Snowflake Badge 1: Data Warehousing Workshop

What benefits do you think Snowflake might offer companies and users, because it is a cloud-based Software-as-a-Service?

- No software installations to be carried out by Tsai and her coworkers (or you and yours)
- No hardware purchases and/or software upgrades or maintenance windows for Tsai and her coworkers (or you and yours)

What benefits do you think Snowflake might offer companies and users, by separating the storage of the data and the engines that perform computations and by using a pay-as-you-go model?

- No need to create data silos (copies of data) to add speed.
- Unlimited flexibility to balance speed and cost as needed (and make the changes immediately).

What benefits do you think Snowflake might offer companies and users, because of it's data sharing technologies?

- Fewer meetings, phone calls, and emails to decide exactly how data will be output by another company and picked up by yours.

- Fewer late night and weekend fire drills when something in the data transfer and load process breaks.
- Fewer emergency situations to get the data loading in time to make sure everything else doesn't also break.
- More time to focus on exploring and analyzing data, because you're not spending time transferring data and loading it.

Database, Schema

Databases are used to group datasets (tables) together. A second-level organizational grouping, within a database, is called a schema. Every time you create a database, Snowflake will automatically create two schemas for you.

The INFORMATION_SCHEMA schema holds a collection of views. The INFORMATION_SCHEMA schema cannot be deleted (dropped), renamed, or moved.

The PUBLIC schema is created empty and you can fill it with tables, views and other things over time. The PUBLIC schema can be dropped, renamed, or moved at any time.

Defining "Warehouse" in Snowflake:

- People who have been working with data for awhile might think of the term "Data Warehouse" as referring to a special collection of data structures, but in Snowflake, warehouses don't store data.
- In Snowflake, Warehouses are "workforces" -- they are used to perform the processing of data.
- When you create a Warehouse in Snowflake, you are defining a "workforce."

Teams are Clusters, Team Members are Servers:

- In the video, the workforce of each warehouse is a team. A small warehouse has a small team, but just one team. An extra-large warehouse has a large team, but just one team.
- Snowflake Warehouse Sizes like eXtra-Small, Small, Medium, etc. all have one cluster. A small warehouse has one cluster made up of just a few servers. A larger warehouse has one cluster, made up of more servers.

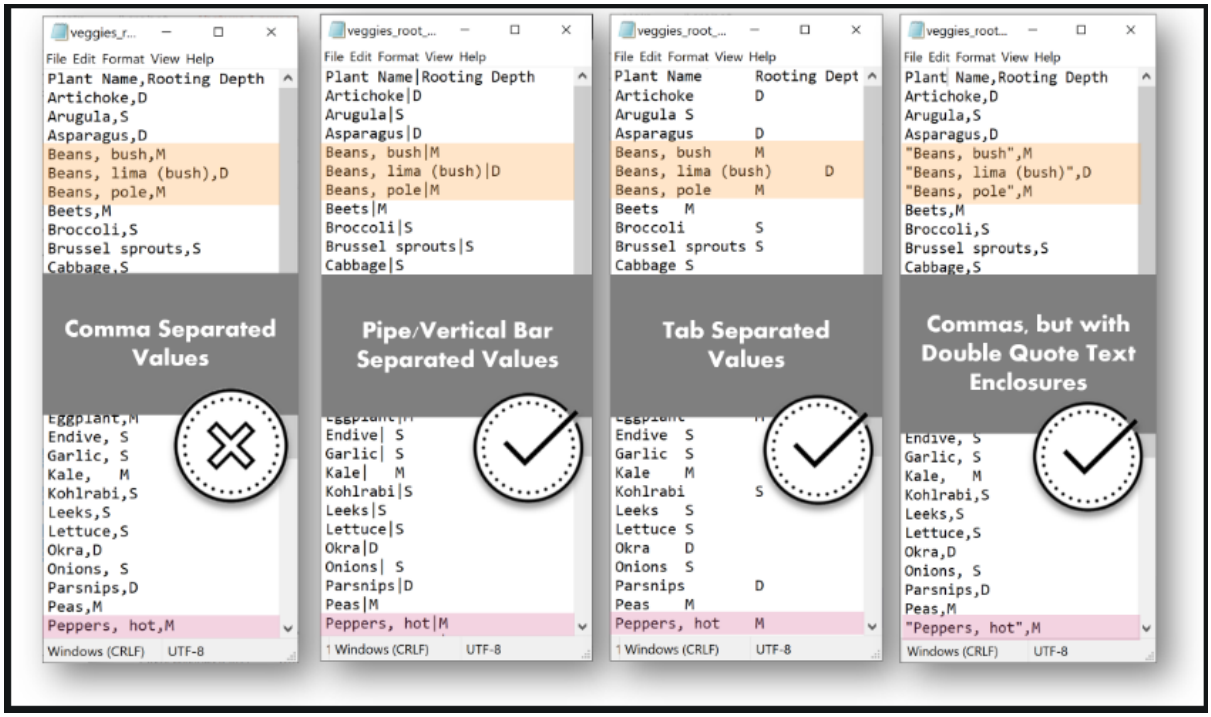
Scaling Up and Down:

- Changing the size of warehouse changes the number of servers in the cluster.
- Changing the size of an existing warehouse is called scaling up or scaling down.

Scaling In and Out:

- If multi-cluster/elastic warehousing is available (Enterprise edition or above) a warehouse is capable of scaling out in times of increased demand. (Adding temporary teams, made up of a collection of temporary workers).
- If multi-cluster scaling out takes place, clusters are added for the period of demand and then clusters are removed (snap back) when demand decreases. (Removing temporary teams).
- The number of servers in the original cluster dictates the number of servers in each cluster during periods where the warehouse scales out by adding clusters.

Loading of file



📓 Using the INFORMATION_SCHEMA to Query Metadata

- The word "metadata" means "data about data."
- The INFORMATION_SCHEMA that gets created in every Snowflake Database holds metadata. In other words, it holds statistics about the number of databases, schemas, tables, views and more. It also holds data about the object names and other object details.
- Notice that in all the queries above, we are using the INFORMATION_SCHEMA to double-check our work and make sure we completed the tasks correctly.

WHAT IS STAGE ?

`Stages` or `Staging area` are places to put things temporarily before moving them to a more stable location.

Snowflake stages are more like cloud folders or directories where you place files so that Snowflake services can pick them up and pull them into database.

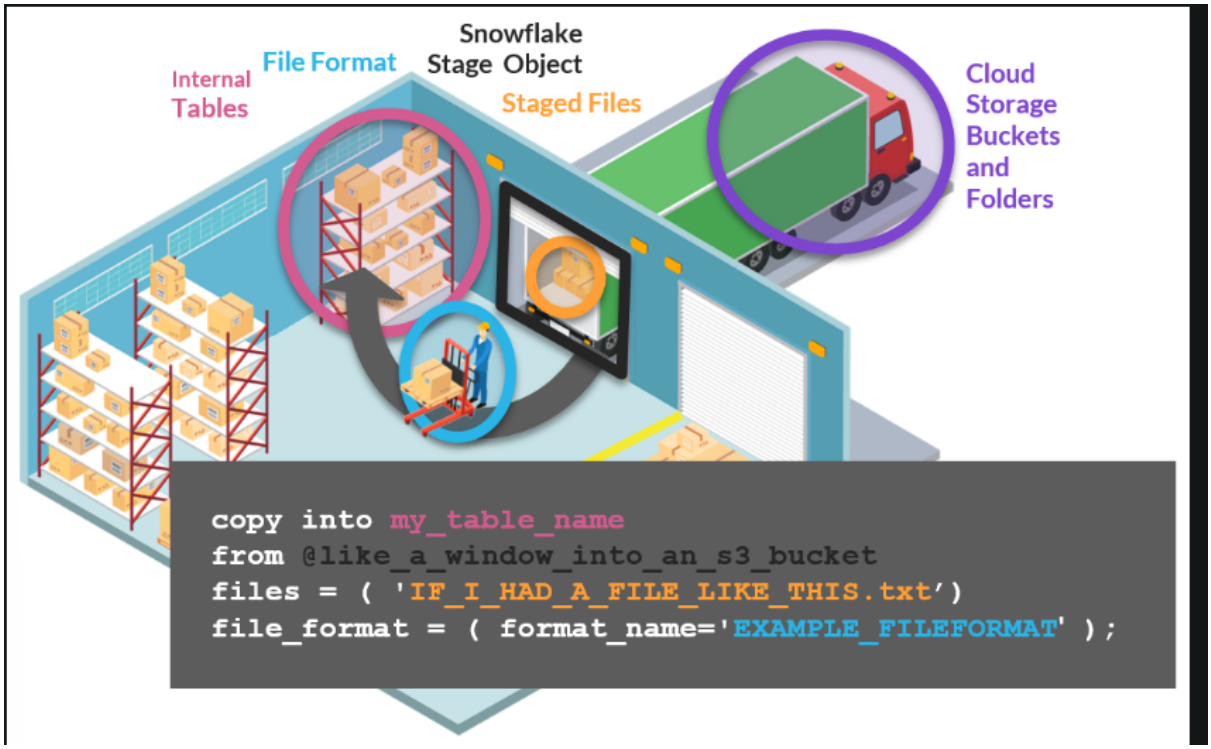
Snowflake has 2 type of stages

1. Internal Stages - which act almost like directories inside a snowflake account's local storage.
2. External Stages - it act more like a secure gateway between cloud storage services and snowflake services
    - To create an external storage based on AWS, you need 3 things
        - Cloud Storage Location, such as an S3 bucket
        - Cloud storage access credentials, AWS IAM user and Policy
        - Stage Definition - define a stage object in snowflake that contains references to those first two things.

📓 A Stage or Not A Stage?

- One weird and confusing thing about stages in Snowflake is that the stage object you just created is not a location at all. The location (the S3 bucket that holds the files) already existed. So what did you just create?
- Well, you created something that tells Snowflake some information about a location where some files are already staged. You didn't create the actual stage location, you created something more like a window into that stage location. Your Snowflake stage object is almost like a File Format in that it holds configuration information that makes loading files easier.
- Sometimes when we define a Snowflake stage we also provide access credentials

Load a File from the S3 bucket into the New Table



Remember that to use the COPY INTO statement, it is best to have 4 things in place:

- **A table**
- **A stage object**
- **A file**
- **A file format**

The file format is sort of optional, but it's a cleaner process if you have one, and we do!

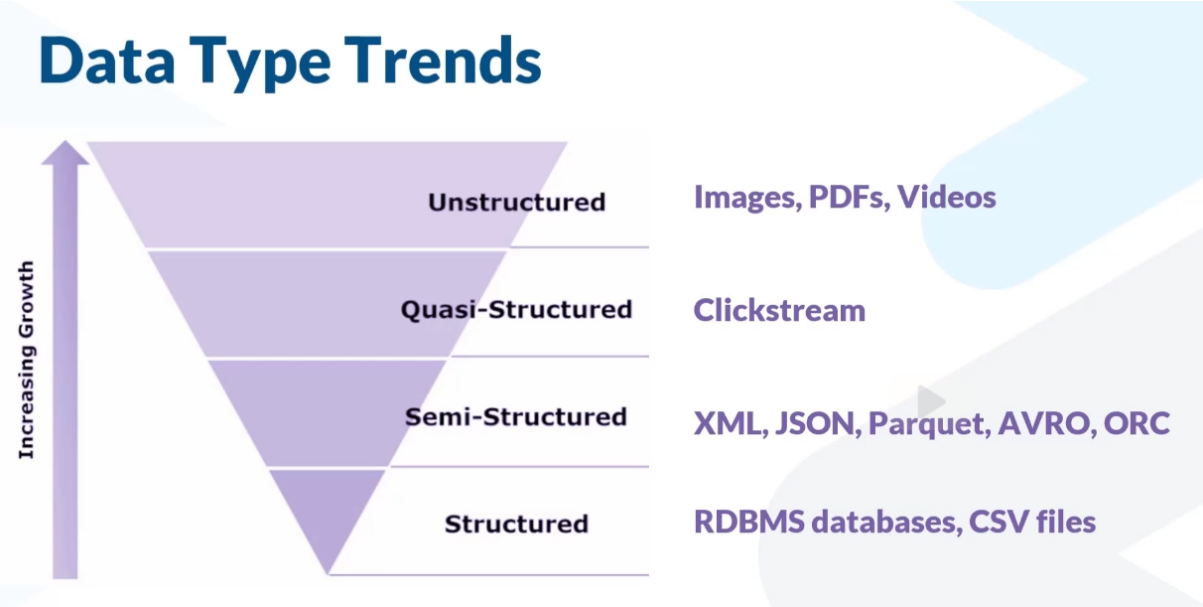
```
copy intomy_table_name
from @like_a_window_into_an_s3_bucket
files = ( 'IF_I_HAD_A_FILE_LIKE_THIS.txt')
file_format = ( format_name='EXAMPLE_FILEFORMAT' );
```

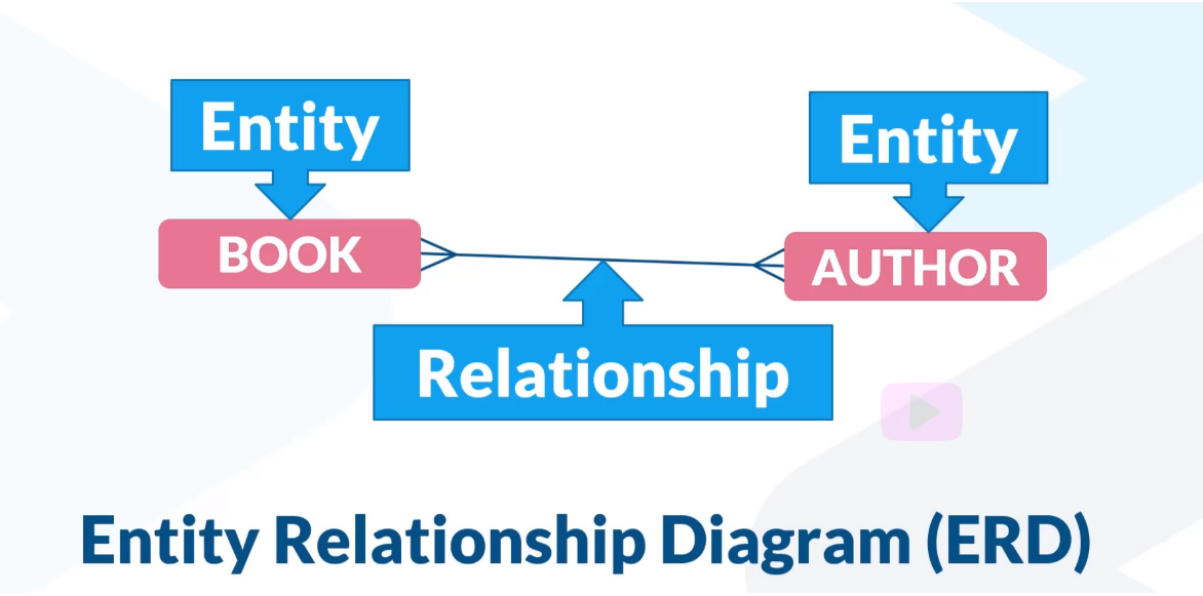
CREATING A FILE FORMAT

```
create file format garden_plants.veggies.PIPECOLSEP_ONEHEADROW
    TYPE = 'CSV'--csv is used for any flat file (tsv, pipe-separated, etc)
```

```
    FIELD_DELIMITER = '|' --pipes as column separators
    SKIP_HEADER = 1 --one header row to skip
    FIELD_OPTIONALLY_ENCLOSED_BY='"'
    ;
```

Data type Trends





Normalized Relational Database Model

BOOK_UID	TITLE	YEAR_PUBLISHED
1	Food	2001
2	Food	2006
3	Food	2008
4	Food	2016
5	Food	2015

AUTHOR_UID	FIRST_NAME	MIDDLE_NAME	LAST_NAME
1	Fiona		Macdonald
2	Gian	Paulo	Faleschini
3	Laura	K	Egendorf
4	Jan		Grover
5	Jennifer		Clapp
6	Kathleen		Petelinsek

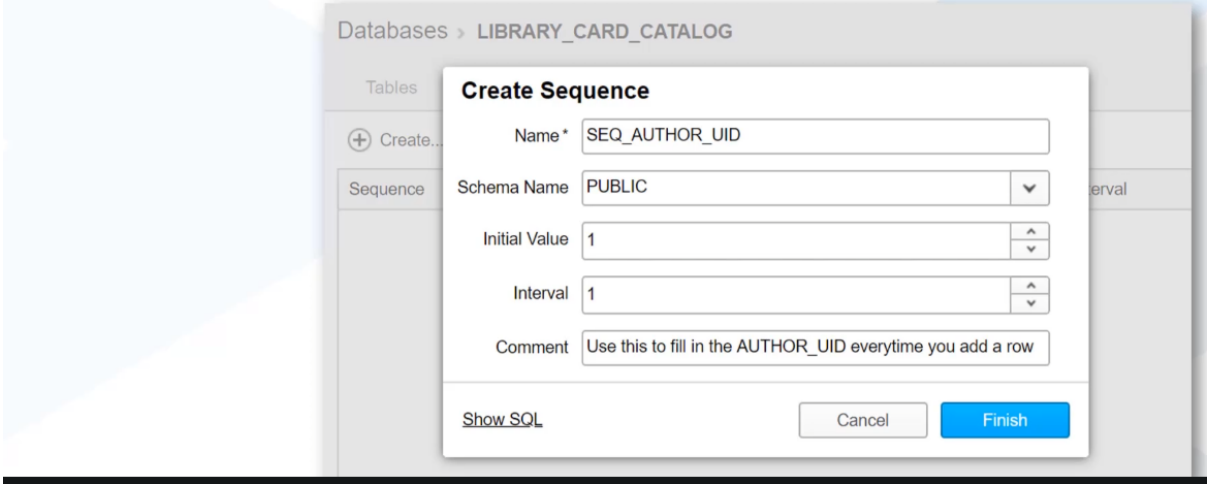
- Different entities in separate tables
- Same info not repeated unnecessarily
- Unique IDs for each row

BOOK_UID	AUTHOR_UID
1	1
1	2
2	3
3	4
4	5
5	6

When it comes to adding unique IDs to tables, Snowflake has a special type of counter that helps you keep track of what numvers should come next.

SEQUENCE OBJECTS

Unique IDs for each row



A sequence is a counter. It can help you create unique ids for table rows. There are ways to create unique ids within a single table called an AUTO-INCREMENT column. Those are easy to set up and work well in a single table. A sequence can give you the power to split information across different tables and put the same ID in all tables as a way to make it easy to link them back together later.