**Interview Questions:**

**What is meant by zero copy cloning?**

Zero copy cloning in snowflake is a feature that basically allows to duplicate the source object without making a physical copy of it/data or adding additional storage costs of it. A snapshot of data in source object is taken when a clone is created and made available to clone object. What it does is instead of copying data, it creates a new metadata information pointing to the source database/tables from clone object. So, whenever a user queries a table from clone object, the cloud service simply fetches the data from the actual source. As this operation involves zero cost, it is called zero copy cloning.

We can clone terabytes of data within seconds without incurring computation or storage costs.

**Time travel in snowflake**

It is a feature in snowflake that allows you to access historical data i.e., data before it is altered or deleted within a specified period.

It allows

* Analyzing data manipulations and usage
* Restoring data objects (tables, schemas, and databases) that are accidentally lost or dropped.
* Backup or duplication of data(clones) at or before specified points in past.

When data retention period is completed, data is moved to fail safe, and all the above actions cannot be performed.

**Fail-safe**

After the data retention period, data is moved to fail safe default period which is basically a 7-day period. During this period, we can recover the data that has been lost or damaged, but it takes several days to complete data recovery. It is performed only when all the other recovery options have been exhausted.

**Data retention period**

It specifies how many days historical data will be preserved, allowing time travel operations (select, create, clone, undrop etc.,) can be performed on it. Snowflake preserves the state of data before it is updated.

For Standard objects, default retention period is 1 day, for enterprise accounts it is 0 to 90 days.

**Types of tables in Snowflake**

**Permanent Table**

* Permanent tables are created in the database and exist until they are deleted or dropped.
* These are default table type and give the highest level of data protection and recovery.
* Time travel is possible in these tables up to 90 days.
* It is Fail-safe and data can be recovered if lost due to fail.

**Temporary Table**

* Temporary tables exist for a session.
* If a user wants a temporary table for his subsequent queries and analytics, then once a session is completed, it automatically drops the temporary table.
* It is mostly used for transitory data like ETL/ELT
* Time travel is 0 to 1 day.
* It is not fail-safe, which means data cannot be recovered automatically.

**Transient Table**

* These tables persist until the users drop or delete them.
* Multiple users can access a transient table.
* It is used where "data persistence" is required but doesn't need "data retention" for a longer period.
* Time travel is possible in transient tables but only for 0 to 1 day.
* It is also not failed safe.

**External Table**

* External tables are outside of snowflake, so they can not be deleted or dopped, they have to be removed.
* Data cannot be directly accessed. It can be accessed in Snowflake via an external stage.
* External tables are only meant for reading.
* Time travel is not possible for external tables.
* It is not fail-safe inside Snowflake environment.

**What happens in zero copy cloning when the data is modified in source?**

Cloned objects are independent of the source object and are therefore writable, and any changes made to either object are not reflected in the other.

**What are tasks in snowflake?**

Snowflake tasks are schedulers that can assist in scheduling a single SQL Query or Stored Procedure. Tasks are Snowflake objects to execute a single command, which could be simple SQL command or calling an extensive stored procedure. Tasks can be scheduled or run on-demand, either within a Snowflake Virtual warehouse or serverless.

**Performance tuning techniques in Spark?**

1. Partitioning: Spark processes data in parallel partitions using repartition or coalesce methods based on the size of the data and cluster resources.
2. Broadcasting: Broadcast variables are read-only shared variables that are cached and available on all nodes in a cluster in-order to access or use by the tasks. Instead of sending this data along with every task, spark distributes broadcast variables to the machine using efficient broadcast algorithms to reduce communication costs.
3. Data Caching: Caching is a performance tuning mechanism used to persist intermediate data in memory or on disk so that it can be reused efficiently across multiple Spark transformations and actions.

To cache a Data Frame or RDD in Spark, persist() or cache() methods are used. The difference between these methods is that persist () allows you to specify the storage level (in-memory or on-disk), while cache () uses the default storage level, which is in-memory.