

Snowflake Hands-on Lab

1. Database creation

Screenshot of DB

YS Yerzhan Sapenov
SYSADMIN

Worksheets

Dashboards

Data

Databases

Private Sharing

Provider Studio

Marketplace

Activity

Admin

Help & Support

Search

> DEMO_DB

> EPAM_LAB

> INFORMATION_SCHEMA

> PUBLIC

> GARDEN_PLANTS

> LIBRARY_CARD_CATALOG

> SNOWFLAKE_SAMPLE_DATA

> UTIL_DB

Databases

6 Databases

NAME ↑	SOURCE
DEMO_DB	Local
EPAM_LAB	Local
GARDEN_PLANTS	Local
LIBRARY_CARD_CA...	Local
SNOWFLAKE_SAM...	Share
UTIL_DB	Local

2. Data loading

Information about storage and stage

All the data is in the Azure blob

snowflakedata

Container

Search

Upload

Change access level

Refresh

Delete

Change tier

Acquire lease

Break lease

View snapshots

Create snapshot

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Access policy

Properties

Metadata

Authentication method: Access key (Switch to Azure AD User Account)

Location: snowflakedata

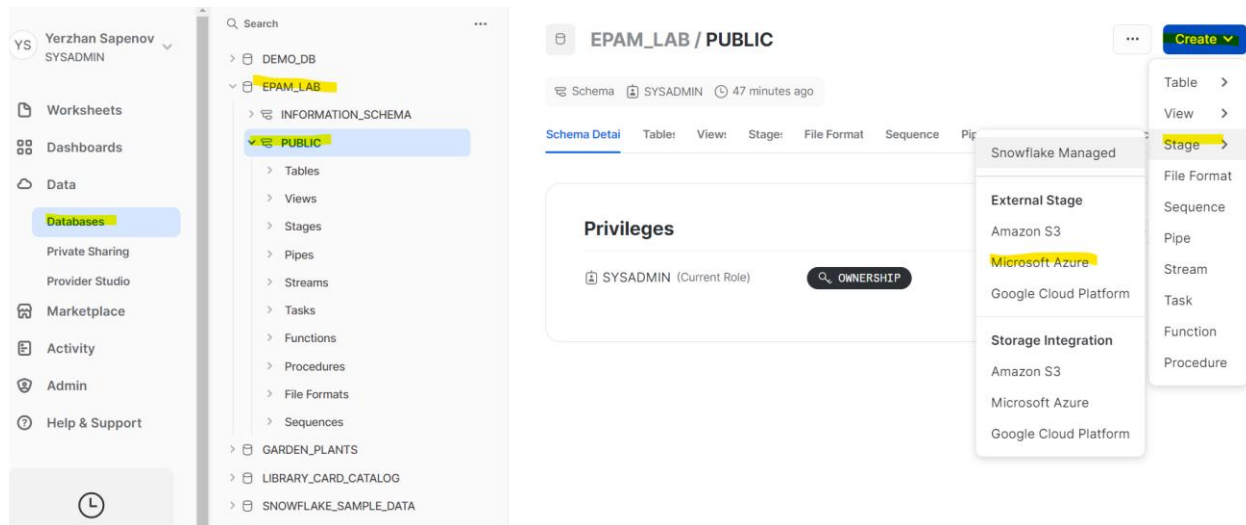
Search blobs by prefix (case-sensitive)

Show deleted blobs

Add filter

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
<input type="checkbox"/> h_customer.dsv	11/15/2022, 10:00:25...	Hot (Inferred)		Block blob	42.15 MiB	Available
<input type="checkbox"/> h_lineitem.dsv	11/15/2022, 2:48:57 ...	Hot (Inferred)		Block blob	1.67 GiB	Available
<input type="checkbox"/> h_nation.dsv	11/15/2022, 9:57:36 ...	Hot (Inferred)		Block blob	2.09 KiB	Available
<input type="checkbox"/> h_order.dsv	11/15/2022, 1:34:54 ...	Hot (Inferred)		Block blob	350.64 MiB	Available
<input type="checkbox"/> h_part.dsv	11/15/2022, 9:59:43 ...	Hot (Inferred)		Block blob	55.94 MiB	Available
<input type="checkbox"/> h_partsupp.dsv	11/15/2022, 10:05:27...	Hot (Inferred)		Block blob	111.69 MiB	Available
<input type="checkbox"/> h_region.csv	11/15/2022, 9:57:36 ...	Hot (Inferred)		Block blob	399 B	Available
<input type="checkbox"/> h_supplier.dsv	11/15/2022, 9:57:47 ...	Hot (Inferred)		Block blob	2.64 MiB	Available
<input type="checkbox"/> tpch_benchmark_querie...	11/15/2022, 12:38:57...	Hot (Inferred)		Block blob	11.94 KiB	Available
<input type="checkbox"/> tpch_ddl.sql	11/15/2022, 12:44:15...	Hot (Inferred)		Block blob	2.24 KiB	Available

Create Stage for Azure Blob by generating SAS token in EPAM_LAB.PUBLIC schema



Here the SAS token is for example, used another one, forgot to make a snapshot.

Create Securable Object in: EPAM_LAB → PUBLIC

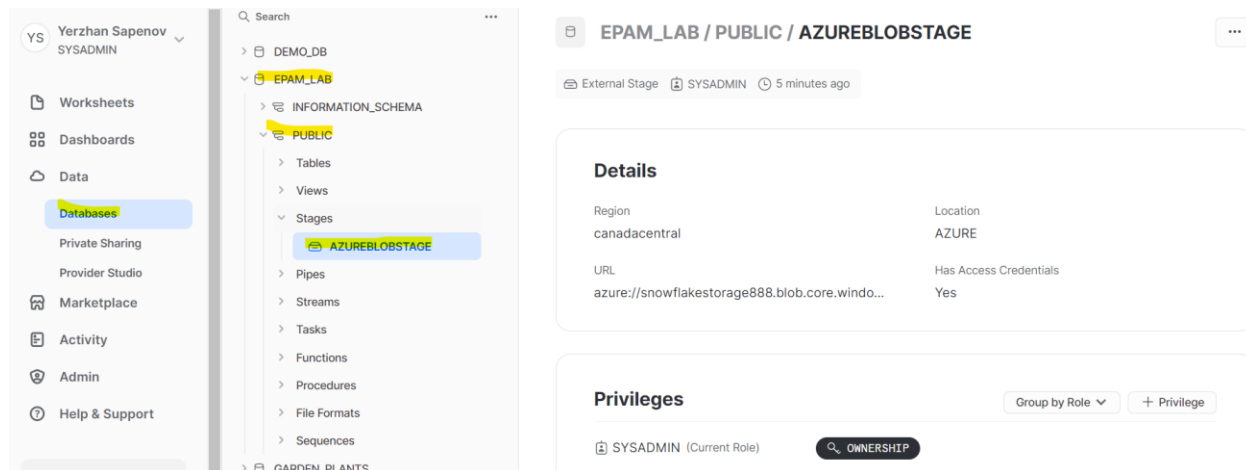
[Open in Worksheets](#)

[View Documentation](#)

SYSADMIN No Warehouse selected

Create Stage

```
1 create stage AZUREBLOBSTAGE[
2     url = 'azure://snowflakestorage888.blob.core.windows.net/snowflakedata'
3     credentials = (azure_sas_token = 'sp=r&st=2022-11-15T05:10:15Z&se=2022-11-15T13:10:15Z&spr=https&sv=2021-06-08&sr=c&sig=aV790mKTdrECN2GHfk380LUokODxgXGRfw73Y009h4Y%3D') ;
```



Created file formats for CSV and DSV

create file format EPAM_LAB.PUBLIC.DSVfileformat

TYPE = 'CSV'

FIELD_DELIMITER = '|'

SKIP_HEADER = 1 --one header row

FIELD_OPTIONALLY_ENCLOSED_BY = ''

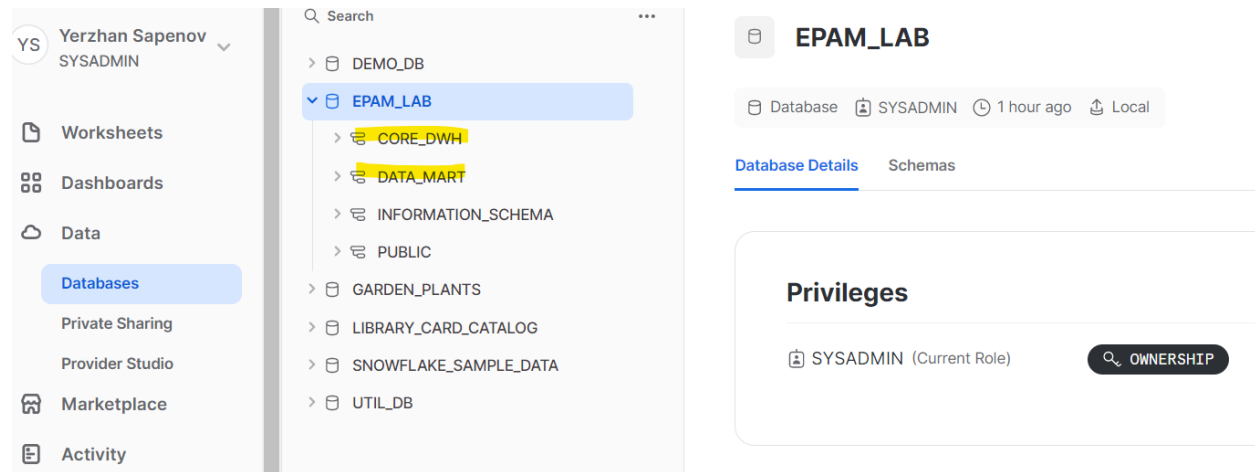
```

;
create file format EPAM_LAB.PUBLIC.CSVfileformat
  TYPE = 'CSV'--csv for comma separated files
  SKIP_HEADER = 1 --one header row
  FIELD_OPTIONALLY_ENCLOSED_BY = ''
;

```

3. ELT Data Workflow

Created 2 schemas in EPAM_LAB database.



Created tables in CORE_DWH, copied data and checked the correctness of file formats

```

CREATE OR REPLACE TABLE epam_lab.core_dwh.h_region
(
  r_regionkey INTEGER,
  r_name CHAR(25),
  r_comment VARCHAR(152)
);

```

```

COPY INTO epam_lab.core_dwh.h_region
FROM @epam_lab.public.azureblobstage
files = ('h_region.csv')
file_format = ( format_name= epam_lab.public.CSVFILEFORMAT );

```

```

CREATE OR REPLACE TABLE epam_lab.core_dwh.h_nation
(
  n_nationkey INTEGER not null,
  n_name CHAR(27),
  n_regionkey INTEGER,
  n_comment VARCHAR(155)
);

```

```

COPY INTO epam_lab.core_dwh.h_nation
FROM @epam_lab.public.azureblobstage
files = ('h_nation.dsv')
file_format = ( format_name= epam_lab.public.DSVFILEFORMAT );

```

```
CREATE OR REPLACE TABLE epam_lab.core_dwh.h_supplier
```

```
(  
  s_supkey  INTEGER not null,  
  s_name    CHAR(25),  
  s_address VARCHAR(40),  
  s_nationkey INTEGER,  
  s_phone   CHAR(15),  
  s_acctbal FLOAT8,  
  s_comment VARCHAR(101)  
);
```

```
COPY INTO epam_lab.core_dwh.h_supplier
```

```
FROM (  
  SELECT t.$1, t.$2, t.$3, t.$4, t.$5,  
         replace(t.$6,',','.'),  
         t.$7  
  FROM @epam_lab.public.azureblobstage/h_supplier.dsv AS t)  
file_format = ( format_name= epam_lab.public.DSVFILEFORMAT );
```

```
CREATE OR REPLACE TABLE epam_lab.core_dwh.h_orders
```

```
(  
  o_orderkey  INTEGER not null,  
  o_custkey   INTEGER not null,  
  o_orderstatus CHAR(1),  
  o_totalprice FLOAT8,  
  o_orderdate  DATE,  
  o_orderpriority CHAR(15),  
  o_clerk      CHAR(15),  
  o_shippriority INTEGER,  
  o_comment    VARCHAR(79)  
);
```

```
COPY INTO epam_lab.core_dwh.h_orders
```

```
FROM (  
  SELECT t.$1, t.$2, t.$3, replace(t.$4,',','.'),  
         TO_DATE(t.$5,'DD.MM.YY'), t.$6, t.$7, t.$8, t.$9  
  FROM @epam_lab.public.azureblobstage/h_order.dsv AS t)  
file_format = ( format_name= epam_lab.public.DSVFILEFORMAT );
```

```
CREATE OR REPLACE TABLE epam_lab.core_dwh.h_partsupp
```

```
(  
  ps_partkey  INTEGER not null,  
  ps_supkey   INTEGER not null,  
  ps_availqty  INTEGER,  
  ps_supplycost FLOAT8 not null,  
  ps_comment  VARCHAR(199)  
);
```

```
COPY INTO epam_lab.core_dwh.h_partsupp
```

```
FROM (  
  SELECT t.$1, t.$2, t.$3,  
         replace(t.$4,',','.'),  
         t.$5  
  FROM @epam_lab.public.azureblobstage/h_partsupp.dsv AS t)
```

```
file_format = ( format_name= epam_lab.public.DSVFILEFORMAT );
```

```
CREATE OR REPLACE TABLE epam_lab.core_dwh.h_part
```

```
(  
  p_partkey  INTEGER not null,  
  p_name     VARCHAR(55),  
  p_mfgr     CHAR(25),  
  p_brand    CHAR(10),  
  p_type     VARCHAR(25),  
  p_size     INTEGER,  
  p_container CHAR(10),  
  p_retailprice INTEGER,  
  p_comment  VARCHAR(23)  
);
```

```
COPY INTO epam_lab.core_dwh.h_part
```

```
FROM @epam_lab.public.azureblobstage
```

```
files = ( 'h_part.dsv' )
```

```
file_format = ( format_name= epam_lab.public.DSVFILEFORMAT );
```

```
CREATE OR REPLACE TABLE epam_lab.core_dwh.h_customer
```

```
(  
  c_custkey  INTEGER not null,  
  c_name     VARCHAR(25),  
  c_address  VARCHAR(40),  
  c_nationkey INTEGER,  
  c_phone    CHAR(15),  
  c_acctbal  FLOAT8,  
  c_mktsegment CHAR(10),  
  c_comment  VARCHAR(117)  
);
```

```
COPY INTO epam_lab.core_dwh.h_customer
```

```
FROM (
```

```
SELECT t.$1, t.$2, t.$3, t.$4, t.$5,  
       replace(t.$6,',','.'),  
       t.$7, t.$8
```

```
FROM @epam_lab.public.azureblobstage/h_customer.dsv AS t)
```

```
file_format = ( format_name= epam_lab.public.DSVFILEFORMAT );
```

```
CREATE OR REPLACE TABLE epam_lab.core_dwh.h_lineitem
```

```
(  
  l_orderkey  INTEGER not null,  
  l_partkey   INTEGER not null,  
  l_suppkey   INTEGER not null,  
  l_linenumbr INTEGER not null,  
  l_quantity  INTEGER not null,  
  l_extendedprice FLOAT8 not null,  
  l_discount  FLOAT8 not null,  
  l_tax       FLOAT8 not null,  
  l_returnflag CHAR(1),  
  l_linestatus CHAR(1),  
  l_shipdate  DATE,  
  l_commitdate DATE,
```

```

l_receiptdate DATE,
l_shipinstruct CHAR(25),
l_shipmode CHAR(10),
l_comment VARCHAR(44)
);

```

```

COPY INTO epam_lab.core_dwh.h_lineitem
FROM (
SELECT t.$1, t.$2, t.$3, t.$4, t.$5,
       replace(t.$6,','),''),
       replace(t.$7,','),''),
       replace(t.$8,','),''),
       t.$9, t.$10,
       TO_DATE(t.$11,'DD.MM.YY'),
       TO_DATE(t.$12,'DD.MM.YY'),
       TO_DATE(t.$13,'DD.MM.YY'),
       t.$14, t.$15, t.$16
FROM @epam_lab.public.azureblobstage/h_lineitem.dsv AS t)
file_format = ( format_name= epam_lab.public.DSVFILEFORMAT );

```

Created denormalized tables in DATA_MART and populated with data from CORE_DWH

Table LINEITEM columns remain the same + Concatenate (PARTKEY, SUPPKEY).

```

CREATE OR REPLACE TABLE epam_lab.DATA_MART.h_lineitem AS
SELECT l_orderkey, l_partkey, l_suppkey, l_linenum, l_quantity, l_extendedprice, l_discount, l_tax, l_returnflag,
l_linestatus, l_shipdate, l_commitdate, l_receiptdate, l_shipinstruct, l_shipmode, l_comment, concat(l_partkey,l_suppkey) AS
l_partsuppkey
FROM epam_lab.core_dwh.h_lineitem;

```

Table ORDERS columns:

orderkey, orderstatus, totalprice, orderdate, order-priority, clerk, ship-priority, comment, custkey, customer_name, customer_address, customer_nation_name, customer_nation_region_name, customer_nation_region_comment, customer_nation_comment, customer_phone, customer_acctbal, customer_mktsegment, customer_comment.

```

CREATE OR REPLACE TABLE epam_lab.DATA_MART.h_orders AS
SELECT o.o_orderkey, o.o_custkey, o.o_orderstatus, o.o_totalprice, o.o_orderdate, o.o_orderpriority, o.o_clerk,
o.o_shippriority, o.o_comment, c.c_name, c.c_address, n.n_name, r.r_name, r.r_comment, n.n_comment, c.c_phone,
c.c_acctbal, c.c_mktsegment, c.c_comment
FROM epam_lab.core_dwh.h_orders AS o LEFT OUTER JOIN
epam_lab.core_dwh.h_customer AS c ON o.o_custkey = c.c_custkey LEFT OUTER JOIN
epam_lab.core_dwh.h_nation AS n ON c.c_nationkey = n.n_nationkey LEFT OUTER JOIN
epam_lab.core_dwh.h_region AS r ON n.n_regionkey = r.r_regionkey;

```

Table PARTSUP columns:

partkey, part_name, part_mfgr, part_brand, part_type, part_size, part_container, part_retailprice, part_comment, suppkey, supplier_name, supplier_address, supplier_nation_name, supplier_nation_region_name, supplier_nation_region_comment, supplier_nation_comment, supplier_phone, supplier_acctbal, supplier_comment, availqty, supplycost, comment, concatenate (partkey, suppkey).

```

CREATE OR REPLACE TABLE epam_lab.DATA_MART.h_partsupp AS
SELECT ps.ps_partkey, p.p_name, p.p_mfgr, p.p_brand, p.p_type, p.p_size, p.p_container, p.p_retailprice, p.p_comment,
ps.ps_suppkey, s.s_name, s.s_address, n.n_name, r.r_name, r.r_comment, n.n_comment, s.s_phone, s.s_acctbal, s.s_comment,
ps.ps_availqty, ps.ps_supplycost, ps.ps_comment, concat(ps.ps_partkey,ps.ps_suppkey) AS ps_partsuppkey

```

```

FROM epam_lab.core_dwh.h_partsupp AS ps LEFT OUTER JOIN
epam_lab.core_dwh.h_part AS p ON ps.ps_partkey = p.p_partkey LEFT OUTER JOIN
epam_lab.core_dwh.h_supplier AS s ON ps.ps_suppkey = s.s_suppkey LEFT OUTER JOIN
epam_lab.core_dwh.h_nation AS n ON s.s_nationkey = n.n_nationkey LEFT OUTER JOIN
epam_lab.core_dwh.h_region AS r ON n.n_regionkey = r.r_regionkey;

```

Process automation

Create Stream, Procedure and Task for tables

```

CREATE OR REPLACE STREAM epam_lab.core_dwh.h_orders_stream on table epam_lab.core_dwh.h_orders;
CREATE OR REPLACE PROCEDURE epam_lab.core_dwh.DWH_TO_MART_ORDERS()
RETURNS VARCHAR
LANGUAGE SQL
AS
BEGIN
    INSERT INTO epam_lab.DATA_MART.h_orders
    SELECT o.o_orderkey, o.o_custkey, o.o_orderstatus, o.o_totalprice, o.o_orderdate, o.o_orderpriority, o.o_clerk,
o.o_shippriority, o.o_comment, c.c_name, c.c_address, n.n_name, r.r_name, r.r_comment, n.n_comment, c.c_phone,
c.c_acctbal, c.c_mktsegment, c.c_comment
    FROM epam_lab.core_dwh.h_orders_stream AS o LEFT OUTER JOIN
    epam_lab.core_dwh.h_customer AS c ON o.o_custkey = c.c_custkey LEFT OUTER JOIN
    epam_lab.core_dwh.h_nation AS n ON c.c_nationkey = n.n_nationkey LEFT OUTER JOIN
    epam_lab.core_dwh.h_region AS r ON n.n_regionkey = r.r_regionkey;
END;
CREATE OR REPLACE TASK epam_lab.core_dwh.DWH_TO_MART_ORDERS_task
warehouse = COMPUTE_WH
schedule = '30 minute'
when
    system$stream_has_data('epam_lab.core_dwh.h_ORDERS_stream')
as
CALL epam_lab.core_dwh.DWH_TO_MART_ORDERS();
ALTER TASK epam_lab.core_dwh.DWH_TO_MART_ORDERS_task RESUME;

CREATE OR REPLACE STREAM epam_lab.core_dwh.h_partsupp_stream on table epam_lab.core_dwh.h_partsupp;
CREATE OR REPLACE PROCEDURE epam_lab.core_dwh.DWH_TO_MART_PARTSUPP()
RETURNS VARCHAR
LANGUAGE SQL
AS
BEGIN
    INSERT INTO epam_lab.DATA_MART.h_PARTSUPP
    SELECT ps.ps_partkey, p.p_name, p.p_mfgr, p.p_brand, p.p_type, p.p_size, p.p_container, p.p_retailprice, p.p_comment,
ps.ps_suppkey, s.s_name, s.s_address, n.n_name, r.r_name, r.r_comment, n.n_comment, s.s_phone, s.s_acctbal, s.s_comment,
ps.ps_availqty, ps.ps_supplycost, ps.ps_comment, concat(ps.ps_partkey,ps.ps_suppkey) AS ps_partsuppkey
    FROM epam_lab.core_dwh.h_PARTSUPP_STREAM AS ps LEFT OUTER JOIN
    epam_lab.core_dwh.h_part AS p ON ps.ps_partkey = p.p_partkey LEFT OUTER JOIN
    epam_lab.core_dwh.h_supplier AS s ON ps.ps_suppkey = s.s_suppkey LEFT OUTER JOIN
    epam_lab.core_dwh.h_nation AS n ON s.s_nationkey = n.n_nationkey LEFT OUTER JOIN
    epam_lab.core_dwh.h_region AS r ON n.n_regionkey = r.r_regionkey;
END;
CREATE OR REPLACE TASK epam_lab.core_dwh.DWH_TO_MART_PARTSUPP_task
warehouse = COMPUTE_WH
schedule = '30 minute'
when
    system$stream_has_data('epam_lab.core_dwh.h_PARTSUPP_stream')

```

```

as
CALL epam_lab.core_dwh.DWH_TO_MART_PARTSUPP();
ALTER TASK epam_lab.core_dwh.DWH_TO_MART_PARTSUPP_task RESUME;

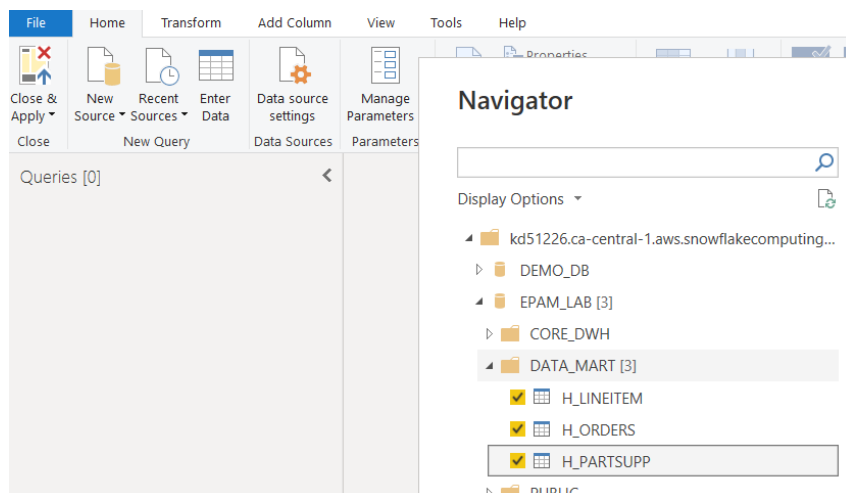
CREATE OR REPLACE STREAM epam_lab.core_dwh.h_lineitem_stream on table epam_lab.core_dwh.h_lineitem;
CREATE OR REPLACE PROCEDURE epam_lab.core_dwh.DWH_TO_MART_LINEITEM()
RETURNS VARCHAR
LANGUAGE SQL
AS
BEGIN
    INSERT INTO epam_lab.DATA_MART.h_LINEITEM
    SELECT l_orderkey, l_partkey, l_suppkey, l_linenumber, l_quantity, l_extendedprice, l_discount, l_tax, l_returnflag,
    l_linestatus, l_shipdate, l_commitdate, l_receiptdate, l_shipinstruct, l_shipmode, l_comment, concat(l_partkey,l_suppkey) AS
    l_partsupkey
    FROM epam_lab.core_dwh.h_LINEITEM_STREAM;
END;
CREATE OR REPLACE TASK epam_lab.core_dwh.DWH_TO_MART_LINEITEM_task
warehouse = COMPUTE_WH
schedule = '30 minute'
when
    system$stream_has_data('epam_lab.core_dwh.h_LINEITEM_stream')
as
CALL epam_lab.core_dwh.DWH_TO_MART_LINEITEM();
ALTER TASK epam_lab.core_dwh.DWH_TO_MART_LINEITEM_task RESUME;

call system$wait(70);

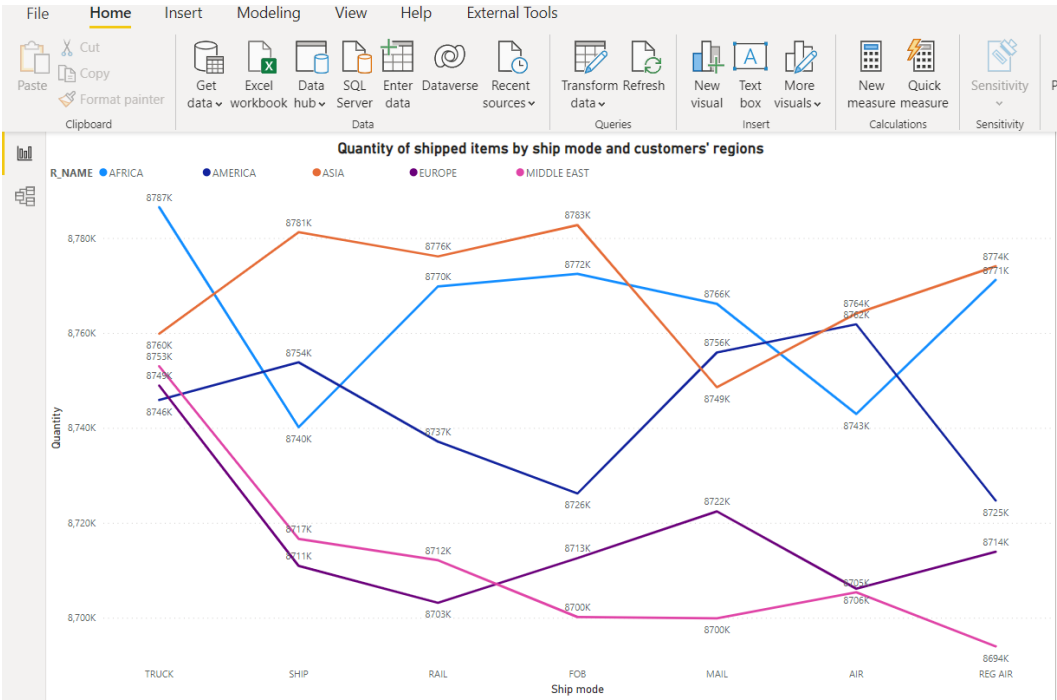
```

4. Snowflake & 3rd party tools

Connection from Power BI



Sample graph from the data



Connection from DBeaver

DBeaver 22.2.4 - <EPAM_LAB> Script

File Edit Navigate Search SQL Editor Database Window Help

SQL | Commit | Rollback | Auto | EPAM_LAB | PUBLIC@EPAM_LAB

Database Navigator | Projects

Enter a part of object name here

- EPAM_LAB - kd51226.ca-central-1.aws.snowflakecomputing.com:443
 - DEMO_DB
 - EPAM_LAB
 - CORE_DWH
 - Tables
 - H_CUSTOMER
 - H_LINEITEM
 - H_NATION
 - H_ORDERS
 - H_PART
 - H_PARTSUPP
 - H_REGION
 - H_SUPPLIER
 - Views
 - Procedures
 - DWH_TO_MART_LINEITEM
 - DWH_TO_MART_ORDERS
 - DWH_TO_MART_PARTSUPP
 - Data Types
 - DATA_MART
 - Tables
 - H_LINEITEM
 - H_ORDERS
 - H_PARTSUPP
 - Views
 - Procedures
 - Data Types
 - INFORMATION_SCHEMA

Project - General | DataSource

Name | DataSource

Bookmarks

```
USE WAREHOUSE COMPUTE_WH;
SELECT L_ORDERKEY, L_PARTKEY, L_SUPPKEY, L_LINENUMBER, L_QUANTITY, L_EXTENDEDPRICE, L_DISCOUNT, L_TAX, L_RETURNFLAG, L_LINESTATUS
FROM EPAM_LAB.DEMO_DB.H_LINEITEM
LIMIT 100;
```

L_ORDERKEY	L_PARTKEY	L_SUPPKEY	L_LINENUMBER	L_QUANTITY	L_EXTENDEDPRICE
2268648	181.644	16.672	3	46	79.31
2269445	378.115	8.152	3	19	22.6
2269729	40.953	10.958	1	19	35.8
2269729	287.656	2.699	2	31	50.9
2270024	225.479	513	6	30	42.1
2270755	395.124	144	1	10	12.1
2270883	149.604	14.612	1	49	81.6
2271014	13.067	18.068	1	17	16.6
2272072	169.031	19.048	1	5	5.5
2274945	27.888	12.890	2	13	23.6

5. Snowflake SQL

Created two warehouses of different sizes

Warehouses

2 Warehouses

NAME ↑	STATUS	SIZE
COMPUTE_WH	Started	X-Small
COMPUTE_WH_XL	Started	X-Large

Used the following command to disable cache results:

```
ALTER SESSION SET USE_CACHED_RESULT=FALSE;
```

Interesting, as expected XS warehouse has longer time of execution than XL warehouse, but it has shorter time of compilation than XL.

select l_orderkey, sum(l_extendedprice * (...	01a86e72-320...	Success	YERZHAN	Compilation	326ms	1.3s
select l_orderkey, sum(l_extendedprice * (...	01a86e71-3200...	Success	YERZHAN	Queued provisioning	80ms	1.1s
select l_orderkey, sum(l_extendedprice * (...	01a86e70-320...	Success	YERZHAN	Execution	694ms	2.2s
ALTER SESSION SET USE_CACHED_RESULT=FALSE;	01a86e70-320...	Success	YERZHAN	Total	1.1s	31ms
select l_orderkey, sum(l_extendedprice * (...	01a86e6c-320...	Success	YERZHAN	COMPUTE_WH_XL		31ms

Snowflake leverages cache, here are the same query with and without cache.

The duration is around 15 times smaller.

SQL TEXT	QUERY ID	STATUS	USER	WAREHOUSE	DURATION
select l_orderkey, sum(l_extendedprice * (...	01a86e64-320...	Success	YERZHAN	COMPUTE_WH	116ms
select l_orderkey, sum(l_extendedprice * (...	01a86e63-320...	Success	YERZHAN	COMPUTE_WH	1.7s

Performance comparison 3NF vs Star Schema

```
SELECT epam_lab.core_dwh.h_nation.n_name,  
       sum(l_extendedprice * (1 - l_discount)) as revenue  
FROM epam_lab.core_dwh.h_lineitem,  
     epam_lab.core_dwh.h_orders,  
     epam_lab.core_dwh.h_customer,  
     epam_lab.core_dwh.h_nation  
WHERE l_orderkey = o_orderkey  
      and l_orderkey = o_orderkey  
      and l_shipdate <= date '1998-12-01'  
      and o_custkey = c_custkey  
      and c_nationkey = n_nationkey  
group by  
       n_name
```

```
order by
    revenue desc;

SELECT epam_lab.data_mart.h_orders.n_name,
    sum(l_extendedprice * (1 - l_discount)) as revenue
FROM epam_lab.data_mart.h_lineitem,
    epam_lab.data_mart.h_orders
WHERE l_orderkey = o_orderkey
    and l_orderkey = o_orderkey
    and l_shipdate <= date '1998-12-01'
group by
    n_name
order by
    revenue desc;
```

The execution duration of query is faster for the star schema, 1.2s versus 1.6s.

SQL TEXT	QUERY ID	STATUS	USER	Query Duration	DURATION
SELECT epam_lab.core_dwh.h_nation.n_name, ...	01a86e99-320...	Success	YERZHAN	Compilation 388ms	2.2s
SELECT epam_lab.data_mart.h_orders.n_name,...	01a86e96-320...	Success	YERZHAN	Queued provisioning 73ms	1.6s
SELECT epam_lab.data_mart.h_orders.n_name,...	01a86e95-320...	Failed	YERZHAN	Execution 1.2s	35ms
SELECT epam_lab.data_mart.h_orders.n_name,...	01a86e90-320...	Failed	YERZHAN	Total 1.6s	23ms

6. Other SnowFlake Features


Object Cloning

```
CREATE TABLE epam_lab.data_mart.h_orders_dev CLONE epam_lab.data_mart.h_orders;
```

It is very fast, less than 1 second.

Time Travel

```
DROP TABLE epam_lab.data_mart.h_orders_dev;
SELECT * FROM epam_lab.data_mart.h_orders_dev LIMIT 10;
```



002003 (42S02): SQL compilation error:

Object 'EPAM_LAB.DATA_MART.H_ORDERS_DEV' does not exist or not authorized.

```
UNDROP TABLE epam_lab.data_mart.h_orders_dev;
```

	status
1	Table H_ORDERS_DEV successfully restored.

```
SELECT * FROM epam_lab.data_mart.h_orders_dev LIMIT 10;
```

	O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE
1	10,702,467	114,445	F	197,071.8303	1993-06-25
2	10,702,469	185,380	F	192,008.179	1993-06-25
3	10,702,470	160,526	F	325,258.8345	1993-06-26
4	10,702,471	57,496	F	6,239.751	1993-06-26
5	10,702,500	168,340	F	277,266.9072	1993-06-27
6	10,702,504	185,531	F	231,749.9791	1993-06-28
7	10,702,533	13,877	F	74,401.1625	1993-06-29

Data Sharing

Created reader account

```
create managed account reader_acct
  admin_name = Yerzhan , admin_password = '****',
  type = reader;
```

status
{"accountName":"AH56885","loginUrl":"https://ah56885.ca-central-1.aws.snowflakecomputing.com"}

Shared data with reader account

Share Data

Sharing as ACCOUNTADMIN

EPAM_LAB

4 Tables

Secure Share Identifier

A secure share that packages the data you selected will be created. [Learn More.](#)

EPAM_LAB_SNOWFLAKE_SECURE_SHARE_1668938481977

Allowed characters: A-Z, 0-9, \$, _

Description (optional)

Add accounts in your region by name

R

READER_ACCT

AWS - Canada (Central)

×

Cancel

Create Share

<div>YS Yerzhan Sapenov</div> <div>ACCOUNTADMIN</div> <div>Worksheets</div> <div>Dashboards</div> <div>Data</div> <div>Databases</div>	Shared With Me Shared By My Account Reader Accounts		
	<div> <div>Q Search</div> <div>Sha</div> </div>		
	TITLE	SHARED WITH	CREATED ↑
	EPAM_LAB_SNOWFLAKE_SECUR...	DQXLFJQ.READER_ACCT	5 minutes ago
		DATA	
		EPAM_LAB	

7. Snowpipe

Created Storage Queue

snowflakestorage888 Queues					
Storage account					
<div> <div>Search</div> <div> <div>Queue</div> <div>Refresh</div> <div>Delete</div> </div> </div>					
<div>Overview</div> <div>Activity log</div> <div>Tags</div>	<div>Search queues by prefix</div> <table> <tr> <th>Queue</th><th>Url</th></tr> <tr> <td> <input type="checkbox"/> queuesnowflake </td><td>https://snowflakestorage888.queue.core.windows.net/queuesnowflake</td></tr> </table>	Queue	Url	<input type="checkbox"/> queuesnowflake	https://snowflakestorage888.queue.core.windows.net/queuesnowflake
Queue	Url				
<input type="checkbox"/> queuesnowflake	https://snowflakestorage888.queue.core.windows.net/queuesnowflake				

Registered Microsoft.EventGrid for subscription

Created Event Subscription

Created an Integration in Snowflake

```
create notification integration SNOWPIPE_DEMO_EVENT
enabled = true
type = queue
notification_provider = azure_storage_queue
azure_storage_queue_primary_uri = 'https://snowflakestorage888.queue.core.windows.net/queuesnowflake'
azure_tenant_id = 'b41b72d0-4e9f-4c26-8a69-f949f367c91d';
```

Permission granted for Snowflake by Azure through URL from command

```
DESC notification integration SNOWPIPE_DEMO_EVENT;
```

Granted Snowflake access to the Storage Queue

4 items (3 Users, 1 Service Principals)				
<input type="checkbox"/>	Name	Type	Role	Scope
Storage Queue Data Contributor				
<input type="checkbox"/>	SnowflakePACInt0692	App	Storage Queue Data Contributor	This resource
			ⓘ	Add

Created Snowpipe

```
CREATE OR REPLACE pipe EPAM_LAB.PUBLIC.SNOWPIPE_ORDERS
auto_ingest = true
integration = 'SNOWPIPE_DEMO_EVENT'
as
COPY INTO
FROM (
```

```
SELECT t.$1, t.$2, t.$3, replace(t.$4, ',', ''),
       TO_DATE(t.$5, 'DD.MM.YY'), t.$6, t.$7, t.$8, t.$9
FROM @epam_lab.public.azureblobstage AS t)
file_format = ( format_name= epam_lab.public.DSVFILEFORMAT );
```

Counted amount of rows in epam_lab.core_dwh.h_orders

COUNT(*)
3,000,000

Uploaded file to Azure Blob Storage with 5 lines.

COUNT(*)
3,000,005


Good step-by-step tutorial [link](#)

8. Additional tasks

In Partner Connect it is possible to use partner applications

Connected to Google drive folder and started sync through Fivetran

✓ Connect source — ✓ Snowflake connected — 3 Complete initial sync



EPAM_3367250862644742496
SnowflakePC

Google Drive
google_drive

Status Logs Schema Usage Setup Connected


Your connector is currently syncing data...

We'll notify you by e-mail once your initial historical sync is complete. Your free trial while you wait? [Add another connector.](#)

↻ Syncing data - First time syncing data for this connector
↻ Historical sync


Connectors 1

Transformations

Destination 

Logs

Explored Matillion



MATILLION

1 million free rows every month
View breakdown →

Start Enterprise trial Talk to sales Upgrade

Pipelines

Agents

Manage

Destinations

Passwords

Cloud credentials


OAuths

Manage Destinations

All destinations ▾

Search

↑ Destinations (1)


SnowflakeDestination
