



# **VIRTUAL HANDS-ON LAB**

## **ADVANCE YOUR ANALYTICS WITH SNOWFLAKE AND TABLEAU**

To participate in the virtual hands-on lab please login to your Snowflake Account and Open Tableau Desktop.

# HANDS-ON LAB SET UP

- Sign-in to your Snowflake free trial account. URL looks something like:  
<https://XY12345.snowflakecomputing.com>
- If you have not already signed up for a free trial account, sign up here:
  - <https://trial.snowflake.com>
  - Please select the region closest to you and the Enterprise Edition
- Open Tableau Desktop
- Install Tableau ODBC Drivers for Snowflake
  - <https://sfc-repo.snowflakecomputing.com/odbc/index.html>
- Download files from On24 resource section
  - SQL script - .sql file - Required



# MEET TODAY'S HANDS-ON LAB LEADERS



David Spezia  
Sales Engineer



Chris Richardson  
Sales Engineer



Jeremy Patoc  
Sales Engineer



# WHAT WE'LL COVER TODAY

- > **Snowflake Query History**
- > **Load Tableau Server JSON data into cloud storage and then perform SQL queries directly with Snowflake**
- > **Load data into Snowflake**
- > **Prepare and optimize data for analytics**
- > **Configure Tableau and connect to Snowflake**
- > **Use Tableau to query semi-structured data, tables, views, and aggregations**
- > **Q&A**



**LET'S DIVE IN**



# Sign in to Your Snowflake Account



Log in to Snowflake

User Name

Password

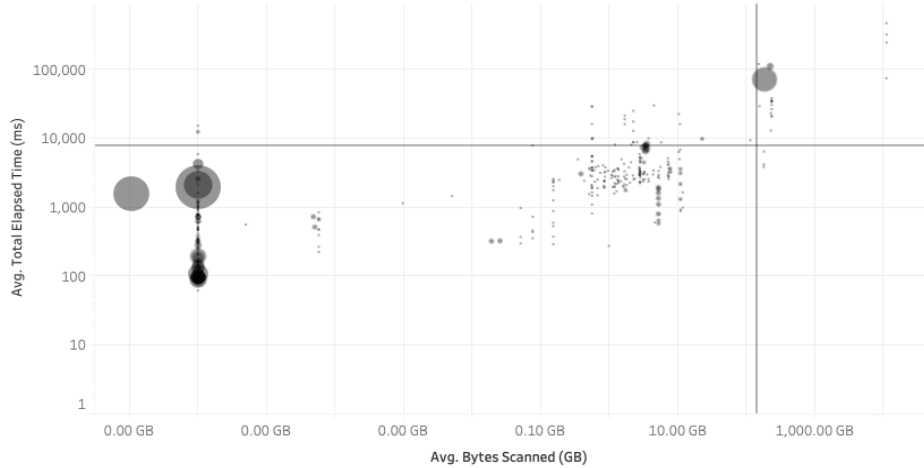
Log In

# Section I

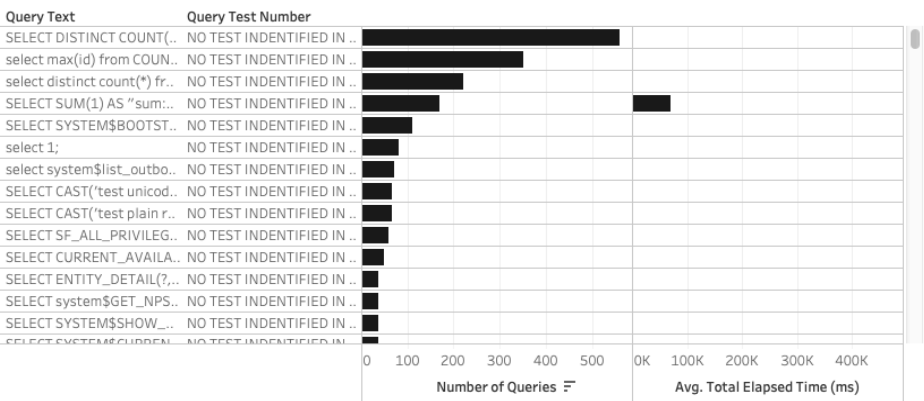
## Query History



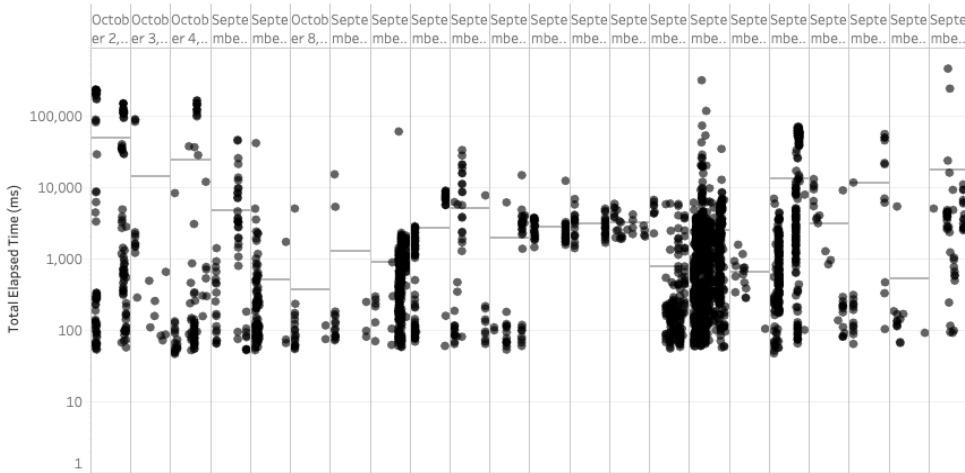
SQT by Elapsed & Bytes - *click to filter*



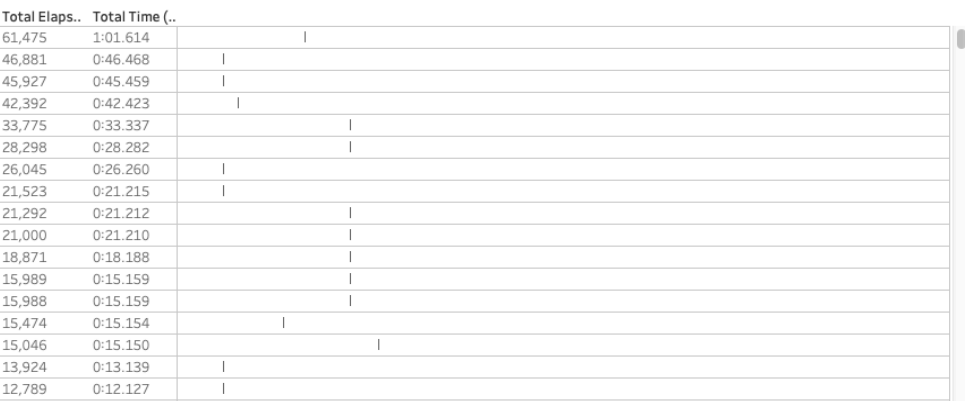
SQT Number & Elapsed- *click to filter*



SQID Over Time



SQID Gantt





# Section II

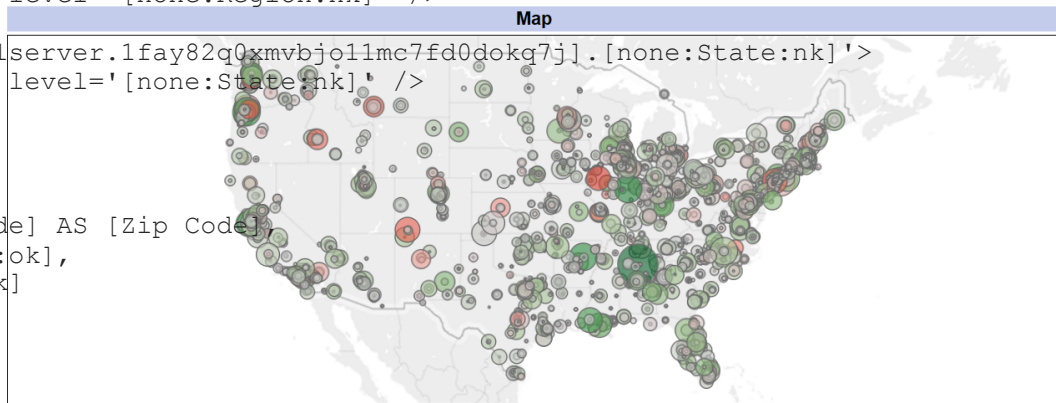
# Tableau Server JSON



# Query from UX

```
cache-hit: none
elapsed: 0.266
owner-component: DataInterpreter
owner-dashboard: Dashboard
owner-worksheet: Map
protocol-id: 3
query-abstract: <query aggregation='true' caching='normal' hierarchize-rows='false' include-empty='false'>
  <fields>
    <field column='[sqlserver.1fay82q0xmvbjo11mc7fd0dokq7j].[none:Zip Code:ok]' />
    <output column='[sqlserver.1fay82q0xmvbjo11mc7fd0dokq7j].[none:Zip Code:ok]' />
    <output column='[sqlserver.1fay82q0xmvbjo11mc7fd0dokq7j].[sum:Profit:qk]' />
    <output column='[sqlserver.1fay82q0xmvbjo11mc7fd0dokq7j].[sum:Sales:qk]' />
    <filter class='categorical' column='[sqlserver.1fay82q0xmvbjo11mc7fd0dokq7j].[none:Region:nk]'>
      <groupfilter function='level-members' level='[none:Region:nk]' />
    </filter>
    <filter class='categorical' column='[sqlserver.1fay82q0xmvbjo11mc7fd0dokq7j].[none:State:nk]'>
      <groupfilter function='level-members' level='[none:State:nk]' />
    </filter>
  </fields>
</query>

query-compiled: SELECT [tbl_Orders].[Zip Code] AS [Zip Code],
  SUM([tbl_Orders].[Profit]) AS [sum:Profit:ok],
  SUM([tbl_Orders].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[tbl_Orders] [tbl_Orders]
GROUP BY [tbl_Orders].[Zip Code]
query-id: 3
```



2391	{\"ts\":\"2019-09-28T17:46:47.294\", \"pid\":13276, \"tid\":\"47c\", \"sev\":\"info\", \"req\":\"XY8P0ZU-KRQ#IBt638WK-AAAGI\", \"sess\":\"85A50268D0DD4776AA6D1A123C845E2C-1.0\", \"site\":\"Default\", \"user\":\"alan\", \"k\":\"ec-load\", \"v\":{\"cns\":\"LQTV1Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"zlib-compress-data\", \"v\":{\"compressed-size-b\":\"5981\", \"eqc-store\",\"v\":{\"class\":\"snowflake\", \"column-count\":\"3\", \"ec-store\",\"v\":{\"cns\":\"LQRV1Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"ec-store\", \"v\":{\"cns\":\"LQTV1Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"ec-load\", \"v\":{\"cns\":\"AQTV5Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"zlib-compress-data\", \"v\":{\"compressed-size-b\":\"6255\", \"eqc-store\",\"v\":{\"class\":\"snowflake\", \"column-count\":\"3\", \"ec-store\",\"v\":{\"cns\":\"AQRV5Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"ec-store\", \"v\":{\"cns\":\"AQTV5Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"qp-query-result-size\", \"v\":{\"num-cols\":\"3\", \"num-rows\":\"546\", \"qp-query-end\", \"v\":{\"elapsed\":\"10.3\", \"owner-component\":\"end-protocol.query\", \"l\":{\"o\",\"a\":{\"depth\":\"17\", \"elapsed-federation-run-query\", \"l\":{\"o\",\"a\":{\"depth\":\"15\", \"end-qp.federated-evaluator\", \"l\":{\"o\",\"a\":{\"depth\":\"14\", \"end-qp.run-query\", \"l\":{\"o\",\"a\":{\"depth\":\"16\", \"elapsed-logical.query\", \"l\":{\"o\",\"a\":{\"depth\":\"13\", \"end-LogicalQueryProcessorHelper.Execute\", \"l\":{\"o\",\"a\":{\"end-AQLTableProcessor.Process\", \"l\":{\"o\",\"a\":{\"depth\":\"10\", \"end-qp.abstract-query\", \"l\":{\"o\",\"a\":{\"depth\":\"10\", \"ec-load\", \"v\":{\"cns\":\"LQTV1Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"zlib-compress-data\", \"v\":{\"compressed-size-b\":\"10167\", \"eqc-store\",\"v\":{\"class\":\"snowflake\", \"column-count\":\"5\", \"ec-store\",\"v\":{\"cns\":\"LQTV1Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"ec-store\", \"v\":{\"cns\":\"LQRV1Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"ec-load\", \"v\":{\"cns\":\"AQTV5Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"zlib-compress-data\", \"v\":{\"compressed-size-b\":\"10362\", \"eqc-store\", \"v\":{\"class\":\"snowflake\", \"column-count\":\"5\", \"ec-store\", \"v\":{\"cns\":\"AQRV5Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"ec-store\", \"v\":{\"cns\":\"AQTV5Z\", \"elapsed-ms\":\"0\", \"key-hashtag\":\"qp-query-result-size\", \"v\":{\"num-cols\":\"5\", \"num-rows\":\"546\", \"qp-query-end\", \"v\":{\"elapsed\":\"10.366\", \"owner-component\":\"qp-batch-summary\", \"v\":{\"elapsed\":\"10.38\", \"elapsed-dash-board-image.Scale Up vs. Out() [FC05E2 msg.v\":{\"Setting WorksheetImage [Compare Rows] Ports VisualModelCache: cache found computing key msg.v\":{\"VisualModelCache: Key found\", \"end-protocol.query\", \"l\":{\"o,\"a\":{\"depth\":\"17\", \"elapsed-federation.run-query\", \"l\":{\"o,\"a\":{\"depth\":\"15\", \"end-qp.federated-evaluator\", \"l\":{\"o,\"a\":{\"depth\":\"14\", \"end-qp.run-query\", \"l\":{\"o,\"a\":{\"depth\":\"16\", \"elapsed-logical.query\", \"l\":{\"o,\"a\":{\"depth\":\"13\", \"end-LogicalQueryProcessorHelper.Execute\", \"l\":{\"o,\"a\":{\"end-AQLTableProcessor.Process\", \"l\":{\"o,\"a\":{\"depth\":\"10\", \"end-qp.abstract-query\", \"l\":{\"o,\"a\":{\"depth\":\"10\", \"begin-dash-board-image.update-zone-layout\", \"l\":{\"o,\"a\":{\"begin-visual-controller.resize-visual-model\", \"l\":{\"o,\"a\":{\"begin-compute-model-task.prepare-minimal-vw-update\", \"l\":{\"o,\"a\":{\"end-compute-model-task.prepare-minimal-vw-update\", \"l\":{\"o,\"a\":{\"end-visual-controller.resize-visual-model\", \"l\":{\"o,\"a\":{\"begin-worksheet.image.begin-compute-visual-model\", \"l\":{\"o,\"a\":{\"begin-visual-controller.update-visual-model\", \"l\":{\"o,\"a\":{\"begin-visual-controller.update-sheet\", \"l\":{\"o,\"a\":{\"end-visual-controller.update-sheet\", \"l\":{\"o,\"a\":{\"depth\":\"10\", \"begin-visual-controller.compute-or-reuse-visual-model\", \"l\":{\"o,\"a\":{\"begin-compute-model-task.execute\", \"l\":{\"o,\"a\":{\"depth\":\"10\", \"begin-compute-model-task.compute-partition-model\", \"l\":{\"o,\"a\":{\"begin-compute-model-task.compute-worksheet-data\", \"l\":{\"o,\"a\":{\"begin-data-interpreter3.prepare-ds\", \"l\":{\"o,\"a\":{\"depth\":\"10\", \"end-data.interpreter3.prepare-ds\", \"l\":{\"o,\"a\":{\"depth\":\"10\", \"begin-query-batch.process\", \"l\":{\"o,\"a\":{\"depth\":\"16\", \"end-query-batch.process\", \"l\":{\"o,\"a\":{\"depth\":\"16\", \"begin-data-interpreter3.process-query-batch\", \"l\":{\"o,\"a\":{\"begin-densification-interpreter.apply-specs\", \"l\":{\"o,\"a\":{\"
------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

# QPBS – Logical Structure

Header	<pre>"ts": "timestamp", "pid": "process id", "tid": "thread id", "sev": "severity", "req": "request", "sess": "session", "site": "site", "user": "user", "k": "key", "v": "value"</pre>
S	Summary
Q	Queries



# QPBS – Logical Structure

Header	<code>"ts", "pid", "tid", "sev", "req", "sess", "site", "user", "k", "v"</code>
Summary	<p><code>"elapsed:"</code> net time in second to complete the entire query batch <math>\max(\text{ts}) - \min(\text{ts})</math></p> <p><code>"elapsed-compute-keys:"</code> net time in seconds to compute and store cache keys</p> <p><code>"elapsed-sum:"</code> gross time in seconds to complete entire query batch <math>\Sigma \text{ elapsed query } 1 \text{ to } N</math></p> <p><code>"job-count:"</code> integer number of jobs in the query batch complexity</p>
Q	Queries



# QPBS – Logical Structure

H

"ts", "pid", "tid", "sev", "req", "sess", "site", "user", "k", "v"

S

"elapsed:", "elapsed-compute-keys:", "elapsed-sum:", "job-count:"

Queries 1 to N

"cache-hit:" type of cache hit including none  
"elapsed:" time in seconds to run the query  
"owner-component:" whether data or quick filters own the query  
"owner-dashboard:" name of dashboard owner of the query  
"owner-worksheet:" name of sheet owner of the query (alpha first if mult)  
"protocol-id:" protocol server used to run the query  
"query-abstract:" XML abstraction of the query  
"query-compiled:" physical query in native DB syntax  
"query-id:" id of the query in the batch stating with 0 to N



# Section III

## Put JSON into Cloud Bucket



# STAGE

The screenshot displays the Snowflake web interface. At the top, the navigation bar includes icons for Databases, Shares, Data Exchange, Warehouses, Worksheets, History, and Account. Below this, the breadcrumb trail shows 'Databases > TABLEAU'. The 'Stages' tab is selected, showing a table with columns: Stage, Schema, Location, Creation Time, Owner, and Comment. A 'Create Stage' dialog box is open in the foreground, titled 'Create Stage' with the instruction 'Choose a location for files to be staged'. It presents three options: 'Snowflake Managed' (with a cloud icon and a blue plus sign), 'Existing Amazon S3 Location' (with the Amazon S3 logo), and 'Existing Microsoft Azure Location' (with the Microsoft Azure logo). Each option has a radio button. At the bottom of the dialog are 'Cancel' and 'Next' buttons.

Stage	Schema	Location	Creation Time	Owner	Comment
TABLEAU_JSON	PUBLIC	Snowflake	8/30/19 12:07:02 PM	ACCOUNTADMIN	

PUT into Stage from SnowSQL

USE DATABASE TABLEAU;

USE SCHEMA TABLEAU;

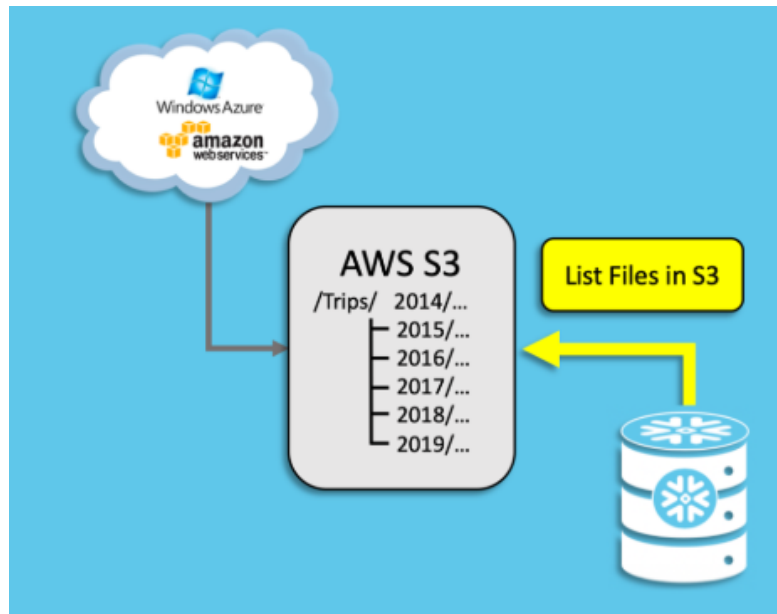
Put

file:///Path/Dir/VizQL\_Lumberjack/SpecificTransactions.json @TABLEAU\_JSON;





# QUERY FROM STAGE



CODE

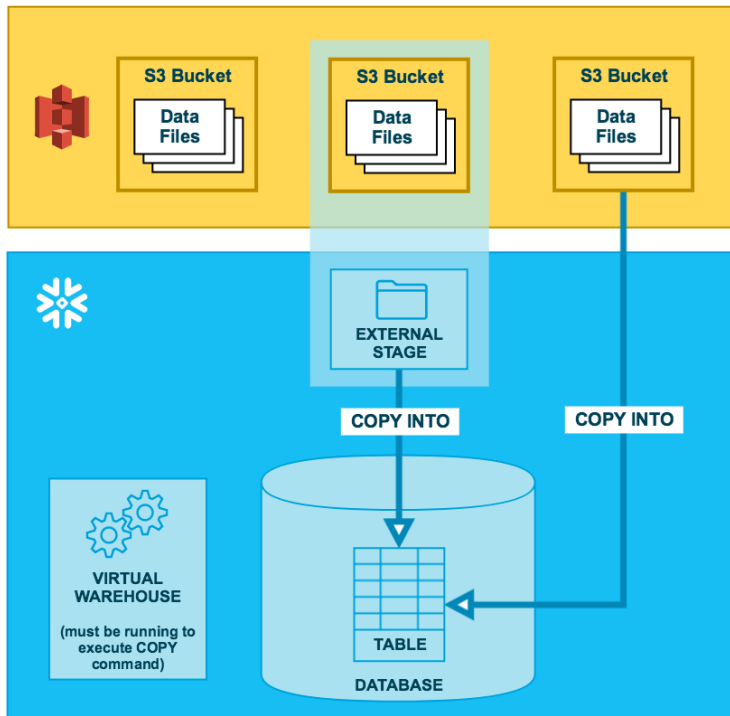
```
show stages;
```

```
Is @TABLEAU_JSON;
```

```
select $1 from  
  @TABLEAU_JSON/nativeapi_vizqlserver_1  
-0_2019_09_28_00_00_00.json (file_format  
=> json) limit 1;
```



# EXTERNAL TABLE



## CODE

```
create or replace external table exttbl_Test(  
  timestamp timestamp_ltz(9) as  
    (current_timestamp),  
  date date as to_date($1:ts::timestamp),  
  time time as to_time($1:ts::timestamp),  
  req string as ($1:req::string),  
  sess string as ($1:sess::string),  
  site string as ($1:site::string))  
Auto_Refresh = False -- Would be True for  
  SQS after PUT  
Location = @TABLEAU_JSON/  
File_Format = (type = json, File_Extension =  
  'json');
```

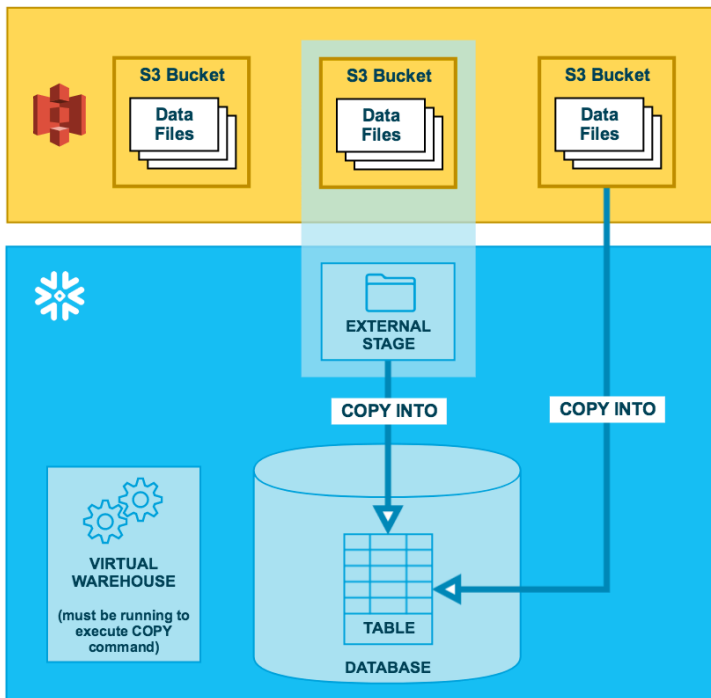


# Section IV

## COPY INTO Snowflake



# COPY INTO



## CODE

```
create or replace sequence counter start = 1  
increment = 1;
```

```
create or replace table stg_Logs (  
  id_json number,  
  log_json variant,  
  dts_json timestamp_ltz(9)  
);
```

```
copy into stg_Logs FROM (select  
  counter.nextval, $1, current_timestamp from  
  @TABLEAU_JSON/ (file_format => json))  
on_error = skip_file;
```

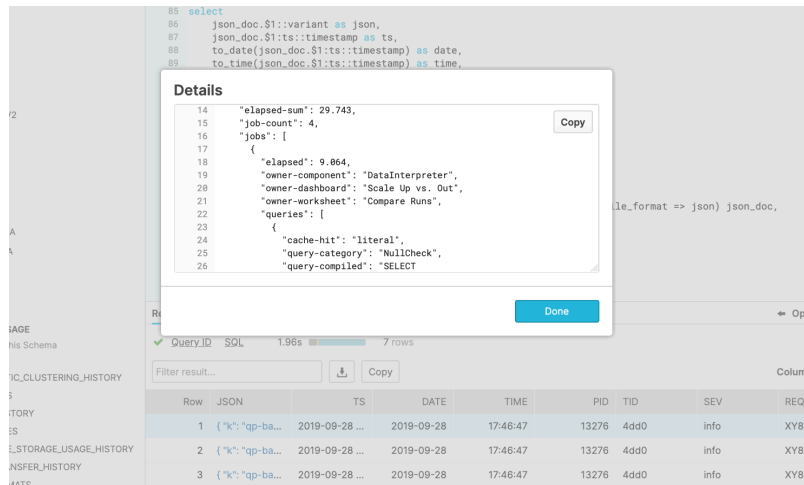


# Section V

## Prepare JSON for Analytics



# OPTIONAL FLATTEN



The screenshot shows the Snowflake SQL Editor interface. At the top, a SQL query is displayed with line numbers 85 to 89. The query uses the `optional flatten` function to parse a JSON document. A 'Details' popup window is open, showing the expanded JSON structure for the first row. Below the popup, a table of results is visible, showing columns for Row, JSON, TS, DATE, TIME, PID, TID, SEV, and REQ. The first three rows of data are shown.

```
85 select
86   json_doc.$1::variant as json,
87   json_doc.$1::timestamp as ts,
88   to_date(json_doc.$1::timestamp) as date,
89   to_time(json_doc.$1::timestamp) as time,
```

**Details**

```
14   "elapsed-sum": 29.743,
15   "job-count": 4,
16   "jobs": [
17     {
18       "elapsed": 9.864,
19       "owner-component": "DataInterpreter",
20       "owner-dashboard": "Scale Up vs. Out",
21       "owner-worksheet": "Compare Runs",
22       "queries": [
23         {
24           "cache-hit": "literal",
25           "query-category": "Nullcheck",
26           "query-compiled": "SELECT"
```

le\_format => json) json\_doc,

Query ID: SQL-1.965 7 rows

Filter result... [Download] [Copy]

Row	JSON	TS	DATE	TIME	PID	TID	SEV	REQ
1	{ "k": "qp-ba...	2019-09-28 ...	2019-09-28	17:46:47	13276	4dd0	info	XY8
2	{ "k": "qp-ba...	2019-09-28 ...	2019-09-28	17:46:47	13276	4dd0	info	XY8
3	{ "k": "qp-ba...	2019-09-28 ...	2019-09-28	17:46:47	13276	4dd0	info	XY8

## CODE

from

```
@TABLEAU_JSON/nativeapi_vizqlserver_1  
-0_2019_09_28_00_00_00.json (file_format  
=> json) json_doc,
```

```
lateral flatten(input =>  
parse_json(json_doc.$1:v:jobs)) jobs,
```

```
lateral flatten(input =>  
parse_json(jobs.value:queries)) queries
```



# OPTIONAL MV

## SNOWFLAKE JUST WORKS!



©2019 Snowflake Computing Inc. All Rights Reserved

20

## CODE

create or replace materialized view  
mv\_Snowjack\_Sessions

(User, Session, Start\_Time, End\_Time,  
Queries)

as

SELECT User, Sess as Session, MIN(TS) as  
Start\_Time, MAX(TS) as End\_Time,  
COUNT(query\_subid) as Queries

FROM tbl\_Snowjack GROUP BY User,  
Sess;

select \*,  
TIMESTAMPDIFF(seconds,Start\_Time,End  
\_Time) As Duration from  
mv\_Snowjack\_Sessions;



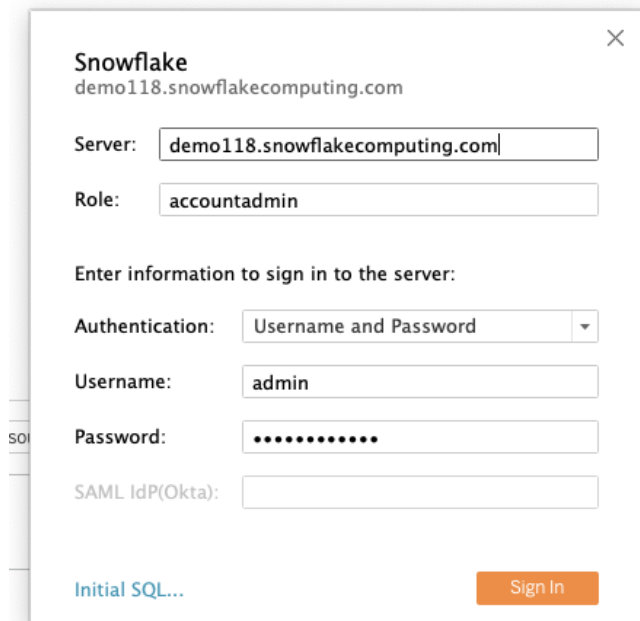
# Section VI

# Connect Tableau to Snowflake





# Connection Window



A screenshot of the Snowflake connection window. The window has a title bar with a close button (X). The main content area is titled "Snowflake" and shows the connection details for "demo118.snowflakecomputing.com". It includes input fields for "Server" (containing "demo118.snowflakecomputing.com"), "Role" (containing "accountadmin"), "Authentication" (a dropdown menu set to "Username and Password"), "Username" (containing "admin"), "Password" (masked with dots), and "SAML IdP(Okta)" (empty). At the bottom, there is a link "Initial SQL..." and a "Sign In" button.

**Snowflake**  
demo118.snowflakecomputing.com

Server: demo118.snowflakecomputing.com

Role: accountadmin

Enter information to sign in to the server:

Authentication: Username and Password

Username: admin

Password: .....

SAML IdP(Okta):

[Initial SQL...](#) [Sign In](#)

Custom SQL

Tables

Views

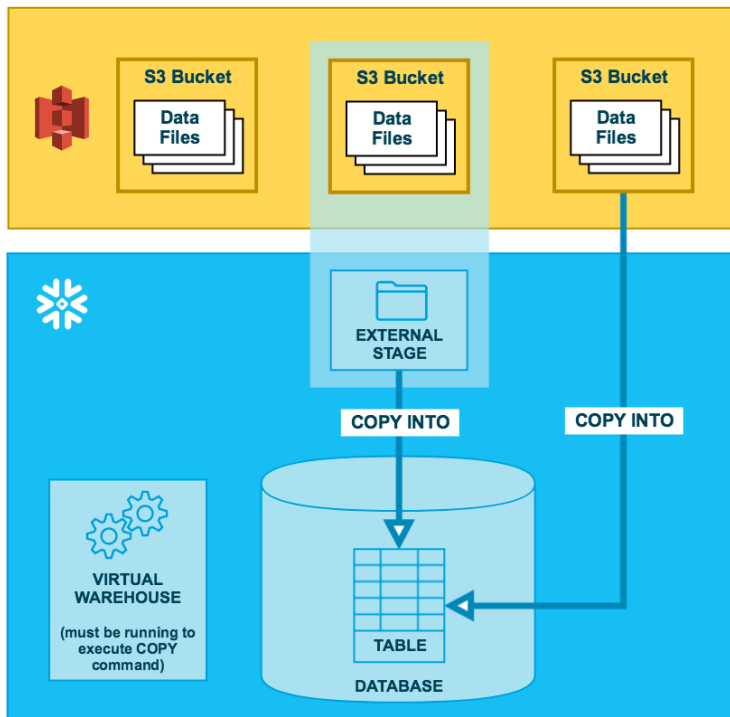


# Section VII

# Query Snowflake with Tableau



# External Table



Connections [Add](#)

demo118.snow...omputing.com  
Snowflake

Warehouse  
TABLEAU\_WH

Database  
TABLEAU

Schema  
TABLEAU

Table [p](#)

- EXTTBL\_TES...TTBL\_TEST)
- MV\_SNOWJAC...SESSIONS)
- STG\_LOGS (T...U.STG\_LOGS)
- TBL\_SNOWJA...SNOWJACK)

New Custom SQL

New Union

EXTTBL\_TEST (TABLEAU.EXTTBL\_TEST) (TABLEAU)

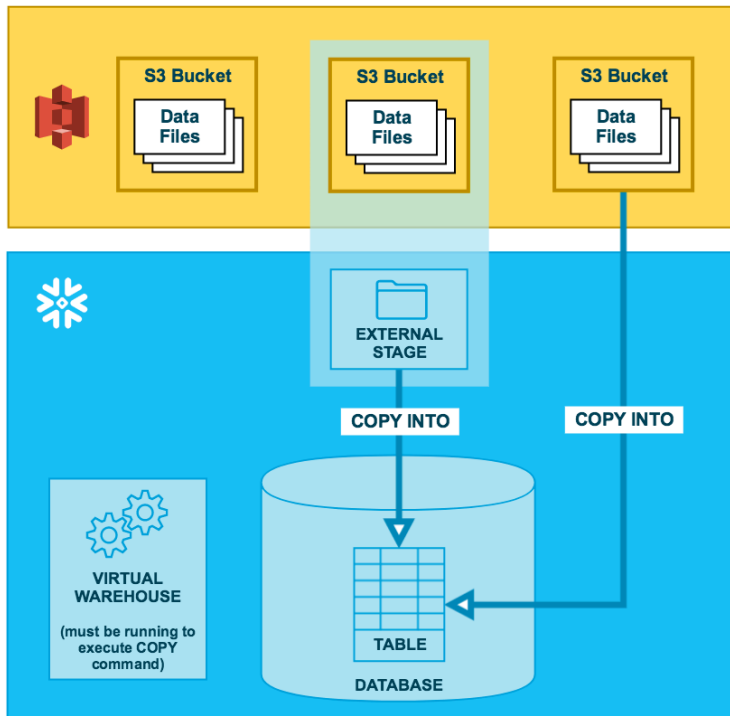
EXTTBL\_TEST

Sort fields | Data source order

Abc EXTTBL_TEST Value	EXTTBL_TEST Timestamp	EXTTBL_TEST Date	EXTTBL_TEST Time	# EXTTBL_TEST PID
1	10/23/2019 8:19:44 ...	null	null	no
[ 0	10/23/2019 8:19:44 ...	null	null	no
[ 0	10/23/2019 8:19:44 ...	null	null	no
[ 0	10/23/2019 8:19:44 ...	null	null	no



# Table



STG\_LOGS (TABLEAU.STG\_LOGS) (TABLEAU)

STG\_LOGS

Connections [Add](#)

demo118.snow...omputing.com  
Snowflake

Warehouse  
TABLEAU\_WH

Database  
TABLEAU

Schema  
TABLEAU

Table

- EXTTBL\_TES...TTBL\_TEST)
- MV\_SNOWJAC...SESSIONS)
- STG\_LOGS (T...U.STG\_LOGS)
- TBL\_SNOWJA...SNOWJACK)

New Custom SQL

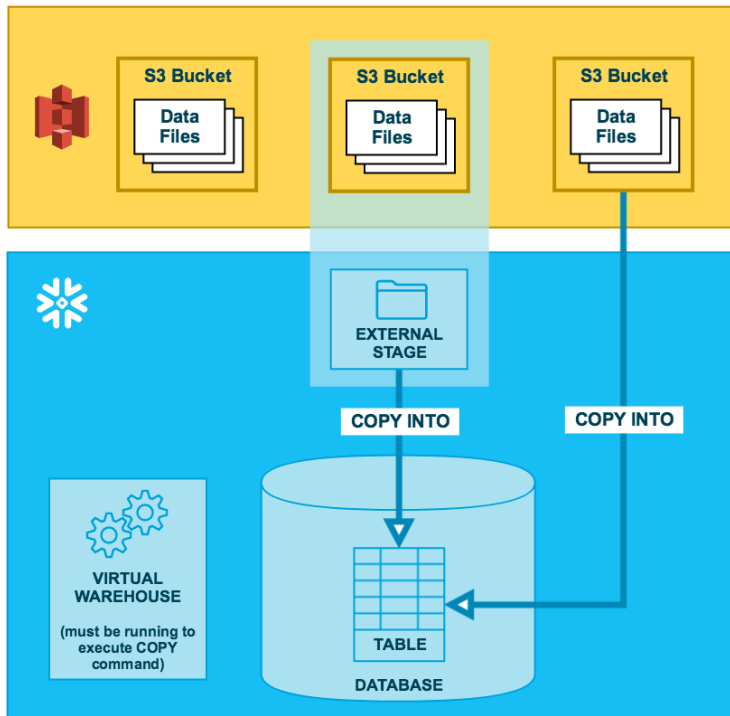
New Union

Sort fields: Data source order

#	STG_LOGS Id Json	STG_LOGS Log Json	STG_LOGS Dts Json
175931	{ "k": "rotate-log",		10/23/2019 7:50:05 ...
175932	{ "k": "msg",		10/23/2019 7:50:05 ...
175933	{ "k": "protocolpool-		10/23/2019 7:50:05 ...
175934	{ "k": "protocolpool-		10/23/2019 7:50:05 ...



# Materialized View



The screenshot shows the Snowflake web interface for configuring a Materialized View. The left sidebar shows the "Connections" section with a dropdown for "demo118.snow...computing.com". The main panel shows the configuration for the "MV\_SNOWJACK\_SESSIONS" table.

**Connections:** demo118.snow...computing.com (Snowflake)

**Warehouse:** TABLEAU\_WH

**Database:** TABLEAU

**Schema:** TABLEAU

**Table:** MV\_SNOWJACK\_SESSIONS (TABLEAU.MV\_SNOWJACK\_SESSIONS) ...

**Sort fields:** Data source order

Abc	Abc	MV_SNOWJACK_SESSIONS	MV_SNOWJACK_SESSIONS	#
USER	SESSION	START_TIME	END_TIME	MV_SNOWJACK_...
alan	67AB30C527DB46A1...	9/28/2019 10:21:26 ...	9/28/2019 10:22:40 ...	11
alan	BDC57C6341414DA5...	9/28/2019 5:55:22 PM	9/28/2019 5:55:22 PM	22
alan	7BCBE8A92FB64882...	10/2/2019 7:46:47 AM	10/2/2019 7:47:01 AM	18
alan	80EB442290444F5C...	10/2/2019 7:47:44 AM	10/2/2019 7:47:48 AM	14
alan	85A50268D8DD4776...	9/28/2019 5:46:47 PM	9/28/2019 5:46:47 PM	7



# Section VIII

# Conclusion



# Conclusion

- At the very top of the UI click on the “Partner Connect” icon to get access to trial/free ETL and Other tools to help you get more data into Snowflake and then analyze it.
- Read the “Definitive Guide to Maximizing Your Free Trial” document at:  
<https://www.snowflake.com/test-driving-snowflake-the-definitive-guide-to-maximizing-your-free-trial/>
- Attend a Snowflake virtual or in-person event to learn more about our capabilities and how customers use us: <https://www.snowflake.com/about/events/>
- Read the “Best Practices for Using Tableau with Snowflake” Whitepaper:  
<https://resources.snowflake.com/ebooks/best-practices-for-using-tableau-with-snowflake>
- See Us at our Tableau Conference Booth





THANK YOU

[snowflake.com/contact](https://snowflake.com/contact)