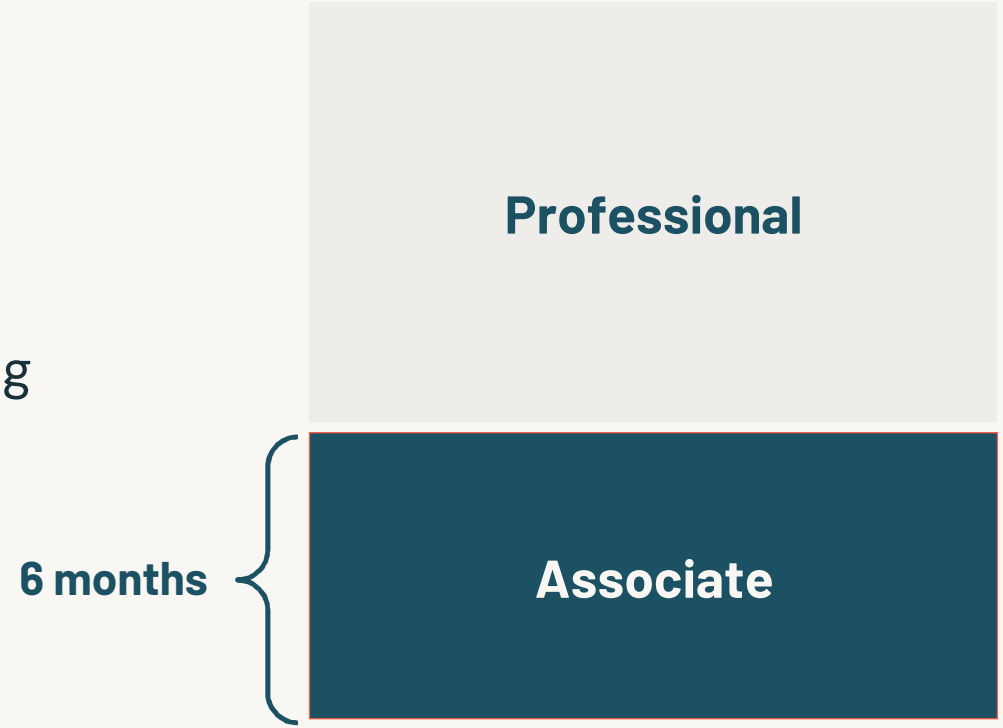
Certification Overview

Databricks Certified Associate Data Engineer Exam

Target Audience

- Data Engineer
- Beginner-level certification
- Assess candidates at a level equivalent to **six months of experience** with data engineering with Databricks



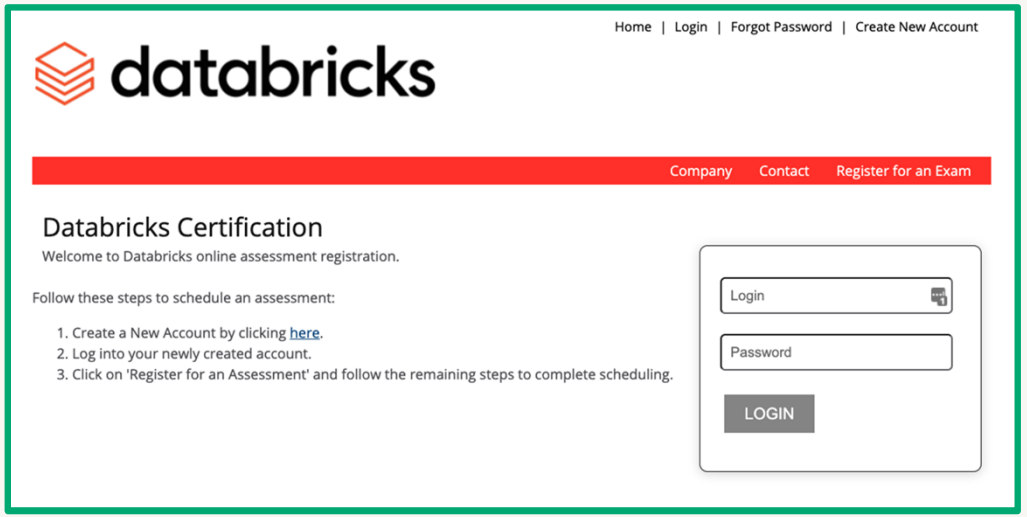
Associate Data Engineer Expectations

Therefore, the following is expected of a Associate-level data engineer:

- Understand how to use and the benefits of using the Databricks Lakehouse Platform and its tools
- Build ETL pipelines using Apache Spark SQL and Python
- Incrementally process data
- Build production pipelines for data engineering applications and Databricks SQL queries and dashboards
- Understand and follow best security practices

Exam Platform

- Databricks Academy certifications are offered through Kryterion's Webassessor platform.
- https://www.webassessor.com/databricks
- Webassessor is a simple, scalable assessment solution resulting in an easy test-taking experience.



Proctoring Details

- During the exam, you will be:
  - Monitored via webcam by a Webassessor proctor.
  - Asked to provide valid, photo-based identification.
- The proctor will:
  - Monitor you during the exam.
  - Answer any exam delivery questions you might have.
  - Provide technical support.
- The proctor will not provide assistance on the content of the exam.
- No test aids will be available during the exam.

Exam Grading

- Certification exams are automatically graded.
- Following the exam, the proctor's session notes and the recorded grade will be reviewed by Databricks Academy
- It will take about one week for you to find out whether or not you passed the exam.

Exam Format and Structure

There are 45 multiple-choice questions on the certification exam. The questions will be distributed by high-level topic in the following way:

A. Databricks Lakehouse Platform – 24% (11/45)

B. ELT with Spark SQL and Python – 29% (13/45)

C. Incremental Data Processing – 22% (10/45)

D. Production Pipelines – 16% (7/45)

E. Data Governance – 9% (4/45)

Databricks Lakehouse Platform (24%)

Databricks Lakehouse Platform (24%)

The minimally qualified candidate should be able to:

Understand how to use, and the benefits of the Databricks Lakehouse Platform, including:

- Lakehouse description and its benefits to Data Teams
- Clusters, Databricks File System (DBFS), Notebooks and Repos
- Delta Lake general concepts, Delta Table Management, Manipulation, and Optimizations

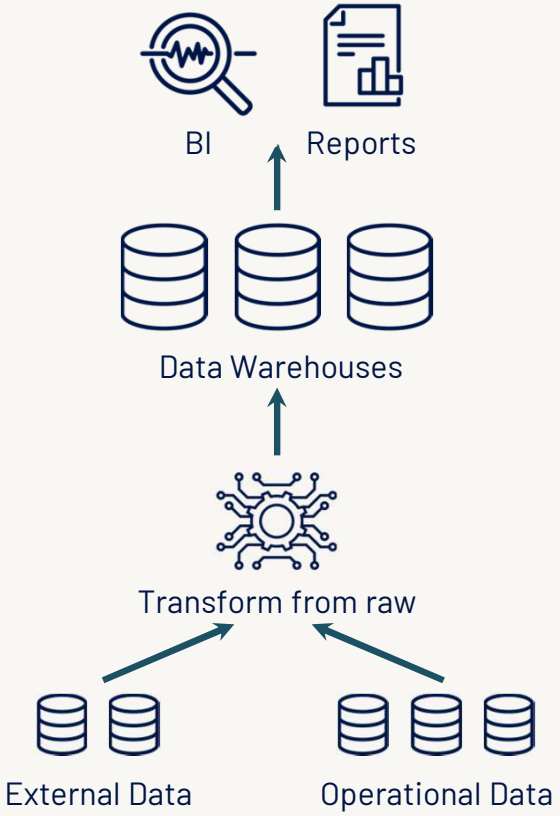
Lakehouse

Data Warehouses

- Purpose built for BI and reporting
- Meant to unify disparate systems

But, DW's have failed to keep up with current use cases:

- Unable to store unstructured data
- Unable to support data science, ML, and streaming
- Natively only support SQL

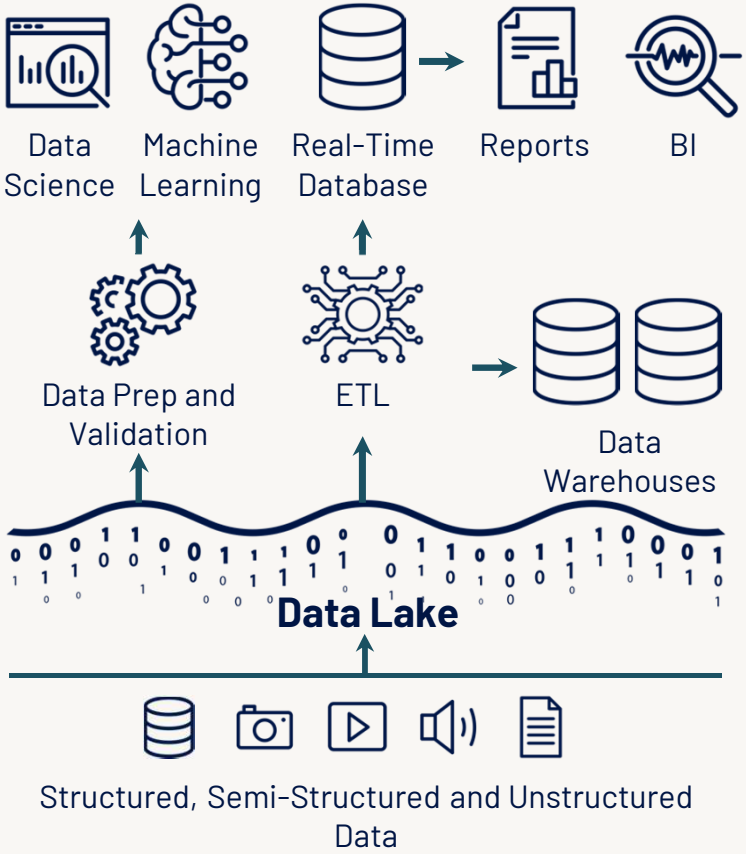


Data Lakes

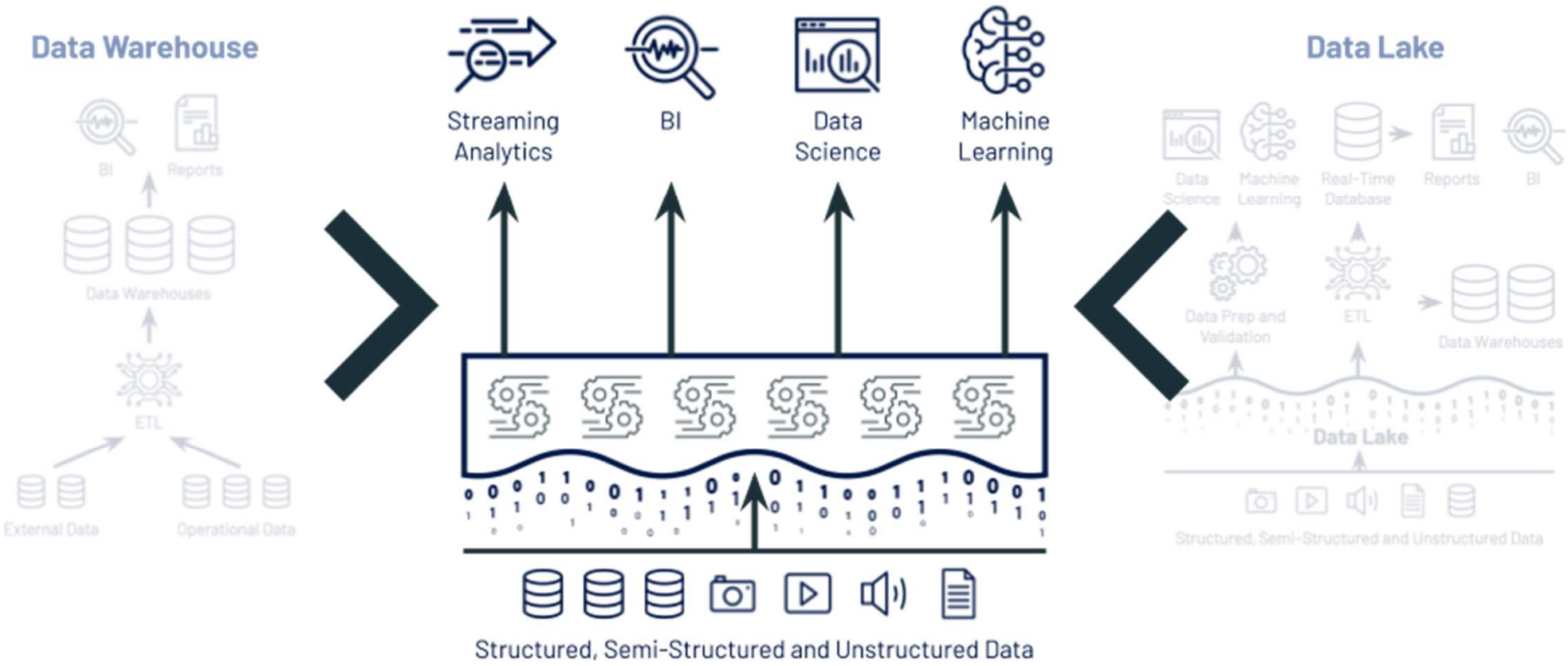
- Store all kinds of data
- Storage is very inexpensive
- Good starting point

However:

- Complex to set up
- Poor BI performance
- Unreliable data swamps



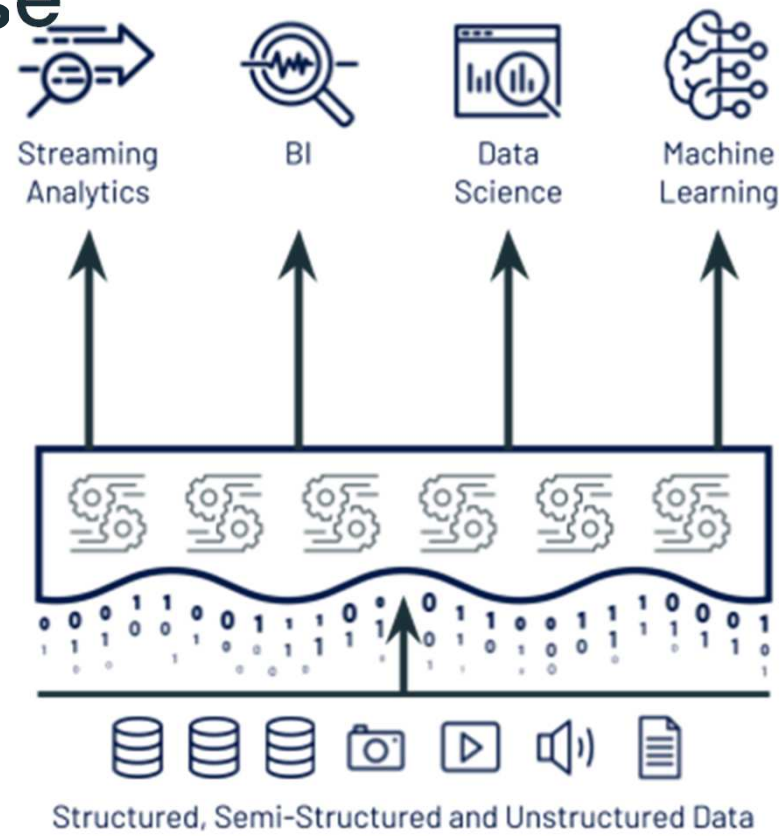
Introducing the Lakehouse



databricks

Key Features of a Lakehouse

- Support for diverse data types and formats
- The ability to use BI tools directly on source data
- Support for diverse workloads (BI, data science, machine learning, and analytics
- Data reliability and consistency



Practice Question 1 - Lakehouse

Which of the following describes a benefit of a data lakehouse that is unavailable in a traditional data warehouse?

A. A data lakehouse provides a relational system of data management.
B. A data lakehouse captures snapshots of data for version control purposes.
C. A data lakehouse couples storage and compute for complete control.
D. A data lakehouse utilizes proprietary storage formats for data.
E. A data lakehouse enables both batch and streaming analytics.

Answer

A. Wrong – A data lakehouse is designed for analytical workloads.
B. Wrong – Snapshots are created as a result of the Lakehouses optimistic concurrency control mechanism.
C. Wrong – Storage and compute are decoupled.
D. Wrong – Data lakehouse uses open-source Parquet storage format.

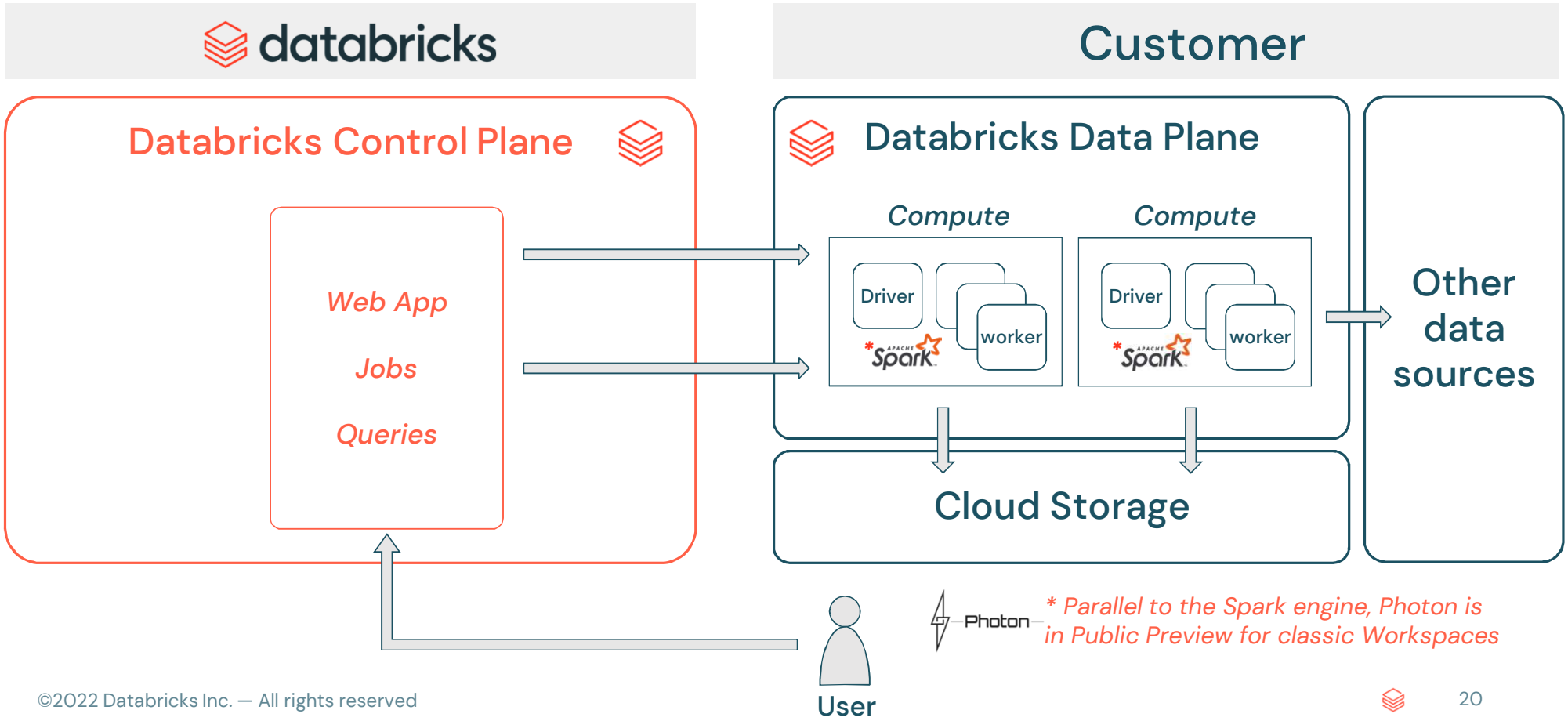
E. Correct – Data lakehouses support both batch and streaming workloads.

Learn More

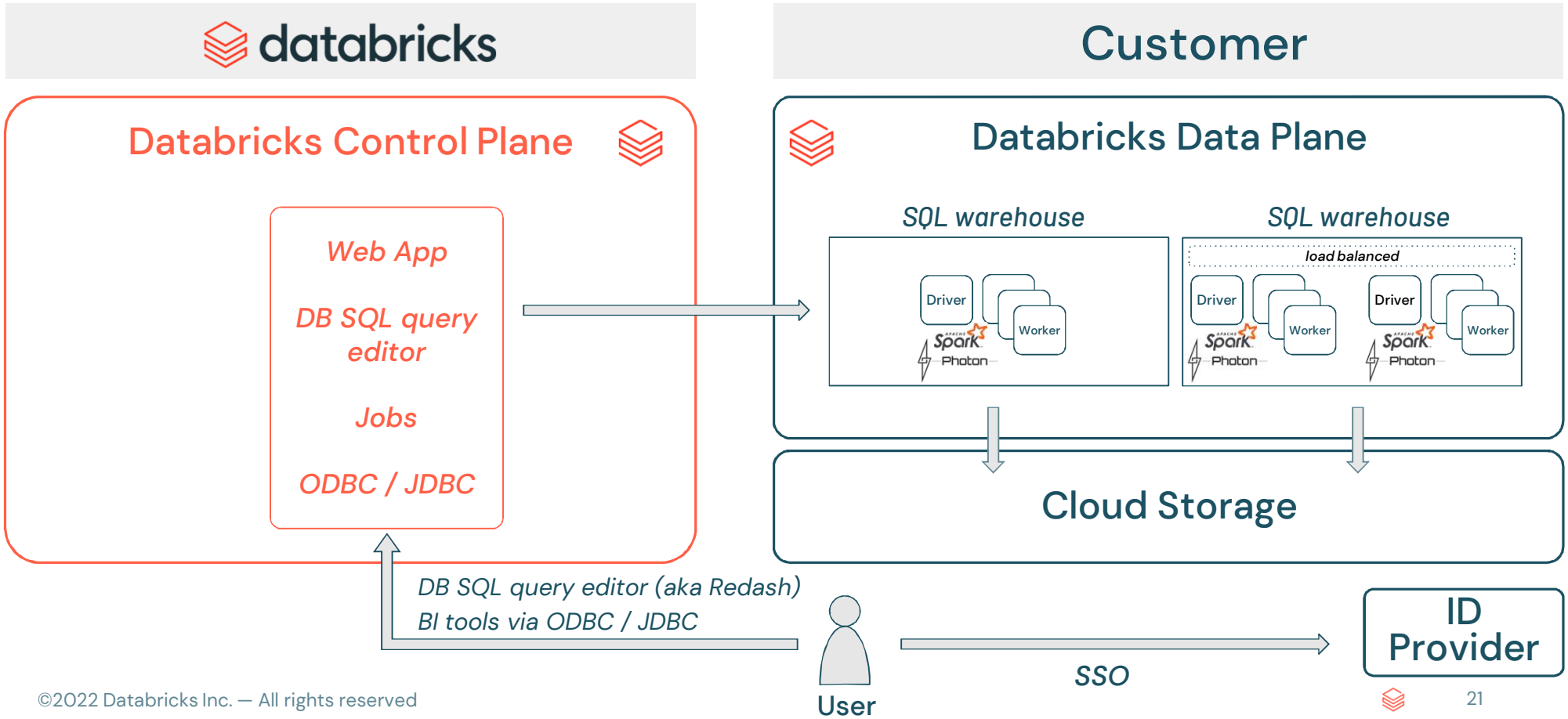
What is the Databricks Lakehouse Platform?

Platform Architecture

High Level Architecture - Databricks Workspaces



High Level Architecture – Databricks SQL



Practice Question 2 – Platform Architecture

Which of the following locations hosts the driver and worker nodes of a Databricks managed cluster?

A. Data plane
B. Control plane
C. Databricks Filesystem (DBFS)
D. JDBC data source
E. Databricks web application

Answer

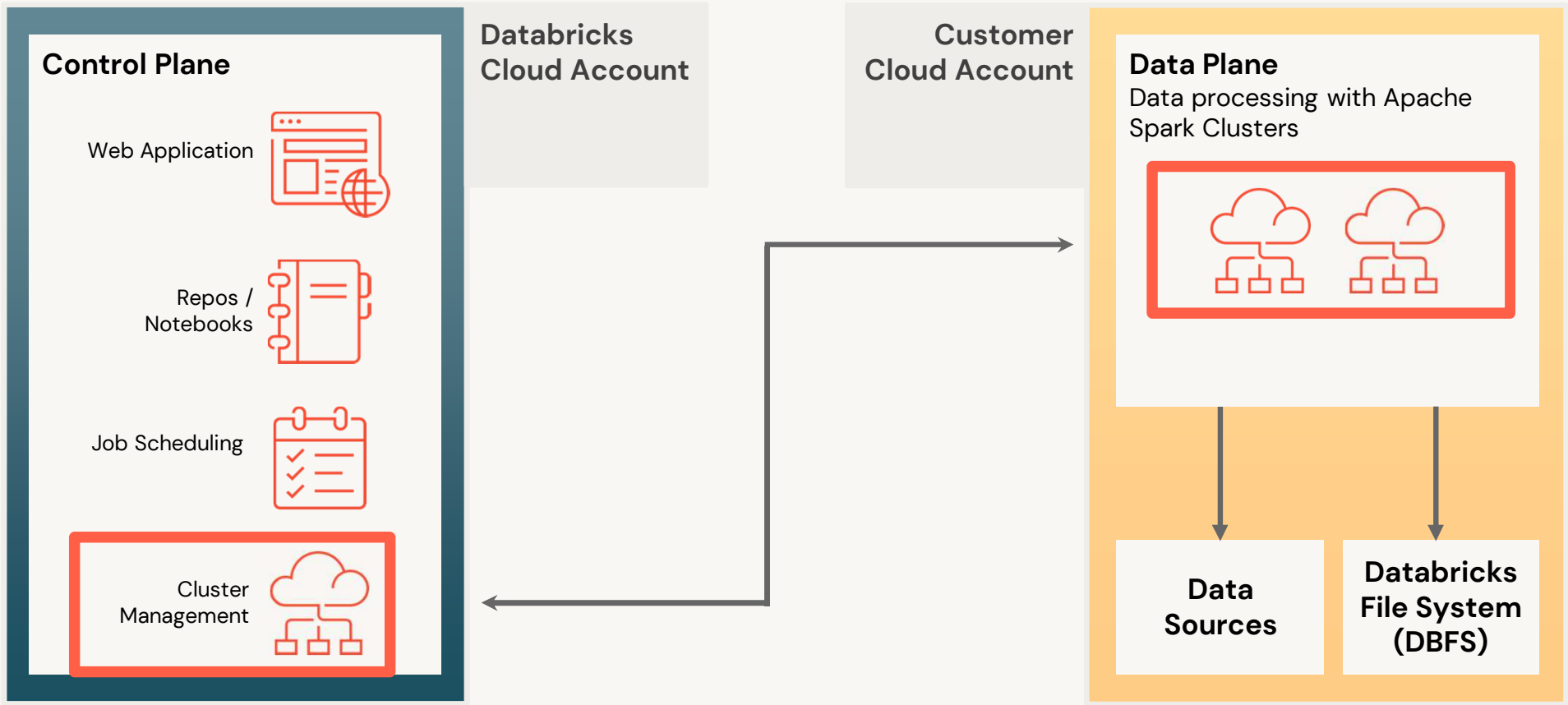
A. Correct – Workspace clusters are always deployed into the data plane in the customer's cloud account.
B. Wrong – The Databricks control plane is where Databricks services are run.
C. Wrong – DBFS is comprised of a services abstraction layer over cloud storage and a physical storage location in the customer's cloud account.
D. Wrong – Databricks' clusters are compute resources, not sources of data.
E. Wrong – The Databricks web application runs in the Databricks control plane.

Learn More

Databricks Architecture and Services

Clusters

Clusters



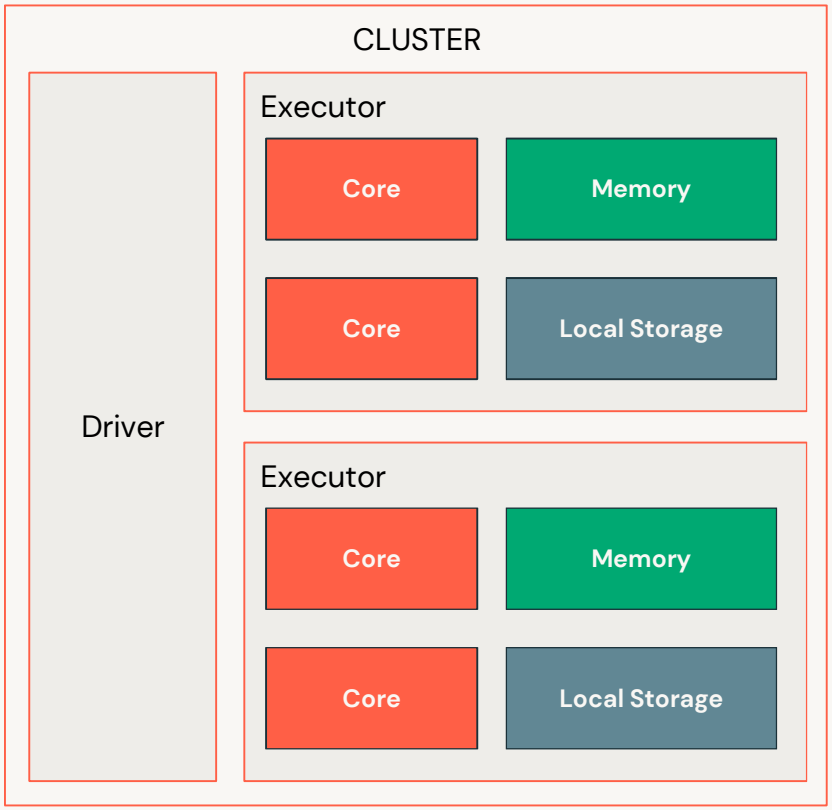
Clusters

Overview

Clusters are made up of one or more virtual machine (VM) instances

Driver coordinates activities of executors

Executors run tasks composing a Spark job



Clusters

Types

All-purpose Clusters

Analyze data collaboratively using interactive notebooks

Create clusters from the Workspace or API

Retains up to 70 clusters for up to 30 days.

Job Clusters

Run automated jobs

The Databricks job scheduler creates job clusters when running jobs.

Retains up to 30 clusters.

Practice Question 3 – Clusters

A data engineer has a job that needs to run on a regular schedule. Looking to save on costs which type of cluster should the data engineer consider?

A. SQL Warehouse Cluster
B. High Concurrency Cluster
C. Single Node Cluster
D. Jobs Cluster
E. Multi-node Scalable All-Purpose Cluster

Answer

A. Wrong – SQL Warehouses are only available in Databricks SQL, and is not present in the Data Engineering Workspace.
B. Wrong – High Concurrency clusters are designed to accommodate multiple users each running separate workloads.
C. Wrong – Single node cluster is still considered all-purpose and costs more to run.
D. Correct – Jobs clusters are designed to run unattended production workloads and are cheaper than all-purpose clusters.
E. Wrong – All-purpose clusters are much costlier than jobs clusters.

Learn More

Databricks Cluster Usage Management

Repos

Databricks Repos

Overview

Git Versioning

Native integration with Github, Gitlab, Bitbucket and Azure Devops

UI-based workflows

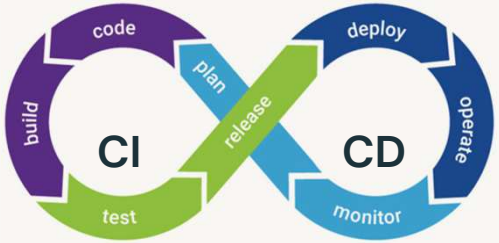
Azure DevOps GitHub

GitLab Bitbucket

CI/CD Integration

API surface to integrate with automation

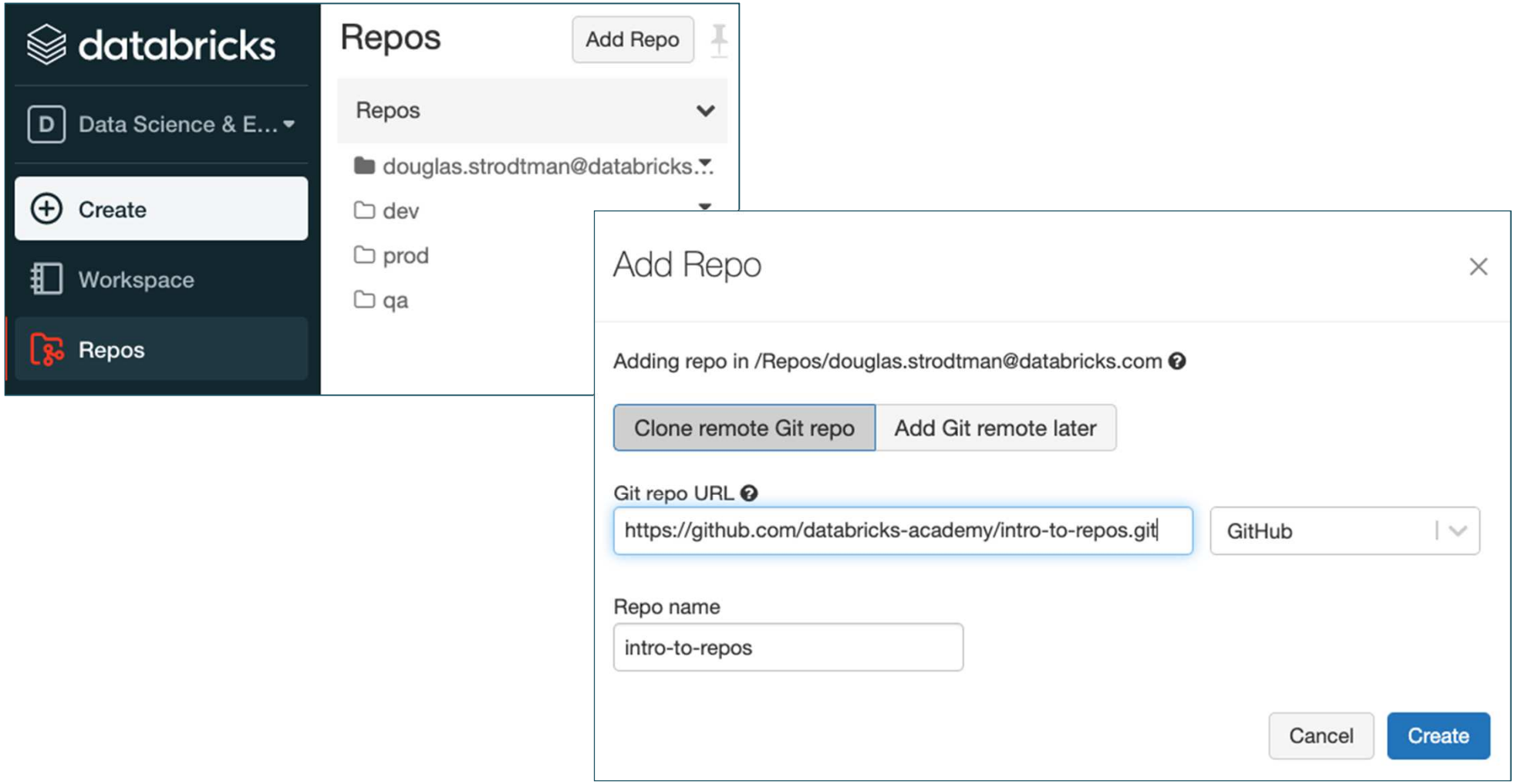
Simplifies the dev/staging/prod multi-workspace story

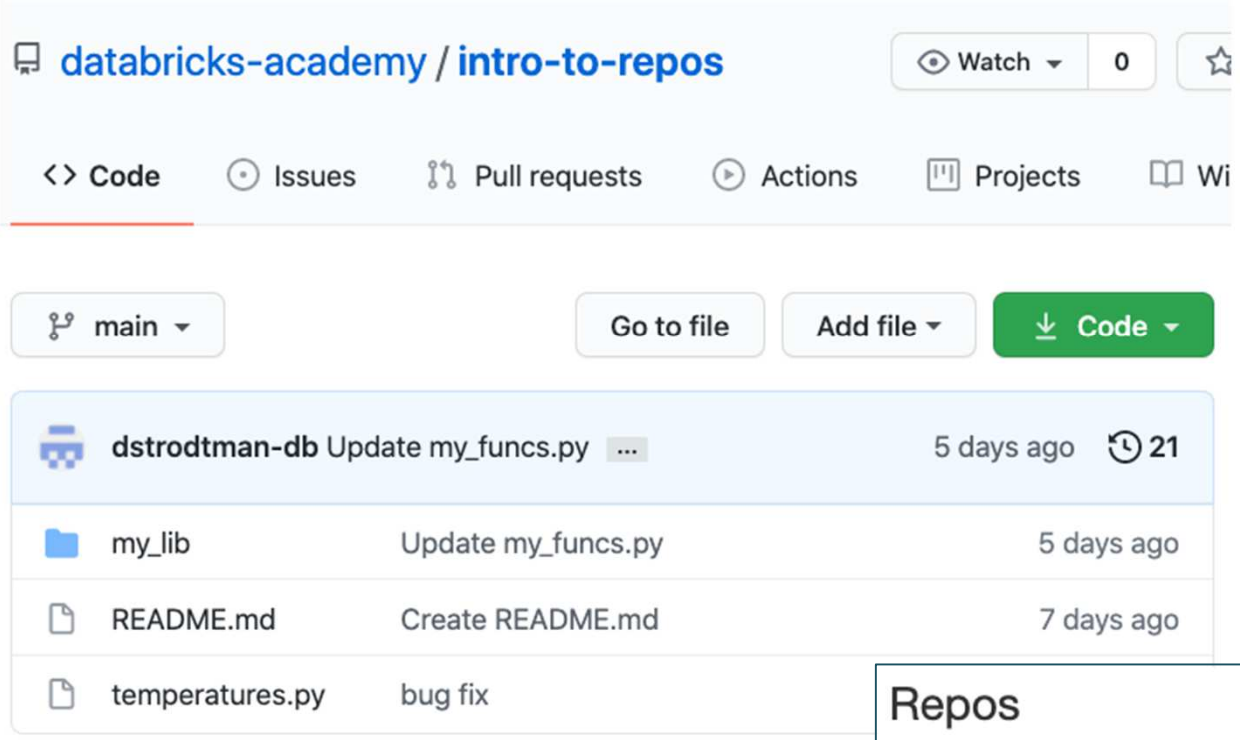


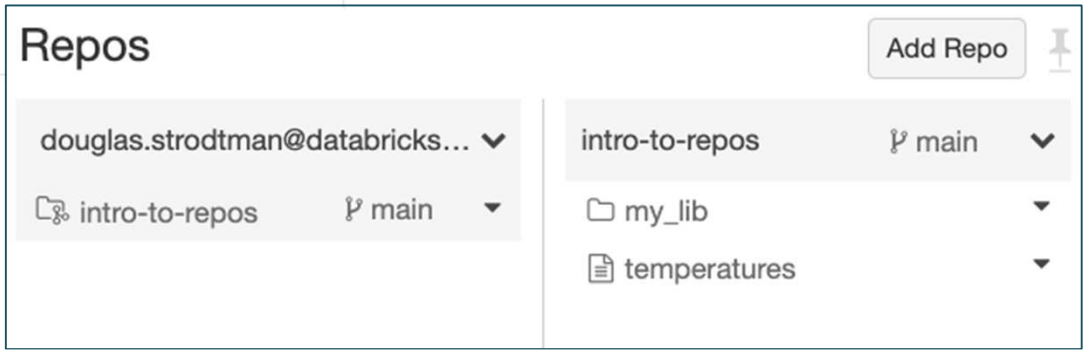
Enterprise ready

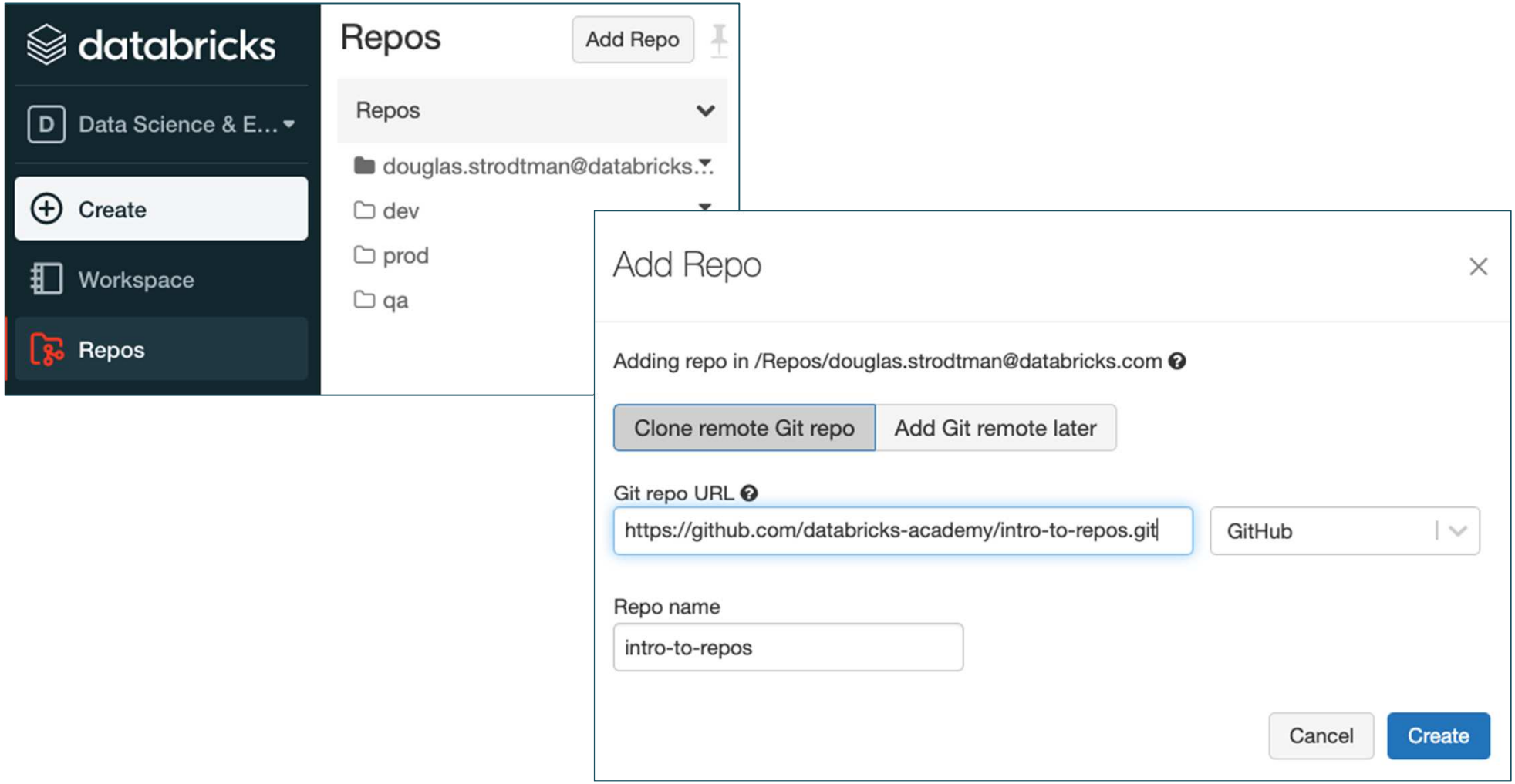
Allow lists to avoid exfiltration

Secret detection to avoid leaking keys


databricks
Data Science & E...
Create
Workspace
Repos
Repos
Add Repo
Repos
douglas.strodtman@databricks...
dev
prod
qa
Add Repo
Adding repo in /Repos/douglas.strodtman@databricks.com
Clone remote Git repo
Add Git remote later
Git repo URL
https://github.com/databricks-academy/intro-to-repos.git
GitHub
Repo name
intro-to-repos
Cancel
Create


databricks-academy / intro-to-repos
Watch
0
Code
Issues
Pull requests
Actions
Projects
Wi
main
Go to file
Add file
Code
dstrodtman-db Update my_funcs.py
5 days ago
21
my_lib
Update my_funcs.py
5 days ago
README.md
Create README.md
7 days ago
temperatures.py
bug fix


Repos
Add Repo
douglas.strodtman@databricks...
intro-to-repos
main
intro-to-repos
main
my_lib
temperatures


databricks
Data Science & E...
Create
Workspace
Repos
Repos
Add Repo
Repos
douglas.strodtman@databricks...
dev
prod
qa
Add Repo
Adding repo in /Repos/douglas.strodtman@databricks.com
Clone remote Git repo
Add Git remote later
Git repo URL
https://github.com/databricks-academy/intro-to-repos.git
GitHub
Repo name
intro-to-repos
Cancel
Create

Practice Question 4 – Repos

Which of the following describes how Databricks Repos can help facilitate CI/CD workflows on the Databricks Lakehouse Platform?

A. Repos facilitate the pull request, review, and approval process before merging branches.
B. Repos can merge changes from a secondary Git branch into a main Git branch.
C. Repos can be used to design, develop, and trigger Git automation pipelines.
D. Repos can store the single-source-of-truth Git repository.
E. Repos can commit or push code changes to trigger a CI/CD process.

Answer

A. Wrong – The pull, review, and approval process is related to Git best practices.
B. Wrong – Merging branches is independent of CI/CD pipelines.
C. Wrong – Pipeline development is facilitated through other tooling such as Azure Devops.
D. Wrong – Repos rely on the Git repository for the single source of truth.
E. Correct – With repos you can trigger a CI/CD pipeline.

Learn More

Introduction to Databricks Repos

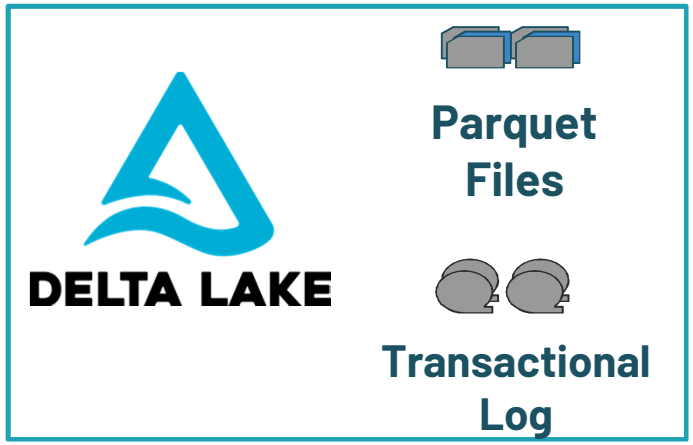
Delta Lake Concepts

Delta Lake ensures data reliability

Batch

Streaming

Updates/ Deletes

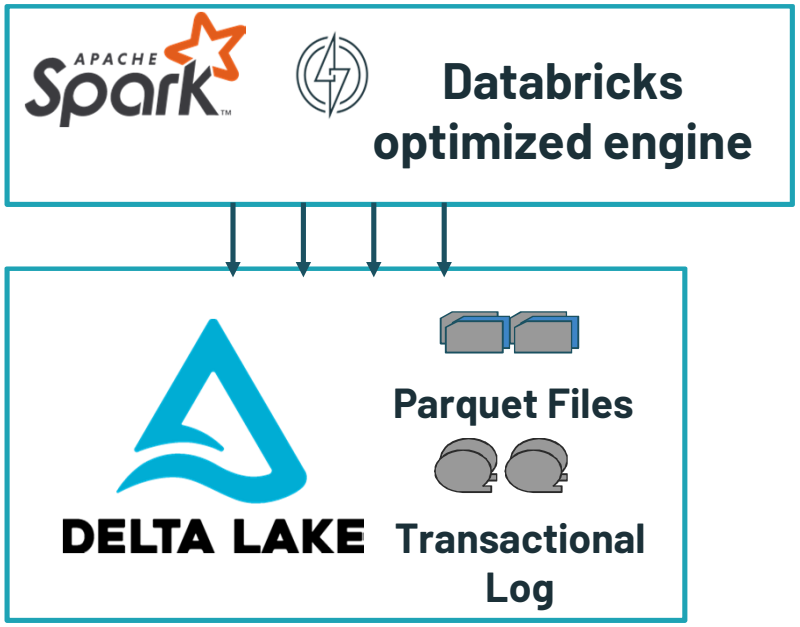


High Quality & Reliable Data
always ready for analytics

Key Features

- ACID Transactions
- Schema Enforcement
- Unified Batch & Streaming
- Time Travel/Data Snapshots

Delta Lake optimizes performance



Highly Performant queries at scale

Key Features

- Indexing
- Compaction
- Data skipping
- Caching

Delta Lake

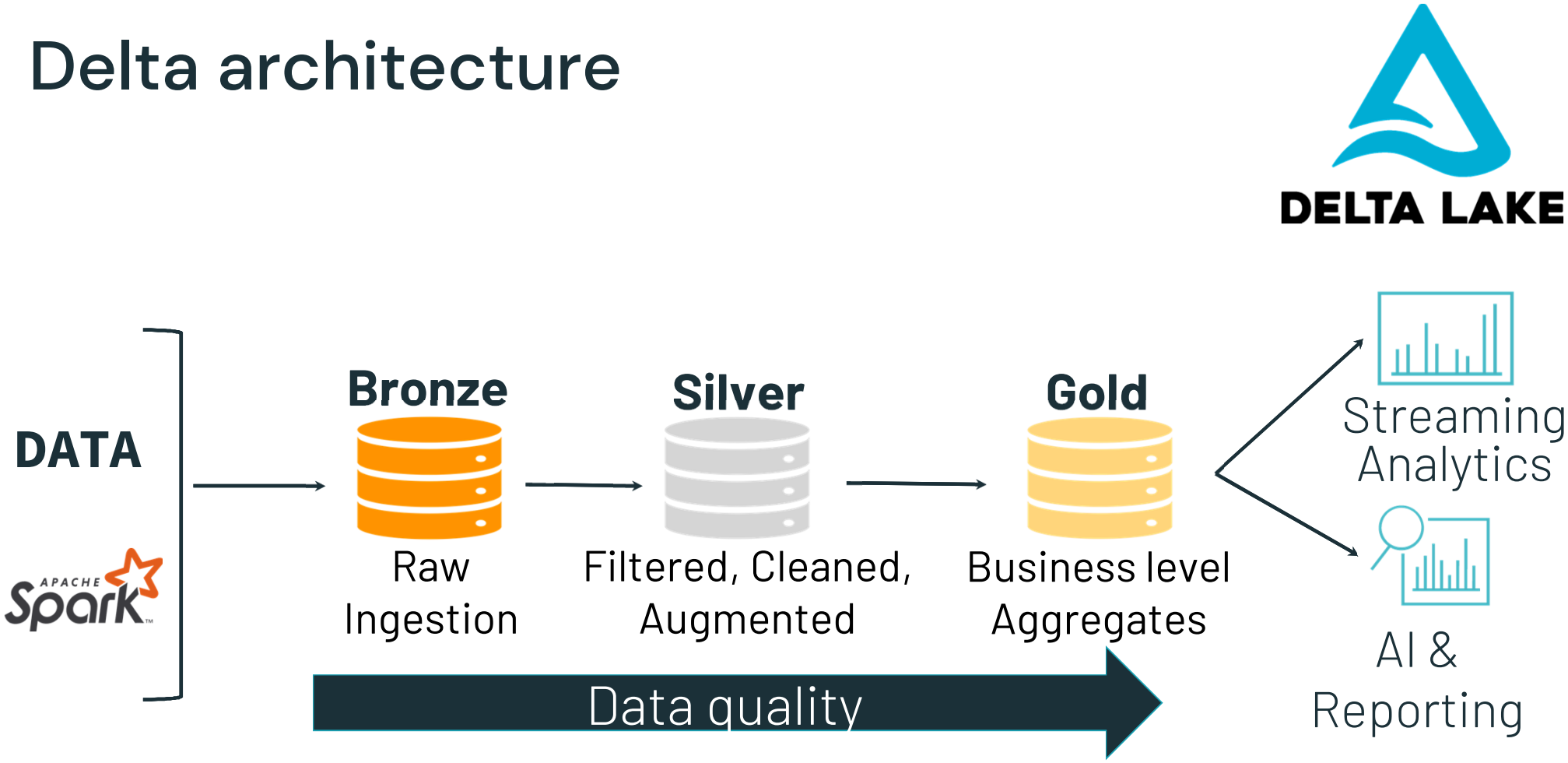
- Core component of a data lakehouse
- Offers guaranteed consistency because it's ACID compliant
- Robust data store
- Designed to work with Apache Spark and Photon

DELTA LAKE

Elements of Delta Lake

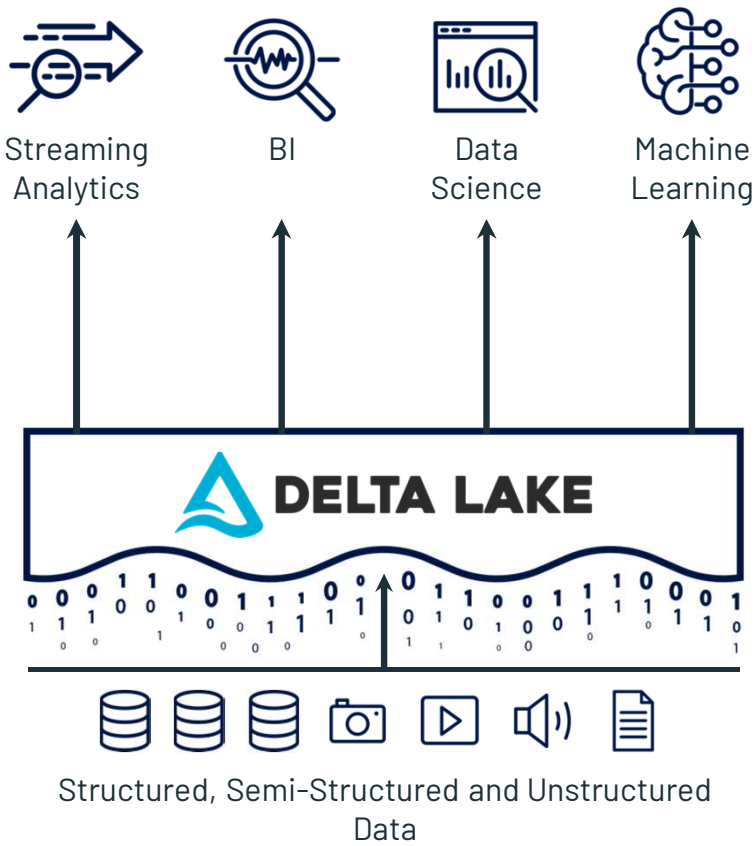
- Delta Architecture
- Delta Storage Layer
- Delta Engine

Delta architecture



databricks

Delta Storage Layer



One platform for every use case

Structured transactional layer

Data Lake for all your data

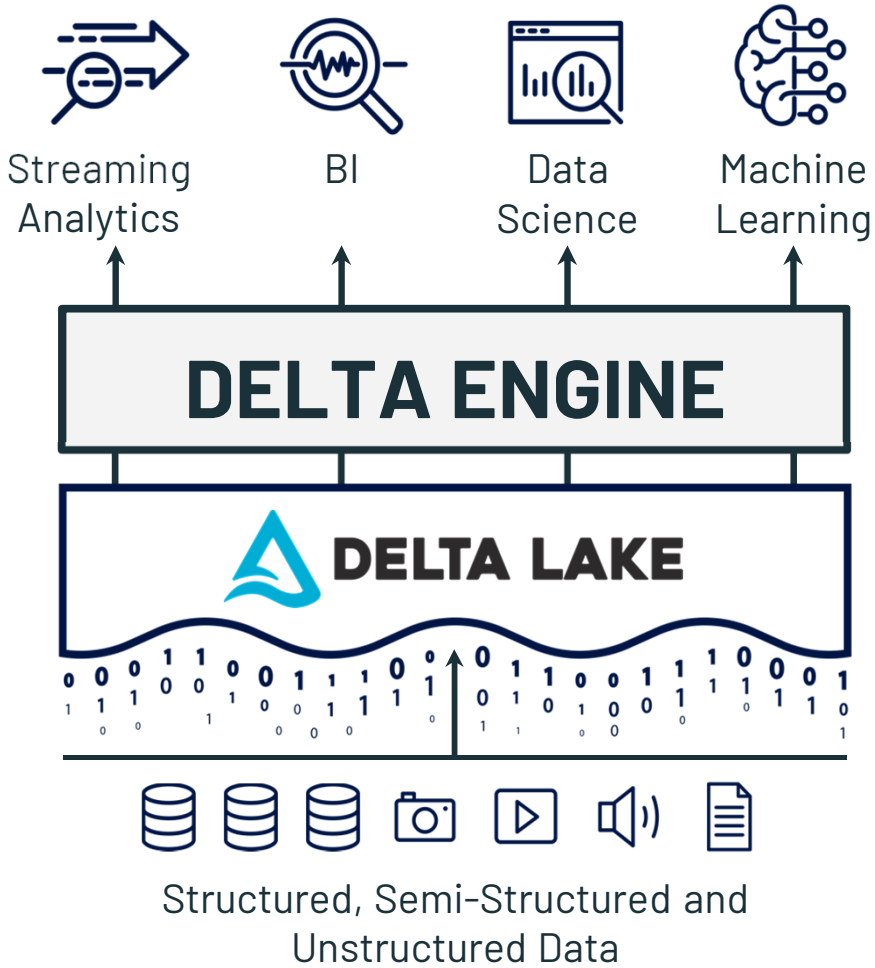
databricks

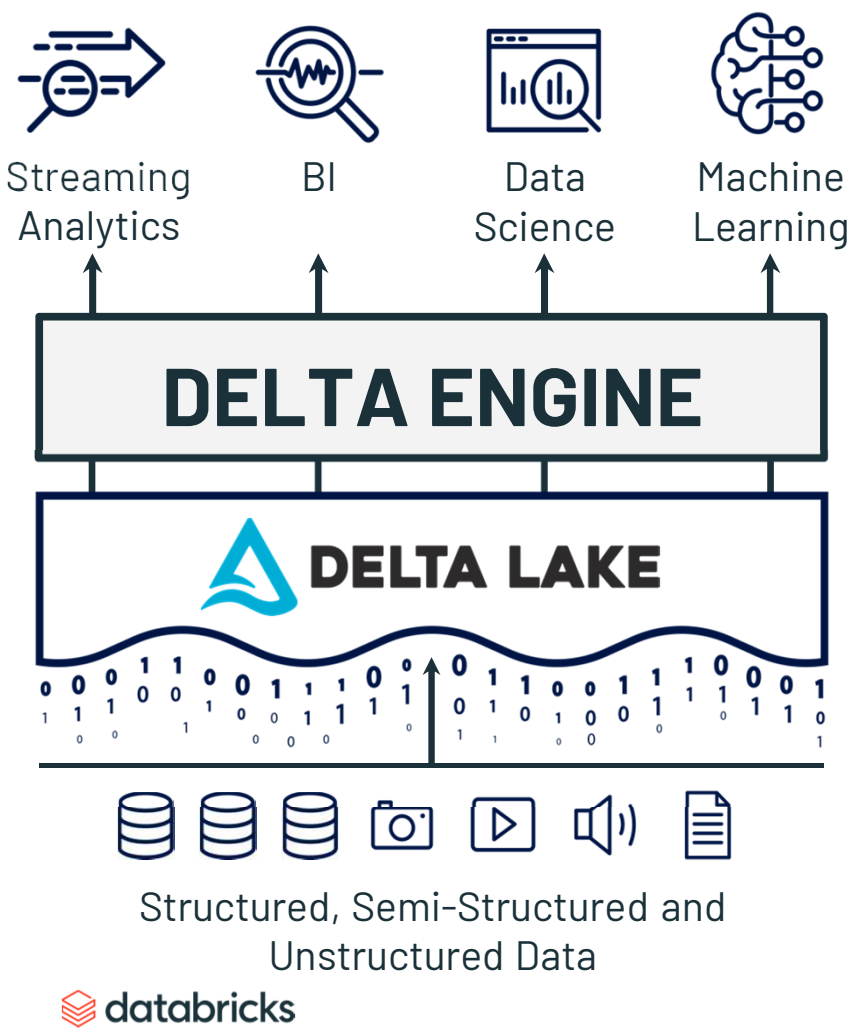
Delta Storage Layer

- Guarantee data is consistent
- Track metadata
- Automatically handle variations in schema
- Enables version control and rollbacks
- Merge and update data as it arrives

Delta Engine

- File management optimizations
- Performance optimization with Delta Caching
- Dynamic File Pruning
- Adaptive Query Execution





One platform for every use case

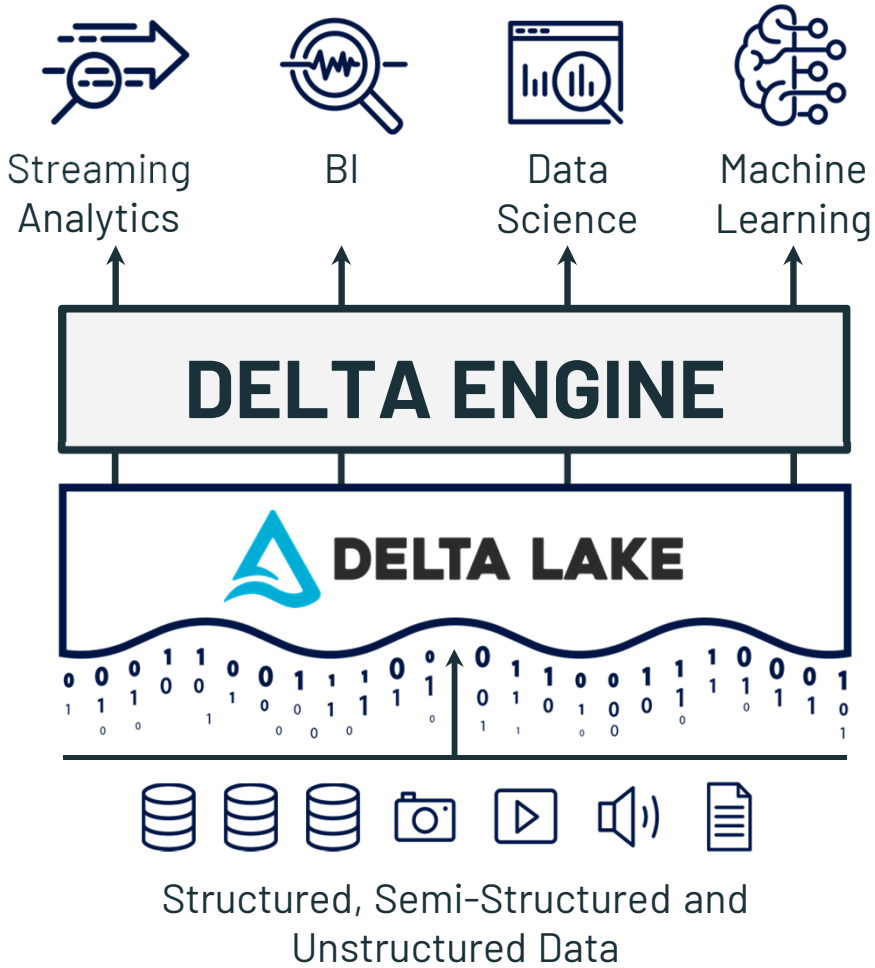
High performance query engine

Structured transactional layer

Data Lake for all your data

Delta Engine

- File management optimizations
- Performance optimization with Delta Caching
- Dynamic File Pruning
- Adaptive Query Execution



Practice Question 5 – Delta Lake

Which of the following describes Delta Lake?

A. Delta Lake is an open-source analytics engine used for big data workloads.
B. Delta Lake is an open format storage layer that delivers reliability, security, and performance.
C. Delta Lake is an open-source platform to help manage the complete machine learning lifecycle.
D. Delta Lake is an open-source data storage format for distributed data.
E. Delta Lake is an open format storage layer that processes data.

Answer

A. Wrong – Delta engine is just one part that makes up Delta Lake. It is more than just an engine.
B. Correct – Merging branches is independent of CI/CD pipelines.
C. Wrong – Delta can be used in machine learning; however, it was not designed for this single purpose.
D. Wrong – Delta lake uses open-source Parquet for its storage.
E. Correct – Delta engine works in conjunction with Apache Spark and Photon to process data. Delta engine by itself cannot process data.

Learn More

What is Delta Lake

Delta Tables

Get Started with Delta using Spark APIs

Instead of `parquet`...

```
CREATE TABLE ...
USING parquet
...

dataframe
        .write
        .format("parquet")
        .save("/data")
```

... simply say `delta`

```
CREATE TABLE ...
USING delta**
...

dataframe
        .write
        .format("delta")
        .save("/data")
```

** If using DBR 8.0 or greater, Delta is the default file format.

Get Started with Delta using SQL

```
CREATE TABLE Customers(
Id INT,
Fname STRING,
Lname STRING,
Address STRING,
State STRING,
Zipcode STRING
) USING DELTA **
```

```
UPDATE Customers
SET Fname = 'Frank' WHERE Id = 25;

DELETE FROM Customers WHERE Id = 4;

MERGE INTO Customers
USING CustomerUpdates
ON Customers.Id = CustomerUpdates.Id
WHEN MATCHED THEN UPDATE SET *
WHEN MATCHED THEN INSERT *
```

With Delta you can update, delete and perform upserts on Delta tables.

** If using DBR 8.0 or greater, Delta is the default file format.

Practice Question 6 – Delta Tables

Which of the following SQL keywords can be used to append new rows into an existing Delta table?

A. UPDATE
B. COPY
C. INSERT INTO
D. DELETE
E. UNION

Answer

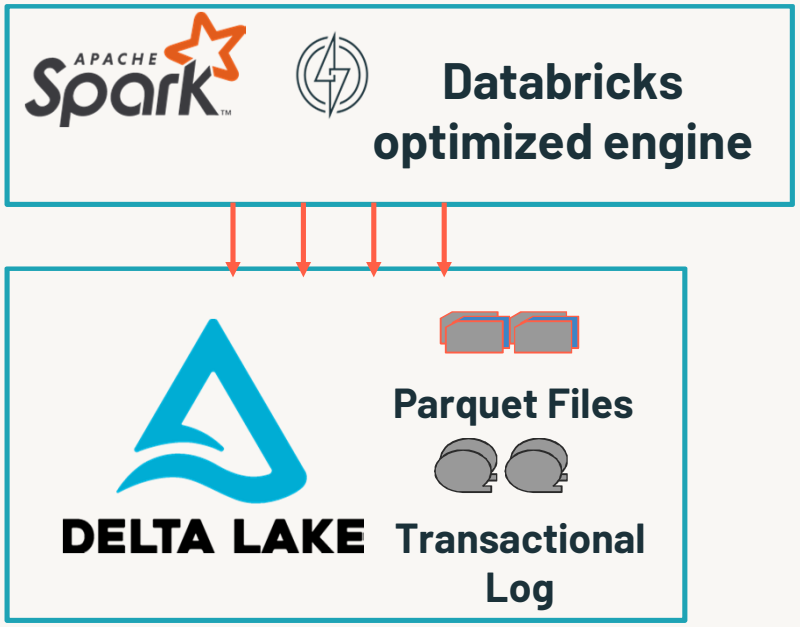
A. Wrong – UPDATE will change existing data.
B. Wrong – COPY by itself is not a valid Delta table command.
C. Correct – INSERT INTO inserts new rows into a Delta table.
D. Wrong – DELETE will remove rows.
E. Correct – UNION returns the result of subquery 1 plus the rows of subquery 2.

Learn More

Managing Delta Tables

Delta Optimizations

Delta Lake optimizes performance



Highly Performant queries at scale

Key Features

- Indexing
- Compaction
- Data skipping
- Caching

Optimizing On Delta

Databricks Delta uses multiple mechanisms to speed up queries

- **Compaction** coalescing small files into larger ones.files are compacted together into new larger files up to 1GB

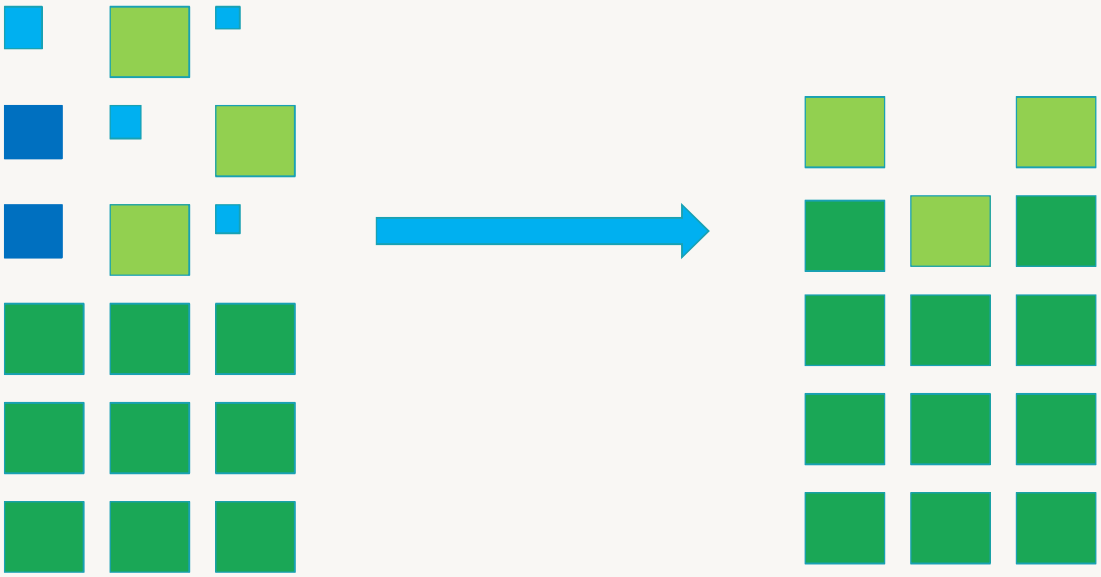
- **Partition Pruning** is a performance optimization that speeds up queries by limiting the amount of data read.

- **Data Skipping** is a performance optimization that aims at speeding up queries that contain filters (WHERE clauses).

- **ZOrdering** is a technique to colocate related information in the same set of files.
- **Caching**, automatically caches input Delta (and Parquet) tables, improving read throughput by 2X to 10X

OPTIMIZE: Compaction Built-in

OPTIMIZE my_table



Partition Pruning

/path/to/deltalake_table/

part=1/part_00001.parquet

part=1/part_00002.parquet

part=2/part_00001.parquet

part=2/part_00002.parquet

```
SELECT * FROM deltalake_table WHERE part = 2
```

What is data skipping?

- Simple, well-known I/O pruning technique used by many DBMSes and Big Data systems
- Idea: track file-level stats like min & max / leverage them to avoid scanning irrelevant files
- Example:

SELECT * FROM table WHERE col = 5

SELECT file_name FROM index
WHERE col_min <= 5 AND col_max >= 5

file_name	col_min	col_max
file1	6	8
file2	3	10
file3	1	4

ZORDER

```
OPTIMIZE my_table ZORDER BY (col1, col2)
```

How it Works:

- Takes existing parquet files within a partition.
- Maps the rows within the parquet files according to Column Specified .
  - In the case of only one column, the mapping above becomes a linear sort.
- Rewrites the sorted data into new parquet files.

Note: Databricks Runtime 7.x+

CACHING

- Caches the data accessed by the specified simple SELECT query in the disk cache.
- You can choose a subset of columns to be cached by providing a list of column names and choose a subset of rows by providing a predicate.
- This enables subsequent queries to avoid scanning the original files as much as possible.
- This construct is applicable only to Delta tables and Parquet tables.
- Views are also supported, but the expanded queries are restricted to the simple queries, as described above.

Practice Question 7 – Delta Optimizations

A data engineering team needs to query a Delta table to extract rows that all meet the same condition. However, the team has noticed that the query is running slowly. The team has already tuned the size of the data files. Upon investigating, the team has concluded that the rows meeting the condition are sparsely located throughout each of the data files.

Based on the scenario, which of the following optimization techniques could speed up the query?

A. Data Skipping
B. Z-Ordering
C. Bin-Packing
D. Write as a Parquet File
E. Tuning File Size

Answer

A. Wrong – Data Skipping is achieved through statistics gathered when Delta is writing out Parquet files.
B. Correct – Z-Ordering will reorganize the data to speed up queries.
C. Wrong – File compaction was already done on the data. Further compaction will have no effect.
D. Wrong – Moving from Delta format to open-source Parquet format will not improve performance, it will most likely make it worse.
E. Correct – The file compaction process done by the team tuned the file size already.

ETL with SPARK SQL (29%)

ETL WITH SPARK SQL (29%)

The minimally qualified candidate should be able to:

- Querying Files Directly
- Delta Tables
- Writing to tables

ETL with Spark SQL

Querying Files Directly
Options for External Sources
Creating Tables
Writing to Tables
Cleaning Data
Advanced Transformations
SQL UDF's

Practice Question 1 - Querying Files Directly

True or False

Is it Possible to write a SQL query directly against a file or a directory of files in Databricks?

Answer

TRUE

Using the following syntax you can run a query directly against a file or a directory of files.

SELECT * FROM file_format./path/to/file

The file path can be a single file or a directory

File Format examples would be json,csv,parquet etc

Practice Question 2 – Querying Files Directly

Which of the following statements would read from a json file and filter for records where the country = “SWE”

A. SELECT * FROM json.`${wasbs://some_account/some_container/countrydata.json` where country = “SWE”
B. Create table as select * from (wasbs://some_account/some_container/countrydata.json).filter(“country=‘SWE’)
C. From (create table using json, location = wasbs://some_account/some_container/countrydata.json) as table1, select * where country =‘swe’

Correct answer = A

Answer

Correct Answer A

Discussion:

The correct syntax for this query is Answer A

Practice Question 2

When reading directly from a file in SQL how does spark determine the schema.

A. The schema must be supplied
B. The schema is inferred
C. The default schema of _c0 String, _c1 String,_c2 String…. is always used

Answer

Correct answer B.

When reading from a file if the file is csv the header of the file will be used for column names.
If the file is JSON the JSON will be parsed to determine the schema.
When reading from parquet, the parquet header will provide the schema

Practice Question 3

You are tasked with reading user data that has a history of having a small but significant percentage of dates formatted incorrectly that when parsed end up being in the future. What strategy might you employ to avoid reading those records.

1. Add a check constraint to the table Add constraint not_in_future(date <= current_date())
2. Select * from source where date <= current_date()
3. Use a Foreign Key constraint
4. Quarantine the source table

Answer

Correct answers 1,2

Discussion:
Check constraints can be added to delta tables to enforce rules that can be expressed as a sql statement.

A filter as describe in answer 2 would also work.

Delta does not at this time enforce Foreign Keys, besides it is hard to imagine how they would prevent date format issues,

Quarantining the source table would prevent any of the records from being read, instead of just the incorrectly formatted dates.

Practice Question 4

Comments can be added as informational fields to which of the following

A. Databases(also known as schemas)
B. Tables
C. Columns
D. All of the above

Answer

Discussion,

Correct answer = all of the above, comments can be added to Tables, Columns and Databases(schemas)

Practice Question 5

Cloning Delta Tables

Which of the following statements is correct:
Definitions used in this question
“source” = table to be cloned
“clone” = a table created using a create table table_name Deep|shallow clone source

A. Modifying the clone may conflict with writes in progress on the source.
B. Time travel on the clone is available to versions of the source created before the clone was created
C. Delta tables with constraints can not be cloned
D. Modification of the clone will never lead to data change on the source

Answer

Answer = D

No operations on the clone will effect the source. It will not conflict with writes on the source, constraints on the source will exist on the clone.

Time travel however on the clone is limited, to either the version that existed from the time of the clone, and any future changes to the clone including an incremental application of deep clone which only copies new data over to the clone from the source

Practice Question 6

You have two tables, one is a delta table named conveniently enough as “delta_table” and the other is a parquet table named once again quite descriptively as parquet_table. Some error in ETL upstream has led to source_table having zero records, when it is supposed to have new records generated daily.

If I run the following statements.
Insert overwrite delta_table select * from source_table;
Insert overwrite parquet_table select * from source table;

Which statement below is correct.

A. Both tables can be restored using “Restore table table_name version as of <previous version>
B. Both tables, delta_table and parquet_table have been completed deleted, with no options to restore
C. The current version of the delta table is a full replacement of the previous version, but it can be recovered through time travel or a restore statement
D. If the table is an external table the data is recoverable for the parquet table

Answer

Answer C.

The delta table can be recovered. The parquet table could only be recovered if it was stored in a location that was backed up in some way. Wether or not the table is external or managed makes no difference in this case.

Incremental Data Processing (22%)

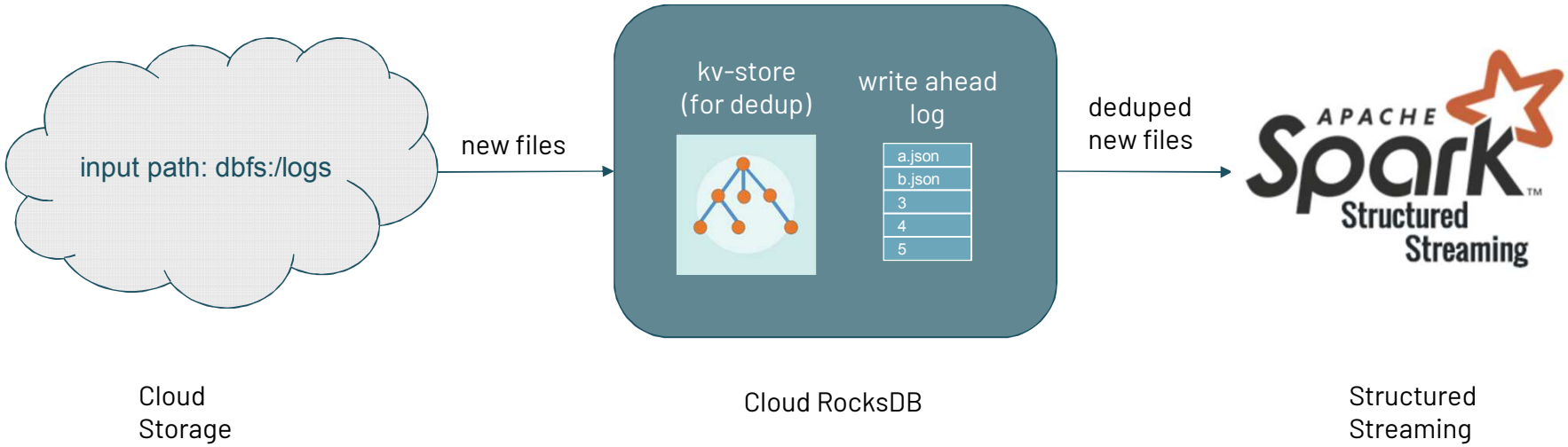
Incremental Data Processing (22%)

The minimally qualified candidate should be able to:

Incrementally process data, including:

- Structured Streaming (general concepts, triggers)
- Auto Loader (streaming reads)
- Multi-hop Architecture (bronze-silver-gold, streaming applications)
- Delta Live Tables (benefits and features)

Overview


input path: dbfs:/logs
new files
kv-store
(for dedup)
write ahead
log
a.json
b.json
3
4
5
deduped
new files
APACHE
Spark
Structured
Streaming
Cloud
Storage
Cloud RocksDB
Structured
Streaming

Auto Loader

Streaming Reads

Incrementally process data to power analytic insights with Spark Structured Streaming and AutoLoader

- Define streaming reads with Auto Loader and Pyspark to load data into Delta
- Define streaming reads on tables for SQL manipulation
- Identifying source locations
- Use cases for using Auto Loader

Getting started with Auto Loader

Load files from Cloud Storage , in Python or Scala

```
df = spark.readStream
            .format("cloudFiles")
            .option("cloudFiles.format", "json")
            .load("/input/path")
            .writeStream
            .option("checkpointLocation","/chk/path")
            .start("/out/path")
```

Also Available in DLT

Practice Questions 1 - Auto Loader

A data engineer has developed a code block to perform a streaming read on a data source. The code is below:

```
(spark
          .read
          .schema(schema)
          .format("CloudFiles")
          .option("cloudFiles.format", "json")
          .load(dataSource)
)
```

The code is returning an error.

Practice Questions 1 - Auto Loader

Which of the following changes should be made to the code block to configure it to successfully perform a streaming read?

A. The .**read** line should be replaced with .**readStream**.
B. A new **.stream** line should be added after the **.read** line.
C. The .**format**("**cloudFiles**") line should be replaced with .**format**("**stream**").
D. A new .**stream** line should be added after the spark line.
E. A new .**stream** line should be added after the .**load**(**dataSource**) line.

Structured Streaming

General Concepts

Programming model

Configuration for reads and writes

End-to-end fault tolerance

Interacting with streaming queries

Triggers

Set up streaming writes with different **.trigger()** behaviors

- Default
- ProcessingTime = "2 minutes"
- Once = True
- AvailabilityNow = True

Output Mode

Complete

Append

Practice Questions 2 - Streaming

A data engineer has configured a Structured Streaming job to read from a table, manipulate the data, and then perform a stream write into a new table.

The code block used as below:

```
(spark.table("sales")
        .withColumn("avg_price", col("sales") / col("units"))
    .writeStream
    .option("checkpointLocation", checkpointPath)
    .ouputMode("complete")
    .____
    .table("new_sales")
```

Practice Questions 2 - Streaming

If the data engineer only wants the query to execute a single micro-batch to process all of the available data, which of the following lines of code should the data engineer use to fill in the blank?

A. trigger (once=True)
B. trigger(continous="once")
C. processingTime("once")
D. trigger(processingTime="once")
E. processingTime(1)

Multi-hop Architecture

Propagate new data through multiple tables in the data lakehouse

Bronze

Bronze vs raw tables

Workloads using bronze tables as source

Silver and Gold

Silver vs gold tables

Workloads using silver table as source

Structured Streaming in Multi-hop

Converting data from bronze to silver levels with validation

Converting data from silver to gold levels with aggregation

Practice Questions 3 - Multi-Hop Architecture

Which of the following data workloads will utilize a Bronze table as its source?

A. A job that aggregates cleaned data to create standard summary statistics
B. A job that queries aggregated data to publish key insights into a dashboard
C. A job that ingests raw data from a streaming source into the lakehouse
D. A job that develops a feature set for a machine learning application
E. A job that enriches data by parsing its timestamps into a human-readable format

Practice Questions 4 - Multi-Hop Architecture

Which of the following Structured Streaming queries is performing a hop from a Bronze table to a Silver table?

A.
```
(spark.table("sales")
        .groupBy("store")
        .agg(sum("sales"))
        .writeStream
        .option("checkpointLocation", checkpointPath)
        .outputMode("complete")
        .table("aggregatedSales")
)
```

Practice Questions 4 - Multi-Hop Architecture

B.
```
(spark.table("sales")
        .agg(sum("sales")
             sum("units"))
        .writeStream
        .option("checkpointLocation", checkpointPath)
        .outputMode("complete")
        .table("aggregatedSales")
)
```

C.
```
(spark.table("sales")
        .withColumn("avgPrice", col("sales") / col("units"))
        .writeStream
        .option("checkpointLocation", checkpointPath)
        .outputMode("append")
        .table("cleanedSales")
```

Practice Questions 4 - Multi-Hop Architecture

D. (spark.readStream.load(rawSalesLocation)
.writeStream
.option("checkpointLocation", checkpointPath)
.outputMode("append")
.table("uncleanedSales")
)

E. (spark.read.load(rawSalesLocation)
.writeStream
.option("checkpointLocation", checkpointPath)
.outputMode("append")
.table("uncleanedSales")
)

Delta Live Tables

Leverage Delta Live Tables to simplify productionalizing SQL data pipelines with Databricks

General Concepts

Benefits of using Delta Live Tables for ETL

Scenarios that benefit from Delta Live Tables

UI

Deploying DLT pipelines from notebooks

Executing updates

Explore and evaluate results from DLT pipelines

SQL Syntax

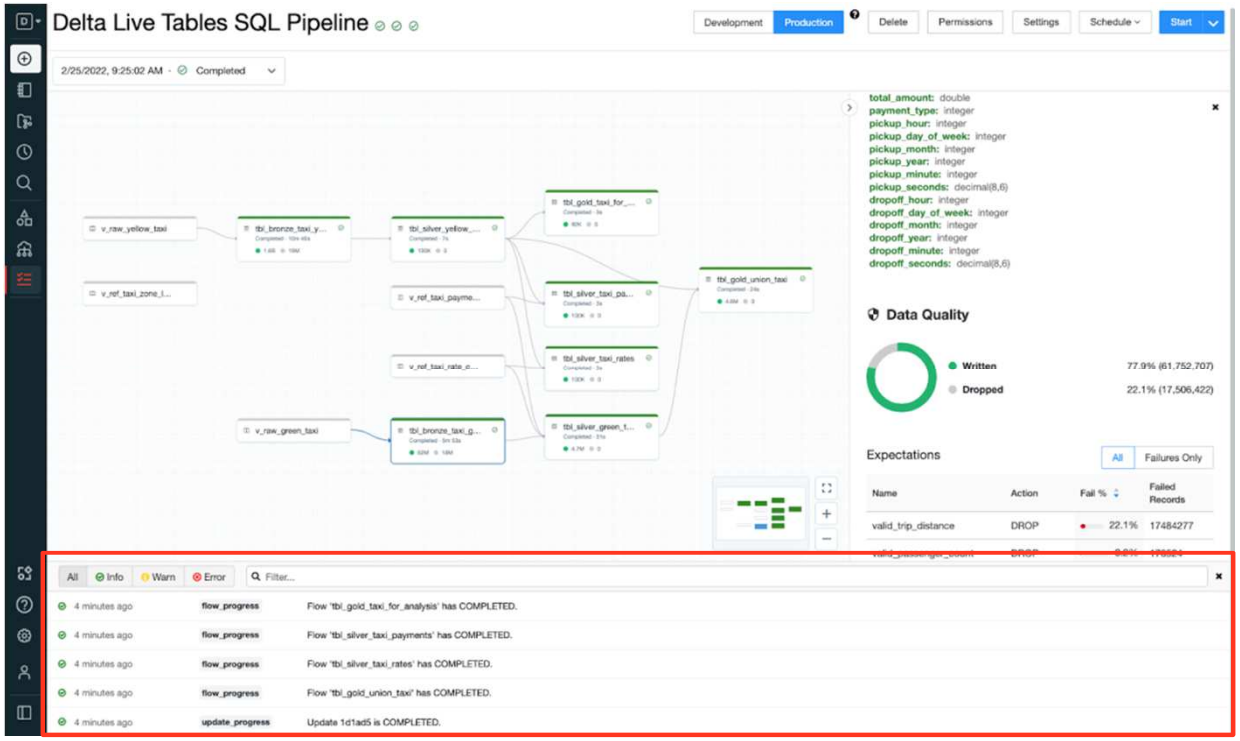
Converting SQL definitions to Auto Loader syntax

Common differences in DLT SQL syntax

Pipelines UI

A one stop shop for ETL debugging and operations

- Visualize data flows between tables
- Discover metadata and quality of each table
- Access to historical updates
- Control operations
- Dive deep into events



What is a Live Table?

Live Tables are materialized views for the lakehouse.

A live table is:

- Defined by a SQL query
- Created and kept up-to-date by a pipeline

```
CREATE ~~OR REPLACE~~ LIVE TABLE report
AS SELECT sum(profit)
FROM prod.sales
```

Live tables provides tools to:

- Manage dependencies
- Control quality
- Automate operations
- Simplify collaboration
- Save costs
- Reduce latency

Declarative SQL & Python APIs

Source

```
/* Create a temp view on the accounts table */
CREATE STREAMING LIVE VIEW account_raw AS
SELECT * FROM cloud_files(“/data”, “csv”);
```

Bronze

```
/* Stage 1: Bronze Table drop invalid rows */
CREATE STREAMING LIVE TABLE account_bronze AS
COMMENT "Bronze table with valid account ids"
SELECT * FROM account_raw ...
```

Silver

```
/* Stage 2:Send rows to Silver, run validation rules */
CREATE STREAMING LIVE TABLE account_silver AS
COMMENT "Silver Accounts table with validation checks"
SELECT * FROM account_bronze ...
```

Gold

- Use intent-driven declarative development to abstract away the **“how”** and define **“what”** to solve
- Automatically generate **lineage** based on table dependencies across the data pipeline
- Automatically checks for errors, missing dependencies and syntax errors

What is a Streaming Live Table?

Based on Spark™ Structured Streaming

A streaming live table is "stateful":

- Ensures exactly-once processing of input rows
- Inputs are only read once

- Streaming Live tables compute results over append-only streams such as Kafka, Kinesis, or Auto Loader (files on cloud storage)
- Streaming live tables allow you to reduce costs and latency by avoiding reprocessing of old data.

```
CREATE STREAMING LIVE TABLE report
AS SELECT sum(profit)
FROM cloud_files(prod.sales)
```

BEST PRACTICE

Development vs Production

Fast iteration or enterprise grade reliability

Development Mode

- Reuses a **long-running cluster** running for fast iteration.
- **No retries** on errors enabling faster debugging.

Production Mode

- Cuts costs by **turning off clusters** as soon as they are done (within 5 minutes)
- **Escalating retries**, including cluster restarts, ensure reliability in the face of transient issues.

In the Pipelines UI: Development | Production

Practice Questions 5 - DLT

A data engineer has three notebooks in an ELT pipeline. The notebooks need to be executed in a specific order for the pipeline to complete successfully. The data engineer would like to use Delta Live Tables to manage this process.

Which of the following steps must the data engineer take as part of implementing this pipeline using Delta Live Tables?

A. They need to create a Delta Live Tables pipeline from the Data page.
B. They need to create a Delta Live Tables pipeline from the Jobs page.
C. They need to refactor their notebook to use Python and the dlt library.
D. They need to refactor their notebook to use SQL and **CREATE LIVE TABLE** keyword.

Practice Questions 6 - DLT

A Delta Live Table pipeline includes two datasets defined using STREAMING LIVE TABLE. Three datasets are defined against Delta Lake table sources using LIVE TABLE.

The table is configured to run in Development mode using the Triggered Pipeline Mode.

Assuming previously unprocessed data exists and all definitions are valid, what is the executed outcome after clicking Start to update the pipeline?

Practice Questions 6 - DLT

A. All datasets will be updated once and the pipeline will shut down. The compute resources will be terminated.

B. All datasets will be updated at set intervals until the pipeline is shut down. The compute resources will be deployed for the update and terminated when the pipeline is stopped.

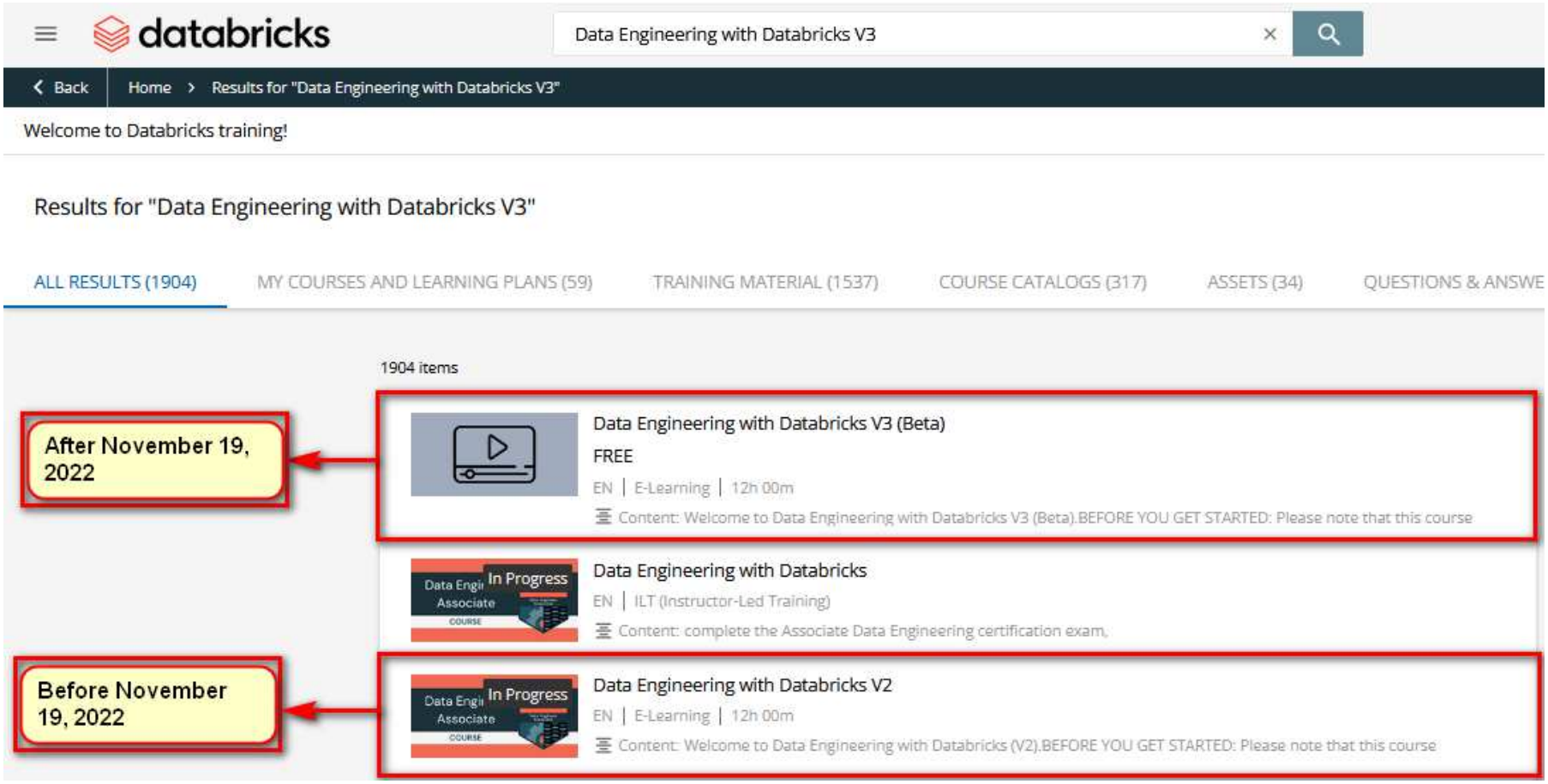
C. All datasets will be updated at set intervals until the pipeline is shut down. The compute resources will persist after the pipeline is stopped to allow for additional testing.

D. All datasets will be updated once and the pipeline will shut down. The compute resources will persist to allow for additional testing.

E. All datasets will be updated continuously and the pipeline will not shut down. The compute resource will persist with the pipeline.

Access On-Demand DEwD from Partner Academy

https://partner-academy.databricks.com/



Additional Study Resource

- Data Engineering with Databricks - Summary Notes
- Code repo
- Certification preparation workshop - ON-demand

Production Pipelines 16%

The minimally qualified candidate should be able to:

Build production pipelines for data engineering applications and Databricks SQL queries and dashboards, including:

- Jobs
  - Automation
  - Task Orchestration
  - UI
- Dashboards
  - SQL Endpoints
  - Query Scheduling
  - Alerting
  - Refreshing

Jobs

Orchestrate tasks with Databricks Jobs

Automation

Setting up retry policies

Using cluster pools and why

Task Orchestration

Benefits of using multiple tasks in Job

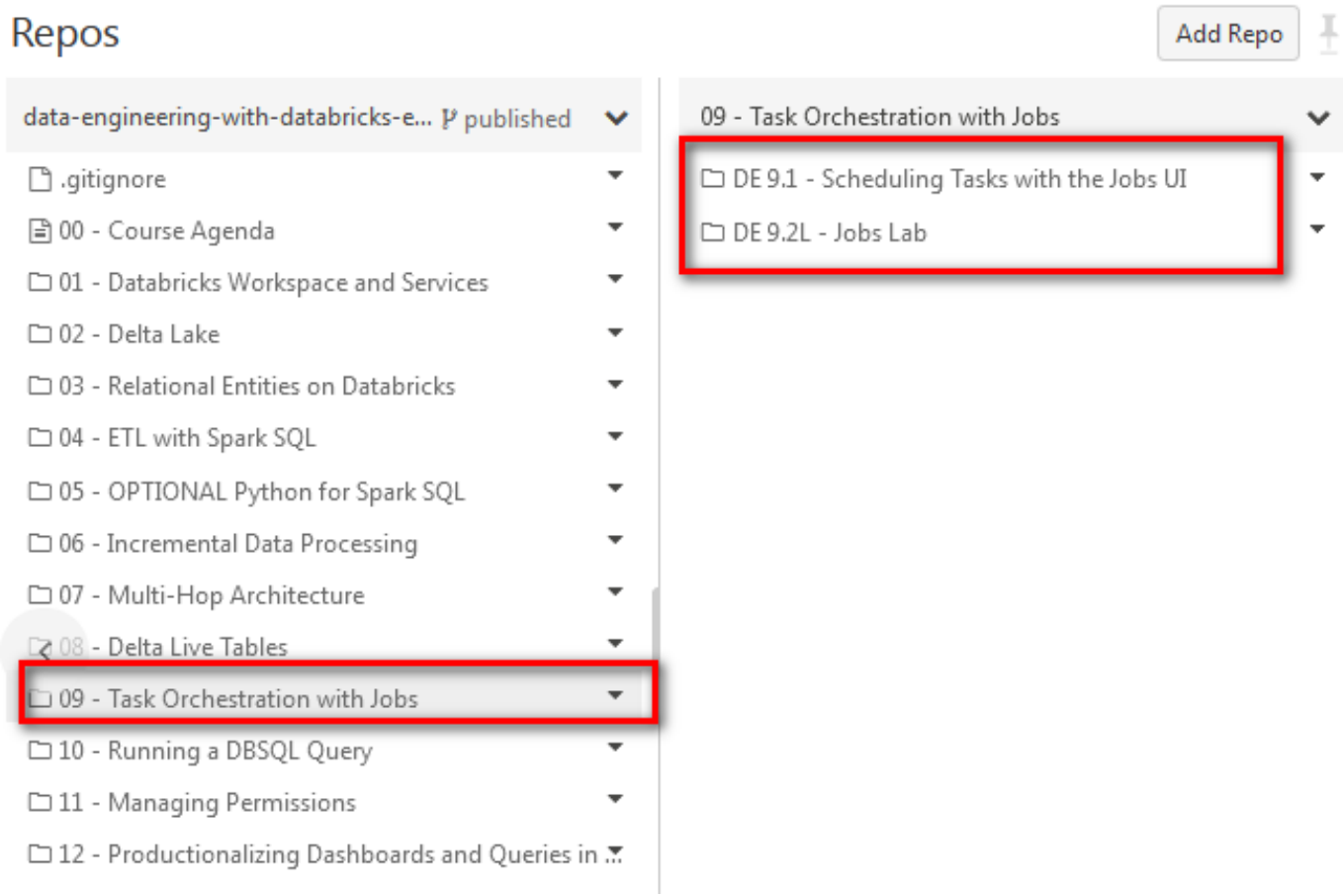
Configuring predecessor tasks

UI

Using notebook parameters in jobs

Locating job failures using Jobs UI

Ensure that you know this section completely



Focus on these links from docs.databricks page

1 https://docs.databricks.com/workflows/index.html

2 https://docs.databricks.com/workflows/jobs/jobs.html

3 https://www.databricks.com/blog/2021/07/13/announcement-orchestrating-multiple-tasks-with-databricks-jobs-public-preview.html

4 https://learn.microsoft.com/en-us/azure/databricks/workflows/jobs/jobs

Practice Question 1

An engineering manager uses a Databricks SQL query to monitor their team's progress on fixes related to customer-reported bugs. The manager checks the results of the query every day, but they are manually rerunning the query each day and waiting for the results.

Which of the following approaches can the manager use to ensure the results of the query are updated each day?

A. They can schedule the query to run every 1 day from the Jobs UI.
B. They can schedule the query to refresh every 1 day from the query's page in Databricks SQL.
C. They can schedule the query to run every 12 hours from the Jobs UI.
D. They can schedule the query to refresh every 1 day from the SQL endpoint's page in Databricks SQL.
E. They can schedule the query to refresh every 12 hours from the SQL endpoint's page in Databricks SQL.

Answer to Practice Question 1

Answer B

Practice Question 2

You have written a notebook to generate a summary data set for reporting, Notebook was scheduled using the job cluster, but you realized it takes 8 minutes to start the cluster, what feature can be used to start the cluster in a timely fashion so your job can run immediately?

A. Setup an additional job to run ahead of the actual job so the cluster is running when the second job starts

B. Use the Databricks cluster pool feature to reduce the startup time

C. Use Databricks Premium Edition instead of Databricks Standard Edition

D. Pin the cluster in the Cluster UI page so it is always available to the jobs

E. Disable auto termination so the cluster is always running.

Answer to Practice Question 2

Answer B

Cluster pools allow us to reserve VM's ahead of time, when a new job cluster is created VM are grabbed from the pool.

Note: when the VM's are waiting to be used by the cluster only cost incurred is Azure. Databricks run time cost is only billed once VM is allocated to a cluster.

https://www.youtube.com/watch?v=FVtlTxOabxg

Practice Question 3

Which of the following approaches can the data engineer use to obtain a version-controllable configuration of the Job's schedule and configuration?

A. They can link the job to notebooks that are a part of a Databricks Repo

B. They can submit the job once on a Job Cluster

C. They can download the JSON equivalent of the job from the Job's page

D. They can submit the Job once on a All-Purpose Cluster

E. They can download the XML description of the job from the Job's Page

Answer to Practice Question 3

Answer C

Dashboards

Use Databricks SQL for on-demand queries

Databricks SQL Endpoints

Creating SQL endpoints for different use cases

Query Scheduling

Scheduling query based on scenario

Query reruns based on interval time

Alerting

Configure notifications for different conditions

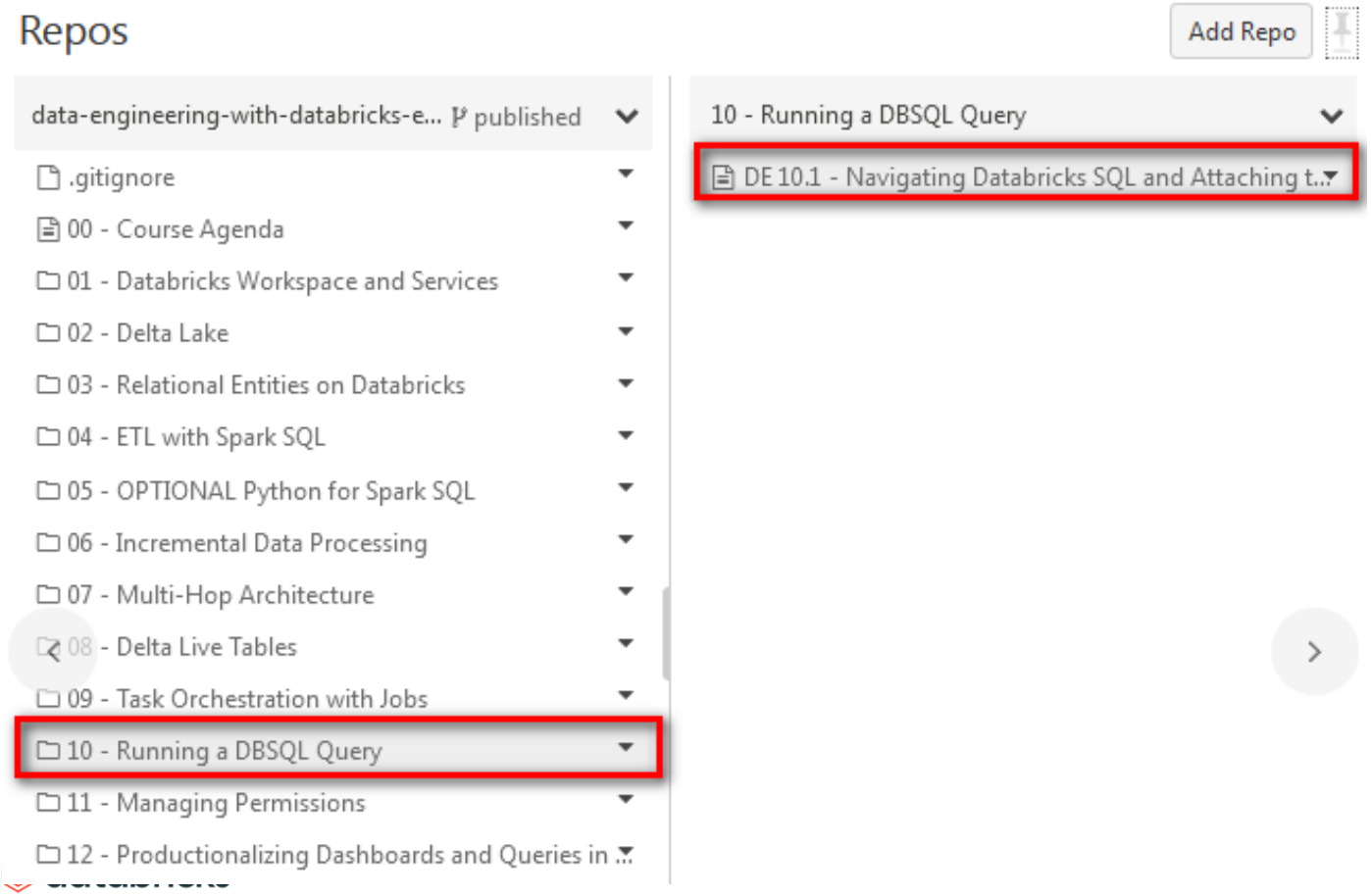
Configure and manage alerts for failure

Refreshing

Scheduling dashboard refreshes

Query reruns impact on dashboard performance

Ensure that you know this section completely



Focus on these links from docs.databricks page

1 https://docs.databricks.com/sql/admin/sql-endpoints.html

2 https://docs.databricks.com/sql/user/queries/schedule-query.html

3 https://docs.databricks.com/sql/user/alerts/index.html

4 https://docs.databricks.com/sql/admin/alert-destinations.html

5 https://docs.databricks.com/sql/language-manual/sql-ref-syntax-aux-cache-refresh-table.html

Practice Question 4

A data analyst has noticed that their Databricks SQL queries are running too slowly. They claim that this issue is affecting all of their sequentially run queries. They ask the data engineering team for help. The data engineering team notices that each of the queries uses the same SQL endpoint, but the SQL endpoint is not used by any other user.

Which of the following approaches can the data engineering team use to improve the latency of the data analyst's queries?

A. They can turn on the Serverless feature for the SQL endpoint.
B. They can increase the maximum bound of the SQL endpoint's scaling range.
C. They can increase the cluster size of the SQL endpoint.
D. They can turn on the Auto Stop feature for the SQL endpoint.
E. They can turn on the Serverless feature for the SQL endpoint and change the Spot Instance Policy to "Reliability Optimized."

Answer to Practice Question 4

Answer C

Practice Question 5

An engineering manager uses a Databricks SQL query to monitor their team's progress on fixes related to customer-reported bugs. The manager checks the results of the query every day, but they are manually rerunning the query each day and waiting for the results.

Which of the following approaches can the manager use to ensure the results of the query are updated each day?

A. They can schedule the query to run every 1 day from the Jobs UI.
B. They can schedule the query to refresh every 1 day from the query's page in Databricks SQL.
C. They can schedule the query to run every 12 hours from the Jobs UI.
D. They can schedule the query to refresh every 1 day from the SQL endpoint's page in Databricks SQL.
E. They can schedule the query to refresh every 12 hours from the SQL endpoint's page in Databricks SQL.

Answer to Practice Question 5

Answer B

Practice Question 6

Data engineering team has provided 10 queries and asked Data Analyst team to build a dashboard and refresh the data every day at 8 AM, identify the best approach to set up data refresh for this dashboard? Which of the following approaches can the manager use to ensure the results of the query are updated each day?

A. Each query requires a separate task and setup 10 tasks under a single job to run at 8 AM to refresh the dashboard

B. The entire dashboard with 10 queries can be refreshed at once, single schedule needs to be setup to refresh at 8 AM.

C. Setup Job with Linear Dependency to load all 10 queries into a table so the dashboard can be refreshed at once.

D. A Dashboard can only refresh one query at a time, 10 schedules to set up the refresh.

E. Use Incremental refresh to run at 8 AM every day

Answer to Practice Question 6

Answer B

Data Governance (9%)

Data Governance

Understand and follow best security practices

Unity Catalog

Benefits of Unity Catalog

Unity Catalog Features

Entity Permissions

Configuring access to production tables and database

Granting different levels of permissions to for users and groups

Table Access Control

Objects

- CATALOG
- DATABASE
- TABLE
- VIEW
- FUNCTION
- ANONYMOUS FUNCTION
- ANY FILE

Privileges

- SELECT
- CREATE
- MODIFY
- READ_METADATA
- CREATE_NAMED_FUNCTION
- ANONYMOUS FUNCTION
- ALL_PRIVILEGES

Table Access Control List

```
GRANT SELECT ON TABLE <schema-name>.<table-name> TO users
```

https://docs.databricks.com/security/access-control/table-acls/object-privileges.html

Practice Questions 1 - Managing permissions

What can you do with the data explorer?

A. Navigate databases tables views
B. Explore data schemas and metadata history
C. Set and modify permissions
D. All of the above

Practice Questions 2 - Managing permissions

The permission of the following objects can be configured:

A. CATALOG, DATABASE, TABLE, VIEW, FUNCTION, ANY FILE
B. CATALOG, DATABASE, TABLE, VIEW, FUNCTION
C. CATALOG, DATABASE
D. DATABASE, TABLE, VIEW, FUNCTION

Practice Questions 3 - Managing permissions

The MODIFY permission gives the ability to:

A. Add, delete and modify
B. Modify
C. Modify and delete
D. Modify and Add

Practice Questions 4 - Managing permissions

The USAGE permission gives

A. Ability to Add, delete and modify
B. No ability, it is an additional requirement to perform any action on a database object
C. Modify and delete
D. Modify and Add

We've reached the end of our course...

At this point, you should be able to:

- Understand the learning context behind the Databricks Certified Associate Data Engineer exam (the exam).
- Describe the format and structure of the exam.
- Describe the topics covered in the exam.
- Recognize the different types of questions provided on the exam.
- Identify resources that can be used to learn the material covered in the exam.