# Snowflake Interview Questions

- What are the different types of Stages?

  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 )

  Data can be retrieved from the stage or transferred to the stage using the COPY INTO command.

  For BULK loading you can use COPY INTO and for continuous data loading you need to use SNOWPIPE, an autonomous service provided by Snowflake.
  To load data from the local file system into snowflake you can use the PUT command.

- What is Unique about Snowflake Cloud Data Warehouse?

  Snowflake has introduced many unique features that are not been used in any of the other data warehouses currently in the market.

  - ❖ Totally cloud-agnostic ( SAAS ) - Snowflake relies on 3 cloud service providers (AWS, Azure, GCP) for its underlying infrastructure. It provides true SAAS functionality where the user does not require to download or install any kind of software to use snowflake or need to worry about any kind of hardware.

  - ❖ Decoupled storage and compute - By decoupled it means storage and computers work separately and work collaboratively with the interface provided by the cloud provider. This helps in decreasing usage costs where the user pays only what he is using.

  - ❖ Zero copy cloning - This feature is used to take a snapshot of the table at the current instance to take a backup of the table. The snapshot taken will not consume any physical space in the data storage unless any changes have been done on the clone object. This will occupy the same columnar

partition used by the source table. Once changes are done on the cloned object they will be stored in the different micro partitions.

- ❖ Secure data sharing - This feature provides secure sharing of the data with different snowflake accounts or users outside of the snowflake account. By secure, it means you can assign authorized users to access any particular table in order to keep the table secured from the rest of the snowflake users. The shared objects are always in Read-only mode. You can create a Reader account to share data with the user who is not using Snowflake.

- ❖ Supports semi-structured data - Snowflake supports file formats such as JSON, AVRO, ORC, PARQUET, and XML. The variant data type is used to load semi-structured data into snowflakes. Once loaded it can be separated into multiple columns as a table.

  The variant has a limit of 16MB for an individual row. Flatten function is used to split the nested attributes into separate columns.

- ❖ Scalability - As Snowflake is built upon cloud infrastructure, it uses cloud services for storage and computing. The warehouse is a VM that is used to carry out the computation required to execute any query. This enables users the ability to scale up resources when they need large amounts of data to be loaded faster and scale back down when the process is finished without any interruption to service.

- ❖ Time-travel and Failsafe - Time-travel is to retrieve snowflake objects which are removed/dropped from snowflakes. You can read/retrieve data that is deleted within a permissible time frame using time travel.

  Using Time Travel, you can perform the following actions within a defined period of time:

  - Query data in the past that has since been updated or deleted.
  - Create clones of entire tables, schemas, and databases at or before specific points in the past.
  - Restore tables, schemas, and databases that have been dropped.

- What are the different ways to access the Snowflake Cloud Data warehouse?

Snowflake provides WebUI to access snowflake as well as SnowSQL to execute SQL queries and perform all DDL and DML operations including data loading and unloading. It also provides native connectors for Python, Spark, Go, Nodejs, JDBC, and ODBC.

- What are the data security features in Snowflake?

  Snowflake provides below security features:

  1. Data encryption
  2. Object-level access
  3. RBAC
  4. Secure data sharing
  5. Masking policies for sensitive data

- What are the benefits of Snowflake Compression?

  Snowflake stores files in storage as compressed by default as gzip format which helps to reduce the storage space occupied by that file and also improves the data loading and unloading performance. It also detects compressed file formats such as gzip,bzip2,deflate,raw_deflate.

- What is Snowflake Caching? What are the different types of caching in Snowflake?

  It comprises three types of caching :

  1. Result cache- This holds the results of every query executed in the past 24 hours.
  2. Local disk cache- This is used to cache data used by SQL queries. Whenever data is needed for a given query it's retrieved from the Remote Disk storage, and cached in SSD and memory.
  3. Remote cache- Which holds the long-term storage. This level is responsible for data resilience, which in the case of Amazon Web Services, means 99.999999999% durability. Even in the event of an entire data center failure.

- Is there a cost associated with Time Travel in Snowflake?

Yes, Time travel is the feature provided by snowflake to retrieve data that is removed from Snowflake databases.

Using time travel you can do :

1. Query data in the past that has since been updated or deleted.
2. Create clones of entire tables, schemas, and databases at or before specific points in the past.
3. Restore tables, schemas, and databases that have been dropped.

Once the Time travel period is over, data is moved to the Fail-safe zone.
For the snowflake standard edition, the default Time travel period is 1.
For the snowflake Enterprise edition,
for transient and temp DB, schema, tables, the default time travel period is 1.
for permanent DB, schema, tables, and views, the default time travel can ranges from 1to 90 days.

- What is fail-safe in Snowflake

  When the time-travel period elapses, removed data moves to Fail-safe zone of 7 days for Ent. edition snowflake and above. Once data goes to Failsafe, we need to contact Snowflake in order to restore the data. It may take from 24 hrs to days to get the data. The charges will occur from where the state of the data is changed on the basis of 24 Hr.

- What is the difference between Time-Travel vs Fail-Safe in Snowflake

  Time travel has a time period ranging from 0 to 90 days for permanent DB, schema, and tables where Fail safe time is of 7 days only.

  Once the table/schema is dropped from the SF account it will get into Time travel according to the time travel duration of that object (0–90) days.
  Once TT has elapsed, objects move into the Fail-safe zone.

  Snowflake provides us with 3 methods of time travel –

  a. Using Timestamp — We can do time travel to any point of time before or after the specified timestamp.
  b. Using Offset — We can do time travel to any previous point in time.

c. Using Query ID — We can do time travel to any point of time before or after the specified Query ID.

- How does zero-copy cloning work and what are its advantage

  Zero copy cloning is just like creating a clone of the snowflake object.
  You can create clones of SF objects such as DB, schema, table, stream, stage, file formats, sequence, and task.

  When you create a clone, Snowflake will point the metadata of the source object to a cloned object depicting cloning until you make any changes to the cloned object.

  1. Main advantage of this is it creates a copy of the object in less time.
  2. It does not consume any extra space if no updates happen on the cloned object.
  3. fast way to take backup of any object.

  Syntax : create table orders_clone clone orders;

- What are Data Shares in Snowflake?

  Data sharing is the feature provided by snowflake to share data across snowflake accounts and people outside of the snowflake accounts. You can share data according to the customized datasets shared. For people outside snowflake, you need to create a reader account with access to only read the data.

  Below are the objects that can be shared:
  - ❖ Tables
  - ❖ External tables
  - ❖ Secure views
  - ❖ Secure materialized views
  - ❖ Secure UDFs

  There are two types of users:

  1. Data provider: The provider creates a share of a database in their account and grants access to specific objects in the database. The provider can also share data from multiple databases, as long as these databases belong to the same account.

2. Data consumer: On the consumer side, a read-only database is created from the share. Access to this database is configurable using the same, standard role-based access control that Snowflake provides for all objects in the system.

- Where is metadata stored in Snowflake?

  Once the table is created in Snowflake, it generates metadata about the table containing a count of the rows, the date-time stamp on which it gets created, and aggregate functions such as sum, min, and a max of numerical columns.

  Metadata is stored in S3 where snowflake manages the data storage. That's why while querying the metadata, there is no need of running a warehouse.

- Briefly explain the different data security features that are available in Snowflake

  Multiple data security options are available in snowflake such as :

  1. Secure view
  2. Reader account
  3. Shared data
  4. RBAC

- What are the responsibilities of a storage layer in Snowflake?

  The storage layer is nothing but the cloud storage service where data resides. It has responsibilities such as :

  1. Data protection
  2. Data durability
  3. Data Encryption
  4. Archival of Data

- Is Snowflake an MPP database

  Yes. By MPP it means Massively Parallel processing. Snowflake is built on the cloud so it inherits the characteristics of the cloud such as scalability. It can handle parallel running queries by adding necessary compute resources. Snowflake supports shared-nothing architecture where the compute env is shared between the users. When the query load increases, it automatically

creates multiple clusters on nodes capable of handling the complex query logic and execution.

- Explain the different table Types available in Snowflake:

  It supports three types of tables :

  1. Permanent :
  Permanent tables are the default type of tables getting created in snowflake. It occupies the storage in cloud storage. The data stored in a permanent table gets partitioned into micro-partitions for better data retrieval. This type of table has better security features such as Time travel
  The default time travel period for the permanent table is 90 days.

  2. Temporary: Unlike permanent tables, temporary tables do not occupy the storage. All the data stays temporarily in the memory. It holds the data only for that particular session.

  3. Transients: Transient tables are similar to temporary with respect to the time travel period but the only difference is transient tables need to be dropped manually. They will not get dropped until explicitly dropped.

- What are Micro-partitions :

  Snowflake has its unique way of storing the data in cloud storage. Snowflake is a columnar data warehouse as it stores data in columnar format. By columnar, it means instead of storing data row-wise it split the table into columnar chunks called Micro-partitions. Why micro because it only limits each partition to be 50 to 500 MB.

  Snowflake doesn't support indexing; instead it manages the metadata of each micro-partition to retrieve data faster. A relational database when queried uses indexes to traverse all the rows to find requested data. The overhead of reading all the unused data makes the data retrieval time consuming and compute-heavy. Contrary to relational DB, snowflake uses the metadata of MP and checks which chunk or MP contains the data requested by the user. Metadata content the offset and the number of rows consist in that particular micro partition. Using the metadata, snowflake manages all micro-partitions for data storage and retrieval.

- What is the default type of table created in Snowflake.

In addition to permanent tables, which is the default table type when creating tables, Snowflake supports defining tables as either temporary or transient. These types of tables are especially useful for storing data that does not need to be maintained for extended periods of time (i.e. transitory data).

- What view types can be created in Snowflake but not in traditional databases?

  Likewise tables in snowflake there are different types of views that can be created, i.e normal Views, Secure Views, and Materialized Views.

  Normal views are similar to the views found in RDBMS where the output data depends on the query it will run on a table or multiple tables. The query needs to be refreshed in order to reflect the updated data.

  Secure Views prevent users from possibly being exposed to data from rows of tables that are filtered by the view. With secure Views, the view definition and details are only visible to authorized users (i.e. users who are granted the role that owns the View).

  A materialized view is a pre-computed dataset derived from a query specification which is nothing but a SELECT query in its definition. The output is stored for later use.

  Since the underlying data of the given query is pre-computed, querying a materialized view is faster than executing the original query. This performance difference can be significant when a query is run frequently or it is too complex.

- Is Snowflake a Data Lake

  A data lake is normally used for dumping all kinds of data coming from various data sources where it can contain text data, chats, files, images, or videos. The data will be unfiltered, unorganized, and difficult to analyze.

  We cannot use this data to carry any information out of it.

  On a similar basis, the snowflake is supporting structured and semi-structured data with scalable cloud storage providing data lake features along with analytical usage of the data.

By choosing snowflake you get the best of both data lake and data warehouse.

- What are the key benefits you have noticed after migrating to Snowflake from a traditional on-premise database.

  - ❖ Cloud agnostic.
  - ❖ Decoupled storage and computation.
  - ❖ Highly scalable.
  - ❖ Query performance.
  - ❖ supports structured and semi-structured data.
  - ❖ Native connectors such as python, scala , R, and JDBC/ODBC.
  - ❖ Secure data sharing.
  - ❖ Materialized views.

- When you execute a query, how does Snowflake retrieve the data as compared to the traditional databases.

  When the end user executes any query, it first goes to the cloud service layer where it gets optimized and restructured for better performance. The query will be tuned in terms of getting data from the underlying data storage. Also the query gets compiled by the query compiler in the same layer.
  after compilation, it goes to metadata cache to check if the cache has stored any data related to that query

- Explain the difference between External Stages and Internal Name Stages:
  Stages denote where you want to stage (hold) the data in a snowflake.

  There are two types of stages exists in snowflake :

1. Internal stage : In this stage , snowflakes provide a place to hold the data within itself. Data never leave snowflake VPC in this kind of stage.

   it's also gets divided into sub categories as :
   1. User : Each user get automatically allocated stage for data loading
   2. Table : Each table get automatically allocated stage for data loading
   3. Named : Named stages can be created manually for data loading.

2. External stage:

In contrast to the Internal stage, external stages point to locations outside on Snowflake. i.e. Cloud storage buckets ( S3, GCS, Azure blob )

You must specify an internal stage in the PUT command when uploading files to Snowflake.

You must specify the same stage in the COPY INTO <table> command when loading data into a table from the staged files.

- Explain the difference between User and Table Stages.

User stages: Each user has a Snowflake stage allocated to them by default for storing files. This stage is a convenient option if your files will only be accessed by a single user, but need to be copied into multiple tables.

User stages have the following characteristics and limitations:

- ❖ User stages are referenced using @~; e.g. use LIST @~ to list the files in a user stage.
- ❖ Unlike named stages, user stages cannot be altered or dropped.
- ❖ User stages do not support setting file format options. Instead, you must specify file format and copy options as part of the COPY INTO <table> command.

This option is not appropriate if:

- ❖ Multiple users require access to the files.
- ❖ The current user does not have INSERT privileges on the tables the data will be loaded into.

Table stage: Each table has a Snowflake stage allocated to it by default for storing files. This stage is a convenient option if your files need to be accessible to multiple users and only need to be copied into a single table.

Table stages have the following characteristics and limitations:

- ❖ Table stages have the same name as the table; e.g. a table named mytable has a stage referenced as @%mytable.
- ❖ Unlike named stages, table stages cannot be altered or dropped.

- ❖ Table stages do not support transforming data while loading it (i.e. using a query as the source for the COPY command).

Note that a table stage is not a separate database object; rather, it is an implicit stage tied to the table itself. A table stage has no grantable privileges of its own. To stage files to a table stage, list the files, query them on the stage, or drop them, you must be the table owner (have the role with the OWNERSHIP privilege on the table).

- What are the constraints which are enforced in Snowflake?

  Normally there no constraints are enforced in snowflake except for NOT NULL constraints, which are always enforced.

  Usually, in traditional databases, there are many constraints being used to validate or restrict the incorrect data from being stored such as primary key, not null, Unique, etc.

- Snowflake provides the following constraint functionality:

  - ❖ Unique, primary, and foreign keys, and NOT NULL columns.
  - ❖ Named constraints.
  - ❖ Single-column and multi-column constraints.
  - ❖ Creation of constraints inline and out-of-line.
  - ❖ Support for creation, modification and deletion of constraints.

- Difference between Snowflake and other databases?

  Snowflake is a cloud-based data warehousing platform that differs from traditional databases in several ways:

  — Architecture: Snowflake uses a unique architecture called multi-cluster, shared data architecture which separates storage and compute, enabling unlimited scalability and concurrency.
  — Separation of storage and compute: In Snowflake, storage and compute resources are decoupled, allowing users to scale each independently.
  — Pay-per-use pricing model: Snowflake offers a consumption-based pricing model where users pay only for the resources they use, making it cost-effective for varying workloads.

— Built for the cloud: Snowflake is designed specifically for the cloud, offering features such as automatic scaling, data sharing, and native support for semi-structured data types like JSON and XML.

- How will you calculate the expense of a query running in snowflake?

  The cost of a query in Snowflake is typically calculated based on the amount of data scanned or processed by the query. Snowflake provides a feature called “Query History” where users can view details about executed queries including the amount of data scanned, execution time, and cost estimation.

- How to load files in Snowflake?

  Files can be loaded into Snowflake using various methods including:

  — Snowflake’s native COPY command for bulk data loading from files stored in cloud storage services like Amazon S3, Google Cloud Storage, or Azure Blob Storage.
  — Snowpipe, a continuous data ingestion service that automatically loads data from files as they are added to a stage.
  — Using external tables to query data directly from files stored in cloud storage without loading it into Snowflake.

- How to share a table in Snowflake other than the data marketplace?

  Tables can be shared in Snowflake using Snowflake’s data sharing feature which allows users to share data securely between different Snowflake accounts or within the same account. This can be done by granting access to specific objects or by creating a secure view that references the shared data.

- How does Snowflake store data?

  Snowflake stores data in a columnar format using a combination of cloud storage (e.g., Amazon S3, Google Cloud Storage, Azure Blob Storage) for persistent storage and virtual warehouses for compute processing. Data is stored in compressed, encrypted micro-partitions which are managed by Snowflake’s storage layer.

- If I faced an error while loading data what will happen?

If an error occurs during the data loading process in Snowflake, the operation will be aborted, and Snowflake will provide an error message indicating the nature of the issue. Depending on the type of error, you may need to troubleshoot and correct the issue before attempting to reload the data.

- What is Snowpipe?

  Snowpipe is a continuous data ingestion service provided by Snowflake that automatically loads data from files stored in cloud storage as soon as they are added to a specified stage. Snowpipe provides real-time data loading capabilities without the need for manual intervention, enabling near real-time analytics and data processing.

- What is materialized view what are its drawbacks?

  A materialized view is a database object that contains the results of a query that has been precomputed and stored for faster query performance. Drawbacks of materialized views include:

  1. Increased storage requirements: Materialized views store redundant data, which can increase storage requirements, especially for complex queries.
  2. Maintenance overhead: Materialized views need to be refreshed periodically to ensure that the data is up-to-date, which can introduce maintenance overhead and potentially impact system performance.
  3. Limited applicability: Materialized views may not be suitable for all types of queries or workloads, and their effectiveness depends on factors such as query patterns and data volatility.

- How can you implement CDC in Snowflake?

  Change Data Capture (CDC) can be implemented in Snowflake using various techniques such as:
  — Using Snowflake's STREAMs feature to capture and propagate changes from source tables to target tables.
  — Implementing custom CDC solutions using Snowflake's Time Travel and Table History features to track and capture changes to data over time.
  — Integrating Snowflake with external CDC tools or services that support Snowflake's data replication capabilities.

- What if one of the source tables added a few more columns how you handle it at the snowflake end?

  If a source table adds additional columns, you can handle it at the Snowflake end by modifying the target table in Snowflake to include the new columns. This can be done using ALTER TABLE statements to add the new columns to the target table schema, ensuring that the data can be properly loaded and queried in Snowflake.

- How to load data from JSON to Snowflake?

  Data from JSON files can be loaded into Snowflake using Snowflake's COPY command or Snowpipe. Before loading JSON data, you may need to preprocess or transform the data to ensure that it conforms to the required schema and format for Snowflake tables.

- What are secure views and why they are used? How is data privacy done here?

  Secure views in Snowflake are database objects that provide controlled access to underlying data by restricting the columns and rows visible to users or roles. Secure views are used to enforce data privacy and security policies by limiting access to sensitive data based on user roles and permissions.

- What are streams?

  Streams in Snowflake are objects that capture and propagate changes made to tables in real-time. Streams allow users to track changes to data, such as inserts, updates, and deletes, and replicate these changes to other tables or downstream systems for further processing or analysis.

- How can you fetch specific data from the variant columns?

  Specific data from variant columns in Snowflake can be fetched using JSON functions and operators such as dot notation to navigate nested JSON structures, bracket notation to access array elements, and functions like GET and ARRAY_CONTAINS to extract specific values or elements from variant columns.

- How do you load semi-structured data in Snowflake?

Semi-structured data such as JSON or XML can be loaded into Snowflake using Snowflake's native support for semi-structured data types. This can be done using Snowflake's COPY command to load data from files stored in cloud storage, or by directly querying semi-structured data using Snowflake's VARIANT data type and related functions.

- How to create a stage in Snowflake?

  A stage in Snowflake is a named location in cloud storage (e.g., Amazon S3, Google Cloud Storage, Azure Blob Storage) where files are stored for loading into Snowflake. Stages can be created using the CREATE STAGE statement in Snowflake, specifying the location and access credentials for the cloud storage provider.

- What is clustering ?

  Clustering in Snowflake refers to the process of organizing data within tables based on one or more clustering keys. Clustering improves query performance by physically ordering data on disk according to the clustering key, reducing the need for data scanning and improving data locality.

- What is automatic clustering ?

  Automatic clustering in Snowflake is a feature that automatically organizes data within tables based on usage patterns and query history. It analyzes query patterns and access patterns to determine the optimal clustering keys for tables, improving query performance by minimizing the need for manual configuration and tuning of clustering keys.

- If I want to fetch data based on timestamp value is it feasible to cluster the data on timestamp?

  Yes, it is feasible to cluster data on a timestamp if you frequently query the data based on timestamp values. Clustering the data on a timestamp column can improve query performance by physically ordering the data on disk according to the timestamp values, reducing the need for data scanning and improving data locality.

- How will you read hierarchical JSON data, I mean in case it is having an array how would you read that data.

In Snowflake, you can read hierarchical JSON data using JSON functions and operators to navigate the nested structure. If the JSON data contains an array, you can use functions such as FLATTEN to unnest the array and retrieve individual elements. Additionally, Snowflake provides functions like GET and ARRAY_CONTAINS to extract specific values or elements from JSON arrays.

- How to disable fail-safe?

  In Snowflake, fail-safe mode is enabled by default to prevent accidental data loss or corruption. Disabling fail-safe mode is not recommended as it compromises data integrity. However, if you still wish to disable fail-safe mode, you can contact Snowflake support for assistance, as this action may have significant implications for data protection and recovery.

- What is the best approach to recover the historical data at the earliest which was accidentally deleted?

  The best approach to recover historical data that was accidentally deleted in Snowflake is to leverage Snowflake's Time Travel and Data Protection features. Time Travel allows you to access historical versions of tables for a specified period, typically ranging from 0 to 90 days, depending on your Snowflake edition. Additionally, Snowflake's data retention policies and continuous backups provide additional safeguards for data recovery. You can contact Snowflake support for assistance with data recovery procedures and options.

- You have created a warehouse using the command create or replace warehouse OriginalWH initially_suspended=true; What will be the size of the warehouse?

  The size of the warehouse created with the specified command will depend on the underlying configuration and resources allocated to it. However, since the warehouse is initially suspended (`initially_suspended=true`), it will not consume any compute resources until it is manually resumed. Once the warehouse is resumed, its size (i.e., the number of virtual warehouses, nodes, and size of each node) will be determined by the warehouse configuration specified in the command.

- Scenario-Based Questions:

- 1. You have observed that a stored procedure that is getting executed daily at 7 AM as part of your batch process is consuming resources and the CPU I/O is showing as 90%, and the other jobs which are getting executed are impacted due to the store procedure. How can you quickly resolve the issue with the stored procedure?

- 2. Some queries are getting executed on a warehouse and you have executed an Alter Warehouse statement to resize the warehouse, how this will affect the queries which are already in the execution state.

- 3. A new business analyst has joined your project, as part of the onboarding process you have sent him some queries to generate some reports, the query took around 5 minutes to get executed, the same query, when executed by other business analysts, has returned the results immediately? What could be the Issue?