Snowflake MasterClass

What will you learn?

- Getting Started
- Architecture
- Loading Data
- Copy Options
- Unstructured Data
- Performance
- Load from AWS
- Load from Azure
- Load from GCP
- Snowpipe
- Time Travel
- Fail Safe
- Table Types
- Zero-Copy Cloning
- Data Sharing
- Data Sampling
- Scheduling Tasks
- Visualizations
- Streams
- Materialized views
- Data Masking
- Access Management
- Partner Connect
- Best practices

Contents

Getting Started

How you can get the most out of this course?

- Make use of the udemy tools
- Practice
- Ask questions
- Own pace
- Resources, quizzes & assignements
- Help others

Enjoy learning!

Best practices

- ✓ Pay only for what you use

Make use of the udemy tools

Own pace

Practice

Resources, quizzes & assignements

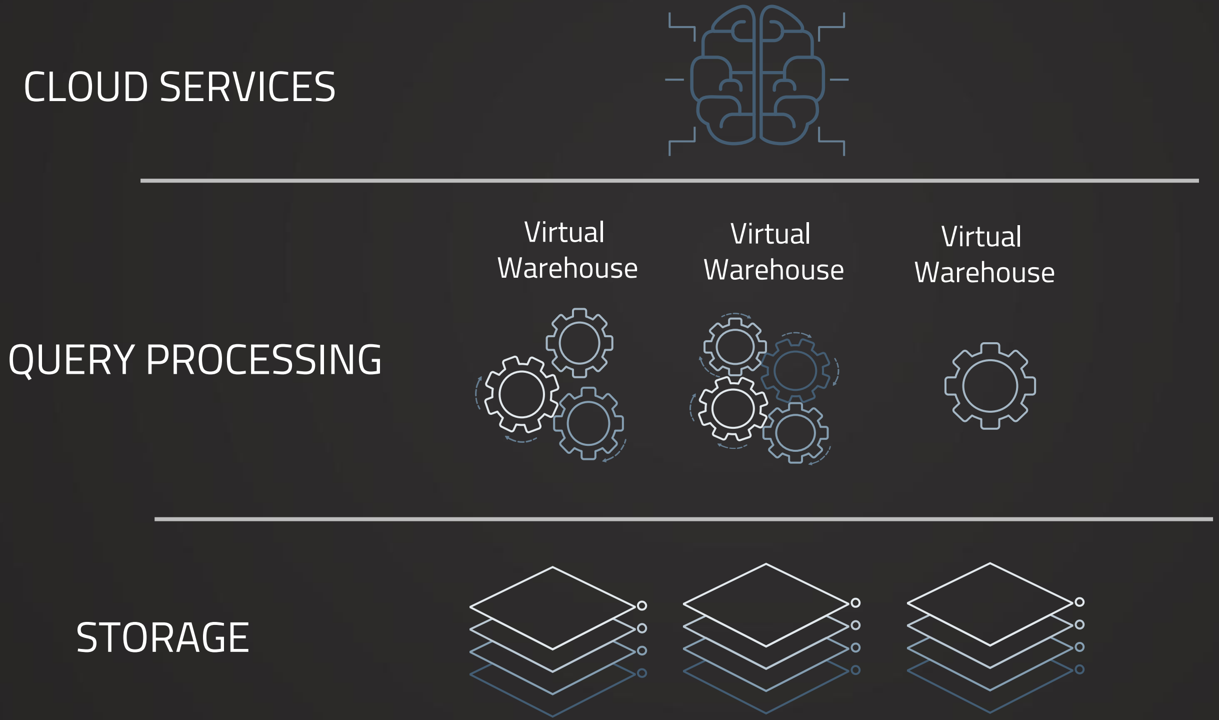
Ask questions

Help others

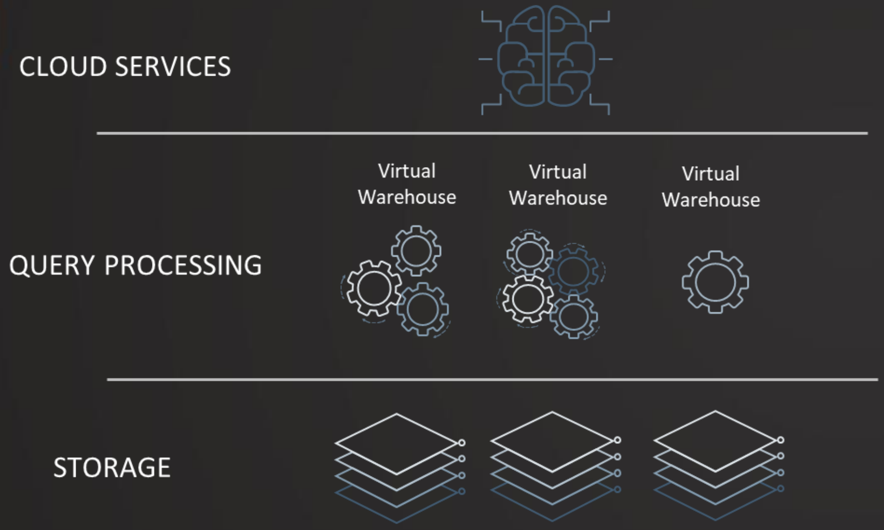
Enjoy learning!

Snowflake Architecture

Snowflake Architecture



Snowflake Architecture



- Brain of the system -
Managing infrastructure, Access control, security, Optimizier, Metadata etc.

- Muscle of the system -
Performs MMP (Massive Parallel Processing)

- Hybrid Columnar Storage -
Saved in blobs

Virtual Warehouse Sizes

XS	1
S	2
M	4
L	8
XL	16
4XL	128

Multi-Clustering

Multi-Clustering

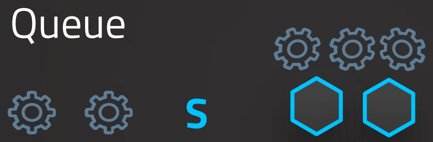
… More queries …

Multi-Clustering

... More queries ...

> Auto-Scaling

Multi-Clustering



Auto-Scaling: When to start an additional cluster?

Scaling policy

Standard

Favors starting additional warehouses

Economy

Favors conserving credits rather than starting additional warehouses

Scaling policy

Policy	Description	Cluster Starts…	Cluster Shuts Down…
Standard (default)	Prevents/minimizes queuing by **favoring starting additional clusters** over conserving credits.	**Immediately when** either a query is **queued** or the system detects that there are more queries than can be executed by the currently available clusters.	**After 2 to 3 consecutive successful checks** (performed at 1 minute intervals), which determine whether the load on the least-loaded cluster could be redistributed to the other clusters
Economy	Conserves credits by favoring keeping running clusters fully-loaded rather than starting additional clusters, *Result*: May result in queries being queued **and taking longer to complete**.	Only if the system estimates there's enough query load to keep the cluster busy for **at least 6 minutes**.	**After 5 to 6 consecutive successful checks …**

Data Warehousing

What is a data warehouse?

What is a data warehouse?

Skip the lecture if you are already familiar with data warehouses

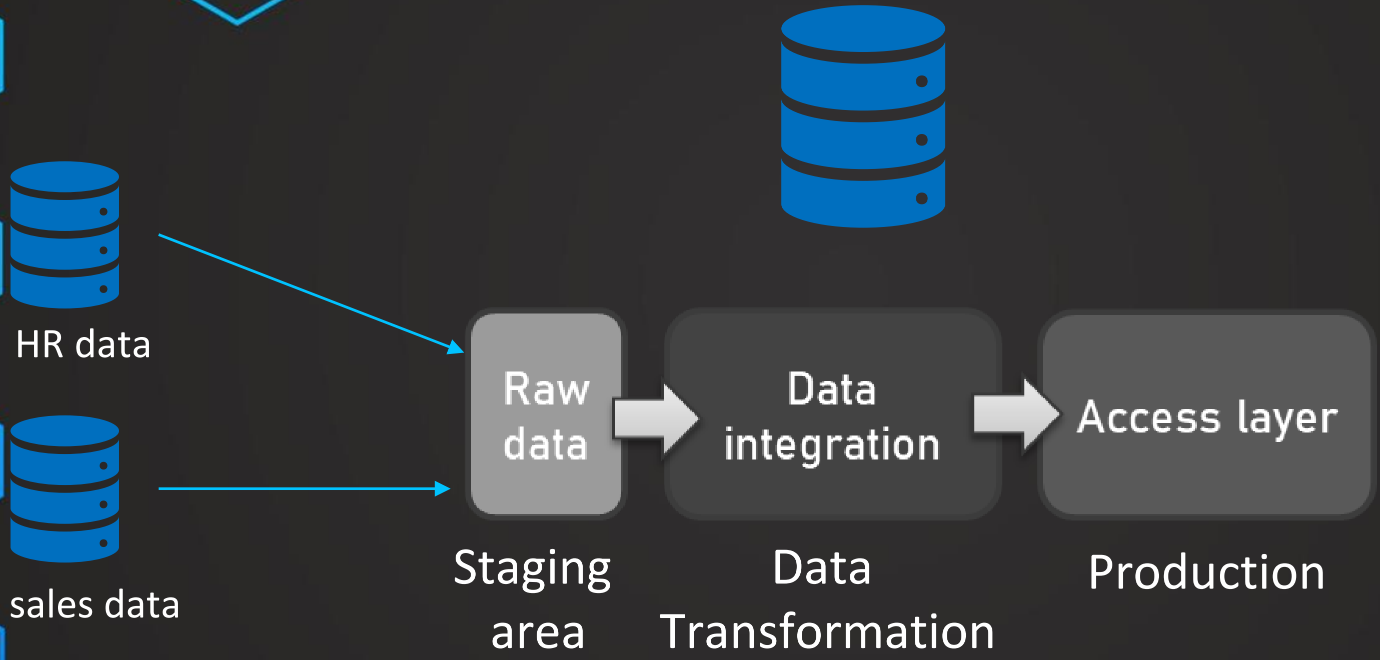
What is a data warehouse?

What is the purpose of a data warehouse?

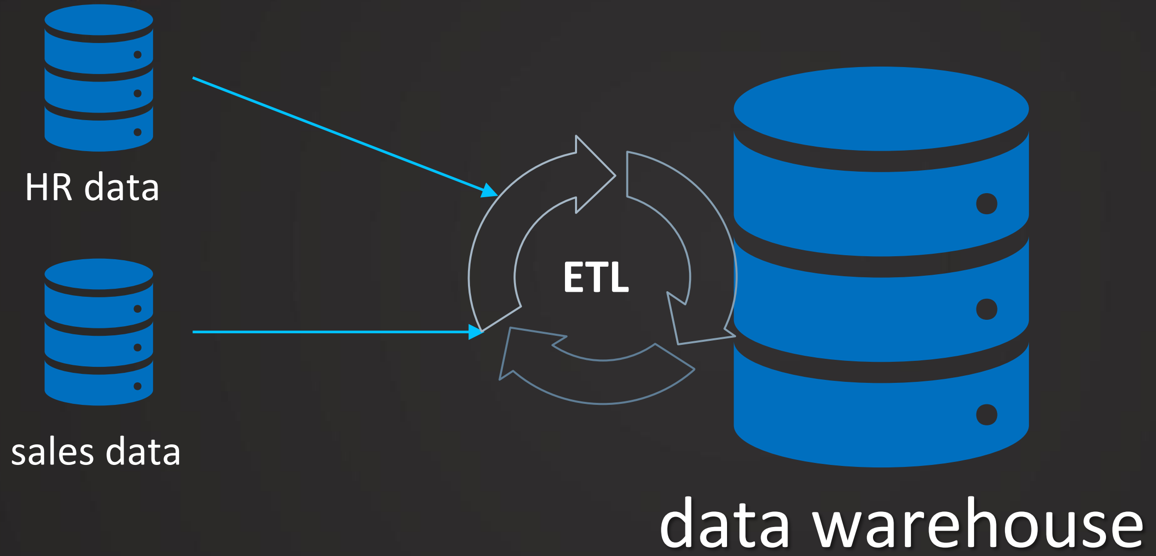
What is a data warehouse?

= Database that is used for reporting and data analysis

Different layers

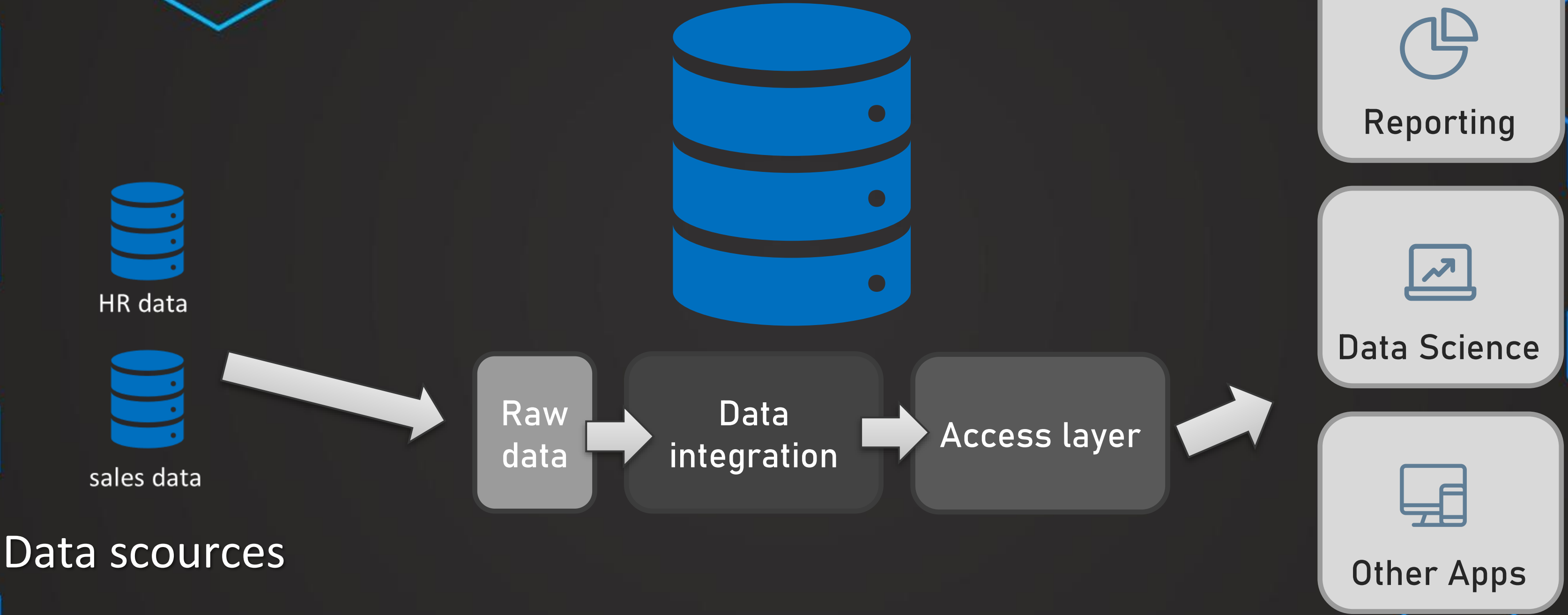


What is a data warehouse?

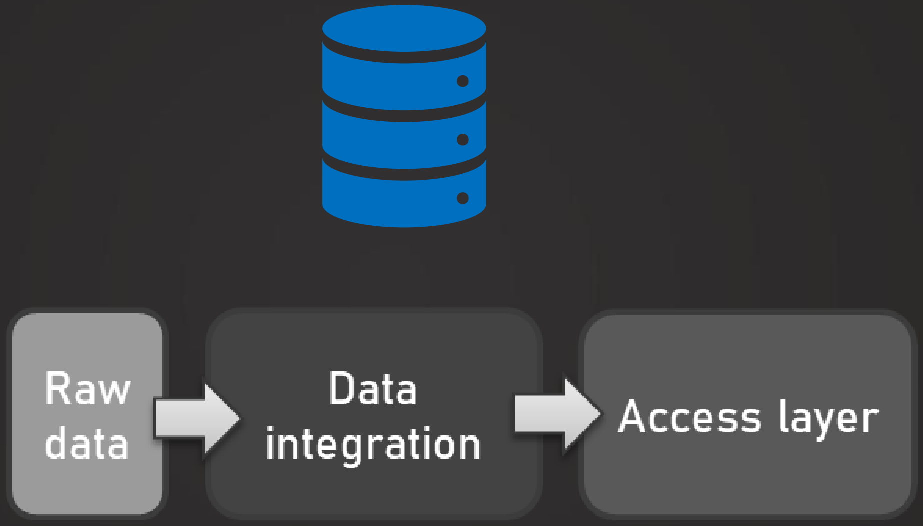


ETL = Extract, Transform & Load

Different layers



Different layers



Staging area

Data Transformation

Cloud Computing

Cloud Computing

Why Cloud Computing?

Cloud Computing

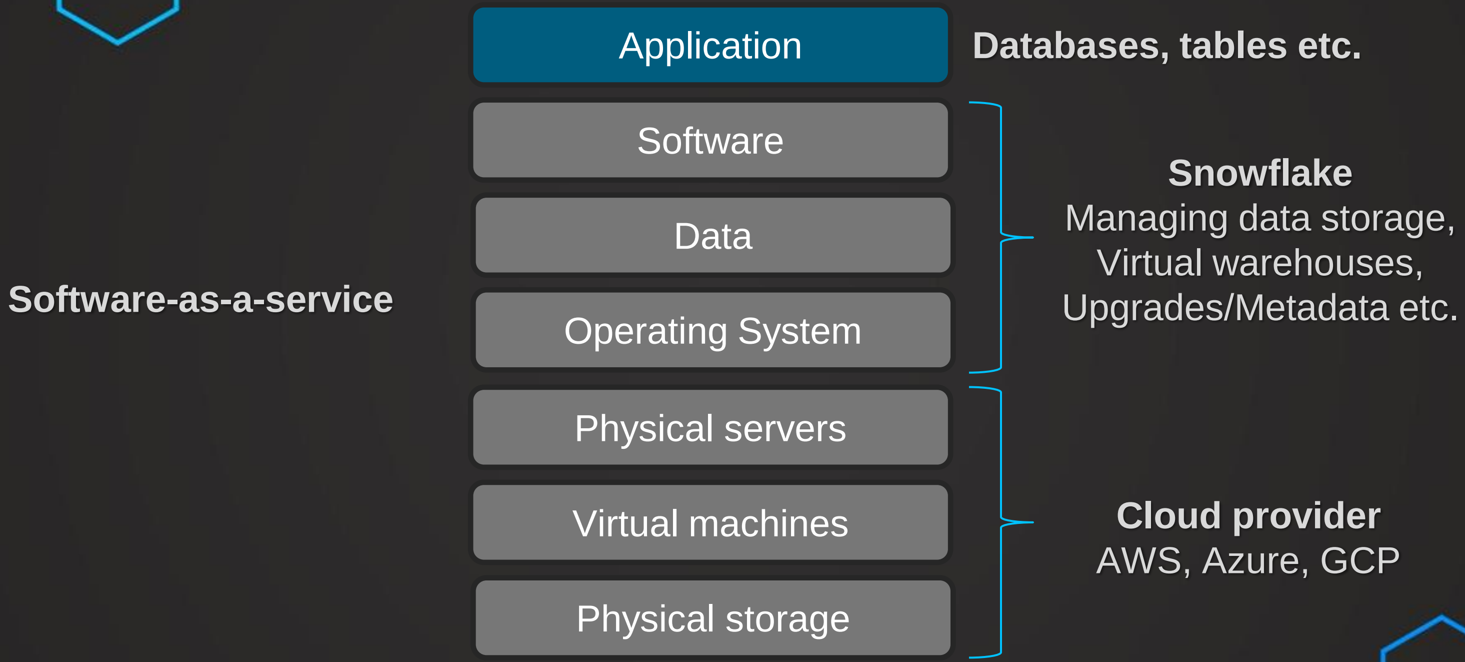
Data Center

MANAGED

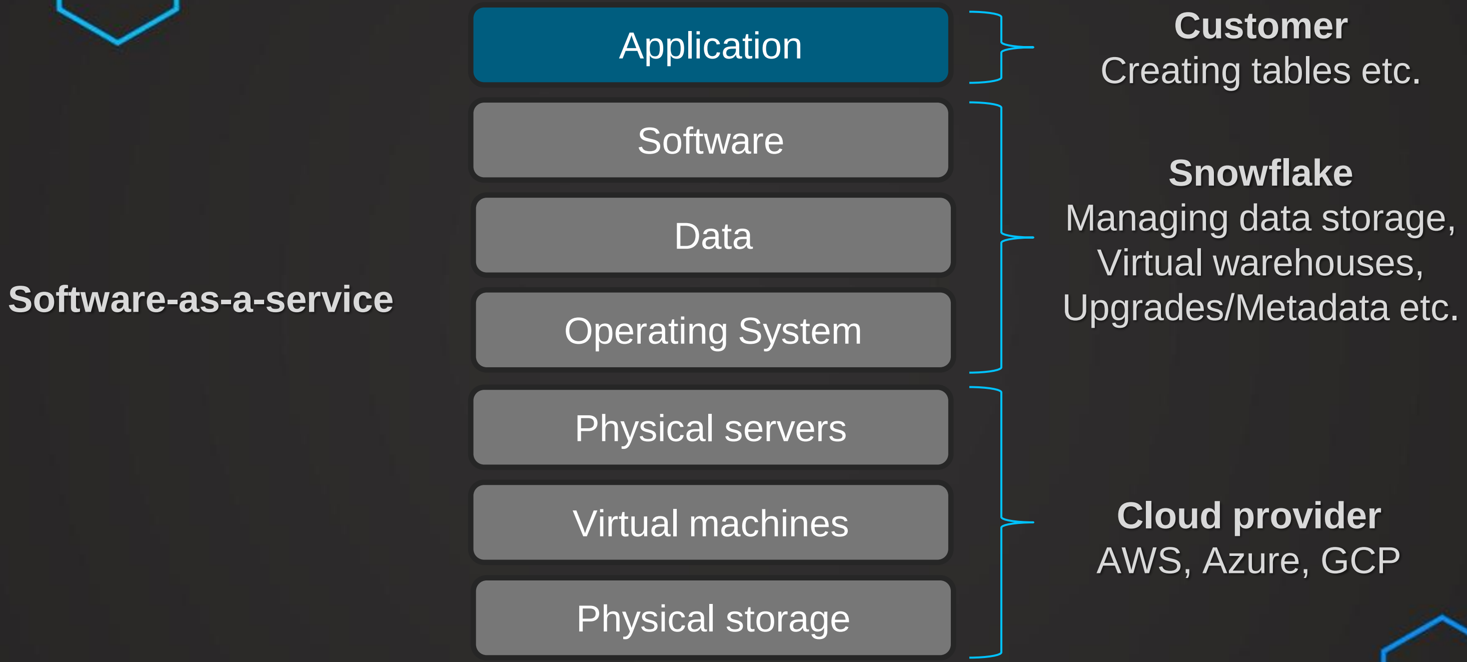
- Infrastucture
- Security
- Electricity
- Software/Hardware upgrades

Software-as-a-Service

Cloud Computing



Cloud Computing



Snowflake Editions

Snowflake Editions

Enterprise

additional features for the needs of large-scale enterprises

Standard

introductory level

Business Critical

even higher levels of data protection for organizations with extremely sensitive data

Virtual Private

highest level of security

Snowflake Editions

Standard

- ✓ Complete DWH
- ✓ Automatic data encryption
- ✓ Time travel up to 1 day
- ✓ Disaster recovery for 7 days beyond time travel
- ✓ Secure data share
- ✓ Premier support 24/7

Enterprise

- ✓ All Standard features
- ✓ Multi-cluster warehouse
- ✓ Time travel up to 90 days
- ✓ Materialized views
- ✓ Search Optimization
- ✓ Column-level security

Business Critical

- ✓ All Enterprise features
- ✓ Additional security features such as Data encryption everywhere
- ✓ Extended support
- ✓ Database failover and disaster recovery

Virtual Private

- ✓ All Business Ciritcal features
- ✓ Dedicated virtual servers and completely seperate Snowflake environment

Snowflake Pricing

Snowflake Pricing

Compute

- ✓ Charged for active warehouses per hour
- ✓ Depending on the size of the warehouse
- ✓ Billed by second (minimum of 1min)
- ✓ Charged in Snowflake credits

Storage

- ✓ Monthly storage fees
- ✓ Based on average storage used per month
- ✓ Cost calculated after compression
- ✓ Cloud Providers

Snowflake Pricing

Compute

Storage

$/€ → Credits

→ *Consumed*

Snowflake Pricing

Standard	Enterprise	Business Critical	Virtual Private
✓ $2.70 / Credit	✓ $4 / Credit	✓ $5.40 / Credit	✓ Contact Snowflake

Region: EU (Frankfurt)

Platform: AWS

Virtual Warehouse Sizes

XS	1
S	2
M	4
L	8
XL	16
4XL	128

Snowflake Pricing

Scenario

On Demand Storage

✓ We think we need 1 TB of storage

Capacity Storage

- Scenario 1: 100GB of storage used

  0.1 TB x $40 = $4

- Scenario 2: 800GB of storage used

  0.8 TB x $40 = $32

- Scenario 1: 100GB of storage used

  1 TB x $23 = $23

- Scenario 2: 800GB of storage used

  0.8 TB x $40 = $23

Region: US East (Northern Virginia)

Platform: AWS

Snowflake Pricing

Scenario

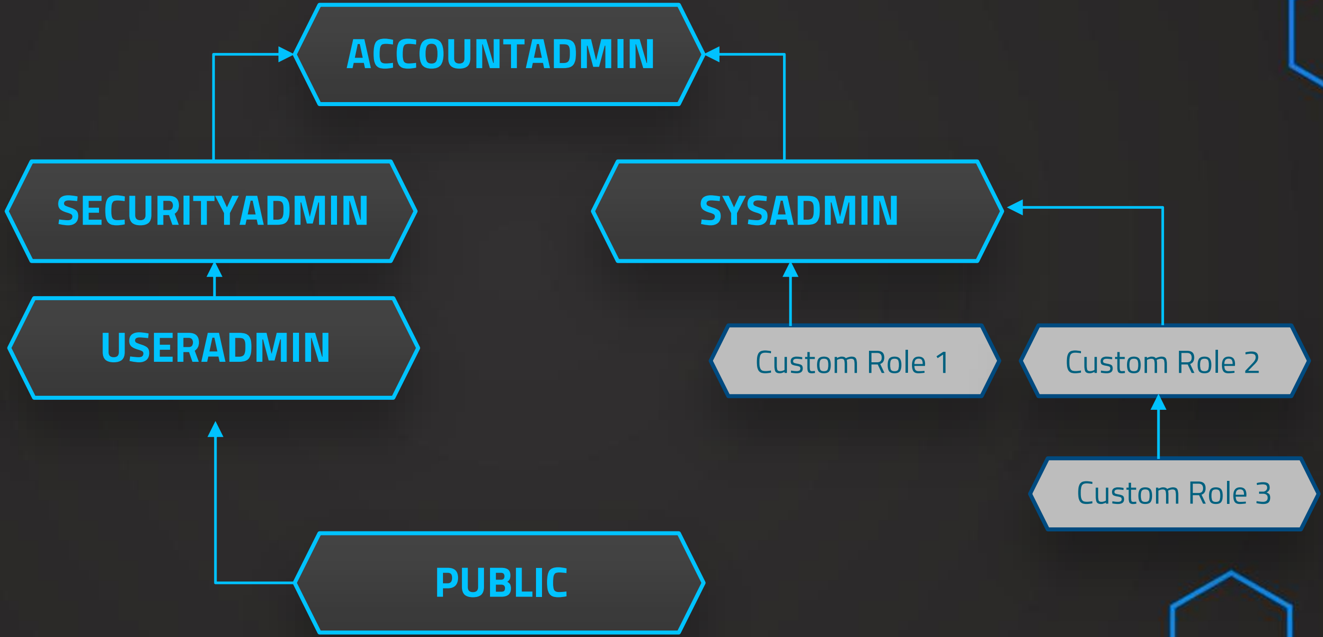
On Demand Storage

Capacity Storage

- ✓ Start with On Demand
- ✓ Once you are sure about your usage use Capacity storage

Snowflake Roles

Snowflake Roles



Snowflake Roles

ACCOUNTADMIN

- ✓ SYSADMIN and SECURITYADMIN
- ✓ top-level role in the system
- ✓ should be granted only to a limited number of users

SECURITYADMIN

- ✓ USERADMIN role is granted to SECURITYADMIN
- ✓ Can manage users and roles
- ✓ Can manage any object grant globally

SYSADMIN

- ✓ Create warehouses and databases (and more objects)
- ✓ Recommended that all custom roles are assigned

USERADMIN

- ✓ Dedicated to user and role management only
- ✓ Can create users and roles

PUBLIC

- ✓ Automatically granted to every user
- ✓ Can create own objects like every other role (available to every other user/role

Loading Data

Loading Data

BULK LOADING

- ✓ Most frequent method
- ✓ Uses warehouses
- ✓ Loading from stages
- ✓ COPY command
- ✓ Transformations possible

CONTINUOUS LOADING

- ✓ Designed to load small volumes of data
- ✓ Automatically once they are added to stages
- ✓ Lates results for analysis
- ✓ Snowpipe (Serverless feature)

Understanding Stages

- ✓ Not to be confused with dataware house stages
- ✓ Location of data files where data can be loaded from

External Stage

Internal Stage

Understanding Stages

External Stage

- ✓ External cloud provider
  - S3
  - Google Cloud Plattform
  - Microsoft Azure
- ✓ Database object created in Schema
- ✓ CREATE STAGE (URL, access settings)

Note: Additional costs may apply if region/platform differs

Internal Stage

- ✓ Local storage maintained by Snowflake

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
copyOptions
```

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
ON_ERROR = CONTINUE
```

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
VALIDATION_MODE =  RETURN_n_ROWS | RETURN_ERRORS
```

- ✓ **Validate the data files instead of loading them**

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
VALIDATION_MODE = RETURN_n_ROWS | RETURN_ERRORS
```

RETURN_N_ROWS (E.G. RETURN_10_ROWS)	VALIDATES & RETURNS THE SPECIFIED NUMBER OF ROWS; FAILS AT THE FIRST ERROR ENCOUNTERED
RETURN_ERRORS	RETURNS ALL ERRORS IN COPY COMMAND

✓ **Validate the data files instead of loading them**

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
SIZE_LIMIT = num
```

- ✓ Specify maximum size (in bytes) of data loaded in that command (at least one file)
- ✓ When the threshold is exceeded, the COPY operation stops loading

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
RETURN_FAILED_ONLY = TRUE | FALSE
```

- ✓ **Specifies whether to return only files that have failed to load in the statement result**
- ✓ **DEFAULT = FALSE**

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
TRUNCATECOLUMNS = TRUE | FALSE
```

- ✓ **Specifies whether to truncate text strings that exceed the target column length**

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
FORCE = TRUE | FALSE
```

- ✓ Specifies to load all files, regardless of whether they've been loaded previously and have not changed since they were loaded
- ✓ Note that this option reloads files, potentially duplicating data in a table

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
TRUNCATECOLUMNS = TRUE | FALSE
```

- ✓ Specifies whether to truncate text strings that exceed the target column length
- ✓ TRUE = strings are automatically truncated to the target column length
- ✓ FALSE = COPY produces an error if a loaded string exceeds the target column length
- ✓ DEFAULT = FALSE

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
SIZE_LIMIT = num
```

- ✓ Specify maximum size (in bytes) of data loaded in that command (at least one file)
- ✓ When the threshold is exceeded, the COPY operation stops loading
- ✓ Threshold for each file
- ✓ DEFAULT: null (no size limit)

Copy Options

```
COPY INTO <table_name>
FROM  externalStage
FILES = ( '<file_name>' ,'<file_name2>')
FILE_FORMAT = <file_format_name>
PURGE = TRUE | FALSE
```

- ✓ specifies whether to remove the data files from the stage automatically after the data is loaded successfully
- ✓ DEFAULT: FALSE

Load unstructured data

Create Stage

Load raw data

Type VARIANT

Analyse & Parse

Flatten & Load

Performance Optimization

Performance Optimization

Make queries run faster

Save costs

Performance Optimization

- ✓ **Add indexes, primary keys**
- ✓ **Create table partitions**
- ✓ **Analyze the query execution table plan**
- ✓ **Remove unnecessary full table scans**

Performance Optimization

- ✓ Add indexes, primary keys
- ✓ Create table partitions
- ✓ Analyze the query execution table plan
- ✓ Remove unnecessary full table scans

How does it work in Snowflake?

- ✓ **Automatically managed micro-partitions**

What is our job?

- ✓ Assigning appropriate data types
- ✓ Sizing virtual warehouses
- ✓ Cluster keys

Performance aspects

Dedicated virtual warehouses

- ✓ Separated according to different workloads

Scaling Up

- ✓ For known patterns of high work load

Scaling Out

- ✓ Dynamically fo unknown patterns of work load

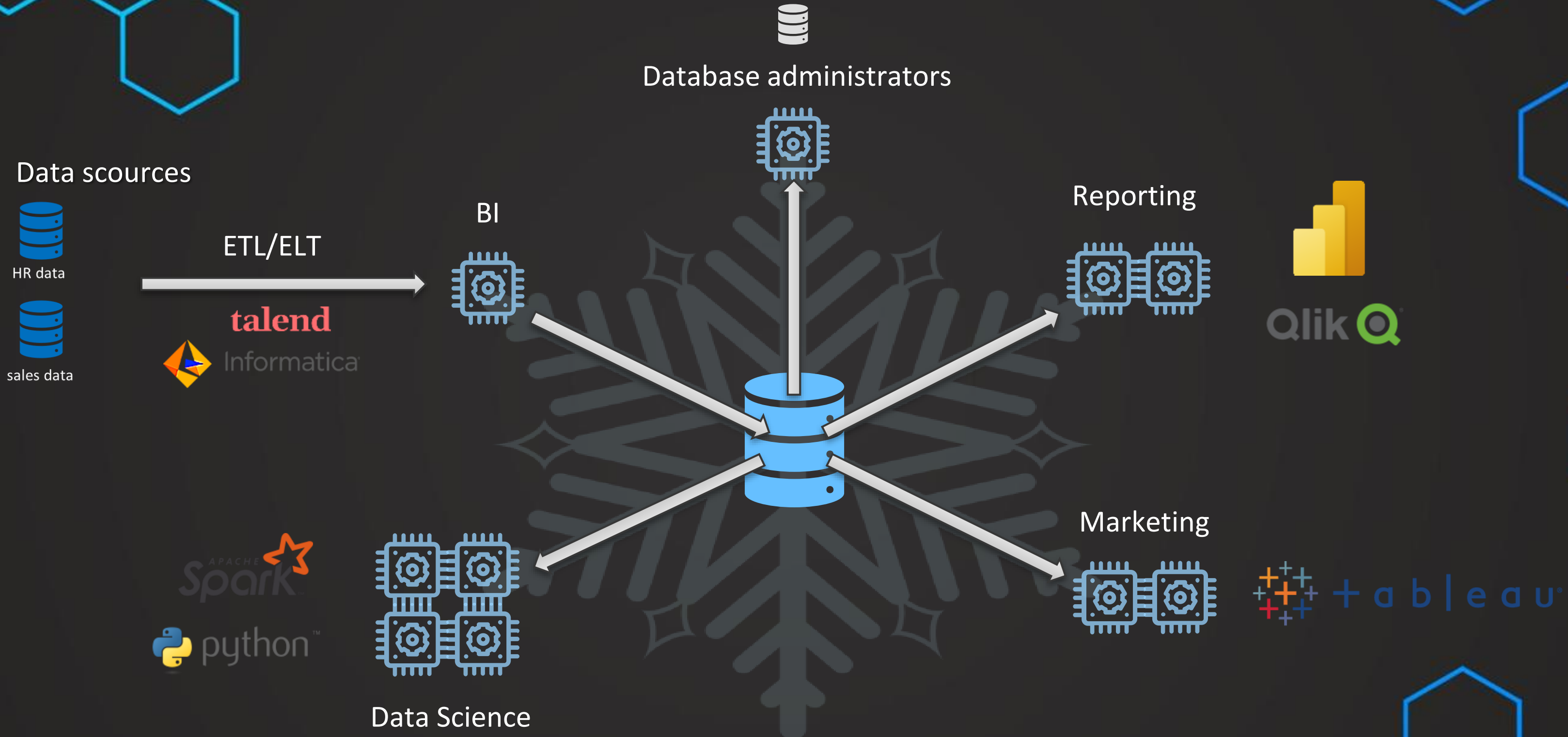
Maximize Cache Usage

- ✓ Automatic caching can be maximized

Cluster Keys

- ✓ For large tables

Dedicated virtual warehouse



Dedicated virtual warehouse

Identify & Classify

- ✓ Identify & Classify groups of workload/users
- ✓ BI Team, Data Science Team, Marketing department

Create dedicated virtual warehouses

- ✓ For every class of workload & assign users

Considerations

Not too many VW

- ✓ Avoid underutilization

Refine classifications

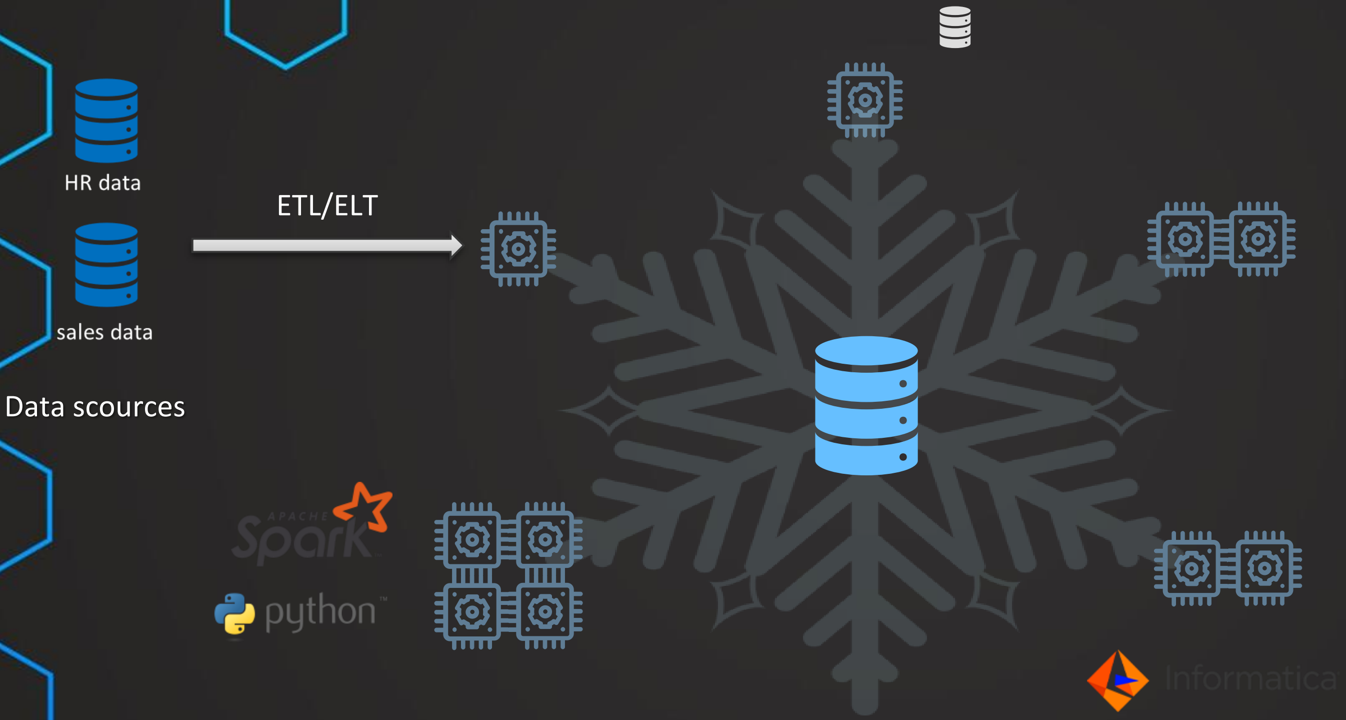
- ✓ Work patterns can change

Considerations

- ✓ If you use at least Entripse Edition *all* warehouses should be Multi-Cluster
- ✓ Minimum: Default should be 1
- ✓ Maximum: Can be very high

Let's practice!

How does it work in Snowflake?



Scaling Up/Down

- ✓ Changing the size of the virtual warehouse depending on different work loads in different periods

Use cases

- ✓ ETL at certain times (for example between 4pm and 8pm)
- ✓ Special business event with more work load

- ✓ NOTE: Common scenario is increased query complexity *NOT* more users (then Scaling out would be better)

Scaling Out

Scaling Up	Scaling Out
Increasing the size of virtual warehouses	Using addition warehouses/ Multi-Cluster warehouses
More complex query	More concurrent users/queries

Scaling Out

- ✓ **Handling performance related to large numbers of concurrent users**
- ✓ **Automation the process if you have fluctuating number of users**

Caching

- ✓ Automatical process to speed up the queries
- ✓ If query is executed twice, results are cached and can be re-used
- ✓ Results are cached for 24 hours or until underlaying data has changed

What can we do?

- ✓ **Ensure that similar queries go on the same warehouse**
- ✓ **Example: Team of Data Scientists run similar queries, so they should all use the same warehouse**

Clustering in Snowflake

- ✓ Snowflake automatically maintains these cluster keys
- ✓ In general Snowflake produces well-clustered tables
- ✓ Cluster keys are not always ideal and can change over time
- ✓ Manually customize these cluster keys

What is a cluster key?

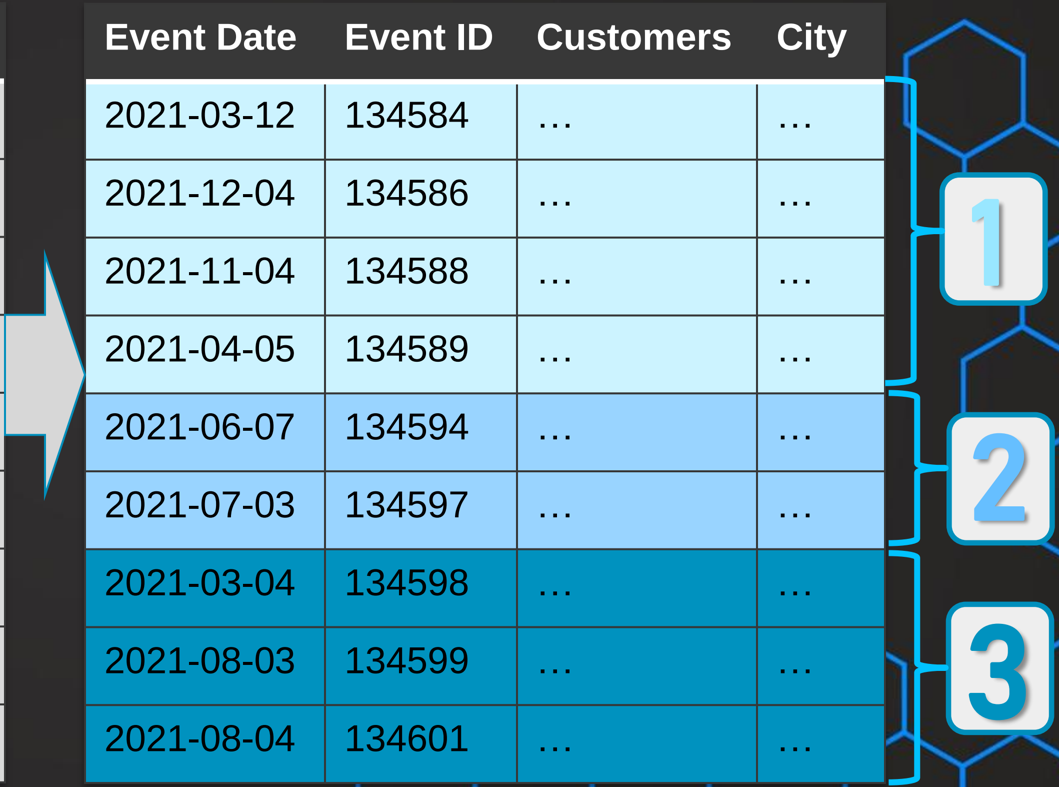
- ✓ **Subset of rows to locate the data in micro-partions**
- ✓ **For large tables this improves the scan efficiency in our queries**

What is a cluster key?

Event Date	Event ID	Customers	City
2021-03-12	134584	…	…
2021-12-04	134586	…	…
2021-11-04	134588	…	…
2021-04-05	134589	…	…
2021-06-07	134594	…	…
2021-07-03	134597	…	…
2021-03-04	134598	…	…
2021-08-03	134599	…	…
2021-08-04	134601	…	…

What is a cluster key?

Event Date	Event ID	Customers	City
2021-03-12	134584	…	…
2021-12-04	134586	…	…
2021-11-04	134588	…	…
2021-04-05	134589	…	…
2021-06-07	134594	…	…
2021-07-03	134597	…	…
2021-03-04	134598	…	…
2021-08-03	134599	…	…
2021-08-04	134601	…	…



Event Date	Event ID	Customers	City
2021-03-12	134584	…	…
2021-12-04	134586	…	…
2021-11-04	134588	…	…
2021-04-05	134589	…	…
2021-06-07	134594	…	…
2021-07-03	134597	…	…
2021-03-04	134598	…	…
2021-08-03	134599	…	…
2021-08-04	134601	…	…

What is a cluster key?

Event Date	Event ID	Customers	City
2021-03-12	134584	…	…
2021-12-04	134586	…	…
2021-11-04	134588	…	…
2021-04-05	134589	…	…
2021-06-07	134594	…	…
2021-07-03	134597	…	…
2021-03-04	134598	…	…
2021-08-03	134599	…	…
2021-08-04	134601	…	…

```
SELECT COUNT(*)
WHERE Event_Date > '2021-07-01'
AND Event_Date < '2021-08-01 '
```

What is a cluster key?

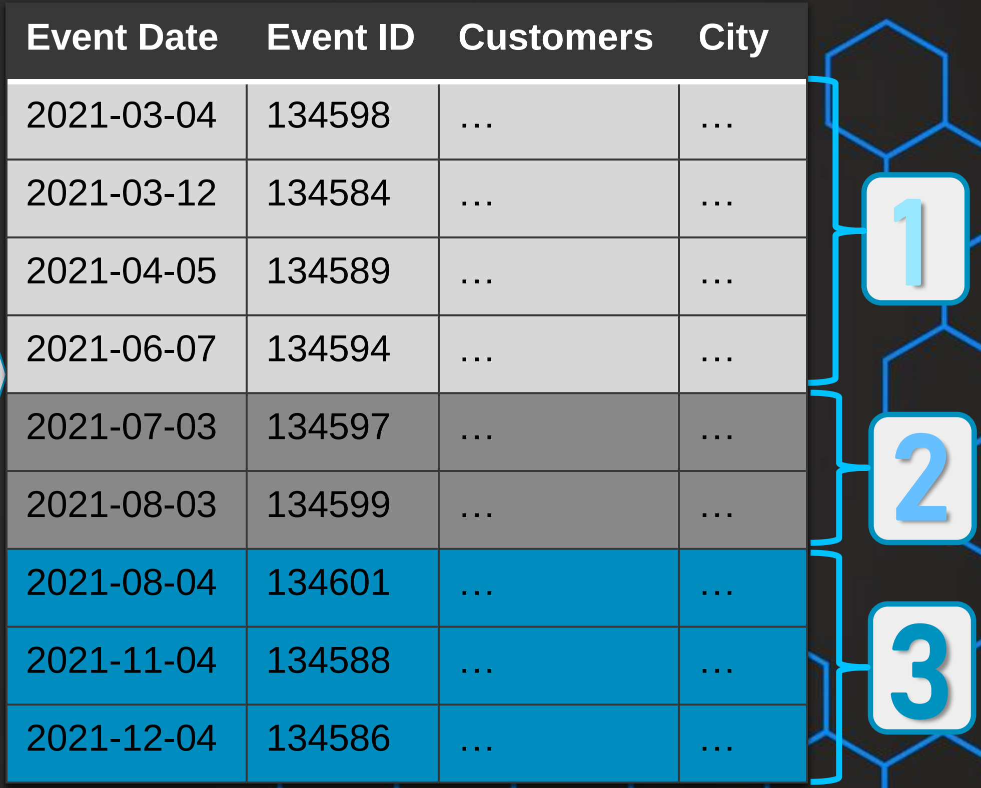
Event Date	Event ID	Customers	City
2021-03-12	134584	…	…
2021-12-04	134586	…	…
2021-11-04	134588	…	…
2021-04-05	134589	…	…
2021-06-07	134594	…	…
2021-07-03	134597	…	…
2021-03-04	134598	…	…
2021-08-03	134599	…	…
2021-08-04	134601	…	…

```
SELECT COUNT(*)
WHERE Event_Date > '2021-07-01'
AND Event_Date < '2021-08-01'
```

All partitions need to be scanned!

What is a cluster key?

Event Date	Event ID	Customers	City
2021-03-12	134584	…	…
2021-12-04	134586	…	…
2021-11-04	134588	…	…
2021-04-05	134589	…	…
2021-06-07	134594	…	…
2021-07-03	134597	…	…
2021-03-04	134598	…	…
2021-08-03	134599	…	…
2021-08-04	134601	…	…



Event Date	Event ID	Customers	City
2021-03-04	134598	…	…
2021-03-12	134584	…	…
2021-04-05	134589	…	…
2021-06-07	134594	…	…
2021-07-03	134597	…	…
2021-08-03	134599	…	…
2021-08-04	134601	…	…
2021-11-04	134588	…	…
2021-12-04	134586	…	…

When to cluster?

- ✓ **Clustering is not for all tables**

- ✓ **Mainly very large tables of <u>multiple terabytes</u> can benefit**

How to cluster?

- ✓ Columns that are used most frequently in WHERE-clauses (often date columns for event tables)
- ✓ If you typically use filters on two columns then the table can also benefit from two cluster keys
- ✓ Column that is frequently used in Joins
- ✓ Large enough number of distinct values to enable effective grouping
  Small enough number of distinct values to allow effective grouping

Clustering in Snowflake

```
CREATE TABLE <name> ... CLUSTER BY ( <column1> [ , <column2> ... ] )
```

```
CREATE TABLE <name> ... CLUSTER BY ( <expression> )
```

```
ALTER TABLE <name> CLUSTER BY ( <expr1> [ , <expr2> ... ] )
```

```
ALTER TABLE <name> DROP CLUSTERING KEY
```

Clustering in Snowflake

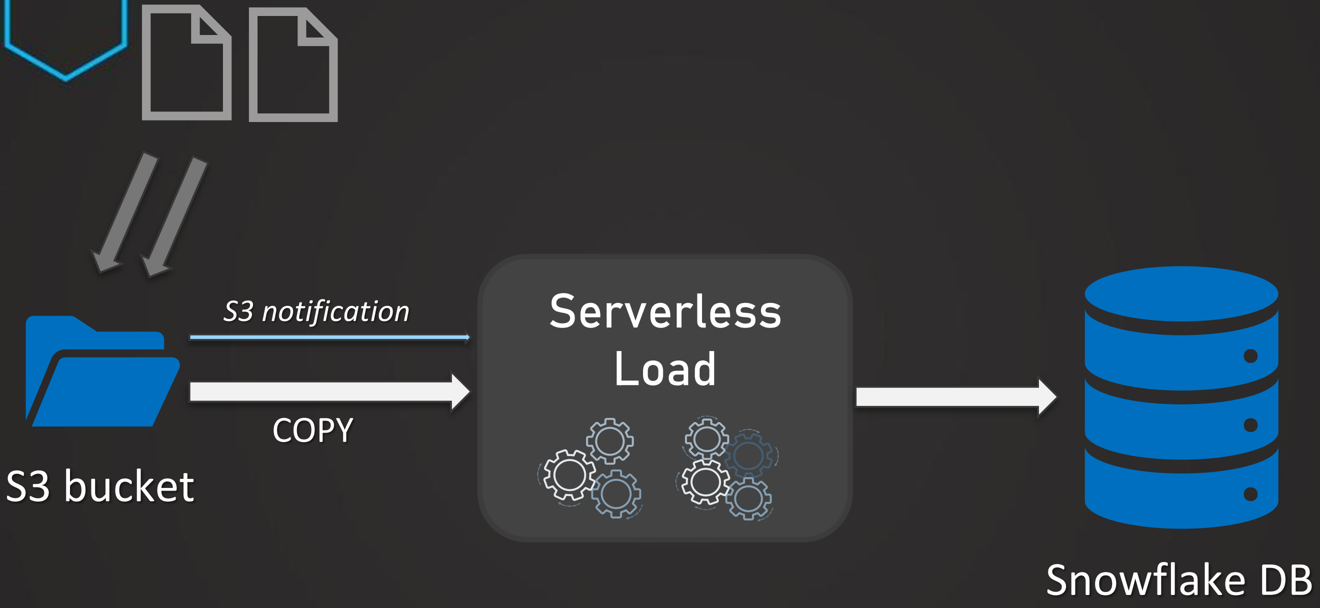
- ✓ Columns that are used most frequently in WHERE-clauses (often date columns for event tables)
- ✓ If you typically use filters on two columns then the table can also benefit from two cluster keys
- ✓ Column that is frequently used in Joins
- ✓ Large enough number of distinct values to enable effective grouping
  Small enough number of distinct values to allow effective grouping

Snowpipe

What is Snowpipe?

- ✓ **Enables loading once a file appears in a bucket**
- ✓ **If data needs to be available immediately for analysis**
- ✓ **Snowpipe uses serverless features instead of warehouses**

Snowpipe


S3 notification
COPY
S3 bucket
Serverless Load
Snowflake DB

Setting up Snowpipe

- **Create Stage** — ✓ To have the connection
- **Test COPY COMMAND** — ✓ To make sure it works
- **Create Pipe** — ✓ Create pipe as object with COPY COMMAND
- **S3 Notification** — ✓ To trigger snowpipe

Fail Safe and Time Travel

Time Travel

Standard	Enterprise	Business Critical	Virtual Private
✓ Time travel up to 1 day	✓ Time travel up to 90 days	✓ Time travel up to 90 days	✓ Time travel up to 90 days

RETENTION PERIODE DEFAULT = 1

Fail Safe

- ✓ Protection of historical data in case of disaster
- ✓ Non-configurable 7-day period for permanent tables
- ✓ Period starts immediately after Time Travel period ends
- ✓ No user interaction & recoverable only by Snowflake
- ✓ Contributes to storage cost

Fail Safe

✓ Access and query data etc.

Current Data Storage

Continuous Data Protection Lifecycle

- ✓ No user operations/queries
- ✓ Recovery beyond Time Travel
- ✓ Restoring only by snowflake support

- ✓ SELECT … AT | BEFORE
- UNDROP

- ✓ Access and query data etc.

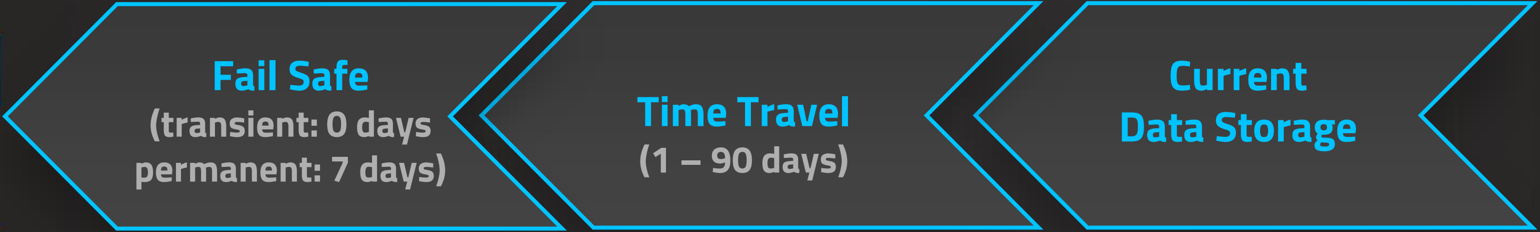


Table Types

Table types

Permanent data	Only for data that does not need to be protected	Non-permanent data
Until dropped	Until dropped	Only in session
Permanent	**Transient**	**Temporary**
`CREATE TABLE`	`CREATE TRANSIENT TABLE`	`CREATE TEMPORARY TABLE`
✓ Time Travel Retention Period 0 – 90 days	✓ Time Travel Retention Period 0 – 1 day	✓ Time Travel Retention Period 0 – 1 day
✓ Fail Safe	× Fail Safe	× Fail Safe

Table types

Permanent data

Until dropped

Permanent

CREATE TABLE

✓ Time Travel Retention Period

0 – 90 days

✓ Fail Safe

Only for data that does not need to be protected

Until dropped

✓ Time Travel Retention Period

0 – 1 day

× Fail Safe

Non-permanent data

Only in session

Temporary

ATE TEMPORARY TABLE

✓ Time Travel Retention Period

0 – 1 day

× Fail Safe

Managing Storage Cost

Table types notes

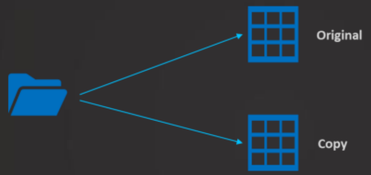
- ✓ **Types are also available for other database objects (database, schema etc.)**
- ✓ **For temporary table no naming conflicts with permanent/transient tables!**

  Other tables will be effectively hidden!

Zero Copy Cloning

Zero-Copy Cloning

- ✓ Create copies of a database, a schema or a table



- ✓ Cloned object is independent from original table
- ✓ Easy to copy all meta data & improved storage management
- ✓ Creating backups for development purposes
- ✓ Works with time travel also

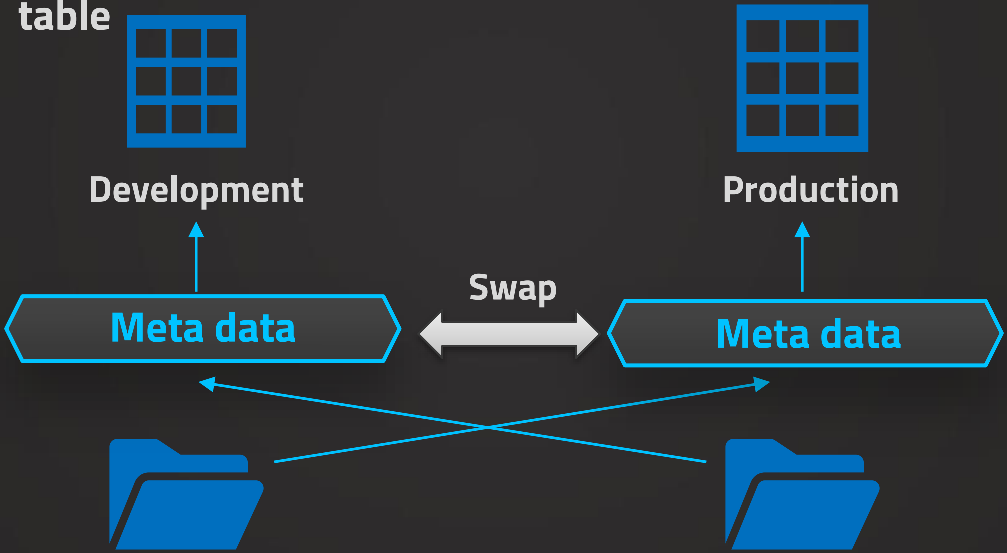
Zero-Copy Cloning

```
CREATE TABLE <table_name> ...
  CLONE  <source_table_name>
  BEFORE ( TIMESTAMP => <timestamp> )
```

Swapping

Swapping Tables

- ✓ **Use-case: Development table into production table**



Swapping

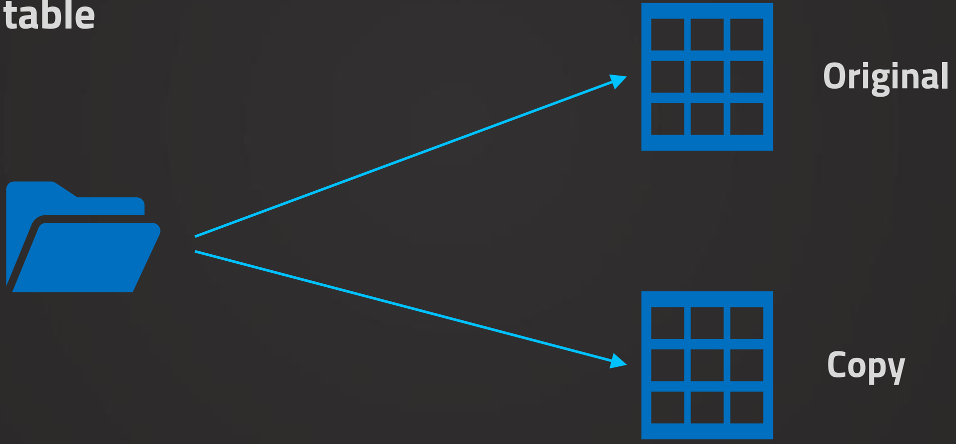
```
ALTER TABLE <table_name> ...
  SWAP WITH  <target_table_name>
```

Swapping

```
ALTER SCHEMA <schema_name> ...
  SWAP WITH  <target_schema_name>
```

Swaping Tables

- ✓ Create copies of a database, a schema or a table



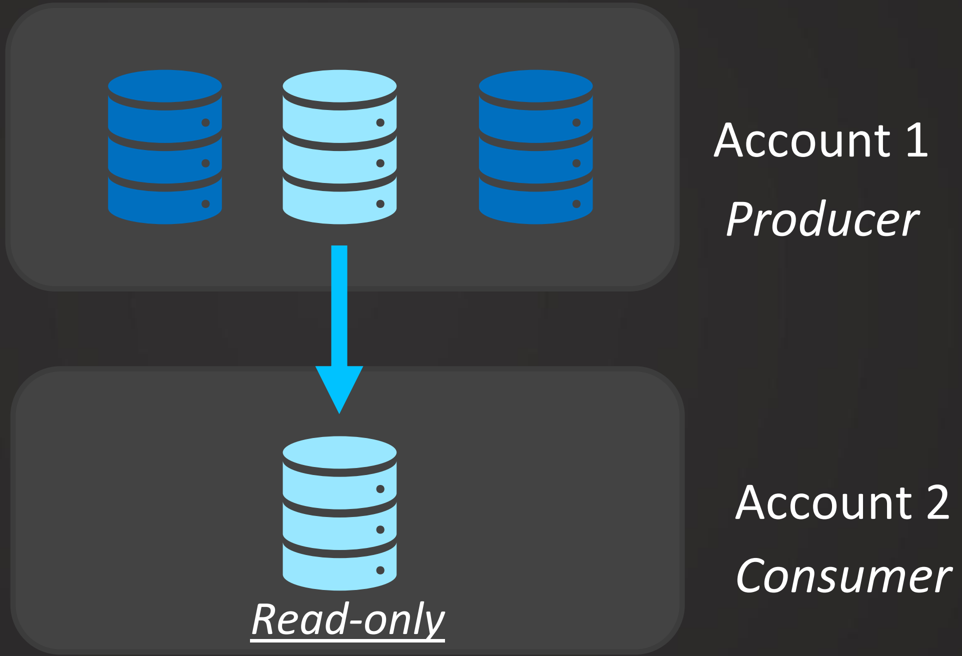
Meta data operation

Data Sharing

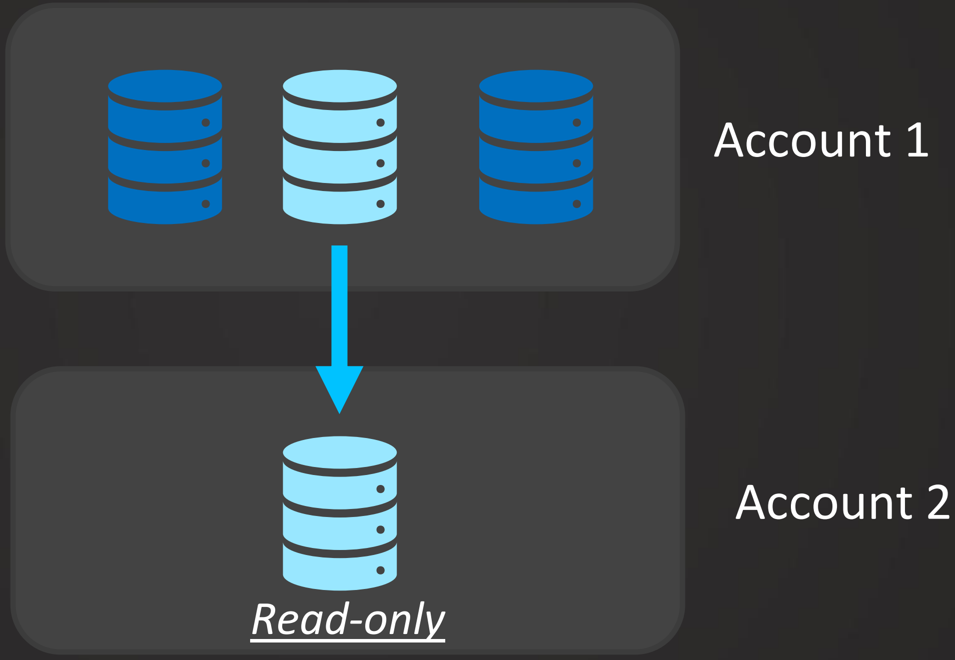
Data Sharing

- ✓ Usually this can be also a rather complicated process
- ✓ Data sharing without actual copy of the data & uptodate
- ✓ Shared data can be consumed by the own compute resources
- ✓ Non-Snowflake-Users can also access through a reader account

Data Sharing

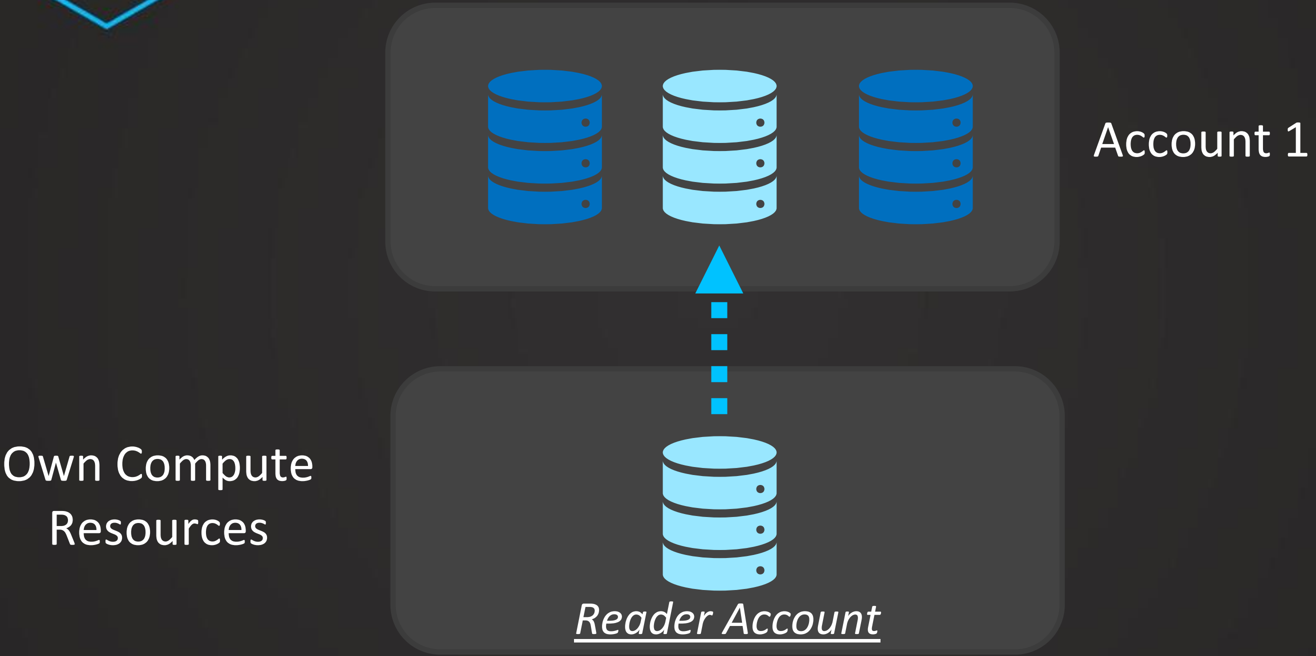


Data Sharing

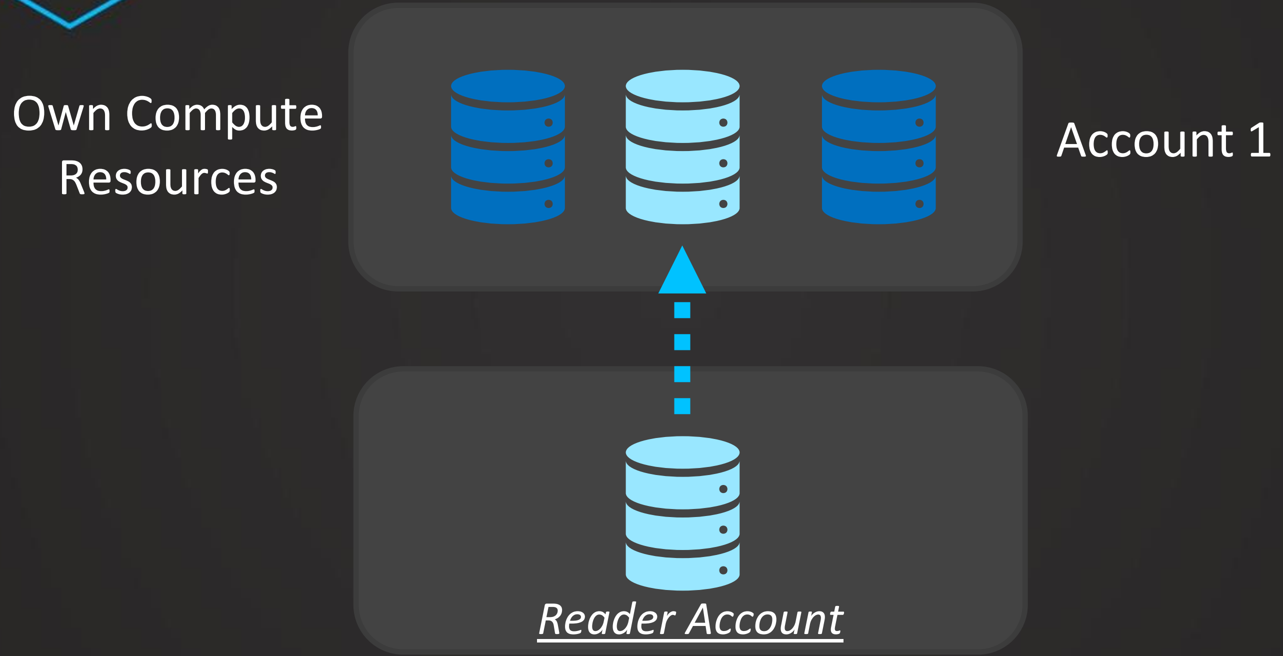


Account 2
Compute Resources

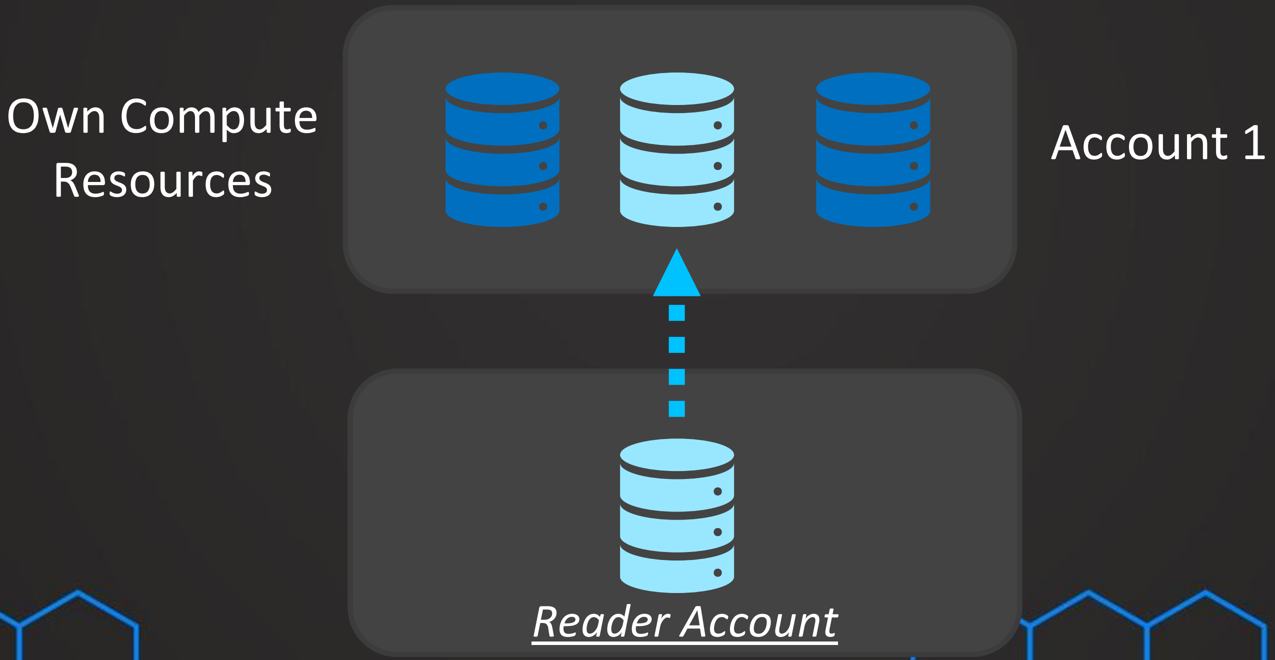
Data Sharing



Data Sharing



Data Sharing with Non Snowflake Users



Sharing with Non Snowflake users

New Reader Account
- ✓ Indepentant instance with own url & own compute resources

Share data
- ✓ Share database & table

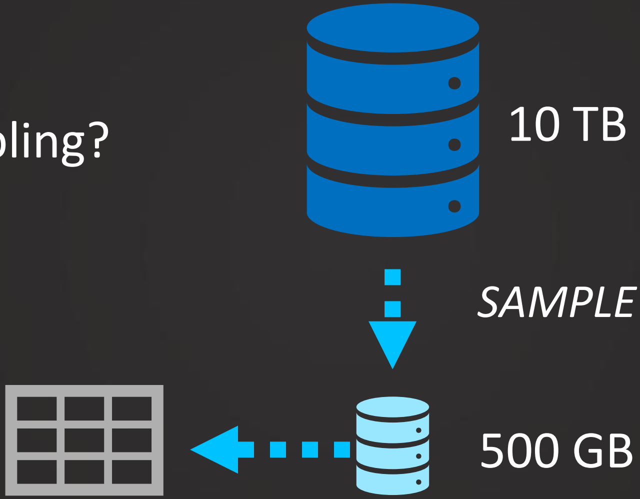
Create Users
- ✓ As administrator create user & roles

Create database
- ✓ In reader account create database from share

Data Sampling

Data Sampling

Why Sampling?



Data Sampling

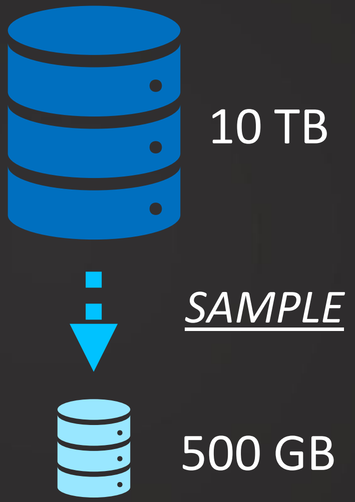
Why Sampling?

- Use-cases: Query development, data analysis etc.

- Faster & more cost efficient (less compute resources)

Data Sampling

Why Sampling?



Data Sampling Methods

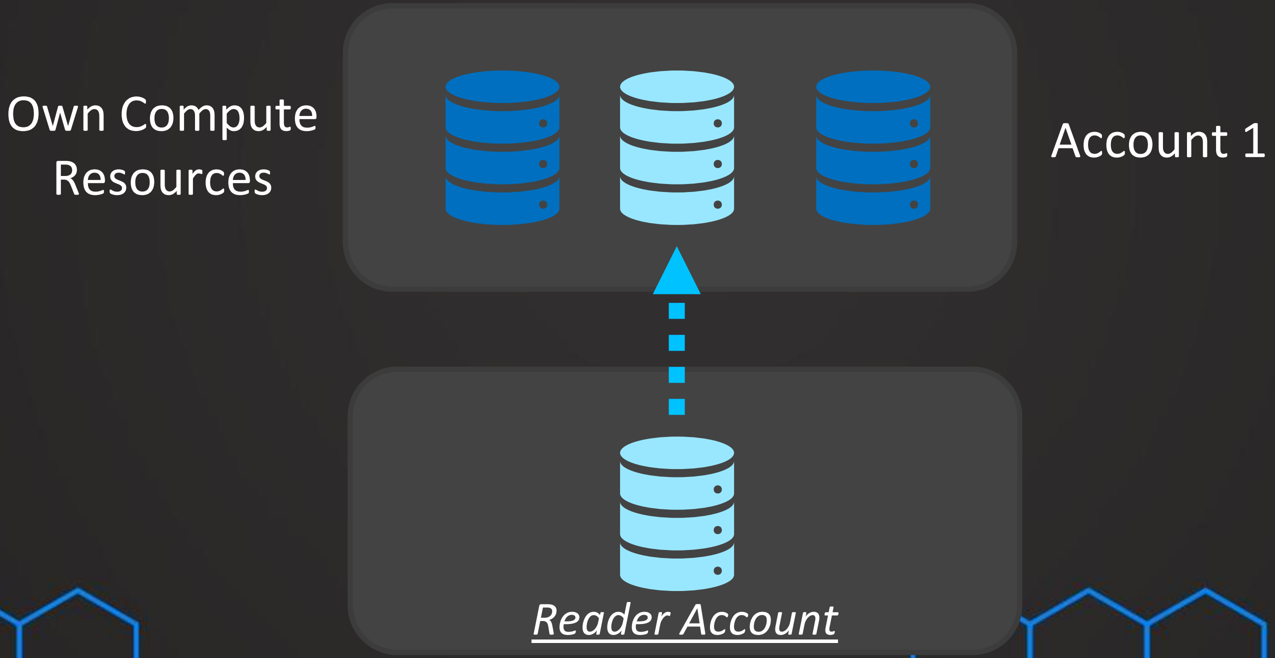
ROW or BERNOULLI method

BLOCK or SYSTEM method

Data Sampling Methods

ROW or BERNOULLI method	BLOCK or SYSTEM method
Every row is chosen with percentage *p*	Every block is chosen with percentage *p*
More "randomness"	More effective processing
Smaller tables	Larger tables

Data Sampling



Tasks & Streams

Scheduling Tasks

- ✓ Tasks can be used to schedule SQL statements
- ✓ Standalone tasks and trees of tasks

Understand tasks

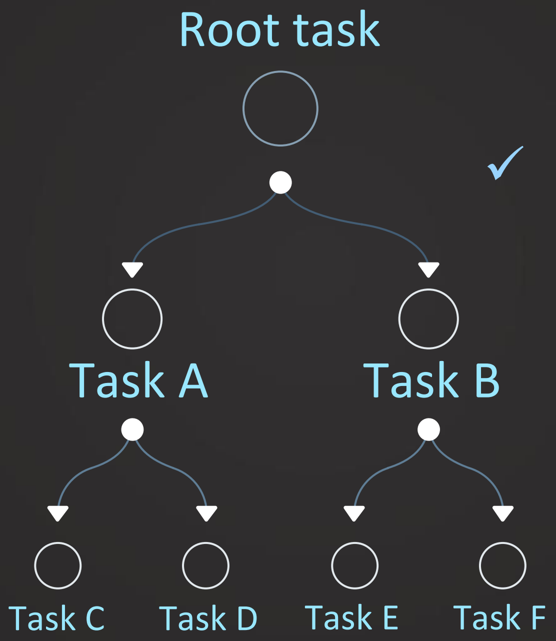
Create tasks

Schedule tasks

Tree of tasks

Check task history

Tree of Tasks



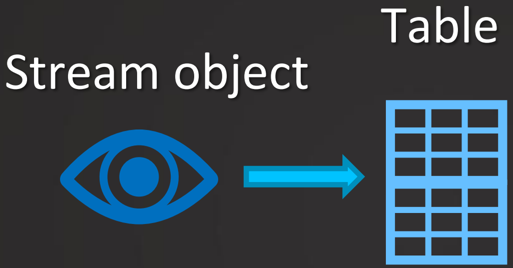
✓ Every task has one parent

Tree of Tasks

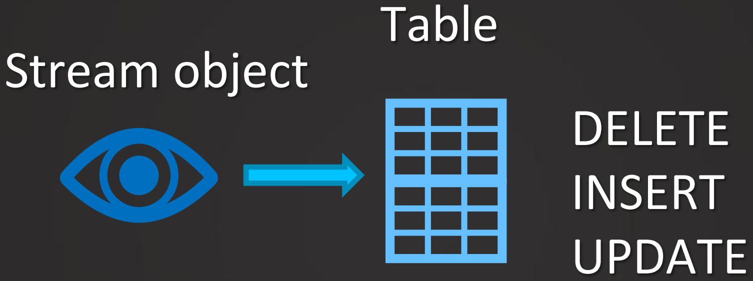
```
CREATE TASK ...
  AFTER  <parent task>
  AS …
```

```
ALTER TASK ...
  ADD AFTER  <parent task>
```

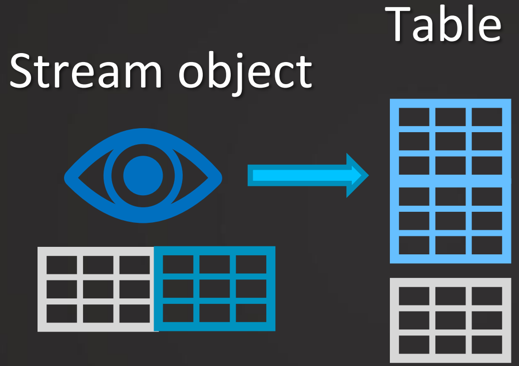
Streams



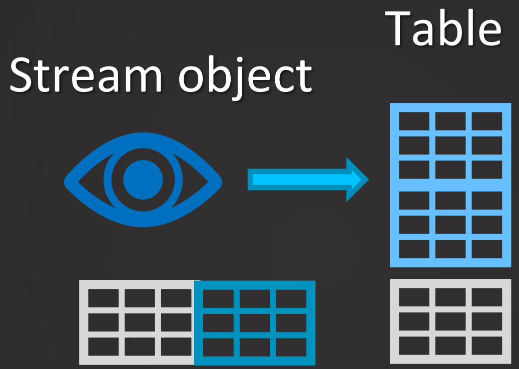
Streams



Streams



Streams



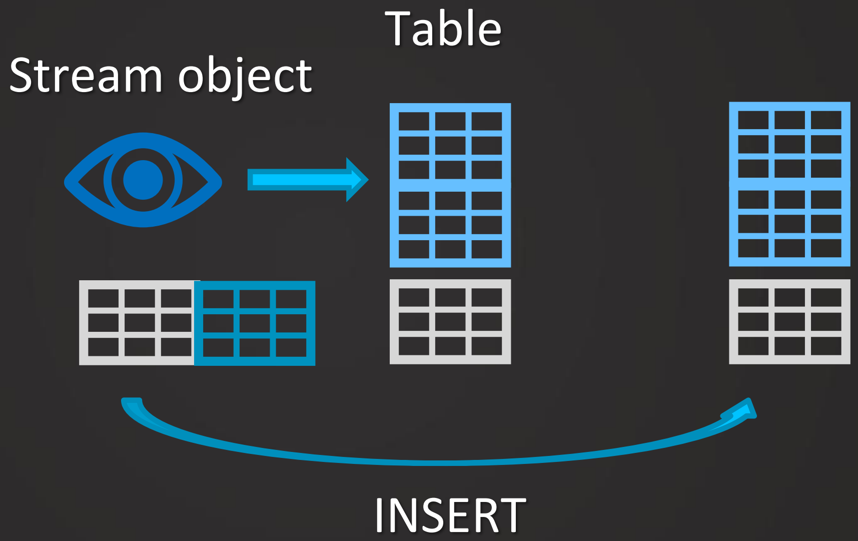
METADATA$ACTION
METADATA$UPDATE
METADATA$ROW_ID

Streams

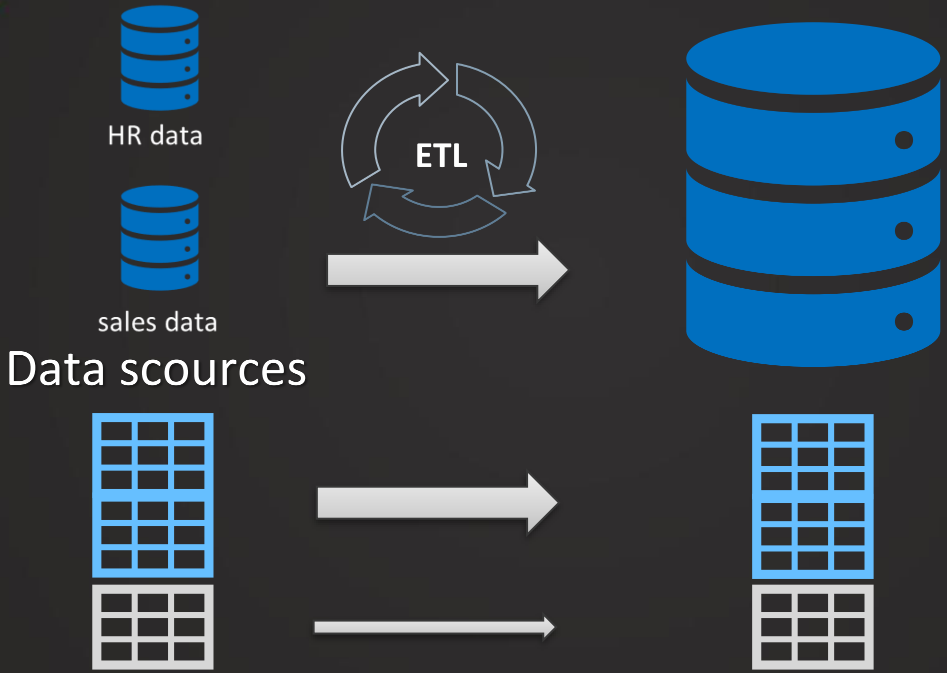
```
CREATE STREAM <stream name>
  ON TABLE  <table name>
```

```
SELECT * FROM <stream name>
```

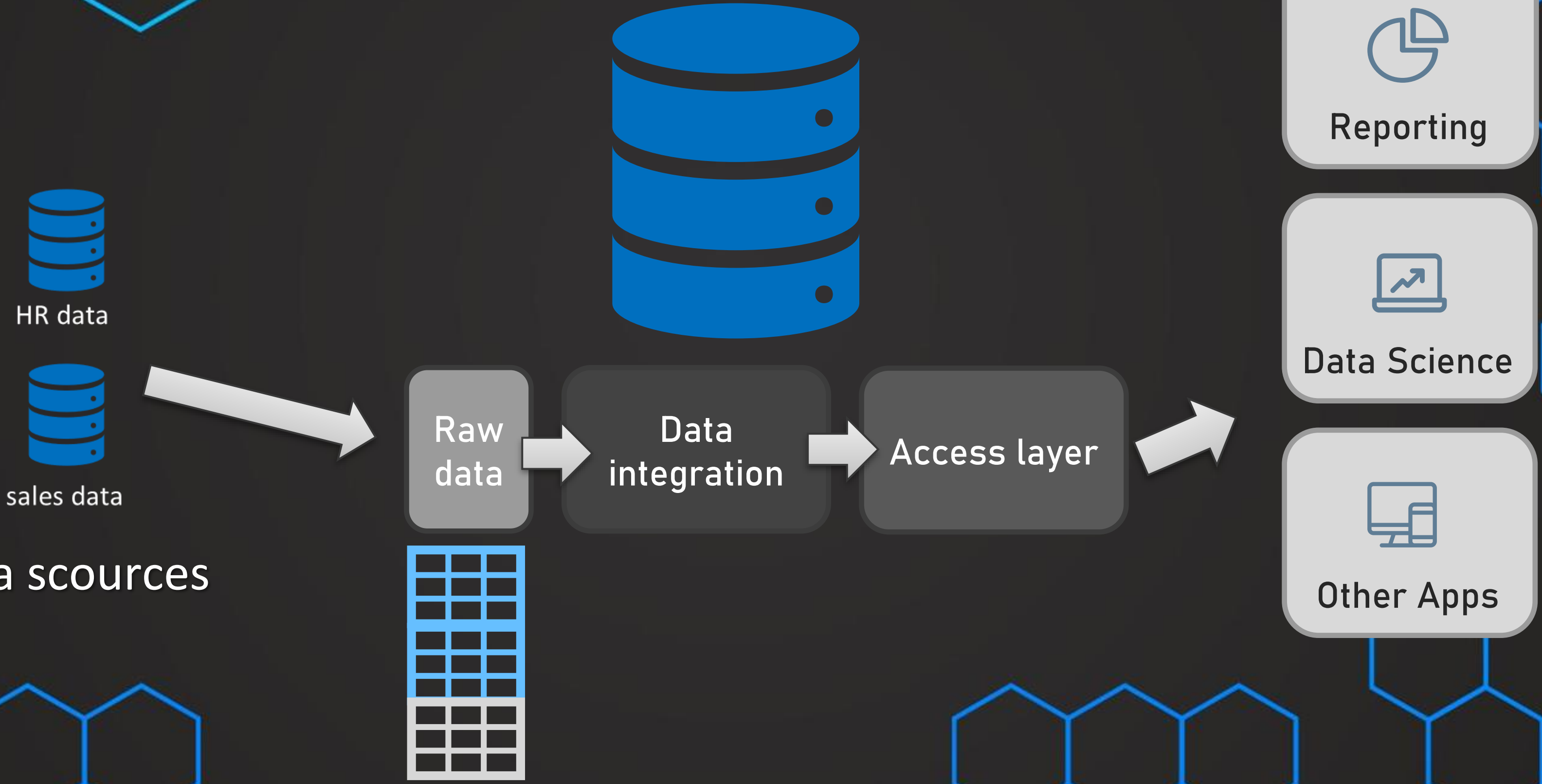
Streams



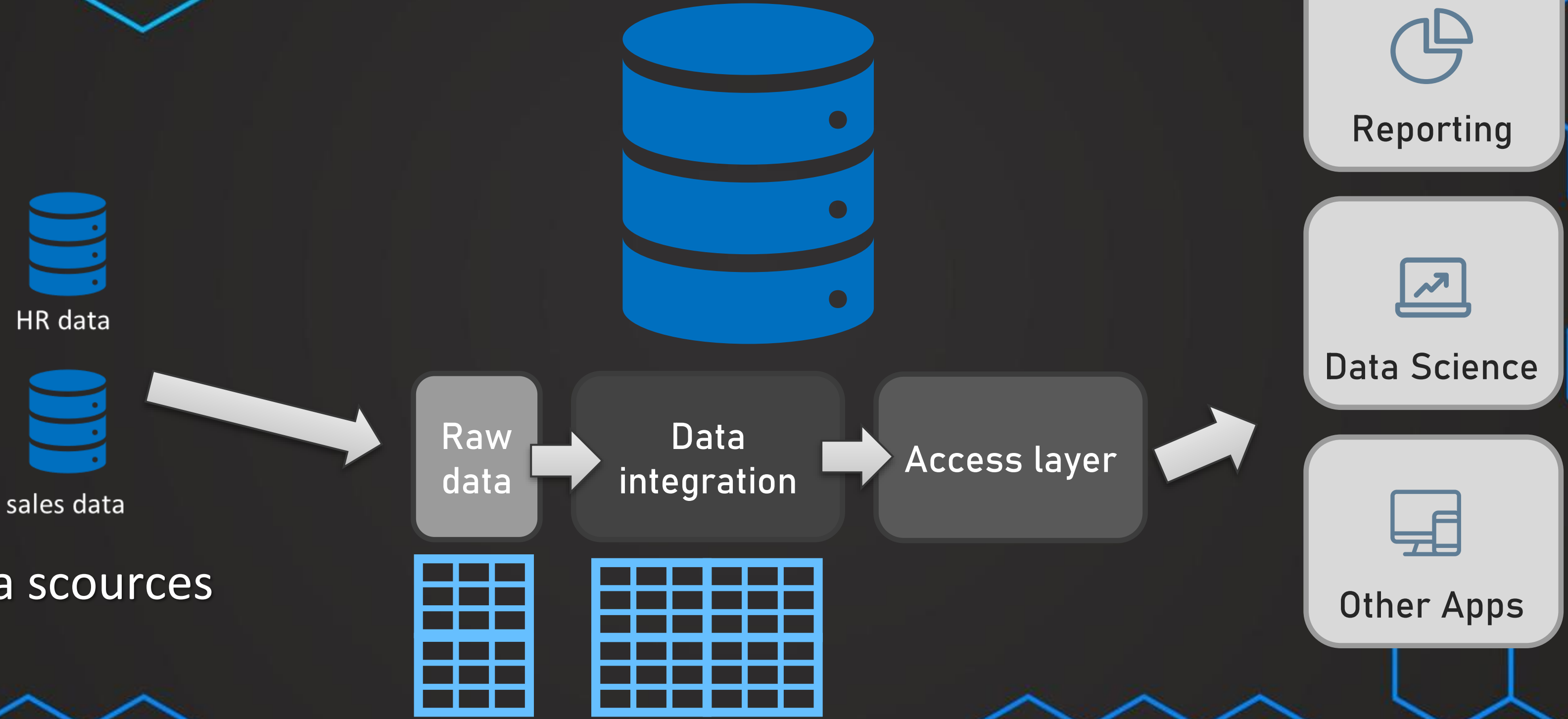
Streams



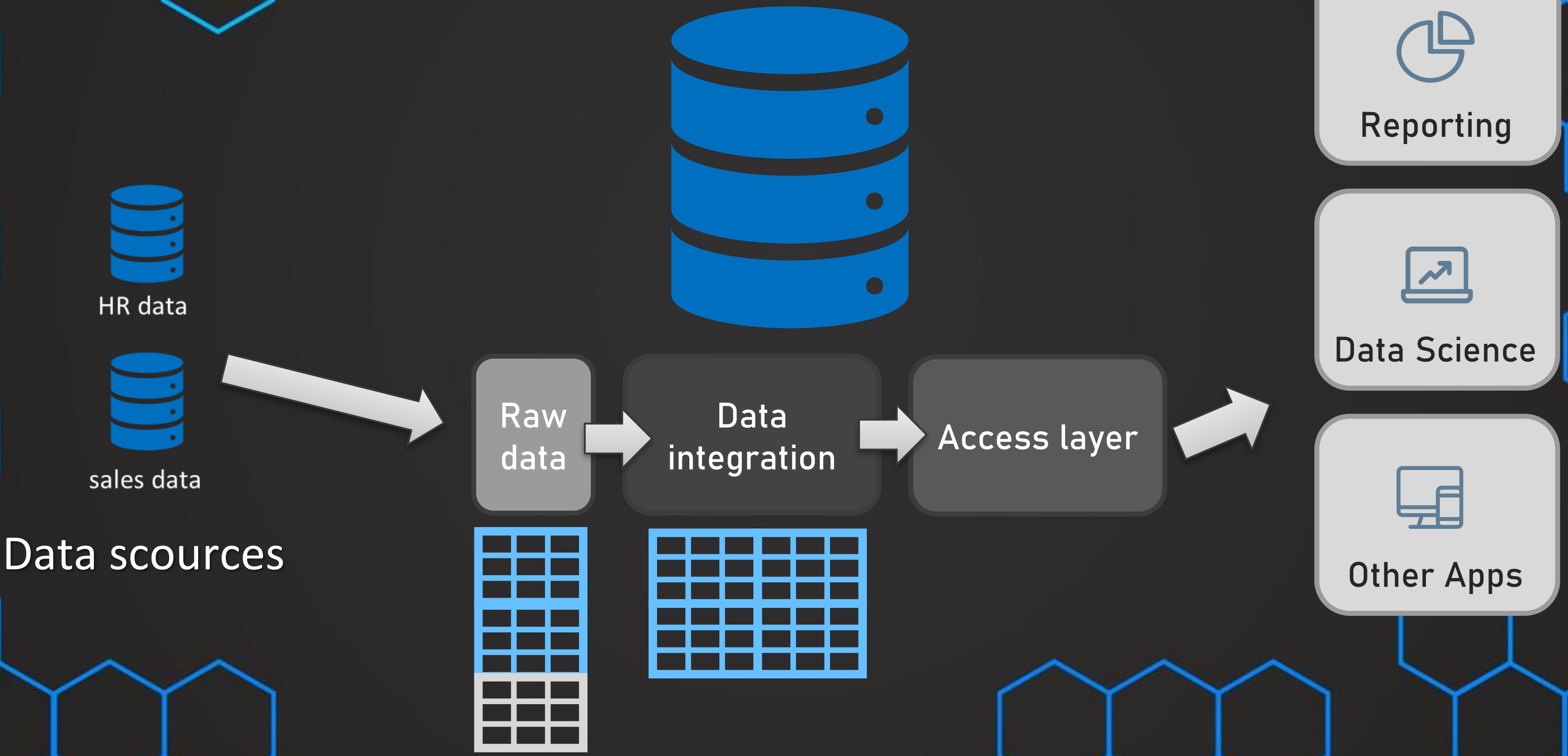
Streams


HR data
sales data
Data scources
Raw data
Data integration
Access layer
Reporting
Data Science
Other Apps

Streams

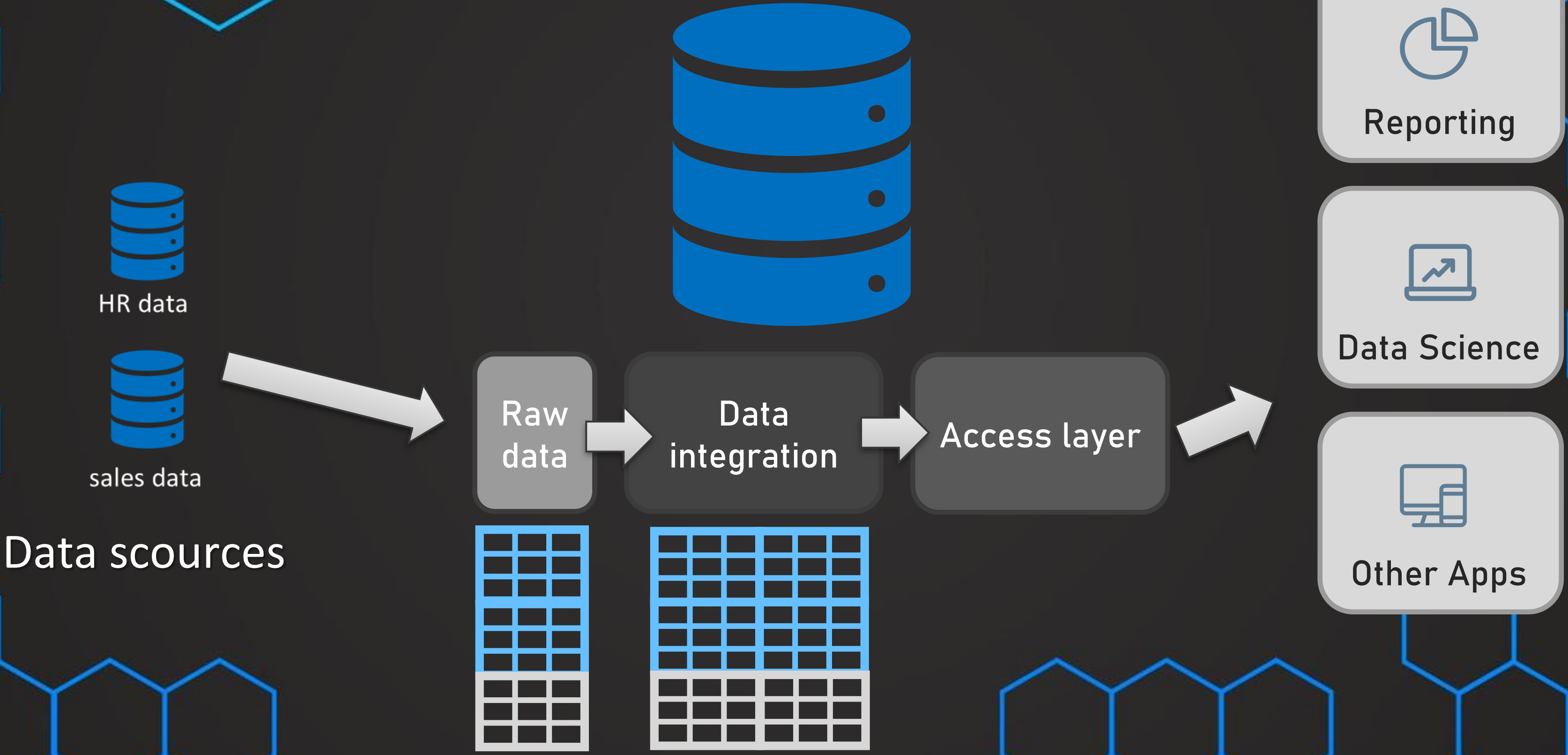

HR data
sales data
Data scources
Raw data
Data integration
Access layer
Reporting
Data Science
Other Apps

Streams



HR data
sales data
Data scources
Raw data
Data integration
Access layer
Reporting
Data Science
Other Apps

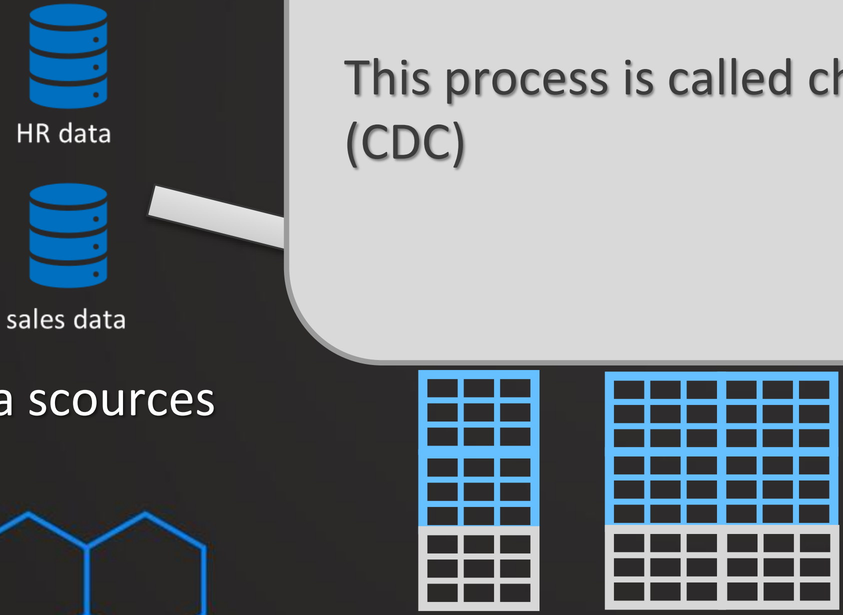
Streams

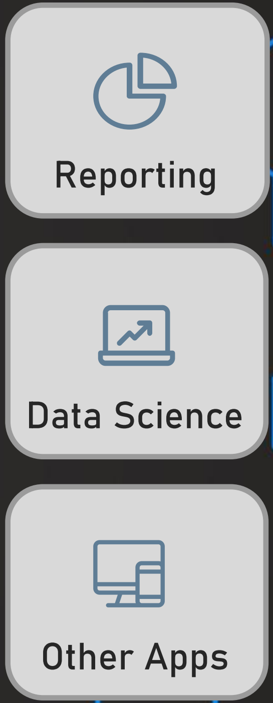

HR data
sales data
Data scources
Raw data
Data integration
Access layer
Reporting
Data Science
Other Apps

Streams

Object that records (DML-)changes made to a table

This process is called change data capture (CDC)





Types of streams

STANDARD	APPEND-ONLY
✓ INSERT	✓ INSERT
✓ UPDATE	
✓ DELETE	

Syntax

```
CREATE STREAM <stream name>
  ON TABLE  <table name>
  APPEND_ONLY = TRUE
```

Materialized Views

Materialized views

- ✓ We have a view that is queried frequently and that a long time to be processed
- ✗ Bad user experience
- ✗ More compute consumption

Materialized views

- ✓ We have a view that is queried frequently and that a long time to be processed
- ✓ We can create a materialized view to solve that problem

What is a materialized view?

- ✓ Use any SELECT-statement to create this MV
- ✓ Results will be stored in a seperate table and this will be updated automatically based on the base table

When to use MV?

- ✓ Benefits
- ✓ Maintenance costs

When to use MV?

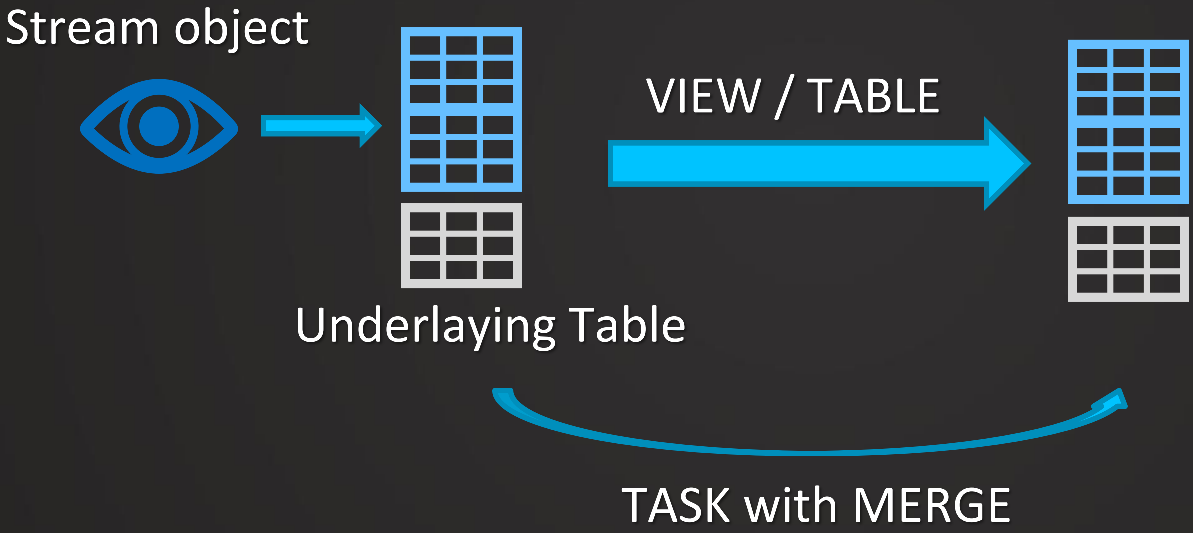
- ✓ View would take a long time to be processed and is used frequently
- ✓ Underlaying data is change not frequently and on a rather irregular basis

When to use MV?

If the data is updated on a very regular basis…

✓ Using *tasks & streams* could be a better alternative

Alternative – streams & tasks



When to use MV?

- ✓ Don't use materialized view if data changes are very frequent
- ✓ Keep maintenance cost in mind
- ✓ Considder leveraging tasks (& streams) instead

Limitations

Only available for Enterprise edition

Limitations

- Joins (including self-joins) are not supported
- Limited amount of aggregation functions

Limitations

- Joins (including self-joins) are
- Limited amount of aggregatio

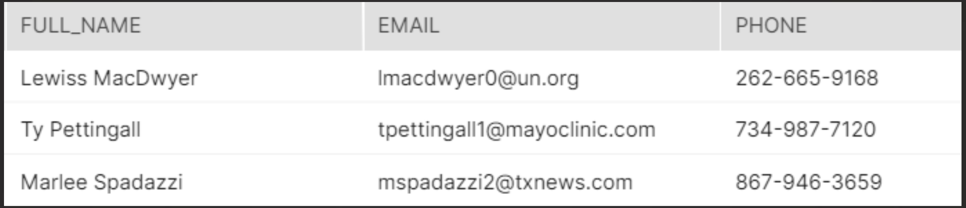
APPROX_COUNT_DISTINCT (HLL).
AVG (except when used in PIVOT).
BITAND_AGG.
BITOR_AGG.
BITXOR_AGG.
COUNT.
MIN.
MAX.
STDDEV.
STDDEV_POP.
STDDEV_SAMP.
SUM.
VARIANCE (VARIANCE_SAMP, VAR_SAMP).
VARIANCE_POP (VAR_POP).

Limitations

- Joins (including self-joins) are not supported
- Limited amount of aggregation functions
- UDFs
- HAVING clauses.
- ORDER BY clause.
- LIMIT clause

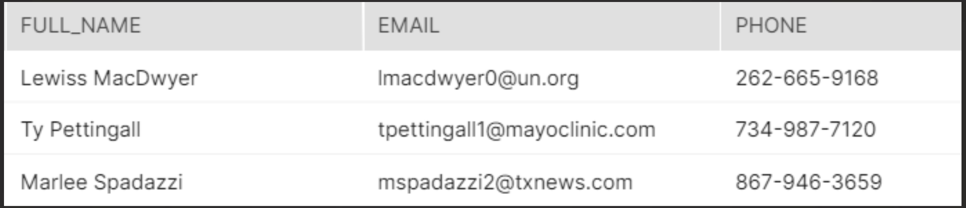
Data Masking

Data Masking

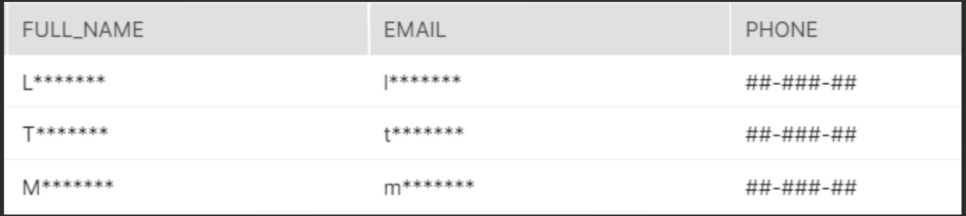


FULL_NAME	EMAIL	PHONE
Lewiss MacDwyer	lmacdwyer0@un.org	262-665-9168
Ty Pettingall	tpettingall1@mayoclinic.com	734-987-7120
Marlee Spadazzi	mspadazzi2@txnews.com	867-946-3659

Data Masking



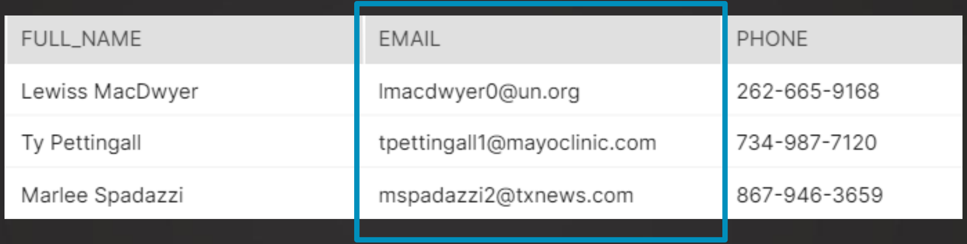
FULL_NAME	EMAIL	PHONE
Lewiss MacDwyer	lmacdwyer0@un.org	262-665-9168
Ty Pettingall	tpettingall1@mayoclinic.com	734-987-7120
Marlee Spadazzi	mspadazzi2@txnews.com	867-946-3659



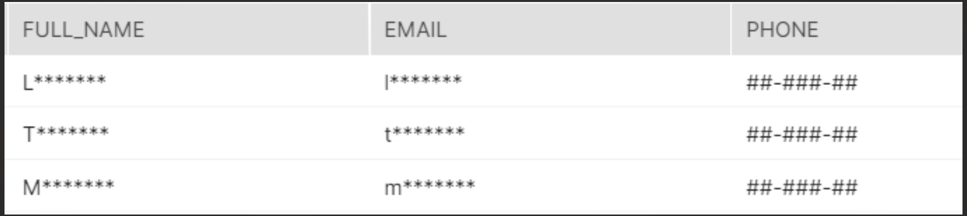
FULL_NAME	EMAIL	PHONE
L*******	l*******	##-###-##
T*******	t*******	##-###-##
M*******	m*******	##-###-##

Data Masking

Column-level Security



FULL_NAME	EMAIL	PHONE
Lewiss MacDwyer	lmacdwyer0@un.org	262-665-9168
Ty Pettingall	tpettingall1@mayoclinic.com	734-987-7120
Marlee Spadazzi	mspadazzi2@txnews.com	867-946-3659



FULL_NAME	EMAIL	PHONE
L*******	l*******	##-###-##
T*******	t*******	##-###-##
M*******	m*******	##-###-##

Access Control

Access Control

- ✓ Who can access and perform operations on objects in Snowflake
- ✓ Two aspects of access control combined

Access Control

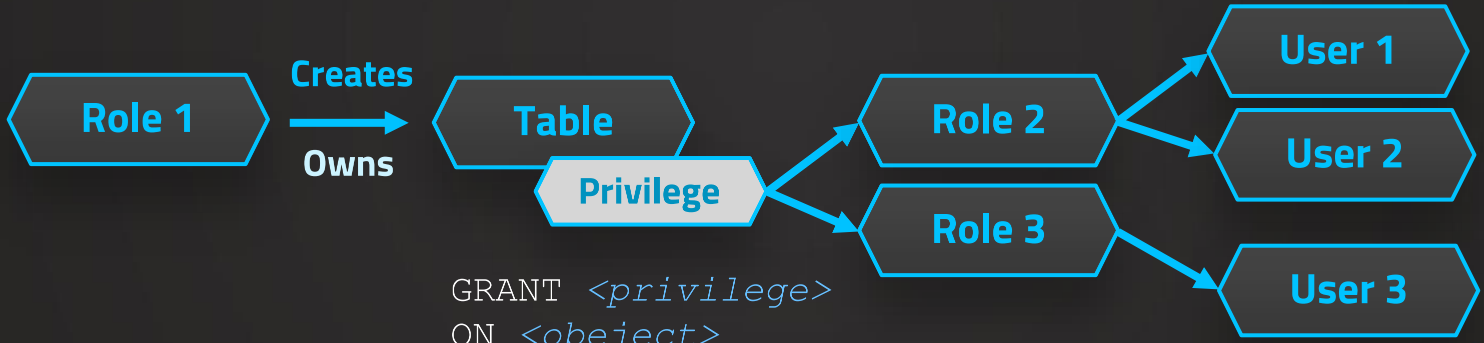
Discretionary Access Control (DAC)

- ✓ Each object has an owner who can grant access to that object

Role-based Access Control (RBAC)

- ✓ Access privileges are assigned to roles, which are in turn assigned to users

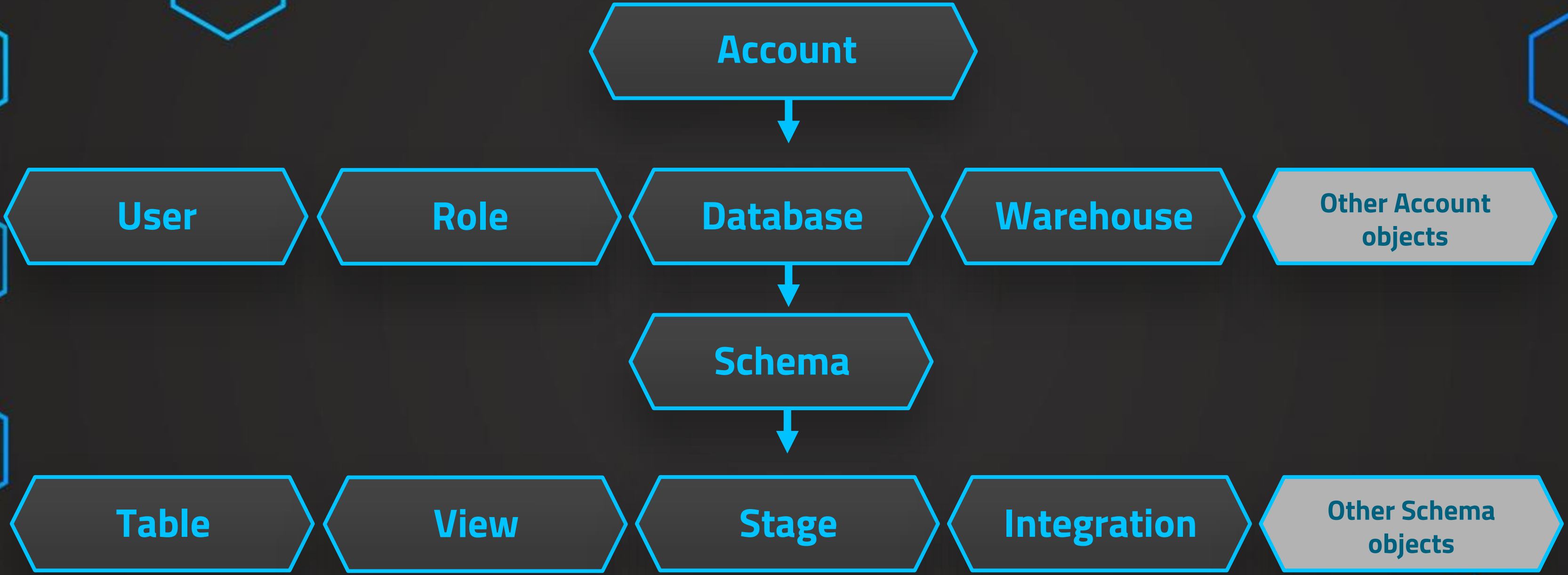
Access Control



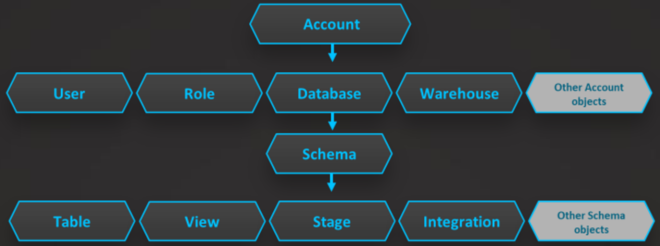
```
GRANT <role>
TO <user>
```

```
GRANT <privilege>
ON <obeject>
TO <role>
```

Securable objects



Access Control



- ✓ Every object owned by a single role (multiple users)
- ✓ Owner (role) has all privileges per default

Key concepts

USER	✓ People or systems
ROLE	✓ Entity to which privileges are granted (role hierarchy)
PRIVILEGE	✓ Level of access to an object (SELECT, DROP, CREATE etc.)
SECURABLE OBJECT	✓ Objects to which privileges can be granted (Database, Table, Warehouse etc.)

Snowflake Roles

ACCOUNTADMIN

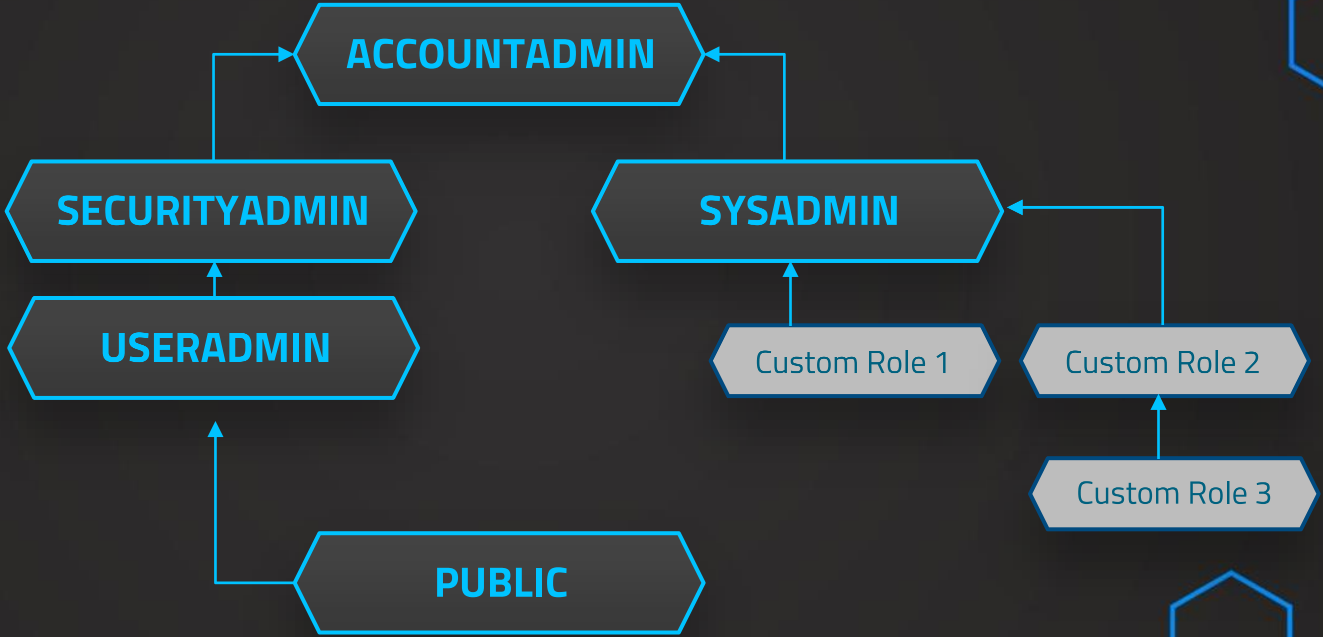
SECURITYADMIN

SYSADMIN

USERADMIN

PUBLIC

Snowflake Roles



Snowflake Roles

ACCOUNTADMIN

- ✓ SYSADMIN and SECURITYADMIN
- ✓ top-level role in the system
- ✓ should be granted only to a limited number of users

SECURITYADMIN

- ✓ USERADMIN role is granted to SECURITYADMIN
- ✓ Can manage users and roles
- ✓ Can manage any object grant globally

SYSADMIN

- ✓ Create warehouses and databases (and more objects)
- ✓ Recommended that all custom roles are assigned

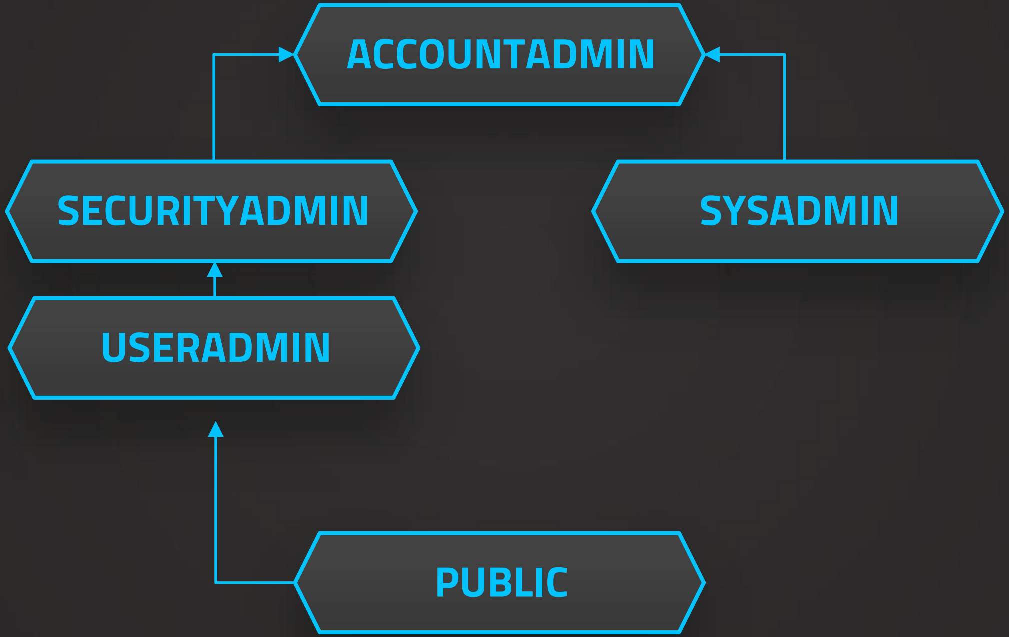
USERADMIN

- ✓ Dedicated to user and role management only
- ✓ Can create users and roles

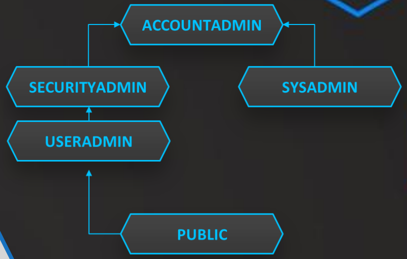
PUBLIC

- ✓ Automatically granted to every user
- ✓ Can create own objects like every other role (available to every other user/role

ACCOUNTADMIN



ACCOUNTADMIN



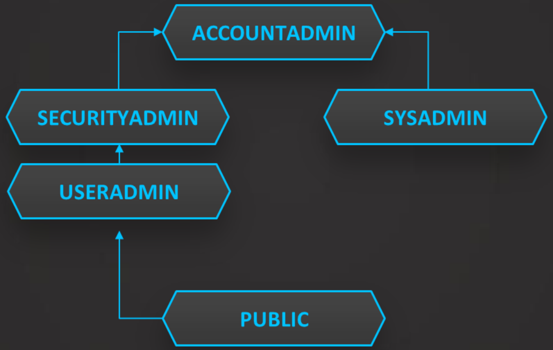
Top-Level-Role

- ✓ Manage & view all objects
- ✓ All configurations on account level
- ✓ Account operations (create reader account, billing etc.)
- ✓ First user will have this role assigned
- ✓ Initial setup & managing account level objects

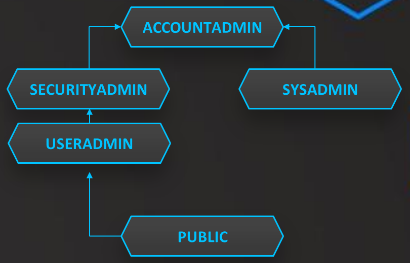
Best practises

- ✓ Very controlled assignment strongly recommended!
- ✓ Multi-factor authentification
- ✓ At least two users should be assigned to that role
- ✓ Avoid creating objects with that role unless you have to

ACCOUNTADMIN

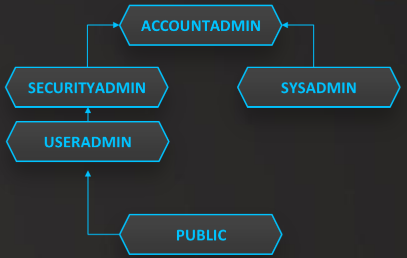


ACCOUNTADMIN



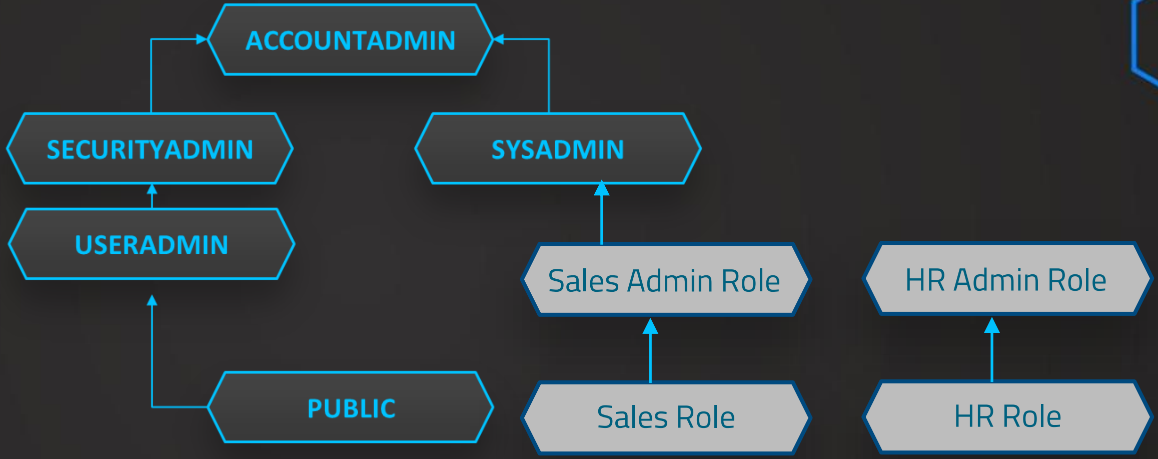
- ✓ Account admin tab
- ✓ Billing & Usage
- ✓ Reader Account
- ✓ Multi-Factor Authentification
- ✓ Create other users

ACCOUNTADMIN

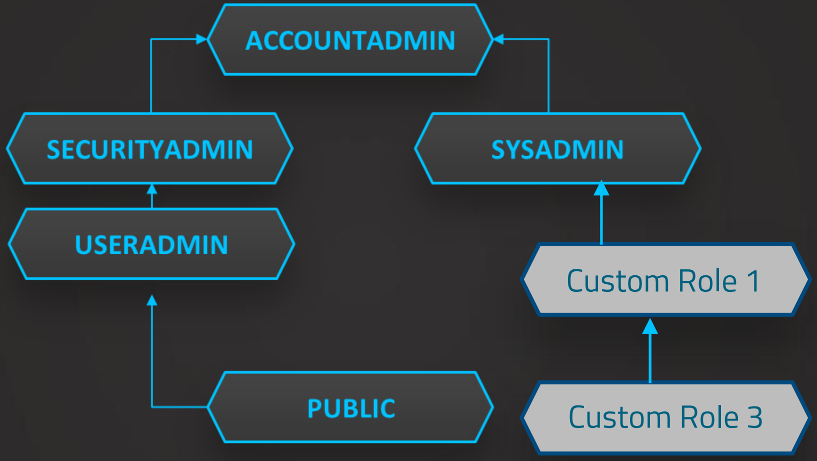


- ✓ USERADMIN role is granted to SECURITYADMIN
- ✓ Can manage users and roles
- ✓ Can manage any object grant globally

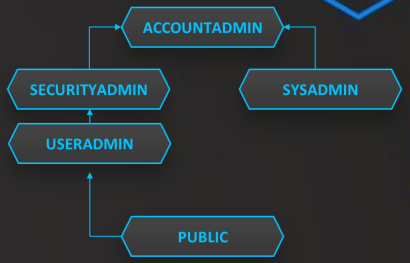
SECURITYADMIN



SECURITYADMIN



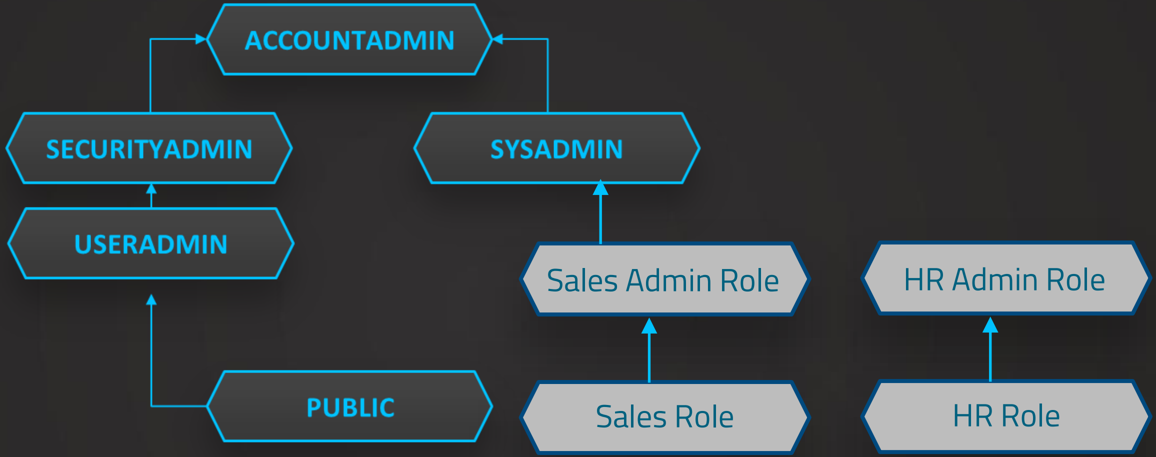
SYSADMIN



- ✓ Create & manage objects
- ✓ Create & manage warehouses, databases, tables etc.
- ✓ Custom roles should be assigned to the SYSADMIN role as the parent

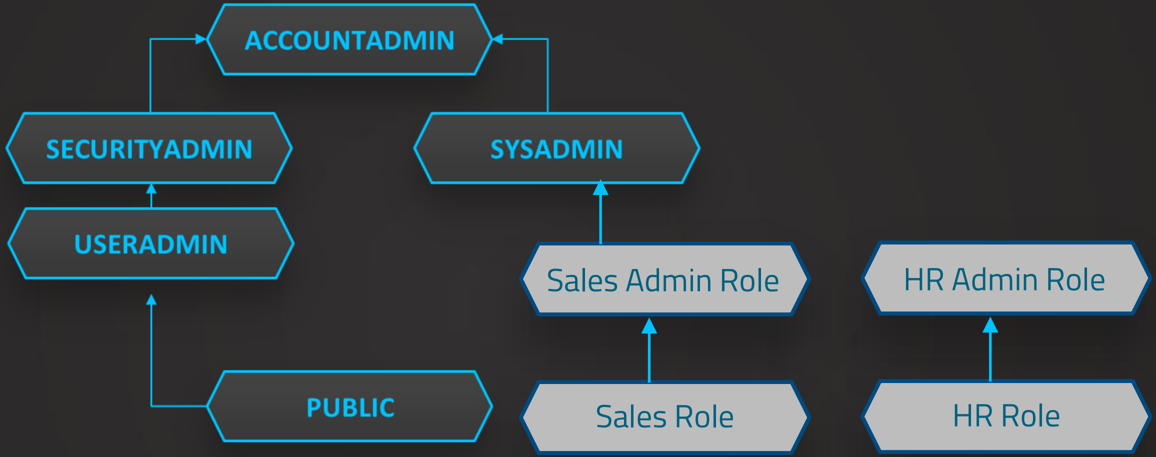
Then this role also has the ability to grant privileges on warehouses, databases, and other objects to the custom roles.

SYSADMIN



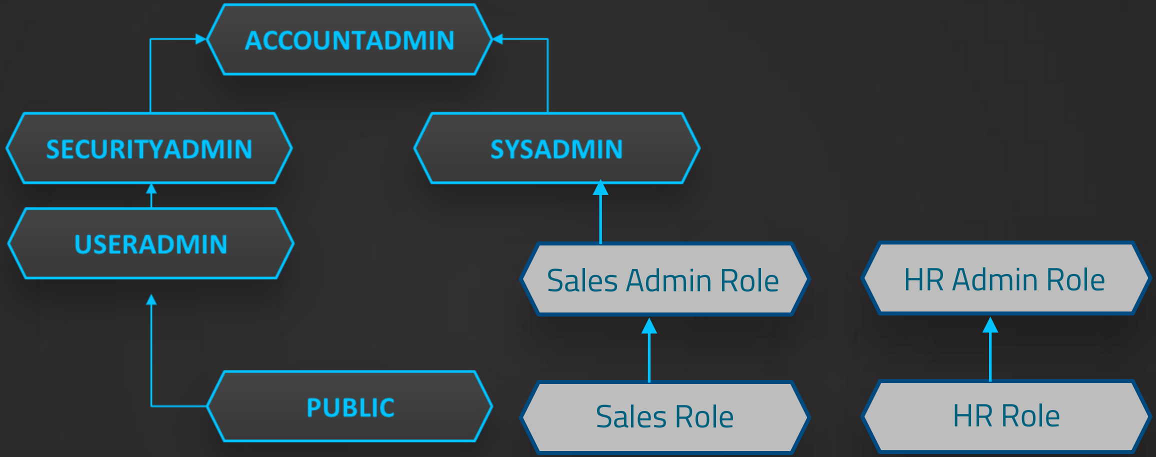
- ✓ Create a virtual warehouse & assign it to the custom roles
- ✓ Create a database and table & assign it to the custom roles

Custom roles



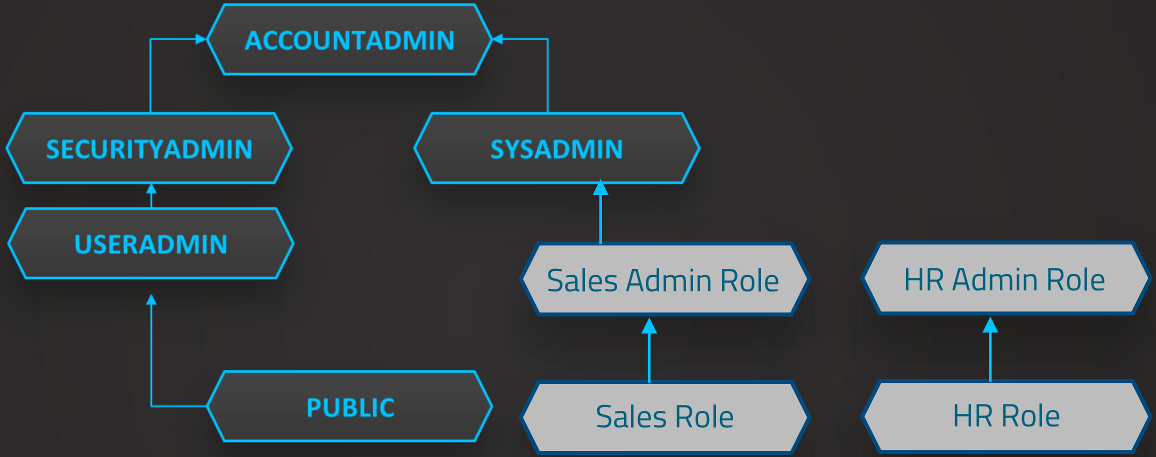
- ✓ Customize roles to our needs & create own hierarchies
- ✓ Custom roles are usually created by SECURITYADMIN
- ✓ Should be leading up to the SYSADMIN role

USERADMIN



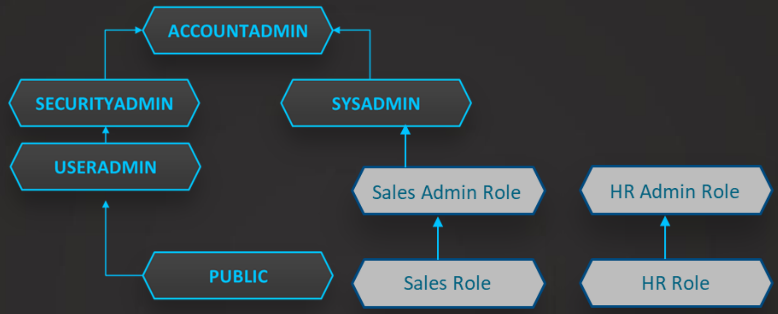
- ✓ Create Users & Roles (User & Role Management)
- ✓ Not for granting privileges (only the one that is owns)

PUBLIC



- ✓ Create Users & Roles (User & Role Management)
- ✓ Not for granting privileges (only the one that is owns)

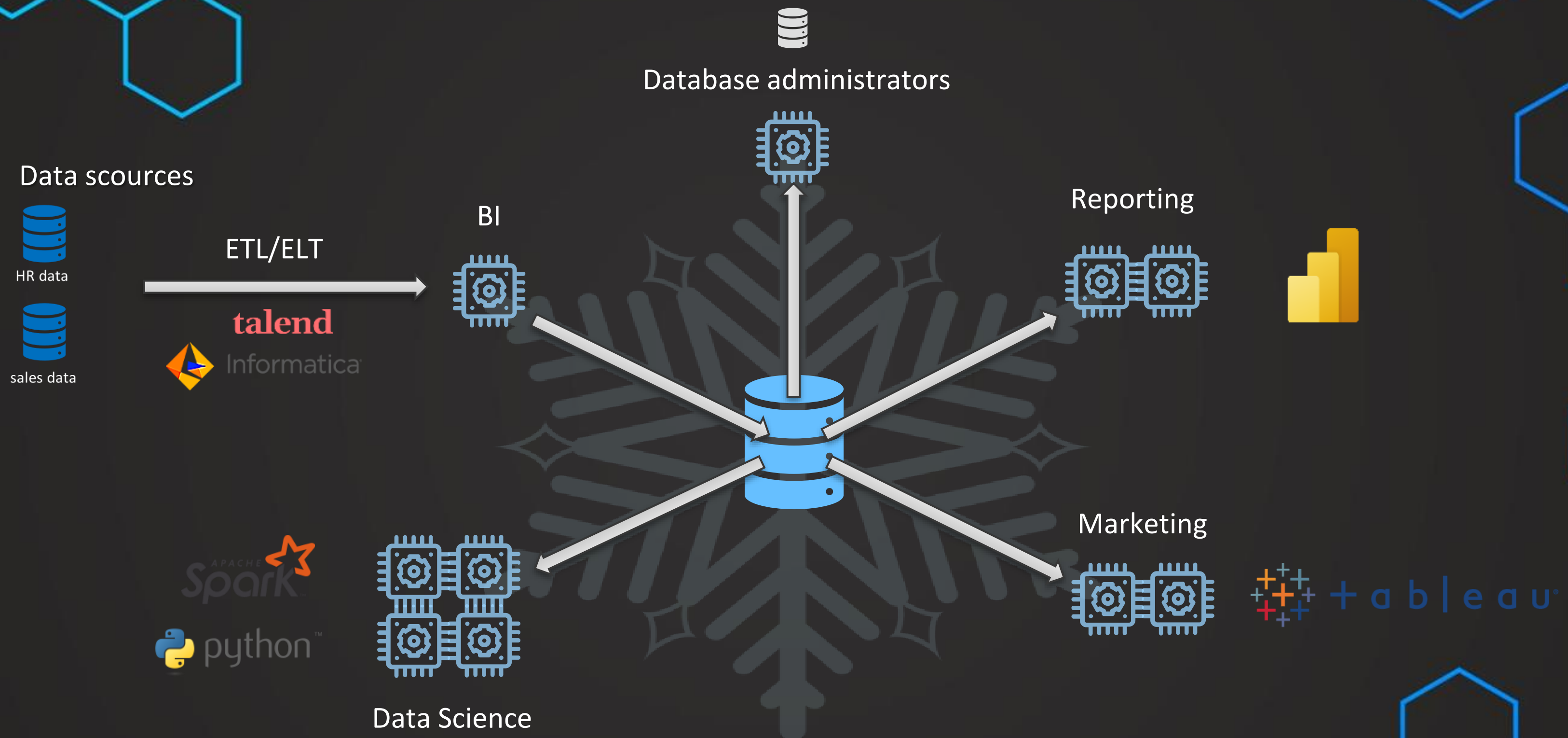
PUBLIC



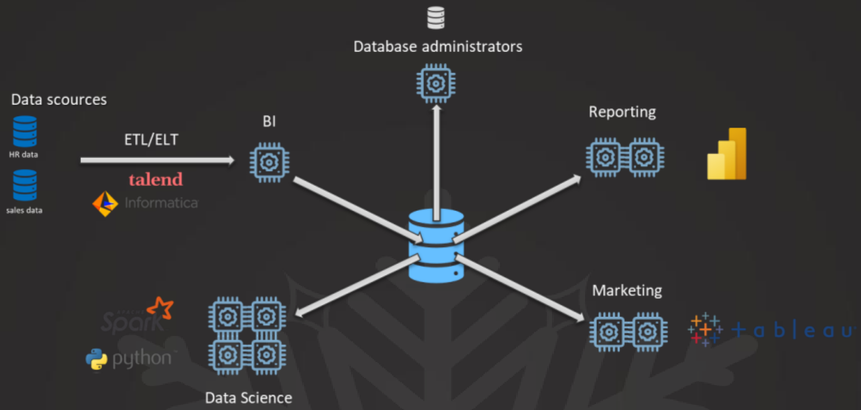
- ✓ Least privileged role (bottom of hierarchy)
- ✓ Every user is automatically assigned to this role
- ✓ Can own objects
- ✓ These objects are then available to everyone

Snowflake & Other Tools

Snowflake & other tools

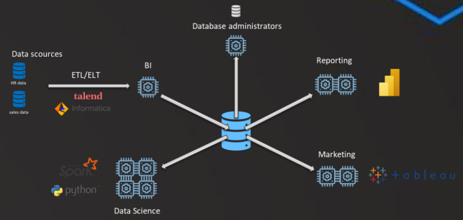


Snowflake & other tools



- ✓ Create easily trial accounts with Snowflake partners
- ✓ Convenient option for trying 3rd-party tools

Snowflake & other tools



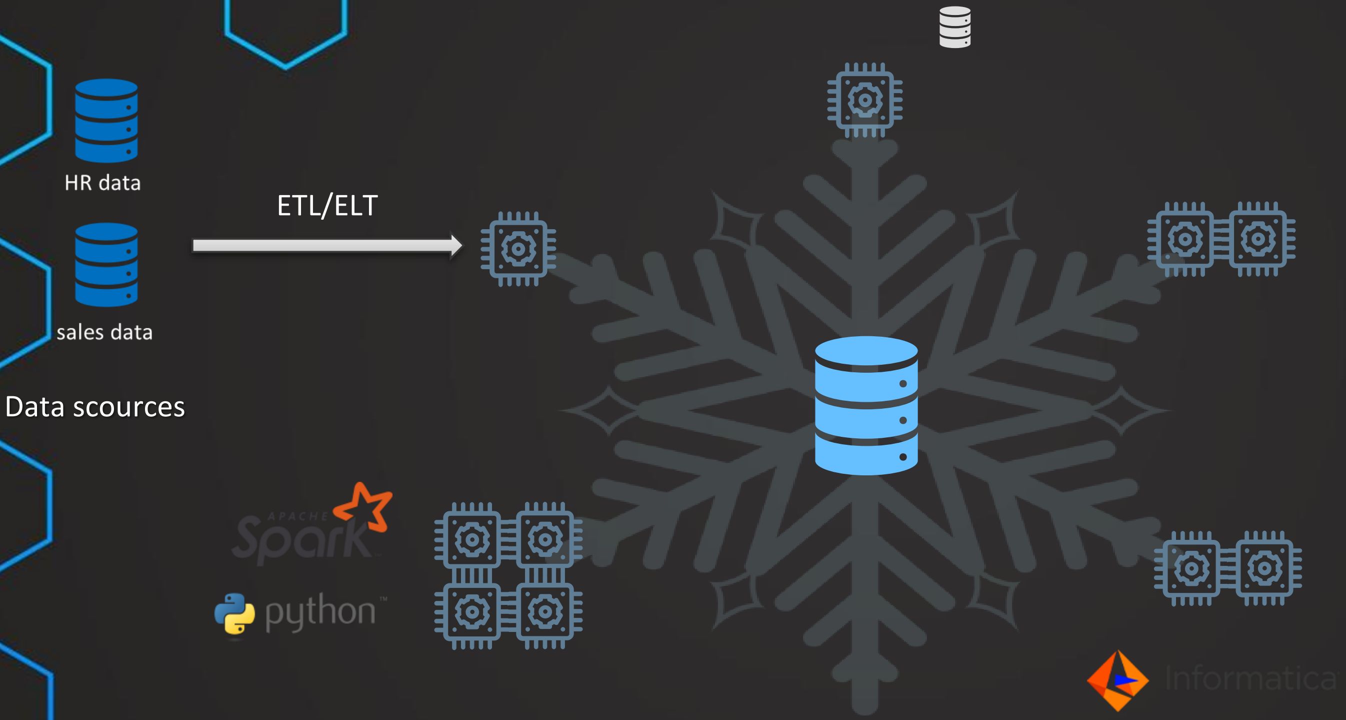
- ✓ **ETL/data integration tools – Moving & transforming data**
- ✓ **Machine Learning & Data Science tools**
- ✓ **Security & Governance**

Best Practices

Best practices

- ✓ Virtual warehouses
- ✓ Table design
- ✓ Monitoring
- ✓ Retention period

How does it work in Snowflake?



Virtual warehouse

- ✓ **Best Practice #1 – Enable Auto-Suspend**
- ✓ **Best Practice #2 – Enable Auto-Resume**
- ✓ **Best Practice #3 – Set appropriate timeouts**

	ETL / Data Loading	BI / SELECT queries	DevOps / Data Science
Timeout	Immediately	10 min	5 min

Table design

- ✓ Best Practice #1 – Appropiate table type
  - ✓ Staging tables – Transient
  - ✓ Productive tables – Permanent
  - ✓ Development tables – Transient

Table design

- ✓ Best Practice #1 – Appropiate table type
- ✓ Best Practice #2 – Appropiate data type
- ✓ Best Practice #3 – Set cluster keys only if necesarry
  - ✓ Large table
  - ✓ Most query time for table scan
  - ✓ Dimensions

Retention period

- ✓ **Best Practice #1: Staging database – 0 days (transient)**
- ✓ **Best Practice #2 – Production – 4-7 days (1 day min)**
- ✓ **Best Practice #3 – Large high-churn tables – 0 days (transient)**

	Active	Time Travel	Fail Safe
Timeout	20GB	400GB	2.8TB