

Table Of Contents

[How to Upload CSV Files to Snowflake: Step-by-Step \(2025\)](#)

Upload CSV to Snowflake —using Snowflake Web Interface

Load CSV from a Snowflake Managed (Internal) stage into a table

Step-by-Step Guide: How to Load CSV to Snowflake —without Snowsight

Conclusion

FAQs

How to Upload CSV Files to Snowflake: Step-by-Step (2025)

[Snowflake](#), renowned for its seamless integration and powerful data handling capabilities, provides a reliable solution for managing large datasets. Importing CSV to Snowflake is a common task for [data professionals, analysts, and engineers](#) seeking to leverage its advanced analytics features.

In this article, we will guide you through the process to upload CSV to Snowflake using various methods. Whether you prefer utilizing the Snowflake Web Interface or performing the task without relying on Snowsight by utilizing SnowSQL and querying it directly, we have you covered.

Upload CSV to Snowflake—using Snowflake Web Interface

One way to upload CSV to Snowflake is by using the Snowflake Web Interface ([Snowsight](#)). Snowsight provides a user-friendly environment that simplifies the entire process of uploading CSV.

Here are the steps to follow:

Loading CSV to Snowflake Using Snowsight

[Snowsight](#), Snowflake's integrated development environment (IDE), offers a convenient way to upload CSV directly from your PC into the Snowflake table. The steps are as follows:

Step 1: Log in to Snowsight.

Step 2: Navigate to the desired database. In the navigation menu, select "Data" and then click on "**Databases**", which will display a list of available databases.

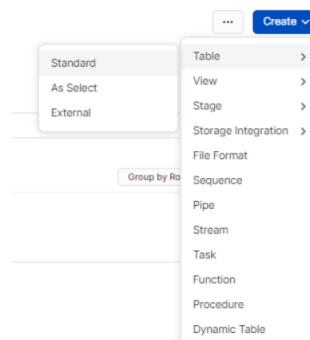


Selecting 'Data' -> 'Databases' to access desired database - upload CSV to Snowflake

Step 3: Select the desired database and schema from the list to set the context for your data loading process.

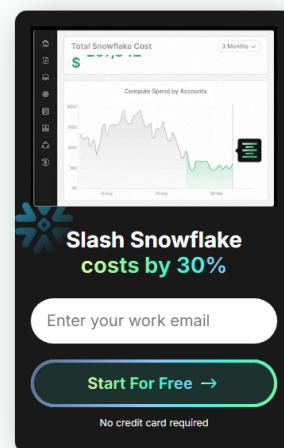
Step 4: Choose the target table or create a new Standard table using the interface.

Step 5: You can now write your own query to create a table. For the purpose of this demo, I will provide an example of creating a table for student records. If you would like to follow along, you can simply copy the following query.



creating a new Standard table using interface - upload CSV to Snowflake

```
create or replace TABLE MY_DATABASE.MY_SCHEMA.CSV_LOAD_DEMO (
    student_id INTEGER,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    date_of_birth DATE,
    email VARCHAR(100),
    address VARCHAR(200)
);
```



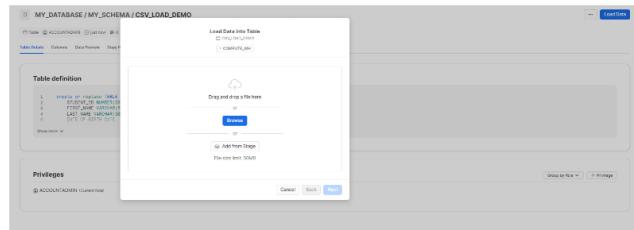
Example query of creating a table for student records

Step 6: Now, select the table where you want to load the data into.



Selecting the table to load the data - upload CSV to snowflake

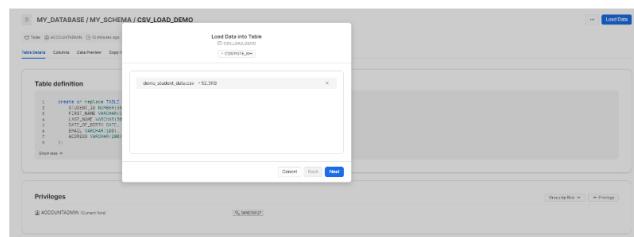
Step 7: Click on "Load Data" to initiate the data loading operation for the selected table.



Click 'Load Data' to initiate data loading process - import CSV to snowflake - upload CSV to snowflake

Step 8: In the "Load Data into Table" dialog, choose the "Upload a file" option to indicate that you want to upload structured or semi-structured data files.

Step 9: Upload your data files by either dragging and dropping them into the designated area or by using the file selection dialog.



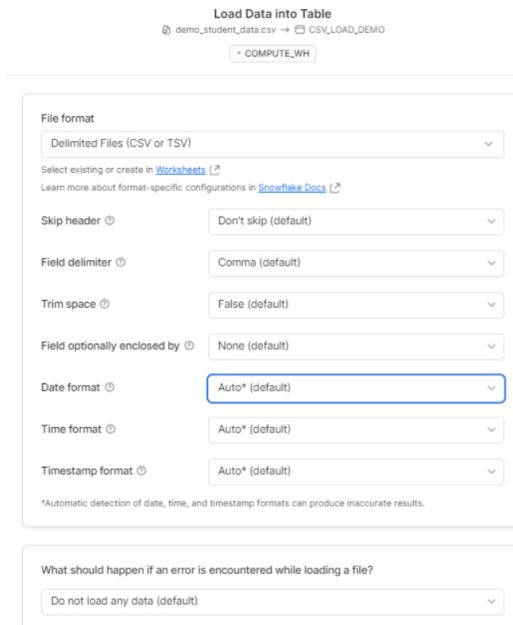
Uploading CSV to Snowflake by dragging or selecting from dialog - snowflake import csv

Step 10: Specify the warehouse (if necessary).

If you haven't set a default warehouse for your user, select a warehouse from the available options to process the data loading operation.

Step 11: Click on the "Next" button to proceed to the next step of the data loading process.

Step 12: Select the appropriate file format for your data. You can either choose a predefined file format from the current database or customize a file type and adjust the relevant settings based on your data.



Load Data into Table
demo_student_data.csv → CSV_LOAD_DEMO
+ COMPUTE_WH

File format
Delimited Files (CSV or TSV)
Select existing or create in Worksheets
Learn more about format-specific configurations in [Snowflake Docs](#)

Skip header
Don't skip (default)

Field delimiter
Comma (default)

Trim space
False (default)

Field optionally enclosed by
None (default)

Date format
Auto* (default)

Time format
Auto* (default)

Timestamp format
Auto* (default)

*Automatic detection of date, time, and timestamp formats can produce inaccurate results.

What should happen if an error is encountered while loading a file?
Do not load any data (default)

Show SQL

Cancel Back Next

Choosing file format: pre-defined or customize settings - load data into snowflake from csv

OR

You can simply type the following SQL query directly into your worksheet

```
COPY INTO "MY_DATABASE"."MY_SCHEMA"."CSV_LOAD_DEMO"
FROM
'@"MY_DATABASE"."MY_SCHEMA".%CSV_LOAD_DEMO"/_snowflake_temp_import_files_/
demo_student_data.csv'
FILE_FORMAT = (
    TYPE=CSV,
    SKIP_HEADER=0,
    FIELD_DELIMITER=',',
    TRIM_SPACE=FALSE,
    FIELD_OPTIONALLY_ENCLOSED_BY=None,
    DATE_FORMAT=AUTO,
    TIME_FORMAT=AUTO,
    TIMESTAMP_FORMAT=AUTO
)
ON_ERROR=ABORT_STATEMENT
PURGE=TRUE
```

Step 13 (optional): Configure error handling.

Specify the desired action if an error occurs during the data loading process. By default, no data is loaded from the file in case of an error.

Step 14: Click on "Next" to move to the next stage of the data loading process.

Step 15: Snowsight will start loading your file and display the number of rows successfully inserted into the target table. You can monitor the progress during this stage.

Step 16: Choose further actions.

After the data loading is complete, you have two options:

- To open a worksheet with SQL syntax for querying your table, select "Query Data."
- To close the dialog and finish the data loading process, select "Done".

Data Partially Loaded into Table
student.csv → CSV_LOAD_DEMO

501 of 502 rows were successfully inserted into the table.

The first error occurs on line 1 at character 1 on column 1 ("STUDENT_ID").

Numeric value 'STUDENT_ID' is not recognized

See more and explore errors in Worksheets [?]

Query Data

Done

Data loaded to the table - snowflake csv

	STUDENT_ID	FIRST_NAME	LAST_NAME	BIRTH_DATE	EMAIL	ADDRESS
1	1	First Name 1	Last Name 1	1999-01-01	email1@example.com	Address 1
2	2	First Name 2	Last Name 2	2000-02-02	email2@example.com	Address 2
3	3	First Name 3	Last Name 3	2001-03-03	email3@example.com	Address 3
4	4	First Name 4	Last Name 4	2001-01-11	email4@example.com	Address 4
5	5	First Name 5	Last Name 5	2001-02-22	email5@example.com	Address 5
6	6	First Name 6	Last Name 6	2000-03-09	email6@example.com	Address 6
7	7	First Name 7	Last Name 7	1999-12-29	email7@example.com	Address 7
8	8	First Name 8	Last Name 8	2000-01-01	email8@example.com	Address 8
9	9	First Name 9	Last Name 9	1999-01-02	email9@example.com	Address 9
10	10	First Name 10	Last Name 10	2000-01-08	email10@example.com	Address 10
11	11	First Name 11	Last Name 11	1999-01-09	email11@example.com	Address 11
12	12	First Name 12	Last Name 12	2000-01-02	email12@example.com	Address 12
13	13	First Name 13	Last Name 13	1999-01-25	email13@example.com	Address 13
14	14	First Name 14	Last Name 14	1999-01-26	email14@example.com	Address 14
15	15	First Name 15	Last Name 15	1999-01-16	email15@example.com	Address 15
16	16	First Name 16	Last Name 16	1997-10-10	email16@example.com	Address 16
17	17	First Name 17	Last Name 17	1997-10-07	email17@example.com	Address 17
18	18	First Name 18	Last Name 18	2001-09-19	email18@example.com	Address 18
19	19	First Name 19	Last Name 19	2000-12-10	email19@example.com	Address 19
20	20	First Name 20	Last Name 20	2000-01-01	email20@example.com	Address 20
21	21	First Name 21	Last Name 21	2000-01-03	email21@example.com	Address 21
22	22	First Name 22	Last Name 22	2001-01-02	email22@example.com	Address 22
23	23	First Name 23	Last Name 23	2000-01-04	email23@example.com	Address 23
24	24	First Name 24	Last Name 24	2001-01-02	email24@example.com	Address 24
25	25	First Name 25	Last Name 25	1999-08-08	email25@example.com	Address 25
26	26	First Name 26	Last Name 26	1999-01-20	email26@example.com	Address 26
27	27	First Name 27	Last Name 27	1999-11-20	email27@example.com	Address 27
28	28	First Name 28	Last Name 28	2000-01-08	email28@example.com	Address 28
29	29	First Name 29	Last Name 29	2000-01-09	email29@example.com	Address 29
30	30	First Name 30	Last Name 30	2000-01-21	email30@example.com	Address 30
31	31	First Name 31	Last Name 31	1999-01-05	email31@example.com	Address 31
32	32	First Name 32	Last Name 32	2000-01-17	email32@example.com	Address 32

Table successfully loaded with data - upload CSV to snowflake

Step 17: If your file encounters any issues during the loading process, such as mismatched columns, an error message will be displayed. To make necessary adjustments, select "Back" and modify your settings accordingly.



Want to take Chaos Genius for a spin?

It takes less than 5 minutes.

[Start Free Trial](#)

Load CSV from a Snowflake Managed (Internal) stage into a table

Loading CSV data from a Snowflake Managed/Internal stage into a table using the Snowflake web interface (Snowsight) has several advantages. These include a user-friendly graphical interface for data loading, a simplified process for transforming and loading data, and the ability to track the progress of the load.

Here are the steps to follow to upload CSV to Snowflake using stage:

Step 1: Log in to Snowsight.

Step 2: Head over to the navigation menu, select "Data" and then click on "Databases", which will display a list of available databases.



Select 'Data' > 'Databases' to view available databases - snowflake import csv

Step 3: Select the desired database and schema from the list to set the context for your data loading process.

Step 4: Choose the 'stage' option and click on "Snowflake Managed" stage.



Selecting 'stage' -> 'Snowflake Managed' option - upload CSV to snowflake

Step 5: Enter a Stage Name. In the "Stage Name" field, provide a descriptive name for your stage. Choose a name that reflects the purpose or nature of the stage.

Create Stage

Creating as: ACCOUNTADMIN

Snowflake

Stage Name: demo_student_stage

Comment (optional): Demo student record

Directory table: Enabling directory table is required to see the files stored on the stage. Warehouse is required. [Learn more](#)

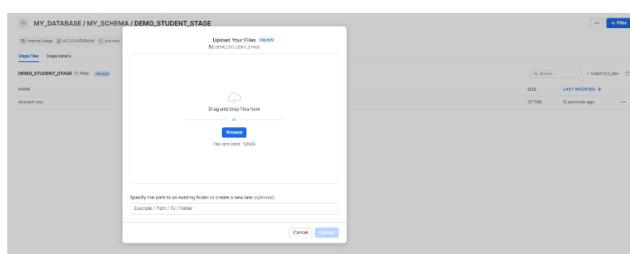
SQL Preview

Create

Entering a descriptive Stage Name - upload CSV to snowflake

Step 6: Once you have filled in the necessary details for your stage, click the "Create" button to create the Snowflake Managed stage.

Step 7: Choose the CSV file to load into Snowflake stage.



Choosing the CSV to load into Snowflake stage - snowflake import csv

Step 8: Select the dropdown menu associated with the selected file, find and select

the "Load into table" option to begin the data loading operation.

The screenshot shows the Snowflake interface with the 'student.csv' file listed in the 'STUDENT_STAGE' stage. The file has a size of 31KB and was last modified 2 minutes ago. A context menu is open next to the file, with the 'Load into table' option highlighted.

Loading CSV to Snowflake table - snowflake import csv

Step 9: A dialog box will appear. In the "Load Data into Table" dialog, choose the desired database, schema, and table where you want to load the selected file.

The 'Load Data into Table' dialog is shown. It displays the selected file 'student.csv'. Below it, the 'Select a database' dropdown is set to 'MY_DATABASE', 'Select a schema' dropdown is set to 'MY_SCHEMA', and the 'Select a table' dropdown is set to 'CSV_LOAD_DEMO'. A note at the bottom states 'Note: Data will be appended to the existing table'. At the bottom right are 'Cancel', 'Back', and a blue 'Next' button.

Loading data into Snowflake table - snowflake import csv

Alternatively,

Step 10: Select the dropdown menu associated with the selected file, find and select the "Copy Path" option.

The screenshot shows the Snowflake interface with the 'student.csv' file listed in the 'STUDENT_STAGE' stage. The file has a size of 31KB and was last modified 2 minutes ago. A context menu is open next to the file, with the 'Copy path' option highlighted.

Copying path of Snowflake stage - snowflake import csv

Step 11: After completing the previous steps, return to the actual table where you want to load the data. Click on "Load Data". A dialog box will appear. Instead of selecting "Browse", choose the option "Add from stage". Paste the link of the path you copied earlier into the designated field.

The 'Load Data into Table' dialog is shown. It features a large input field with a cloud icon labeled 'Drag and drop a file here' and a 'Browse' button below it. To the right, there is a section titled 'Add from Stage' with a 'Add' button. A path 'f0_STUDENT_STAGE/students.csv' is entered into the 'Add from Stage' field. At the bottom right are 'Cancel', 'Back', and a blue 'Next' button.

Adding stage path to load data - snowflake import csv

Step 10: Select the appropriate file format for your data. You can either choose a predefined file format from the current database or customize a file type and adjust the relevant settings based on your data.

The 'Load Data into Table' dialog is shown. It displays the selected file 'demo_student_data.csv'. Below it, the 'File format' dropdown is set to 'Delimited Files (CSV or TSV)'. At the bottom right are 'Cancel', 'Back', and a blue 'Next' button.

Select existing or create in Worksheets [?](#)
Learn more about format-specific configurations in [Snowflake Docs](#) [?](#)

Skip header	Don't skip (default)
Field delimiter	Comma (default)
Trim space	False (default)
Field optionally enclosed by	None (default)
Date format	Auto* (default)
Time format	Auto* (default)
Timestamp format	Auto* (default)

*Automatic detection of date, time, and timestamp formats can produce inaccurate results.

What should happen if an error is encountered while loading a file?

Do not load any data (default)

Show SQL

[Cancel](#) [Back](#) [Next](#)

Choosing file format: pre-defined or customize settings - snowflake import csv

Step 14: After the data loading is complete, you have two options:

- To open a worksheet with SQL syntax for querying your table, select "Query Data"
- To close the dialog and finish the data loading process, select "Done".

MY_DATABASE / MY_SCHEMA / CSV_LOAD_DEMO

Table Details | Columns | Data Preview | Edit History

COMPUTED_IN_DATABASE

STUDENT_ID	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	EMAIL	ADDRESS
1	First Name 1	Last Name 1	2002-01-01	email1@sample.com	Address 1
2	First Name 2	Last Name 2	2002-01-02	email2@sample.com	Address 2
3	First Name 3	Last Name 3	2002-01-03	email3@sample.com	Address 3
4	First Name 4	Last Name 4	2002-01-04	email4@sample.com	Address 4
5	First Name 5	Last Name 5	2002-01-05	email5@sample.com	Address 5
6	First Name 6	Last Name 6	2002-01-06	email6@sample.com	Address 6
7	First Name 7	Last Name 7	2002-01-07	email7@sample.com	Address 7
8	First Name 8	Last Name 8	2002-01-08	email8@sample.com	Address 8
9	First Name 9	Last Name 9	2002-01-09	email9@sample.com	Address 9
10	First Name 10	Last Name 10	2002-01-10	email10@sample.com	Address 10
11	First Name 11	Last Name 11	2002-01-11	email11@sample.com	Address 11
12	First Name 12	Last Name 12	2002-01-12	email12@sample.com	Address 12
13	First Name 13	Last Name 13	2002-01-13	email13@sample.com	Address 13
14	First Name 14	Last Name 14	2002-01-14	email14@sample.com	Address 14
15	First Name 15	Last Name 15	2002-01-15	email15@sample.com	Address 15
16	First Name 16	Last Name 16	2002-01-16	email16@sample.com	Address 16
17	First Name 17	Last Name 17	2002-01-17	email17@sample.com	Address 17
18	First Name 18	Last Name 18	2002-01-18	email18@sample.com	Address 18
19	First Name 19	Last Name 19	2002-01-19	email19@sample.com	Address 19
20	First Name 20	Last Name 20	2002-01-20	email20@sample.com	Address 20
21	First Name 21	Last Name 21	2002-01-21	email21@sample.com	Address 21
22	First Name 22	Last Name 22	2002-01-22	email22@sample.com	Address 22
23	First Name 23	Last Name 23	2002-01-23	email23@sample.com	Address 23
24	First Name 24	Last Name 24	2002-01-24	email24@sample.com	Address 24
25	First Name 25	Last Name 25	2002-01-25	email25@sample.com	Address 25
26	First Name 26	Last Name 26	2002-01-26	email26@sample.com	Address 26
27	First Name 27	Last Name 27	2002-01-27	email27@sample.com	Address 27
28	First Name 28	Last Name 28	2002-01-28	email28@sample.com	Address 28
29	First Name 29	Last Name 29	2002-01-29	email29@sample.com	Address 29
30	First Name 30	Last Name 30	2002-01-30	email30@sample.com	Address 30
31	First Name 31	Last Name 31	2002-01-31	email31@sample.com	Address 31
32	First Name 32	Last Name 32	2002-02-01	email32@sample.com	Address 32
33	First Name 33	Last Name 33	2002-02-02	email33@sample.com	Address 33
34	First Name 34	Last Name 34	2002-02-03	email34@sample.com	Address 34
35	First Name 35	Last Name 35	2002-02-04	email35@sample.com	Address 35
36	First Name 36	Last Name 36	2002-02-05	email36@sample.com	Address 36
37	First Name 37	Last Name 37	2002-02-06	email37@sample.com	Address 37
38	First Name 38	Last Name 38	2002-02-07	email38@sample.com	Address 38
39	First Name 39	Last Name 39	2002-02-08	email39@sample.com	Address 39
40	First Name 40	Last Name 40	2002-02-09	email40@sample.com	Address 40

Table successfully loaded with data - snowflake import csv

Step-by-Step Guide: How to Load CSV to Snowflake—without Snowsight

System Requirements + Setup

Step 1: Install SnowSQL. Head over to the Snowflake website and [download SnowSQL](#) and then follow the installation instructions specific to your operating system.

Step 2: [Configure SnowSQL](#).

Step 3: Now, select the target database and schema.

```
USE MY_DATABASE.MY_SCHEMA
```

```
pramitdemo#COMPUTE_WH@MY_DATABASE.PUBLIC>use MY_DATABASE.MY_SCHEMA;
+-----+
| status |
+-----+
| Statement executed successfully. |
+-----+
1 Row(s) produced. Time Elapsed: 0.118s
pramitdemo#COMPUTE_WH@MY_DATABASE.MY_SCHEMA>CREATE OR REPLACE stage student_demo_record_stage;
+-----+
| status |
+-----+
| Stage area STUDENT_DEMO_RECORD_STAGE successfully created. |
+-----+
1 Row(s) produced. Time Elapsed: 3.675s
pramitdemo#COMPUTE_WH@MY_DATABASE.MY_SCHEMA>
```

Selecting target database and schema - upload CSV to snowflake

Step 4: Create a FILE FORMAT. Run the following command in SnowSQL to create a file format:

```
CREATE OR REPLACE FILE FORMAT student_csv_format
TYPE = CSV
FIELD_DELIMITER = ','
```

```
pramitdemo@COMPUTE_WH@MY_DATABASE.MY_SCHEMA>CREATE OR REPLACE FILE FORMAT student_csv_format type = 'csv' Field_delimiter = ',';
| status
| File Format STUDENT_CSV_FORMAT successfully created
| 1 Row(s) produced. Time Elapsed: 0.193s
pramitdemo@COMPUTE_WH@MY_DATABASE.MY_SCHEMA>
```

creating CSV file format 'student_csv_format' - upload CSV to snowflake

Creating the Target Table

Step 5: Using the CREATE TABLE create a table in Snowflake that matches the structure of your CSV file.

Run the following command in SnowSQL to create the table:

```
CREATE OR REPLACE TABLE STUDENT_CSV_LOAD_DEMO2 (
    student_id INTEGER,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    date_of_birth DATE,
    email VARCHAR(100),
    address VARCHAR(200)
);
```

```
pramitdemo@COMPUTE_WH@MY_DATABASE.MY_SCHEMA>CREATE OR REPLACE TABLE STUDENT_CSV_LOAD_DEMO2 (
    student_id INTEGER,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    date_of_birth DATE,
    email VARCHAR(100),
    address VARCHAR(200)
);
| status
| Table STUDENT_CSV_LOAD_DEMO2 successfully created.
| 1 Row(s) produced. Time Elapsed: 1.478s
pramitdemo@COMPUTE_WH@MY_DATABASE.MY_SCHEMA>
```

Creating table matching CSV file structure - upload CSV to snowflake

Table creation in Snowsight console

Loading CSV Data

Step 6: Uploading CSV to Snowflake stage.

- Use the PUT command to upload your CSV file to the stage.
- Run the following command in SnowSQL:

```
put file:///C:/Users/default.LAPTOP-A2VTF9HN/Desktop/cg/csvtest/student.csv
@student_demo_record_stage;
```

```
pramitdemo@COMPUTE_WH@MY_DATABASE.MY_SCHEMA> put file:///C:/Users/default.LAPTOP-A2VTF9HN/Desktop/cg/csvtest/student.csv @student_demo_record_stage;
| source | target | source_size | target_size | source_compression | target_compression | status | message |
| student.csv | student.csv.gz | 38597 | 8348 | NONE | GZIP | UPLOADED | 
| Row(s) produced. Time Elapsed: 3.945s
pramitdemo@COMPUTE_WH@MY_DATABASE.MY_SCHEMA>
```

Uploading CSV to Snowflake stage using PUT command - snowflake import csv

Checking the Stage Data

Step 7: Now, verifying if the Snowflake stage is populated with the CSV data to do so Run the following command to check if the Snowflake stage is populated with the data from the file:

```
SELECT
    col.$1,
    col.$2,
    col.$3,
    col.$4,
    col.$5,
    col.$6
FROM @student_demo_record_stage (file_format => student_csv_format) col;
```

```
pramitdemo@COMPUTE_WH@MY_DATABASE.MY_SCHEMA>select
    col.$1,
    col.$2,
    col.$3,
    col.$4,
    col.$5,
    col.$6
    from @student_demo_record_stage (file_format => student_csv_format) col;
```

STUDENT_ID	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	EMAIL	ADDRESS
1	First Name 1	Last Name 1	1997-09-01	email1@example.com	Address 1
2	First Name 2	Last Name 2	2002-05-09	email2@example.com	Address 2
3	First Name 3	Last Name 3	2001-10-13	email3@example.com	Address 3
4	First Name 4	Last Name 4	2001-06-11	email4@example.com	Address 4
5	First Name 5	Last Name 5	2002-11-16	email5@example.com	Address 5
6	First Name 6	Last Name 6	2005-05-09	email6@example.com	Address 6
7	First Name 7	Last Name 7	1997-12-09	email7@example.com	Address 7
8	First Name 8	Last Name 8	2001-11-06	email8@example.com	Address 8
9	First Name 9	Last Name 9	1998-08-08	email9@example.com	Address 9
10	First Name 10	Last Name 10	2000-09-08	email10@example.com	Address 10
11	First Name 11	Last Name 11	2001-01-07	email11@example.com	Address 11
12	First Name 12	Last Name 12	2004-10-02	email12@example.com	Address 12
13	First Name 13	Last Name 13	1999-04-25	email13@example.com	Address 13
14	First Name 14	Last Name 14	1998-07-17	email14@example.com	Address 14
15	First Name 15	Last Name 15	1998-05-18	email15@example.com	Address 15
16	First Name 16	Last Name 16	1997-10-18	email16@example.com	Address 16
17	First Name 17	Last Name 17	1997-08-18	email17@example.com	Address 17
18	First Name 18	Last Name 18	2001-05-18	email18@example.com	Address 18
19	First Name 19	Last Name 19	2002-12-10	email19@example.com	Address 19
20	First Name 20	Last Name 20	2003-11-23	email20@example.com	Address 20
21	First Name 21	Last Name 21	2004-01-01	email21@example.com	Address 21
22	First Name 22	Last Name 22	2001-06-02	email22@example.com	Address 22
23	First Name 23	Last Name 23	2000-11-03	email23@example.com	Address 23
24	First Name 24	Last Name 24	2001-02-02	email24@example.com	Address 24
25	First Name 25	Last Name 25	1999-08-06	email25@example.com	Address 25
26	First Name 26	Last Name 26	1999-05-09	email26@example.com	Address 26
27	First Name 27	Last Name 27	1998-12-15	email27@example.com	Address 27
28	First Name 28	Last Name 28	2003-03-09	email28@example.com	Address 28
29	First Name 29	Last Name 29	2000-09-30	email29@example.com	Address 29
30	First Name 30	Last Name 30	2002-02-29	email30@example.com	Address 30
31	First Name 31	Last Name 31	1998-03-05	email31@example.com	Address 31
32	First Name 32	Last Name 32	2002-09-17	email32@example.com	Address 32
33	First Name 33	Last Name 33	2004-08-01	email33@example.com	Address 33
34	First Name 34	Last Name 34	2002-04-24	email34@example.com	Address 34
35	First Name 35	Last Name 35	2004-02-23	email35@example.com	Address 35
36	First Name 36	Last Name 36	2003-05-07	email36@example.com	Address 36
37	First Name 37	Last Name 37	2004-03-20	email37@example.com	Address 37
38	First Name 38	Last Name 38	1999-01-01	email38@example.com	Address 38
39	First Name 39	Last Name 39	2000-02-08	email39@example.com	Address 39
40	First Name 40	Last Name 40	2002-12-10	email40@example.com	Address 40
41	First Name 41	Last Name 41	2004-01-01	email41@example.com	Address 41
42	First Name 42	Last Name 42	1999-05-05	email42@example.com	Address 42
43	First Name 43	Last Name 43	2001-09-29	email43@example.com	Address 43
44	First Name 44	Last Name 44	2001-09-06	email44@example.com	Address 44
45	First Name 45	Last Name 45	2001-10-05	email45@example.com	Address 45
46	First Name 46	Last Name 46	1999-06-06	email46@example.com	Address 46

verifying stage data population- snowflake import csv

The screenshot shows the Snowflake UI with the database set to 'MY_DATABASE' and schema to 'MY_SCHEMA'. A table named 'STUDENT_DEMO_RECORD_STAGE' is displayed, showing 46 rows of data. The columns are labeled 'NAME' and 'LAST_MODIFIED'. The table has 46 rows, each corresponding to one of the 46 entries in the CSV file above.

verifying stage data population - upload CSV to Snowflake

Loading data from the stage into a Snowflake table

Step 8: To load the data from the stage into a Snowflake table you can use the COPY INTO command. Choose the appropriate command based on your requirements:

- To load data as it is organized in the CSV file:

```
COPY INTO STUDENT_CSV_LOAD_DEMO2 FROM @student_demo_record_stage;
```

- To import only specific columns from the stage:

```
COPY INTO STUDENT_CSV_LOAD_DEMO2
FROM (
  SELECT
    col.$1",
    col.$2",
    col.$3",
    col.$4",
    col.$5",
    col.$6"
  FROM @student_demo_record_stage (file_format => student_csv_format) col
)
ON_ERROR = 'CONTINUE';
```



Loading data from stage to Snowflake table - upload CSV to snowflake

Validating the data

Step 9: Run a SELECT query to check if the Snowflake database table is populated with the data:

```
SELECT * from STUDENT_CSV_LOAD_DEMO2;
```

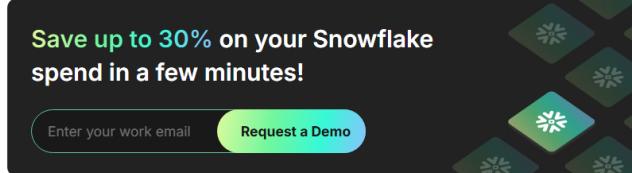
STUDENT_ID	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	EMAIL	ADDRESS
1	First Name 1	Last Name 1	1997-09-01	email1@example.com	Address 1
2	First Name 2	Last Name 2	2002-05-09	email2@example.com	Address 2
3	First Name 3	Last Name 3	2001-10-13	email3@example.com	Address 3
4	First Name 4	Last Name 4	2001-06-11	email4@example.com	Address 4
5	First Name 5	Last Name 5	2002-11-16	email5@example.com	Address 5
6	First Name 6	Last Name 6	2005-05-09	email6@example.com	Address 6
7	First Name 7	Last Name 7	1997-12-09	email7@example.com	Address 7
8	First Name 8	Last Name 8	2001-07-06	email8@example.com	Address 8
9	First Name 9	Last Name 9	1998-04-29	email9@example.com	Address 9
10	First Name 10	Last Name 10	2000-09-08	email10@example.com	Address 10
11	First Name 11	Last Name 11	2001-01-07	email11@example.com	Address 11
12	First Name 12	Last Name 12	2004-10-02	email12@example.com	Address 12
13	First Name 13	Last Name 13	1999-04-25	email13@example.com	Address 13

14	First Name 14	Last Name 14	1998-07-17	email14@example.com	Address 14
15	First Name 15	Last Name 15	1998-05-16	email15@example.com	Address 15
16	First Name 16	Last Name 16	1997-10-18	email16@example.com	Address 16
17	First Name 17	Last Name 17	1997-06-29	email17@example.com	Address 17
18	First Name 18	Last Name 18	2001-05-18	email18@example.com	Address 18
19	First Name 19	Last Name 19	2002-12-10	email19@example.com	Address 19
20	First Name 20	Last Name 20	2003-11-23	email20@example.com	Address 20
21	First Name 21	Last Name 21	2004-08-15	email21@example.com	Address 21
22	First Name 22	Last Name 22	2005-09-02	email22@example.com	Address 22
23	First Name 23	Last Name 23	2006-11-03	email23@example.com	Address 23
24	First Name 24	Last Name 24	2001-02-02	email24@example.com	Address 24
25	First Name 25	Last Name 25	1999-08-06	email25@example.com	Address 25
26	First Name 26	Last Name 26	1999-05-20	email26@example.com	Address 26
27	First Name 27	Last Name 27	1998-12-15	email27@example.com	Address 27
28	First Name 28	Last Name 28	2003-03-08	email28@example.com	Address 28
29	First Name 29	Last Name 29	2000-09-30	email29@example.com	Address 29
30	First Name 30	Last Name 30	2001-10-21	email30@example.com	Address 30
31	First Name 31	Last Name 31	1999-03-05	email31@example.com	Address 31
32	First Name 32	Last Name 32	2002-09-17	email32@example.com	Address 32
33	First Name 33	Last Name 33	2004-08-01	email33@example.com	Address 33
34	First Name 34	Last Name 34	2001-11-24	email34@example.com	Address 34
35	First Name 35	Last Name 35	2004-02-23	email35@example.com	Address 35
36	First Name 36	Last Name 36	2003-05-07	email36@example.com	Address 36
37	First Name 37	Last Name 37	2004-03-28	email37@example.com	Address 37
38	First Name 38	Last Name 38	1999-07-20	email38@example.com	Address 38
39	First Name 39	Last Name 39	2000-02-08	email39@example.com	Address 39
40	First Name 40	Last Name 40	2002-12-10	email40@example.com	Address 40
41	First Name 41	Last Name 41	2004-01-01	email41@example.com	Address 41
42	First Name 42	Last Name 42	1997-10-30	email42@example.com	Address 42
43	First Name 43	Last Name 43	2001-06-29	email43@example.com	Address 43
44	First Name 44	Last Name 44	2001-09-06	email44@example.com	Address 44
45	First Name 45	Last Name 45	2001-10-05	email45@example.com	Address 45
46	First Name 46	Last Name 46	1999-04-06	email46@example.com	Address 46
47	First Name 47	Last Name 47	2001-11-15	email47@example.com	Address 47
48	First Name 48	Last Name 48	2004-07-28	email48@example.com	Address 48
49	First Name 49	Last Name 49	1998-06-30	email49@example.com	Address 49
50	First Name 50	Last Name 50	2003-07-07	email50@example.com	Address 50
51	First Name 51	Last Name 51	1998-07-27	email51@example.com	Address 51
52	First Name 52	Last Name 52	2004-08-11	email52@example.com	Address 52
53	First Name 53	Last Name 53	2001-12-30	email53@example.com	Address 53
54	First Name 54	Last Name 54	2000-08-10	email54@example.com	Address 54
55	First Name 55	Last Name 55	1998-03-02	email55@example.com	Address 55
56	First Name 56	Last Name 56	1997-06-28	email56@example.com	Address 56

Verifying if the Snowflake database table is populated with the data - upload CSV to snowflake

The screenshot shows the Snowflake Web Interface with the database set to 'MY_DATABASE / MY_SCHEMA / STUDENT_CSV_LOAD_DEMO'. A table named 'STUDENT' is displayed with columns: STUDENT_ID, FIRST_NAME, LAST_NAME, DATE_OF_BIRTH, EMAIL, and ADDRESS. The table contains 56 rows of student data, each with a unique ID and name combination, along with their birthdate, email address, and address.

Table successfully loaded with data - snowflake import csv



Conclusion

And that's it—your CSV file is now securely stored and ready to be analyzed in Snowflake! So by following these steps carefully, you can easily unlock the potential of your data and open up a whole new world of insights.

This powerful platform makes it easy to unlock the potential of your data and gain valuable insights. Whether you used the visual Web Interface or wrote a script to programmatically load your data, Snowflake's flexible import options enable you to get your data in quickly and easily.

The real fun begins now. You can start querying your data, building dashboards and reports, and turbocharging your analytics. Snowflake gives you the performance, scalability, and tools you need to gain valuable business insights fast. So what are you waiting for? Start exploring your data today!

FAQs

Can I load multiple CSV at once into Snowflake?

Yes, Snowflake allows you to load multiple CSV simultaneously using the appropriate commands or by selecting multiple files in the Snowflake Web Interface.

Are there any file size limitations when loading / uploading CSV to Snowflake?

Snowflake supports loading large CSV, ranging from kilobytes to terabytes in size. The actual limits depend on your Snowflake account configuration and the resources available.

Can I automate the process of importing / uploading CSV to Snowflake?

Yes, you can automate the process by utilizing Snowflake's SnowSQL command-

line client or by integrating Snowflake with other ETL (Extract, Transform, Load) or data integration tools.

Does Snowflake provide error handling mechanisms during the CSV import process?

Yes, Snowflake provides detailed error messages and logs to help identify and resolve any issues that may occur during the CSV import process. These error messages assist in troubleshooting and ensuring successful data loading.

Can I import / upload CSV to Snowflake with varying column structures?

Yes, Snowflake accommodates flexible column structures when importing CSV files. You can define the appropriate table structure and map the CSV columns to the corresponding table columns during the loading process.

What are the benefits to upload CSV to Snowflake?

Uploading CSV to Snowflake allows for easy data integration and analysis. Snowflake's cloud-based data warehousing platform can handle large amounts of data and allows for real-time querying and analysis. Additionally, uploading CSV files to Snowflake can save time and resources compared to traditional data warehousing methods.

How can I troubleshoot errors when uploading CSV to Snowflake? Some common errors when uploading CSV to Snowflake include formatting issues, incorrect file encoding, and missing or incorrect headers. To troubleshoot these errors, you can check the file format and encoding, review the headers and data in the file, and use Snowflake's error messages and logs to identify specific issues.

Does Snowflake support CSV?

Yes, Snowflake supports CSV file format for data loading. You can easily upload CSV files to Snowflake using the COPY command or Snowflake's web interface. Snowflake also supports other file formats such as JSON, Avro, Parquet, and more.

Can we load an Excel file in Snowflake?

Yes, you can load Excel files in Snowflake, but you need to first save the Excel file as a CSV file. Snowflake supports loading data from CSV files, which can be easily created from Excel files. Once you have saved the Excel file as a CSV file, you can upload it to Snowflake.

Tags

[Snowflake Upload CSV](#) [Import CSV to Snowflake](#)

[Load Data Into Snowflake From CSV](#) [Snowflake CSV](#)

[Snowflake File Format CSV](#) [Snowflake Import CSV](#) [Data Loading](#)

[Snowflake File](#)



Pramit Marattha

Technical Content Lead

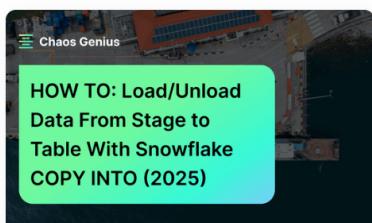
Pramit is a Technical Content Lead at Chaos Genius.

People who are also involved

[Preeti Shrimai](#), [Varun P](#), [Kartikay Bagla](#)

Continue Reading

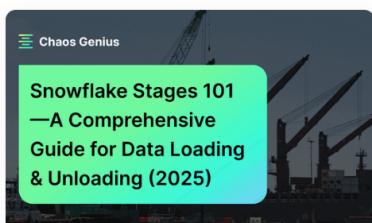
[See More →](#)



HOW TO: Load/Unload Data From Stage to Table With Snowflake COPY INTO (2025)

[Snowflake COPY INTO](#)

HOW TO: Use Snowflake COPY INTO to Load/Unload Data (2025)



Snowflake Stages 101 —A Comprehensive Guide for Data Loading & Unloading (2025)

[Snowflake Stages](#)

Snowflake Stages 101: A Comprehensive Guide (2025)



HOW TO: Create and Manage Snowflake File Formats (2025)

[Snowflake File Formats](#)

HOW TO: Create and Manage Snowflake File Formats (2025)

Jan 8, 2024 • 29 min read

Dec 18, 2023 • 32 min read

Aug 31, 2023 • 15 min read

"Chaos Genius has been a game-changer for our DataOps at NetApp. Thanks to the precise recommendations, intuitive interface and predictive capabilities, we were able to lower our **Snowflake costs by 28%, yielding us a 20X ROI!**"



Anju Mohan
Director, IT

Join today to get upto
30% Snowflake savings

Unlock Snowflake Savings

Chaos Genius

Autonomous Agents for Snowflake & Databricks
Cost Optimization.



Copyright © 2025 GoodHealth Technologies, Inc. All rights reserved.

Company

[About Us](#)

Legal

[Terms](#)

[Security](#)

[Privacy](#)

[Cookies](#)

Resources

[Blog](#)

[Documentation](#)

Contact

founders@chaosgenius.io

Backed By



[Subscribe](#)