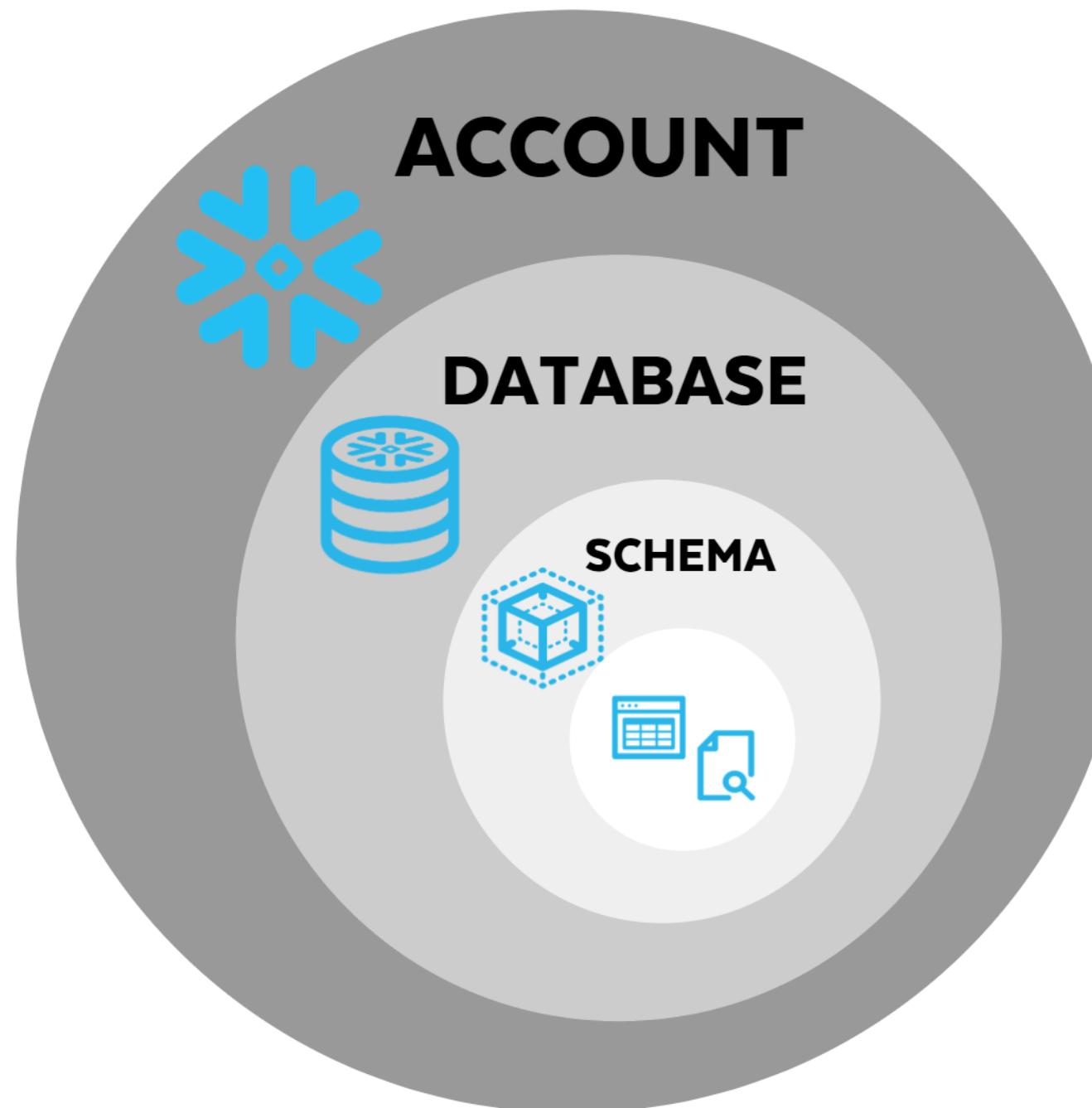


STORAGE HIERARCHY



Snowflake Accounts

- A customer can have as many accounts as they want
- Each account has its own URL – which uses the name of the account.
- Each account is deployed on a single cloud provider platform (AWS, Azure, GCP)
- Each account exists in a single geographic region
- Each account exists with a single Snowflake Edition (e.g. Standard, Premier, etc.)

Accounts Contain Databases

- Each database belongs to a single Snowflake account
- Databases can be replicated to other accounts, but they cannot SPAN multiple accounts

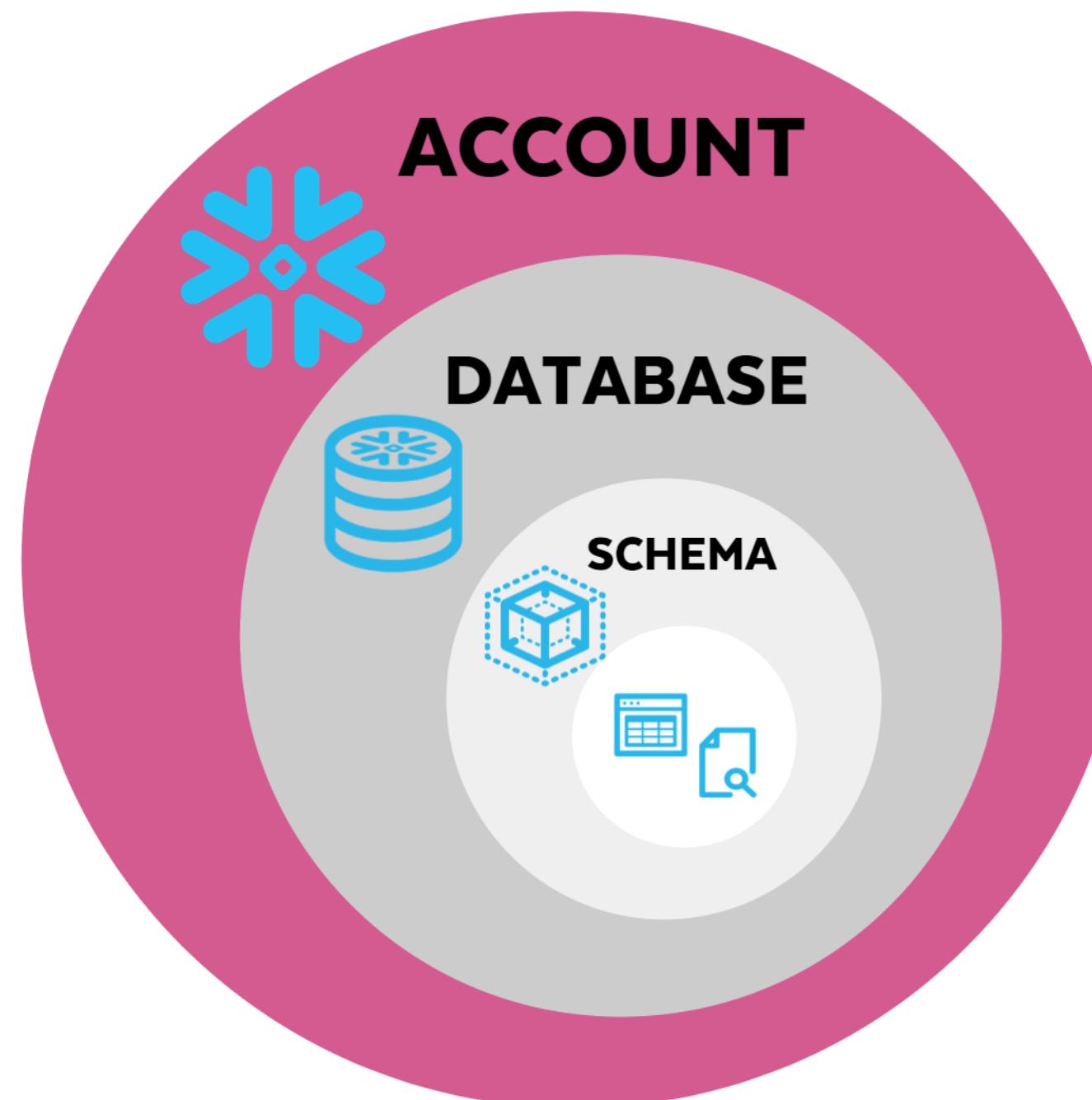
Databases Contain Schemas

- Each schema belongs to a single Snowflake database, in a single account
- Schemas can be replicated to other accounts or databases, but they cannot SPAN accounts or databases

Schemas Contain Other Objects

- Objects include tables, views, file formats, sequences, UDFs, & stored procs
- Objects belong to a single schema, in a single database, in a single account

STORAGE CONTAINER HIERARCHY



READ MORE ABOUT REGIONS HERE:

<https://docs.snowflake.net/manuals/user-guide/intro-regions.html>

Snowflake Accounts

The account shown here is the account NS61174. Most trial accounts start with two letters followed by 5 numbers. Trial accounts can be converted to paid accounts.

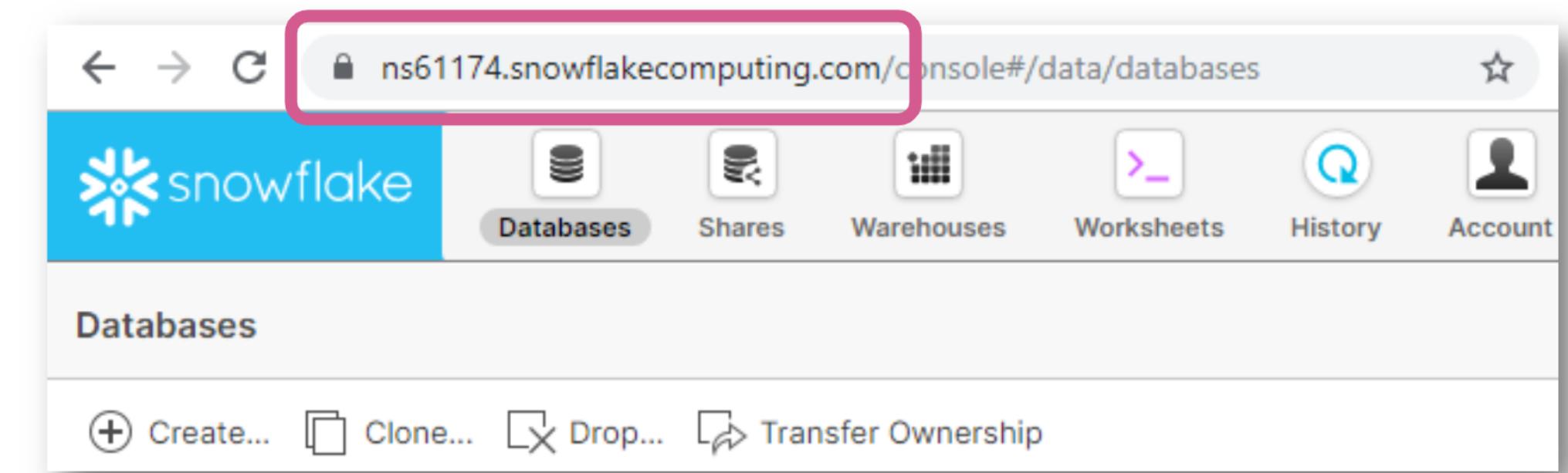
Customers can request a “vanity” address like:

<https://robertsfishandchips.snowflakecomputing.com>

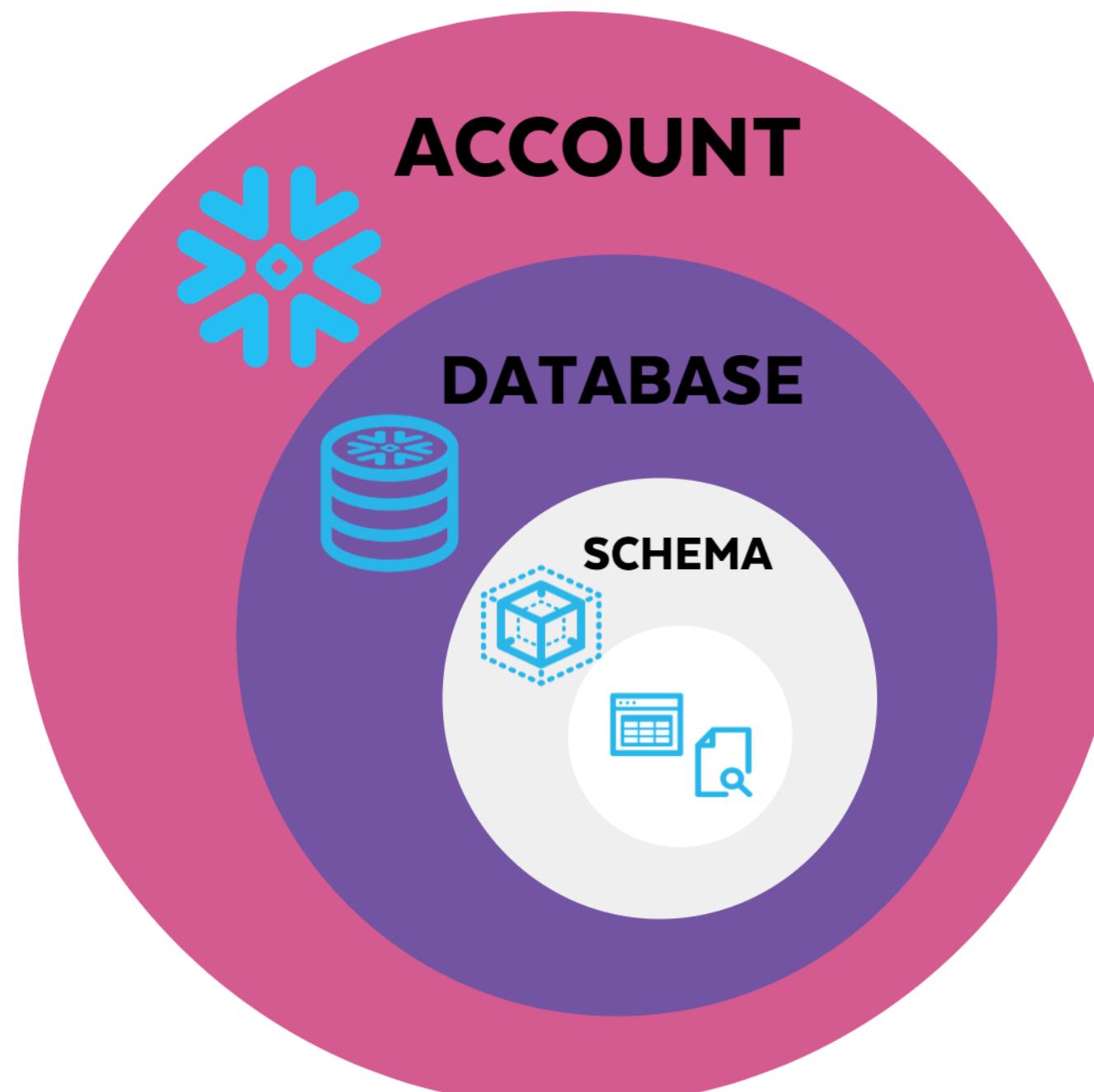
When an account has just the account name, followed by “snowflakecomputing.com”, we know the cloud provider is **AWS** and the region is **us-west-2**. This is because this was the first cloud provider and region on which Snowflake deployed.

Later, when Snowflake expanded to other **regions** and **providers**, the region name was added to the account URL. For example, an address like:

<https://xy12345.east-us-2.azure.snowflakecomputing.com>



STORAGE CONTAINER HIERARCHY



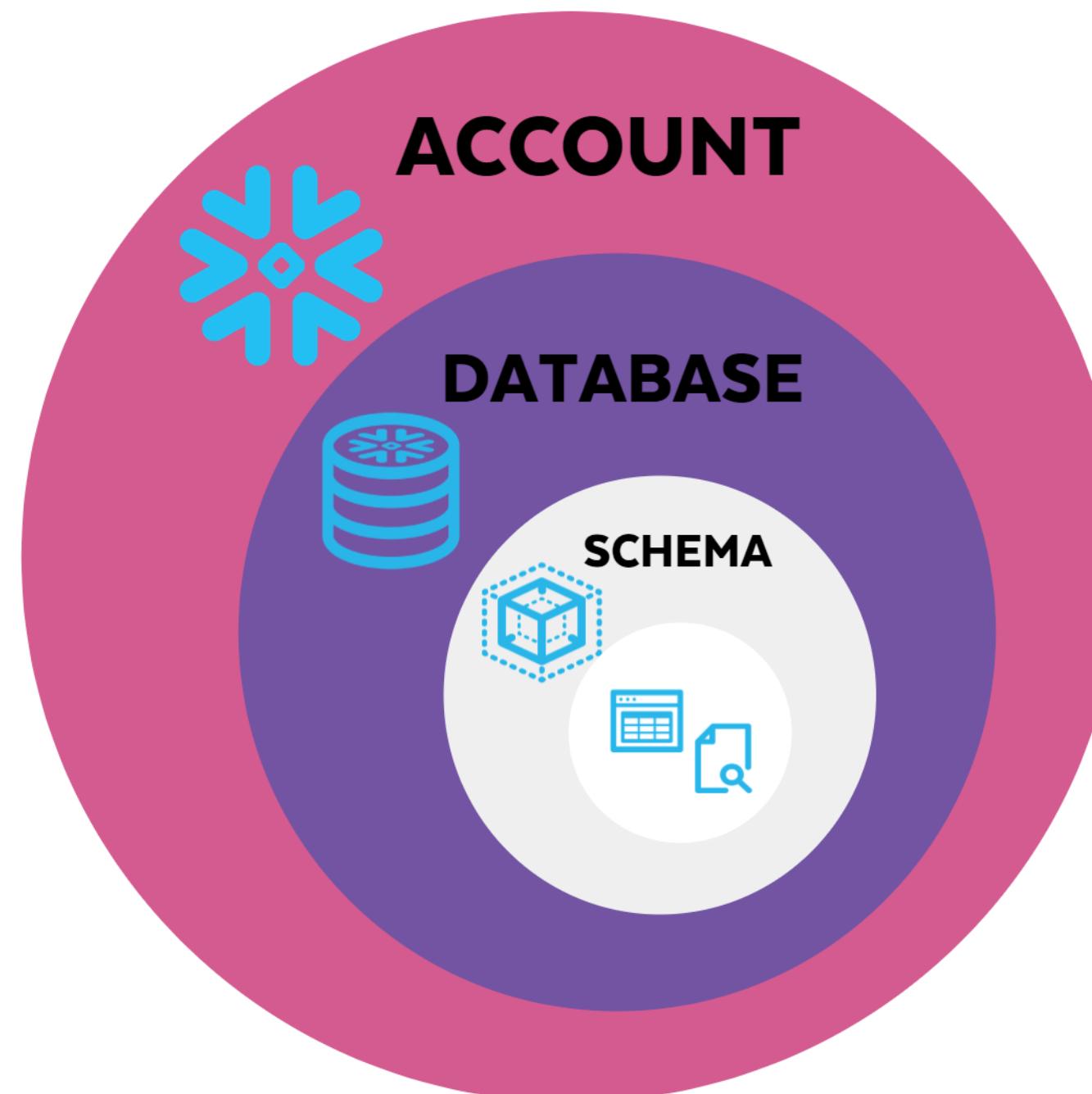
Accounts Contain Databases

- In this example, account NS61174 has three databases.
- The **SNOWFLAKE** database contains information about account usage. It is automatically added by Snowflake to each new account. It is sometimes called the “Account Usage Share” because it is shared by Snowflake so customers can check usage.
- The **SNOWFLAKE_SAMPLE_DATA** is also shared by Snowflake. This database allows for performance testing and tutorials. Customers are not charged for storing this data., because they are NOT storing it. The database is a shared database.

Screenshot of the Snowflake web console showing the 'Databases' page for account ns61174. The browser address bar shows 'ns61174.snowflakecomputing.com/console#/data/databases'. The page includes a navigation bar with 'snowflake' logo, 'Databases', 'Shares', 'Warehouses', 'Worksheets', 'History', and 'Account' buttons. Below the navigation bar is a 'Databases' section with buttons for 'Create...', 'Clone...', 'Drop...', and 'Transfer Ownership'. A table lists three databases: MY_DATABASE, SNOWFLAKE_SAMPLE_DATA, and SNOWFLAKE. The table columns are Database, Origin, Creation Time, and Owner. The SNOWFLAKE database is highlighted with a purple box.

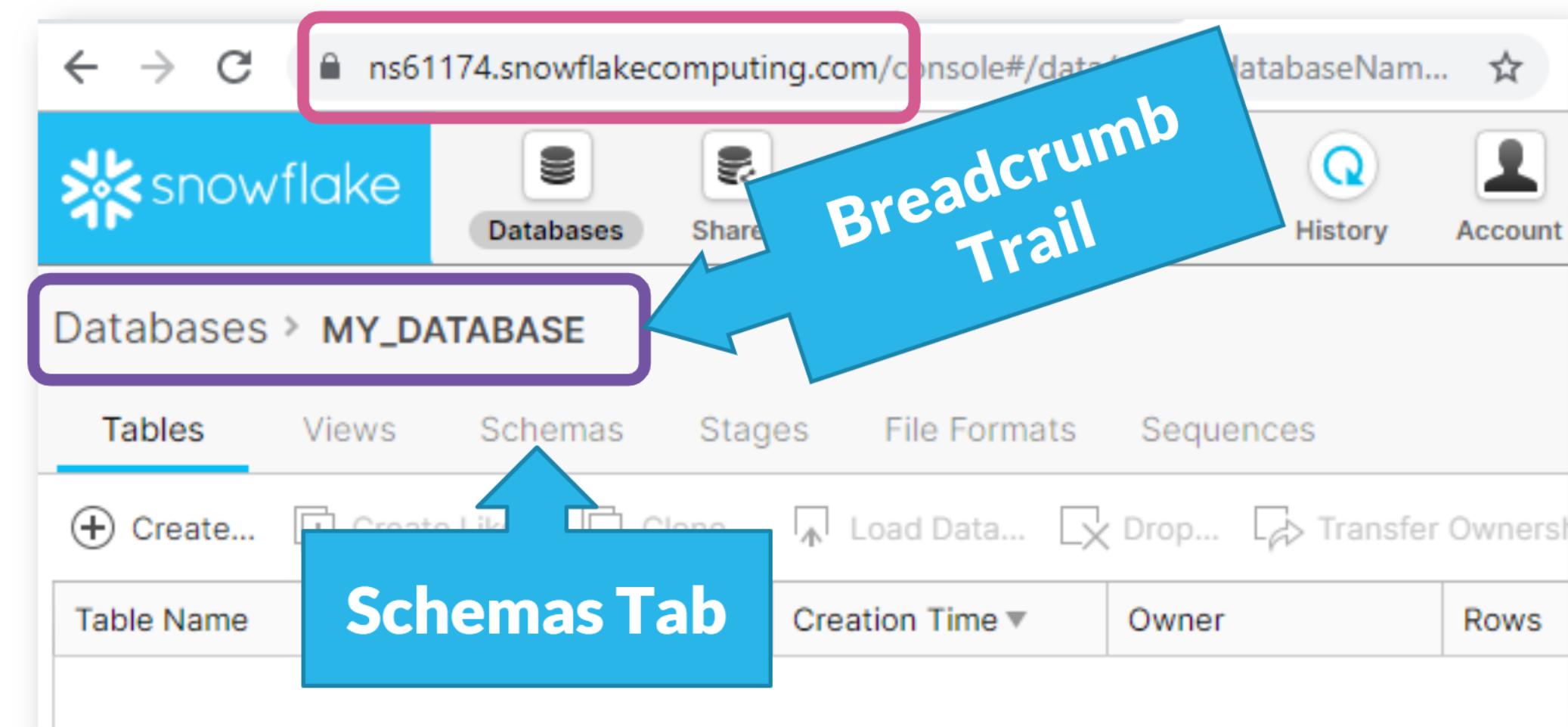
Database	Origin	Creation Time	Owner
MY_DATABASE		7:06:25 AM	ACCOUNTADMIN
SNOWFLAKE_SAMPLE_DATA	SFC_SAMPLES.SA...	6/1/19 10:51:59 PM	ACCOUNTADMIN
SNOWFLAKE	SNOWFLAKE.ACCTO...	6/1/19 10:51:47 PM	

STORAGE CONTAINER HIERARCHY

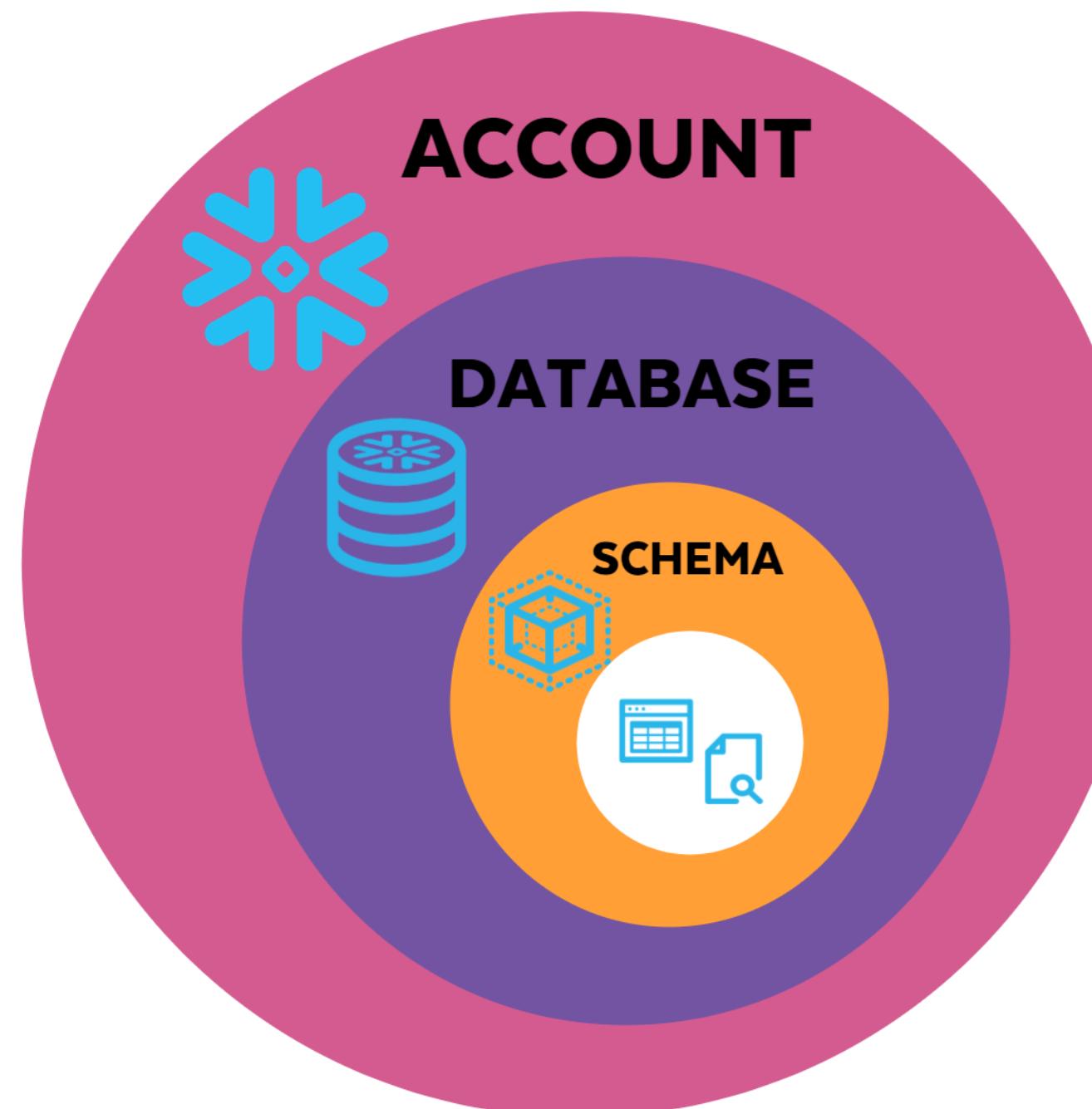


Accounts Contain Databases

When we drill into any database, the name of the database gets added to the breadcrumb trail and a set of tabs appears showing Schemas and other object types.



STORAGE CONTAINER HIERARCHY

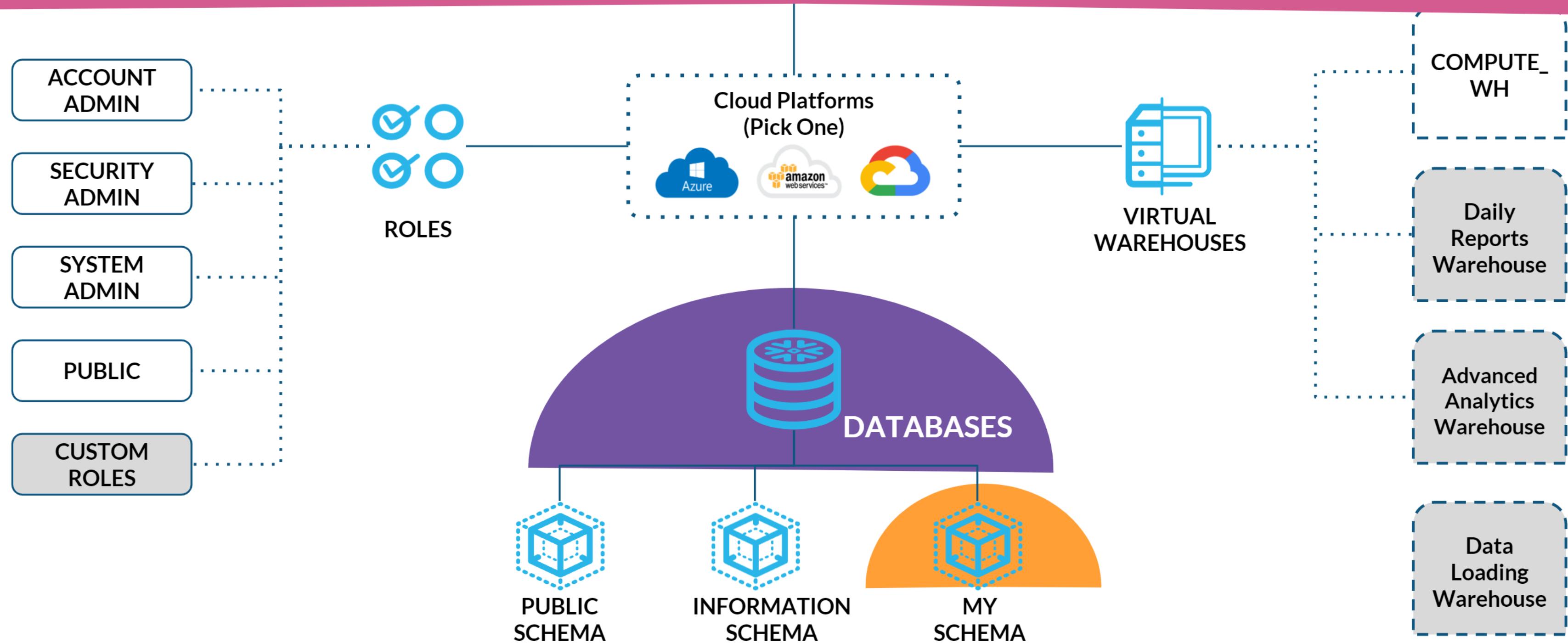


Databases Contain Schemas

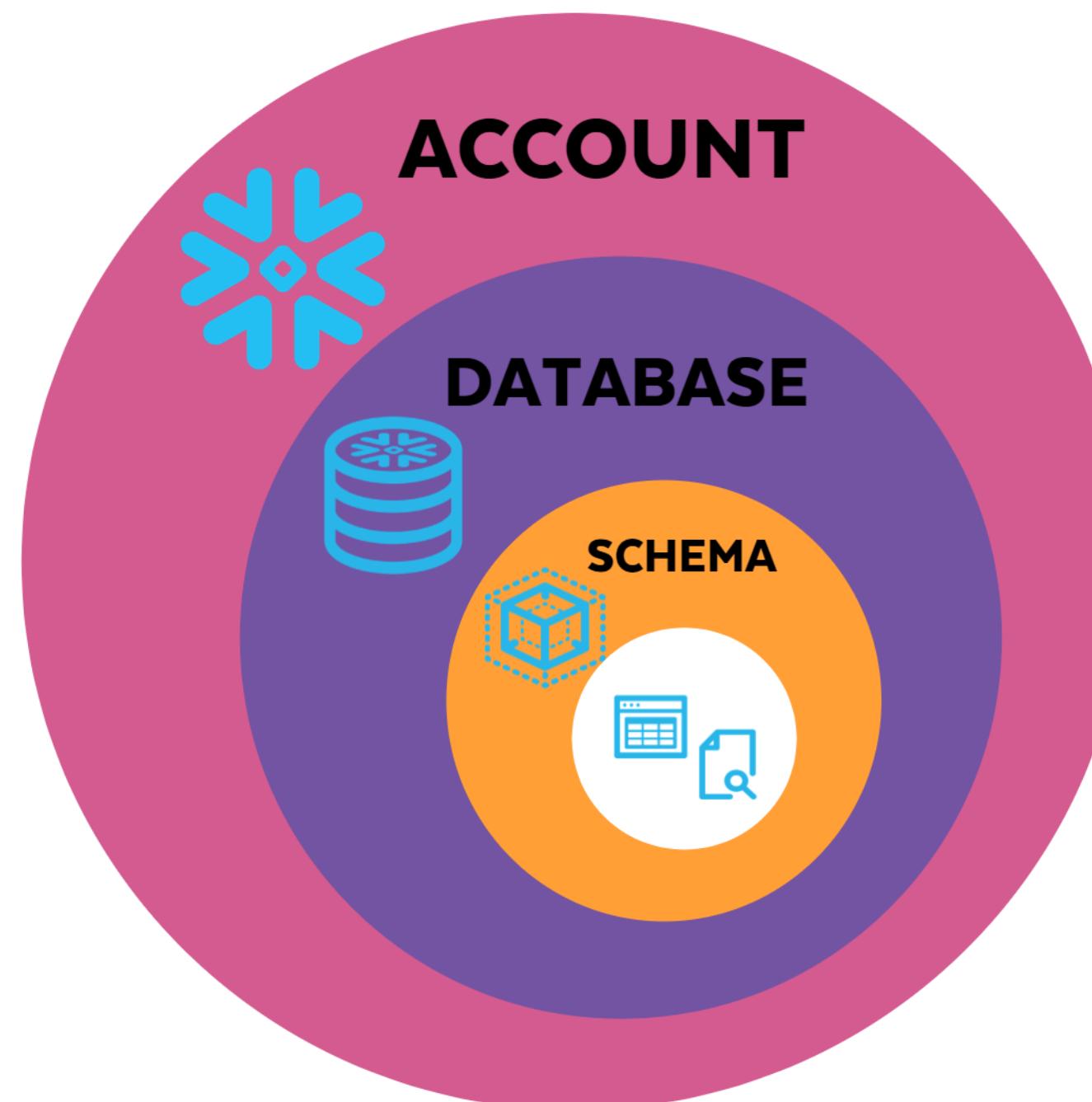
- The set of tabs below the breadcrumb trail makes it look as though Schemas are at the same level as Tables, Views, Stages, File Formats, and Sequence but this is not an accurate way to think of Schemas.
- Schemas are a logical grouping one level above the other objects listed.

Schema	Creation Time	Owner	Managed Access	Comment
INFORMATION_SCHEMA	7:21:55 AM			
MY_SCHEMA	7:06:39 AM	ACCOUNTADMIN		
PUBLIC	7:06:25 AM	ACCOUNTADMIN		

SNOWFLAKE ACCOUNT



STORAGE CONTAINER HIERARCHY



Tables, Views, Stages, File Formats, and Sequences are situated within Schemas

- The tables shown below are in different schemas.
- Snowflake allows you to bypass the schema during the drill-down process, making it seem as if tables and other objects are not within a schema, but they are.

The screenshot shows the Snowflake web interface. At the top, there's a navigation bar with the Snowflake logo, followed by tabs for 'Databases', 'Shares', 'Warehouses', and 'Workshops'. Below the navigation bar, the path 'Databases > MY_DATABASE' is highlighted. Underneath, there are tabs for 'Tables', 'Views', 'Schemas', 'Stages', 'File Formats', and 'Sequences', with 'Tables' being the active tab. Below these tabs are buttons for 'Create...', 'Create Like...', 'Clone...', 'Load Data...', and 'Drop...'. The main area displays a table with three rows of data:

Table Name	Schema	Creation Time
MY_SECOND_TABLE	MY_SCHEMA	9:00:11 AM
MY_FIRST_TABLE	PUBLIC	8:59:55 AM

The 'Schema' column for both rows is highlighted with an orange border.

SCHEMA OBJECTS

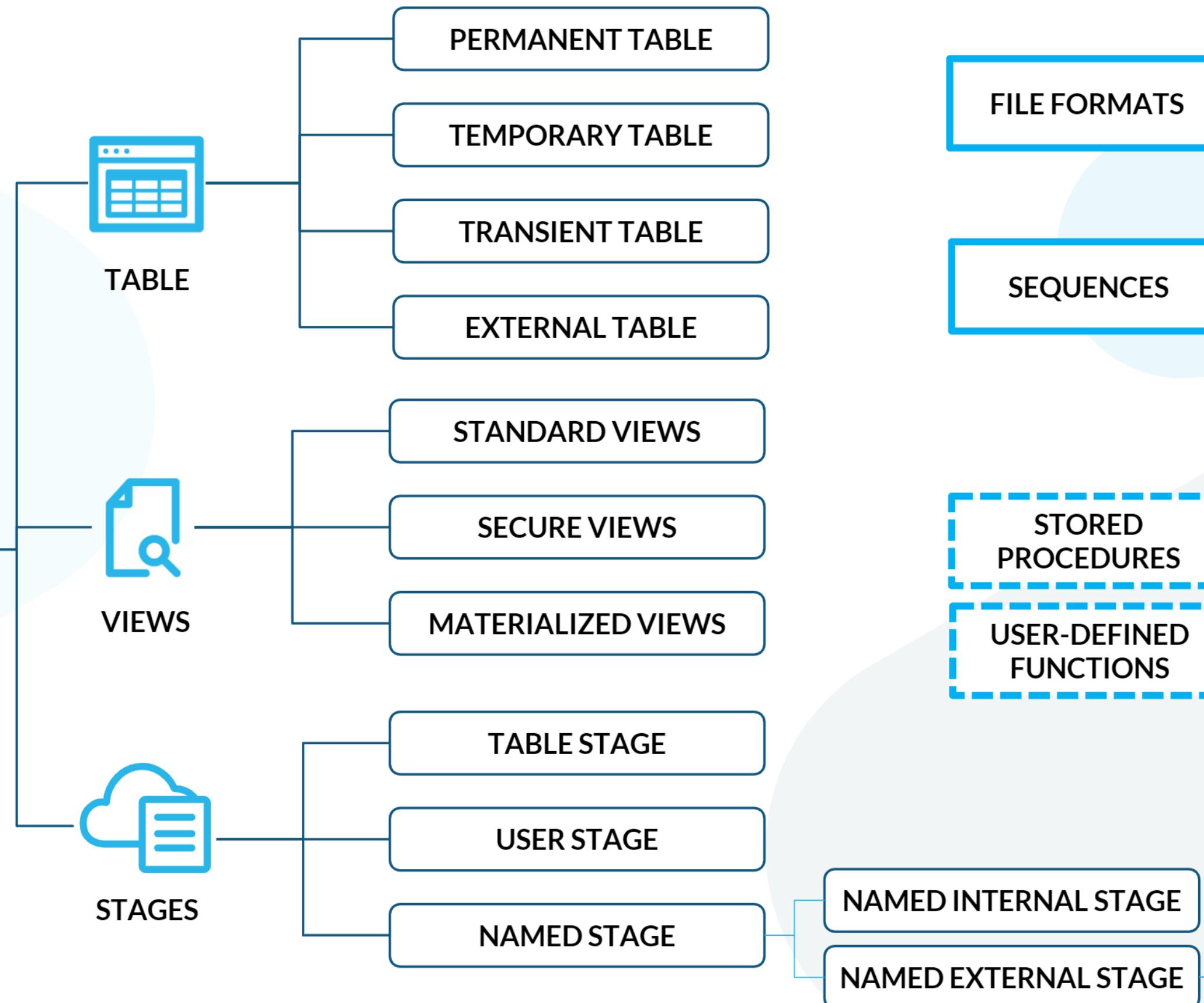
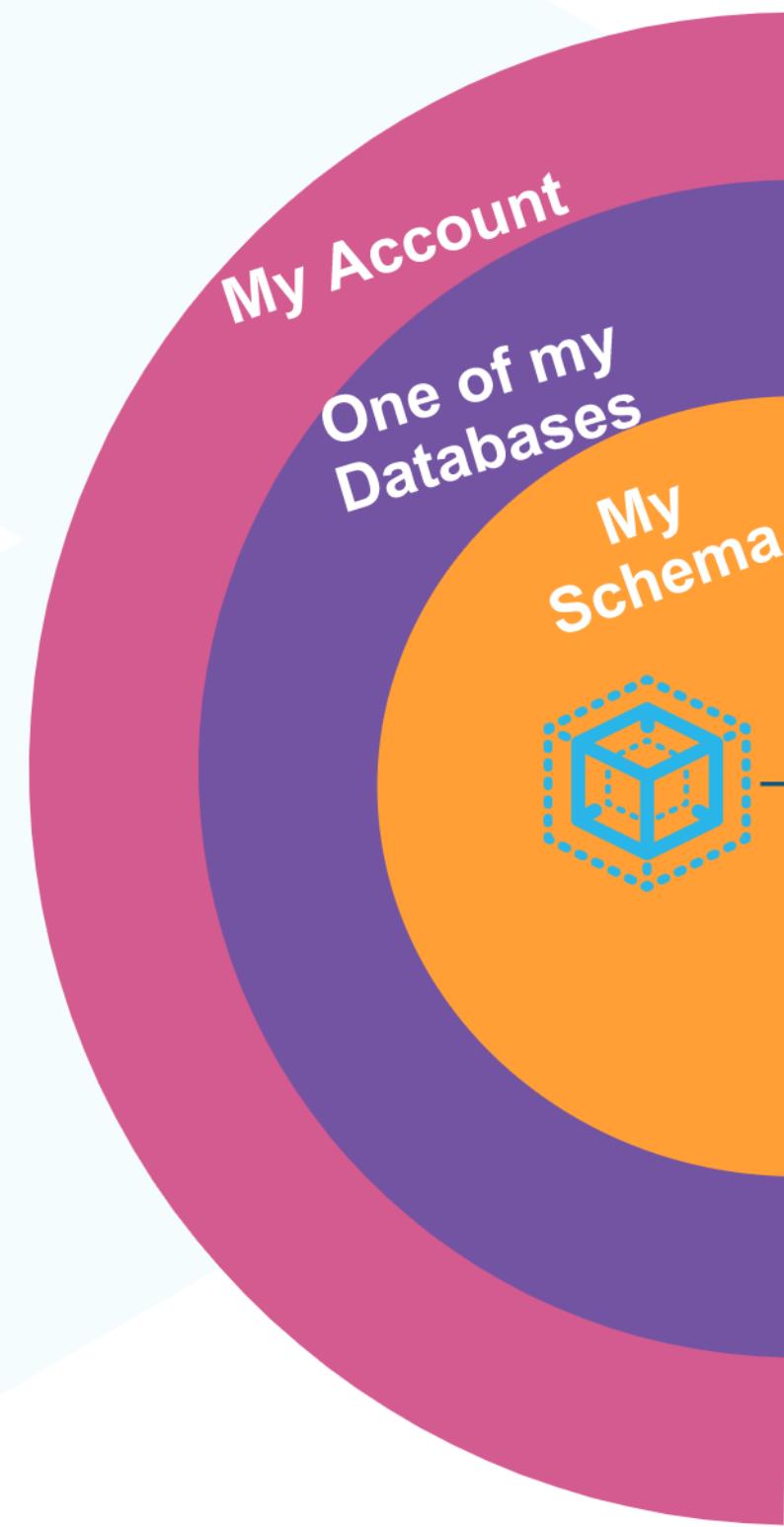
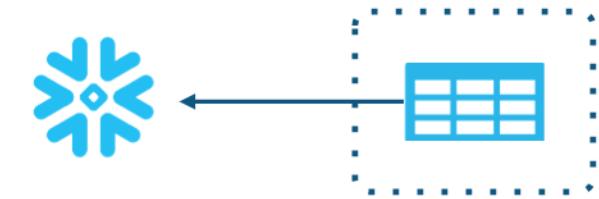


TABLE TYPES



PERMANENT

- Standard, default table type.
- Most commonly used.
- Default Time-Travel
- 7 Day Fail-Safe – non-configurable

TEMPORARY

- Lasts single session
- Used for storing non-permanent, transitory data
- Can be used for moving data from one permanent to another with some transformation within the table
- Never accrue storage costs

TRANSIENT

- Lasts longer than the session
- Recovery snapshot is not retained even for 24 hour minimum provided on Permanent.
- Usually less costly than permanent tables.

EXTERNAL

- Stored in External Stages
- Are read only (no DML SQL)
- DDL SQL works: Selects, Aggregations, Joins
- Can use in a View
- No cloning, time travel or fail-safe
- Great for Semi-structured

Time-Travel



Fail-Safe



VIEW TYPES



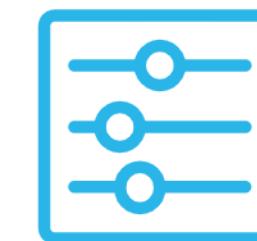
Standard/Regular View

- Default View type
- Most commonly used
- Executes as the executing Role
- Underlying DDL available to any Role with access to the View



Secure View

- The underlying query definition and details are only visible to authorized users
- Executes as the owning role
- Snowflake query optimizer behaves differently when running the SQL contained in a secure view so that it doesn't accidentally reveal things about the underlying data
- May not run as fast as a Standard view, because the optimizer isn't used the same way



Materialized View

- It has “view” in the name, but it behaves more like a table
- Results are stored just as a table would be
- In Snowflake, these are auto-refreshed
- Can provide cost savings but are inflexible and somewhat limited in use

STAGE TYPES

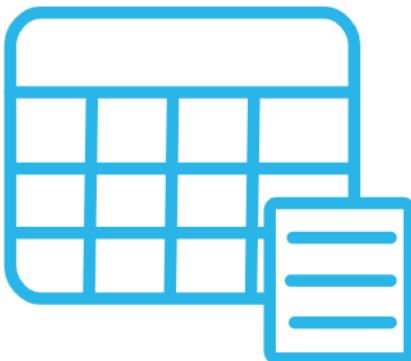
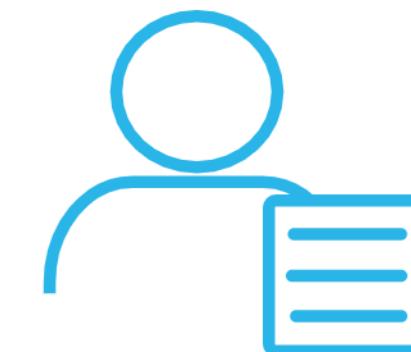


Table Stage @%[TABLE_NAME]

- Automatically defined – No set up needed
- Not viewable as objects on Stages tab of WebUI
- You must specify certain file format-like parameters as part of the command during each COPY INTO event



User Stage @~[LOGIN]



Named Internal Stage @[STAGE_NAME]

- Definable via WebUI
- More flexible than Table and User Stages
- Viewable as objects on Stages tab of Database Area of WebUI
- Have embedded default File Format definitions or you can specify named File Formats for use



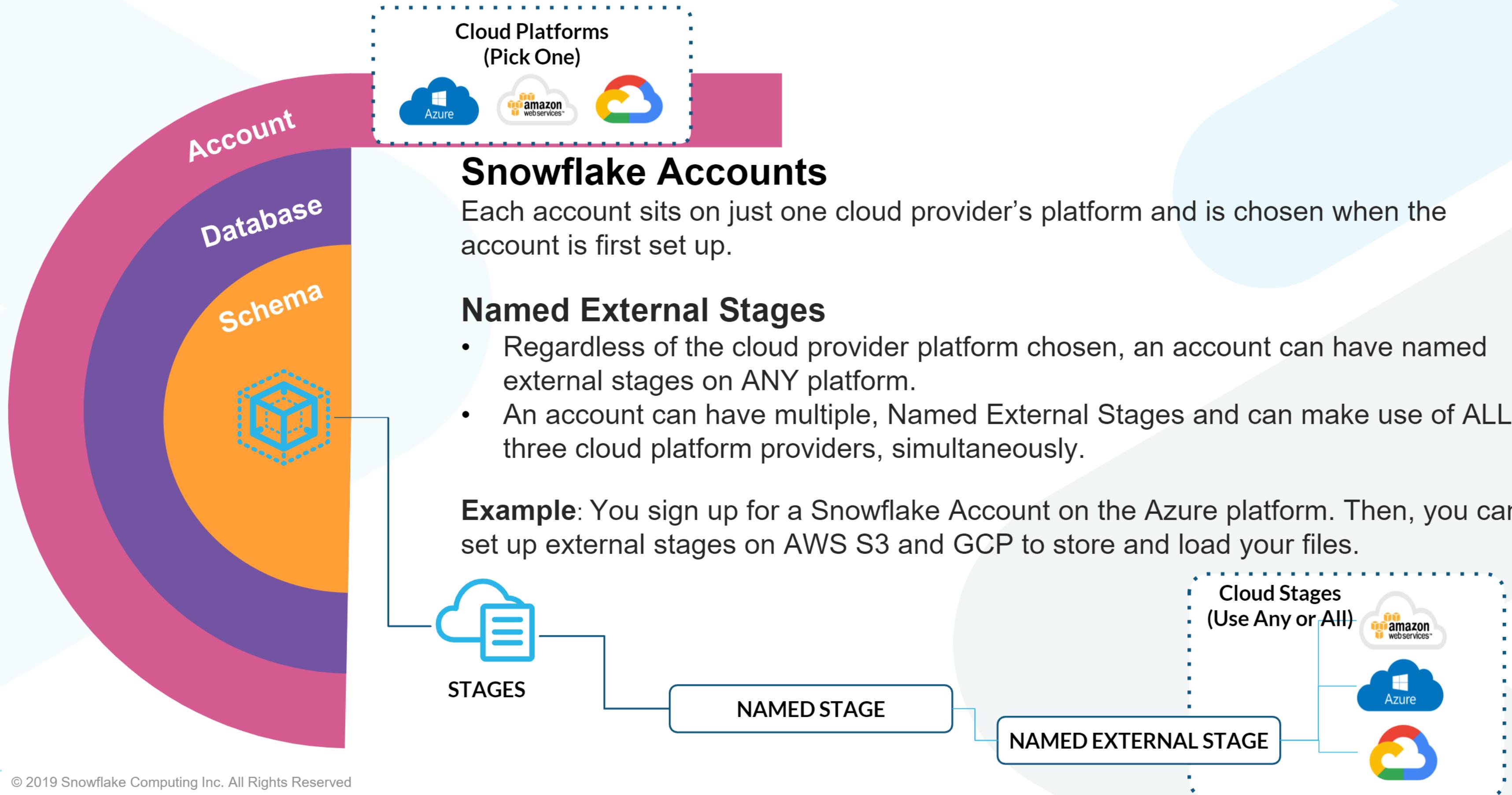
Named External Stage @[STAGE_NAME]

GETTING FILES INTO AND OUT OF STAGES

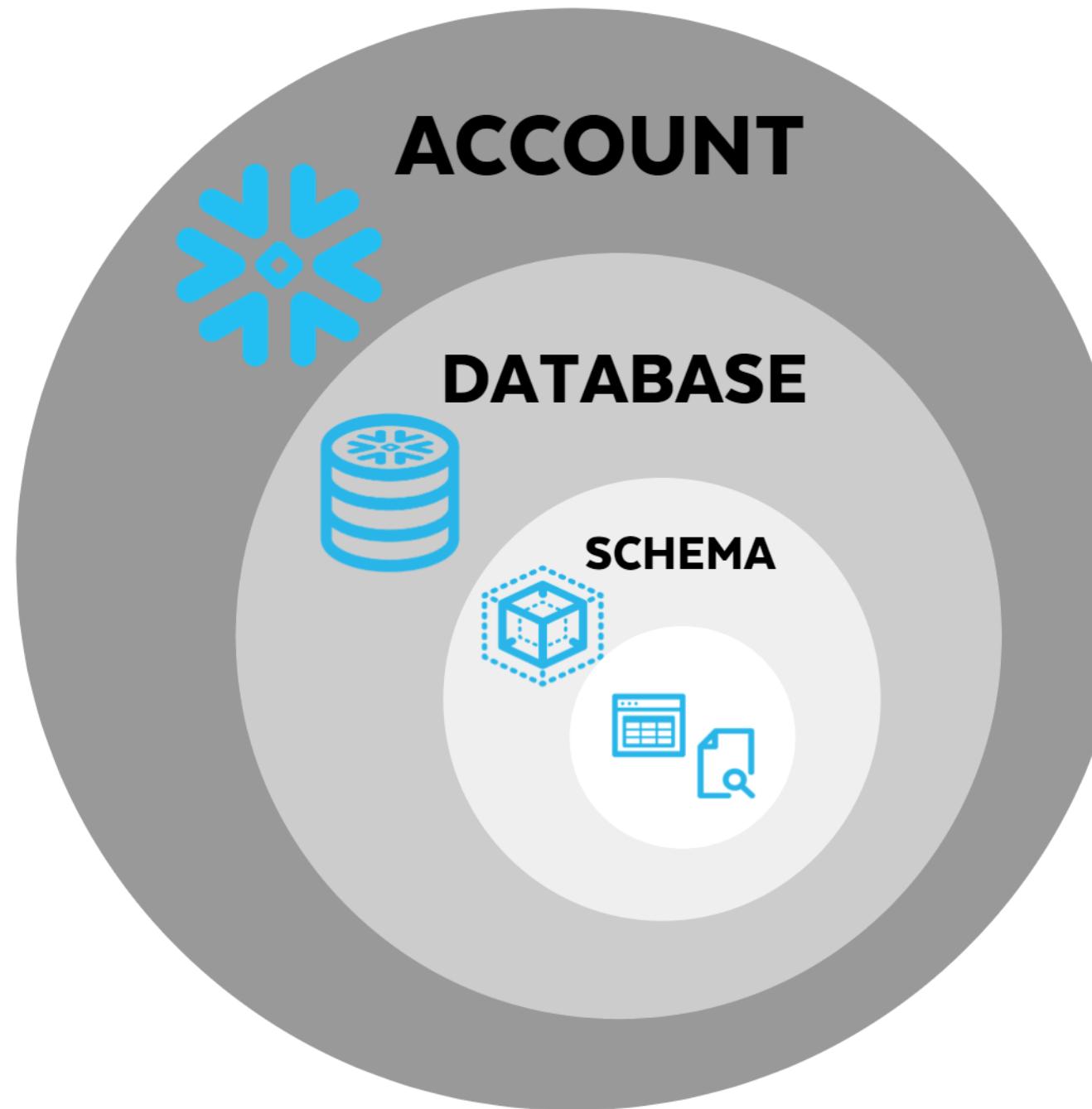
- Use the PUT command to load files into any type of stage*
- Use the COPY INTO command to move files from any stage into a table
- Use the COPY INTO command to unload files from tables and put them into any kind of stage

* The PUT command cannot be run from WebUI Worksheet. The PUT command can be run from SnowSQL, the Snowflake CLI client. The PUT command can be run via other programmatic interfaces as well.

PLATFORM VS. STAGES



STORAGE HIERARCHY



Further Reading:

DATABASES AND SCHEMAS

<https://docs.snowflake.net/manuals/sql-reference/ddl-database.html#database-schema-share-ddl>

TABLES

<https://docs.snowflake.net/manuals/user-guide/tables-temp-transient.html#comparison-of-table-types>

VIEWS

<https://docs.snowflake.net/manuals/user-guide/views-introduction.html>

STAGES

<https://docs.snowflake.net/manuals/user-guide/data-load-local-file-system-create-stage.html#types-of-stages>

