

DataKern Azure Data Engineering Curriculum Dossier

AI writes the code. You learn to read it, reason about it, and defend the engineering choices behind it.

This dossier describes the Azure Data Engineering track the same way it is taught live: as an engineering discipline shaped by platform decisions, delivery discipline, and production accountability. The goal is not to rush through service names. The goal is to understand how Azure components fit together, where trade-offs appear, and how to explain your work with professional clarity.

Program Brief

	Details
Total Hours	Approximately 30 hours
Session Length	90 minutes
Total Sessions	20 live sessions
Phases	5
Mini Projects	5
Audience	Professionals moving into Azure batch data engineering
Investment	249 EUR or equivalent local currency
Platform Focus	Azure batch data engineering
Delivery Style	Live, guided, reasoning-first
Experience Required	No prior coding experience required

How the Program Runs

Every session follows the same operating rhythm:

1. A concept is introduced in plain language.
2. We implement it live with AI tools in the room.
3. We stop to reason about design, failure points, and trade-offs.
4. You practise before the session ends, so implementation stays tied to understanding.

What This Track Is For

- Professionals who want a serious entry point into Azure batch data engineering.
- Learners who want to understand storage, orchestration, Spark, delivery, and architecture as one connected system.
- People who want to learn how to read AI-generated code and defend engineering decisions rather than memorise service definitions.

What This Track Is Not For

- People expecting instant job conversion without disciplined follow-through.
- Learners looking for a generic analytics overview without implementation depth.
- Anyone who wants cloud vocabulary without understanding delivery, reliability, and operational ownership.

Phase 1 - Foundations and Data Engineering Mindset

DURATION	6 hours (4 sessions)
----------	----------------------

This opening phase establishes the mental model first. Before pipelines or platform services appear, you clarify what data engineering teams actually own, how Azure fits into that world, and why batch-first delivery still matters in many production environments.

Session 1 - What Data Engineering Is and Why It Matters

- Understand the role of data engineering in modern organizations.
- Distinguish Data Engineer from Analyst, Data Scientist, and ML Engineer.
- Set expectations for batch-first enterprise delivery.

Session 2 - The Data Lifecycle and Core Terminology

- Understand ingest, transform, store, serve, and monitor.
- Learn ETL, ELT, lineage, schema, idempotency, and orchestration language.
- Build the mental model behind reliable pipelines.

Session 3 - Azure Data Platform Big Picture

- Map ADLS, ADF, Databricks, Key Vault, and Azure DevOps into one platform view.
- Understand Bronze, Silver, and Gold architecture.
- Learn where each service earns its place.

Session 4 - Environment Setup and Team Workflow

- Set up Azure free-tier resources, Databricks, GitHub, and VS Code.
- Configure budget alerts before meaningful resource usage.
- Establish the working baseline for the rest of the program.

MINI PROJECT	Azure data platform blueprint for a fictional enterprise.
--------------	---

Phase 2 - Python, SQL, Spark, and Databricks

DURATION	9 hours (6 sessions)
----------	----------------------

Platform work without coding confidence creates fragility fast. This phase builds the practical language of data delivery so you can read transformations, reason about them, and move from isolated scripts toward pipeline-ready logic.

Session 5 - Python for Data Engineers I

- Understand variables, collections, file handling, and control flow.
- Read AI-generated Python with confidence.
- Learn to reason about what each block of code is doing.

Session 6 - Python for Data Engineers II

- Work with functions, modular structure, and exception handling.
- Use Pandas for practical transformation tasks.
- Learn how to shape scripts into reusable pipeline logic.

Session 7 - SQL for Transformation and Data Quality I

- Query, filter, join, aggregate, and order data confidently.
- Understand how SQL supports real reporting and transformation tasks.
- Build fluency in reading and correcting SQL logic.

Session 8 - SQL for Transformation and Data Quality II

- Apply window functions, validation patterns, and quality checks.
- Learn how to detect duplicates, null issues, and inconsistent records.
- Treat SQL as a production tool rather than an exam topic.

Session 9 - Spark and Distributed Thinking

- Understand why Spark exists and when it matters.
- Learn the Spark mental model: lazy execution, partitions, and transformations.
- Compare local data handling with distributed execution.

Session 10 - Databricks and PySpark in Practice

- Build transformations inside Databricks notebooks.
- Read schemas, clean data, and write Delta outputs.
- Prepare the technical foundation for Azure integration work.

MINI PROJECT	PySpark transformation notebook from raw input to trusted Delta output.
--------------	---

Phase 3 - Azure Services and Platform Integration

DURATION	6 hours (4 sessions)
----------	----------------------

This phase brings the platform together. You move from isolated components into connected Azure delivery, where storage strategy, orchestration, security, and integration quality begin to matter as part of one coherent system.

Session 11 - Azure Data Lake Storage Design

- Organize storage zones and folder strategy deliberately.
- Understand file formats, partition strategy, and naming conventions.
- Design storage that supports reliable downstream pipelines.

Session 12 - Azure Data Factory Orchestration

- Build pipelines, datasets, linked services, and triggers.
- Understand when orchestration belongs in ADF and when it does not.
- Design for repeatable, observable batch movement.

Session 13 - Secrets and Secure Connectivity

- Use Key Vault and managed access patterns for credential handling.
- Remove hardcoded secrets from engineering workflows.

- Learn practical security discipline for Azure delivery.

Session 14 - End-to-End Azure Integration

- Connect ADLS, ADF, Databricks, and Key Vault in one batch pipeline.
- Trace data movement and control flow across the platform.
- Validate the system as one coherent delivery path.

MINI PROJECT	End-to-end Azure batch pipeline with secure integration patterns.
--------------	---

Phase 4 - Production Delivery and Architecture Discipline

DURATION	6 hours (4 sessions)
----------	----------------------

The difference between a working demo and an engineering deliverable is usually discipline. This phase moves the work into version control, release thinking, observability, and the architecture conversations expected in professional settings.

Session 15 - Git, Branching, and Team Delivery

- Use Git intentionally for working changes, review, and rollback safety.
- Understand branching habits appropriate for engineering teams.
- Prepare pipeline assets for collaborative delivery.

Session 16 - CI/CD with Azure DevOps

- Build release logic for notebooks, configs, and pipeline assets.
- Understand environment promotion and deployment validation.
- Move from local experimentation to controlled delivery.

Session 17 - Testing, Observability, and Runtime Quality

- Add data quality checks, failure alerts, and monitoring signals.
- Learn how production teams discover pipeline degradation early.
- Treat testing and monitoring as core delivery work, not extras.

Session 18 - Architecture Reviews and Trade-Offs

- Read and produce architecture diagrams clearly.
- Explain service choices, failure points, cost risks, and scalability trade-offs.
- Practise the kind of reasoning expected in senior interviews.

MINI PROJECT	Production delivery pack with CI/CD, data quality, monitoring, and architecture notes.
--------------	--

Phase 5 - Portfolio and Interview Readiness

DURATION	3 hours (2 sessions)
----------	----------------------

The final phase is about communication quality. You translate the technical work into project narratives, architecture explanations, and interview answers that sound credible to hiring teams rather than rehearsed from a checklist.

Session 19 - Resume, Portfolio, and Project Positioning

- Turn project work into credible portfolio artifacts.
- Write concise explanations focused on business value and engineering rigor.
- Prepare a data engineering profile that can stand up in interviews.

Session 20 - Mock Interviews and Career Strategy

- Practise technical and behavioral interview responses.
- Defend design choices around Azure, Spark, storage, and orchestration.
- Build a realistic application plan for the next step in your career.

MINI PROJECT	Interview pack with project summary, resume points, and architecture talking points.
--------------	--

Learning Outcomes

By the end of the program, you will be able to:

- Explain the end-to-end Azure batch data platform clearly and credibly.
- Build transformations with Python, SQL, Spark, and Databricks.
- Connect core Azure data services into a secure batch pipeline.
- Deliver pipeline assets through Git and Azure DevOps workflows.
- Add testing, monitoring, and architecture documentation to your work.
- Present your engineering decisions with stronger clarity in interviews.

Prerequisites

- Curiosity about data platforms and how production systems behave.
- A laptop, internet connection, and willingness to practise between sessions.
- Readiness to engage live, ask questions, and reason through code carefully.

Azure free-tier setup and budget alerts are handled early in the program so resource usage stays disciplined.

Tooling Notes

This curriculum uses practical, low-friction tooling:

- Azure Data Lake Storage
 - Azure Data Factory
 - Azure Databricks
 - Azure Key Vault
 - Azure DevOps
 - GitHub
 - VS Code
 - AI assistants such as GitHub Copilot or ChatGPT
-

Curriculum Version 3.0 | DataKern Azure Data Engineering | Approximately 30 Hours | 20 Sessions | 5 Phases | 5 Mini Projects