DATA · TAUGHT WITH APACHE SPARK

Data Engineering

Ingestion, storage formats, orchestration and scale. You start on one laptop with DuckDB, move to a lakehouse with real table formats, and reach for Spark only when the data earns it.

Intermediate | 5 phases | 22 sessions | 33 contact hours

Syllabus

5 phases, 22 sessions. Each phase ends in something you have built.

1 Ground Level

Move real data end to end on one laptop, with no cloud account and nothing to pay for.

5 sessions
$72 alone

1.1 What a Data Engineer Owns

Files and events in, trusted tables out, and who complains when they arrive late.

FREE PREVIEW

1.2 An Environment That Starts

Terminal, Git, uv, and Docker Compose: the lab you can tear down and rebuild in one command.

1.3 Files Are the Substrate

CSV, JSON, Parquet, compression, and why columnar changes every decision downstream.

1.4 SQL Straight Over Files

DuckDB querying Parquet and CSV in place: no server, no cluster, no excuses.

1.5 Python That Moves Data

Polars frames, HTTP pulls, retries, and a script that survives being run twice.

By the end: **A local pipeline turning messy source files into queryable tables.**

2 Pipelines That Repeat

Turn a script you babysit into a job that runs on a schedule without you.

4 sessions
$72 alone

2.1 Ingestion with dlt

Pull REST APIs and databases into typed tables with schema inference and incremental state.

2.2 Idempotence and Incrementals

Watermarks, merge keys, and re-running yesterday without writing a single duplicate.

2.3 Orchestration with Airflow 3

Assets, dependencies, retries, and backfills you trigger on purpose instead of by accident.

2.4 Tests, Contracts, Freshness

Row-level checks, schema contracts, and catching a broken source before the 9am meeting.

By the end: **A scheduled pipeline with incremental loads, tests, and a backfill that does not duplicate rows.**

3 The Lakehouse

Store tables that several engines can read, edit, and roll back without stepping on each other.

4 sessions
$72 alone

3.1 Laying Out Parquet

File sizes, partition columns, sort order, and the small-file problem you are about to create.

3.2 Iceberg Tables and Catalogs

Snapshots, manifests, and a REST catalog running locally beside object storage you host yourself.

3.3 Evolution and Time Travel

Add a column, rename it, delete rows, then read the table exactly as it was last Tuesday.

3.4 Table Maintenance

Compaction, expiring snapshots, orphan files, and why nobody notices until queries crawl.

By the end: **Iceberg tables in a local REST catalog, plus a maintenance job that keeps them fast.**

4 Spark When One Machine Is Not Enough

Reach for a cluster only when the data has earned it, then make the job fast.

4 sessions
$72 alone

4.1	**Why Spark, and When Not To**	The size and shape of job that beats DuckDB, and a Spark Connect session that starts without a JVM fight.
4.2	**DataFrames and Spark SQL**	The transformations you already wrote in Polars, expressed once and run across a cluster.
4.3	**Partitions, Shuffles, Skew**	The three reasons a distributed job runs slower than your laptop, and the fix for each.
4.4	**Spark on the Lakehouse**	Reading and writing Iceberg from Spark, then reading the query plan and the Spark UI to find the cost.

By the end: **A Spark job over your Iceberg tables, tuned, with the before and after explained.**

5 Streams and Operations

Keep unbounded data landing correctly, on time, and at a cost you can defend.

5 sessions
$72 alone

5.1	**Kafka Without ZooKeeper**	Topics, partitions, offsets, consumer groups, and a KRaft broker you start and break yourself.
5.2	**Event Time and Late Data**	Windows, watermarks, out-of-order events, and the delivery guarantee you actually get.
5.3	**Streaming Into Tables**	Structured Streaming from Kafka into Iceberg, with checkpoints and restarts that do not lose data.
5.4	**Lineage, Governance, PII**	OpenLineage, column lineage, table ownership, and deciding who is allowed to read what.
5.5	**SLAs, Alerts, and Cost**	Freshness targets, alerts that fire once, and an honest number for what your compute bills.

By the end: **A streaming pipeline into Iceberg with a published freshness SLA and a runbook.**

Tools you will use

Python | DuckDB | Polars | dlt | Apache Airflow 3 | Apache Iceberg | Apache Spark 4 | Apache Kafka

What you will build

1. **Raw Files to Trusted Tables**
   One laptop, no cloud account, tests included
2. **The Backfill Drill**
   Re-run any date twice, get the same rows
3. **Kafka to Iceberg**
   Late events, compaction, and a freshness SLA you publish

Registration

Phase 1: Ground Level	$72
Phase 2: Pipelines That Repeat	$72
Phase 3: The Lakehouse	$72
Phase 4: Spark When One Machine Is Not Enough	$72
Phase 5: Streams and Operations	$72
Whole track, all 5 phases	**$300**

Buying every phase separately costs $360, so the whole track saves $60.

No payment is taken online. Registering creates an order and payment instructions are sent with it. Access opens once payment is confirmed.