

Databricks: End-to-End Data Engineering – from the source to the Gold layer

Most ETL processes work – until the first rerun. In this training you build a process that can be run a second time without duplicates and handed over for maintenance.

Databricks: End-to-End Data Engineering is a workshop on building a complete data processing workflow in Databricks. The program focuses on one real problem: loading data from a source file and its increments, processing it into the Silver layer, checking its quality and preparing a stable Gold table for a business user.

Participants go through the whole process: from the data contract and design decisions, through Bronze and Silver with deduplication and MERGE, to a Gold table with metrics, a separate path for invalid records and running everything as a Lakeflow Job. Besides writing transformations in SQL and PySpark, the course shows how to build a process that can be run safely many times and handed over for maintenance. The process is built for RetailOps (a fictional retail company whose data we use for all exercises).



Training recipients

The training is intended for technical staff who build or maintain ETL/ELT processes in Databricks.

The course is particularly suitable for:

- data engineers responsible for flows from the source to the Gold layer,
- ETL developers migrating processes from classic tools to Databricks,
- technical analysts preparing data for reports and models,
- teams that want a consistent way of writing transformations and handling errors.

The training will be especially valuable for people who already know Delta Lake and want to see how the

individual pieces fit together into one repeatable process.



Benefits

- Designing an end-to-end process – you will learn to start from the data contract, the data delivery deadline (SLA) and business key before the first line of code is written.
- Ingestion with metadata – you will learn how to load data in batches and incrementally and how to choose a pattern for API, JDBC and CDC sources.
- Transformations in the Silver layer – you will learn standardization, deduplication, joining data and handling late-arriving data.
- Process repeatability (idempotency) – a rerun does not duplicate or corrupt data; you will learn to design MERGE and change reads this way.
- Data quality – you will build validation rules and a separate path for rejected records – an error table (quarantine).
- A Gold layer for the business – you will prepare a table with metrics and test it against edge cases.
- Running and maintenance – you will learn how to parameterize the process, set up retries and hand it over for maintenance.



Training program

1. Module – Process design: data contract and the end-to-end pattern

Module goal: the participant starts by agreeing terms with the data source and the consumer before writing any code.

- Data contract – what we get from the source: structure, types, key, how often and with what delay
- Data delivery deadline (SLA) – when the consumer needs the data and what happens when the source is late
- Division of responsibility – which errors the source system fixes and which the data platform handles
- Reruns and late-arriving data – decisions that have to be made at the design stage
- The RetailOps process pattern: raw → cleaned → reporting data, plus a separate table for invalid records
- Lab: the participant describes the orders data source and records the design decisions

2. Module – Loading data into the Bronze layer

Module goal: the participant loads data from the source in a repeatable way, with full information about where it came from.

- One-time and incremental loading – when reading a file is enough and when you need to detect new files (loading mechanisms are covered in detail in DBX-LKH)
- Metadata – file name, load time and batch ID stored with every record

- Other sources: APIs, databases, change streams (CDC) – which pattern to choose (demo of a ready-made solution)
- The Bronze layer rule – we store data as it arrived, without fixing it
- Lab: the participant loads the orders data into Bronze together with the metadata

3. Module – Transformations in the Silver layer (SQL and PySpark)

Module goal: the participant turns raw data into consistent, cleaned tables.

- Standardizing data – types, date formats, letter case, codes and column names
- Empty and invalid values – filling them in, flagging them or rejecting them
- Removing duplicates – picking the latest version of a record by key (window functions)
- Joining with lookup tables and handling records with no match
- Nested data from JSON and APIs – unpacking lists and structures into regular columns (overview with an example)
- Late-arriving data – updating periods that have already been processed
- Lab: the participant builds a transformation that correctly handles repeated and late-arriving records

4. Module – Updating tables and process repeatability

Module goal: the participant builds a process that gives the same result when it is run again.

- A short recap of MERGE from DBX-LKH; the focus is on updating for every batch of incremental data (foreachBatch)
- Change Data Feed – passing only the changed records on to the next layer; the new automatic version does not need to be enabled on each table (Databricks Runtime 19+, row tracking); the exercises use the classic version, which runs on LTS runtimes
- Repeatability test – running the process several times on the same data and comparing the results
- Lab: the participant runs the process three times and checks that the record count does not change

5. Module – Data quality and the error path

Module goal: the participant knows how to detect invalid data and what to do with it so it does not stop the whole process.

- Quality rules – completeness, value range, format, existence of the related key
- Constraints on Delta tables – what they block on write and what they will not catch
- The quarantine pattern – invalid records go to a separate table with the reason described, valid ones move on
- Quality metrics saved after every run – the basis for reporting on data quality
- Lab: the participant adds validation rules and routes invalid records to an error table

6. Module – The Gold layer and running the process

Module goal: the participant prepares data for the business and runs the whole process as a scheduled job.

- Gold tables – sales metrics, period-over-period comparisons, moving averages
- Edge case tests – an empty delivery, returns, order corrections
- Lakeflow Jobs – running the process as a job with a date parameter – full orchestration in DBX-LKF
- Workshop: participants put the steps built in modules 2-5 together into one Bronze → Silver → Gold process with an error table and run it as a Lakeflow Job. The outcome is a process that can be run many times with the same result.

- Code review and a list of good practices for the team's repository



Expected preparation of the participant

- Completion of DBX-LKH (Lakehouse and Delta Tables) or equivalent knowledge: Delta Lake, MERGE, Auto Loader
- Practical knowledge of SQL
- Basic knowledge of Python; PySpark elements (DataFrame, window functions) are covered in class, and starter code is provided

To increase the comfort of work and training's effectiveness we suggest using an additional monitor. The lack of additional monitor does not exclude participation in the training, however, it significantly influences the comfort of work during classes.

You can find information and requirements of participation in Distance Learning trainings at:

<https://www.altkomakademia.pl/en/how-it-works/about-courses/>



Training Includes

Training includes

- access to the Altkom Akademia student portal
- a prepared Databricks environment with RetailOps scenario data for the duration of the training
- notebooks and exercise materials in electronic form

Note: During selected training sessions, the Academy provides Participants with access to paid tools and services necessary to complete the training program (in particular Azure, Fabric, and other cloud resources). The Participant is required to use the provided resources solely within the scope of the training agenda and in accordance with the trainer's instructions. Use of these resources beyond the training scope—particularly for private projects, testing additional services, modifying the environment configuration, or undertaking activities that generate costs unrelated to the exercises—will result in the Participant being charged for costs exceeding the allocated limit, based on expenses incurred by the Academy.



Duration

1 days / 7 hours

Language

- Training: English
- Materials: English