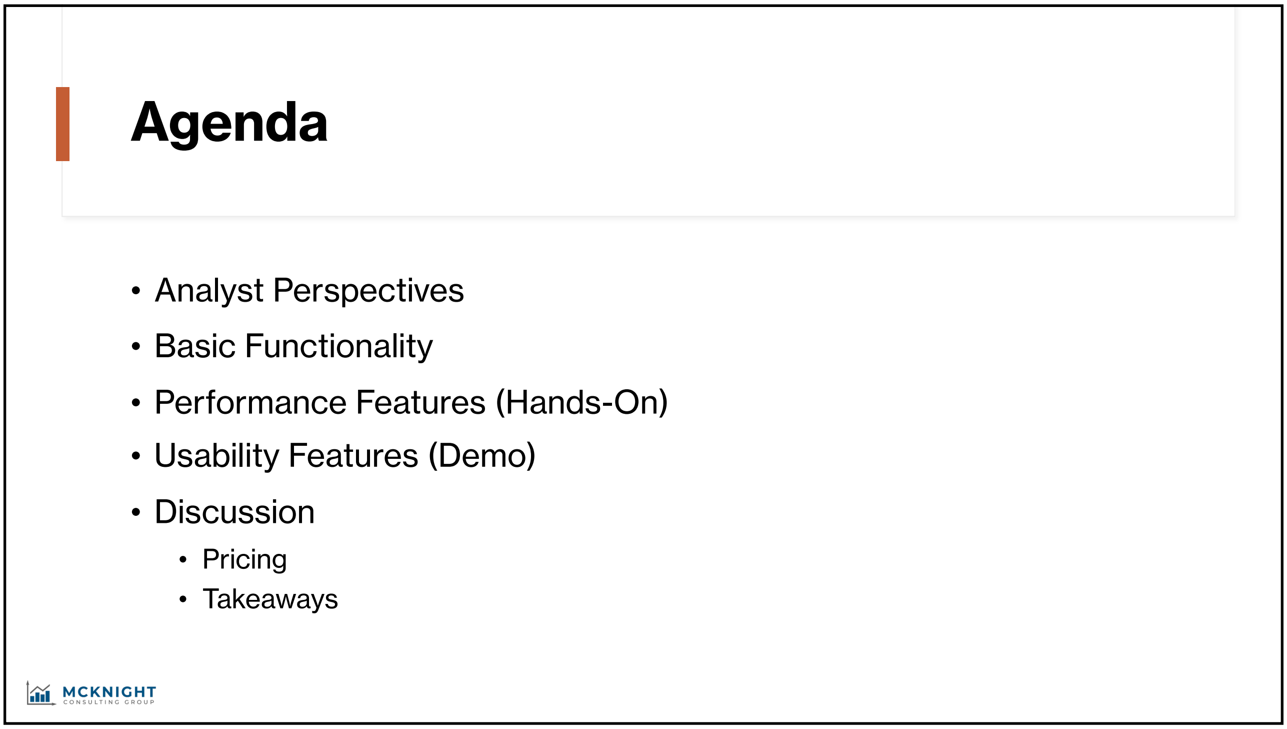
databricks

Competitive Hands-On Training

2023 Q1

McKnight Consulting Group

1

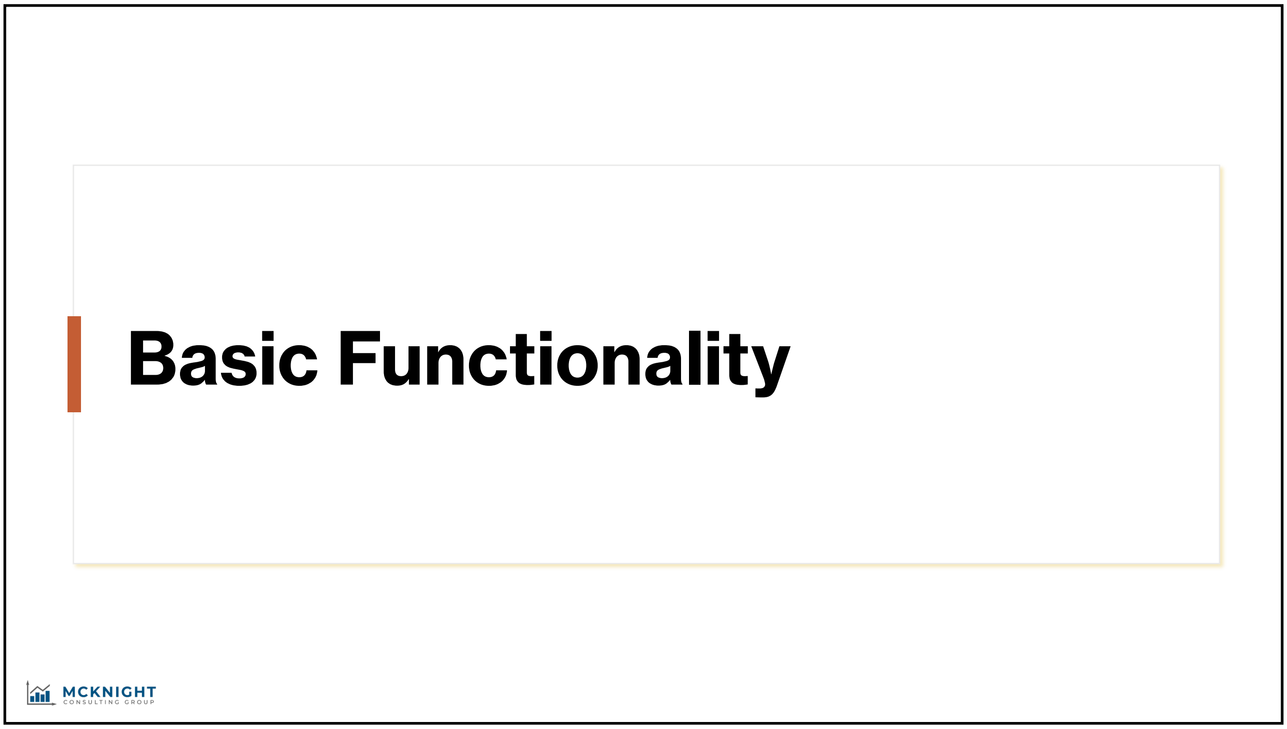


Agenda

- Analyst Perspectives
- Basic Functionality
- Performance Features (Hands-On)
- Usability Features (Demo)
- Discussion
  - Pricing
  - Takeaways

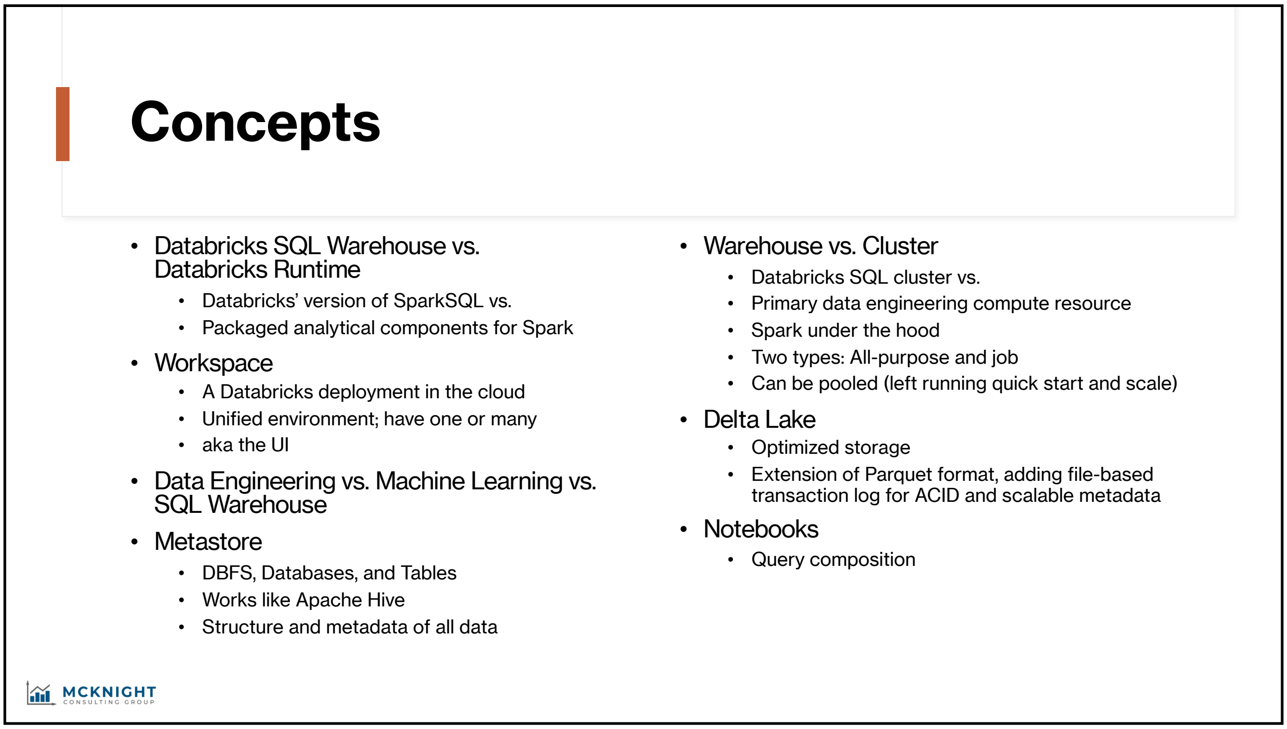
MCKNIGHT CONSULTING GROUP

2



Basic Functionality

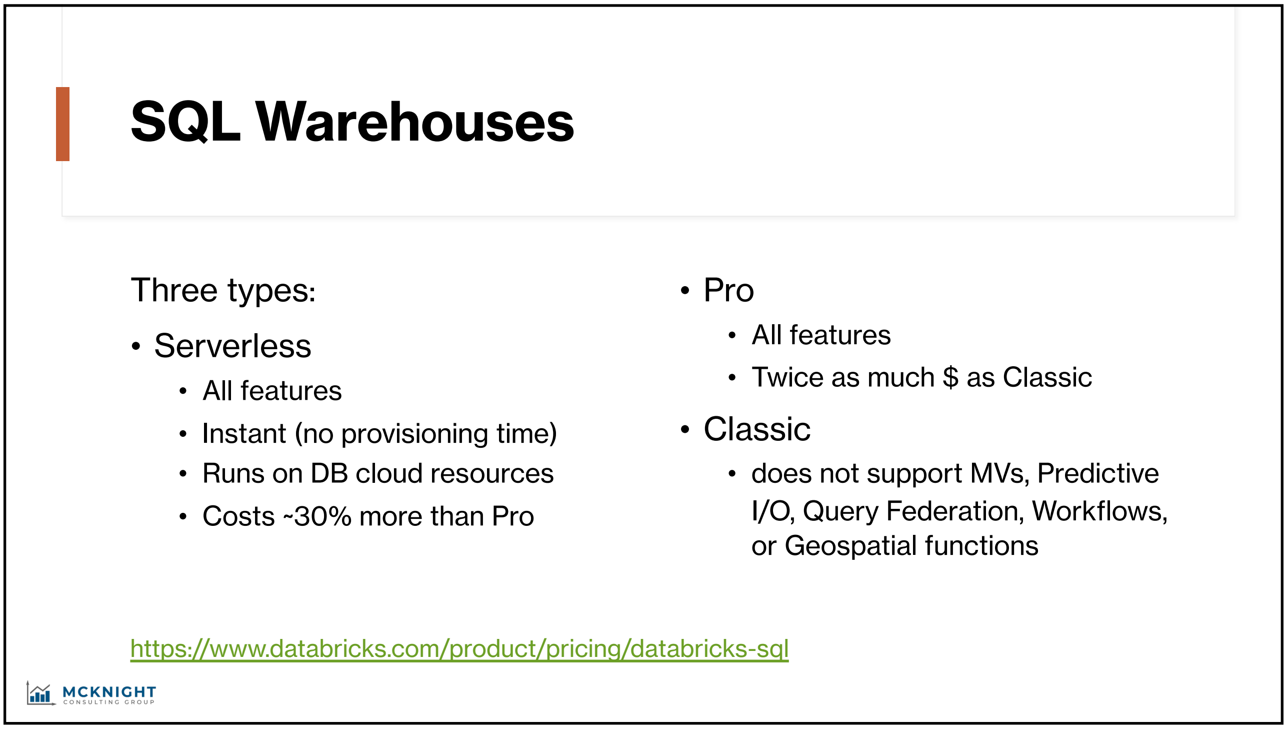
3



Concepts

- Databricks SQL Warehouse vs. Databricks Runtime
  - Databricks' version of SparkSQL vs.
  - Packaged analytical components for Spark
- Workspace
  - A Databricks deployment in the cloud
  - Unified environment; have one or many
  - aka the UI
- Data Engineering vs. Machine Learning vs. SQL Warehouse
- Metastore
  - DBFS, Databases, and Tables
  - Works like Apache Hive
  - Structure and metadata of all data
- Warehouse vs. Cluster
  - Databricks SQL cluster vs.
  - Primary data engineering compute resource
  - Spark under the hood
  - Two types: All-purpose and job
  - Can be pooled (left running quick start and scale)
- Delta Lake
  - Optimized storage
  - Extension of Parquet format, adding file-based transaction log for ACID and scalable metadata
- Notebooks
  - Query composition

4



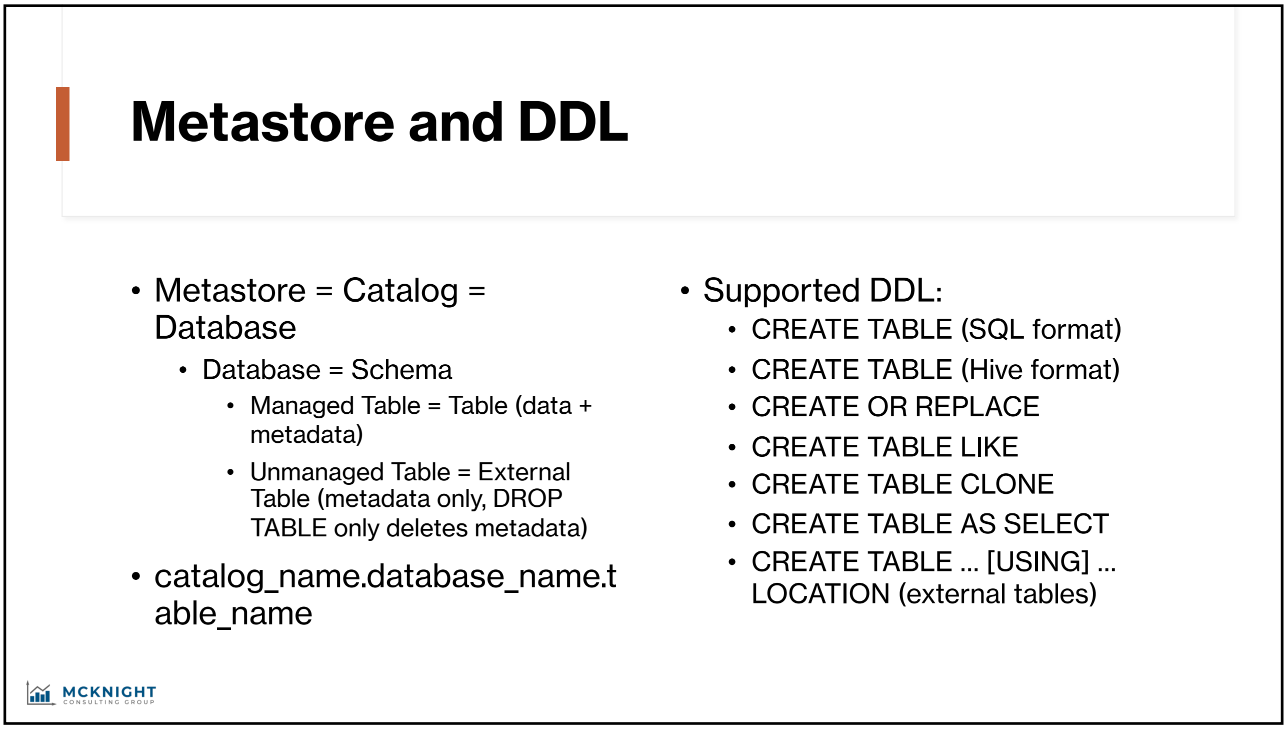
SQL Warehouses

Three types:

- Serverless
  - All features
  - Instant (no provisioning time)
  - Runs on DB cloud resources
  - Costs ~30% more than Pro
- Pro
  - All features
  - Twice as much $ as Classic
- Classic
  - does not support MVs, Predictive I/O, Query Federation, Workflows, or Geospatial functions

https://www.databricks.com/product/pricing/databricks-sql

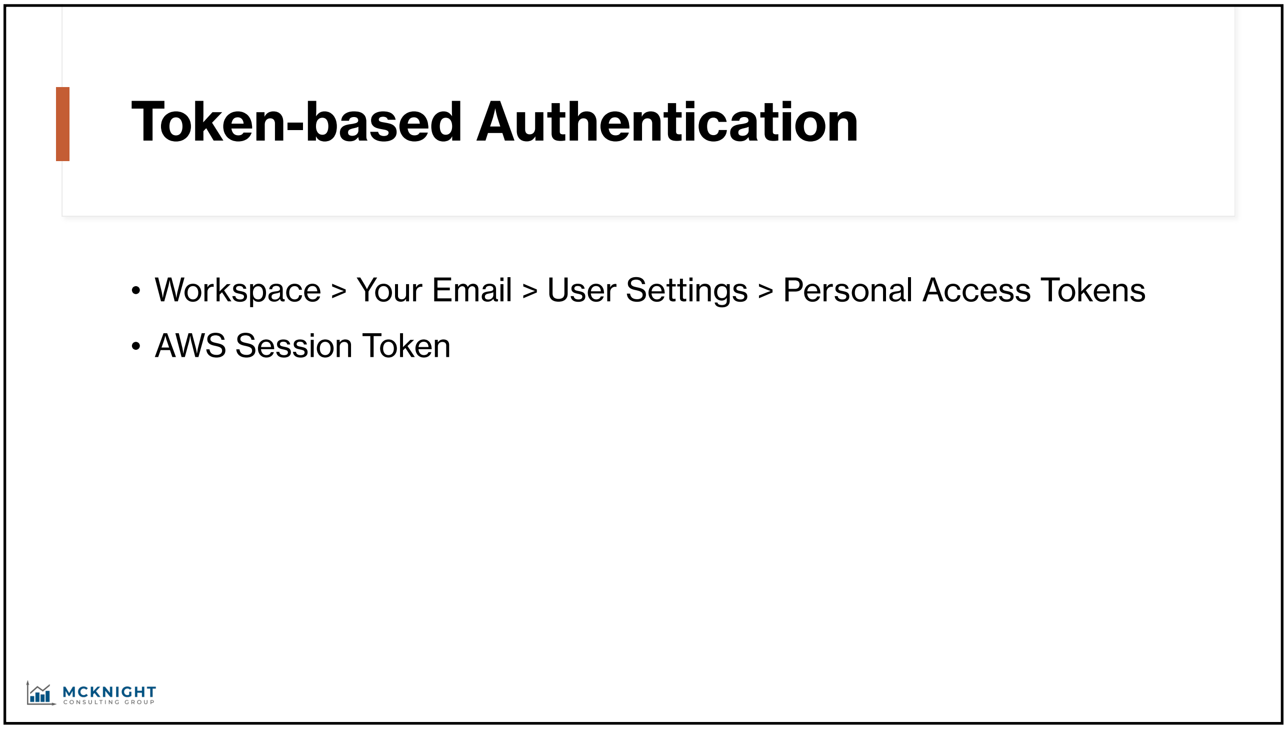
5



Metastore and DDL

- Metastore = Catalog = Database
  - Database = Schema
    - Managed Table = Table (data + metadata)
    - Unmanaged Table = External Table (metadata only, DROP TABLE only deletes metadata)
- catalog_name.database_name.table_name
- Supported DDL:
  - CREATE TABLE (SQL format)
  - CREATE TABLE (Hive format)
  - CREATE OR REPLACE
  - CREATE TABLE LIKE
  - CREATE TABLE CLONE
  - CREATE TABLE AS SELECT
  - CREATE TABLE ... [USING] ... LOCATION (external tables)

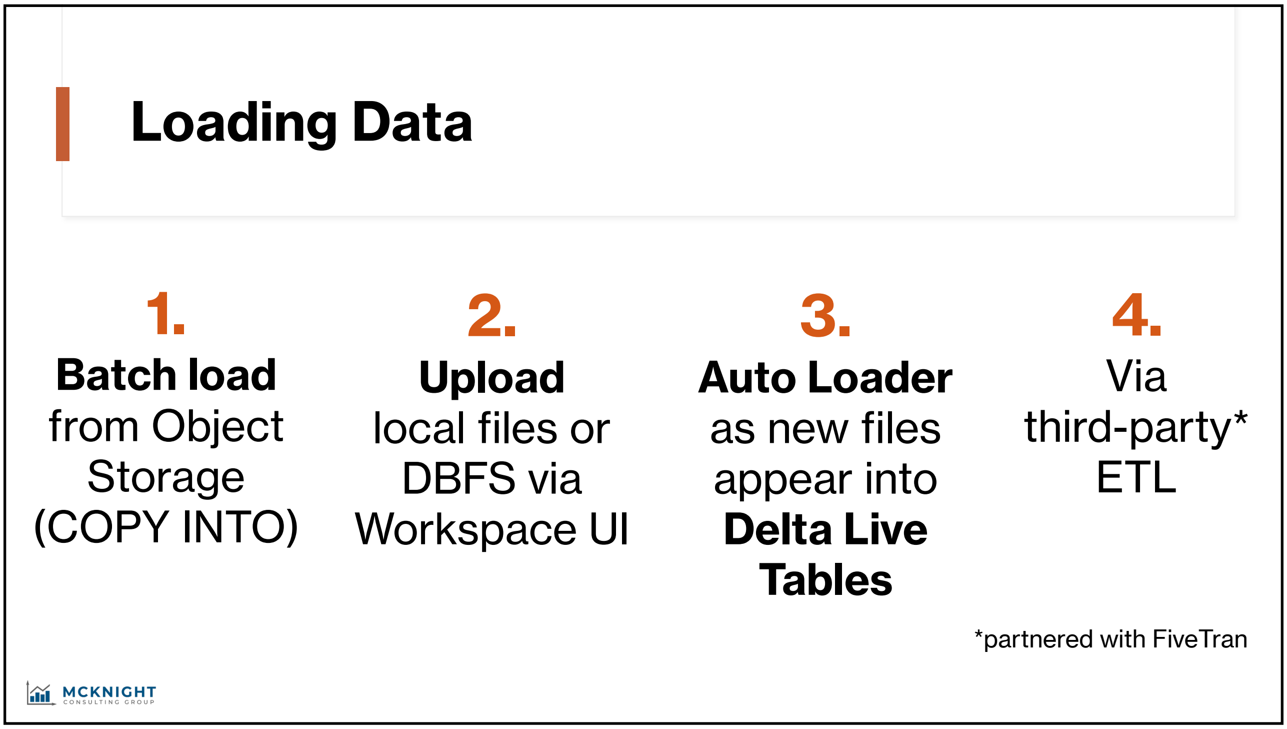
6



Token-based Authentication

- Workspace > Your Email > User Settings > Personal Access Tokens
- AWS Session Token

7



Loading Data

1. **Batch load** from Object Storage (COPY INTO)

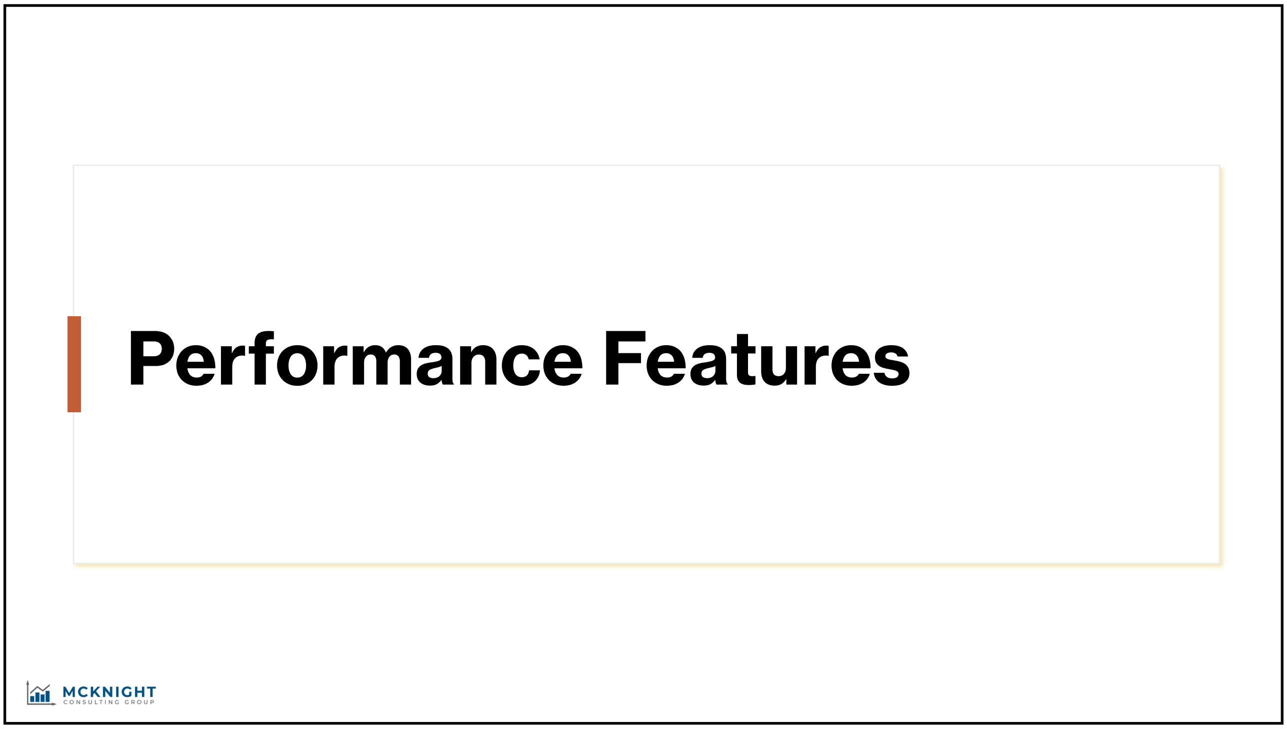
2. **Upload** local files or DBFS via Workspace UI

3. **Auto Loader** as new files appear into **Delta Live Tables**

4. Via third-party* ETL

*partnered with FiveTran

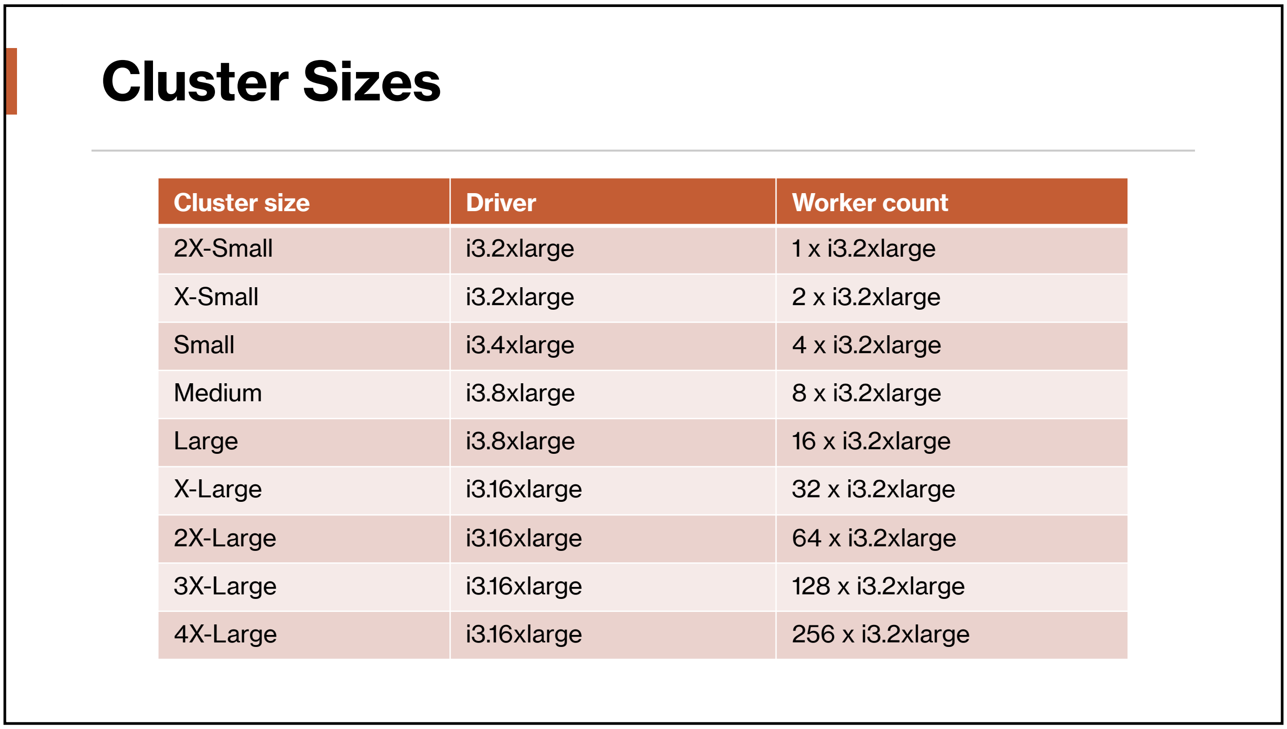
8



Performance Features

MCKNIGHT CONSULTING GROUP

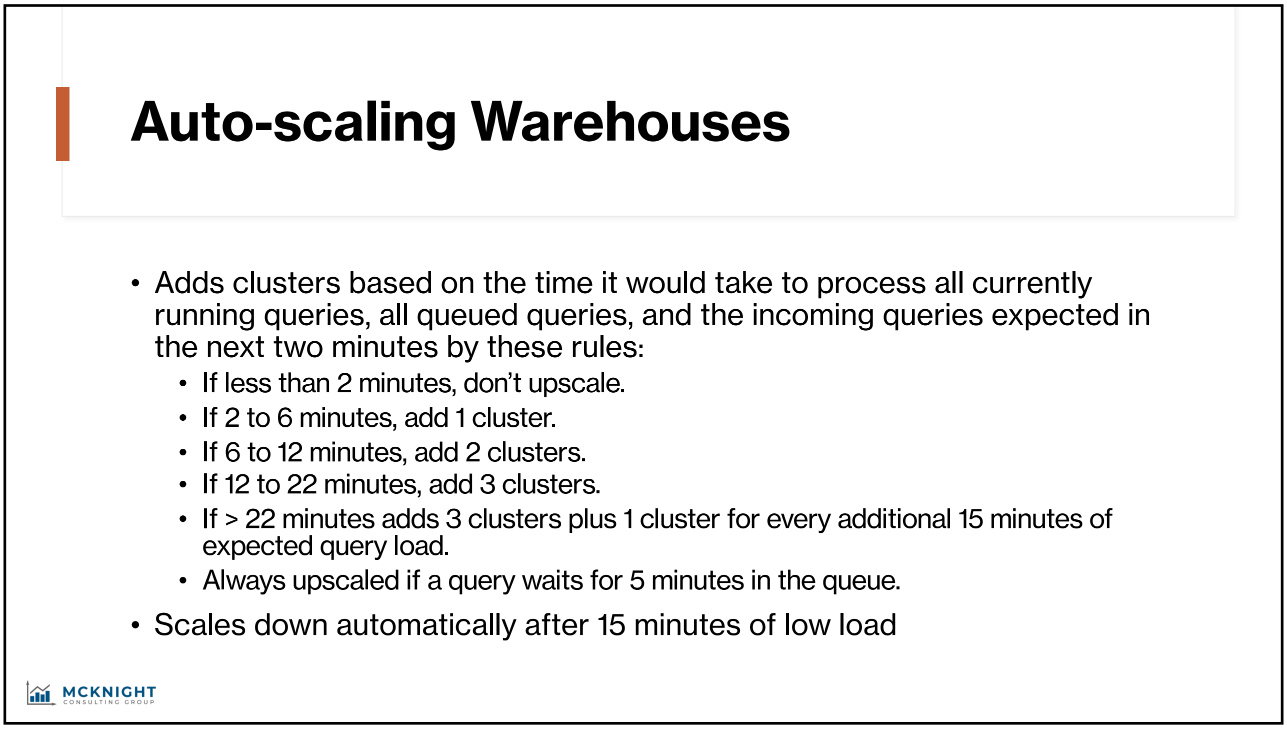
9



Cluster Sizes

Cluster size	Driver	Worker count
2X-Small	i3.2xlarge	1 x i3.2xlarge
X-Small	i3.2xlarge	2 x i3.2xlarge
Small	i3.4xlarge	4 x i3.2xlarge
Medium	i3.8xlarge	8 x i3.2xlarge
Large	i3.8xlarge	16 x i3.2xlarge
X-Large	i3.16xlarge	32 x i3.2xlarge
2X-Large	i3.16xlarge	64 x i3.2xlarge
3X-Large	i3.16xlarge	128 x i3.2xlarge
4X-Large	i3.16xlarge	256 x i3.2xlarge

10

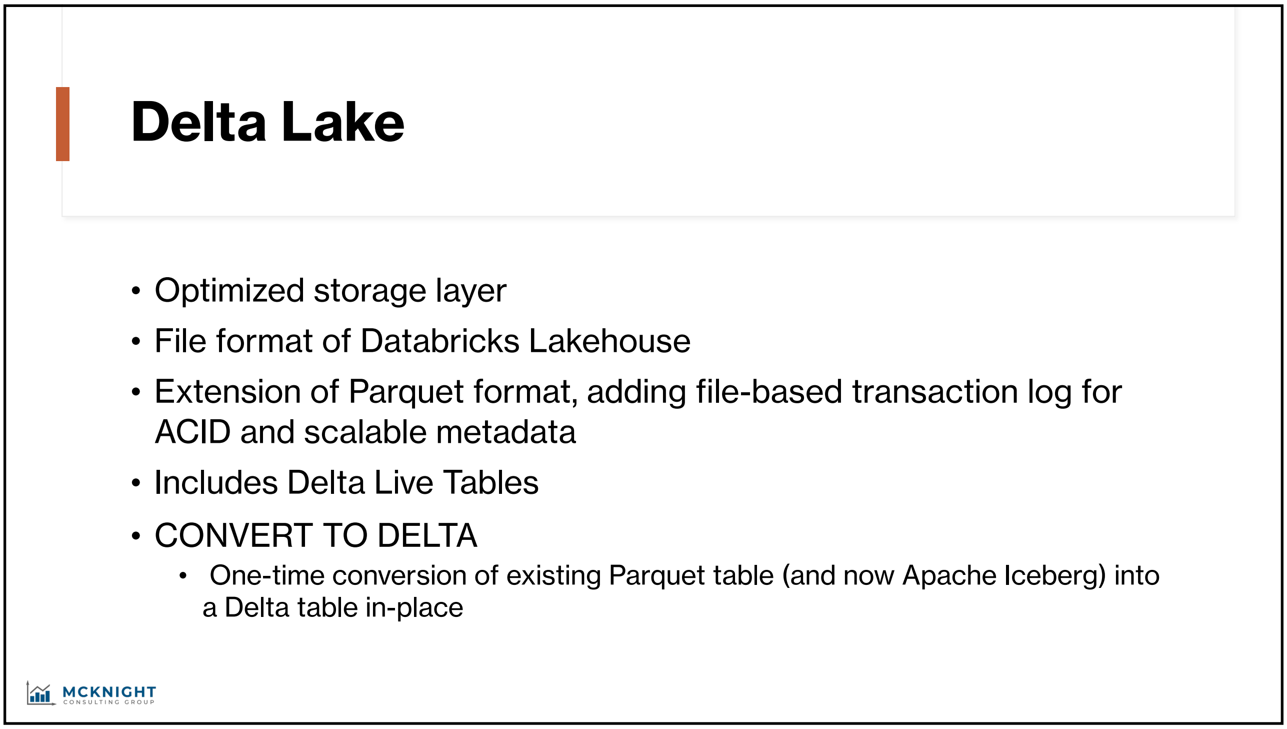


Auto-scaling Warehouses

- Adds clusters based on the time it would take to process all currently running queries, all queued queries, and the incoming queries expected in the next two minutes by these rules:
  - If less than 2 minutes, don't upscale.
  - If 2 to 6 minutes, add 1 cluster.
  - If 6 to 12 minutes, add 2 clusters.
  - If 12 to 22 minutes, add 3 clusters.
  - If > 22 minutes adds 3 clusters plus 1 cluster for every additional 15 minutes of expected query load.
  - Always upscaled if a query waits for 5 minutes in the queue.
- Scales down automatically after 15 minutes of low load

MCKNIGHT CONSULTING GROUP

11

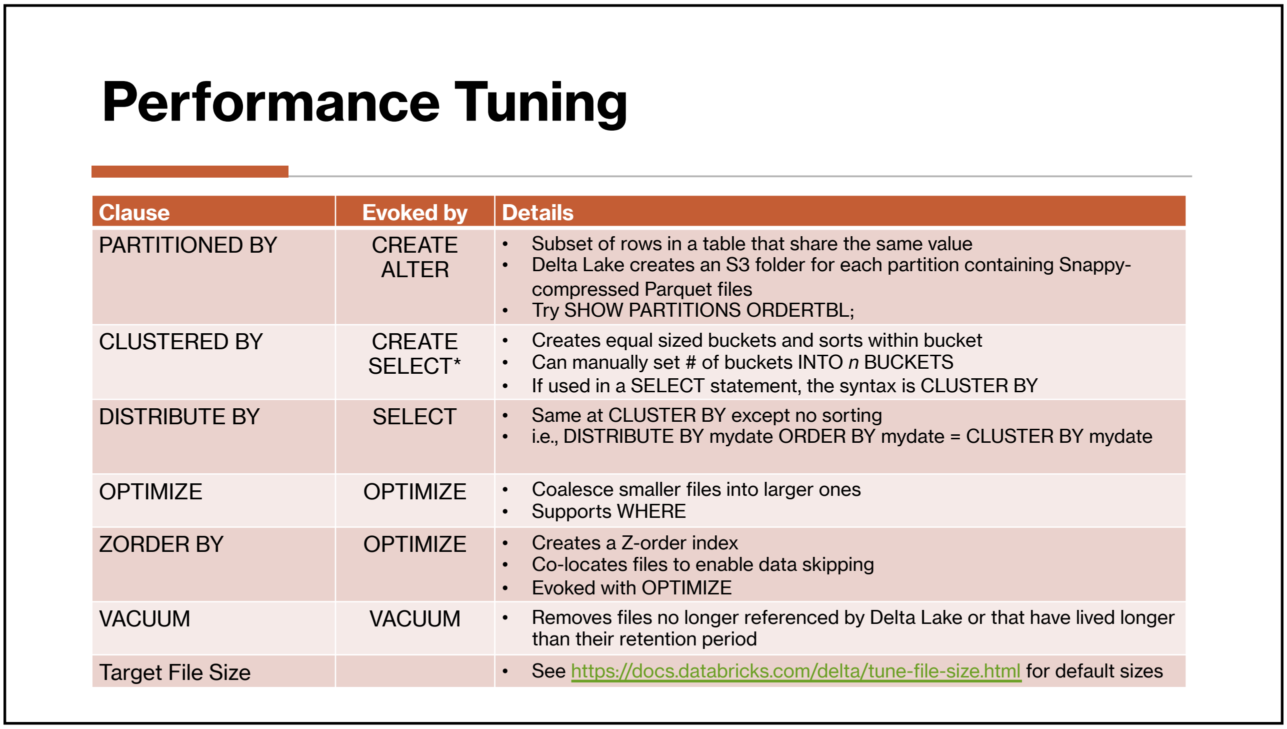


Delta Lake

- Optimized storage layer
- File format of Databricks Lakehouse
- Extension of Parquet format, adding file-based transaction log for ACID and scalable metadata
- Includes Delta Live Tables
- CONVERT TO DELTA
  - One-time conversion of existing Parquet table (and now Apache Iceberg) into a Delta table in-place

MCKNIGHT CONSULTING GROUP

12



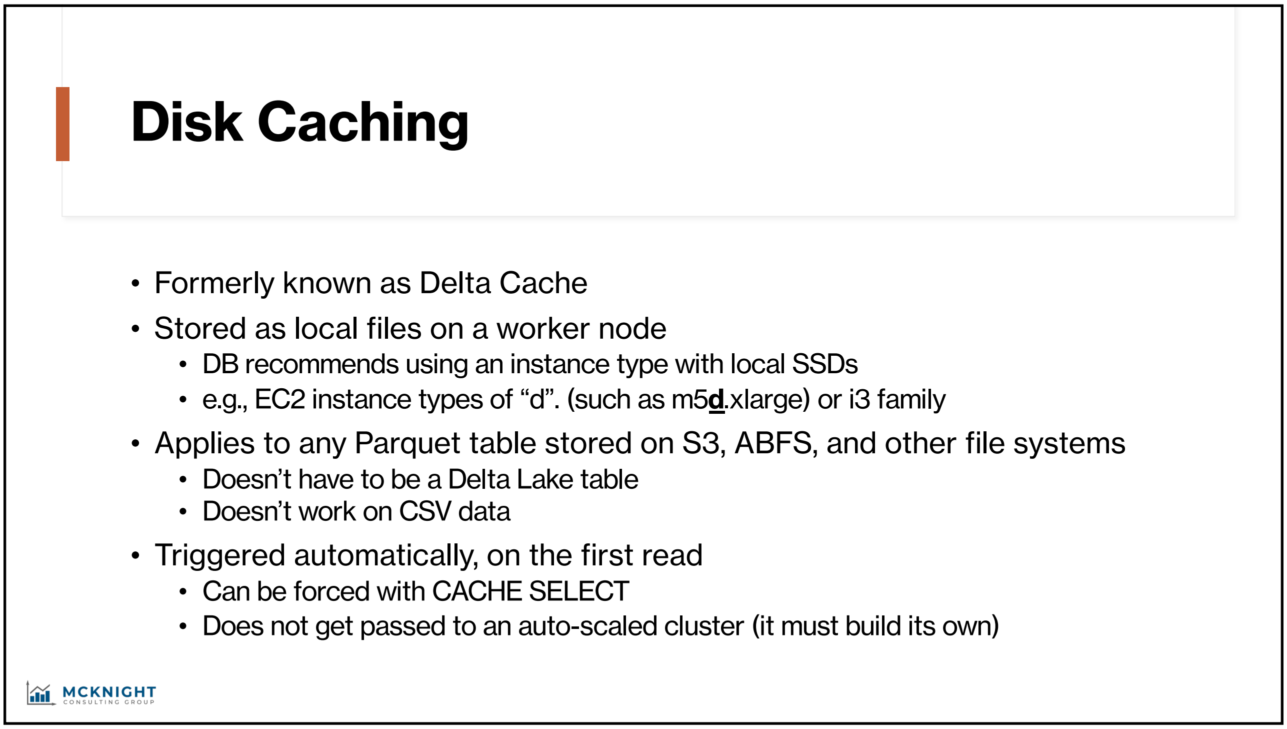
Performance Tuning

Clause	Evoked by	Details
PARTITIONED BY	CREATE ALTER	• Subset of rows in a table that share the same value • Delta Lake creates an S3 folder for each partition containing Snappy-compressed Parquet files • Try SHOW PARTITIONS ORDERTBL;

CLUSTERED BY	CREATE
	SELECT*
		• Creates equal sized buckets and sorts within bucket
		• Can manually set # of buckets INTO *n* BUCKETS
		• If used in a SELECT statement, the syntax is CLUSTER BY

DISTRIBUTE BY	SELECT	• Same at CLUSTER BY except no sorting • i.e., DISTRIBUTE BY mydate ORDER BY mydate = CLUSTER BY mydate
OPTIMIZE	OPTIMIZE	• Coalesce smaller files into larger ones • Supports WHERE
ZORDER BY	OPTIMIZE	• Creates a Z-order index • Co-locates files to enable data skipping • Evoked with OPTIMIZE
VACUUM	VACUUM	• Removes files no longer referenced by Delta Lake or that have lived longer than their retention period
Target File Size		• See https://docs.databricks.com/delta/tune-file-size.html for default sizes

13



Disk Caching

- Formerly known as Delta Cache
- Stored as local files on a worker node
  - DB recommends using an instance type with local SSDs
  - e.g., EC2 instance types of “d”. (such as m5**d**.xlarge) or i3 family
- Applies to any Parquet table stored on S3, ABFS, and other file systems
  - Doesn’t have to be a Delta Lake table
  - Doesn’t work on CSV data
- Triggered automatically, on the first read
  - Can be forced with CACHE SELECT
  - Does not get passed to an auto-scaled cluster (it must build its own)

MCKNIGHT CONSULTING GROUP

14

Usability Features

MCKNIGHT CONSULTING GROUP

15

Filters, Parameters, and Visualizations

16

Scheduled Queries and Alerts

- Scheduled query executions keep dashboards updated or enable routine alerts
- Saved SQL queries from Query Editor
- Supported intervals: 1-30 min, 1-12 hr, 1 or 30 days, 1 or 2 weeks
- Supports “Run as Owner” or “Run as Viewer” privileges

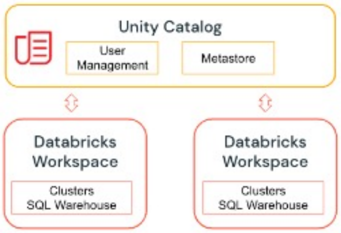
- Alerts notify people when a field returned by a scheduled query meets a threshold
- Supports custom message <html> markup
- Supported destinations:
    - Email
    - Slack
    - Webhook
    - PagerDuty
    - Teams

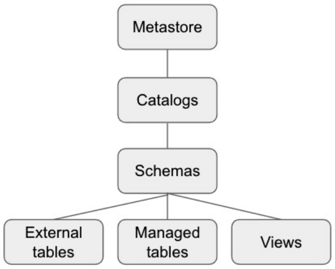
MCKNIGHT CONSULTING GROUP

17

Unity Catalog

- Unity Catalog is the Databricks data governance solution for the Lakehouse
- Manage centrally across all the workspaces in a Databricks account
- Users in different workspaces can share access to the same data, depending on privileges granted
- Define once, secure everywhere model





18

Dynamic Views

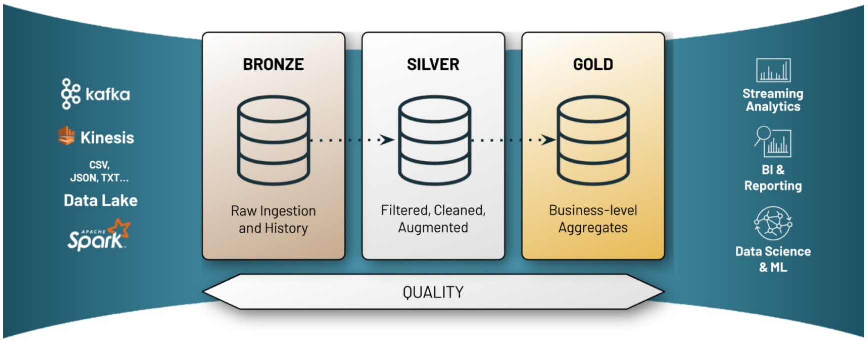
- Used for data masking
- Supports three methods:
  - current_user()
  - is_account_group_member()
  - is_member()
- Evoked by CASE WHEN THEN ELSE END
- For column-level masking, put the CASE statement in the SELECT columns
- For row-level masking, put the CASE statement in the WHERE clause
- For custom masks, use regexp_extract()
- Scala, R, and Machine Learning Runtime workloads not supported

MCKNIGHT CONSULTING GROUP

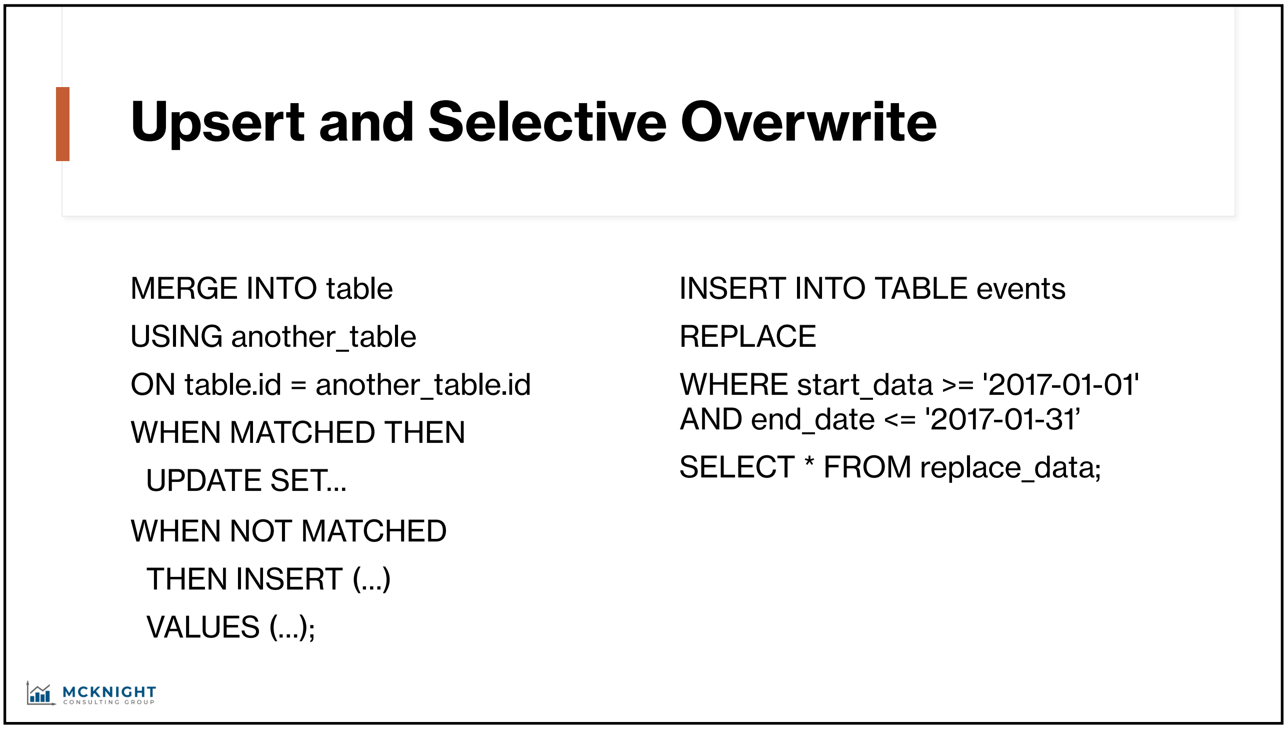
19

Delta Live Tables

- Framework for building reliable, maintainable, and testable data processing pipelines
- DB's version of Materialized Views
- Built with Pipelines



20

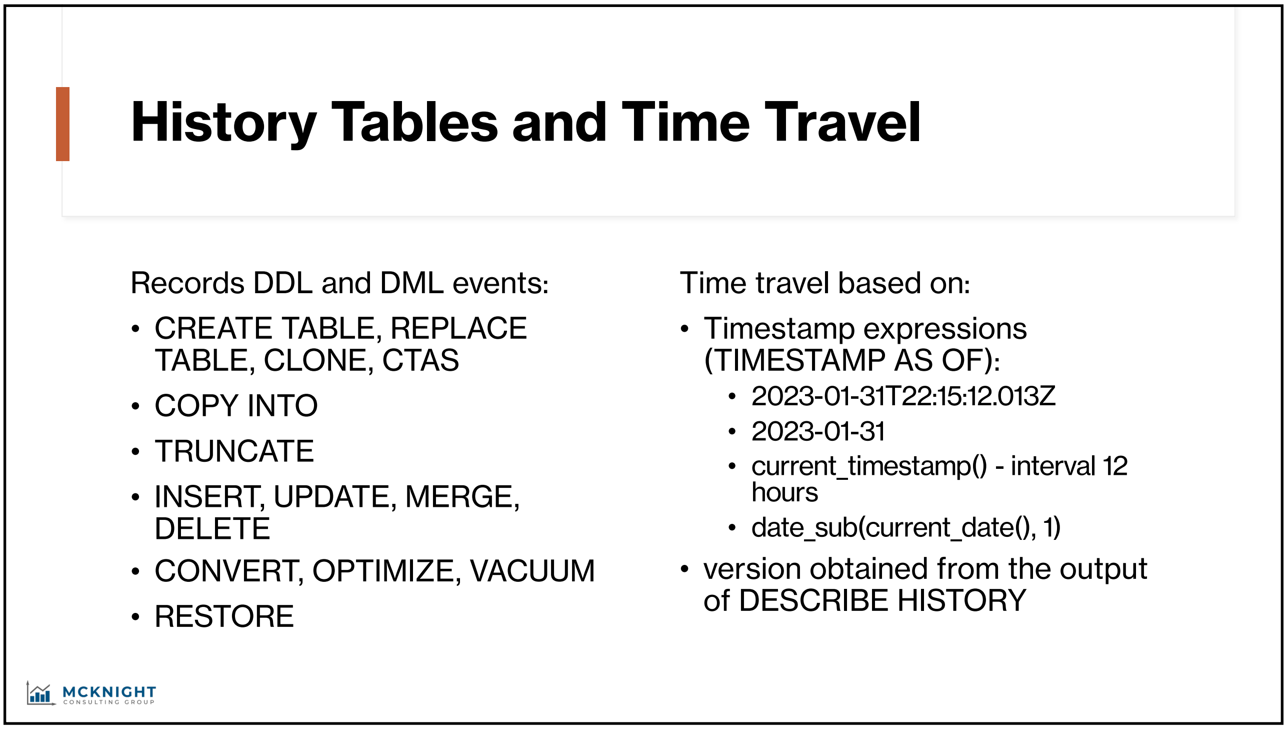


Upsert and Selective Overwrite

```
MERGE INTO table
USING another_table
ON table.id = another_table.id
WHEN MATCHED THEN
  UPDATE SET...
WHEN NOT MATCHED
  THEN INSERT (...)
  VALUES (...);
```

```
INSERT INTO TABLE events
REPLACE
WHERE start_data >= '2017-01-01'
AND end_date <= '2017-01-31'
SELECT * FROM replace_data;
```

21



History Tables and Time Travel

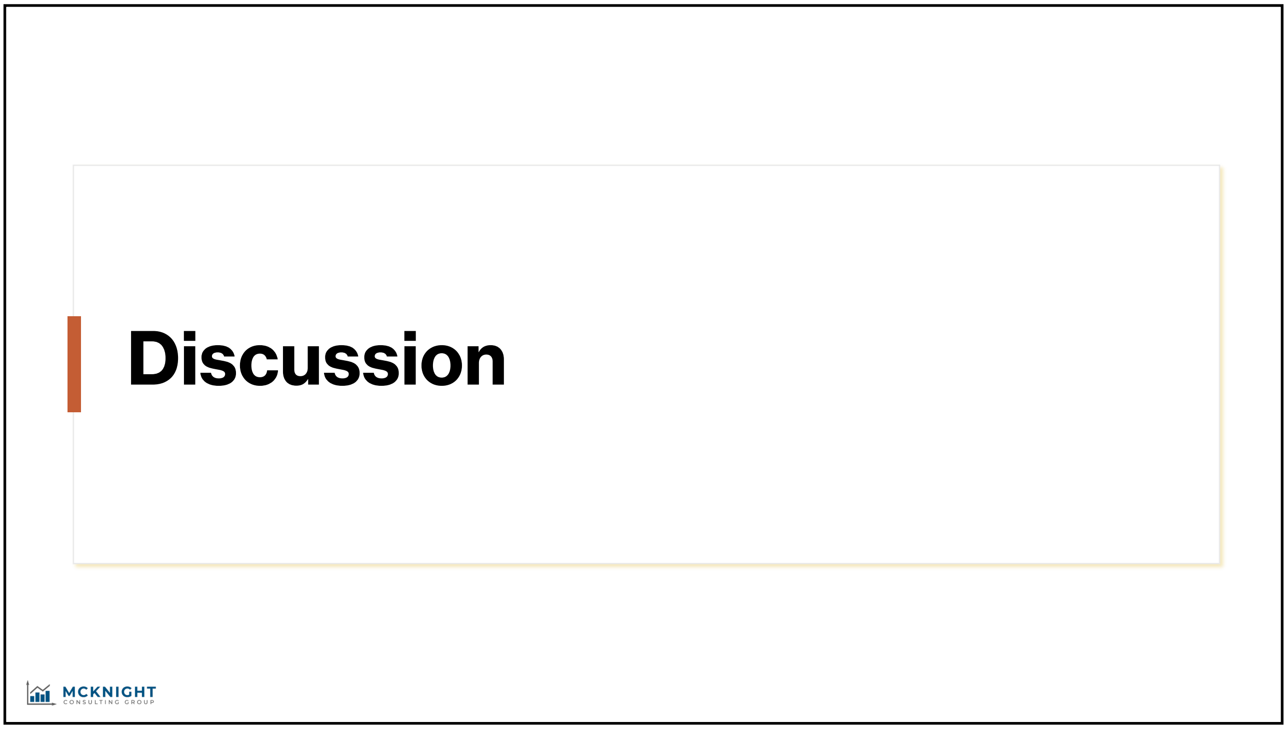
Records DDL and DML events:

- CREATE TABLE, REPLACE TABLE, CLONE, CTAS
- COPY INTO
- TRUNCATE
- INSERT, UPDATE, MERGE, DELETE
- CONVERT, OPTIMIZE, VACUUM
- RESTORE

Time travel based on:

- Timestamp expressions (TIMESTAMP AS OF):
  - 2023-01-31T22:15:12.013Z
  - 2023-01-31
  - current_timestamp() - interval 12 hours
  - date_sub(current_date(), 1)
- version obtained from the output of DESCRIBE HISTORY

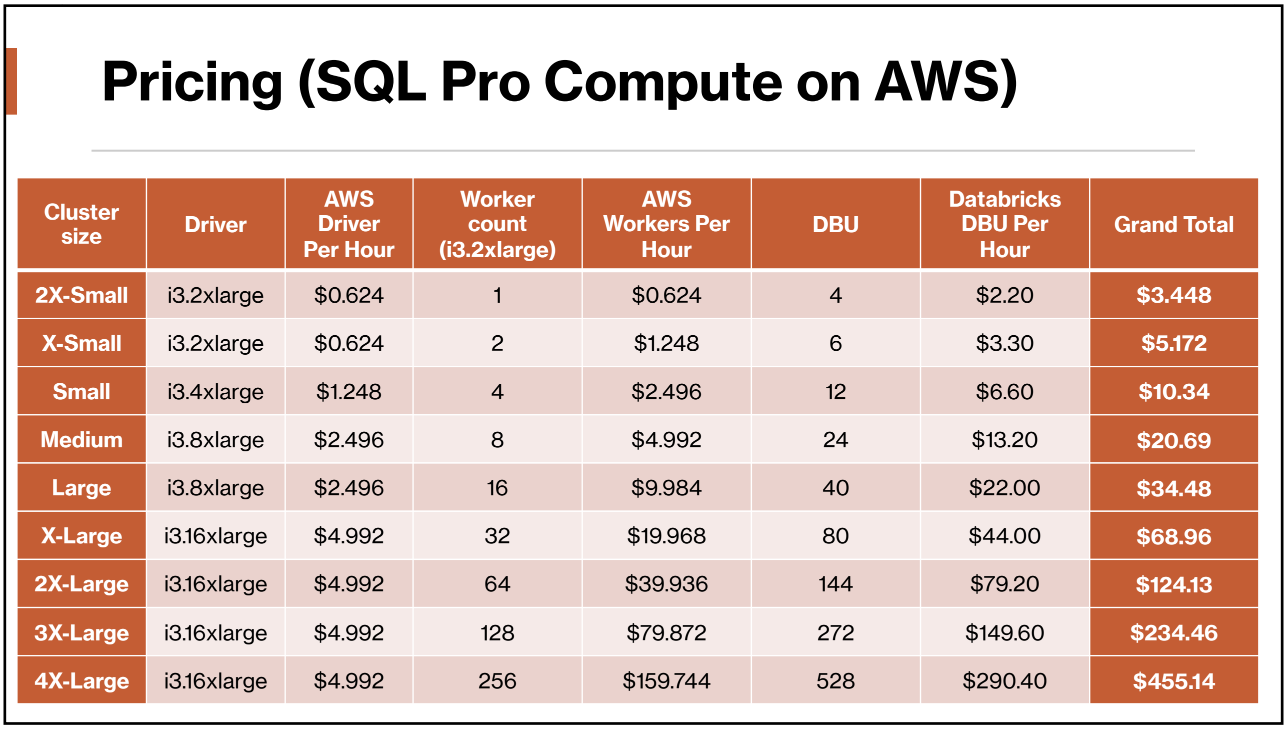
22



Discussion

MCKNIGHT CONSULTING GROUP

23



Pricing (SQL Pro Compute on AWS)

Cluster size	Driver	AWS Driver Per Hour	Worker count (i3.2xlarge)	AWS Workers Per Hour	DBU	Databricks DBU Per Hour	Grand Total
2X-Small	i3.2xlarge	$0.624	1	$0.624	4	$2.20	**$3.448**
X-Small	i3.2xlarge	$0.624	2	$1.248	6	$3.30	**$5.172**
Small	i3.4xlarge	$1.248	4	$2.496	12	$6.60	**$10.34**
Medium	i3.8xlarge	$2.496	8	$4.992	24	$13.20	**$20.69**
Large	i3.8xlarge	$2.496	16	$9.984	40	$22.00	**$34.48**
X-Large	i3.16xlarge	$4.992	32	$19.968	80	$44.00	**$68.96**
2X-Large	i3.16xlarge	$4.992	64	$39.936	144	$79.20	**$124.13**
3X-Large	i3.16xlarge	$4.992	128	$79.872	272	$149.60	**$234.46**
4X-Large	i3.16xlarge	$4.992	256	$159.744	528	$290.40	**$455.14**

24