<https://www.snowflake.com/blog/seven-tips-to-creating-a-successful-customer-advocacy-program/>

<https://community.snowflake.com/s/article/Checklist-Query-Performance>

https://community.snowflake.com/s/article/Troubleshooting-SnowSupport

https://docs.snowflake.com/en/user-guide/tasks-ts

https://community.snowflake.com/s/article/how-to-diagnose-a-slow-running-query

https://careers.snowflake.com/us/en/job/7063170002/Senior-Cloud-Support-Engineer-SQL

[**https://community.snowflake.com/s/article/Snowflake-Client-Connectivity-Troubleshooting**](https://community.snowflake.com/s/article/Snowflake-Client-Connectivity-Troubleshooting)

[**https://docs.snowflake.com/en/sql-reference/functions/system\_allowlist#examples**](https://docs.snowflake.com/en/sql-reference/functions/system_allowlist#examples)

[**https://docs.snowflake.com/en/user-guide/snowcd**](https://docs.snowflake.com/en/user-guide/snowcd)

**https://anywhere.epam.com/en/blog/data-engineer-interview-questions**

**https://www.google.com/search?q=senior+cloud+engineer+sql+snowflake&rlz=1C1CHBF\_enUS877US877&oq=senior+cloud+engineer+sql+snowflake&gs\_lcrp=EgZjaHJvbWUyBggAEEUYOTIGCAEQRRhAMgYIAhAjGCcyBggDEEUYPDIGCAQQRRg9MgYIBRBFGDzSAQg3MDA0ajBqNKgCALACAA&sourceid=chrome&ie=UTF-8&ibp=htl;jobs&sa=X&ved=2ahUKEwirxrHzj6WEAxVHkYkEHSeHAY0Qkd0GegQIEBAB#fpstate=tldetail&htivrt=jobs&htiq=senior+cloud+engineer+sql+snowflake&htidocid=tcYlfXZwToU4BTNgAAAAAA%3D%3D&sxsrf=ACQVn0\_upT9avlxwh-oU4kOWcoC\_3wvi\_w:1707717625151**

1. Stay updated with customer trends.

* Have data office hours with them.
* Keep myself accessible and available.
* Did a course on customer experience.
* Did a data modeling activity to create column names and tables names in a way where the customer experience is excellent
* Introduction pitch.

**Query Performance** :-

- A big part of my role has been addressing customer requests and cases that are assigned to me on the basis of severity and workload.

- In this support role, I work with business users to see what kind of data issues they are encountering. Sometimes it could be query optimization.

**Snowflake as a product**

**What are the different types of Stages?**

**1.** Stages are commonly referred to as the storage platform used to store the files. In Snowflake, there are two types of stages:  
 1. Internal stage — Resides in the Snowflake storage  
 2. External stage — Resides in any of the cloud object storage (AWS S3, Azure Blob, GCP bucket )  
  
([link](https://medium.com/@sanket.prabhu34/commonly-asked-snowflake-interview-questions-e3863732c53f))

**2.** **What is Unique about Snowflake Cloud Data Warehouse?**

- SAAS

- Decoupled storage and compute

- Zero copy cloning.

- Secure data sharing.

- Support semi structured data

- Scalability

- Protection from cloud downtime with replication abilities, to copy the database into different regions (replicated data is read only)

3. A data engineering team has successfully migrated a lot of data into a stage database. The average size of the tables have reached more than 2 tb. The data analytic team is facing performance issues while connecting this database while running complex BI queries. What could be possible issues for query performance and what are the solutions.

Ans: implement clustering keys in medium cardinality columns

4. Privileges to execute snowpipe.([link](https://soumakdas94.medium.com/snowflake-scenario-based-interview-questions-be881a0f8a1e))

1. Ownership of the **snowpipe**
2. USAGE and Read on the source **stage.**
3. **USAGE on the target schema and database**
4. **Insert and select on the target able**

**Why would you like to work for snowflake ?**

- Great experience as a customer where you guys have built out an innovative product, and I would like to be associated with a product that appeals to their customers.

- Secondly, I believe it’s a high growth company where I would get to solve problems of the growing customer base directly while working in the support role.

**Questions to ask to the interviewing panel** ➖

**Customer advocacy**

1. Snowflake usage across industry and regions
2. What metrics or KPIs does the company use to measure the success of its customer advocacy efforts?

**What are the differences between function and stored procedure ?**

* Function returns a value and it doesn’t perform at sort of DML tasks such as insert, updates etc.
* Stored procedures can have an out parameter and it also can apply dml tasks on the tables.
* Stored procedures can be used for transactions on a database.
* We can call a function within stored procedure but we cannot call a stored procedure in a function.
* A function can be used in a stored procedure.
* We can use a function with Select, Having and where statements.

#### 1. What experience do you have in customer service and advocacy?

* Thank you for your question,
* In my current role, 3 yrs of hands on experience in the data engineering in handling internal company wide customers who can range from an analyst to a ceo, as I am actively involved in building out data products for our company’s business users.
* In this role, my customer is the business itself, I enable my coworkers to do their jobs, as almost everyone in the organization is touched up by the data products that I build.
* As a data steward,I actively meet with stakeholders to understand their pain points, and how I can make their life easier.

1. Data quality issues.
2. Query performance issues.
3. Improving the SLA of our datasets.

* I am also part of the incident management team, where I address any incident in the SLA, based on the severity (P0, P1, P2, P3).
* Also I have weekly data office hours on tuesday and thursday, where business users reach out to me via. Live google meet where I help them with data questions, usually it surrounds sql optimization or data dictionary.
* I try to keep myself highly available and highly accessible, so that business users can reach out to me anytime.

#### 2. How would you handle a situation where a customer is angry or frustrated with your company’s product or services?

* Unfortunately, I have been in a position where the SLA delivery data of our critical data pipelines was impacted.
* Just to give business context, we are credit card issuing company and as the nature of the business
* Collection reminders for customers that are unable to pay their debt.
* For that, we have a dependency on external credit data such as fiserv.
* And we have very strict SLAs of 7 am everyday.
* On one of the days, when I was on-call, we had a schema change in the credit data that we receive from fiserv.
* As a result of that our collection reminder pipeline failed at the data ingestion time, and it delayed our anticipated file delivery time.
* As I was on-call, I reached out to the engineering team, to support and it was determined that the work around to apply this fix would require some time.
* Hence I created a work-around in a local notebook, where I manually updated the new columns list in the source file, to ingest the source file from fiserv.
* Eventhough we weren’t able to meet the sla, and the collection reminder was generated at 9.30 am, I was able to make the pipeline work.
* And while working with customers, my goal is to provide a working solution to my customer, eventhough if it is a suboptimal solution but things have to keep moving always.

3. Project that you are most proud about

Project where you went above and beyond for the customer.

* This is good question, I closely work with customers, and a lot of times I get an adhoc request from our customers.
* But I would like to highlight a reporting pipeline that I built out for our marketing team, this one is really special to me.
* To explain the business side, business side of things, we have external partner companies that provides us with potential leads, and we evaluate these leads to decide whether we have to issue a loan or a credit card to them.
* From our side, we have to provide this application decisions data back to our partner then they are able to feed this information into their machine learning models.
* In simple terms, we had about 38 different pipelines where we would send the data to our partners, so maintaining and tracking these projects was sometimes difficult.
* To correct this setup, I took an initiative to create a centralized repository for all the reporting pipelines into one place, and this singular pipeline delivered the data to our partners.
* Eased marketing team’s monitoring and updation process for these pipelines.
* Allowed me to have a strong relationship with our vendors.
* I was able to demonstrate that I am able to come up with creative solutions for our customers, without concrete requirements.
* **Disk Spilling**

1. When the size of the data processed in a query is high then in that scenario the data spills over to the local dish of the warehouse and then the local storage, retrieving data from these datasets is pretty slow.

**Usage and purpose of the execution plan**

1. **Access Path**

* It provides visibility regarding the mechanism of access for the data.
* **Full Table Scan**

1. It indicates whether the full table was searched for getting the required records.
2. Full table scan is inefficient for large tables but it might unavoidable

* **Index Scan**.

1. Much faster than the table scan.
2. If one of the columns has table scan then it is going to locate the suitable records by utilizing the index scan field.
3. Non-clustered indexes are ideal for instances where there are frequent updates to the table.

* **Index Seek**

1. More optimized where it filters the database by going to the exact required location for a fewer records. (B+tree traverses from the root node)
2. Index scan is more suited when we have queries that are suited for range values close to each other. (B+ tree traverses from the leaf pages that are ordered and linked)
3. This is typically more efficient than an index scan, especially for queries that retrieve a small subset of rows.

* **Clustered Index**

1. Special type of index which orders the table based on the primary key.
2. There can be only one clustered index on the table.
3. Beneficial for tables that have frequent range queries as it keeps the data sorted.

* **Bitmap Index Scan:**

1. In some databases, bitmap indexes are used to efficiently handle queries involving multiple columns or conditions.
2. Bitmap index scans involve using bitmap vectors to quickly identify the rows that satisfy the query's criteria.

* **Table or Index Partitioning:**

1. A partitioned table is divided into segments, called partitions, that make it easier to manage and query your data. By dividing a large table into smaller partitions, you can improve query performance and control costs by reducing the number of bytes read by a query.
2. **JOIN Method**

* What type of join is being
* Nested loops, hash joins, and merge joins are different algorithms used by database systems to perform join operations between tables efficiently. Each algorithm has its own characteristics and is suitable for different scenarios.

**Usage**: Nested loops join is efficient when one of the tables is significantly smaller than the other, or when there are suitable indexes to support the join condition.

**Usage**: Hash join is effective when both input tables are relatively large and there's enough memory available to build the hash table. It's particularly efficient for equijoins (joins based on equality conditions).

**Usage**: Merge join is efficient when both input tables are already sorted on the join key, which can often occur when the tables are indexed appropriately or when the optimizer chooses a sort-merge join.

1. **Filtering and Sorting:** The plan will show any filters applied to the data, such as WHERE clauses, and how the database performs sorting operations, if necessary.
2. **Indexes:** If indexes are used, the plan will indicate which indexes are utilized and how they are accessed. This is crucial for optimizing query performance, as using indexes efficiently can significantly speed up data retrieval.
3. **Costs and Cardinality:** Most execution plans include estimates of the cost of each operation and the cardinality, or the number of rows processed by each step. These estimates help in understanding the relative expense of different parts of the query and identifying potential bottlenecks.
4. **Parallelism:** In some cases, the execution plan may involve parallel operations, where the database uses multiple threads or processes to execute parts of the query concurrently. This can improve performance on multi-core systems.

**Cardinality**

* Generally speaking, efficient indexes are high cardinality indexes.
* A good index will have a small number of table values per index value and return a very specific result.
* A good example for a typical high cardinality index are primary keys..

**Micro-partitions**

* Saves data with small blocks of 50 to 500 mb
* It has metadata with information distinct values in the table

**Data Clustering**

* Typically, data stored in tables is sorted/ordered along natural dimensions (e.g. date and/or geographic regions).
* This “clustering” is a key factor in queries because table data that is not sorted or is only partially sorted may impact query performance, particularly on very large tables.

**Trigger vs Constraint**

* Trigger is a special stored procedure that executes certain tasks when there has been an action produced on a table (data inserted into a table)
* Constraint mandates requirements on certain columns like non-null value in the pk value, if the new data doesn’t meet that criteria then it doesn’t get inserted into the table

**Technical issues encountered.**

1. Small file problem

* When ingesting streaming data the data size of the files, maybe smaller than the ideal size of 128 mb, compacting many smaller files causes a slowdown of the process.

1. A business user has a query that runs for a business within 15 mins but for another user it takes 30 mins.

**Query problems examples**

1. A business user who was trying to query a large and wide table, he encountered a long run time, so when I checked the execution plan I could see that in his query there was a disk spilling issue as he was encountering an increased datascan, in that case I reduced his data pull by limiting it to the fields that he needed to be pulled, as well as filtering on an index field.

* Checked the queue
* Checked whether there was a simultaneous write operation not operating on the table
* Checked whether

1. PII access error.
2. Incorrect role assignment leading to incorrect queue assignment.
3. Database engine not online, network connectivity issue that we encounter.
4. Business query not working with date conversion

**Trouble shoot**

1. **Fragmantation error** - caused by small file problem
2. **Compilation ERROR**

* SQL

1. **Out of memory error**
2. **Access issue** - PII data, user role needs to be accessed.
3. **CONNECTION ERROR**: Error setting up remote fragment execution
4. Role wise assignment of warehouse, sometimes a person was assigned a different role when compared to the user. ( this query uses PII data and it seems like you would need to request PII access)
5. Cast error for datetime values, incorrect conversion for Unix\_timestamp values for conversion. ( UNIX\_TIMESTAMP(final\_application\_decision\_time) to (epoch/1800)\*1800)
6. VALIDATION ERROR: Column 'userid' not found in table 'be' SQL Query

**Query Optimization**

1. Understanding the business context behind the query
2. Analyze the execution plan of the query to see whether there are full table scan, index scan etc. that might be taking too long.
3. We can check whether the query is spending a lot of time in queuing, then look for a possibility of separate queue.
4. Review table schema and indexes
5. Review the query to check whether we are pulling on the required data in the query
6. Ensure appropriate join in being used.
7. Limit the results set
8. Check the where condition, whether it can be optimized.
9. Check whether there is disk spilling to local disk and remote storage (then optimize the query make, pull only the data that is needed, decrease the number of parallel queries, increase warehouse loadize).

**Steps to approach**

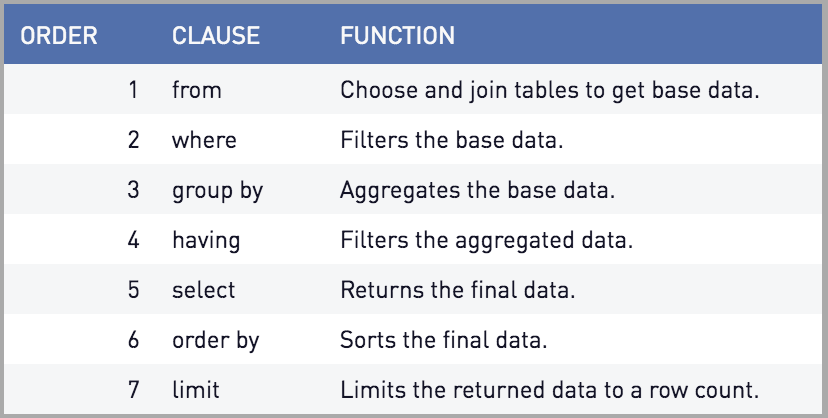
* Fetching more rows than needed
* Fetching all columns from a multi-table join
* Avoid distinct, as it causes the results to be sorted.
* Whether we can add a date filter to the table.
* Whether we can avoid date conversion within the sql
* Converting a complex query to a small queries
* Avoiding multi table join
* Avoid outer join, try to convert it into inner join.
* Sorting results can be a costly operation, so you can often improve performance by avoiding sorts or by performing them on fewer rows.
* Prefer exists over in
* Check for unutilized indexes on the table

Things to check in execution plan

1. Execution time
2. Number of rows examined
3. Number of rows returned

Things to do:-

1. Query performance improvement
2. Research more about the error encountered in sql.
3. SQL questions revisions
4. Triggers vs constraints
5. Function vs stored procedure
6. Order of execution
7. Snowflake
8. High I/O



* Feedback

1. Try to stay calm and understand the question better.
2. Question surrounding the compilation plan.(not comfortable, regarding solving that)
3. Execution plan was the weak part.
4. Speak in more technical terms.\

**Things to remember**

1. Remember to revise the basics.
2. Keep the focus on despite pressure.
3. Try to keep your head down and keep on working.
4. Revise inner join
5. Indexing questions
6. ELT vs ETL
7. Cte vs temp table
8. Materialized view vs view
9. Need to anticipate the behaviorial questions better
10. Columnar storage vs row storage.
11. Data modelling

#### 1. What experience do you have in customer service and advocacy?

Thank you for your question. In my current role, I've had the privilege of collaborating with a diverse range of internal stakeholders over the past three years as a data engineer. From analysts to CEO, my focus has been on developing data products that empower our company's business users to excel in their roles.

Recognizing that our entire organization relies on the data products I build, I approach my work as a team effort. I see my coworkers as my customers, and my goal is to enable them to perform their jobs more effectively by addressing their needs and challenges. This involves actively engaging with stakeholders to understand their pain points, whether it's related to data quality issues, query performance optimization, or enhancing the SLA of our datasets.

As a data steward, I'm deeply involved in incident management, where I work collaboratively with cross-functional teams to address any issues impacting our SLAs. I prioritize incidents based on severity and work diligently to ensure minimal disruption to our operations.

Additionally, I host weekly data office hours on Tuesdays and Thursdays, providing a platform for business users to seek assistance with their data-related queries. Whether it's SQL optimization or navigating the data dictionary, I'm committed to being highly available and accessible to support my colleagues whenever they need assistance.

Through these collaborative efforts, I strive to foster a culture of teamwork and mutual support, recognizing that our collective success hinges on our ability to work together effectively.

#### 2. How would you handle a situation where a customer is angry or frustrated with your company’s product or services?

"In my current role at Avant, I encountered a significant challenge when the SLA delivery of our critical data pipelines was impacted due to a schema change in the credit data received from our external provider, Fiserv. This issue directly affected the generation of collection reminders for customers unable to pay their debts, a process vital for our business operations given our strict SLAs of 7 am daily.

As part of the incident response team, I immediately engaged with the engineering team to address the issue collaboratively. We recognized that resolving the issue would require a workaround while a permanent fix was developed. Leveraging my expertise as a data engineer, I took the initiative to develop a workaround solution using a local notebook. I manually updated the column list in the source file to ensure the ingestion of data from Fiserv, despite the unexpected schema change.

Although the workaround allowed us to generate collection reminders by 9:30 am, missing our SLA, it was a testament to the collective effort and collaboration of the entire team. While my role involved developing the technical solution, it was the combined expertise and teamwork of the engineering team that enabled us to implement and validate the workaround efficiently.

Throughout this process, my focus remained on ensuring minimal disruption to our operations and providing a working solution to our customers. While the workaround may have been suboptimal, it demonstrated our team's ability to adapt and overcome challenges to keep critical processes moving forward.

This experience reinforced the importance of teamwork and collaboration in resolving complex technical issues, highlighting how collective effort and shared expertise can mitigate the impact of unforeseen challenges on business operations."

3. Project that you are most proud about

Project where you went above and beyond for the customer.

This is an excellent question, and it's one that resonates deeply with me as someone who values collaborative efforts. In my role, I'm constantly engaged with our customers, addressing their ad hoc requests and ensuring their needs are met in a timely manner.

One project that stands out as a testament to our team's collaborative spirit is the reporting pipeline I spearheaded for our marketing team. This project holds a special place in my heart because it exemplifies the power of teamwork in solving complex challenges.

On the business side of things, we collaborate with external partner companies who provide us with potential leads. These leads undergo evaluation to determine whether we issue loans or credit cards to them. Subsequently, we must provide application decision data back to our partners for integration into their machine learning models.

Previously, managing the data pipelines for these processes was cumbersome, with 38 different pipelines in place. Recognizing the need for streamlining and efficiency, I took the initiative to create a centralized repository for all reporting pipelines. This consolidated approach not only simplified monitoring and updates for our marketing team but also fostered stronger relationships with our vendors.

Through this collaborative effort, we not only achieved operational efficiency but also demonstrated our ability to innovate and adapt to meet the evolving needs of our customers. It's a testament to the power of teamwork and creative problem-solving, even in the absence of concrete requirements."

This revised response highlights the collaborative effort involved in the project, emphasizing the teamwork aspect and its impact on achieving the desired outcomes.

"In my role as a data engineer for a snowflake, I was tasked with leading a project to develop a new data analytics platform that would streamline our customer support processes and enhance our ability to provide personalized assistance to users worldwide.

From the outset, I recognized that effective communication would be essential for the success of this project, given the diverse geographic locations and cultural backgrounds of our team members. With team members spread across North America, Europe, and Asia, I knew that clear and efficient communication channels would be critical to ensuring alignment and collaboration.

To facilitate communication, I established a structured approach that incorporated both synchronous and asynchronous communication methods. We held regular video conference meetings twice a week to discuss project updates, address any roadblocks, and solicit input from team members. These meetings were scheduled at a time that accommodated the majority of team members across different time zones, ensuring broad participation and engagement.

In addition to scheduled meetings, I maintained open lines of communication through instant messaging platforms, such as Slack, where team members could ask questions, share updates, and seek clarification on project requirements in real-time. This allowed for quick feedback loops and enabled us to address issues promptly, regardless of geographical location.

Recognizing the importance of documentation in facilitating understanding and alignment, I created a centralized repository where all project-related documentation, including design specifications, data models, and implementation plans, were stored and accessible to all team members. This ensured that everyone had access to the latest information and could easily reference key documents as needed.

Throughout the project, I also made a concerted effort to foster a culture of transparency and inclusivity, encouraging team members to share their ideas, voice concerns, and provide feedback openly. By actively soliciting input from team members and incorporating their perspectives into our decision-making process, we were able to leverage the collective expertise of our global team and arrive at more robust solutions.

As a result of our effective communication and collaboration, we successfully delivered the data analytics platform on time and within budget. The platform has since been instrumental in improving the efficiency of our customer support operations and enhancing the overall user experience for our customers worldwide. This experience underscored the importance of effective communication in overcoming geographic barriers and achieving shared goals as a global team."