EPAM's Snowflake Hands-on Lab

Lab Overview

For those who begin to study Snowflake from scratch, it is recommended to start with "Hands-On Lab Guide for Snowflake Free Trial" that describes how to work with the main database features in the form of step-by-step guide.

This Lab (prepared by your EPAM colleagues) offers a high-level description of the practical task for self-directed learning.

The target group for the Lab are DWBI engineers with experience in building Data Warehouses using other databases (Oracle, MS SQL, Teradata, etc.).

Lab Data Set

Data set from TPC-H benchmark is proposed for the Lab. TPC-H allows you to generate data for 8 tables. The data volume (in gigabytes) is defined by scale factor (SF). For the Lab purpose, you can download prepared in advance data set (2 GB of raw data, SF=2):

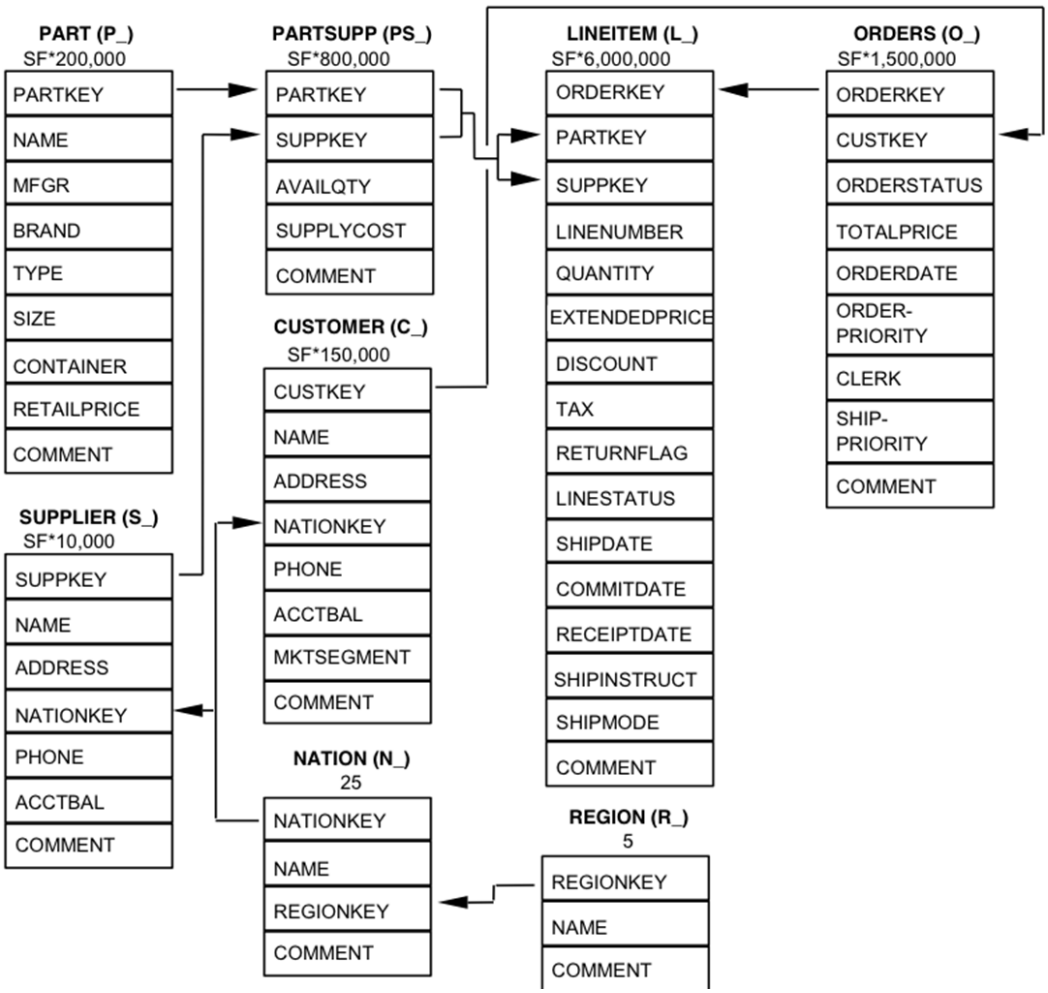


Table	Number of rows
H_LINEITEM	11 996 782
H_ORDER	3 000 000
H_PARTSUPP	1 600 000
H_PART	400 000
H_CUSTOMER	300 000
H_SUPPLIER	20 000
H_NATION	25
H_REGION	5

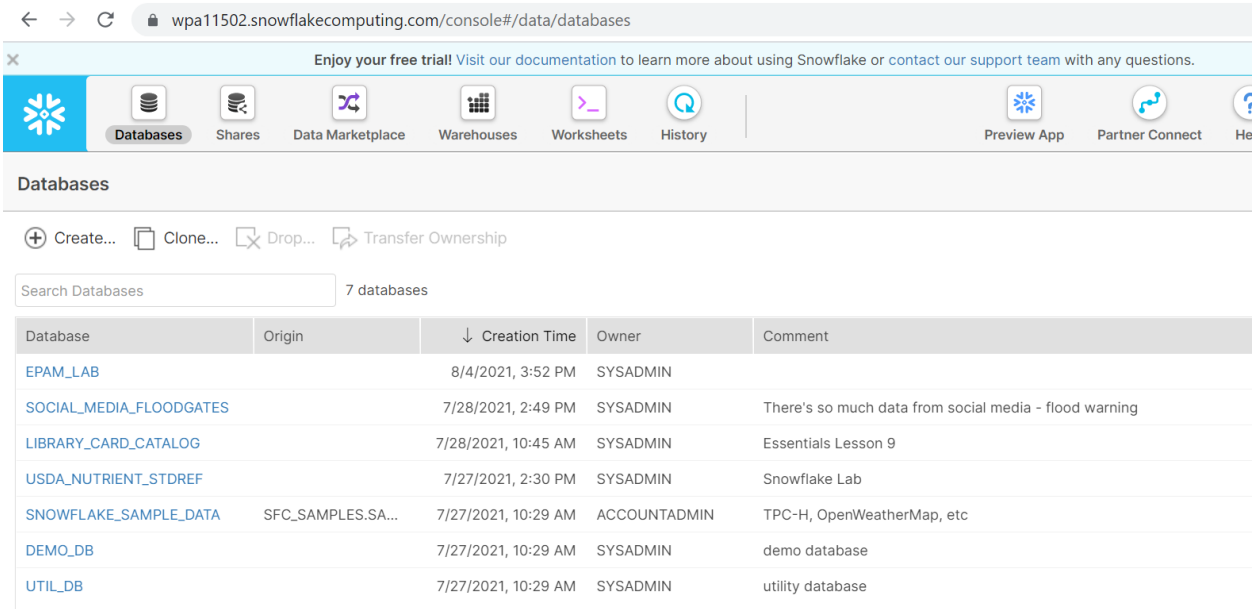
In the shared folder, you can also find DDL script for the tables: *tpch_ddl.sql*.

Lab Description

Hands-on-lab is considered as completed if you score >= 60 points.
(Tasks 1, 8 – 5 points each, Tasks 2, 4, 5, 6 – 10 points each, Task 3 – 30 points, Task 7 – 20 points).

1. Database creation

First, you need to create a separate database EPAM_LAB in Snowflake.



2. Data loading

In this step, you need to load Lab data set to internal (Snowflake) or external stage. If you have an existing account in AWS/GCP/Azure cloud, external stage would be preferable. Please note that you may need some data preparation steps before loading.

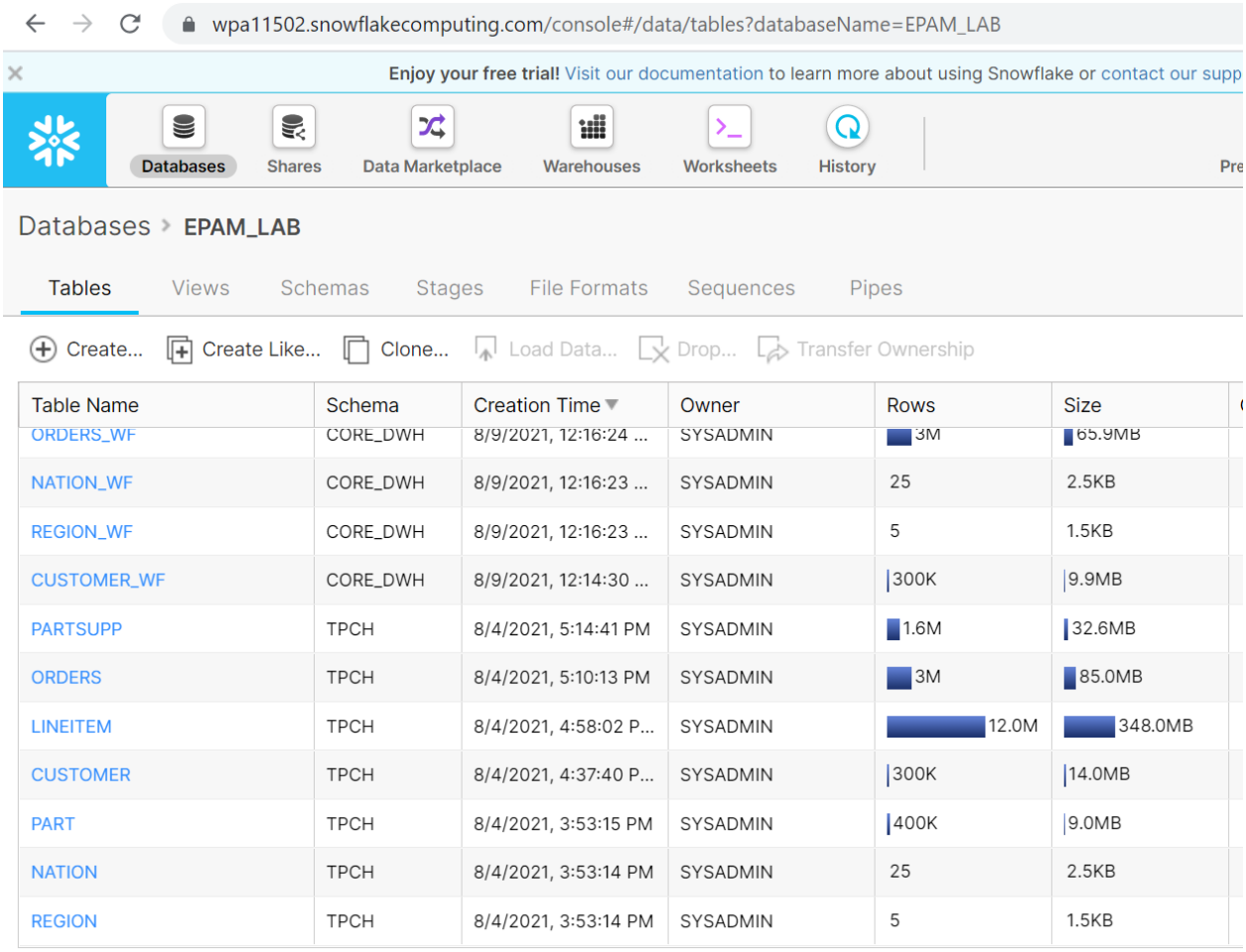


Table Name	Schema	Creation Time	Owner	Rows	Size
ORDERS_WF	CORE_DWH	8/9/2021, 12:16:24 ...	SYSADMIN	3M	65.9MB
NATION_WF	CORE_DWH	8/9/2021, 12:16:23 ...	SYSADMIN	25	2.5KB
REGION_WF	CORE_DWH	8/9/2021, 12:16:23 ...	SYSADMIN	5	1.5KB
CUSTOMER_WF	CORE_DWH	8/9/2021, 12:14:30 ...	SYSADMIN	300K	9.9MB
PARTSUPP	TPCH	8/4/2021, 5:14:41 PM	SYSADMIN	1.6M	32.6MB
ORDERS	TPCH	8/4/2021, 5:10:13 PM	SYSADMIN	3M	85.0MB
LINEITEM	TPCH	8/4/2021, 4:58:02 P...	SYSADMIN	12.0M	348.0MB
CUSTOMER	TPCH	8/4/2021, 4:37:40 P...	SYSADMIN	300K	14.0MB
PART	TPCH	8/4/2021, 3:53:15 PM	SYSADMIN	400K	9.0MB
NATION	TPCH	8/4/2021, 3:53:14 PM	SYSADMIN	25	2.5KB
REGION	TPCH	8/4/2021, 3:53:14 PM	SYSADMIN	5	1.5KB

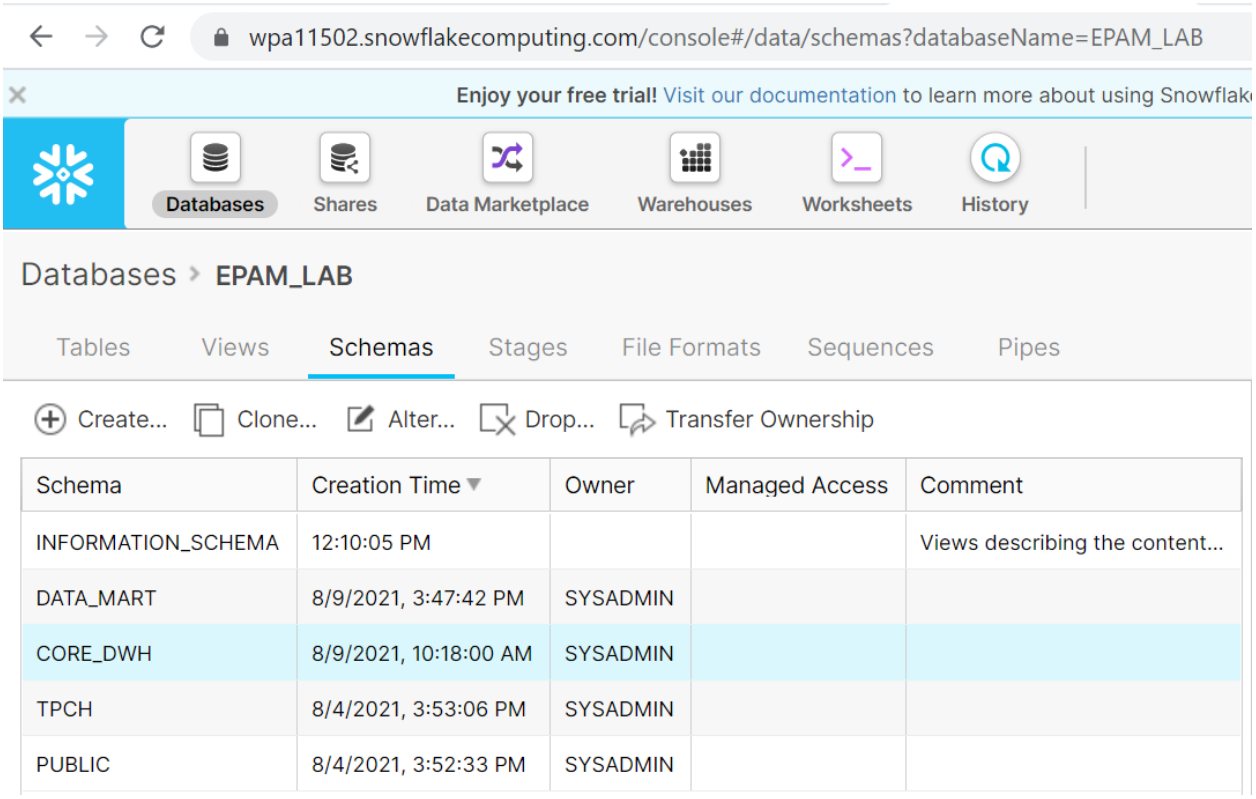
The data load was done using SnowSQL, in the following file you will find the code for loading through the internal stage.

Data Load Point 2.txt

3. ELT Data Workflow

Create two schemas in the DB you created before:

- CORE_DWH
- DATA_MART



Develop the following automated data workflow:

Stage -> CORE_DWH -> DATA_MART

Data in CORE_DWH should be modeled according to 3NF (as is - no transformation). Star Schema is a target data model for DATA_MART (data should be transformed accordingly).

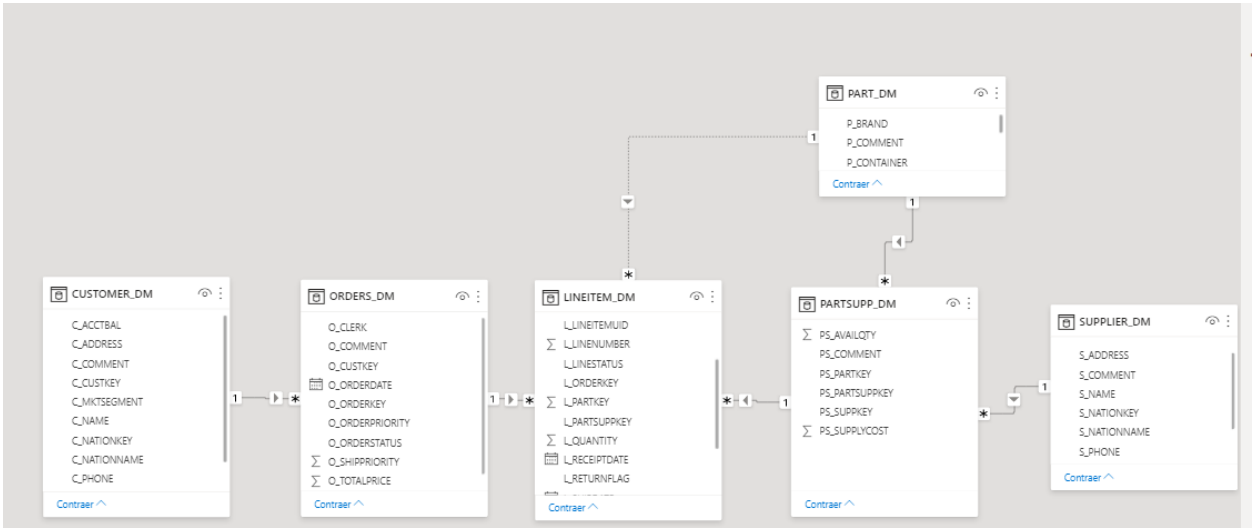
The following Snowflake features should be used:

- Orchestration Tasks
- Stored Procedures
- Tables Streams

The dataflow was created, I was able to create a simple stored procedure due to the fact that my knowledge in JavaScript is fairly limited. Here you can see the code used:

Data Load - WF.txt

And here's the final Star Schema created, the image is taken from the Power BI service:



4. Snowflake & 3rd party tools

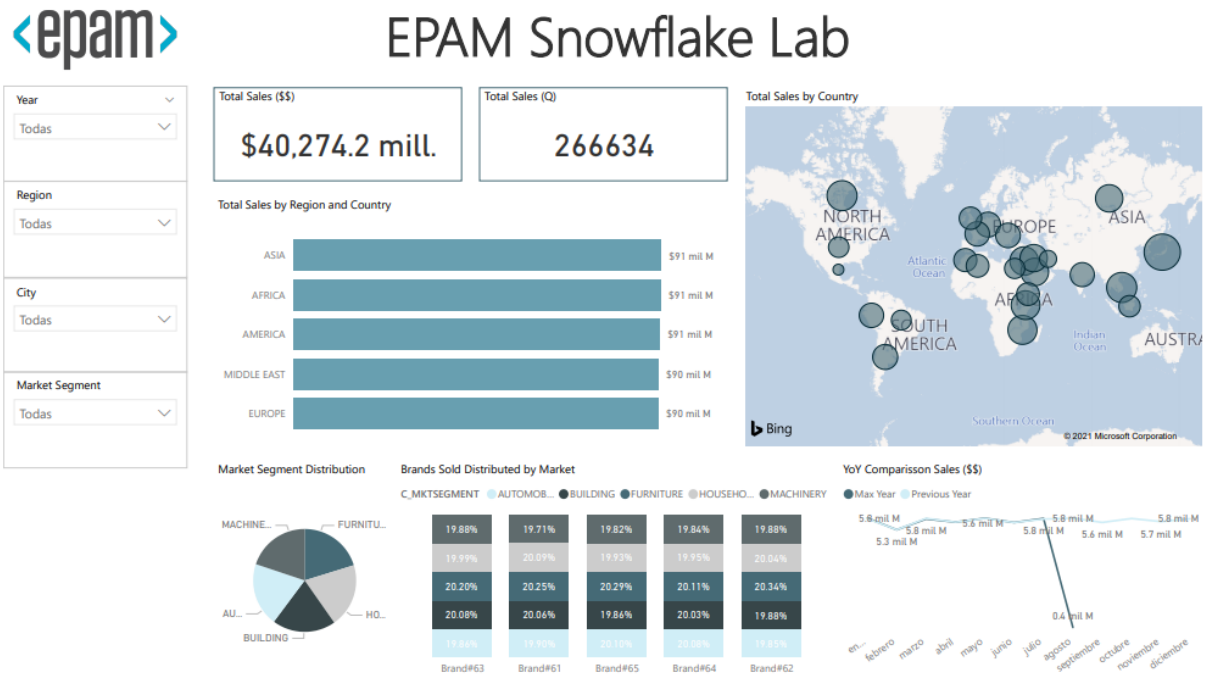
When the data is loaded to DATA_MART schema, connect Snowflake as a data source from any BI tool (Tableau, PowerBI, Qlik Sense, etc.) and create a simple dashboard.

Also, try connecting to Snowflake from any SQL editor (e.g. DBeaver).

The following files include a .pbix file and a .pdf:

EPAM_LAB_DASHBOARD.pdf

EPAM_LAB_DASHBOARD.pbix



5. Snowflake SQL

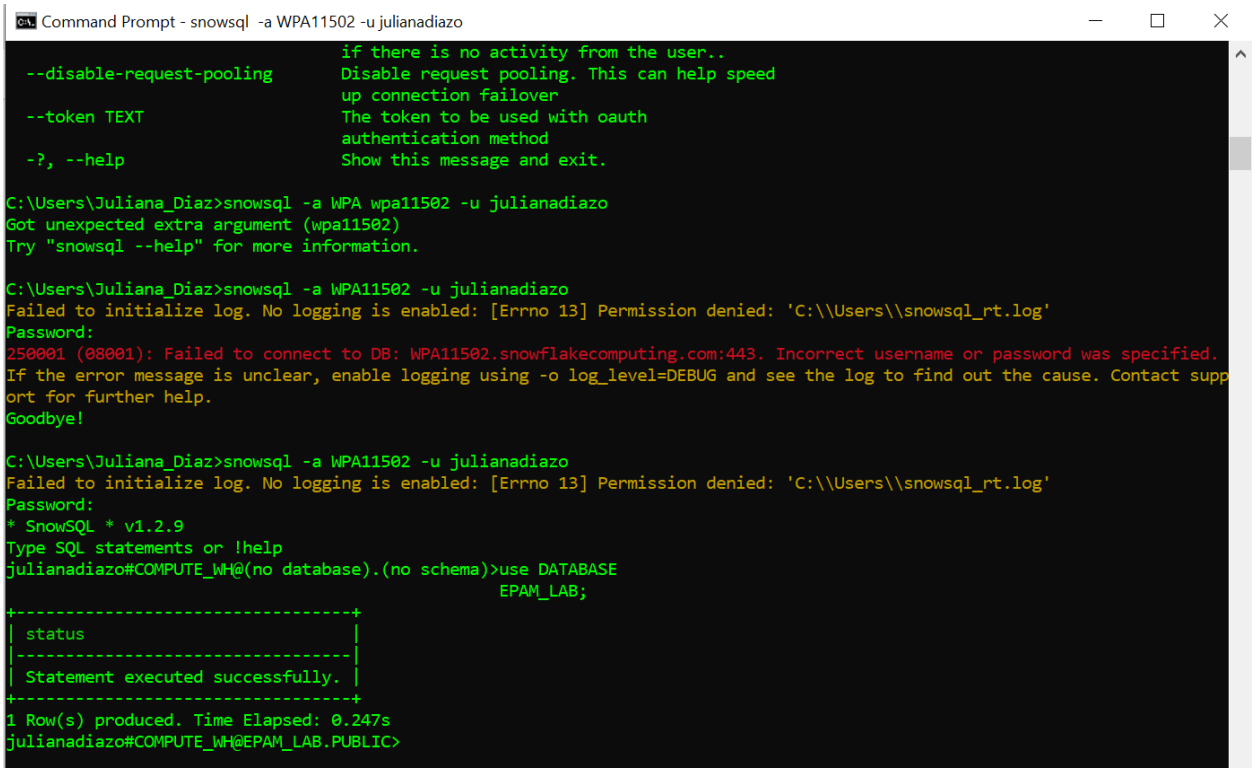
From the shared folder you can also download the file with 22 TPC-H benchmarking queries (tpch_benchmark_queries.sql). Please note that the queries were modified to execute in AWS RedShift database, so some of them may require modifications for Snowflake. Use the queries to test how Snowflake works:

- Create several warehouses of different sizes and compare their performance;
- Test how Snowflake leverages different types of cashes;
- Rewrite a couple of queries to execute on Start Schema data model and compare performance (3NF vs Star Schema);
- Execute queries using SnowSQL (CLI Client).

Data load code using SnowSQL.

Data Load Point 2.txt

Screen shot of the SnowSQL client connected



Queries:

```
julianadiazo#COMPUTE_WH@EPAM_LAB.TPCH>Select * from nation;
+-------------+---------------------------+-------------+--------------------------------------------------------------------------------------------------------+
| N_NATIONKEY | N_NAME                    | N_REGIONKEY | N_COMMENT                                                                                              |
|-------------+---------------------------+-------------+--------------------------------------------------------------------------------------------------------|
|           0 | ALGERIA                   |           0 | packages sleep .                                                                                       |
|           1 | ARGENTINA                 |           1 | quickly final instructions wake alongside of .                                                         |
|           2 | BRAZIL                    |           1 | carefully ironic ideas after affix quickly above .                                                     |
|           3 | CANADA                    |           1 | packages cajole carefully furiously even pinto beans .                                                 |
|           4 | EGYPT                     |           4 | blithely ironic pinto beans along haggle carefully ruthlessly special Ti                               |
|           5 | ETHIOPIA                  |           0 | regular, ironic deposits across wake after .                                                           |
|           6 | FRANCE                    |           3 | quickly even platelets among sleep about .                                                             |
|           7 | GERMANY                   |           3 | packages about use blithely furiously regular ideas .                                                  |
|           8 | INDIA                     |           2 | blithely express pinto beans along use blithely packages .                                             |
|           9 | INDONESIA                 |           2 | final, ironic deposits poach ruthlessly across :                                                       |
|          10 | IRAN                      |           4 | express, silent deposits cajole carefully ironic pinto beans .                                         |
|          11 | IRAQ                      |           4 | blithely final theodolites haggle carefully against .                                                  |
|          12 | JAPAN                     |           2 | quickly express platelets integrate quickly .                                                          |
|          13 | JORDAN                    |           4 | packages sleep about .                                                                                 |
|          14 | KENYA                     |           0 | regular, bold deposits sleep .                                                                         |
|          15 | MOROCCO                   |           0 | fluffily bold dolphins haggle carefully quickly regular instructions .                                 |
|          16 | MOZAMBIQUE                |           0 | silent accounts use blithely according to .                                                            |
|          17 | PERU                      |           1 | quickly bold instructions sleep alongside of .                                                         |
|          18 | CHINA                     |           2 | blithely furious theodolites cajole quickly bold instructions .                                        |
|          19 | ROMANIA                   |           3 | evenly bold pains sleep special, ironic deposits .                                                     |
|          20 | SAUDI ARABIA              |           4 | even accounts could cajole .                                                                           |
|          21 | VIETNAM                   |           2 | blithely even instructions use blithely .                                                              |
|          22 | RUSSIA                    |           3 | blithely silent pinto beans nag blithely .                                                             |
|          23 | UNITED KINGDOM            |           3 | blithely regular theodolites mold slowly :                                                             |
|          24 | UNITED STATES             |           1 | regular accounts was quickly even, express deposits .                                                  |
+-------------+---------------------------+-------------+--------------------------------------------------------------------------------------------------------+
```

```
julianadiazo#COMPUTE_WH@EPAM_LAB.TPCH> SELECT * FROM ORDERS LIMIT 4;
+-----------+----------+-------------+---------------+---------------+---------------+-----------------------------------------+-------------+----------
---+
| O_ORDERKEY | O_CUSTKEY | O_ORDERSTATUS | O_ORDERPRIORITY | O_CLERK         | O_SHIPPRIORITY | O_COMMENT                                 | O_TOTALPRICE | O_ORDERDA
TE |
|-----------+----------+-------------+---------------+---------------+---------------+-----------------------------------------+-------------+----------
---|
|    8236871 |    143534 | F             | 3-MEDIUM        | Clerk#000001814 |             0 | carefully ironic foxes haggle carefully after .   |   40297.3235 | 1995-05-1
4  |
|    8236903 |     81907 | P             | 3-MEDIUM        | Clerk#000000951 |             0 | pending requests above cajole furiously bold pint | 164358.9028 | 1995-05-1
6  |
|    8236930 |    217568 | P             | 4-NOT SPECIFIED | Clerk#000000399 |             0 | carefully brave ideas sleep .                     | 290492.3663 | 1995-05-1
7  |
|    8236997 |    284752 | P             | 4-NOT SPECIFIED | Clerk#000001273 |             0 | quickly final dependencies wake bold accounts .   |   90221.7012 | 1995-05-2
2  |
+-----------+----------+-------------+---------------+---------------+---------------+-----------------------------------------+-------------+----------
---+
4 Row(s) produced. Time Elapsed: 2.475s
julianadiazo#COMPUTE_WH@EPAM_LAB.TPCH>SELECT SUM (O_TOTALPRICE) AS PRICEPERPRIORITY,
                                     O_ORDERPRIORITY
                                     FROM ORDERS
                                     GROUP BY O_ORDERPRIORITY;
+-----------------+-----------------+
| PRICEPERPRIORITY | O_ORDERPRIORITY |
|-----------------+-----------------|
| 90825667127.2229 | 3-MEDIUM        |
| 90713197066.928  | 1-URGENT        |
| 90526540370.2413 | 2-HIGH          |
| 90640568676.5932 | 5-LOW           |
| 90683397435.5099 | 4-NOT SPECIFIED |
+-----------------+-----------------+
5 Row(s) produced. Time Elapsed: 1.690s
julianadiazo#COMPUTE_WH@EPAM_LAB.TPCH>
```

6. Other Snowflake features

Learn and test other interesting Snowflake features:

- Object Cloning;

```
164 CREATE TABLE lineitem_dm CLONE "EPAM_LAB"."CORE_DWH"."LINEITEM_WF";
165
166 ALTER TABLE "EPAM_LAB"."DATA_MART"."LINEITEM_DM"
167 ADD L_EXTENDEDPRICE1 FLOAT8,
168     L_DISCOUNT1 FLOAT8,
169     L_TAX1 FLOAT8,
170     L_SHIPDATE1 DATE,
171     L_COMMITDATE1 DATE,
172     L_RECEIPTDATE1 DATE,
173     L_PARTSUPPKEY VARCHAR,
174     L_LINEITEMUID VARCHAR;
```

- Time Travel;

```
select * from "EPAM_LAB"."DATA_MART"."PARTSUPP_DM" at(offset => -60*5);
select * from "EPAM_LAB"."DATA_MART"."PARTSUPP_DM" at(timestamp => 'Mon, 16 August 2021 16:20:00 -0500'::timestamp_tz);
```

ts Data Preview — Open History

uery ID SQL 4.58s 1,600,000 rows

result... Copy Columns

Row	PS_PARTKEY	PS_SUPPKEY	PS_AVAILQTY	PS_COMMENT	PS_SUPPLYCOST	PS_PARTSUPPKEY
1	191111	6139	506	quickly final instructi...	51.49	191111-6139
2	191113	11114	1473	packages are about .	148.16	191113-11114

- Data Sharing - share your DATA_MART schema with a colleague who helps you with this Lab.

7. Snowpipe

Automated incremental data loading using Snowpipe. Split lineitem & order files into several parts and simulate their sequential loading to stage buckets.

```
//STAGE CREATION

CREATE OR REPLACE STAGE Snowpipe_Epam
url='s3://snowpipe-epamlab'
credentials=(aws_key_id='AKIASDGNXN52OH2BXWKP' aws_secret_key='pVzjY6Lhzw5Xw8D38Risx4pKEbqQzaYzvK8aQW/P')
file_format= DSV;

SHOW STAGES;

//PIPE CREATION
CREATE OR REPLACE pipe epam_lab_pipe auto_ingest=true AS
COPY INTO snowpipe.nation
FROM @Snowpipe_Epam
file_format=DSV;

SHOW pipes;

list @snowpipe_Epam;

alter pipe epam_lab_pipe refresh;

SELECT COUNT (*) FROM nation;
```

esults Data Preview — Open History

Query ID SQL 44ms 1 rows

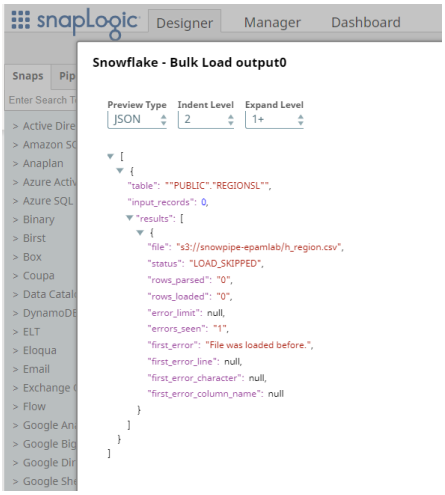
Filter result... Copy Columns

Row	COUNT (*)
1	25

8. Additional tasks

Connect your Snowflake account with partner applications available for a free trial (e.g. Fivetran, Periscope Data, Matillion in Partner Connect menu). Explore how selected tools work.

My account is currently connected to SnapLogic, in the following JSON snap we can see the table Region loaded through Snaps:


snapLogic
Designer
Manager
Dashboard
Snaps
Pip
Enter Search T
> Active Dire
> Amazon S
> Anaplan
> Azure Activ
> Azure SQL
> Binary
> Birst
> Box
> Coupa
> Data Catal
> DynamoDB
> ELT
> Eloqua
> Email
> Exchange
> Flow
> Google An
> Google Big
> Google Dir
> Google She
Snowflake - Bulk Load output0
Preview Type
JSON
Indent Level
2
Expand Level
1+
[
{
"table": ""PUBLIC"."REGIONSL"",
"input_records": 0,
"results": [
{
"file": "s3://snowpipe-epamlab/h_region.csv",
"status": "LOAD_SKIPPED",
"rows_parsed": "0",
"rows_loaded": "0",
"error_limit": null,
"errors_seen": "1",
"first_error": "File was loaded before.",
"first_error_line": null,
"first_error_character": null,
"first_error_column_name": null
}
]
}
]