1. Snowflake:
   1. Without the use of External storage with the use of URL, key\_id, and secret\_id:

* Create a table with the necessary columns in Snowflake.
* Create a stage and give the URL of the s3 bucket. Then we create an IAM user with full S3 access and then create an access CLI key where we get credentials of the AWS key ID and AWS secret key which we will be using for accessing the s3 bucket for copying data from s3 🡪 staging in Snowflake 🡪 table we created in snowflake and using aws key id and secret key is not the recommended way because it will increase the vulnerability of our AWS resources.
* Before loading data we first need to create a staging where we load the raw data and then we transfer data from the staging 🡪 table using copy command.

**(code of this can be found in the code file)**

* 1. With the use of External storage without the use of key\_id and secret\_id:
* Create a table with the necessary columns in Snowflake.
* Create a stage and give the URL of the s3 bucket. Then we create an IAM role with full S3 access by attaching a policy created with full access to the s3 bucket in this method we will transfer the data from s3 to staging by creating internal storage We will make use of key components that will give an overview of the type of transferring we are going on with like from which AWS resource we are making use off key’s like TYPE, STORAGE\_PROVIDER, STORAGE\_AWS\_ROLE\_ARN, STORAGE\_LOCATION.
* Moving on after processing the above step we will create staging and then go for coping data from s3 🡪 staging then from staging 🡪 table.

**(code of this can be found in the code file)**

* 1. With the use of pipe in Snowflake to perform auto data ingestion:
* Table creation and use of s3 URL is the same in this method we will create a pipe to perform auto ingestion of data from s3 🡪 staging 🡪 table.
* If new data gets added to the s3 bucket till the pipe is live data gets ingested to the destination table.

**(code of this can be found in the code file)**