~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Running the following SQL commands, **This does not require a Snowflake virtual warehouse to be active.**

CREATE OR REPLACE DATABASE Production; USE DATABASE Production;

* **CREATE OR REPLACE DATABASE** is a metadata operation. Snowflake handles metadata operations (like creating databases, schemas, and tables) through its control plane, not the compute plane. **USE DATABASE** is just a session context change; it tells Snowflake which database to operate on next.

**When do you need a warehouse:** You need a virtual warehouse **only for operations that require compute**, such as:

* Running SELECT, INSERT, UPDATE, or DELETE statements
* Creating or modifying tables with data operations (e.g., copying data from files)
* Executing stored procedures or UDFs that involve data processing

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

**Warehouse information will be displayed below, I can monitor by warehouse.**

select warehouse\_name, cluster\_number,

count(\*) NBR\_of\_Queries,

sum(total\_elapsed\_time/1000) ELAPSED\_TIME,

SUM(compilation\_time) compilation\_time\_ms,

sum((queued\_provisioning\_time+queued\_repair\_time+queued\_overload\_time)/1000) queued\_time\_sec

from table(information\_schema.query\_history\_by\_warehouse('COMPUTE\_WH'))

GROUP BY warehouse\_name,cluster\_number;

**We can see query text, which db running, query type, etc. Also we see if its long running queries, we will kill that. (TOTAL\_ELAPSED\_TIME), ERROR\_MESSAGE,**

Select \* from table(information\_schema.query\_history\_by\_warehouse(‘compute\_wh’));

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

*Stage the data files  
→Snowflake supports loading data from files that have been staged in either an internal (Snowflake) stage or external (Amazon S3, Google Cloud Storage, or Microsoft Azure) stage.  
In this tutorial, we will upload (stage) the sample data files to an internal table stage. The command used to stage files is PUT.*

**SnowSQL command line:**

> snowsql –version

> snowsql -a typeyoursnowflake identifier link

**It will ask username & password,**

**snowsql commandline will open:**

**now, there are 3 types of internal stages, user stage, table stage, named stage; we can use any stage and store our data;**

**To see files in user stage:**

**We are leveraging user\_stage, copied files into user stage by using PUT command and leveraged COPY INTO to load data into tables.**

> list @~;

To copy data into user stage, use put command

> use warehouse compute\_wh

> use role accountadmin;

> use warehouse compute\_wh

> use ingest\_data;

>create or replace table emp\_basic(first\_name string, last\_name string, start\_date date);

> show tables;

Now, we will load into this table from internal stage;

> put [file://c:\temp\employee\employees0\*.csv](file://c:\temp\employee\employees0*.csv) @~;

>list @~;

Copy data into target table

> copy into emp\_basic from @~ file\_format = (type = csv field\_optionally\_enclosed\_by = ‘ “ ‘)

pattern = ‘.employee0[1-5].csv.gz’

on\_error = ‘skip\_file’;

> alter warehouse compute\_wh resume;

> select \* from emp\_basic;

We can also use table\_stage;

> list @%emp\_basic;

> put [file://c:\temp\employee\employees0\*.csv](file://c:\temp\employee\employees0*.csv) @%emp\_basic;

> copy into emp\_basic from @%emp\_basic file\_format = (type = csv field\_optionally\_enclosed\_by = ‘ “ ‘)

pattern = ‘.employee0[1-5].csv.gz’

on\_error = ‘skip\_file’;

Create NAMED\_STAGE;

> create stage emp\_stage;

> list @emp\_stage;

> put [file://c:\temp\employee\employees0\*.csv](file://c:\temp\employee\employees0*.csv) @emp\_stage;

> copy into emp\_basic from @emp\_stage file\_format = (type = csv field\_optionally\_enclosed\_by = ‘ “ ‘)

pattern = ‘.employee0[1-5].csv.gz’

on\_error = ‘skip\_file’;

Internal Stage – Loading JSON file

---Loading the JSON Data using Internal Stage

> CREATE OR REPLACE STAGE my\_json\_stage file\_format = (type = json);

---Copy the JSON Data

> PUT file://c:\temp\family.json @my\_json\_stage;

> CREATE OR REPLACE TABLE relations\_json\_raw ( json\_data\_raw VARIANT

);

---Load the JSON Data

> COPY INTO relations\_json\_raw from @my\_json\_stage;

---Display the JSON Table

>SELECT

json\_data\_raw: Name,

json data raw:address,

VALUE: Name:: String,

VALUE: Relationship:: String

relations\_json\_raw

lateral flatten (input => json\_data\_raw:family\_detail);

--Load Data into target table

--Now we have analyzed and extracted information. We can load the extracted data into the target table.

> CREATE OR REPLACE TABLE candidate\_family\_detail AS

SELECT json\_data\_raw:Name AS candidate\_name;

VALUE:Name::String

VALUE:Relationship::String

FROM relations\_json\_raw, lateral flatten(input => json\_data\_raw:family\_detail);

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

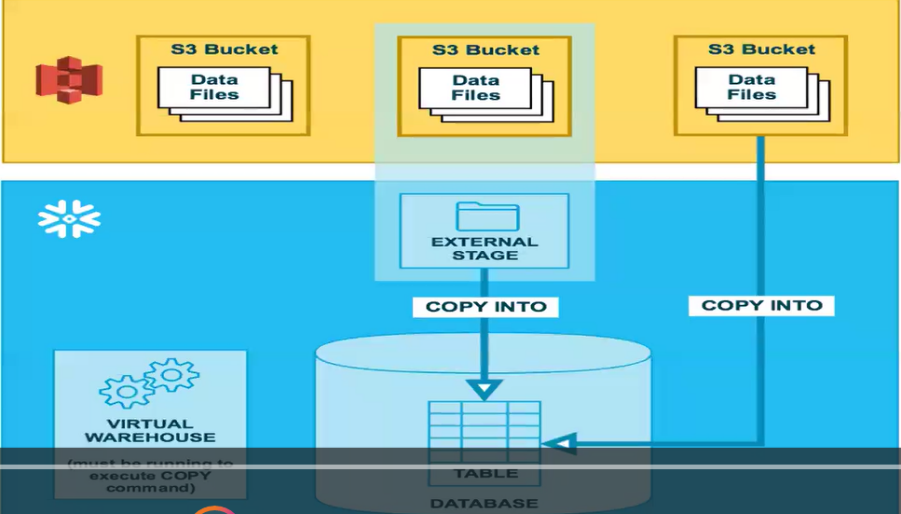
When use named stage, table stage, user stage

Share files with other ppl in team , if data is not much sensitive, If data is sensitive – use user stage.

When we use internal stages, when copy only once, when we don’t have cloud subscriptions.

If you store files in internal stages, storage cost is applicable. So once loaded remove the files, remove @stagename.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~



> We first establish connections between S3 bucket and External Stage.

> Once files are into external stage, as regular process files will load into table.

> In internal stage we need to have files available, then using copy into command load into tables.

We will establish connection between S3 Bucket and Snowflake location. For this we will create external stage.

As soon files are loaded into external stages, as regular process we will use copy into command to load files into respective tables. It will pickup files and load into tables.

There are sequence of steps: