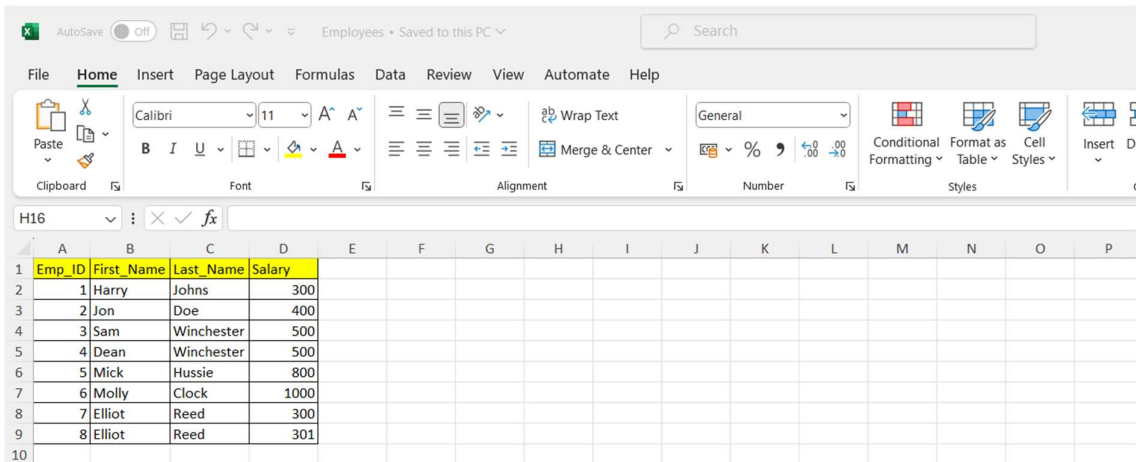


Load Data into Snowflake

1. In this example we are using the below csv file.



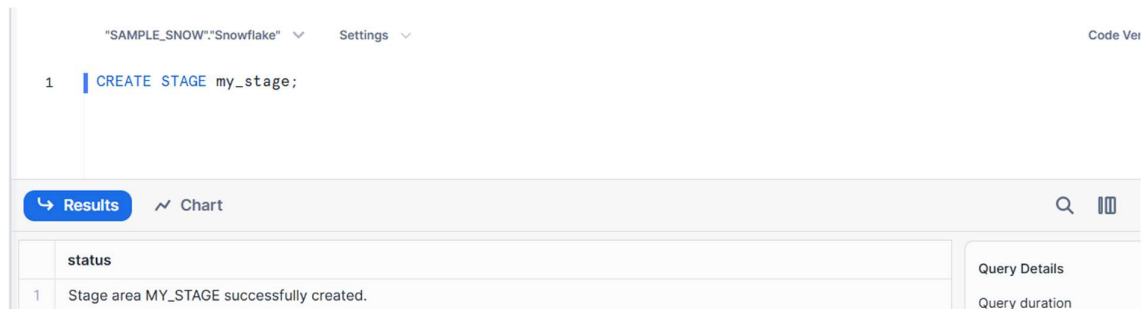
The screenshot shows a Microsoft Excel spreadsheet with the following data:

Emp_ID	First_Name	Last_Name	Salary
1	Harry	Johns	300
2	Jon	Doe	400
3	Sam	Winchester	500
4	Dean	Winchester	500
5	Mick	Hussie	800
6	Molly	Clock	1000
7	Elliot	Reed	300
8	Elliot	Reed	301

CREATE STAGE

2. Create a named location (stage) in Snowflake where you can store data files to be loaded into tables.

Query: CREATE STAGE my_stage;



The screenshot shows the Snowflake query editor with the following query:

```
1 CREATE STAGE my_stage;
```

The results pane shows the status of the query:

status
1 Stage area MY_STAGE successfully created.

Query Details: Query duration

CREATE FILE FORMAT

3. Define the format of your data files, such as CSV, JSON, or Parquet.
4. Use this to specify the file format of your "Employees" file.

Query: CREATE FILE FORMAT my_csv_format
TYPE = 'CSV'
FIELD_OPTIONALLY_ENCLOSED_BY = ''''
SKIP_HEADER = 1;



The screenshot shows the Snowflake query editor with the following query:

```
1 CREATE FILE FORMAT my_csv_format
2 TYPE = 'CSV'
3 FIELD_OPTIONALLY_ENCLOSED_BY = ''''
4 SKIP_HEADER = 1;
```

The results pane shows the status of the query:

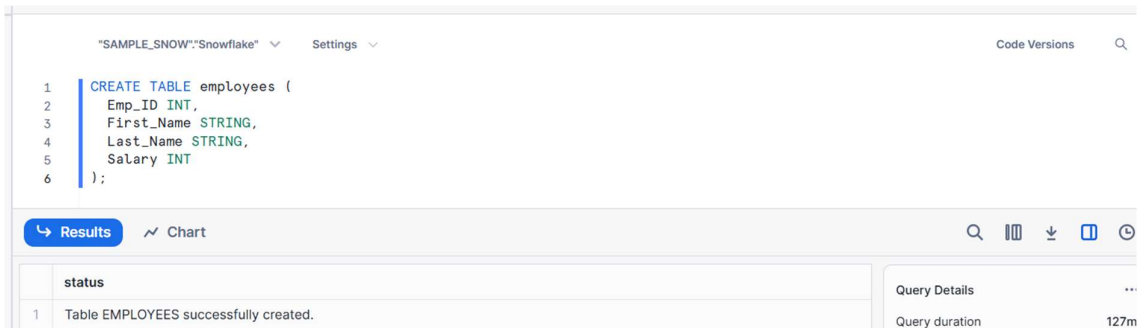
status
1 File format MY_CSV_FORMAT successfully created.

Query Details: Query duration 4s, Rows

CREATE TABLE

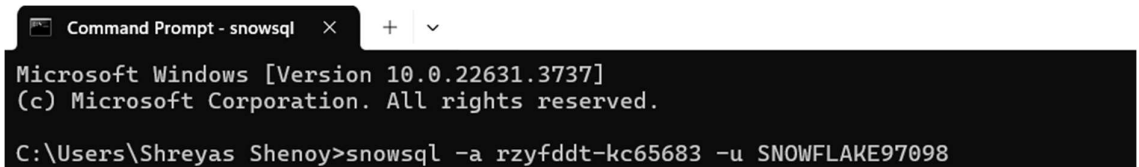
5. Create a table in Snowflake where you can load your data.
6. Use this to create the target table where the data will be loaded.

Query: CREATE TABLE employees (
Emp_ID INT,
First_Name STRING,
Last_Name STRING,
Salary INT);



7. Open the command prompt and give the following command as shown below.

Command syntax: snowsql -a <Account> -u <Username>



8. Give the account Password and click enter.



9. Now to connect with the Warehouse give the below command.

Command: use warehouse COMPUTE_WH;

10. I am using my Warehouse.

```
* SnowSQL * V1.2.32
Type SQL statements or !help
SNOWFLAKE97098#COMPUTE_WH@(no database).(no schema)>use warehouse COMPUTE_WH;
+-----+
| status |
+-----+
| Statement executed successfully. |
+-----+
1 Row(s) produced. Time Elapsed: 0.131s
```

11. Now to connect with the database give the below command.

Command: Use database SAMPLE_SNOW;

```
1 Row(s) produced. Time Elapsed: 0.131s
SNOWFLAKE97098#COMPUTE_WH@(no database).(no schema)>Use database SAMPLE_SNOW;
+-----+
| status |
+-----+
| Statement executed successfully. |
+-----+
1 Row(s) produced. Time Elapsed: 0.121s
```

12. Now to connect with the schema give the below command.

Command: Use schema "Snowflake";

```
1 Row(s) produced. Time Elapsed: 0.121s
SNOWFLAKE97098#COMPUTE_WH@SAMPLE_SNOW.PUBLIC>Use schema "Snowflake";
+-----+
| status |
+-----+
| Statement executed successfully. |
+-----+
1 Row(s) produced. Time Elapsed: 0.165s
```

PUT

13. Upload local files to the Snowflake stage.

14. Use this to upload your "Employees" file from your local system to the Snowflake stage.

Command: PUT file://C:\Files\Source\Employees.csv @my_stage;

```
1 Row(s) produced. Time Elapsed: 0.165s
SNOWFLAKE97098#COMPUTE_WH@SAMPLE_SNOW.Snowflake>PUT file://C:\Files\Source\Employees.csv @my_stage;
+-----+-----+-----+-----+-----+-----+-----+-----+
| source | target | source_size | target_size | source_compression | target_compression | status | message |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Employees.csv | Employees.csv.gz | 195 | 192 | NONE | GZIP | UPLOADED | |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 Row(s) produced. Time Elapsed: 1.434s
```

COPY INTO

15. Loads data from a stage into a Snowflake table.
16. Use this to copy the data from the stage to the Snowflake table.

Command: `COPY INTO employees FROM @my_stage FILE_FORMAT = (FORMAT_NAME = my_csv_format);`

```
1 Row(s) produced. Time Elapsed: 1.434s
SNOWFLAKE97098#COMPUTE_WH$SAMPLE_SNOW.Snowflake>COPY INTO employees FROM @my_stage FILE_FORMAT = (FORMAT_NAME = my_csv_format);
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| file                                | status | rows_parsed | rows_loaded | error_limit | errors_seen | first_error | first_error_line | first_error_character | first_error_column_name |
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| my_stage/Employees.csv.gz          | LOADED | 8           | 8           | 1           | 0           | NULL       | NULL             | NULL                 | NULL                    |
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 Row(s) produced. Time Elapsed: 0.848s
```

17. Now go back to Snowflake and check the employee's data.

"SAMPLE_SNOW"."Snowflake" Settings Code Versions

```
1 Select * from EMPLOYEES
```

	EMP_ID	FIRST_NAME	LAST_NAME	SALARY
1	1	Harry	Johns	300
2	2	Jon	Doe	400
3	3	Sam	Winchester	500
4	4	Dean	Winchester	500
5	5	Mick	Hussie	800
6	6	Molly	Clock	1000
7	7	Elliot	Reed	300
8	8	Elliot	Reed	301

Results Chart

Query Details

Query duration 865ms

Rows 8

Query ID 01b515a6-0001-2587-0...

EMP_ID #

1 8

CREATE PIPE

18. Automates the loading of data from a stage into a Snowflake table.
19. Use this if you want to continuously load data from your stage into a Snowflake table as new files arrive.

Query: `CREATE OR REPLACE PIPE my_pipe`
`AS COPY INTO employees`
`FROM @my_stage`
`FILE_FORMAT = (FORMAT_NAME = my_csv_format);`

"SAMPLE_SNOW"."Snowflake" Settings

```
1 CREATE OR REPLACE PIPE my_pipe
2 AS COPY INTO employees
3 FROM @my_stage
4 FILE_FORMAT = (FORMAT_NAME = my_csv_format);
```

Results Chart

status
1 Pipe MY_PIPE successfully created.

Query Details

Query duration

20. Check Pipe Status, To see the status of the pipe.

Query: SHOW PIPES;

The screenshot shows the Snowflake query interface with the query 'SHOW PIPES' executed. The results table has columns: created_on, name, database_name, schema_name, and definition. One result is shown for a pipe named MY_PIPE in the SAMPLE_SNOW database and Snowflake schema. The definition is 'COPY INTO employees FROM @my_stage'. The query duration is 43ms.

	created_on	name	database_name	schema_name	definition
1	2024-06-17 21:55:38.818 -0700	MY_PIPE	SAMPLE_SNOW	Snowflake	COPY INTO employees FROM @my_stage

Query Details: Query duration 43ms

INSERT OVERWRITE

21. Insert data into a Snowflake table. INSERT OVERWRITE replaces existing data in the table.

22. Use this to insert new rows into the table or overwrite existing data.

23. Create another table with the same employee's fields.

Query: CREATE TABLE sf_employees (
Emp_ID INT,
First_Name STRING,
Last_Name STRING,
Salary INT);

The screenshot shows the Snowflake query interface with the query 'CREATE TABLE sf_employees (...)' executed. The results table has a column 'status' and one result row stating 'Table SF_EMPLOYEES successfully created.' The query duration is 129ms.

	status
1	Table SF_EMPLOYEES successfully created.

Query Details: Query duration 129ms

24. Insert a record in the new table.

Query: INSERT INTO sf_employees (Emp_ID, First_Name, Last_Name, Salary) VALUES (2, 'John', 'Doe', 60000);

The screenshot shows the Snowflake query interface with the query 'INSERT INTO sf_employees (...)' executed. The results table has a column 'number of rows inserted' and one result row showing the value 1. The query duration is 129ms.

	number of rows inserted
1	1

Query Details: Query duration 129ms

25. This statement is inserted into the sf_employees table using the OVERWRITE clause.

Query: INSERT OVERWRITE INTO sf_employees
SELECT * FROM EMPLOYEES
WHERE EMP_ID <= 5;

The screenshot shows a Snowflake query editor with the following SQL query:

```
1 INSERT OVERWRITE INTO sf_employees
2 SELECT * FROM EMPLOYEES
3 WHERE EMP_ID <= 5;
```

The results pane shows a single row with the column "number of rows inserted" and the value 5. The query duration is 436ms.

number of rows inserted
5

26. INSERT used the OVERWRITE option, the old row(s) of sf_employees are gone.

The screenshot shows a Snowflake query editor with the following SQL query:

```
1 SELECT * FROM sf_employees
```

The results pane shows a table with 5 rows and 4 columns: EMP_ID, FIRST_NAME, LAST_NAME, and SALARY. The query duration is 181ms.

EMP_ID	FIRST_NAME	LAST_NAME	SALARY
1	Harry	Johns	300
2	Jon	Doe	400
3	Sam	Winchester	500
4	Dean	Winchester	500
5	Mick	Hussie	800

Unload Data

1. Create a New Stage.

Query: CREATE STAGE my_unload_stage;

The screenshot shows a Snowflake query editor with the following SQL query:

```
1 CREATE STAGE my_unload_stage;
```

The results pane shows a single row with the column "status" and the value "Stage area MY_UNLOAD_STAGE successfully created." The query duration is 47ms.

status
Stage area MY_UNLOAD_STAGE successfully created.

COPY INTO

- Exports the results of a query or a table to one or more files in a stage.
- Use COPY INTO to export data from Snowflake into files stored in a specified stage.
- This can be used to create exports for backups, data sharing, or further processing outside of Snowflake.

Command: COPY INTO @my_unload_stage/employees_export.csv FROM employees
FILE_FORMAT = (FORMAT_NAME = my_csv_format);

```
SNOWFLAKE97098#COMPUTE_WH@SAMPLE_SNOW.Snowflake>COPY INTO @my_unload_stage/employees_export.csv FROM employees FILE_FORMAT = (FORMAT_NAME = my_csv_format);
```

rows_unloaded	input_bytes	output_bytes
8	188	143

```
1 Row(s) produced. Time Elapsed: 0.459s
```

LIST

- Lists files in a stage or a named internal stage.
- Use LIST to view the files present in a Snowflake stage before downloading or performing other operations on them.

Command: LIST @my_unload_stage;

```
1 Row(s) produced. Time Elapsed: 0.459s
SNOWFLAKE97098#COMPUTE_WH@SAMPLE_SNOW.Snowflake>LIST @my_unload_stage;
```

name	size	md5	last_modified
my_unload_stage/employees_export.csv_0_0.csv.gz	144	756b7145ed30632b4bf20c4f6ccf1e21	Tue, 18 Jun 2024 06:01:25 GMT

```
1 Row(s) produced. Time Elapsed: 0.106s
```

GET

- Retrieves a file from a stage into your local file system.
- After data has been staged in Snowflake, use GET to download specific files from the Snowflake stage to your local file system.

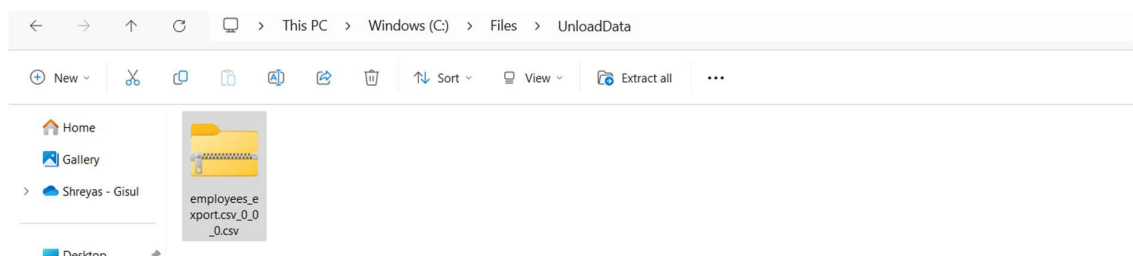
Command: GET @my_unload_stage/employees_export.csv file://C:/Files/UnloadData/;

```
SNOWFLAKE97098#COMPUTE_WH@SAMPLE_SNOW.Snowflake>GET @my_unload_stage/employees_export.csv file://C:/Files/UnloadData/;
```

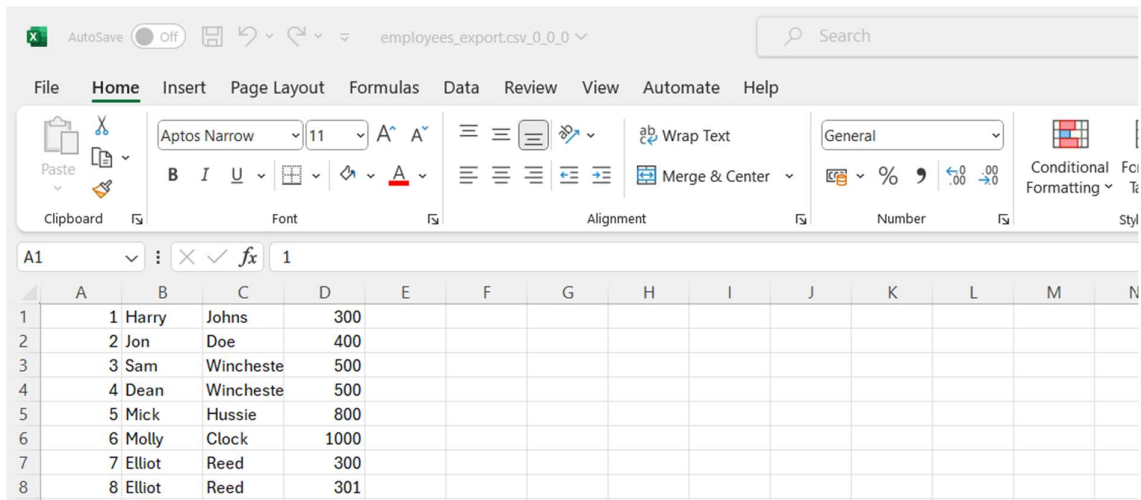
file	size	status	message
employees_export.csv_0_0.csv.gz	143	UPLOADED	

```
1 Row(s) produced. Time Elapsed: 1.403s
SNOWFLAKE97098#COMPUTE_WH@SAMPLE_SNOW.Snowflake>
```

- Go to the File location and check the file.



6. Extract the zip file and open the csv file.



The screenshot shows the Microsoft Excel interface with the 'employees_export.csv_0_0_0' file open. The ribbon is set to 'Home', and the 'Font' group is active. The spreadsheet contains the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		1 Harry	Johns	300										
2		2 Jon	Doe	400										
3		3 Sam	Wincheste	500										
4		4 Dean	Wincheste	500										
5		5 Mick	Hussie	800										
6		6 Molly	Clock	1000										
7		7 Elliot	Reed	300										
8		8 Elliot	Reed	301										