

## Loading Of Unstructured Data Using JASON In Snowflake

### Step 1: Creating Stage and Raw File :

#### Code:

#### // First step: Load Raw JSON

```
CREATE OR REPLACE stage MANAGE_DB.EXTERNAL_STAGES.JSONSTAGE  
    url='s3://bucketsnowflake-jsondemo';
```

```
CREATE OR REPLACE file format MANAGE_DB.FILE_FORMATS.JSONFORMAT  
    TYPE = JSON;
```

```
CREATE OR REPLACE table OUR_FIRST_DB.PUBLIC.JSON_RAW (  
    raw_file variant);
```

```
COPY INTO OUR_FIRST_DB.PUBLIC.JSON_RAW  
    FROM @MANAGE_DB.EXTERNAL_STAGES.JSONSTAGE  
    file_format= MANAGE_DB.FILE_FORMATS.JSONFORMAT  
    files = ('HR_data.json');
```

```
SELECT * FROM OUR_FIRST_DB.PUBLIC.JSON_RAW;
```

- [HR\\_data.json](#) – This is the HR data file which is basically your JASON file
- The Data is been Loaded as you can see from the below image

The screenshot shows a SQL query execution interface. The query is as follows:

```

1 // First step: Load Raw JSON
2
3 CREATE OR REPLACE stage MANAGE_DB.EXTERNAL_STAGES.JSONSTAGE
4   url='s3://bucketandflake-jsondemo';
5
6 CREATE OR REPLACE file format MANAGE_DB.FILE_FORMATS.JSONFORMAT
7   TYPE = JSON;
8
9
10 CREATE OR REPLACE table OUR_FIRST_DB.PUBLIC.JSON_RAW (
11   raw_file variant);
12
13 COPY INTO OUR_FIRST_DB.PUBLIC.JSON_RAW
14   FROM @MANAGE_DB.EXTERNAL_STAGES.JSONSTAGE
15   file_format= MANAGE_DB.FILE_FORMATS.JSONFORMAT
16   files = ('HR_data.json');
17
18
19 SELECT * FROM OUR_FIRST_DB.PUBLIC.JSON_RAW;

```

The results table shows the following data:

file	status	rows_parsed	rows_loaded	error_limit	errors_seen	first_
s3://bucketandflake-jsondemo/HR_data.json	LOADED	200	200	1	0	Full

Query Details:

- Query duration: 967ms
- Rows: 1
- Query ID: 01b5117a-0001-a498-Q

- Once the table is well executed you will be able to see like the given image below it is correctly interrupter

The screenshot shows a SQL query execution interface. The query is as follows:

```

1 // First step: Load Raw JSON
2
3 CREATE OR REPLACE stage MANAGE_DB.EXTERNAL_STAGES.JSONSTAGE
4   url='s3://bucketandflake-jsondemo';
5
6 CREATE OR REPLACE file format MANAGE_DB.FILE_FORMATS.JSONFORMAT
7   TYPE = JSON;
8
9
10 CREATE OR REPLACE table OUR_FIRST_DB.PUBLIC.JSON_RAW (
11   raw_file variant);
12
13 COPY INTO OUR_FIRST_DB.PUBLIC.JSON_RAW
14   FROM @MANAGE_DB.EXTERNAL_STAGES.JSONSTAGE

```

The results table shows the following data:

RAW_FILE
{ "city": "Bakersfield", "first_name": "Portia", "gender": "Male", "id": 1, "job": { "salary": 32000, "title": "Financial Analyst" }, "last_name": "Gloani", "prev_company": [] }
{ "city": "Louny", "first_name": "Dag", "gender": "Male", "id": 2, "job": { "salary": 43000, "title": "Clinical Specialist" }, "last_name": "Gloani", "prev_company": [] }
{ "city": "Wates", "first_name": "Heath", "gender": "Female", "id": 3, "job": { "salary": 40800, "title": "Research Assistant I" }, "last_name": "Gloani", "prev_company": [] }
{ "city": "Umeå", "first_name": "Dita", "gender": "Female", "id": 4, "job": { "salary": 14400, "title": "Assistant Media Planner" }, "last_name": "Gloani", "prev_company": [] }
{ "city": "Shangdian", "first_name": "Nikki", "gender": "Female", "id": 5, "job": { "salary": 34600, "title": "Administrative Officer" }, "last_name": "Gloani", "prev_company": [] }
{ "city": "Saurimo", "first_name": "Austin", "gender": "Female", "id": 6, "job": { "salary": 25800, "title": "Compensation Analyst" }, "last_name": "Gloani", "prev_company": [] }
{ "city": "Kangar", "first_name": "Tessa", "gender": "Female", "id": 7, "job": { "salary": 30600, "title": "Data Coordinator" }, "last_name": "Gloani", "prev_company": [] }

Query Details:

- Query duration: 967ms
- Rows: 1
- Query ID: 01b5117a-0001-a498-Q

## STEP 2 : Parsing And Analyzing :

The screenshot shows a Snowflake SQL editor interface. The left sidebar displays a database schema with a table named `OUR_FIRST_DB.PUBLIC.JSON_RAW`. The main editor contains a SQL query:

```
1 // Second step: Parse & Analyse Raw JSON
2 // Selecting attribute/column
3
4 SELECT RAW_FILE:city FROM OUR_FIRST_DB.PUBLIC.JSON_RAW;
5
6 SELECT $1:first_name FROM OUR_FIRST_DB.PUBLIC.JSON_RAW;
7
8
9 // Selecting attribute/column - formatted
10
11
12 SELECT RAW_FILE:first_name::string as first_name FROM OUR_FIRST_DB.PUBLIC.JSON_RAW;
13
14 SELECT RAW_FILE:id::int as id FROM OUR_FIRST_DB.PUBLIC.JSON_RAW;
15
```

The **Results** pane at the bottom shows a table with one column, `FIRST_NAME`, and one row containing the value `Portia`.

## Well Structured Data :

The screenshot shows a Snowflake SQL editor interface. The main editor contains a SQL query:

```
10 // Selecting attribute/column - formatted
11
12 SELECT RAW_FILE:first_name::string as first_name FROM OUR_FIRST_DB.PUBLIC.JSON_RAW;
13
14 SELECT RAW_FILE:id::int as id FROM OUR_FIRST_DB.PUBLIC.JSON_RAW;
15
16
17 SELECT
18   RAW_FILE:id::int as id,
19   RAW_FILE:first_name::STRING as first_name,
20   RAW_FILE:last_name::STRING as last_name,
21   RAW_FILE:gender::STRING as gender
22 FROM OUR_FIRST_DB.PUBLIC.JSON_RAW;
23
24
```

The **Results** pane at the bottom shows a table with four columns: `ID`, `FIRST_NAME`, `LAST_NAME`, and `GENDER`. The table contains 7 rows of data:

	ID	FIRST_NAME	LAST_NAME	GENDER
1	1	Portia	Gioani	Male
2	2	Dag	Cronay	Male
3	3	Heathi	Lackmann	Female
4	4	Dita	Deering	Female
5	5	Nikki	McCosh	Female
6	6	Austina	Laux	Female
7	7	Tyrene	Nelson	Female

The **Query Details** pane on the right shows the query duration as 59ms, the number of rows as 200, and the query ID as `01b51188-0001-e495-0-`. A histogram of the `ID` column is also displayed.