**Snowflake Architecture**

What’s unique with snowflake architecture?

How snowflake architecture is different from traditional data bases(RDBMS like SQL Server, Oracle, My SQL) ?

What’s your understanding of Snowflake architecture?

Tell me about snowflake architecture?

It's a hybrid of shared-disk and shared-nothing architecture

Basically its Multi cluster shared data architecture. Storage and compute are separate and can be scaled independently to any extent and also it charges separately for storage and compute

(i.e. storage and compute are decoupled)

It’s 3 layer architecture

Storage layer => that stores data in hybrid columnar format and saved in cloud

Query Processing => Virtual Warehouse to run queries/for compute resources (muscle)

Cloud Services => brain of snowflake architecture coordinates between storage and query processing layers

/\*Data is stored in the cloud storage and works as a shared-disk model

(i.e. centralized data store like RDBMS)

Compute it will take advantage of performance and scale-out benefits of shared nothing architecture

(i.e. compute is distributed like hadoop) \*/

Why to go with Snowflake instead of other Cloud Data warehouses like AWS Redshift, Azure Synapse, Google Big query?

Snowflake is available on all cloud platforms like AWS, Azure and GCP (Multi cloud). Other cloud Data warehouses were restricted to that one cloud example: Redshift will be available in only AWS, Big Query will be available in GCP.

Unique features like Data Market place, Zero Copy Clone, Time travel, Data sharing, Streams for CDC

Separate Storage and compute costs, Zero Administration/Maintenance (partitioning, data backups, index maintenance, performance tuning will be taken care by snowflake itself).No software or hardware required to be installed or upgraded as snowflake is SaaS (Software as a service). Synapse and BigQuery are PaaS (Platform as a service)

How Snowflake charges for Storage and compute?

Snowflake Charges separately for Storage and separately for compute

**Virtual Warehouse(Query Processing)**

How is compute done in Snowflake?

Using virtual warehouses

What is Virtual Warehouse?

Cluster of compute nodes/machines

Cluster => group of nodes/machines connected together

Compute Resources => CPU + Memory+ Temporary storage

Different sizes of warehouses?

XS(1) will have one compute nodes/machines, S(2) will have 2 compute nodes, M(4), L(8), XL(16), 2XL(32), 3XL(64), 4XL(128) -- generally available

5XL(256) and 6 XL(512) are in preview state

What is the default Auto suspend time in a Warehouse?

10 minutes (By default, auto-suspend is enabled. Snowflake automatically suspends the warehouse if it is IDLE for the specified period of time)

What is Auto resume in Virtual Warehouse?

If the warehouse is in suspended state and if we set Auto Resume as true then as soon as a query is ran using this warehouse it will automatically get resumed

(i.e. from suspended state it becomes active)

How many Virtual Warehouses have you used in your project?

ETL team ETL\_S\_WH

ETL\_L\_WH

Reporting\_WH

Datascience\_WH

Testing\_WH

We were using Snowflake\_WH warehouse

Maximum Warehouse size that we used in our project is XL or 2XL

ETL tools => Informatica, Microsoft Onpremises ETL tool SSIS, ADF (azure data factory), Datastage

Reporting tools => Power BI (Microsoft’s reporting tool), Tableau

Reporting\_WH

Testing\_WH

Datascience\_WH

ML\_WH

Can we increase or decrease the warehouse size any time?

Yes on fly (instantly) it will get resized

Can we increase or decrease the warehouse size while warehouse is in suspended?

Yes we can increase/decrease warehouse size irrespective whether your warehouse is in running state or suspended state

Can we start or stop the warehouse any time?

Yes, suspend will stop it and resume will start it again

We were using Snowflake\_WH warehouse for our development

What’s the Maximum Warehouse size that you have used in your project XL?

Can we increase or decrease the warehouse size any time?

Yes on fly (instantly) it will get resized

What will happen to already running queries if we resize the warehouse?

Already running queries will still use the old warehouse size but subsequent queries in queue will run with the resized warehouse

Can we start or stop the warehouse any time?

Yes, suspend will stop it and resume will start it again

What is difference between Scale up and scale out?

Increasing the warehouse size is Scale up and it’s to increase performance of complex queries and for large volumes of data (when remote disk spillage happens) (from S to X)

Scale down => decreasing the warehouse size (for example from L to S)

Adding clusters to a warehouse is scale out and it is to handle concurrency (parallel queries) issues

When to go with scale up, when to with scale out?

to handle long running/complex queries, we will increase the warehouse size i.e scale up

to handle concurrency issues(parallel users), we add clusters to a warehouse i.e scale out

how to set a warehouse to auto scale mode?

While creating a warehouse If you set max\_cluster\_count (10)> min\_cluster\_count(1) then the Virtual warehouse will be in auto scale mode

it will start with minimum number of clusters and scale out to maximum clusters based on workloads on demand

**Maximized Mode**=> max\_cluster\_count(5)= min\_cluster\_count (5)

Snowflake starts all the clusters so that maximum resources are available while the warehouse is running.

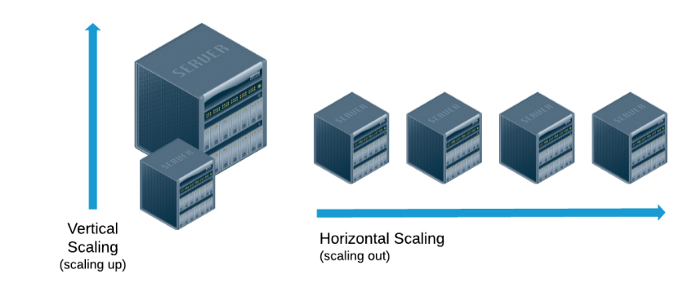
what is horizontal scaling and vertical scaling?

Scale up/scale down=> Horizontal scaling

Scale out/scale in=> Vertical scaling

What is the maximum clusters that can be added in case of scale out?

Maximum 10 clusters can be added



Can we automate scale up? No, we have to manual resize(increase/decrease) the warehouse

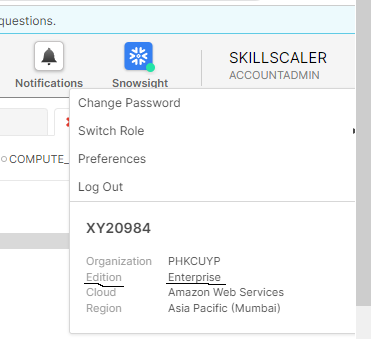
Scaling out => automated process taken care by snowflake based on your work load

**Scaling policy is used to determine when to start or shut down a warehouse.**

**Standard** (default policy) – performance (starts immediately when a query is queued)

**Economy** -- conserves credits (cluster starts only if it estimates there is enough query load to keep the cluster busy for at least 6 minutes)

Where can we see the edition/cloud/region details of your snowflake account?

right top when we expand on User and Role tab

our edition details will be available in service agreement also or we can see it when we create reader accounts if we have account admin access

Table Types

What’s are different types of tables

Permanent, Temporary, transient and external tables

What’s different between permanent, transient and temporary tables?

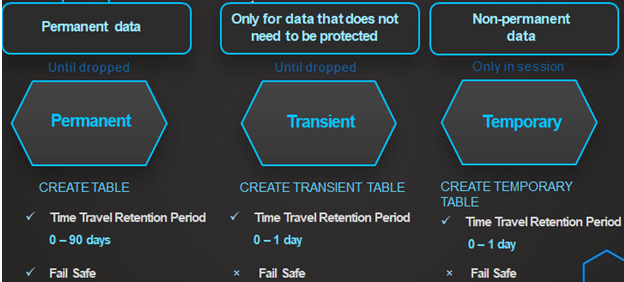
Retention period=> 0-90 days for permanent tables, 0-1 days for temporary and transient tables

Fail safe=> not available for temporary and transient tables

Scope=> until explicitly dropped for permanent and transient tables, Available only for that session for temporary tables

What is time travel in snowflake?

Time travel enables accessing historical data (i.e. data that has been changed or deleted) at any point within a defined period based on retention period. If we drop a table, with time travel we can restore it if it is with in retention period



What’s the maximum retention period for permanent tables?

Standard edition => 1 day

Enterprise edition and above => 90 days

What’s the maximum retention period for temporary tables and transient tables? 1 day

What’s the default retention period in snowflake? 1 day

What is failsafe?

Fail-safe offers protection of data in case of disaster/hardware failure, manages 7 days of historical data by snowflake

Can we configure failsafe period?

Its Non-configurable 7 days period and can be defined only for permanent

Can we disable Fail-safe in Snowflake?

Fail safe cannot be disabled on a permanent table

Transient

Can we interact/read/access data from in failsafe?

No user interaction and recoverable only by snowflake, We need to approach snowflake team to get data from failsafe

If you change the retention period at the account level, all databases, schemas, and tables that do not have an explicit retention period automatically inherit the new retention period

Account level = 1 day

Retention period at database level => 10 days

Schema => 15 days

For one table under this database I have set retention period as 30 days=> it will have 30 days of time travel

Another table I’m not setting any retention period at table level => it takes next level hierarchy I.e schema level retention period I.e 15 days

What are the differences between Time Travel and Fail-safe?

Time travel: Helps the user to query the data, how does it looks some time before and we can restore the previous state of the table.

Fail safe: internally used by snowflake to restore the data during the times of hardware failure/disaster, user can't access/read data directly from Fail safe we will have to approach snowflake team

After Time travel period is over, next 7 days data will be stored in Snowflake

Will time travel contribute to additional storage costs?

Yes, since it has to maintain snapshots/historical data

Will failsafe contribute to additional storage costs?

Yes, since it has to maintain historical data for 7 days

Which scenario did you use time travel? Restore data that got accidentally deleted using UNDROP, Backing up of data from key points in the past i.e data a week back, month back. Query data in the past that has since been updated or deleted.

Why Fail-safe?

Although data is backed up at regular intervals, there can be situations between the two back-ups during which the data has been changed accidentally or deleted due to some mistake. So the fail-safe provides a quick and efficient way to recover such data with minimal cost and minimal effort.

What are the different ways of getting/accessing/querying data from Time travel? It can be done in 3 ways

1. **By timestamp :** check data at a specific point in time

SELECT \* FROM my\_table **AT(timestamp => 'Mon, 01 May 2021 08:00:00 -0700'::timestamp\_tz);**

SELECT \* FROM my\_table **BEFORE(timestamp => 'Mon, 01 May 2021 08:00:00 -0700'::timestamp\_tz);**

1. **By offset →** In this example, we select the historical data from a table as of 15 minutes ago:

SELECT \* FROM my\_table **AT(offset => -60\*15);**

SELECT \* FROM my\_table **BEFORE(offset => -60\*15);**

1. **By query statement ID** → We can see the Query Statement ID in the history tab

SELECT \* FROM my\_table **BEFORE(STATEMENT => '8e5d0ca9-005e-44e6-b858-a8f5b37c5726');**

SELECT \* FROM my\_table **AT(STATEMENT => '8e5d0ca9-005e-44e6-b858-a8f5b37c5726');**

Accessing data with **AT => Data at that point**

Accessing data with **BEFORE => Data before that point**

Failsafe period starts immediately after Time travel ends

Retention period as 90 days for a table, we will have 90 days of historical data in time travel another 7 days data in fail safe

Initially for 10 days retention period/timetravel => 10 days of data will be there in Time travel and 7 days of 11 -17 of historical data will be there in failsafe

After a month=> 10 days of historical data in time travel, 11 -17 days of historical data will be there in fail safe

1 day 2nd to 8th day

1. If we have a table with a 10-day retention period/time travel and you decrease the retention period to 1-day, data from days 2 to 8 will be moved into Fail-safe, leaving only the data of day 1 accessible through Time Travel

9 AM

1. 1 day => 10 days

May 2nd

Mar 3rd

Mar 4th

Initially I have 1 day of retention period on a table and you increased it to 8 days then immediately we will not able access historical data from time travel. As days progress you will be able to access historical data for more days.

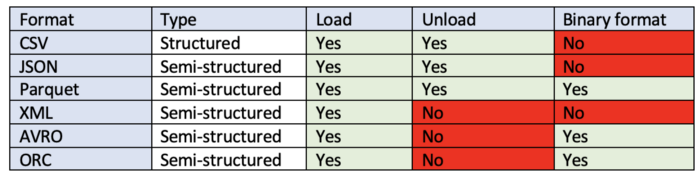
Through WebUI/Wizard also we can load data but we didn’t use it much in our projects) – Maximum 50 MB files can only be loaded using UI

**File formats:**

Named file format object describes and stores the format information like delimiter of the file, skip header, file type and other properties

Delimiter=> Separator (column/row)

Snowflake  supports most of the common file formats used for loading data. These file formats include:



Delimited files (CSV files any valid delimiter is supported; the default is a comma)

How Data will be organized in Data lake?

**s3://bucketname/databasename/schemaname/tablename/date**

Set up connection

Establish connection

How will you connect from snowflake to AWS S3/Azure blob/Google cloud storage?

1. We can create external stage and pass s3 path url and AWS key Access key and secret key
2. Create storage integration and pass AWS IAM ROLE ARN and s3 path or url and use that storage integration in external stage (we are going to cover storage integration option in later sessions)

Data Loading Options

1. Bulk Data loading with Copy Command
2. Continuous data loading with Snowpipe

(Through WebUI/Wizard also we can load data but we didn’t use it much in our projects) – Maximum 50 MB files can only be loaded using UI)

Steps involved in Data loading:

1. Create Database, schema, table
2. Create File Format
3. Create Stage
4. Load data using Copy command/Snowpipe
5. Schedule loads using tasks/scheduling tools like Control M, Orchestration tools like Airflow

As soon as you load data into snowflake, it gets encrypted and compressed

Encryption=> way of translating data for Data Security purpose

Compression=> **reduces** file size (by default gets GZIP compressed)

Can we reload a file that is already loaded into a table using copy command?

No, because based on filename, Metadata information will stored for 64 days

If you have to reload the same file (same name) again, how can we do that?

In copy command, we will specify FORCE=TRUE

CREATE [ OR REPLACE ] FILE FORMAT [ IF NOT EXISTS ] <name>

                      TYPE = { CSV | JSON | AVRO | ORC | PARQUET | XML } [ formatTypeOptions ]

                      [ COMMENT = '<string\_literal>' ]

formatTypeOptions ::=

-- If TYPE = CSV

     COMPRESSION = AUTO | GZIP | BZ2 | BROTLI | ZSTD | DEFLATE | RAW\_DEFLATE | NONE

     RECORD\_DELIMITER = '<character>' | NONE

     FIELD\_DELIMITER = '<character>' | NONE

     FILE\_EXTENSION = '<string>'

     SKIP\_HEADER = <integer>

     SKIP\_BLANK\_LINES = TRUE | FALSE

     DATE\_FORMAT = '<string>' | AUTO

     TIME\_FORMAT = '<string>' | AUTO

     TIMESTAMP\_FORMAT = '<string>' | AUTO

     BINARY\_FORMAT = HEX | BASE64 | UTF8

     ESCAPE = '<character>' | NONE

     ESCAPE\_UNENCLOSED\_FIELD = '<character>' | NONE

     TRIM\_SPACE = TRUE | FALSE

     FIELD\_OPTIONALLY\_ENCLOSED\_BY = '<character>' | NONE

     NULL\_IF = ( '<string>' [ , '<string>' ... ] )

     ERROR\_ON\_COLUMN\_COUNT\_MISMATCH = TRUE | FALSE

     REPLACE\_INVALID\_CHARACTERS = TRUE | FALSE

     VALIDATE\_UTF8 = TRUE | FALSE

     EMPTY\_FIELD\_AS\_NULL = TRUE | FALSE

     SKIP\_BYTE\_ORDER\_MARK = TRUE | FALSE

     ENCODING = '<string>' | UTF8

If comma is defined as column delimiter and if we get extra comma in the data how to handle this scenario?

Add ESCAPE\_UNENCLOSED\_FIELD and FIELD\_OPTIONALLY\_ENCLOSED\_BY as single quotes, double quotes in file format options

Snowflake stores all data internally in the UTF-8 character set, IN SQL Server it will be in UTF-16 encoding so while migrating from there you might face issue so change it to UTF-16

Set up connection

Establish connection

How will you connect from snowflake to AWS S3/Azure blob/Google cloud storage?

1. We can create external stage and pass s3 path url and AWS key Access key and secret key
2. Create storage integration and pass AWS IAM ROLE ARN and s3 path or url and use that storage integration in external stage (it will be discussed in our further sessions)

Can we reload a file that is already loaded into a table using copy command?

No, because based on filename, Metadata information will stored for 64 days

If you have to reload the same file (same name) again, how can we do that?

In copy command, we will specify FORCE=TRUE

How to handle new columns being added in the file and existing columns removed from file?

**ERROR\_ON\_COLUMN\_COUNT\_MISMATCH = TRUE | FALSE**

If set to FALSE, an error is not generated and the load continues. If the file is successfully loaded:

If the input file contains records with more fields than columns in the table, the matching fields are loaded in order of occurrence in the file and the remaining fields are not loaded.

If the input file contains records with fewer fields than columns in the table, the non-matching columns in the table are loaded with NULL values.

Can we read data directly from S3 bucket/External/Internal Stage?

With out loading data into snowflake can we directly read data from the file that is there in S3 bucket?

Select $1,$2,$3 from @my\_external\_stage/dataload/emp.csv.gz (file\_format => 'my\_csv\_format')

Select $1,$2,$3 from @my\_internal\_stage/dataload/emp.csv.gz (file\_format => 'my\_csv\_format')

Can we set File format at Stage level?

File format can be set at External and Internal stage level, Table level and COPY command level.

File format defined at Copy command level takes highest priority, 2nd priority is Stage level file format and last priority will be the file format at table level

What transformations will COPY command supports?

Column reordering (select $1, $4,$2 from @my\_external\_stg)

column omission (avoid loading few fields)

casts using a SELECT statement.

**How can we add name of the file from which data got loaded to the table?**

Using META$FILENAME in COPY command as shown below

copy into emp (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO,FILENAME) from (Select $1,$2,$3,$4,$5,$6,$7,$8,METADATA$FILENAME from @my\_external\_stg/employee/ (file\_format => 'my\_csv\_format')) FORCE=TRUE;

How to skip the file as soon as error is encountered?

ON\_ERROR= SKIP\_FILE

ON\_ERROR = { CONTINUE | SKIP\_FILE | SKIP\_FILE\_<num> | SKIP\_FILE\_<num>% | ABORT\_STATEMENT } --

SKIP\_FILE\_num (e.g. SKIP\_FILE\_10)

Skip a file when the number of error rows found in the file is equal to or exceeds the specified number.

SKIP\_FILE\_num% (e.g. SKIP\_FILE\_10%)

Skip a file when the percentage of error rows found in the file exceeds the specified percentage.

What is the default on\_error option for COPY command in case if you are not specifying it?

-Default option will be ABORT STATEMENT

How to check if file has got any errors before loading the files to Snowflake tables?

Validation mode= return\_errors| return\_all\_errors| return\_n\_rows

How to capture error records into error table during data loads? – Validate table function

How to continue load even in case of error ?

ON\_ERROR = CONTINUE in copy command, file will get partially loaded

insert into emp\_err\_log select \* from table(validate(emp, job\_id=>'\_last'));

where can we see bulk data load history in snowflake?

-- information\_schema.copy\_history table function stores copy history information for 14 days including snowpipe and copy commands

select \* from table(information\_schema.copy\_history(table\_name=>table\_name , start\_time=>date&time));

-- information\_schema.load\_history view stores copy history for 14 days and view does not return the history of data loaded using Snowpipe

select \* from information\_schema.load\_history order by last\_load\_time desc limit 10;

-- snowflake.account\_usage.copy\_history will have 365 days of history

select \* from snowflake.account\_usage.copy\_history order by last\_load\_time desc limit 10;

where can we query details that got executed in your account in last 365 days?

select \* from snowflake.account\_usage.QUERY\_HISTORY order by start\_time desc;

in History tab we will see latest 14 days of history only

what is the difference between LOAD\_HISTORY and COPY\_HISTORY ?

COPY\_HISTORY includes the information of files that got loaded through Snowpipe and regular copy command both whereas LOAD\_HISTORY includes only information of files that got loaded through regular COPY command but snowpipe

How to see the errors/validate files before loading data into a table?

Using validation\_mode=return\_errors in copy command we can see the errors in those files that COPY command is trying to load

If we use validation mode in copy command it will never load any data into the table It just for validation mode