

## ***Chapter: 7 Stages & Storage Integration***

*Seventh Chapter of the Snowflake SnowPro Core Certification Complete Course.*

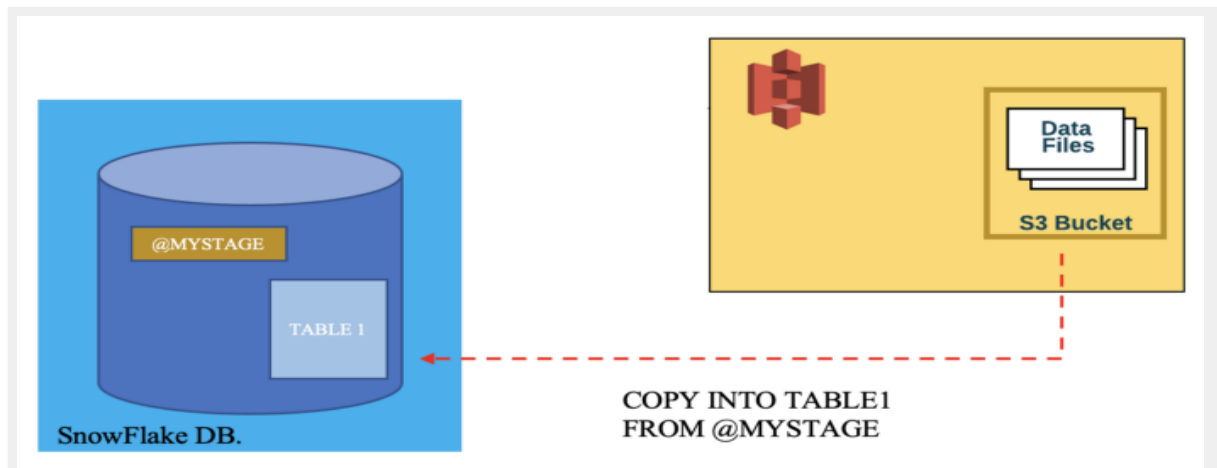
*Stages in Snowflake specify where data files are stored (staged) so that we can load the data in the files into a table. It is where the files are before moving them to Snowflake tables. There are two types of stages, internal and external stages; let's see them.*

- 1. [External Stages](#)*
- 2. [Internal Stages](#)*
- 3. [Stage Metadata](#)*
- 4. [Storage Integration](#)*
- 5. [Typical Exam Questions](#)*

### ***EXTERNAL STAGES***

*External Stages, or Named External Stages, reference data files stored outside of Snowflake. Named Stages are Database Objects. The following Cloud Storage Services are supported (where the files to be copied to Snowflake tables will be located):*

- Amazon S3 Buckets for AWS.*
- Google Cloud Storage Buckets for Google Cloud.*
- Microsoft Azure Containers for Azure.*



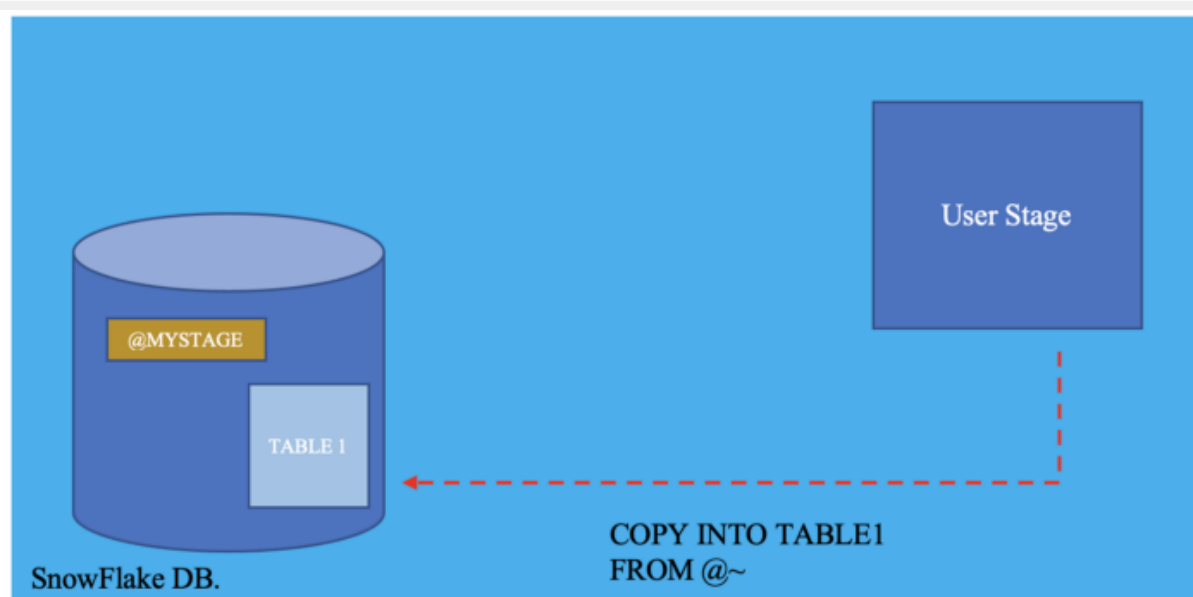
*External Stage in Snowflake.*

*As we can see, the files are in AWS S3 and we'll copy them into a Snowflake table using the COPY command that we will study in the next chapter.*

## *INTERNAL STAGES*

*Stores data files internally within Snowflake. There are the following internal stages:*

- *User (Represented with "@~") → Each user has a Snowflake personal stage allocated to them by default for storing files, and no one can access them except the user it belongs to. You can use it when a single user will only access his files, but he will copy them into several tables. You cannot either see them in the Snowflake interface or drop/alter them; you can only access them using the CLI (SnowSQL). As we can see in the following picture, the data that we copy to the table remains in Snowflake; we don't get it from a Cloud Provider.*



### *User Internal Stage in Snowflake.*

In the first example, we are copying the file "myCSV.csv" from the user's local directory (we will see in the next chapter how to use the PUT command) to his user internal Snowflake stage. Using the second command, we are copying data from the User Internal Stage of the user to a table called "mytable". It's important to realize that we use two commands, one to move the data from local to Snowflake and another one to copy from the internal user stage to the Snowflake table.

```
PUT file:///c:/myData/myCSV.csv @~;
```

```
copy into mytable from @~/staged file_format = (format_name =
'my_csv_format');
```

### *User Internal Stage example in Snowflake.*

- **Table (Represented with "@%")** → Each table has a Snowflake stage allocated to it by default for storing files. This stage is a convenient option if your files must be accessible to multiple users and they only need to be in a single table. You cannot see them in the Snowflake interface; only access them using the CLI. In the following example, you are copying a file from the user's local directory to the table stage of a table called "myTable".

```
PUT file:///c:/myData/myCSV.csv @%myTable;
```

*Table Internal Stage Example in Snowflake.*

- **Named Internal Stage** → Cloud Storage Location managed by Snowflake. Named stages are database objects, both the external and internal ones. If you plan to stage data files that will be loaded only by you, or will be loaded only into a single table, then you may prefer to simply use either your user stage or the stage for the table into which you will be loading data. Named Internal stages can be either permanent or temporary.

## STAGE METADATA

Snowflake automatically generates metadata for files in internal or external stages. This metadata is “stored” in the following virtual columns:

- **METADATA\$FILENAME** → Name of the staged data file that the current row belongs to. It includes the path to the data file in the stage.
- **METADATA\$FILE\_ROW\_NUMBER** → Row number for each record in the container staged data file.

In the following example, we can see how to query the metadata from the files in “@mystage1”:

```
select metadata filename, metadata$file_row_number, t.$1, t.$2 from
@mystage1 (file_format => myformat) t;
```

METADATA\$FILENAME	METADATA\$FILE_ROW_NUMBER	\$1	\$2
data2.csv.gz	1	e	f
data2.csv.gz	2	g	h
data1.csv.gz	1	a	b
data1.csv.gz	2	c	d

*Example of how to select the metadata from files in a Stage in Snowflake.*

## STORAGE INTEGRATION

*A storage integration is a Snowflake object that stores a generated **identity and access management (IAM)** entity for your external cloud storage, along with an optional set of allowed or blocked storage locations (Amazon S3, Google Cloud Storage, or Microsoft Azure). This option will enable users to avoid supplying credentials when creating stages or when loading or unloading data.*

*Storage integrations are **objects at the Account Level, NO Database Level**. They can support several stages. In the following example, we can see how to create it for Amazon S3:*

```
create storage integration <name>
type = external_stage
storage_provider = s3
storage_aws_role_arn = 'arn:aws:iam::001234567890:role/myrole'
enabled = true
storage_allowed_locations = ('s3://mybucket1/path1/',
's3://mybucket2/path2/');
```

*How to create a Storage Integration.*

## TYPICAL EXAM QUESTIONS

1. What are the different stages available in Snowflake

1. User
2. Table
3. Named internal
4. Named external
5. Account
6. Schema

Solution: 1, 2, 3, 4

2. Which are the two metadata columns for staged files?

1. METADATA\$FILENAME
2. METADATA\$FILEFORMAT
3. METADATA\$FILE\_ROW\_NUMBER

Solution: 1, 3

3. Which character identifies a table stage?

1. "@"
2. "%"
3. "/"
4. "@%"

*Solution: 4*

*4. Which character identifies a user stage?*

1. "@~"
2. "~"
3. "@"

*Solution: 1*

*5. When staging uncompressed files in a Snowflake stage, are the files automatically compressed using gzip, unless compression is explicitly disabled?*

1. True
2. False

*Solution: 1.*

*6. Which Snowflake object stores a generated identity and access management (IAM) entity for your external cloud storage, along with an optional set of allowed or blocked storage locations (Amazon S3, Google Cloud Storage, or Microsoft Azure)?*

1. *Storage Schema*
2. *Storage Integration*
3. *User Stage*

*Solution: 2*

*7. Can a single storage integration support multiple external stages?*

1. *True*
2. *False*

*Solution: 1*

*8. You have two types of named stages, one is an external stage, and the other one is an internal stage. Will external stages always require a cloud storage provider?*

1. *True*
2. *False*

*Solution: 1*

*Thanks for Reading!*