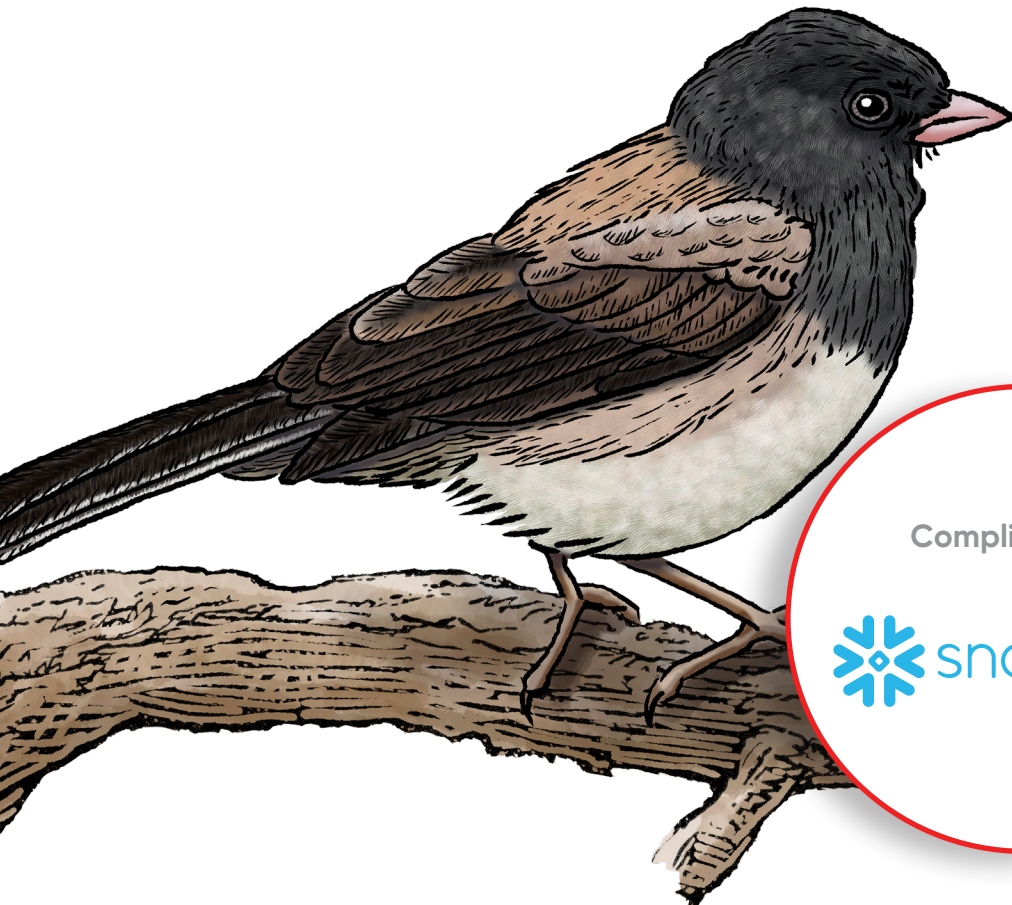


O'REILLY®

# Snowflake

# The Definitive Guide

Architecting, Designing, and Deploying  
on the Snowflake Data Cloud



Compliments of



Joyce Kay Avila

---

# Answers to the Knowledge Check Questions

## Chapter 1

1. The feature to create a support case from within Snowflake is not available for trial or reader accounts. However, there are still plenty of ways to get help (see the following answer).
2. If you are unable to create a support case within Snowflake, you have several options:
  - Consult the [Snowflake documentation](#).
  - Consult the [Snowflake Knowledge Base](#).
  - Solicit support from the [Snowflake Community](#).
  - Review the [Stack Overflow Q&A](#).
  - Obtain direct support (for activation, password, billing, and cancellation requests only) from within the Snowflake Community:
    1. Log in to the [Snowflake Data Heroes Community](#).
    2. Click the “Submit a Case” Quicklink.
    3. Select the account type “I am a 30-day trial user”.
3. Setting the context for a Snowflake worksheet means that you’ll need to let Snowflake know which role, virtual warehouse, and database you want to use in the worksheet. You can set the context through drop-down menus or with the `USE` statements in the worksheet (i.e., `USE ROLE`, `USE WAREHOUSE`, `USE DATABASE`).
4. The “Format query” option, available in the drop-down menu, gives you the ability to neatly format your SQL code. It does not correct spelling errors.

5. A Snowflake certification is valid for two years, after which time a recertification exam must be passed. Note that passing any advanced Snowflake certification resets the clock on the SnowPro core certification, so you will have two more years before you'll need to recertify.
6. You can execute a SQL query in a Snowflake worksheet in two ways:
  - Put your cursor at the beginning of the query and click the blue arrow (the Run button).
  - Highlight all statements and press the Ctrl + Enter keys rather than clicking the Run button.
7. The House icon in the upper-left corner of the screen will return you to the Main menu.
8. We'll be using all uppercase letters for naming our Snowflake objects because Snowflake will convert these names to uppercase anyway, even if they are initially given in lowercase or mixed case. The exception is if you enclose your object name in quotes, in which case the object name will not automatically be converted to uppercase. Instead, it will be saved exactly as you entered it in between the quotes. Note that if you elect to enclose an object name in quotes when you create the object, you'll also need to refer to the object name using quotes in future SQL statements.
9. In the book, we use the `CREATE OR REPLACE` syntax so that you can always go back to any point in the chapter and start at that point without having to drop the objects first. Note, however, that this is a practice you won't want to use in a production environment, because using that statement will cause you to end up overwriting the object. If you mistakenly use the `OR REPLACE` syntax in production, there are Time Travel capabilities you can use to restore the object.
10. When you first create a worksheet, the name defaults to the current date and time.

## Chapter 2

1. The three Snowflake architecture layers are the cloud services layer, the query processing (virtual warehouse) compute layer, and the centralized (hybrid-columnar) database storage layer.
2. The cloud services layer and the centralized (hybrid columnar) database storage layer are multitenant. The query processing (virtual warehouse) compute layer is not multitenant. Note that Snowflake is a multitenant service and the cloud object store is a multitenant service. As such, data is not truly isolated at the public cloud level. It is encryption that creates the isolation. Snowflake's hybrid

tenancy policy, at the public cloud level, uses a multitenant table to consolidate storage but allocates dedicated compute resources to each tenant.

3. The virtual warehouse cache is located in the compute layer. The result cache is located in the cloud services layer. The metadata storage cache layer is located in the cloud services layer.
4. You might want to investigate the following if you are experiencing higher than expected costs for Snowflake cloud services:
  - Simple queries accessing session information or using session variables
  - Large, complex queries with many joins
  - Single-row inserts, as compared to bulk or batch loading
  - Frequent use of `INFORMATION_SCHEMA` commands
  - Frequent use of metadata-only commands such as the `SHOW` command
5. Scaling up is a manual resizing of a virtual warehouse to a larger or smaller size and is most often undertaken to improve query performance and handle large workloads. Scaling out is an automatic process of increasing and decreasing the number of compute clusters and is more often used to maximize concurrency. Scaling out is achieved using multicloud virtual warehouses, which can automatically scale if the number of users and/or queries tends to fluctuate.
6. Compute can be scaled up, down, in, or out. In all cases, there is no effect on storage used.
7. The scalability problem is the main challenge that has been difficult for architectures to solve. Platform architectures need to be scalable to support sharing of the same data at the same time with data-driven teams, large and small, near or far from the data.
8. When Auto-scale mode is selected, the choice of scaling policy is either Standard or Economy:
  - With the Standard scaling policy, the first virtual warehouse immediately starts when a query is queued, or the Snowflake system detects that there is more than one query than the currently running clusters can execute.
  - With the Economy scaling policy, a virtual warehouse starts only if the Snowflake system estimates the query load can keep the virtual warehouse busy for at least six minutes. The goal of the Economy scaling policy is to conserve credits by keeping virtual warehouses fully loaded.
9. You need to configure the following components for multicloud virtual warehouses:

- Mode
    - Auto-scale; can be set to Standard or Economy
    - Maximized; maximized when the Min Clusters value is greater than 1 and both the Min Clusters and Max Clusters values are equal
  - Min Clusters
  - Max Clusters
10. You can change the virtual warehouse via the drop-down menu. You can use the `USE WAREHOUSE` SQL command in the worksheet.

## Chapter 3

1. We can create two main types of databases: permanent (persistent) databases (default) and transient databases. We can create two main types of schemas: permanent (persistent) schemas (default if within a permanent database) and transient schemas (default if within a transient database). And we can create four main types of tables: permanent (persistent) tables (default if within a permanent schema), transient tables (default if within a transient schema), temporary tables, and external tables.
2. A scalar UDF returns a single value whereas a tabular UDF returns multiple values.
3. Unlike user-defined functions (UDFs), a stored procedure can be used to perform database operations such as `SELECT`, `DELETE`, and `CREATE`.
4. If we use the `CREATE DATABASE` command and the database we want to create already exists, we'll receive an error message. If a database already exists and we use the optional keywords `OR REPLACE`, an error will not be returned. Instead, the existing database will be completely overwritten. If you don't want to have an error but also don't want to overwrite an existing database, you can use the `IF NOT EXISTS` optional statement.
5. The default data retention time for a database is one day. We can change the retention time up to 90 days for a permanent database, assuming we have an Enterprise or higher Snowflake edition. However, we cannot change the default retention time; the default retention time will always be one day.
6. The `TRUNCATE TABLE` command removes data from a table whereas the `DROP TABLE` command removes the actual table itself. Note that both the `TRUNCATE` and `DELETE` commands remove data from a table but do not remove the table object itself. The `TRUNCATE` command clears the table load history metadata whereas the `DELETE` command retains the metadata. Therefore, using the `TRUNCATE` command allows an eternal file or a file in a stage to be loaded into the table again.

7. Regular views incur compute costs but not storage costs. Materialized views incur both compute costs and storage costs. A materialized view is a precomputed data set derived from a query specification and stored for later use, thus incurring storage costs. Materialized views are a serverless feature, utilizing Snowflake-managed compute services.
8. A fully qualified object name includes all objects needed to fully identify the object, all separated by a period. An example in which we would want to use the fully qualified name for a table would be `<database_name>.<schema_name>.<table_name>`. A partially qualified object name, without all the qualifiers identified, is allowed but you should understand how partially qualified objects will be resolved. When the database name is omitted, the object is augmented with the current database. When single identifiers are used, how the unqualified objects are resolved depends on whether the object appears in a Data Definition Language (DDL) or Data Manipulation Language (DML) statement, or in a query. More details can be found in the [Snowflake documentation](#).
9. The default file format for stages is CSV. However, you can create file formats for other formats, such as JSON, Avro, ORC, Parquet, and XML.
10. The SNOWFLAKE database is owned by Snowflake Inc. and is a system-defined, read-only shared database which provides object metadata and usage metrics about your account. Unlike the SNOWFLAKE\_SAMPLE\_DATA database imported into your account at the time of setup, the SNOWFLAKE database cannot be deleted from your account.
11. There are two ways to trigger a Snowflake task: tasks can run on a schedule defined at the time the task is created, or task dependencies can be established whereby a task can be triggered by a predecessor task. Note that there is no event source that can trigger a task. A task that is not triggered by a predecessor task must be run on a schedule.
12. METADATA\$ACTION specifies the INSERT or DELETE action. METADATA\$ISUPDATE specifies whether the INSERT or DELETE action is part of an UPDATE applied to the rows in the source table or view. METADATA\$ROW\_ID specifies the unique and immutable ID for the row, which can be used to track changes to specific rows over time.
13. Snowflake streams are a powerful way to handle changing data sets. In Snowflake, one of the most important reasons for using table streams is to keep the staging table and production table in sync. Using a staging table, along with streams, helps to protect undesired changes from being made to the production table. Snowflake table streams are also often used in conjunction with other features, such as Snowflake pipelines and Snowflake tasks.

## Chapter 4

1. The following can be used to make a line of text be a comment instead of being treated as code:

- The COMMENT command
- `/* <commented out code> */`
- `// <commented out code>`

Note that you can comment or uncomment multiple lines by highlighting the lines and then pressing Cmd + / (macOS) or Ctrl + / (Windows). The highlighted lines are commented out.

2. Snowflake's string data types are supported by string constants which are always enclosed between delimiters, either single quotes or dollar signs. Using dollar signs as delimiters is especially useful when the string contains many quote characters.
3. External functions make it possible to use existing machine learning services to extract text from images, or to process PDF files to extract key-value pairs. In an external function, you can use any of the AWS, Azure, or GCP functionalities, such as AWS Rekognition, or Azure Cognitive Services. External functions executed on unstructured data, whether stored within internal or external stages, can be used to eliminate the need to export and reimport data.
4. The default duration for long-running queries is two days. You can use the `STATEMENT_TIMEOUT_IN_SECONDS` duration value at an account, session, object, or virtual warehouse level.
5. Because of the inexact nature of floating-point data types, floating-point operations could have small rounding errors and those errors can accumulate, especially when using aggregate functions to process a large number of rows.
6. Snowflake window functions are a special type of aggregate function that can operate on a subset of rows. This subset of related rows is called a window. Unlike aggregate functions, which return a single value for a group of rows, a window function will return an output row for each input row. The output depends not only on the individual row passed to the function but also on the values of the other rows in the window passed to the function. Window functions are commonly used for finding a year-over-year percentage change, a moving average, a running or cumulative total, and rank rows by groupings or custom criteria.
7. Yes, Snowflake supports unstructured data types.

8. Snowflake supports the following semi-structured data types:
  - VARIANT
  - OBJECT
  - ARRAY
9. Snowflake's `TIMESTAMP` data type supports local time zones. `TIMESTAMP_LTZ` is an internal UTC time with a specified precision; it is a `TIMESTAMP` with local time zone. Daylight saving time is not supported. `TIMESTAMP_TZ` values are compared based on their times in UTC, which does not account for daylight saving time. This is important because, at the moment of creation, `TIMESTAMP_TZ` stores the offset of a given time zone, not the actual time zone.
10. Derived columns, sometimes called computed column values or virtual column values, are not physically stored in a table but are instead recalculated each time they are referenced in a query. Derived columns can be used to calculate another derived column, can be consumed by the outer `SELECT` query, or can be used as part of the `WITH` clause.
11. Following are the three ways you can gain access to unstructured data files in Snowflake:
  - Stage file URLs
  - Scoped URLs
  - Presigned URLs
12. Examples of unstructured data types include video, audio or image files, logfiles, sensor data, and social media posts. Unstructured data, which can be human generated or machine generated, has an internal structure but not one storable in a structured database format.
13. A Snowflake directory table is a built-in read-only table.

## Chapter 5

1. Inherent privileges are those privileges granted to every user who has been assigned a role. More specifically, certain system-defined roles come with inherent privileges. For example, the privileges for resource monitors are inherent to the `ACCOUNTADMIN` role. Note that privileges can also be assigned and inherited, in addition to being inherent to a role.
2. Following are the Snowflake system-defined account roles:
  - `ORGADMIN`
  - `ACCOUNTADMIN`
  - `SYSADMIN`



- SECURITYADMIN
- USERADMIN
- PUBLIC

Most custom roles should be assigned to the SYSADMIN role.

3. Adding defaults for the user makes it easier for the user but adding a default doesn't verify that the privilege has actually been granted to the user.
4. Only an ACCOUNTADMIN can create a new resource monitor.
5. For the ability to view data in a table, a role needs privileges to use the database and schema in which the table resides as well as the ability to use the SELECT command on the table.
6. By default, only the ACCOUNTADMIN role has access to the SNOWFLAKE database.
7. The maximum number of accounts in a Snowflake organization cannot exceed 25 by default; however, you can contact Snowflake support to have the limit raised.
8. Privileges with grant options can be granted to other system-defined or custom roles.
9. The SHOW command returns results based on the role of the user executing the command, so users with different roles could get different results if they have access to different virtual warehouses or different databases, for example.
10. Snowflake has no concept of a super user or a super role. All access to securable objects, even by the account administrator, requires access privileges to be granted explicitly or implicitly by being in a higher hierarchy role.

## Chapter 6

1. Following are the three Snowflake data types used for semi-structured data:

- VARIANT
- OBJECT
- ARRAY

VARIANT is the universal data type.

2. The types of Snowflake stages are as follows:

- Internal stages: user stages, table stages, and internal named stages
- External stages

Internal named stages and external stages are database objects. A user stage and table stages cannot be altered or dropped and, thus, are not separate database objects.

3. The `INSERT ALL` command can be used for either unconditional or conditional multitable inserts.
4. Whenever the key-value pair includes a null value, the key-value pair is not inserted.
5. Following are the error handling options when using the Load Data wizard:
  - Do not load any data in the file.
  - Stop loading, roll back, and return the error (default).
  - Do not load any data in the file if the error count exceeds `<threshold>`.
  - Continue loading valid data from the file.
6. While using the `COPY INTO` command, it is possible to perform basic transformations such as reordering columns or performing casts using a `SELECT` command.
7. Snowflake supports the following semi-structured data types:
  - Loading data types: JSON, Parquet, XML, Avro, ORC (semi-structured) and CSV/TSV (structured)
  - Unloading data types: JSON and Parquet (semi-structured) and CSV/TSV (structured)
8. Following are details regarding Snowpipe REST:
  - Available for use with both internal and external stages
  - Manually calls the Snowpipe REST API endpoint, with the name of a pipe and a list of filenames
  - Passes a list of files in the stage location
  - Best option for use cases in which data arrives randomly and/or if preprocessing needs require using an ETL or ELT tool, or in situations when an external stage is unavailable

Following are details regarding Snowpipe `AUTO_INGEST` (the more scalable approach):

- Available for use with external stages only
  - A notification is received from a cloud provider when a new file arrives
  - Processes new files when awakened
  - Best option for uses cases in which files arrive continuously
9. To prevent resource contention, be sure to isolate data load jobs from queries by dedicating separate virtual warehouses for each. Rather than assuming a large

virtual warehouse will necessarily load massive data files any faster than a small virtual warehouse (it likely won't), make sure to instead try splitting large files into small files of about 100 to 250 MB in size. Remember that the number of files being loaded and the size of each file influence performance more than the size of the virtual warehouse.

10. The `PUT` command loads data into a stage (then the `COPY INTO` command loads data from a stage to a table). The `GET` command unloads data from a stage (after the `COPY INTO` command unloads data from a table to a stage).

## Chapter 7

1. Multifactor authentication (MFA) is intended to be used in addition to strong passwords. Snowflake offers self-service MFA to all account levels so that users can enroll themselves.
2. There is no limit to the number of account network policies that can exist, but there can only be one account-level network policy activated at a time. There is also no limit to the number of user network policies that can exist, but there can only be one user-level network policy, per user, activated at a time.
3. Snowflake's root key is in a hardware security module and is the only key stored in clear text.
4. Data masking policies include dynamic masking, conditional masking, and static masking. Data masking is column based whereas row access policies provide dynamic row-level security. Note that a column cannot have both a masking policy and a row access policy at the same time.
5. The Time Travel retention period is automatically enabled for all Snowflake accounts, and the default is 24 hours but can be set to zero at the account and object levels. For Snowflake Enterprise Edition and higher orgs, the retention period can be set up to 90 days for permanent databases, schemas, and tables.

By setting the retention time as 90 days for a database, all the database objects will also have a 90-day retention time period.

Changing the data retention time at the schema level will result in all tables in the schema inheriting the schema's retention period unless the table is explicitly given a different retention period. Another thing you should keep in mind is that the order in which you drop objects does affect the retention period if there are differences. When you drop a schema, all the existing tables will be available for the same time period as the schema. If you want to make sure the data retention period for child objects is honored, you'll need to drop them prior to dropping the parent object.

6. A tag does not need to be enclosed in quotes. If you do enclose a tag identifier in double quotes, it is case sensitive and can include blank spaces. If an identifier is unquoted, it cannot be case sensitive, cannot contain blank spaces, and must begin with a letter or underscore.

Snowflake's automatic data tagging can be of two different types:

- Semantic
- Privacy

7. We need to flatten query output whenever there is a nested data layer object. This will allow us to convert a nested data layer object into a new object with only one layer of key-value pairs.

In this chapter, we used the `FLATTEN` function to flatten the `DIRECT_OBJECTS_ACCESSED` and `BASE_OBJECTS_ACCESSED` columns in the Snowflake `ACCESS_HISTORY` view.

8. Cloned Snowflake objects are replicated physically, rather than logically, to secondary databases. This means that there will be additional data storage costs for any cloned objects that are replicated.
9. The specific ways we can use Time Travel to access historical data are `SELECT AT/BEFORE`, `CLONE AT/BEFORE`, and `UNDROP`.
10. Time Travel and fail-safe, along with active storage, are part of the total calculated storage that incurs storage costs. Replication and failover/failback charges include data transfer costs and the cost of compute resources. For replications of cloned objects, there will also be additional data storage costs.

## Chapter 8

1. The three major categories of a Snowflake monthly bill are storage fees, data transfer costs, and credits consumed.
2. While storage pricing is a consideration of which region to select for your Snowflake account, you'll also want to consider which region could minimize latency and could give you access to any required features you may need. It's important to note that if you decide later to move your data to a different region, data transfer costs will be incurred.
3. The price of on-demand Snowflake credits is dependent on the cloud provider, region, and service edition. Alternatively, you can prepurchase credits at a negotiated discounted rate.

4. The three properties of a Snowflake resource monitor are the credit quota, credit usage, and triggers. The credit quota and credit usage properties are both required to create a resource monitor. If a schedule is not explicitly stated, the default schedule applies. By default, the credit quota resets at the beginning of each calendar month. Setting triggers is optional.
5. There is no limit to the number of virtual warehouse–level resource monitors that can be created, but there can be only one account-level resource monitor.
6. There is a limit of five Notify actions, one Suspend action, and one Suspend Immediately action for each resource monitor.
7. Making changes to databases presents a unique set of challenges, especially for traditional on-premises databases which have to deal with the problem of down time for hardware and explicitly having to back up databases beforehand so that they will be available for restoration, if needed. In the event a rollback is needed, an operations freeze with no work happening during the restoration process is required. In addition, there is the problem of drift whereby developers could be working on a different version of the database than the version in production, and there is also the problem of load balancing. With Snowflake, these top challenges no longer exist for users. Database change management processes can be handled quite simply in Snowflake by creating a lightweight DevOps framework for small deployments. For a more heavyweight Snowflake DevOps framework, there are database change management tools that can be used.
8. Zero-copy cloning is a metadata-only operation; thus, zero-copy cloning offers the ability to duplicate an object without actually creating a physical copy. As such, there is no additional storage charge for Snowflake cloned objects unless and until changes are made.

Zero-copy clones are frequently used to support working in development and test environments as part of the development lifecycle.
9. An empty table is created using the `CREATE TABLE LIKE` command whereas the `CREATE TABLE AS SELECT` command creates a populated table.
10. Assigning a virtual warehouse resource monitor to a virtual warehouse that is already assigned to a resource monitor will override the previous assignment.

## Chapter 9

1. The `QUERY_HISTORY_BY_USER` function narrows the focus of the query history. `QUERY_HISTORY_BY_SESSION` and `QUERY_HISTORY_BY_WAREHOUSE` are also available for use.
2. The Snowflake `HASH()` function is not a cryptographic hash function. Instead, it is a utility function that returns information such as the descriptions of queries.

3. The two traditional methods by which relational databases can be partitioned are horizontal and vertical.
4. All Snowflake data is stored in database tables. Logically, tables are structured as collections of rows and columns. This logical structure of rows and columns is mapped to the Snowflake physical table structures known as micro-partitions. Snowflake's unique micro-partitioning approach is transparent to users, unlike traditional data warehouse partitioning approaches whereby the user has to independently manipulate partitions by using DDL commands.

A Snowflake micro-partition is inherently small in size and is stored as a columnarized set of rows that holds about 16 MB of compressed data. Storing the data in this way, where columns are stored independently within the micro-partitions, allows for efficient column scanning and filtering. This is especially beneficial for very large tables. Using micro-partitions allows for granular pruning on large tables. Effective and efficient pruning is important because scanning entire tables is not optimal for most queries.

5. DML operations, like an update, will add a new partition block and remove an existing partition block. Micro-partition additions and deletions are tracked in the Snowflake metadata.
6. Snowflake can determine how well clustered a table is by considering how many partitions overlap. Micro-partitions can overlap with each other in a partition; the number of overlaps is known as the clustering width. Micro-partitions can overlap at a specific point in the partition; the number of overlaps at that specific point determines the clustering depth.

Ultimately, the best indicator of a well-clustered table is its query performance.

7. To see the clustering depth in a table, use the `SYSTEM$CLUSTERING_INFORMATION` function. If a table has been clustered, you don't need to specify the column. If it has not been clustered, you'll need to be sure to include the column(s) in your statement.
8. When deciding on a clustering key, a good rule of thumb is to select a key that has enough distinct values for effective pruning, yet not too many values that Snowflake is still able to effectively group rows in the same micro-partitions.
9. When creating the clustering key, it is recommended that you choose no more than three or four columns or expressions. While the number of columns is an important consideration, it is most important to focus on selecting the right columns or expressions for the clustering key and putting them in the best order.
10. Materialized view and the search optimization service are serverless. Table clustering incurs compute costs and is not a serverless feature.

## Chapter 10

1. Direct Share, Data Exchange, Global Marketplace, and Data Clean Room
2. Unlike most traditional databases, Snowflake supports cross-database joins, including joins with databases built from inbound shares. This means that consumers, other than those with reader accounts, can also create views in their own database which can combine their own data with data from the shared database. This provides great benefit because the consumer can enrich their own data to make it even more valuable.
3. An inbound share can only contain one database, but it is possible for a producer to share data from multiple databases by using a secure view. A Snowflake provider can create a secure view that references schemas, tables, and other views from several databases as long as those databases belong to the same account.
4. The Snowflake Marketplace and Data Exchange both utilize Snowflake secure data sharing to connect providers of data with consumers. The Marketplace is a public global marketplace that facilitates sharing among third parties. The Data Exchange is your own private hub for securely collaborating around data, with a selected group of members that you invite.
5. By default, the ACCOUNTADMIN has the inherent privileges to create and maintain data shares.
6. Only the data provider is assessed data storage charges in relation to a data share.
7. The data consumer pays for charges incurred to query a data share. Whenever a reader account is used to query a data share, the charges are paid for by the data provider.
8. *Snowgrid* is the term to describe data replication for the purpose of facilitating data sharing. With the power of secure data sharing and Snowgrid, Snowflake providers can create and share any number of data shares with data consumers worldwide. Snowgrid spans globally, seamlessly connecting Snowflake users who may be separated by region or cloud. Snowgrid achieves this through replication via auto-fulfillment. Even when data is shared across the cloud and regions, the shares are transactionally consistent, meaning the source of truth is still maintained.
9. Snowflake allows you to share the following objects:
  - Databases
  - Tables
  - External tables

- Secure views
- Secure materialized views
- Secure UDFs

Best practices include the following:

- Creating a separate schema for each listing is recommended, but not required. The reason for creating separate schemas is because this could reduce the possibility of confusion and accidentally sharing an object.
  - To ensure that sensitive data in a shared database is not exposed to users in consumer accounts, it is strongly recommended that you share secure views or secure UDFs rather than directly sharing tables.
  - When sharing data from massively large tables, it is recommended that you, as the provider, define clustering keys on the tables.
10. The following requirements must be met in order to receive approval to list on the Snowflake Marketplace:
- Data must be fresh and nonstatic.
  - Data cannot consist of only sample, mocked-up data.
  - Data cannot contain private data.
  - You must possess the right to make the data available.

## Chapter 11

1. You can instruct Snowflake to return a subset of rows via sampling technique in the following ways:
  - Fixed-size sampling based on a specific number of rows (Bernoulli sampling)
  - Fraction-based sampling based on a percentage of table rows:
    - Bernoulli sampling (default)
    - System sampling

There are limits to the number of records that can be used for Snowsight visualizations. The number of rows for automatic statistics visualizations, as well as for charts and dashboards, must be less than one million. Thus, it is often impossible to visualize a full data set in Snowsight. In these situations, it is important to obtain a sample of the data to visualize.

2. A SQL query is not needed to preview the data. You just need to click the Magnifying Glass icon to preview the data. When we preview the data, we see the first 100 rows. We do not need an active virtual warehouse to preview the data.



3. When you rerun a query that uses the `LIMIT` clause, you will get the same rows returned the second time. However, when you rerun a sampling query, you will *not* get the same rows returned the second time as were returned the first time. Sample query results are unique in that the results are not cached.
4. An active virtual warehouse is required to run a sampling query but is not required to preview the data using the Magnifying Glass.
5. You can bucket the data in Snowsight in the following ways:
  - None
  - Date
  - Second
  - Minute
  - Hour
  - Week
  - Month
  - Quarter
  - Year
  - Via filter
6. Deleting a tile from a dashboard permanently deletes the underlying worksheet. You will not be able to retrieve the tile or worksheet after you perform a delete action.
7. Some of the use cases for using Snowsight visualizations include the following:
  - When individuals or small groups of users need to undertake ad hoc data analysis
  - Performing data validation while loading data
  - Quickly creating minor charts and dashboards that can be shared as a team
8. One alternative to deleting a tile is to simply unplace the tile so that it still exists but is not viewable on the dashboard.
9. You can create an unlimited number of charts from one worksheet.
10. Snowflake allows for visualization from internal data sources and external data sources.

## Chapter 12

1. Every week, Snowflake deploys two planned/scheduled releases which may include the following:
  - Full release (deployed any day of the week except Friday)
    - New features
    - Feature enhancements or updates
    - Fixes
  - Patch release

Every month, Snowflake deploys one behavior change release.
2. Snowpark supports the following client-side open API languages:
  - Node.js
  - .NET
  - Go
  - Java
  - Python
  - SQL
3. The Snowflake data platform is said to be developer friendly because it allows app developers to choose among several programming languages.
4. We set `USE_CACHED_RESULT` to `FALSE` because we don't want to take advantage of the cache results for our examples since we are testing the query performance for clustering and the search optimization service. If we didn't change the value to false, we would get accurate performance testing results if we reran the query two or more times.
5. It is more appropriate to use search optimization, rather than clustering, for point lookup queries. Regarding clustering, Snowflake produces well-clustered data in tables, in general. However, over time, particularly as DML occurs on very large tables, the data in some table rows might no longer cluster optimally on desired dimensions. Also, for very large tables in which the ordering is not ideal, there could be a better way to co-locate the data in the same in the same micro-partitions. In both cases, that is where Snowflake's clustering can help.
6. The cybersecurity workload often involves teams who need fast response times for critical dashboards with highly selective filters which often necessitate point lookup queries. As such, the search optimization service, which is useful for point lookup queries, can be used to support the Snowflake cybersecurity workload.

7. The Data Vault modeling approach can take advantage of Snowflake's massively parallel processing compute clusters and optimized columnar storage format.
8. The Data Vault approach and the Kimball approach are relatively simple and require less time to implement as compared to the Inmon methodology. The Kimball approach is based on a four-step normalization process whereas Data Vault uses an agile methodology. In general, the cost to implement the Data Vault modeling approach is the lowest, but Data Vault does require more storage, and is not the best choice when the data is straightforward and auditing of the data is not needed.
9. Data engineers are not responsible for creating governance rules and regulations, but they are tasked with implementing them.
10. The two common pricing strategies for data monetization are cost pricing and value pricing.