

## Chapter 6: Tables and Views.

As we saw in the chapter on [Snowflake Architecture](#), a Database contains Schemas, which contain other objects. This chapter will discuss two of the most important ones in more detail: the tables and the views.

### Tables:

Tables are database objects that contain all the data, like in any other SQL language. There are different types of tables, and there are some differences between them, which are shown in the following diagram:

	Time Travel	Fail Safe	Cloning	Create views
Permanent	0-90 days.	Si	Permanent Transient Temporary	Yes
Transient	0-1 days.	No	Transient Temporary	Yes
Temporary	0-1 days.	No	Transient Temporary	Yes
External	No	No	No	Yes

Difference between Snowflake tables.

### Permanent tables.

The tables that we can create on Snowflake by default. They have the following characteristics:

1. 0-90 days of Time Travel (Enterprise Edition necessary, 0-1 in Standard Edition).
2. Seven days FAIL-SAFE.
3. You can create a temporary, transient, or permanent table as a clone from permanent tables. If you clone a table to Transient or Temporary, the old partitions will remain permanent, but the new partitions we add to this clone will be Transients/Temporary.

### *Transient tables.*

*Transient tables persist until explicitly dropped and are available to all users with the appropriate privileges. They are similar to permanent tables, with the key difference that they do not have Fail-safe, and they have a shorter Time Travel period. For that reason, we use them for transitory data that we need to maintain beyond each session.*

1. 0-1 day of Time Travel. You CANNOT set 90 days..
2. No Fail-safe.
3. You can create a temporary or transient table as a clone, but not a permanent table as a clone

### *Temporary tables.*

*They store non-permanent data, and they only exist within the session they were created. They only persist for the remainder of the session. As such, they are not visible to other users or sessions. Once the session ends, data stored in the table is purged completely from the system.*

1. 0-1 day of Time Travel. You CANNOT set 90 days..
2. No Fail-safe.
3. You can create a temporary or transient table as a clone, but not a permanent table as a clone

### *External tables.*

*In external tables, the data is stored in an external stage. They store file-level metadata about the data files, enabling querying data stored in files in an external stage as if it were inside a database. You can only perform read operations on these tables, for example, joins or select queries.*

1. No Time Travel.
2. No Fail-safe.
3. No Cloning.

External tables are helpful when you have objects in the Cloud Provider that cannot be copied elsewhere by regulations, or you have lots of data in the Cloud Provider, and you only need a part of it in Snowflake.

### Views:

A view allows us to access the result of a query as a table. An example of using a view could be when you have a BI tool that has to access a variant column (column with non-structured data like JSON, which is also possible in Snowflake, as we will study later). You can use a view almost anywhere where a table can be used (joins, subqueries, etc.). We have two types of views, Non-Materialized and Materialised views, which can be Secure or Not Secure. In this diagram, we can see the differences, and we will see them below in more depth:

	Edition	Storage	Clustering	Sharing	Stream
Regular	All editions	No	No	No	No
Materialized	Enterprise	Yes	Yes	No	No
Secure	All editions, but you need Enterprise for secure materialized views.	Regular no Materialized yes	Regular no Materialized yes	Yes	No

Different types of views in Snowflake.

### Regular View (Non-Materialized Views):

The most common type of view. The results are NOT stored for future use, so the performance is slower than materialized views.

1. You don't store the data.
2. No clustering as you don't store the data.
3. You can only share these views if they are secure.
4. You cannot create streams from any view.

Streams from views have been added as a preview, so we will probably see them this year on Snowflake!

## *Materialized View:*

*In many ways, it behaves more like a table. The results are stored to allow faster access but require storage space and active maintenance, which incur additional costs. You need an enterprise edition to be able to create them. Changes to a base table are not automatically propagated to materialized views based on that table (for example, if you add a column in the table, this column is not propagated here. If you delete a column or even the table, this view will give you an error). You can also do clustering in Materialized views.*

- 1. You store the data.*
- 2. You can do clustering as you store the data.*
- 3. You can only share these views if they are secure.*
- 4. You cannot create streams from any kind of view\**

*This view incurs both storage and computing costs. Storage because the information is stored as if it were a table, as we said before, and compute costs because they are automatically and transparently maintained by Snowflake (new data that enters on the table will be propagated here). When should we use materialized views?*

*Query results contain a small number of rows and/or columns relative to the base table (the table on which the view is defined). Query results contain results that require significant processing, including the analysis of semi-structured data or aggregates that take a long time. The query is on an external table, which might have slower performance. The view's base table does not change frequently.*

## *Secure View :*

*For security or privacy reasons, you might not wish to expose the underlying tables or internal structural details for a view. The view definition and details are only visible to authorized users with secure views. You cannot see the syntax of how the view was created from the table. They are mandatory to share views (Sharing process that we will see later). Both Regular and Materialized views can also be at the same time Secure Views.*

*You store the data depending on if they are materialized or no-materialized. You do clustering depending on whether they are materialized or no-materialized. You can share these views You cannot create streams from any view*

## TYPICAL EXAM QUESTIONS

1. What types of tables are available in Snowflake:

1. Permanent
2. Temporary
3. Transient
4. External/Internal

Solution: 1, 2, 3, 4

2. What types of views are available in Snowflake?

1. Regular
2. Secure View
3. Table View
4. Materialized View
5. External View

Solution: 1, 2, 4

3. We need to temporarily store intermediate data, which will only be used by an ETL process. We don't need the data outside the ETL process. If you want to optimize storage cost, what type of table will you create to store this data?

1. Permanent
2. Temporary
3. Transient
4. External

Solution: 2.

4. Can you use streams to track changes in materialized view?

1. True
2. False

*Solution: 2. This is a limitation of the views.*

5. In which of the below scenarios will you use an external table?

1. You have data on the cloud providers, but the data cannot be copied or moved to any other location due to compliance regulations.
2. You have a high volume of data on the cloud providers, but we only need a part of the data in Snowflake.
3. You have data on the cloud providers which needs to be updated by Snowflake.
4. You have XML data on the cloud provider.

*Solution: 1, 2. The third answer is not correct as external tables can only read data. They also don't support XML.*

6. Select two limitations with materialized views:

1. Time Travel is not supported.
2. We cannot define Cluster Keys.
3. Streams cannot track changes.

*Solution: 1, 3*

7. Can temporary tables be created with a clustering key defined?

1. True
2. False
- 3.

*Solution: 1*

*Thanks for Reading!*

Let me know if this was helpful. If you have any concerns and suggestions please feel free to reach out. Whatsapp +91-7999498574

LinkedIn: <https://www.linkedin.com/in/avinash-sharma-553378151/>

*Regards,*

*Avinash Sharma*