

JSON_SIMPLE AND NESTED

To Create a Table

Steps:

Step-1 Snowsql -v

Step-2 Snowsql -a ____ -u ____ -p ____

Step-3 Create database emp_json

Step-4 Create schema emp_schema

Step-5 Create table employees_json(raw variant);

Output:

```
C:\Users\admin>snowsql -v
Version: 1.4.5

C:\Users\admin>snowsql -a FSBBSTN-IH05083 -u ROSHINI
Password:
ROSHINI#COMPUTE_WH@EMP_JSON.PUBLIC>create database emp_json;
+-----+
| status
+-----+
| Database EMP_JSON successfully created.
+-----+
1 Row(s) produced. Time Elapsed: 0.200s
ROSHINI#COMPUTE_WH@EMP_JSON.PUBLIC>create schema emp_schema;
+-----+
| status
+-----+
| Schema EMP_SCHEMA successfully created.
+-----+
1 Row(s) produced. Time Elapsed: 0.168s
ROSHINI#COMPUTE_WH@EMP_JSON.EMP_SCHEMA>create table employees_json (raw var
iant);
+-----+
```

Output for Creating New Table

Step-6 Create stage emp_stage;

Step-7 Put 'file://C:/Users/admin/Downloads/employees.json' @emp_stage;

**Step-8 create or replace file format ff_json_flat type = 'JSON'
strip_outer_array = false;**

**Step-9 copy into employees_json from @emp_stage/employees.json
file_format =
(format_name= ff_json_flat);**

Output:

```
1 Row(s) produced. Time Elapsed: 0.208s
ROSHINI#COMPUTE_WH@EMP_JSON.EMP_SCHEMA>create stage emp_stage;
+-----+
| status
| Stage area EMP_STAGE successfully created.
+-----+
1 Row(s) produced. Time Elapsed: 0.337s
ROSHINI#COMPUTE_WH@EMP_JSON.EMP_SCHEMA>put 'file://C:/Users/admin/Downloads
/employees.json' @emp_stage;
+-----+
| source      | target      | source_size | target_size | source_compression | target_compression | status   | message
| employees.json | employees.json.gz |       707 |       288 | NONE           | GZIP            | UPLOADED |
+-----+
1 Row(s) produced. Time Elapsed: 1.383s
ROSHINI#COMPUTE_WH@EMP_JSON.EMP_SCHEMA>create or replace file format ff_js
n_flat type = 'JSON' strip_outer_ar
ay = false;
+-----+
| status
| File format FF_JSON_FLAT successfully created.
+-----+
1 Row(s) produced. Time Elapsed: 0.619s
ROSHINI#COMPUTE_WH@EMP_JSON.EMP_SCHEMA>copy into employees_json from @emp_s
tage/employees.json file_format = {
    format_name = ff_json_flat} on_error
    = 'continue';
+-----+
| file          | status | rows_parsed | rows_loaded | error_limit | errors_seen | first_error | first_error_line | first_error_character | fi
rst_error_column_name |
| emp_stage/employees.json.gz | LOADED |      1 |       1 |       1 |       0 | NULL     |           NULL |           NULL | NULL
+-----+
```

Output for Copy the data to snowflake

Step-10 `select * from employees_json;`

```
Row(s) produced. Time Elapsed: 2.405s
OSHINI#COMPUTE_WH@EMP_JSON.EMP_SCHEMA>select * from employees_json;
+-----+
| RAW |
+-----+
[{"id": 101, "name": "John Doe", "salary": 5000, "department": "HR"}, {"id": 102, "name": "Jane Smith", "salary": 6000, "department": "Finance"}, {"id": 103, "name": "Michael Brown", "salary": 5500, "department": "IT"}, {"id": 104, "name": "Linda White", "salary": 5200, "department": "Marketing"}, {"id": 105, "name": "James Black", "salary": 5800, "department": "IT"}, {"id": 106, "name": "Emma Green", "salary": 5100, "department": "HR"}, {"id": 107, "name": "David Blue", "salary": 5300, "department": "Finance"}]
```

Output for normal json

In this result i got this output but i want in j_son method so i modify the select query into **Json_simple**.

```

Select      Value:ID::Int As ID,
              Value:Name:: String As Name,
              Value:Department:: String As Department,
              Value:Salary:: Int As Salary

from Employee.json,
Lateral Flatten,(input=>raw);

```

Output:

```

ROSHINI#COMPUTE_WH@EMP_JSON.EMP_SCHEMA>SELECT
                                              value:id::INT AS ID,
                                              value:name::STRING AS Name,
                                              value:department::STRING AS Dept,
                                              value:salary::INT AS Salary
FROM employees_json,
      LATERAL FLATTEN(input => raw);

+----+-----+-----+-----+
| ID | NAME    | DEPT   | SALARY |
+----+-----+-----+-----+
| 101 | John Doe | HR     | 5000   |
| 102 | Jane Smith | Finance | 6000   |
| 103 | Michael Brown | IT     | 5500   |
| 104 | Linda White | Marketing | 5200   |
| 105 | James Black | IT     | 5800   |
| 106 | Emma Green | HR     | 5100   |
| 107 | Robert King | Finance | 6200   |
| 108 | Olivia Scott | Marketing | 5300   |
| 109 | William Lee | IT     | 5600   |
| 110 | Sophia Turner | HR     | 5000   |
+----+-----+-----+-----+

```

Output for select in json_simple

These are the step for select query:

- Step-1** The Table (Employee.json) contains JSON data.
- Step-2** FLATTEN(input => raw) breaks the JSON array into separate rows.
- Step-3** value:ID, value:Name, etc., extract fields from the JSON.
- Step-4** ::INT or ::STRING converts them into proper data types.
- Step-5** The query returns clean columns like ID, Name, Department, Salary

LATERAL FLATTEN is used to pull values out of a JSON/XML array and show each item as its own row.

JSON_NESTED

I already created database so i use that database and i will change the table and stage names.

Step-1 Create table nested_json;

Step-2 Create or replace stage nested_stage;

Step-3 put 'file://C:/Users/admin/Downloads/nested_json.txt'
@nested_stage;

Output:

```
ROSHINI#COMPUTE_WH@EMP_JSON.JSON_SCH>create table nested_json(raw variant);
+-----+
| status
+-----+
| Table NESTED_JSON successfully created.
+-----+
1 Row(s) produced. Time Elapsed: 0.218s
ROSHINI#COMPUTE_WH@EMP_JSON.JSON_SCH>create or replace stage nested_stage;
+-----+
| status
+-----+
| Stage area NESTED_STAGE successfully created.
+-----+
```

```
ROSHINI#COMPUTE_WH@EMP_JSON.JSON_SCH>put 'file://C:/Users/admin/Downloads/nested.json' @nested_stage;
+-----+-----+-----+-----+-----+-----+-----+-----+
| source | target | source_size | target_size | source_compression | target_compression | status | message |
+-----+-----+-----+-----+-----+-----+-----+-----+
| nested.json | nested.json.gz | 1473 | 496 | NONE | GZIP | UPLOADED | |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Output for Copy the data to snowflake

Step-4 create or replace file format ff_json_flat type = 'JSON' strip_outer_array = false;

Step-5 copy into nested_json from @nested_stage/nested.json file_format = (format_name = 'ff_json_flat');

```

ROSHINI#COMPUTE_WH@EMP_JSON.JSON_SCH>create or replace file format ff_json_flat type = 'JSON' strip_outer_array = false;
+-----+
| status |
+-----+
| File format FF_JSON_FLAT successfully created. |
+-----+
1 Row(s) produced. Time Elapsed: 0.903s
ROSHINI#COMPUTE_WH@EMP_JSON.JSON_SCH>copy into nested_json from @nested_stage/nested.json file_format = (format_name = 'ff_json_flat');
+-----+
| file | status | rows_parsed | rows_loaded | error_limit | errors_seen | first_error | first_error_line | first_error_ch
| character | first_error_column_name |
+-----+
| nested_stage/nested.json.gz | LOADED | 1 | 1 | 1 | 0 | NULL | NULL |
+-----+

```

Output for file_format and copy into snowsql

Step-6 select*from nested_json:

```

ROSHINI#COMPUTE_WH@EMP_JSON.JSON_SCH>select*from nested_json;
+-----+
| RAW |
+-----+
[ {
  "id": 101,
  "name": "John Doe",
  "projects": [
    {
      "duration": 3,
      "project_name": "Website Redesign"
    },
    {
      "duration": 4,
      "project_name": "Employee Portal"
    }
  ],
  "id": 102,
  "name": "Jane Smith",
  "projects": [
    {
      "duration": 5,
      "project_name": "Financial Report"
    },
    {
      "duration": 2,
      "project_name": "Audit Compliance"
    }
  ],
  "id": 103,
  "name": "Michael Brown",
  "projects": [
    {
      "duration": 6,
      "project_name": "System Upgrade"
    }
  ]
}
]
```

Select raw:id:int as id,

Raw:name:string as dept **from nesed_json:**

Output:

```

1 Row(s) produced. Time Elapsed: 1.928s
ROSHINI#COMPUTE_WH@EMP_JSON.JSON_SCH>SELECT
          raw:id::INT AS ID,
          raw:name::STRING AS Name,
          raw:department::STRING AS Dept
        FROM NESTED_JSON;
+---+---+---+
| ID | NAME | DEPT |
+---+---+---+
| NULL | NULL | NULL |
+---+---+---+

```

In this part i got **null** records because of my json path is wrong.After i checked and give right json path it will come like this.

For right query in nested_json:

```
Select emp.value:id::INT AS Employee_ID,
       emp.value:name::STRING AS Employee_Name,
       proj.value:project_name::STRING AS Project_Name,
       proj.value:duration::INT AS Duration
  FROM NESTED_JSON,
       LATERAL FLATTEN(input => raw) emp,          -- first flatten employees
       LATERAL FLATTEN(input => emp.value:projects) proj; -- second flatten projects
```

EMPLOYEE_ID	EMPLOYEE_NAME	PROJECT_NAME	DURATION
101	John Doe	Website Redesign	3
101	John Doe	Employee Portal	4
102	Jane Smith	Financial Report	5
102	Jane Smith	Audit Compliance	2
103	Michael Brown	IT Security	6
103	Michael Brown	Cloud Migration	8
104	Linda White	Marketing Campaign	3
104	Linda White	Social Media Outreach	2
105	James Black	Mobile App Dev	4
105	James Black	Server Optimization	5
106	Emma Green	HR Portal	3
106	Emma Green	Employee Training	2
107	Robert King	Financial Audit	4
107	Robert King	Budget Planning	3
108	Olivia Scott	Marketing Analytics	3
108	Olivia Scott	Campaign Review	2
109	William Lee	Server Maintenance	5
109	William Lee	Network Upgrade	4
110	Sophia Turner	Employee Onboarding	2
110	Sophia Turner	Benefits Portal	3

Output for select query in nested_json

XML-SIMPLE

steps:

- Step-1** **CREATE OR REPLACE TABLE** xml_table (id
 INTEGER AUTOINCREMENT,xml_doc VARIANT);
- Step-2** **create or replace stage** xml_stages;
- Step-3** **put 'file://C:/Users/admin/Downloads/employees.xml' @xml_stages;**

output:

```
1 Row(s) produced. Time Elapsed: 0.817s
ROSHINI#COMPUTE_WH@SIMPLE_XML.PUBLIC>CREATE OR REPLACE TABLE xml_table (id
                  INTEGER AUTOINCREMENT,xml_doc VARIANT)
                  ;
+-----+
| status
| Table XML_TABLE successfully created.
+-----+
1 Row(s) produced. Time Elapsed: 0.869s
ROSHINI#COMPUTE_WH@SIMPLE_XML.PUBLIC>create or replace stage xml_stages;
+-----+
| status
| Stage area XML_STAGES successfully created.
+-----+
1 Row(s) produced. Time Elapsed: 1.165s
ROSHINI#COMPUTE_WH@SIMPLE_XML.PUBLIC>put 'file://C:/Users/admin/Downloads/e
                  mployees.xml' @xml_stages;
+-----+
| source      | target      | source_size | target_size | source_compression | target_compression | status      | message
| employees.xml | employees.xml.gz |      1111 |      304 |      NONE |      GZIP | UPLOADED |
+-----+
```

Step-4 **CREATE OR REPLACE FILE FORMAT** my_xml_format TYPE =
'XML' STRIP_OUTER_ELEMENT = TRUE COMPRESSION = 'AUTO';

Step-5 **Copy into** xml_table(xml_doc)from @xml_stages/employees.xml
file_format = (format_name = 'my_xml_format');

Output:

Step-6 **select *** from xml_table;

```
1 Row(s) produced. Time Elapsed: 3.398s
ROSHINI#COMPUTE_WH@SIMPLE_XML.PUBLIC>select * from xml_table;
+----+-----+
| ID | XML_DOC
+----+-----+
1   | <employee id="101">
    |   <name>John Doe</name>
    |   <department>HR</department>
    |   <salary>5000</salary>
    | </employee>
2   | <employee id="102">
    |   <name>Jane Smith</name>
    |   <department>Finance</department>
    |   <salary>6000</salary>
    | </employee>
3   | <employee id="103">
    |   <name>Michael Brown</name>
    |   <department>IT</department>
    |   <salary>5500</salary>
    | </employee>
4   | <employee id="104">
    |   <name>Linda White</name>
    |   <department>Marketing</department>
    |   <salary>5200</salary>
    | </employee>
5   | <employee id="105">
    |   <name>James Black</name>
    |   <department>IT</department>
    |   <salary>5800</salary>
    | </employee>
6   | <employee id="106">
    |   <name>Emma Green</name>
    |   <department>HR</department>
    |   <salary>5100</salary>
    | </employee>
7   | <employee id="107">
    |   <name>Robert King</name>
    |   <department>Finance</department>
    |   <salary>6200</salary>
    | </employee>
8   | <employee id="108">
    |   <name>Olivia Scott</name>
    |   <department>Marketing</department>
    |   <salary>5300</salary>
    | </employee>
9   | <employee id="109">
    |   <name>William Lee</name>
    |   <department>IT</department>
    |   <salary>5600</salary>
    | </employee>
10  | <employee id="110">
    |   <name>Sophia Turner</name>
    |   <department>HR</department>
    |   <salary>5000</salary>
    | </employee>
```

Step-7 **CREATE OR REPLACE TABLE** EMPLOYEES_FINAL (ID INT,NAME STRING,DEPARTMENT STRING,SALARY INT);

Output:

```
ROSHINI#COMPUTE_WH@SIMPLE_XML.PUBLIC>CREATE OR REPLACE TABLE EMPLOYEES_FINAL (ID INT,NAME STRING,DEPARTMENT STRING,SALARY INT);
+-----+
| status |
+-----+
| Table EMPLOYEES_FINAL successfully created. |
+-----+
```

Step-8 **INSERT INTO** EMPLOYEES_FINAL (ID, NAME, DEPARTMENT, SALARY)

```
SELECT
    XML_DOC:@"@id"::string::int AS ID,
    XMLGET(XML_DOC, 'name'):"$"::string AS NAME,
    XMLGET(XML_DOC,'department'):"$"::string AS
DEPARTMENT,
    XMLGET(XML_DOC,'salary'):"$"::int AS SALARY
FROM XML_TABLE;
```

Step-9 **SELECT * FROM** EMPLOYEES_FINAL;

Output:

```

10 Row(s) produced. Time Elapsed: 5.439s
ROSHINI#COMPUTE_WH@SIMPLE_XML.PUBLIC>SELECT * FROM EMPLOYEES_FINAL;
+-----+-----+-----+-----+
| ID   | NAME    | DEPARTMENT | SALARY |
+-----+-----+-----+-----+
| 101  | John Doe | HR          | 5000   |
| 102  | Jane Smith | Finance    | 6000   |
| 103  | Michael Brown | IT          | 5500   |
| 104  | Linda White | Marketing  | 5200   |
| 105  | James Black | IT          | 5800   |
| 106  | Emma Green | HR          | 5100   |
| 107  | Robert King | Finance    | 6200   |
| 108  | Olivia Scott | Marketing  | 5300   |
| 109  | William Lee | IT          | 5600   |
| 110  | Sophia Turner | HR          | 5000   |

```

XML - NESTED

Steps:

Step-1 `create database nest_xml;`

Step-2 `create schema nest_schema;`

step-3 `CREATE OR REPLACE TABLE xml_table (id INTEGER AUTOINCREMENT,xml_doc VARIANT);`

Step-4 `create or replace stage xml_stage;`

```

ROSHINI#COMPUTE_WH@(no database).(no schema)>create database nest_xml;
+-----+
| status
+-----+
| Database NEST_XML successfully created.
+-----+
1 Row(s) produced. Time Elapsed: 0.260s
ROSHINI#COMPUTE_WH@NEST_XML.PUBLIC>create schema nest_schema;
+-----+
| status
+-----+
| Schema NEST_SCHEMA successfully created.
+-----+
1 Row(s) produced. Time Elapsed: 0.208s
ROSHINI#COMPUTE_WH@NEST_XML.NEST_SCHEMA>CREATE OR REPLACE TABLE xml_table (id INTEGER AUTOINCREMENT,xml_doc VARIANT);
+-----+
| status
+-----+
| Table XML_TABLE successfully created.
+-----+
1 Row(s) produced. Time Elapsed: 0.566s
ROSHINI#COMPUTE_WH@NEST_XML.NEST_SCHEMA>create or replace stage xml_stage;
+-----+
| status
+-----+
| Stage area XML_STAGE successfully created.

```

Step-5 `putfile://C:/Users/admin/Downloads/emp_project.xml' @xml_stage;`

Step-6 `CREATE OR REPLACE FILE FORMAT my_xml_format TYPE = 'XML' STRIP_OUTER_ELEMENT = TRUE COMPRESSION = 'AUTO';`

Step-7 Copy into xml_table(xml_doc)from @xml_stage/emp_project.xml file_format = (format_name = 'my_xml_format');

```
ROSHINI#COMPUTE_WH@NEST_XML.NEST_SCHEMA>put'file:///C:/Users/admin/Downloads/emp_project.xml' @xml_stage;
+-----+
| source | target | source_size | target_size | source_compression | target_compression | status | message |
+-----+
| emp_project.xml | emp_project.xml.gz | 2695 | 544 | NONE | GZIP | UPLOADED |          |
+-----+
1 Row(s) produced. Time Elapsed: 2.756s
ROSHINI#COMPUTE_WH@NEST_XML.NEST_SCHEMA>CREATE OR REPLACE FILE FORMAT my_xml_format TYPE = XML STRIP_OUTER_ELEMENT = TRUE COMPRESSION = 'AUTO';
+-----+
| status |
+-----+
| File format MY_XML_FORMAT successfully created. |
+-----+
1 Row(s) produced. Time Elapsed: 0.739s
ROSHINI#COMPUTE_WH@NEST_XML.NEST_SCHEMA>Copy into xml_table(xml_doc)from @xml_stage/emp_project.xml file_format = (format_name = 'my_xml_format');
+-----+
| file | status | rows_parsed | rows_loaded | error_limit | errors_seen | first_error | first_error_line | first_error_character | f |
| inst_error_column_name |          |          |          |          |          |          |          |          |          |
+-----+
| xml_stage/emp_project.xml.gz | LOADED |      10 |       10 |        1 |         0 | NULL |          NULL |          NULL | N |
+-----+
```

Step-8 select*from xml_table;

```
ROSHINI#COMPUTE_WH@NEST_XML.NEST_SCHEMA>select*from xml_table;
+-----+
| ID | XML_DOC |
+-----+
| 1  | <employee id="101">
    |   <name>John Doe</name>
    |   <projects>
    |     <project>
    |       <project_name>Website Redesign</project_name>
    |       <duration>3</duration>
    |     </project>
    |     <project>
    |       <project_name>Employee Portal</project_name>
    |       <duration>4</duration>
    |     </project>
    |   </projects>
    | </employee>
| 2  | <employee id="102">
    |   <name>Jane Smith</name>
    |   <projects>
    |     <project>
    |       <project_name>Financial Report</project_name>
    |       <duration>5</duration>
    |     </project>
    |     <project>
    |       <project_name>Audit Compliance</project_name>
    |       <duration>2</duration>
    |     </project>
    |   </projects>
    | </employee>
| 3  | <employee id="103">
    |   <name>Michael Brown</name>
    |   <projects>
    |     <project>
    |       <project_name>IT Security</project_name>
    |       <duration>6</duration>
    |     </project>
    |     <project>
    |       <project_name>Cloud Migration</project_name>
    |     </project>
+-----+
```

Step-9 `create or replace table employee_projects_final(emp_id int,emp_name string,project_name string,duration int);`

```
10 Row(s) produced. Time Elapsed: 1.500s
ROSHINI#COMPUTE_WH@NEST_XML.NEST_SCHEMA>create or replace table employee_projects_final(emp_id int,emp_name string,project_name string,duration int);
+-----+
| status |
+-----+
| Table EMPLOYEE_PROJECTS_FINAL successfully created. |
```

Step-10 `INSERT INTO EMPLOYEE_PROJECTS_FINAL (EMP_ID, EMP_NAME, PROJECT_NAME, DURATION)`

```
SELECT XML_DOC:"@id":int AS EMP_ID,
       XMLGET(XML_DOC,'name'):$string AS EMP_NAME,
       XMLGET(XML_DOC,'project_name'):$string AS
PROJECT_NAME,
       XMLGET(XML_DOC, 'duration'):$int AS DURATION FROM
XML_TABLE;
```

Step-11 `Select*from EMPLOYEE_PROJECTS_FINAL;`

```
ROSHINI#COMPUTE_WH@NEST_XML.NEST_SCHEMA>INSERT INTO EMPLOYEE_PROJECTS_FINAL
(EMP_ID, EMP_NAME, PROJECT_NAME, DURATION)
SELECT
XML_DOC:"@id":int AS EMP_ID,
XMLGET(XML_DOC,'name'):$string
AS EMP_NAME,
XMLGET(XML_DOC,'project_name'):$string
AS PROJECT_NAME,
XMLGET(XML_DOC, 'duration'):$int
AS DURATION
FROM XML_TABLE;
+-----+
| number of rows inserted |
+-----+
| 10 |
+-----+
10 Row(s) produced. Time Elapsed: 0.525s
ROSHINI#COMPUTE_WH@NEST_XML.NEST_SCHEMA>select*from EMPLOYEE_PROJECTS_FINAL
;
+-----+
| EMP_ID | EMP_NAME      | PROJECT_NAME | DURATION |
+-----+
| 101   | John Doe      | NULL        | NULL     |
| 102   | Jane Smith    | NULL        | NULL     |
| 103   | Michael Brown | NULL        | NULL     |
| 104   | Linda White   | NULL        | NULL     |
| 105   | James Black   | NULL        | NULL     |
| 106   | Emma Green    | NULL        | NULL     |
| 107   | Robert King   | NULL        | NULL     |
| 108   | Olivia Scott  | NULL        | NULL     |
| 109   | William Lee   | NULL        | NULL     |
| 110   | Sophia Turner | NULL        | NULL     |
+-----+
```

Through i tried lateral flatten i got same null values.

```
10 Row(s) produced. Time Elapsed: 0.670s
ROSHINI#COMPUTE_WH@NEST_DATA.NESTSCHEMA>INSERT INTO EMPLOYEE_PROJECTS_FINAL (EMP_ID, EMP_NAME, PROJECT_NAME, DURATION)
SELECT
    XML_DOC:@id::int AS EMP_ID,
    XMLGET(XML_DOC, 'name'):$::string AS EMP_NAME,
    XMLGET(project.value, 'project_name'):$::string AS PROJECT_NAME,
    XMLGET(project.value, 'duration'):$::int AS DURATION
FROM XML_TABLE,
    LATERAL FLATTEN(input => XML_DOC:projects.project) project;

+-----+
| number of rows inserted |
+-----+
|          0 |
+-----+
0 Row(s) produced. Time Elapsed: 1.343s
ROSHINI#COMPUTE_WH@NEST_DATA.NESTSCHEMA>select *from EMPLOYEE_PROJECTS_FINAL;
+-----+-----+-----+-----+
| EMP_ID | EMP_NAME | PROJECT_NAME | DURATION |
+-----+-----+-----+-----+
| 101 | John Doe | NULL | NULL |
| 102 | Jane Smith | NULL | NULL |
| 103 | Michael Brown | NULL | NULL |
| 104 | Linda White | NULL | NULL |
| 105 | James Black | NULL | NULL |
| 106 | Emma Green | NULL | NULL |
| 107 | Robert King | NULL | NULL |
| 108 | Olivia Scott | NULL | NULL |
| 109 | William Lee | NULL | NULL |
| 110 | Sophia Turner | NULL | NULL |
+-----+-----+-----+-----+
```