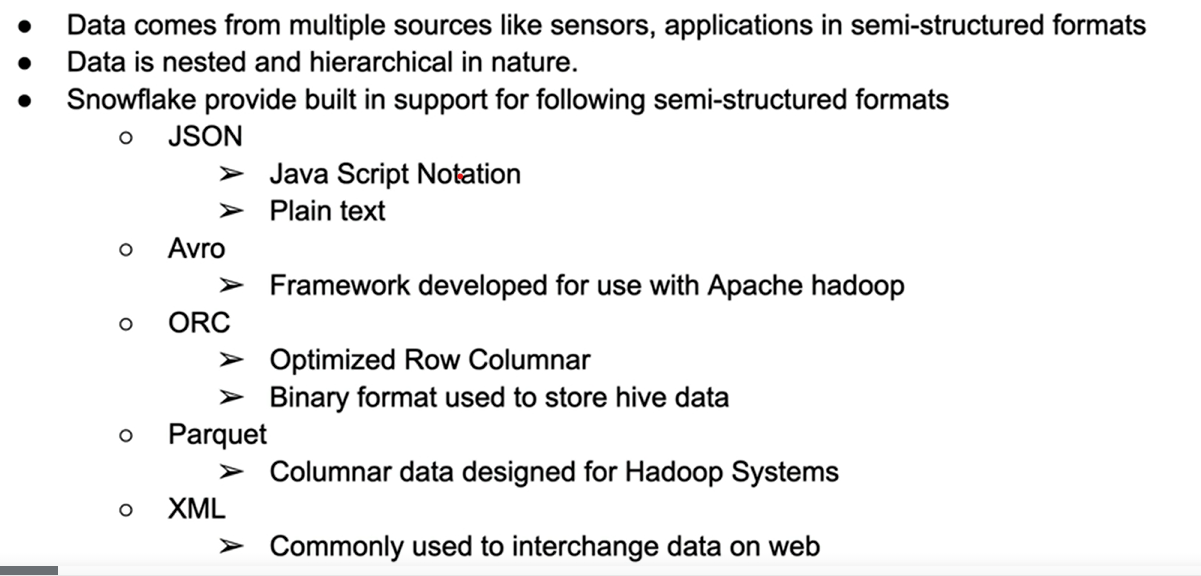
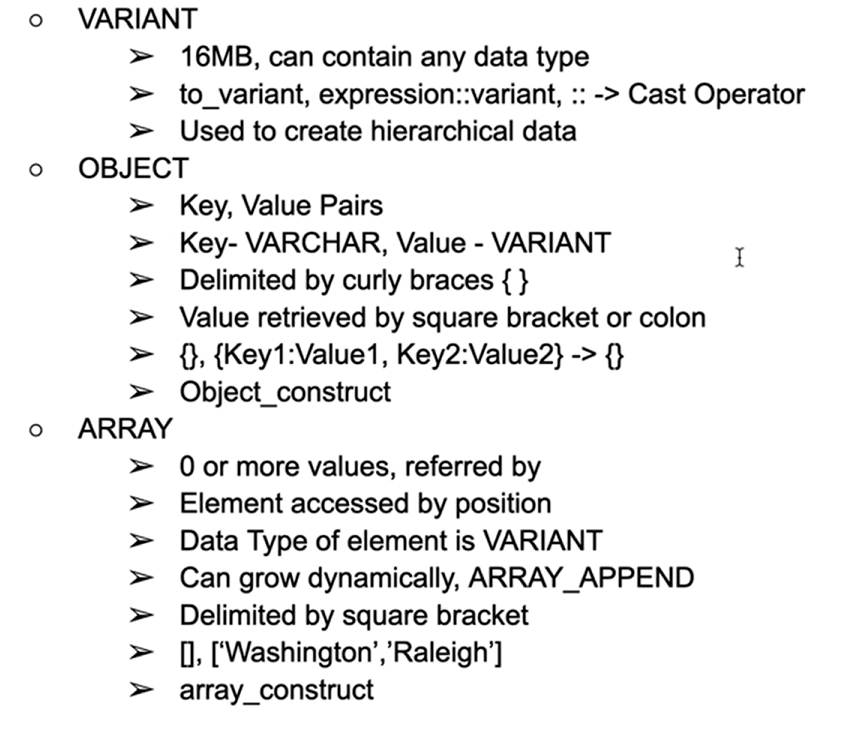
**LOADING SEMI STRUCTURED DATA**

**Semi Structured Data:**



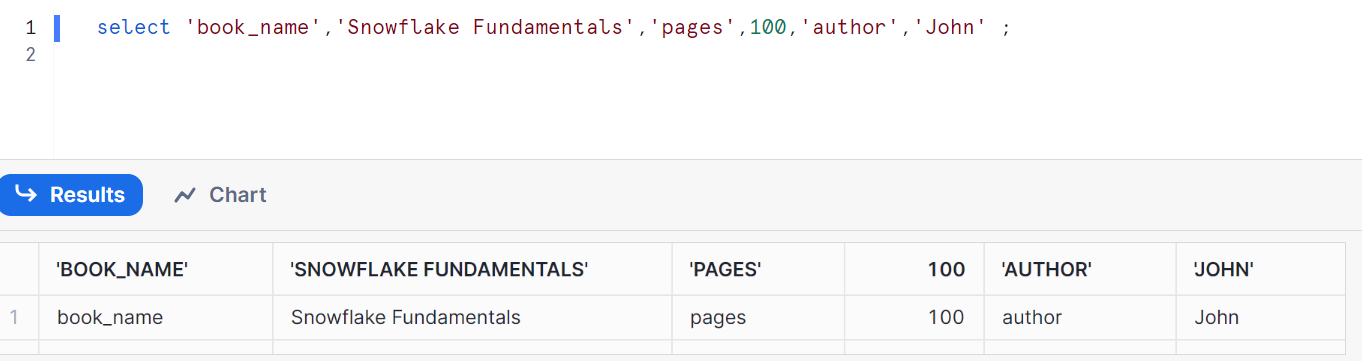
**Data Types to support semi –structured data:**



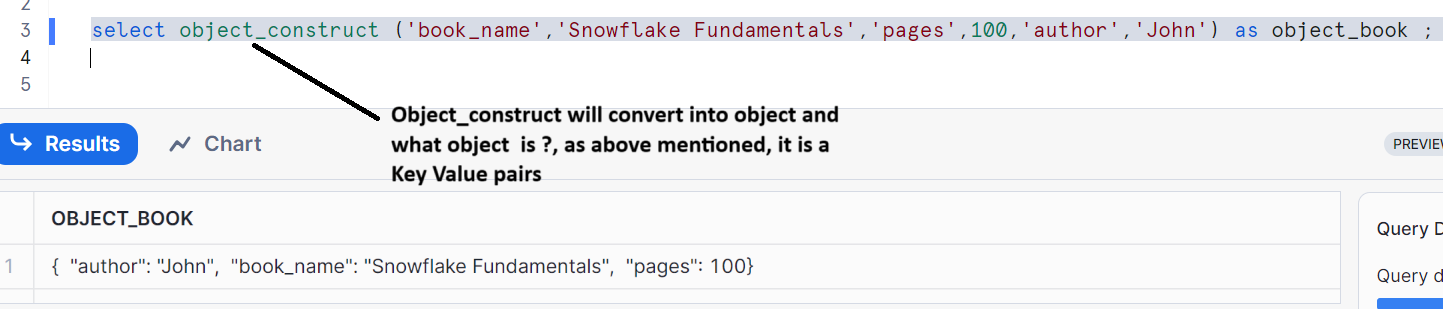
Let’s do some hands on in snowflake:

Plain text data

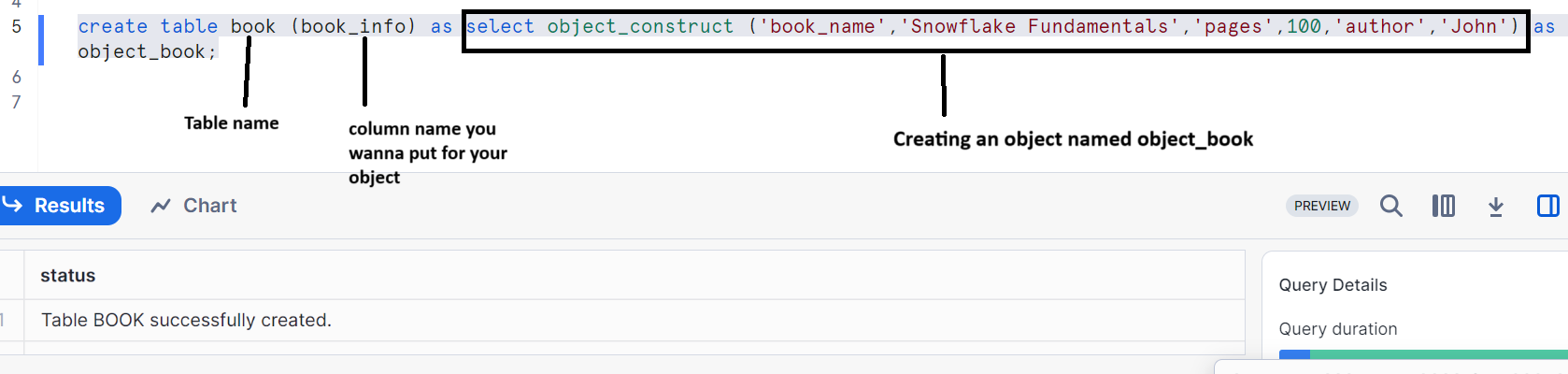
You get an organized tabular format when you give a key & value pair like the below:

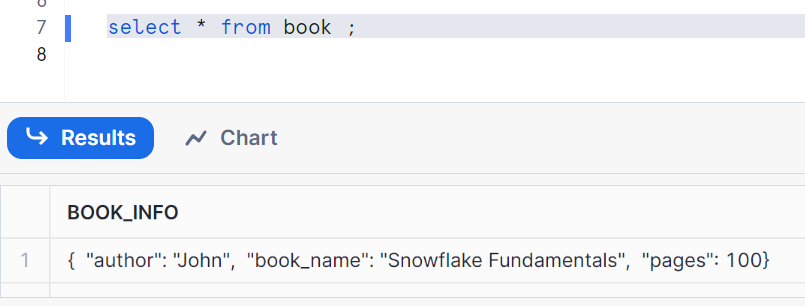


Constructs:

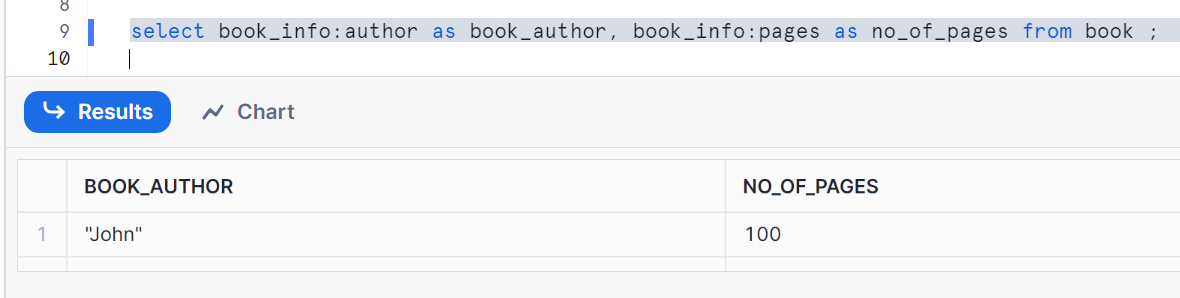


Let’s create a table for object

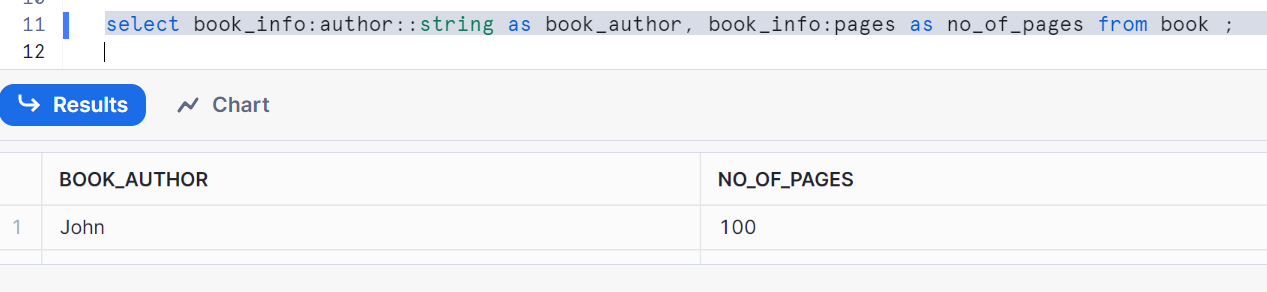




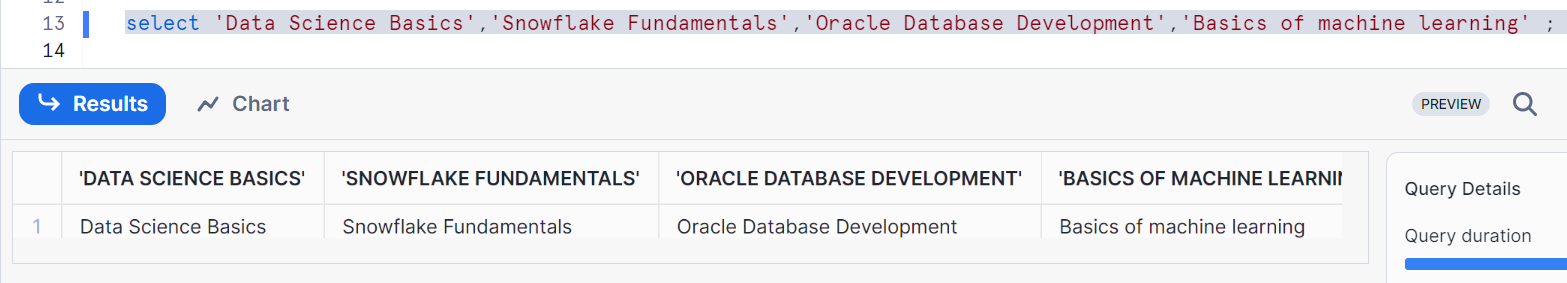
From your column name in your table, pick whatever you want pick from your object



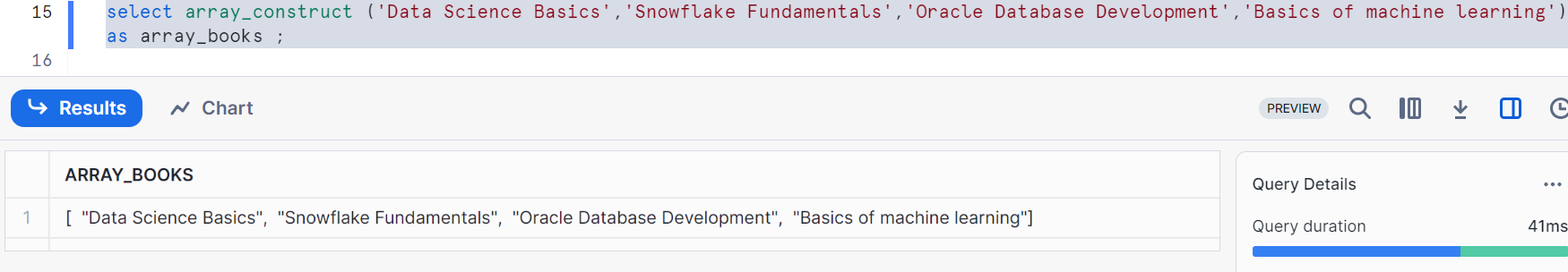
As we can see above that john is coming in double quotes. Hence in order to remove, we convert into string



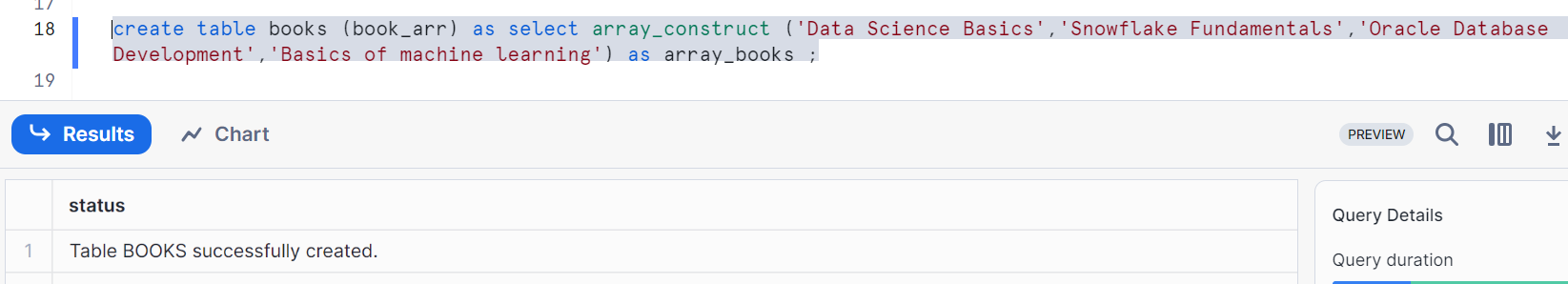
Let’s do the same for Arrays with an example

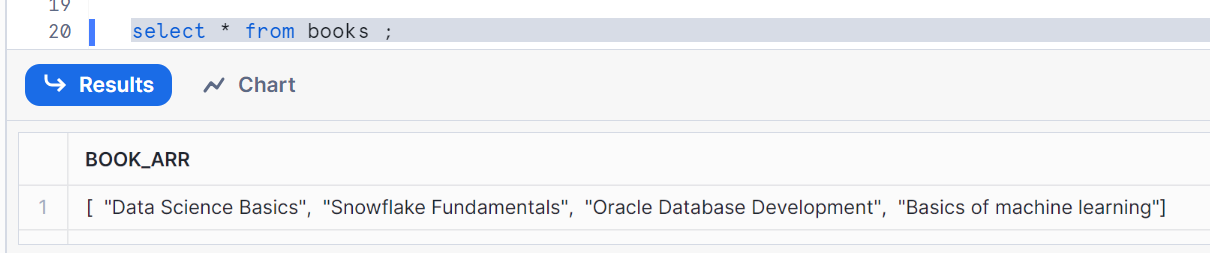


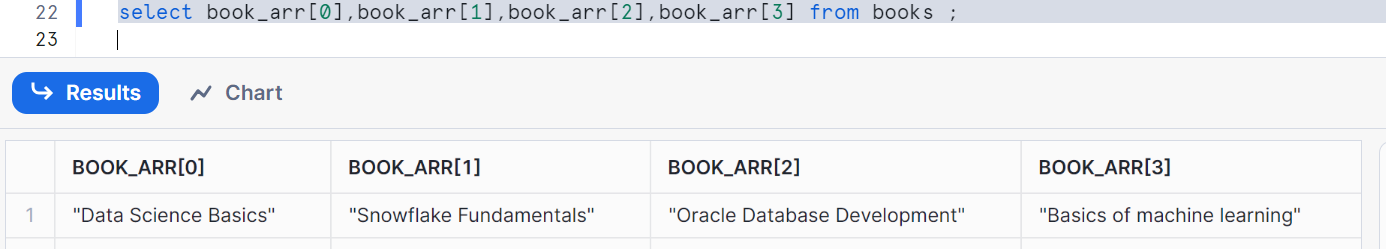
Array\_construct will convert into array (list of elements)

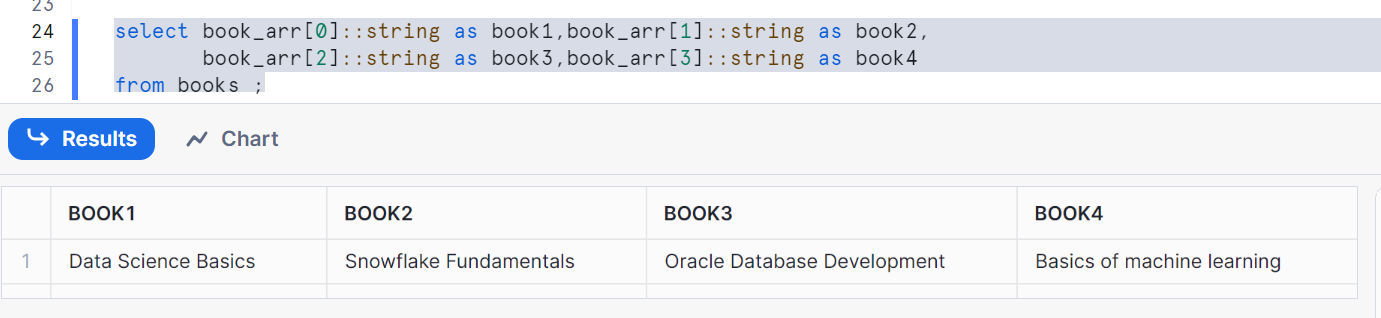


Creating a table







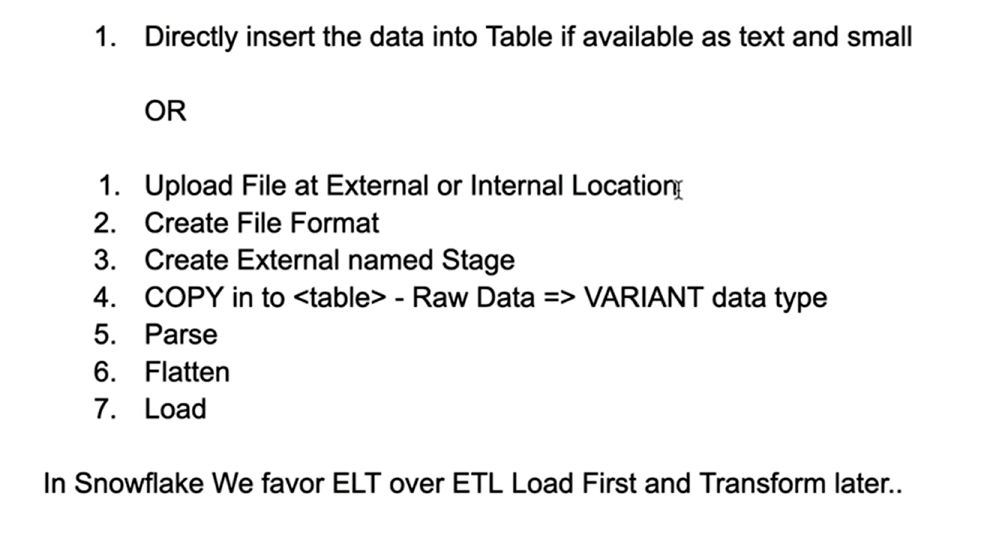


Cleanup:

Drop table book;

Drop table books;

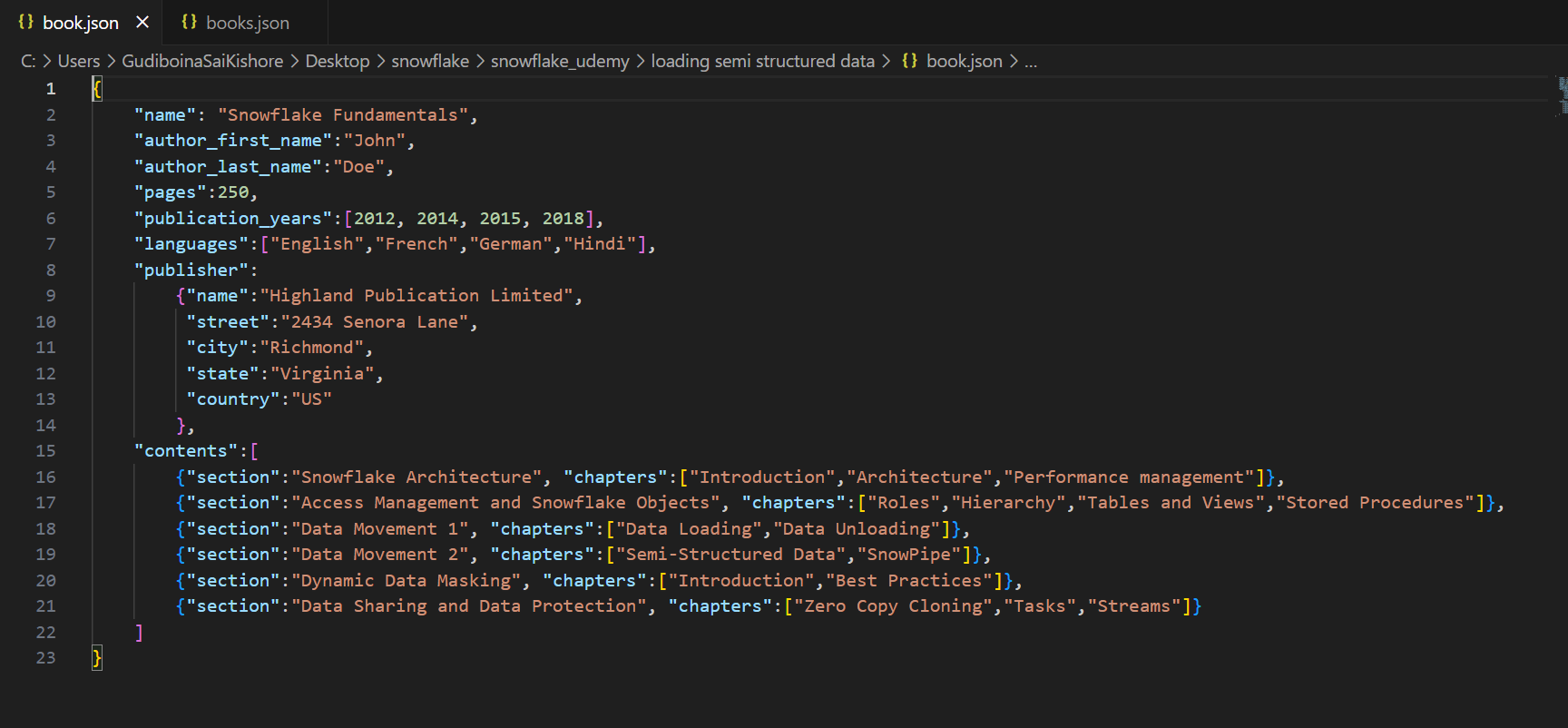
**When we deal with semi-structured data loading, we have to do following**



Let’s do hands on

First let us check the JSON data

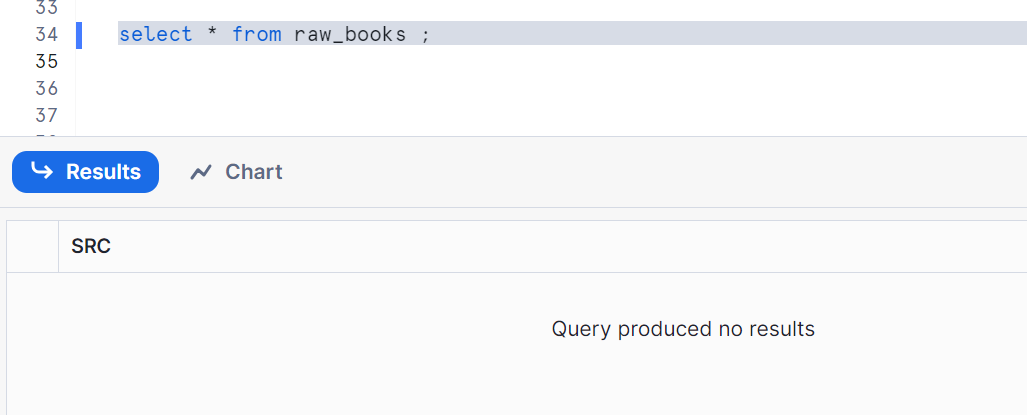
Here is my data file named book.json



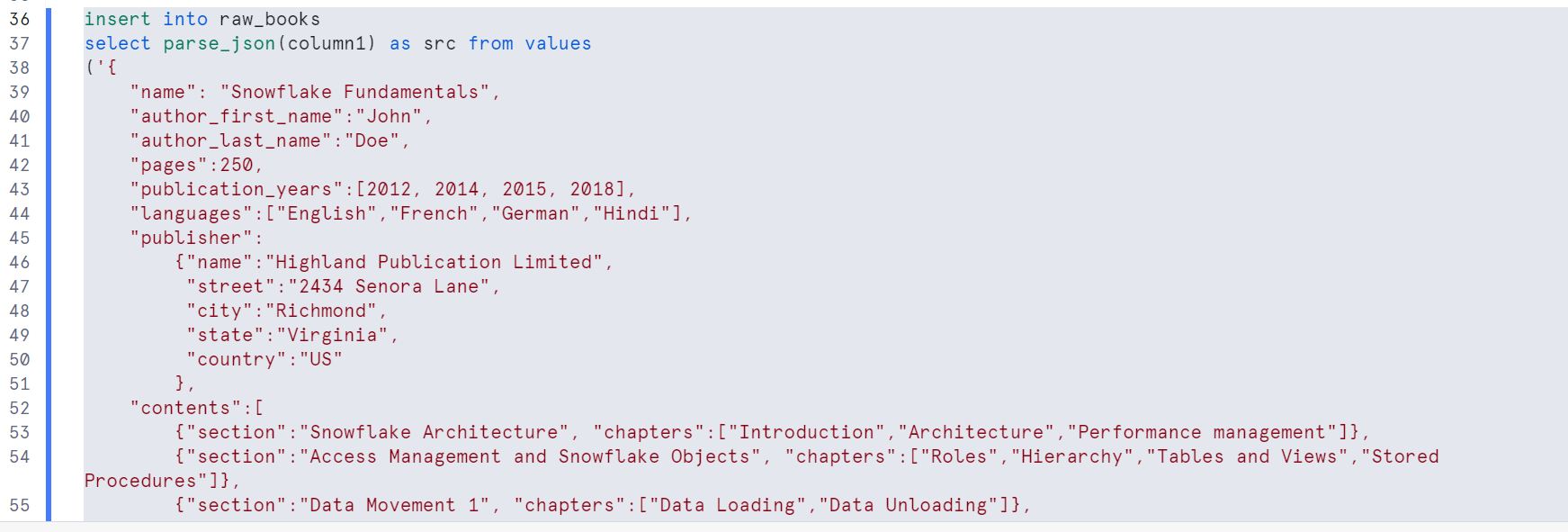
Create table to load the **JSON text data (note: as we are doing it manually for JSON text data to load)**

Src is a column name with the data type variant

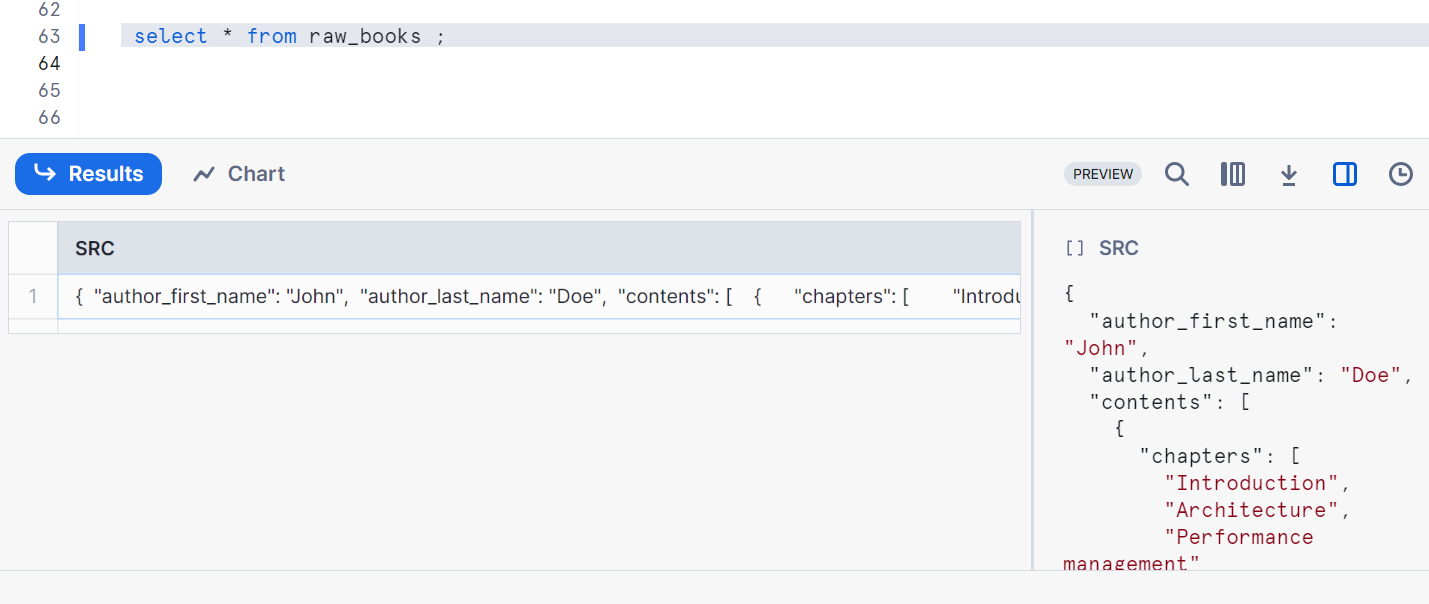




Insert data using parse\_json



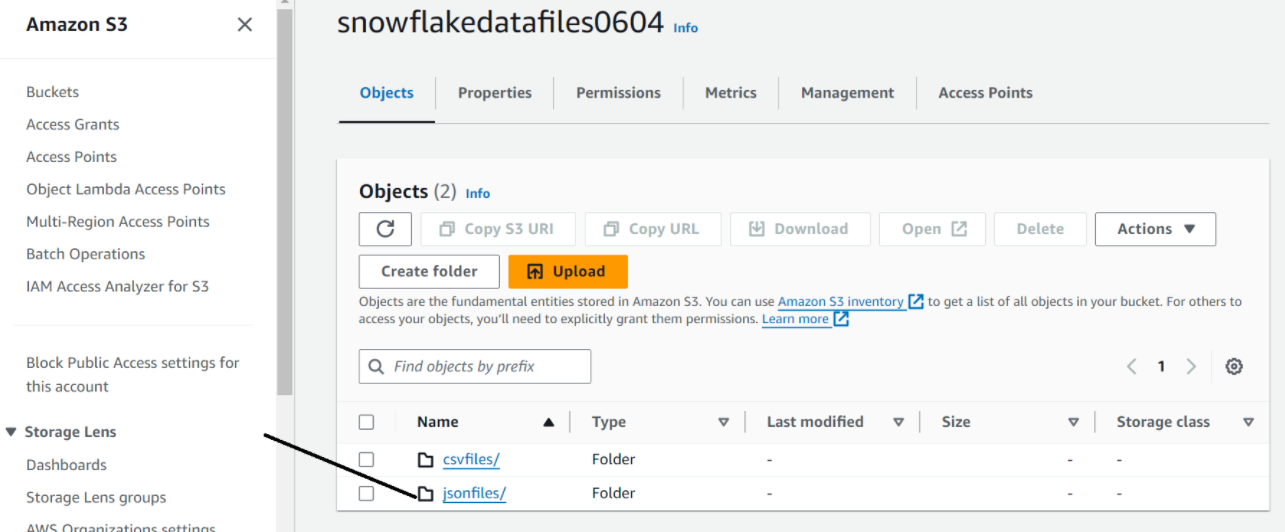
Field names will be in sorted order as you see in below image



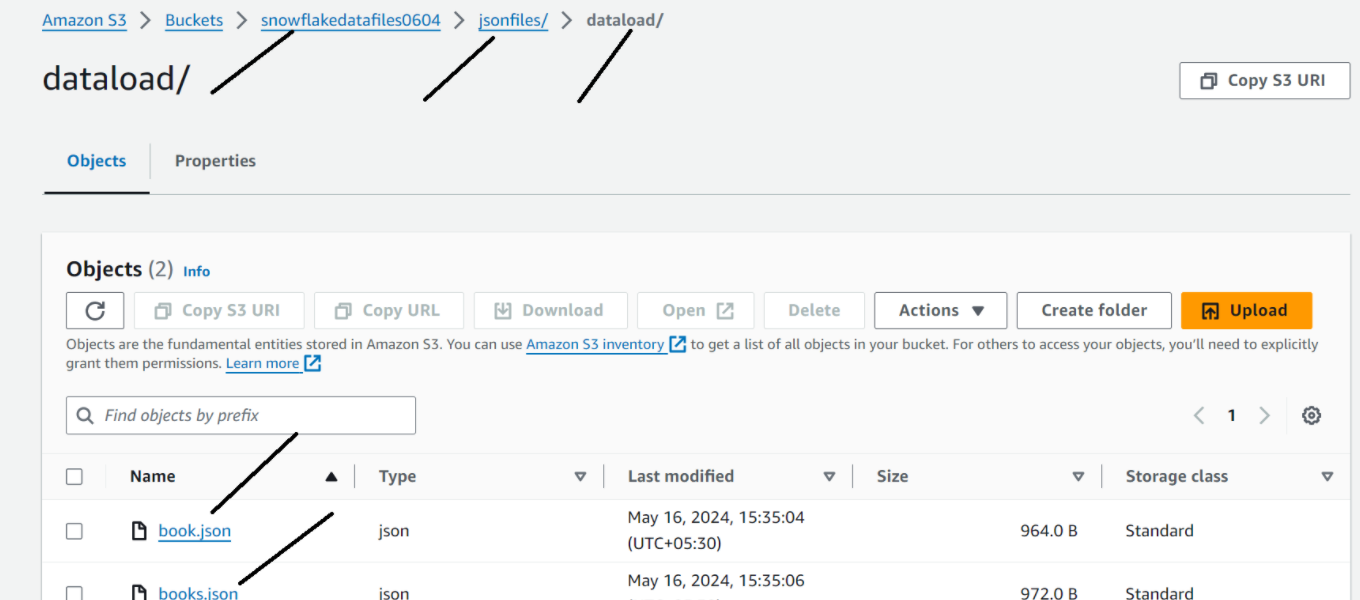
Now let’s see how to load json file data from external location

Create an external file location i.e. in AWS

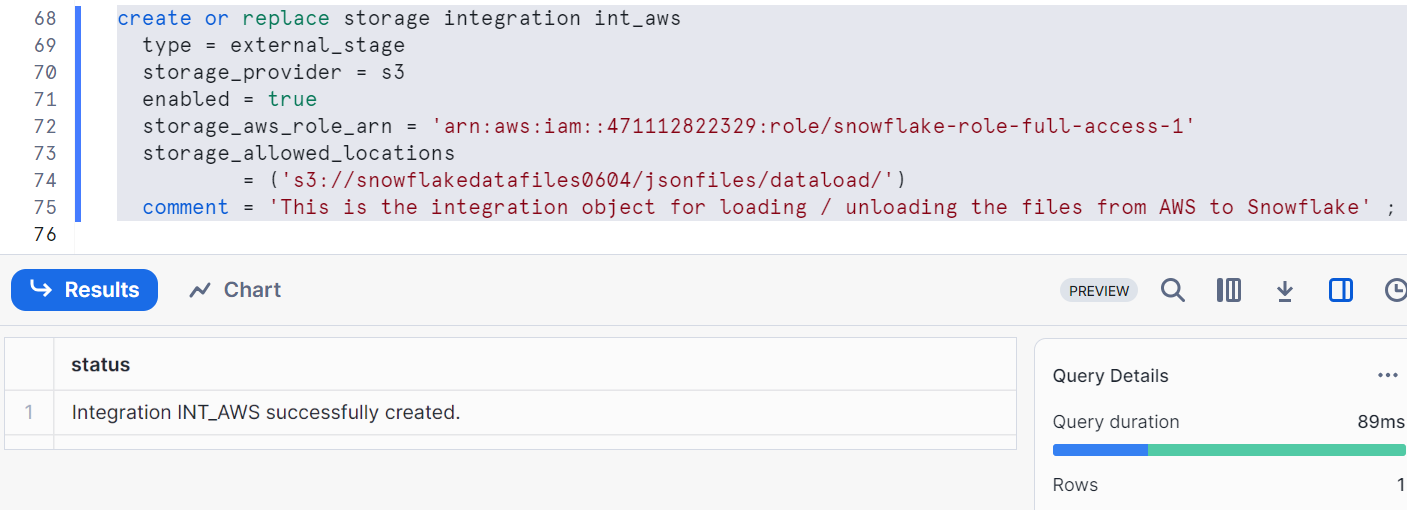
Please refer latest post snowflake\_23 (I explained it in detail)



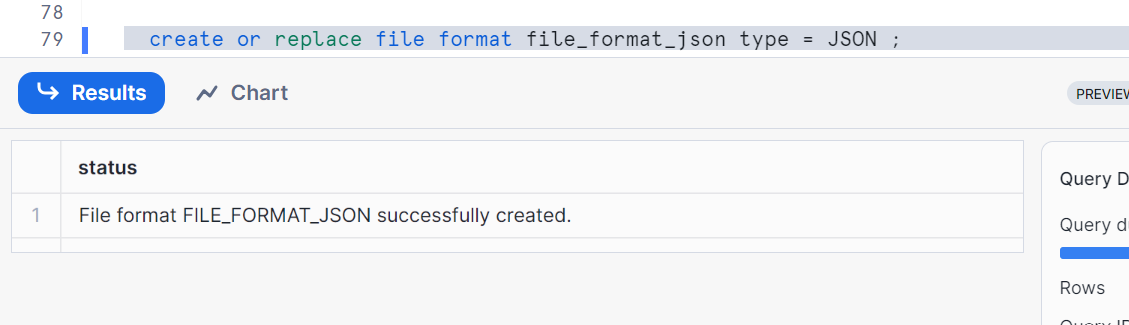
Create a folder dataload in it and add files in it

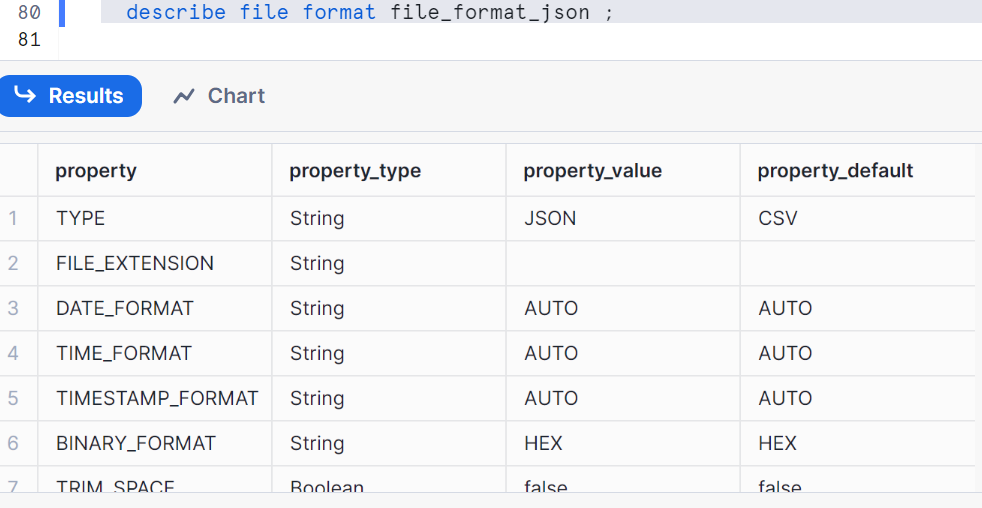


Create integration object

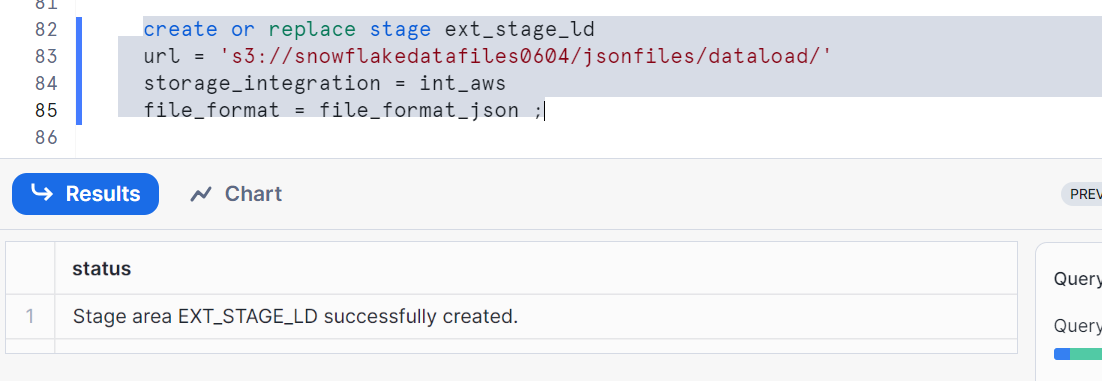


Creating file format





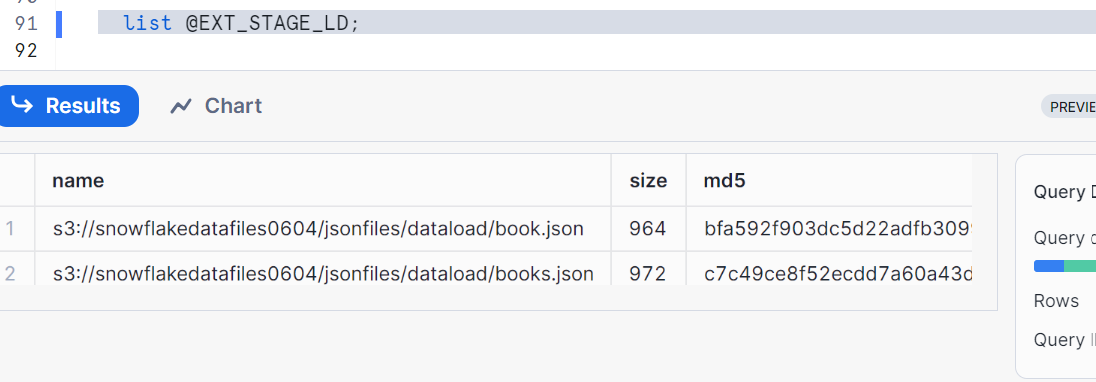
Create external stage for loading data

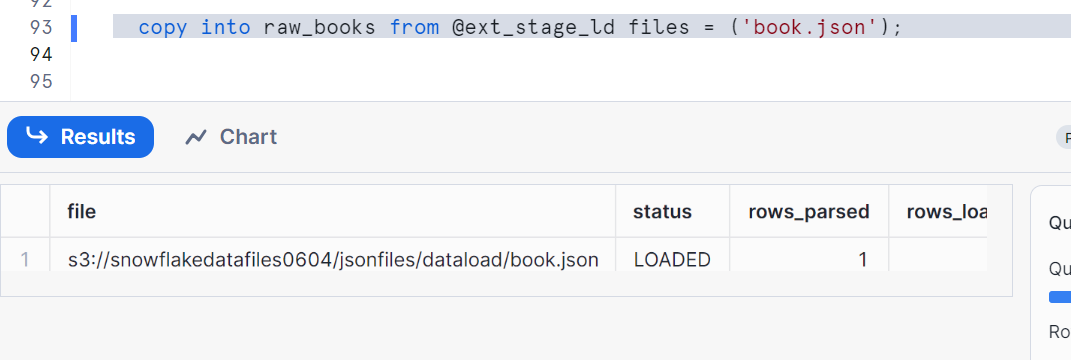


In AWS, edit the trust policy in your IAMrole









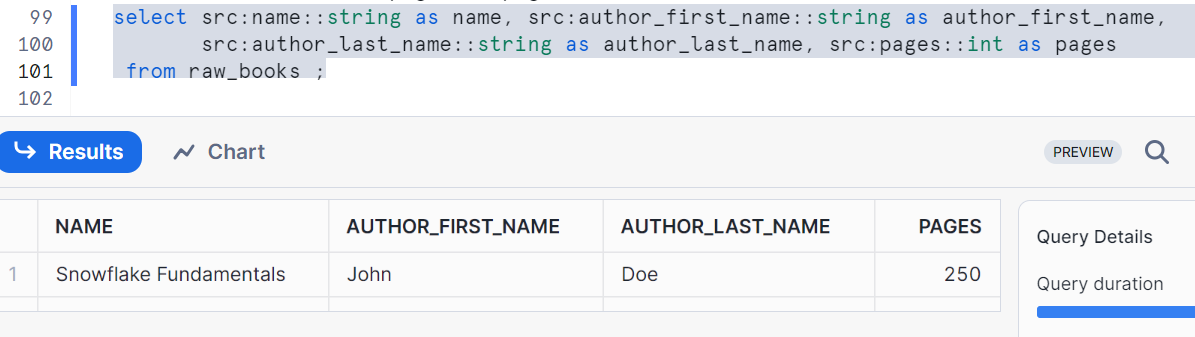




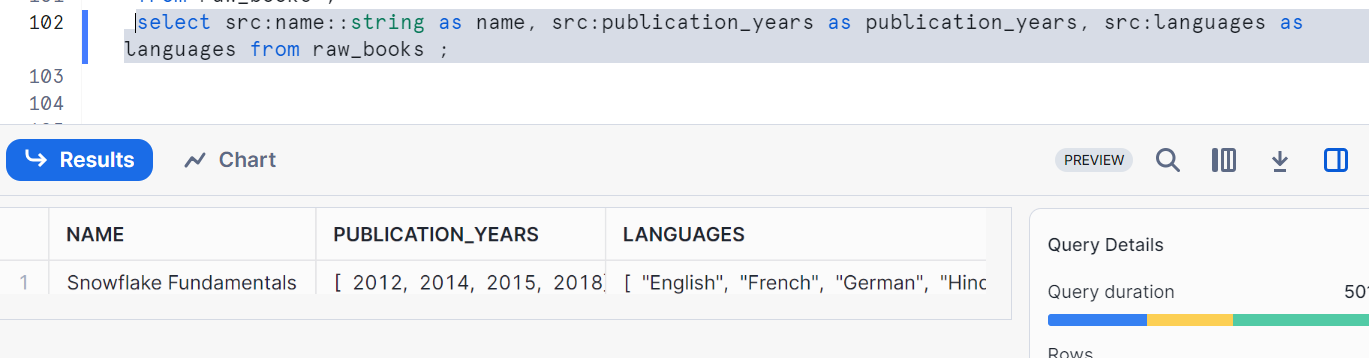
Parsing the JSON data

Select specific columns





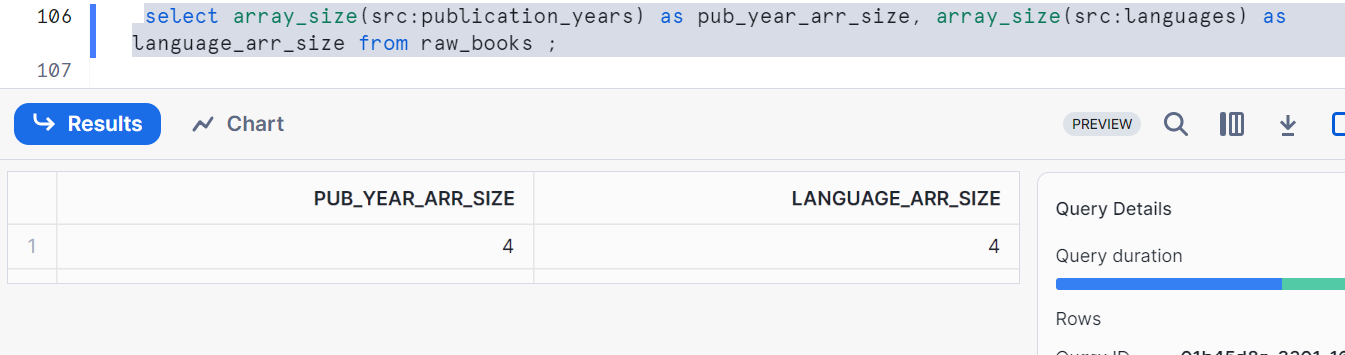
Handling Array - selecting the string array and number array



Selecting first element of Array



ARRAY\_SIZE - to check the length of array and do operations traversing in loop



**Flatten function in snowflake:**

It explodes the compound values into the multiple rows and it has been provided by snowflake to handle the semi structured data

It accepts variant, object and array column and produces a lateral view

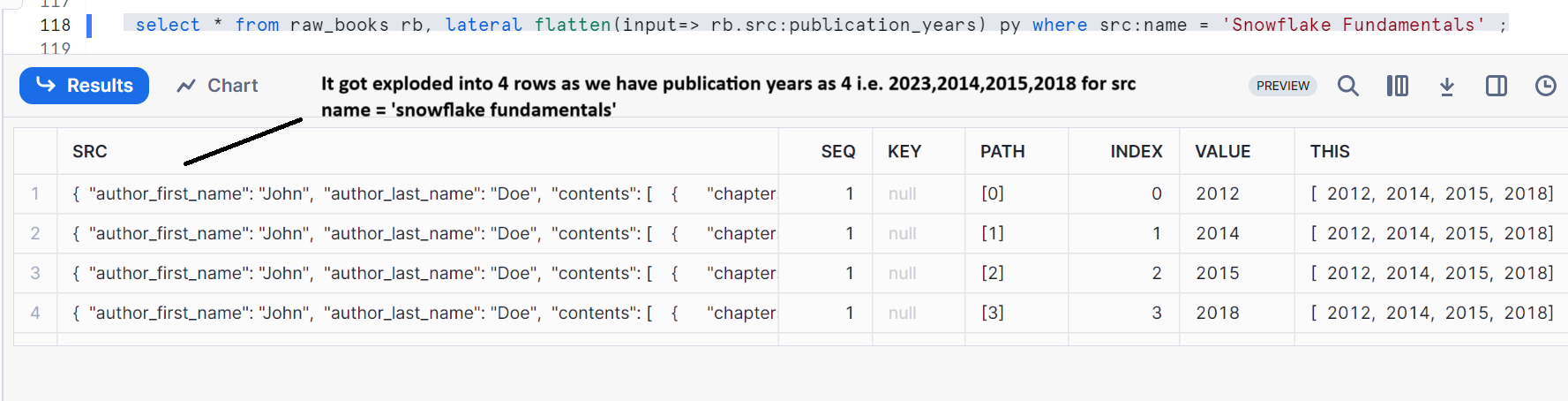
Used for converting semi-structured to relational data

Input is expression -> can be object, array or variant. This will be converted to rows and will be a mandatory parameter

Output is - (sequence, key, path, index, value,this) i.e. (Generated number, key of exploded expression, path of the element, index if array, value, element)



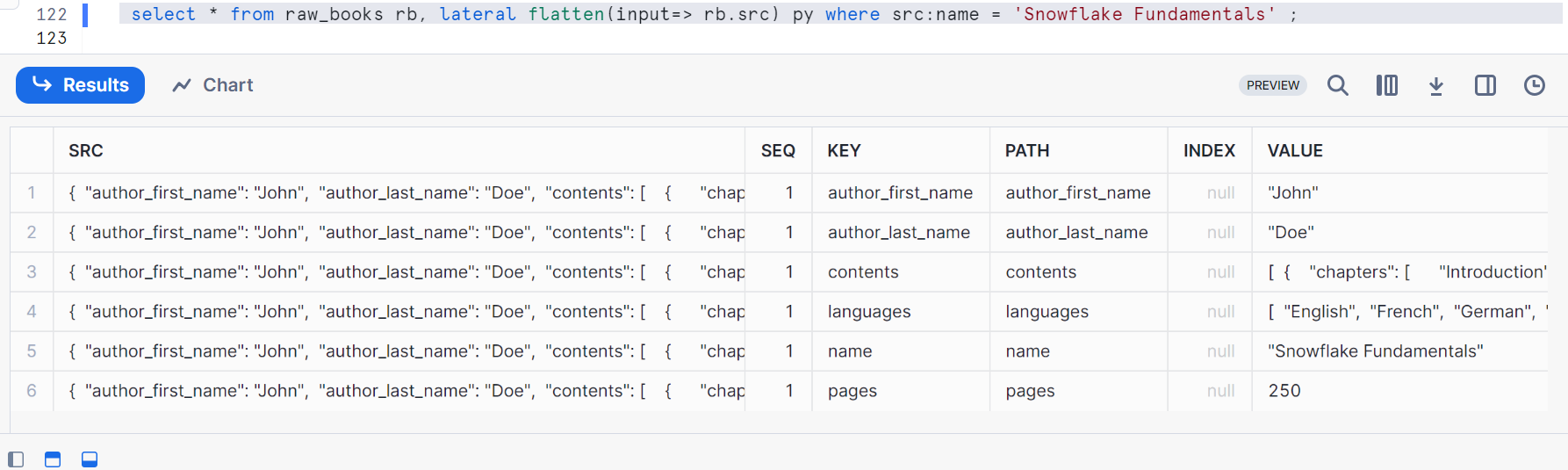
Now let’s see the example of publication\_years by using flatten function



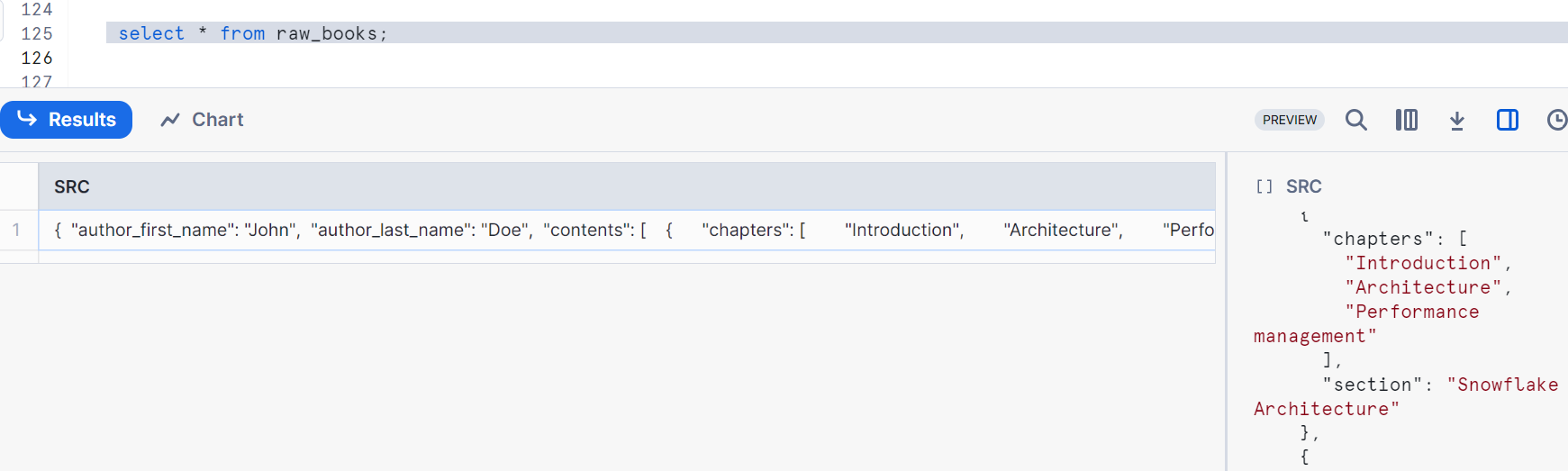
Now let’s pass object into flatten function and observe



Now let’s pass variant into flatten function and observe



Now let’s load all the Json data into table format in snowflake

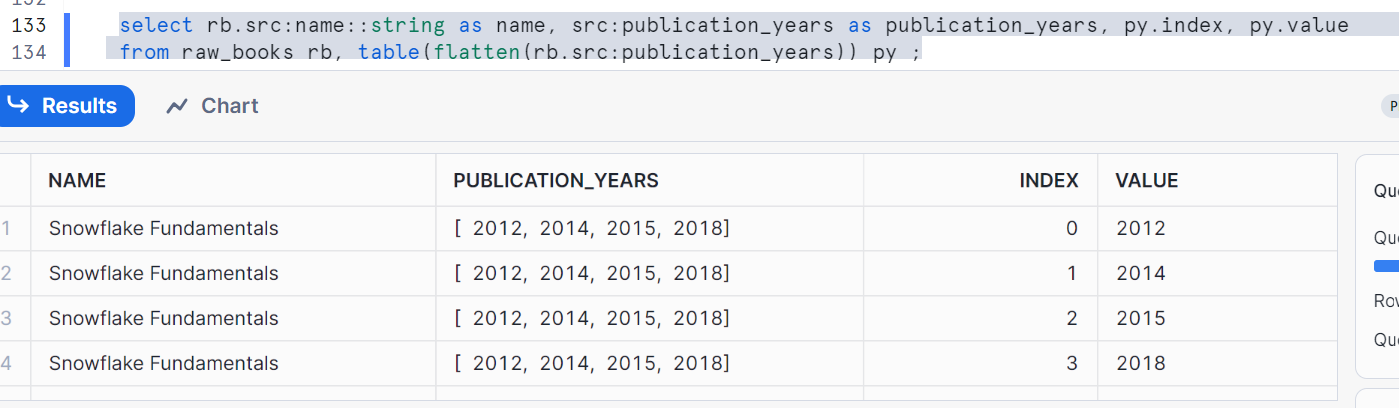


As it is not possible to show the complete output, I am showing the file data again which table raw\_books contains



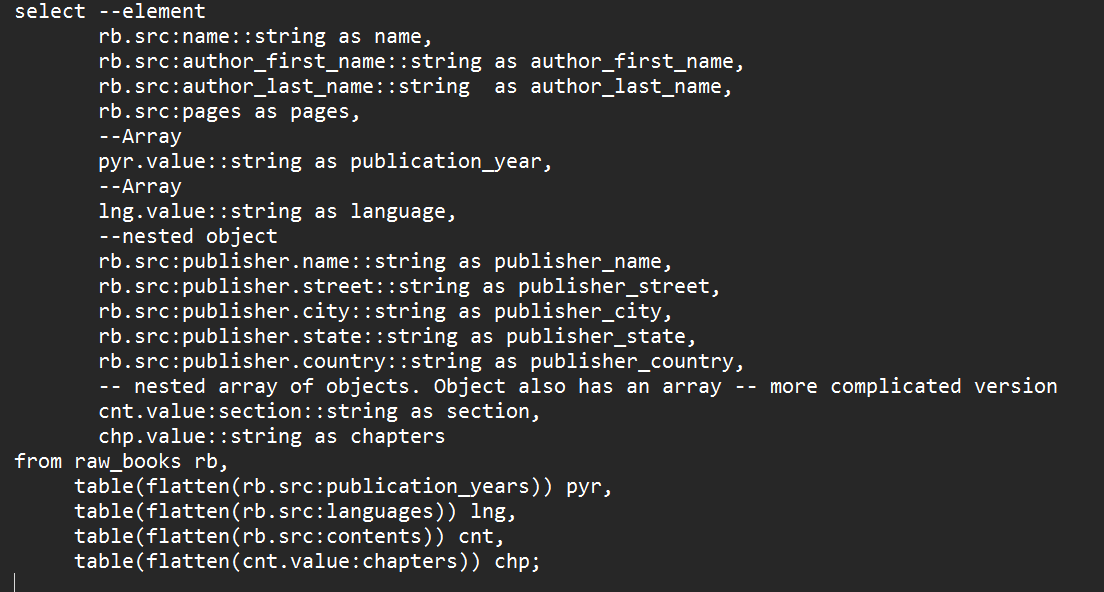


You can use as table as well if you want



Now let’s write the syntax to convert entire JSON data into table, please follow step by step & understand it with respect to input JSON data

Run it in SF

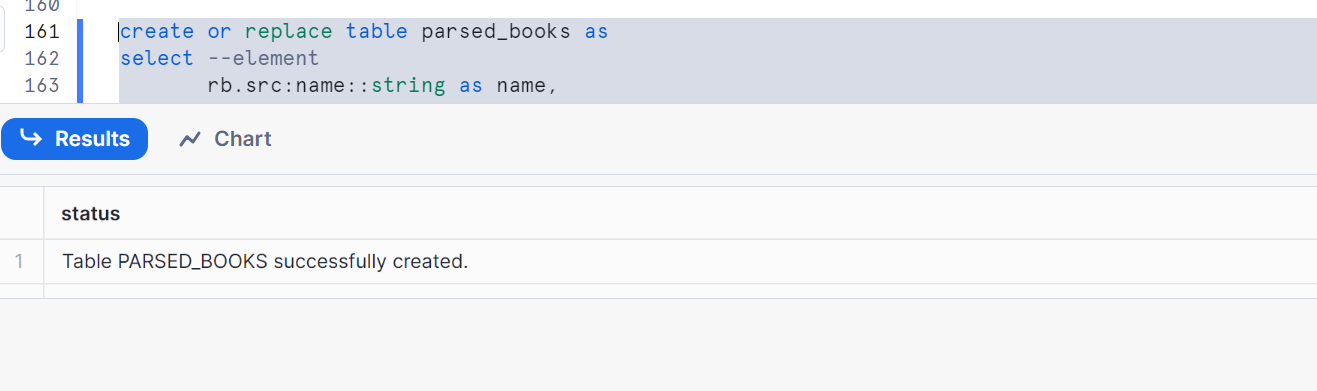


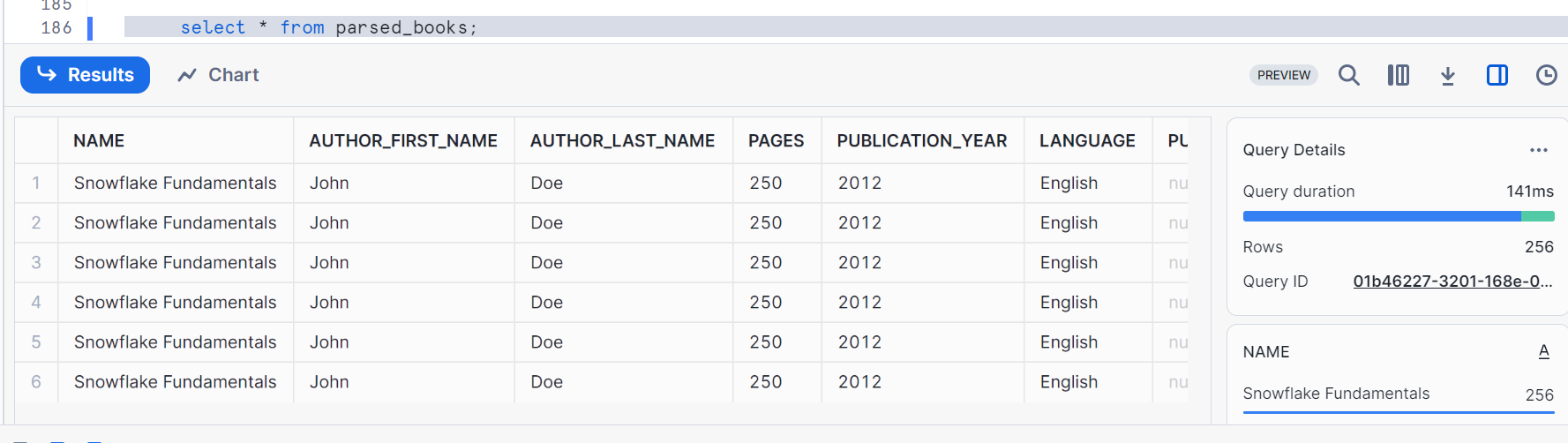
The output is as follows where I had downloaded the output in Snowflake and opened in excel to show you



Therefore entire Json is been converted to tabular format in snowflake with appropriate column names

Now lets create a table by using the above query





Now query as you wanted



Clean up:

Drop table raw\_books;

Drop table parsed\_books;

Drop stage ext\_stage\_ld;

Drop storage integration int\_aws;

Drop file format file\_format\_json;