

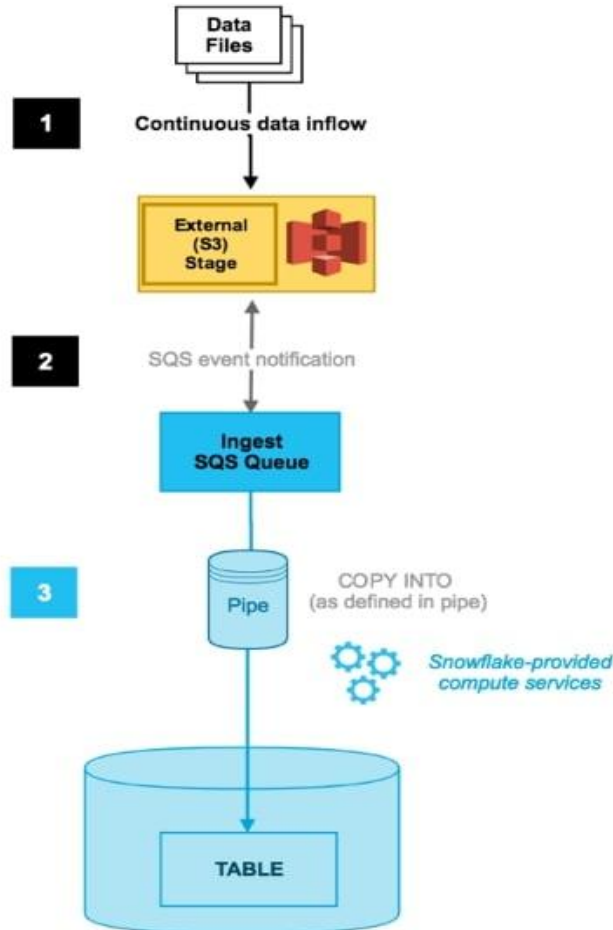
Semi-Structured Data

Data loading from AWS S3 to Snowflake through SnowPipe


Steps For Migration

1. Creating AWS S3 bucket
2. Creating Database , Schema & Table in SnowFlake
3. Creating File Format
4. Creating Named Stage
5. Creating Pipe
6. Setting notification in AWS
7. Converting loaded data into Relational Table

Architecture Diagram Of Snowpipe



Creating Bucket in AWS S3

 Services [Alt+S]

Global ▼ mmk999mmk ▼

Amazon S3 ×

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

Access analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 3

▶ AWS Marketplace for S3






Amazon S3

▶ Account snapshot


Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

Buckets (2) [Info](#)

  Copy ARN  Empty  Delete  Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

< 1 > 

	Name ▲	AWS Region ▼	Access ▼	Creation date ▼
<input type="radio"/>	mymanojbucket	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	September 29, 2021, 11:09:04 (UTC+05:30)
<input type="radio"/>	rahul-project	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	November 26, 2021, 12:31:00 (UTC+05:30)

Create Database , Schema and Table

The screenshot displays the Snowflake SQL Worksheet interface. The top navigation bar includes icons for Databases, Shares, Data Marketplace, Warehouses, Worksheets (active), History, Partner Connect, Help, and Snowsight. The user is identified as RAHUL SYSADMIN.

The main workspace shows a query titled "JSON" with a green checkmark indicating successful execution. The query text is:

```
--Create Temporary Table for JSON
create or replace table PROJECT.MINIPROJECT.REST(json_data variant);
```

The left sidebar shows the database hierarchy: DEMO_DB, PROJECT, INFORMATION_SCHEMA, MINIPROJECT (expanded to show Tables and Views), PUBLIC, SNOWFLAKE_SAMPLE_DATA, and UTIL_DB.

Below the query editor, the "Results" tab is active, showing a single row of results:

Row	status
1	Table REST successfully created.

Additional details in the Results section include a green checkmark, "Query ID", "SQL", execution time of "139ms", and "1 rows". There is also a "Filter result..." input field and a "Copy" button.

Create File Format

The screenshot displays the Snowflake Snowsight web interface. The top navigation bar includes icons for Databases, Shares, Data Marketplace, Warehouses, Worksheets (active), and History. On the right, there are links for Partner Connect, Help, and Snowsight, along with the user name RAHUL SYSADMIN.

The main workspace shows a query titled "JSON" with a green checkmark indicating successful execution. The query text is as follows:

```
1 --Create Temporary Table for JSON
2 create or replace table PROJECT.MINIPROJECT.REST(json_data variant);
3
4 --Create JSON File Format
5 create or replace file format JSON
6 type="JSON"
7 strip_outer_array = true;
```

Below the query editor, the "Results" tab is active, showing a single row of results:

Row	status
1	File format JSON successfully created.

The interface also features a left sidebar with a "Find database objects" search bar and a tree view of the database structure, including DEMO_DB, PROJECT (with INFORMATION_SCHEMA, MINIPROJECT, and PUBLIC sub-databases), and SNOWFLAKE_SAMPLE_DATA. The bottom of the results section includes a "Filter result..." input, a "Copy" button, and a "Columns" dropdown.

Create Named Stage

The screenshot displays the Snowflake Snowsight interface. The top navigation bar includes icons for Databases, Shares, Data Marketplace, Warehouses, Worksheets (active), and History. On the right, there are links for Partner Connect, Help, and Snowsight, along with the user name RAHUL SYSADMIN.

The main workspace is titled "JSON" and shows a SQL query being executed. The query is as follows:

```
1 --Create Temporary Table for JSON
2 create or replace table PROJECT.MINIPROJECT.REST(json_data variant);
3
4 --Create JSON File Format
5 create or replace file format JSON
6 type="JSON"
7 strip_outer_array = true;
8
9
10 --Create External Named Stage
11 create or replace stage MYS3
12 url = 's3://rahul-project'
13 credentials =(aws_key_id= 'AKIA2NXYEXDPWI4M2H4M'
14               aws_secret_key = 'gvf0CBDrxxXgKyckS3Ma4AULwwHGc5qcGGaK+fWV' );
15
16
```

The query execution results are shown in a table with the following data:

Row	status
1	Stage area MYS3 successfully created.

The interface also includes a sidebar for finding database objects, a top bar with tabs for CustomerPipe, Order Review, and JSON, and a bottom bar with a "Results" tab and a "Data Preview" section.

Create Pipe

The screenshot displays the Snowflake SQL Editor interface. The top navigation bar includes icons for Databases, Shares, Data Marketplace, Warehouses, Worksheets (active), History, Partner Connect, Help, and Snowsight. The user is identified as RAHUL SYSADMIN.

The editor is titled "CustomerPipe" and shows a query in the "JSON" format. The query is as follows:

```
7 strip_outer_array = true,
8
9
10 --Create External Named Stage
11 create or replace stage MYS3
12 url = 's3://rahul-project'
13 credentials =(aws_key_id= 'AKIA2NXYEXDPWI4M2H4M'
14               aws_secret_key = 'gvf0CBDrxXxgKyckS3Ma4AULwwHGc5qcG6aK+fWV');
15
16 --Create Pipe
17 create or replace pipe REST_SNOWPIPE
18 auto_ingest = true as
19 copy into REST
20 from @MYS3
21 file_format= JSON;
22
23
24
```

The query has been executed successfully, as indicated by the "Run" button and the "Results" section. The results show a single row with the status "Pipe REST_SNOWPIPE successfully created."

The left sidebar shows the database structure, including DEMO_DB, PROJECT, INFORMATION_SCHEMA, MINIPROJECT, PUBLIC, SNOWFLAKE_SAMPLE_DATA, and UTIL_DB.

The bottom section displays the "Results" tab, showing the query ID, SQL, execution time (150ms), and the number of rows (1 row). A filter result... input field and a "Copy" button are also present.

Row	status
1	Pipe REST_SNOWPIPE successfully created.

Use 'show pipes' command to see pipe and copy notification_channel

The screenshot displays the Snowflake SQL Editor interface. The top navigation bar includes icons for Databases, Shares, Data Marketplace, Warehouses, Worksheets (active), History, Partner Connect, Help, and Snowsight. The user is identified as RAHUL SYSADMIN. The current worksheet is titled 'CustomerPipe' and shows a query execution history with a 'Run' button and a status of 'All Queries | Saved 16 seconds ago'.

The SQL query executed is as follows:

```
--Create External Named Stage
create or replace stage MYS3
url = 's3://rahul-project'
credentials =(aws_key_id= 'AKIA2NXYEXDPWI4M2H4M'
aws_secret_key ='gvf0CBDrxXxgKyckS3Ma4AULwwHGc5qcGGaK+fWV' );


--Create Pipe
create or replace pipe REST_SNOWPIPE
auto_ingest = true as
copy into REST
from @MYS3
file_format= JSON;

show pipes;
```

The query results are displayed in a table with 11 columns: Row, created_on, name, database_name, schema_name, definition, owner, notification_channel, comment, integration, and pattern. The results show one row of data for a pipe named 'CUSTOMER_PIPE'.

Row	created_on	name	database_name	schema_name	definition	owner	notification_channel	comment	integration	pattern
1	2021-12-01 ...	CUSTOMER_PIPE	PROJECT	MINIPROJECT	COPY INTO ...	SYSADMIN	arn:aws:sqs:...		NULL	NULL

Setting Notification

i Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. [Learn more](#) 

Destination

Choose a destination to publish the event. [Learn more](#) 

- ☐ **Lambda function**
Run a Lambda function script based on S3 events.
- ☐ **SNS topic**
Send notifications to email, SMS, or an HTTP endpoint.
- ☒ **SQS queue**
Send notifications to an SQS queue to be read by a server.

Specify SQS queue

- ☐ Choose from your SQS queues
- ☒ Enter SQS queue ARN





SQS queue


arn:aws:sqs:ap-south-1:759614567829:sf-snowpipe-AIDA3BXEOOWKQBCMGU5ZU-G


Cancel

Save changes



Uploading Data

 Services [Alt+S]   Global 

 **Upload succeeded**
View details below.

 The information below will no longer be available after you navigate away from this page.

Summary


Destination s3://rahul-project	Succeeded  1 file, 1.8 MB (100.00%)	Failed  0 files, 0 B (0%)
---	---	---

Files and folders

Configuration

Files and folders (1 Total, 1.8 MB)

 < 1 >

Name	Folder	Type	Size	Status	Error
Sample Json Data.json	-	application/json	1.8 MB	 Succeeded	-

Verify loaded data

The screenshot displays the Snowflake SQL Editor interface. The top navigation bar includes icons for Databases, Shares, Data Marketplace, Warehouses, Worksheets (active), and History. On the right, there are links for Partner Connect, Help, and Snowsight, along with the user name RAHUL SYSADMIN.

The main workspace shows a query titled "CustomerPipe" with a status of "JSON". The query text is as follows:

```
--Create Pipe
17 create or replace pipe REST_SNOWPIPE
18 auto_ingest = true as
19 copy into REST
20 from @MYS3
21 file_format= JSON;
22
23
24 show pipes;
25
26
27 ALTER PIPE REST_SNOWPIPE refresh;
28
29 --Verifying loaded data
30 select * from REST;
31
```

Below the query editor, the "Results" tab is active, showing a "Data Preview" of the query results. The preview indicates that the query executed successfully in 1.36s and returned 1 row. The data is displayed in a table with the following structure:

Row	JSON_DATA
1	{ "crs": { "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84", "type": "name" }, "features": [{ "geometry": { "coordinates": [-122.318798848, 47.613361605000101], "t...

Transforming loaded data into Relational Table

The screenshot displays the Snowflake SQL Editor interface. The top navigation bar includes icons for Databases, Shares, Data Marketplace, Warehouses, Worksheets (active), and History. On the right, there are links for Partner Connect, Help, and Snowsight, along with the user name RAHUL SYSADMIN.


The left sidebar shows the database hierarchy: DEMO_DB, PROJECT, INFORMATION_SCHEMA, MINIPROJECT (expanded to show Tables and Views), ML, PUBLIC, SNOWFLAKE_SAMPLE_DATA, and UTIL_DB.

The main editor area shows a SQL query with line numbers 33 to 56. The query is as follows:


```
33 -- Convert into Relational Table using FLATTEN Function
34
35 select
36 value: geometry:coordinates[0] as C1,
37 value: geometry:coordinates[1] as C2,
38 value: geometry:type as Type,
39 value: properties:CreationDate as CreationDate,
40 value: properties:EditDate as EditDate,
41 value: properties:address_help_your_customer_find as Address,
42 value: properties:are_you_open_for as OpenFor,
43 value: properties:business_name as BusinessName,
44 value: properties:business_website as BusinessWebsite,
45 value: properties:globalid as GlobalID,
46 value: properties:objectid as ObjectID,
47 value: properties:phone_number_to_place_an_order as PhoneNumber,
48 value: properties:provide_details_for_you_busines as Details
49 from REST ,
50 lateral flatten(input =>json_data:features);
51
52
53
54
55
56
```

Below the query editor, there are tabs for 'Results' (active) and 'Data Preview'. The 'Results' tab is currently empty. The bottom right corner has a link to 'Open History'.


Relational Table




Databases




Shares




Data Marketplace



Warehouses




Worksheets



History

Partner Connect

Help

Snowsight

RAHUL
SYSADMIN

CustomerPipe

JSON

+ ▼

Find database objects

Starting with...

DEMO_DB

PROJECT

INFORMATION_SCHEMA

MINIPROJECT

Tables

Views

ML

PUBLIC

SNOWFLAKE_SAMPLE_DATA

UTIL_DB

Run

☐ All Queries

Saved 29 seconds ago

SYSADMIN

COMPUTE_WH (XS)

PROJECT

MINIPROJECT

...

```
33 -- Convert into Relational Table using FLATTEN Function
34
35 select
36 value: geometry:coordinates[0] as C1,
37 value: geometry:coordinates[1] as C2,
38 value: geometry:type as Type,
```

Results

Data Preview

Open History

Query ID

SQL

280ms

2,880 rows

Filter result...

Download

Copy

Columns ▼

↗

Cleaning Transformation

The screenshot displays the Snowflake Data Cloud interface. The top navigation bar includes icons for Databases, Shares, Data Marketplace, Warehouses, Worksheets (active), and History. On the right, there are links for Partner Connect, Help, and Snowsight, along with the user name RAHUL SYSADMIN.

The main workspace shows a worksheet named "CustomerPipe" with a JSON data source. The "Run" button is highlighted, and the status indicates "All Queries" and "Saved 1 minute ago".

The left sidebar lists the database objects: DEMO_DB, PROJECT, INFORMATION_SCHEMA, MINIPROJECT (with sub-items Tables and Views), ML, PUBLIC, SNOWFLAKE_SAMPLE_DATA, and UTIL_DB.

The SQL query in the editor is as follows:

```
51
52 --Final Table after Cleaning our Data with Casting
53
54 select
55 value: geometry:coordinates[0]::number as C1,
56 value: geometry:coordinates[1]::number as C2,
57 value: geometry:type::String as Type,
58 value: properties:CreationDate::datetime as CreationDate,
59 value: properties:EditDate::datetime as EditDate,
60 value: properties:address_help_your_customer_find::String as Address,
61 value: properties:are_you_open_for::String as OpenFor,
62 value: properties:business_name::String as BusinessName,
63 value: properties:business_website::String as BusinessWebsite,
64 value: properties:globalid::String as GlobalID,
65 value: properties:objectid::number as ObjectID,
66 value: properties:phone_number_to_place_an_order::String as PhoneNumber,
67 value: properties:provide_details_for_you_business::String as Details
68 from REST ,
69 lateral flatten(input =>json_data:features);
70
71
72
73
74
75
```

At the bottom, the "Results" tab is active, and a "Data Preview" button is visible. An "Open History" link is located in the bottom right corner.

Final Relational Table

Databases

Shares

Data Marketplace

Warehouses

Worksheets

History

Partner Connect

Help

Snowsight

RAHUL
SYSADMIN

CustomerPipe

JSON

+ -

Find database objects

Starting with...

DEMO_DB

PROJECT

INFORMATION_SCHEMA

MINIPROJECT

- Tables
- Views

ML

PUBLIC

SNOWFLAKE_SAMPLE_DATA

UTIL_DB

Run

All Queries

Saved 6 minutes ago

SYSADMIN

COMPUTE_WH (XS)

PROJECT

MINIPROJECT

...

```
51
52 --Final Table after Cleaning our Data with Casting
53
54 select
55 value: geometry:coordinates[0]::number as C1,
56 value: geometry:coordinates[1]::number as C2,
57 value: geometry:type::String as Type,
```

Results

Data Preview

Open History

Query ID

SQL

50ms

2,880 rows

Filter result...

Download

Copy

Columns

...

Row	C1	C2	TYPE	CREATIONDATE	EDITDATE	ADDRESS	OPENFOR	BUSINESSNAME	BUSINESSWEBS	GLOBALID
1	-122	48	Point	2020-03-27 ...	2020-06-04...	1040 E Unio...	curbside	Bateau	https://www....	12aba10d-78.
2	-122	48	Point	2020-03-27 ...	2020-03-27 ...	1519 14th Ave	takeout,delv...	Nue	https://www....	4b271d79-3..
3	-122	48	Point	2020-03-27 ...	2020-03-27 ...	1501 Melros...	takeout,curb...	Terra Plata	http://www.t...	f4928649-8..
4	-122	48	Point	2020-03-27 ...	2020-06-04...	419 6th Ave ...	takeout,third...	A+ Hong Ko...	https://www....	0745b48d-8..
5	-122	48	Point	2020-03-27 ...	2020-05-11 ...	2415 Airport...	delivery	Gourmondo	https://cater...	0afb2fdb-9e.
6	-122	48	Point	2020-03-27 ...	2020-03-27 ...	4523 Califor...	takeout	Lady Jaye	https://www....	70882582-e..
7	-122	48	Point	2020-03-27 ...	2020-06-04...	2460 Wester...	takeout,delv...	Buddha Bru...	https://www....	995642d3-0.