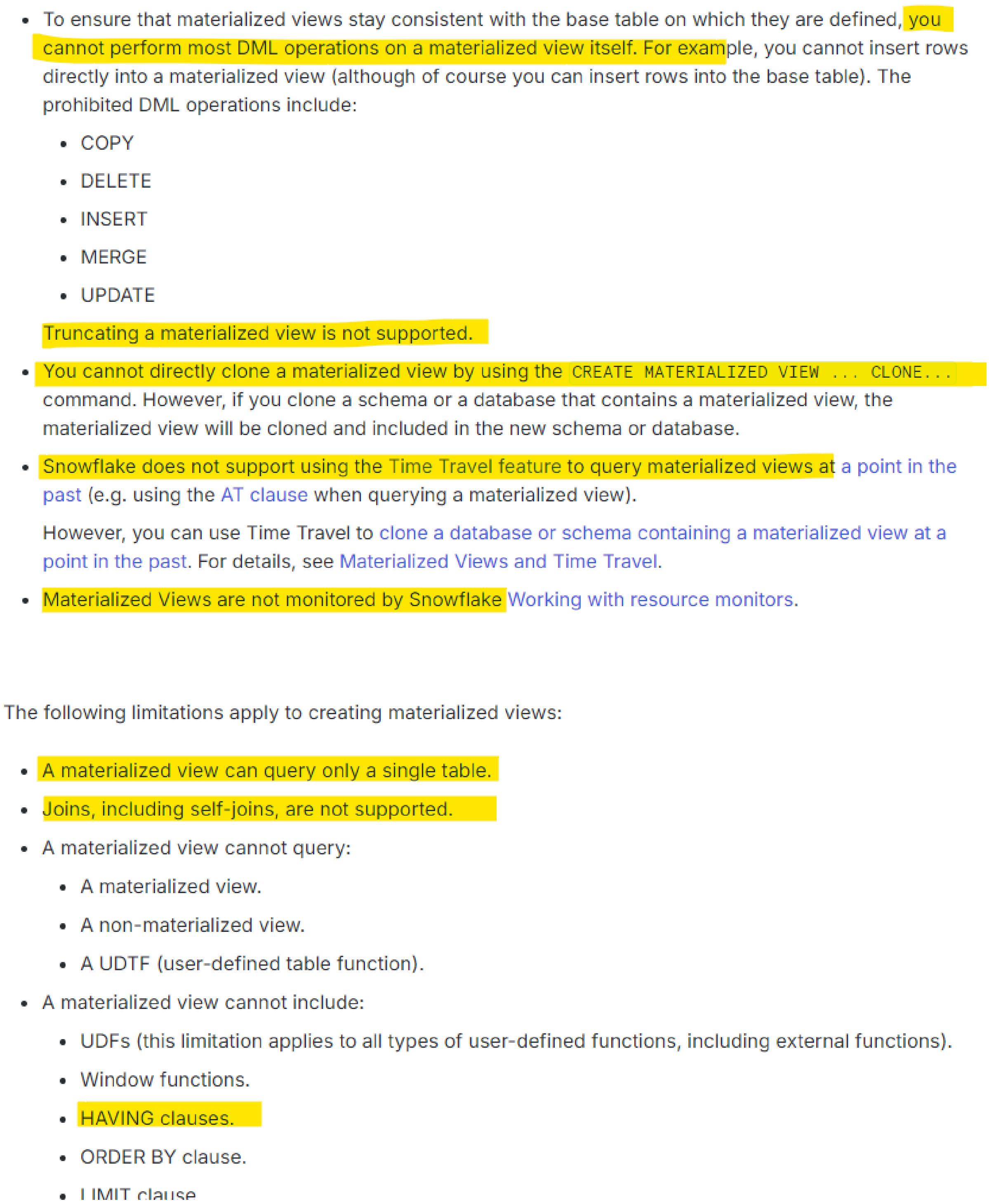
6. A Data Engineer identified a use case where he decided to use materialized views for query performance.

Which one is not a limitation he must be aware of before using materialized views in their use case?

A. Truncating a materialized view is not supported.
B. Time travel is not currently supported on materialized views.
C. You cannot directly clone a materialized view by using the CREATE MATERIALIZED VIEW .. CLONE .. command.
D. A materialized view can query only a single table and joins, including self-joins, are not supported.
E. A materialized view does not support clustering.
F. Materialized views cannot be created on shared data.
G. A materialized view cannot include HAVING clauses or ORDER BY clauses.
H. Context functions like CURRENT_TIME or CURRENT_TIMESTAMP are not permitted.

Answers : E

Explanation - https://docs.snowflake.com/en/user-guide/views-materialized



- To ensure that materialized views stay consistent with the base table on which they are defined, you cannot perform most DML operations on a materialized view itself. For example, you cannot insert rows directly into a materialized view (although of course you can insert rows into the base table). The prohibited DML operations include:
    - COPY
    - DELETE
    - INSERT
    - MERGE
    - UPDATE

  Truncating a materialized view is not supported.
- You cannot directly clone a materialized view by using the `CREATE MATERIALIZED VIEW ... CLONE...` command. However, if you clone a schema or a database that contains a materialized view, the materialized view will be cloned and included in the new schema or database.
- Snowflake does not support using the Time Travel feature to query materialized views at a point in the past (e.g. using the AT clause when querying a materialized view).

  However, you can use Time Travel to clone a database or schema containing a materialized view at a point in the past. For details, see Materialized Views and Time Travel.
- Materialized Views are not monitored by Snowflake Working with resource monitors.

The following limitations apply to creating materialized views:

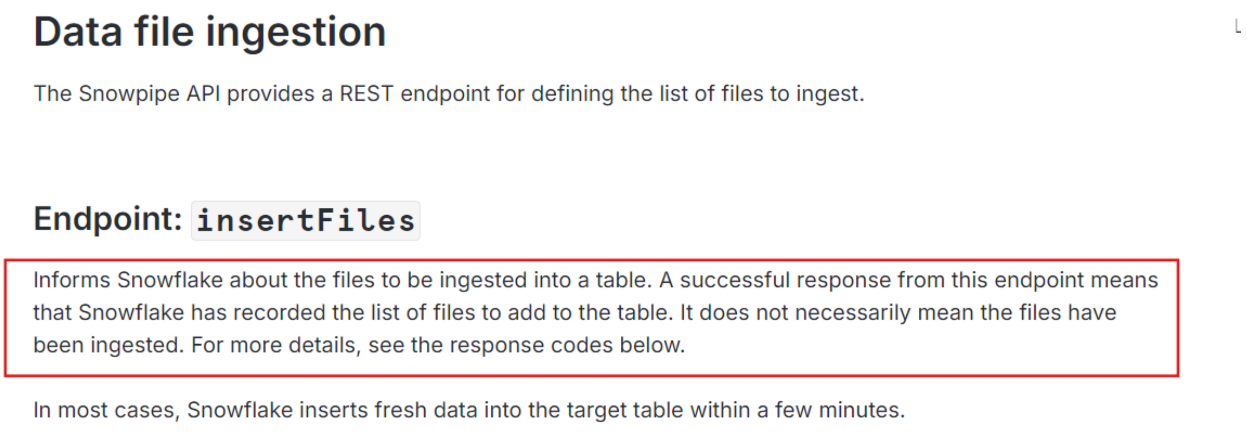
- A materialized view can query only a single table.
- Joins, including self-joins, are not supported.
- A materialized view cannot query:
    - A materialized view.
    - A non-materialized view.
    - A UDTF (user-defined table function).
- A materialized view cannot include:
    - UDFs (this limitation applies to all types of user-defined functions, including external functions).
    - Window functions.
    - HAVING clauses.
    - ORDER BY clause.
    - LIMIT clause

7. Mark the correct statements about cache.

A. Materialized views are more flexible than, but typically slower than, cached results.
B. Materialized views are faster than tables because of their "cache" (i.e., the query results for the view); in addition, if data has changed, they can use their "cache" for data that hasn't changed and use the base table for any data that has changed.
C. For persisted query results of all sizes, the cache expires after 24 hours.
D. The size of the warehouse cache is determined by the compute resources in the warehouse.
E. Warehouse cache is dropped when the warehouse is suspended, which may result in slower initial performance for some queries after the warehouse is resumed.

Answers : A, B, D

Explanation :



Data file ingestion

The Snowpipe API provides a REST endpoint for defining the list of files to ingest.

Endpoint: `insertFiles`

Informs Snowflake about the files to be ingested into a table. A successful response from this endpoint means that Snowflake has recorded the list of files to add to the table. It does not necessarily mean the files have been ingested. For more details, see the response codes below.

In most cases, Snowflake inserts fresh data into the target table within a few minutes.

8. Michael, a Data Engineer, is running a data query to achieve the union of datasets coming from multiple data sources. Later, he figured out that the data processing query is taking more time than expected. He started analyzing the query performance using the query

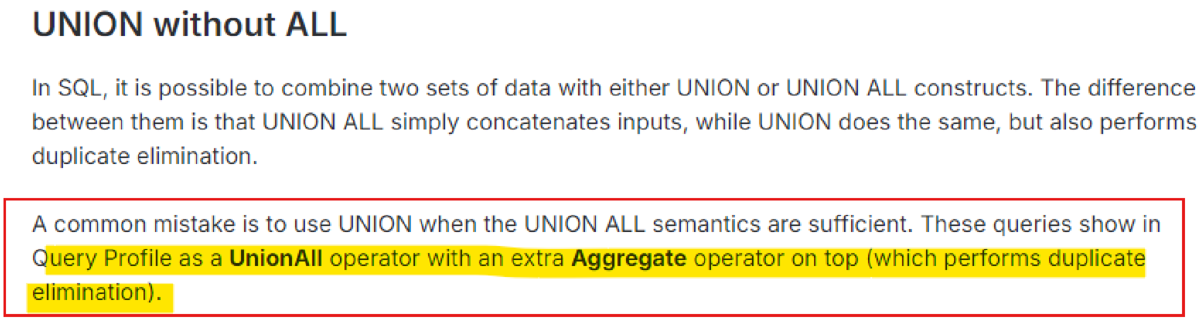
profile interface. He discovered and realized that he used UNION when the UNION ALL semantics were sufficient.

Which extra data processing operator did Michael figure out while doing query profile analysis in this case, which helps him to identify this performance bottleneck?

A. Aggregate
B. UNION ALL
C. Flatten
D. Join
E. Filter

Answer : A

Explanation : https://docs.snowflake.com/en/user-guide/ui-snowsight-activity



UNION without ALL

In SQL, it is possible to combine two sets of data with either UNION or UNION ALL constructs. The difference between them is that UNION ALL simply concatenates inputs, while UNION does the same, but also performs duplicate elimination.

A common mistake is to use UNION when the UNION ALL semantics are sufficient. These queries show in Query Profile as a **UnionAll** operator with an extra **Aggregate** operator on top (which performs duplicate elimination).

---

9. While working with multi-cluster warehouses, select the incorrect understanding of a Data Engineer about its usage.

A. Multi-cluster warehouses are designed specifically for handling queuing and performance issues related to large numbers of concurrent users and/or queries.
B. Unless you have a specific requirement for running in Maximized mode, multi-cluster warehouses should be configured to run in Auto-scale mode, which enables Snowflake to automatically start and stop clusters as needed.
C. When choosing the minimum number of clusters for a multi-cluster warehouse, keep the default value as 1.
D. Multi-cluster warehouses generally improve query performance, particularly for larger, more complex queries.

E. When choosing the maximum number of clusters for a multi-cluster warehouse, set its value as large as possible.

Answer : D

Explanation : Multi-cluster warehouses are used to clear the queue when volume of queries goes up. In case the complexity of the query goes up, it's better to go for a higher power virtual warehouse.

---

10. As a Data Engineer, you have a requirement to query the most recent data from a large dataset that resides in external cloud storage. How would you design your data pipelines keeping in mind the fastest time to delivery?

A. Data pipelines would be created to first load data into internal stages and then into a permanent table with SCD Type 2 transformation.

B. Direct querying external tables on top of existing data stored in external cloud storage for analysis without first loading it into Snowflake.

C. Unload data into Snowflake internal data storage using the PUT command.

D. Snowpipe can be leveraged with streams to load data in micro-batch fashion with CDC streams that capture the most recent data only.

E. External tables with materialized views can be created in Snowflake

Answer : E

Explanation : Since data is outside Snowflake, External tables are a good choice. Since fastest time to delivery is the consideration, creating a Materialized view on top of External tables will help solve the problem

---

11. SYSTEM$CLUSTERING_INFORMATION functions return clustering information, including average clustering depth, for a table based on one or more columns in the table. The function returns a JSON object containing average_overlaps name/value pairs.

Does high average_overlaps indicate well-organized clustering?

A. YES
B. NO

Answer : NO

---

12. Jane, a Data Engineer, is tasked with optimizing the performance of a data pipeline that loads semi-structured data (JSON) into Snowflake. She notices that the pipeline is taking longer than expected. Which of the following actions should she take to improve performance?

A. Increase the size of the virtual warehouse used for the data load.
B. Use the STRIP_OUTER_ARRAY option in the COPY command to flatten nested JSON structures.
C. Convert JSON files to CSV before loading into Snowflake.
D. Increase the data retention period for the staging area.

Answer : B

---

13. John is optimizing a query that joins large tables in Snowflake. He notices that the query performance is poor. Which approach should he take to improve the performance?

A. Increase the size of the virtual warehouse.
B. Create clustered tables based on the join keys.
C. Use the STREAMING join option in the SQL query.
D. Disable auto-scaling for the virtual warehouse.

Answer : B

---

14. Tom, a Data Engineer, wants to automate the process of refreshing a materialized view in Snowflake whenever new data is loaded. What should he do?

A. Schedule a task that refreshes the materialized view using the REFRESH MATERIALIZED VIEW command.

B. Enable auto-refresh on the materialized view during creation.

C. Manually refresh the materialized view after each data load.

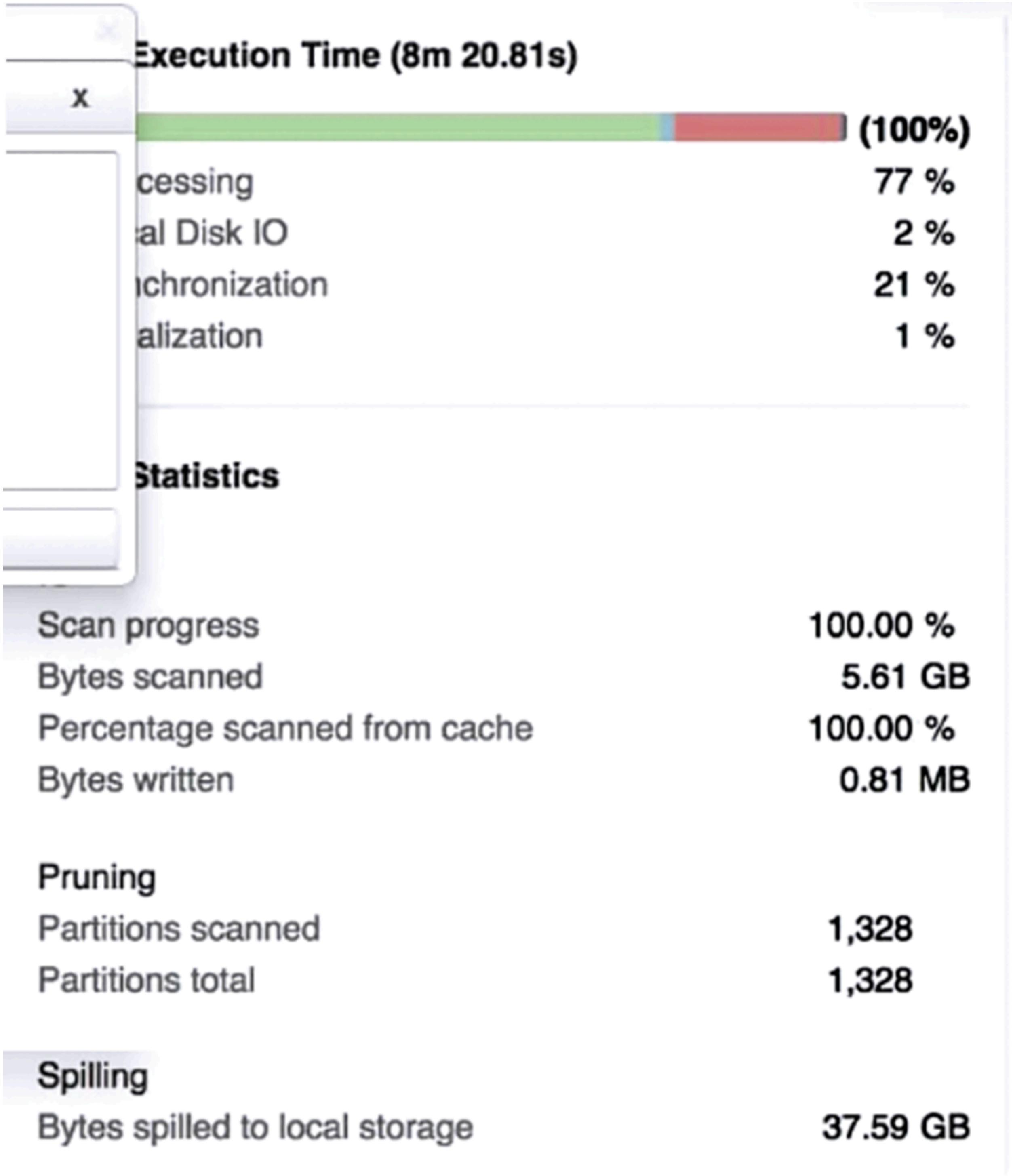
D. Use the CREATE OR REPLACE VIEW command to refresh the view.

Answer : B

Explanation : Materialized views get auto refreshed leveraging Snowflake Serverless compute

---

15. A Data Engineer is investigating a query that is taking a long time to return. The Query Profile shows the following :



Execution Time (8m 20.81s)

	(100%)
cessing	77 %
al Disk IO	2 %
chronization	21 %
alization	1 %

Statistics

Scan progress	100.00 %
Bytes scanned	5.61 GB
Percentage scanned from cache	100.00 %
Bytes written	0.81 MB

Pruning

Partitions scanned	1,328
Partitions total	1,328

Spilling

Bytes spilled to local storage	37.59 GB

What step should the Engineer take to increase the query performance?

A. Add additional virtual warehouses.

B. Increase the size of the virtual warehouse.

C. Rewrite the query using Common Table Expressions (CTEs).

D. Change the order of the joins and start with smaller tables first.

Answer : B

Explanation - Data Spillage happening to local storage which means the VW is not sufficient capacity

---

16. A Data Engineer executes a complex query and wants to make use of Snowflake’s query results caching capabilities to reuse the results.

Which conditions must be met? (Choose three.)

A. The results must be reused within 72 hours.

B. The query must be executed using the same virtual warehouse.

C. The USED_CACHED_RESULT parameter must be included in the query.

D. The table structure contributing to the query result cannot have changed.

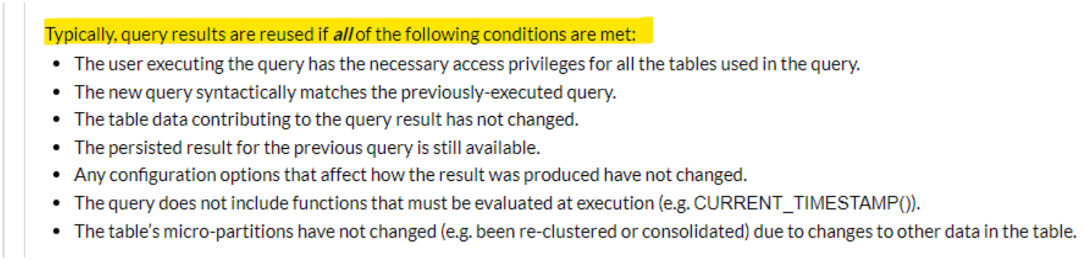
E. The new query must have the same syntax as the previously executed query.

F. The micro-partitions cannot have changed due to changes to other data in the table.

Answer : D, E, F

Explanation :

https://community.snowflake.com/s/article/Understanding-Result-Caching



Typically, query results are reused if *all* of the following conditions are met:

- The user executing the query has the necessary access privileges for all the tables used in the query.
- The new query syntactically matches the previously-executed query.
- The table data contributing to the query result has not changed.
- The persisted result for the previous query is still available.
- Any configuration options that affect how the result was produced have not changed.
- The query does not include functions that must be evaluated at execution (e.g. CURRENT_TIMESTAMP()).
- The table's micro-partitions have not changed (e.g. been re-clustered or consolidated) due to changes to other data in the table.

---

17. Given the table SALES which has a clustering key of column CLOSED_DATE, which table function will return the average clustering depth for the SALES_REPRESENTATIVE column for the North American region?

A. select system$clustering_information('Sales', 'sales_representative', 'region = ''North America''');

B. select system$clustering_depth('Sales', 'sales_representative', 'region = ''North America''');

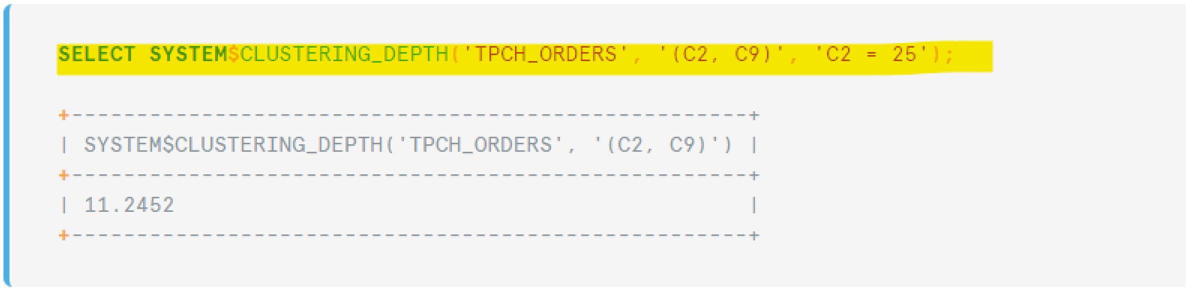
C. select system$clustering_depth('Sales', 'sales_representative') where region = 'North America';

D. select system$clustering_information('Sales', 'sales_representative') where region = 'North America’;

Answer : B

Explanation :

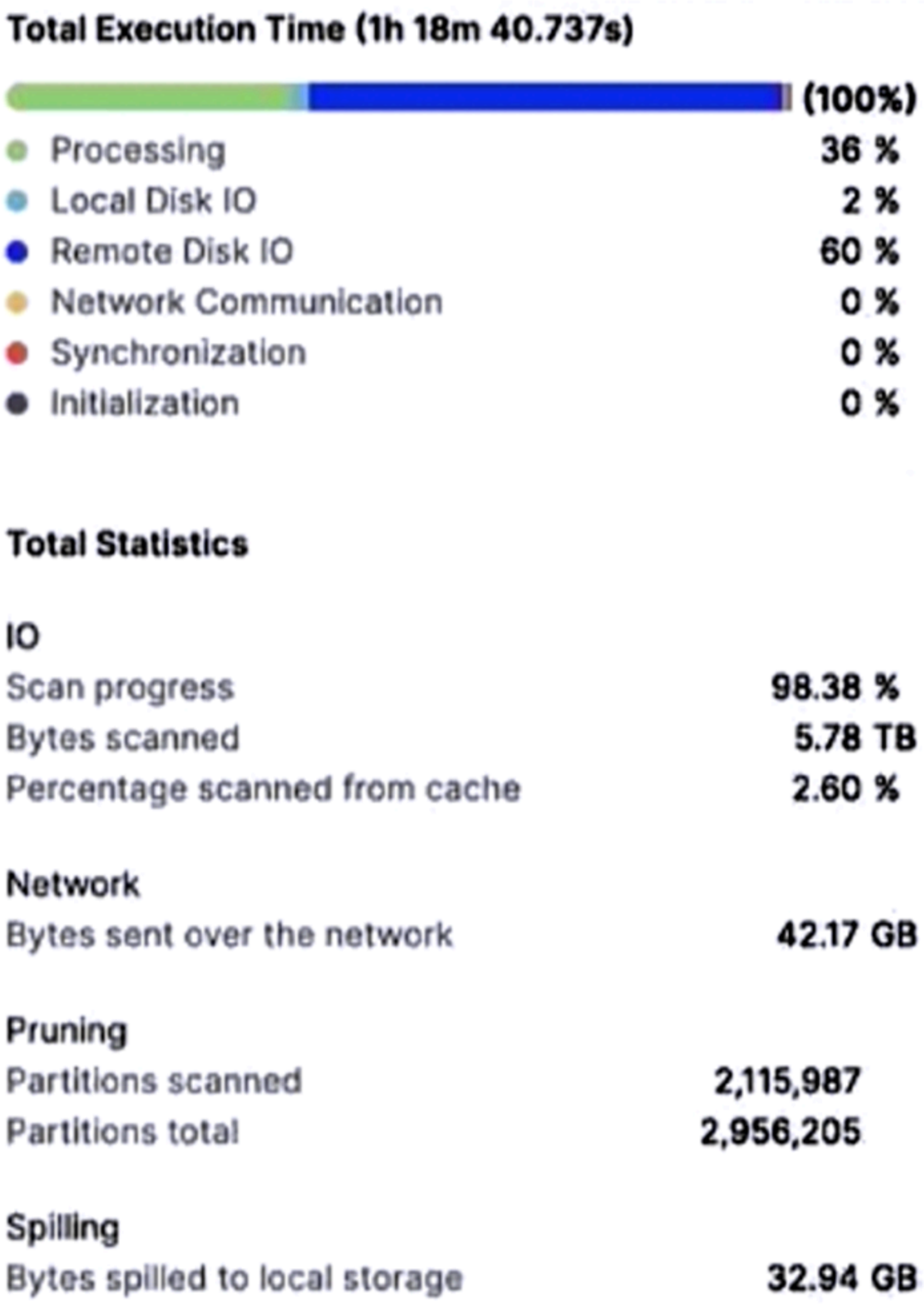
Same as the previous example, but with a predicate on one of the columns:



```
SELECT SYSTEM$CLUSTERING_DEPTH('TPCH_ORDERS', '(C2, C9)', 'C2 = 25');

+--------------------------------------------------+
| SYSTEM$CLUSTERING_DEPTH('TPCH_ORDERS', '(C2, C9)') |
+--------------------------------------------------+
| 11.2452                                          |
+--------------------------------------------------+
```

18. A large table with 200 columns contains two years of historical data. When queried, the table is filtered on a single day. Below is the Query Profile :



Total Execution Time (1h 18m 40.737s)

	(100%)
● Processing	36 %
● Local Disk IO	2 %
● Remote Disk IO	60 %
● Network Communication	0 %
● Synchronization	0 %
● Initialization	0 %

Total Statistics

IO	
Scan progress	98.38 %
Bytes scanned	5.78 TB
Percentage scanned from cache	2.60 %
Network	
Bytes sent over the network	42.17 GB
Pruning	
Partitions scanned	2,115,987
Partitions total	2,956,205
Spilling	
Bytes spilled to local storage	32.94 GB

Using a size 2XL virtual warehouse, this query took over an hour to complete.
What will improve the query performance the MOST?

A. Increase the size of the virtual warehouse.

B. Increase the number of clusters in the virtual warehouse.

C. Implement the search optimization service on the table.

D. Add a date column as a cluster key on the table.

Answer : D

Explanation : The querying is happening by filtering data for a day. So, improvement in performance can come if we create a cluster on the date key.

---

19. The following is returned from SYSTEM$CLUSTERING_INFORMATION() for a table named ORDERS with a DATE column named O_ORDERDATE:

```
{
 "cluster_by_keys" : "LINEAR(YEAR(O_ORDERDATE))",
 "total_partition_count" : 536,
 "total_constant_partition_count" : 493,
 "average_overlaps" : 0.1716,
 "average_depth" : 1.0914,
 "partition_depth_histogram" : {
   "00000" : 0,
   "00001" : 491,
   "00002" : 41,
    "00003" : 4,

    "00004" : 0,

    "00005" : 0,

    "00006" : 0,

    "00007" : 0,

    "00008" : 0,

    "00009" : 0,

    "00010" : 0,

    "00011" : 0,

    "00012" : 0,

    "00013" : 0,

    "00014" : 0,

    "00015" : 0,

    "00016" : 0

  }

}
```

What does the total_constant_partition_count value indicate about this table?

A. The table is clustered very well on O_ORDERDATE, as there are 493 micro-partitions that could not be significantly improved by reclustering.

B. The table is not clustered well on O_ORDERDATE, as there are 493 micro-partitions where the range of values in that column overlap with every other micro-partition in the table.

C. The data in O_ORDERDATE does not change very often, as there are 493 micro-partitions containing rows where that column has not been modified since the row was created.

D. The data in O_ORDERDATE has a very low cardinality, as there are 493 micro-partitions where there is only a single distinct value in that column for all rows in the micro-partition.

Answer : A

Explanation : asdsad

How is Snowflake's virtual warehouse data cache used?

1. It is used to store statistics when data is loaded.
2. It is used to persist query results when the underlying data has not changed.
3. It is used for `SHOW` commands.
4. It is used to minimize how much data needs to be read from cloud storage.

Answer : 4

Explanation :

A Data Engineer is determining how to cluster the following table, which is used to record orders from a food delivery service application.

```
CREATE TABLE orders_dashboard (

        id NUMBER,

        driver_id NUMBER,

        customer_id NUMBER,

        restaurant_id NUMBER,

        ordered_at TIMESTAMP,

        item_count INTEGER,

        total_cost NUMBER,

        order_location_state CHAR(2),

        order_comments TEXT);
```

The purpose of this table is to provide metrics for regional sales managers to understand how deliveries in each region are performing over various timeframes.

Given that queries will be run against this table, what would be the MOST efficient clustering statement for this table?

1. alter table orders_dashboard cluster by (ordered_at, order_location_state);
2. alter table orders_dashboard cluster by (DATE(ordered_at), order_location_state);
3. alter table orders_dashboard cluster by (order_location_state, DATE(ordered_at));
4. alter table orders_dashboard cluster by (order_location_state, ordered_at);

A Data Engineer executes the below query and notices that the execution takes longer than expected.

select col1, col3 from table1 where col2 > 1000;

Which statement will display the number of scanned micro-partitions?

1. select parse_json(select SYSTEM$EXPLAIN_PLAN_JSON(last_query_id())):"GlobalStats":"partitionsAssigned";
2. select parse_json(select SYSTEM$EXPLAIN_PLAN_JSON(last_query_id())):"GlobalStats":"partitionsTotal";
3. select parse_json(select system$clustering_information('table1', '(col1, col3)')):"total_partition_count";
4. select parse_json(select system$clustering_information('table1', '(col2)')):"total_partition_count";

Explanation :

How do you enable search optimization service ?

A) ALTER TABLE my_table ADD SEARCH OPTIMIZATION;

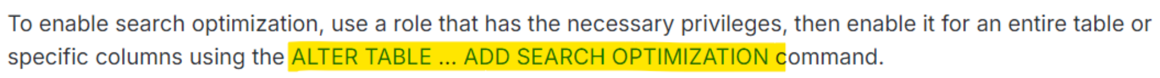
B) ALTER TABLE MODIFY my_table ADD SEARCH OPTIMIZATION;

C) ALTER TABLE my_table SET SEARCH OPTIMIZATION;

D) ALTER TABLE my_table SET SEARCH OPTIMIZATION=TRUE;

Answer : A

Explanation :

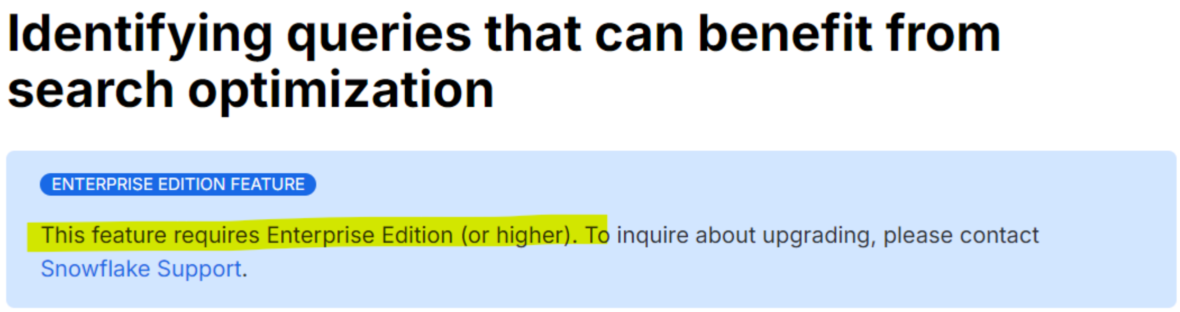


To enable search optimization, use a role that has the necessary privileges, then enable it for an entire table or specific columns using the ALTER TABLE … ADD SEARCH OPTIMIZATION command.

Mark the edition where search optimization service is NOT supported

A) Standard
B) Enterprise
C) Business Critical
D) Virtual Private Snowflake

Answer : A



Identifying queries that can benefit from search optimization

ENTERPRISE EDITION FEATURE

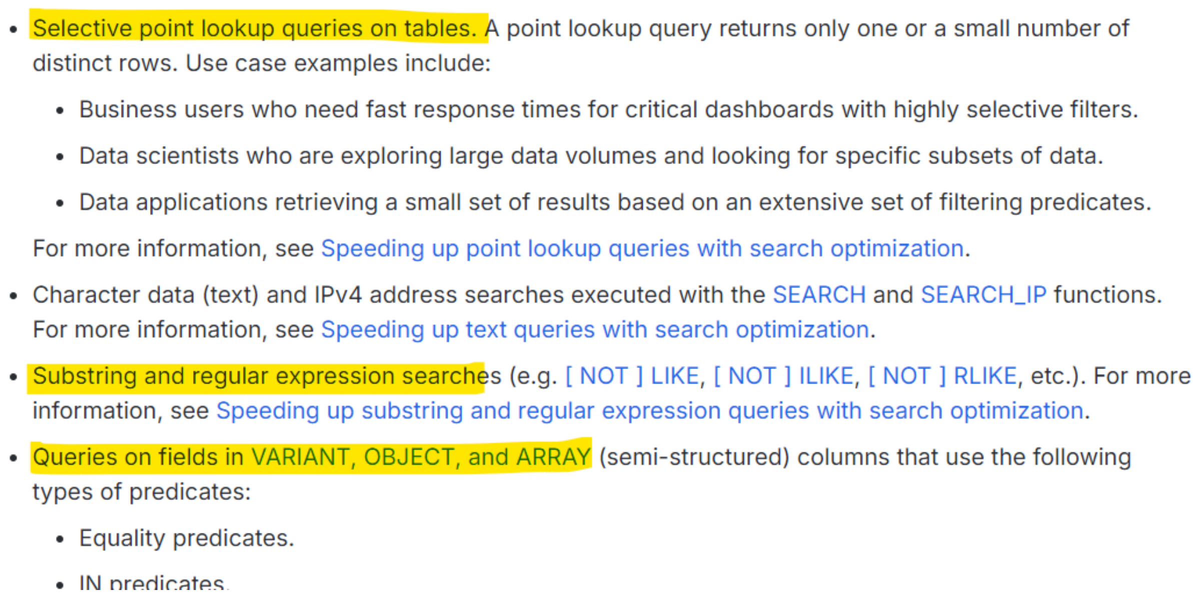
This feature requires Enterprise Edition (or higher). To inquire about upgrading, please contact Snowflake Support.

Mark all the cases where Search Optimization Service will significantly improve query performance

A) Selective point look up queries
B) Substring and regular expression search
C) Aggregation functions
D) Queries on fields of types VARIANT, OBJECT and ARRAY
E) Non-deterministic functions like CURRENT_DATE() etc.

Answer : A, B, D

Explanation : https://docs.snowflake.com/en/user-guide/search-optimization-service



- Selective point lookup queries on tables. A point lookup query returns only one or a small number of distinct rows. Use case examples include:
  - Business users who need fast response times for critical dashboards with highly selective filters.
  - Data scientists who are exploring large data volumes and looking for specific subsets of data.
  - Data applications retrieving a small set of results based on an extensive set of filtering predicates.

  For more information, see Speeding up point lookup queries with search optimization.
- Character data (text) and IPv4 address searches executed with the SEARCH and SEARCH_IP functions. For more information, see Speeding up text queries with search optimization.
- Substring and regular expression searches (e.g. [NOT] LIKE, [NOT] ILIKE, [NOT] RLIKE, etc.). For more information, see Speeding up substring and regular expression queries with search optimization.
- Queries on fields in VARIANT, OBJECT, and ARRAY (semi-structured) columns that use the following types of predicates:
  - Equality predicates.
  - IN predicates.