



Azure Data Engineer – Crash Course (ADF+SQL)

Days 1–2: Foundations + SQL basics:

Day 1 – Cloud, Azure Setup + Core SQL

Theory (2–3h):

- Why data engineering; overview of data pipeline with ADF + Data Lake + Azure SQL.
- Cloud basics, Azure portal tour; what is Azure SQL and what is a table/row/column.
- ETL vs ELT, simple diagram: CSV → Data Lake → ADF → Azure SQL.
- SQL basics: SELECT, WHERE, ORDER BY, DISTINCT; data types; NULL concept.

Lab (3–4h):

- Each student: create resource group, Storage account, Azure SQL DB (or share one per team).
- In query editor: CREATE TABLE StudentDemo (Id, Name, Dept). INSERT a few rows, run SELECT * with WHERE and ORDER BY.
- Give a sample CSV (students table / products table).
- Students: Design table columns and types.
- CREATE TABLE, INSERT sample rows (script provided).
- Practice 10+ queries using SELECT + WHERE + ORDER BY + DISTINCT.

Day 2 – Data Lake Design + SQL Filtering/Aggregates

Theory

- What is Data Lake Gen2, containers, folders; raw vs processed vs curated (Bronze, Silver, Gold architecture – Data Lake house architecture).
- DDL vs DML: CREATE/ALTER/DROP vs INSERT/UPDATE/DELETE with simple examples.
- Good folder structures and naming (by date, source).
- SQL predicates: IN, BETWEEN, LIKE, IS NULL
- Intro to aggregate functions (COUNT, SUM) and GROUP BY.

Lab

- Instructor-guided: create Data Lake with hierarchical namespace, containers raw, processed, curated.
- Students: Upload CSVs into raw.
- In Azure SQL: Practice multiple ALTER TABLE (add new column), DROP column/table in a sandbox DB. INSERT, UPDATE, DELETE on their sample data (10–15 small exercises).
- Students design and create project folders: raw/sales/yyyy/mm, processed/sales/...SQL practice set: 5–6 queries with IN, BETWEEN, LIKE, IS NULL.
- Simple GROUP BY (e.g., count students by department, sum sales by product).

Days 3–6: ADF pipelines + stronger SQL:

Day 3 – ADF basics + Copy to Lake

Theory

- ADF components: linked service, dataset, pipeline, activity; concept of Integration Runtime.
- Copy Activity: patterns file → lake, file → SQL (high-level).

Lab

- Create ADF instance and linked services (Storage + Azure SQL).
- Build first pipeline: HTTP/local CSV → Data Lake raw container.
- Verify in portal that files landed correctly; students explore Data Lake structure.

Day 4 – ADF parameters + SQL aggregations

Theory

- ADF parameters for folder/file names; basics of triggers (schedule).
- Aggregates in SQL: COUNT, SUM, AVG, MIN, MAX; GROUP BY / HAVING with simple examples.

Lab

- Parameterize source folder name and run pipeline for two different days.
- In Azure SQL (data either manually loaded or via a simple pipeline): 8–10 queries with GROUP BY and HAVING (e.g., total sales per day, departments having more than X students).

Day 5 – Mapping Data Flows + Joins

Theory

- Mapping Data Flows: concept, using them for light transformations (filter, derive column, aggregate).
- Joins in SQL: INNER vs LEFT JOIN, join condition on keys.

Lab

- Build a Data Flow: Data Lake raw → processed with: Removing bad rows (filter).
- Converting data types, trimming strings.
- In Azure SQL: Create simple DimProduct, FactSales tables (script given). 6–8 join queries combining dimension and fact tables.

Day 6 – Control flow + Stored Procedure (SP) intro

Theory

- Control flow in ADF: activity dependencies, If Condition, on-failure path; monitoring runs.
- SQL: Why we use SPs for readability; simple SP examples, very light.

Lab

- Extend an existing pipeline: add a step that on failure writes a log entry to a Data Lake folder.
- In SQL: Convert a “long” query into a SP. 3–4 small exercises where students use SP for intermediate results (e.g., top products, then filter).

Days 7–10: Azure SQL focus + project:

Day 7 – Lake → Azure SQL load + validation

Theory

- Common pattern: Data Lake processed → Azure SQL tables via ADF; staging vs final tables (concepts only).
- Data validation using SQL: row counts, NULL checks in important columns.

Lab

- Build pipeline: Data Lake processed → Azure SQL final tables (Copy Activity).
- In SQL: Compare counts between two tables.
- Check for NULLs in key fields; basic quality queries like min/max date ranges.

Day 8 – Schema design + basic indexes (concept)

Theory

- Star schema: fact table + dimension tables, why it is used in reporting.

- Azure SQL basics: logical server, DB, compute tiers (very high-level, no pricing deep dive).
- Concept: what an index is and why it helps; no deep internals.

Lab

- Students design a small star schema for their project domain (e.g., ecommerce, student marks).
- Implement schema with CREATE TABLE + primary/foreign keys in Azure SQL (guided template).
- Optional demo: instructor shows creating one index and simple before/after query timing on large demo table (students observe).

Day 9 – Window functions (light) + analytics queries

Theory

- Window functions concept: partition, order, frame (only intuitively).
- Key functions: ROW_NUMBER, RANK, SUM OVER (PARTITION BY).

Lab

- On project tables: Use ROW_NUMBER to pick latest record per customer or per day.
- Use RANK to find top N items per category.
- Use SUM OVER (PARTITION BY, ORDER BY) for running total of sales/marks.
- Students write 5–6 analytics queries and save them in a .sql file for Day 12.

Day 10 – End-to-end mini-project presentations - Project scope (team of 3–4):

- Domain examples: sales, attendance, library, food delivery, etc.
- Required flow: Source files (CSV/JSON).
- ADF ingestion to Data Lake raw.
- ADF Data Flow or pipeline to processed/curated.
- Copy into Azure SQL star schema.
- Answer business questions using SQL joins, GROUP BY, and at least one window function.
- Deliverables per team Architecture diagram (one slide).
- Folder structure screenshot and ADF pipeline screenshots.
- SQL artifacts: CREATE TABLE scripts.5–7 queries: joins, aggregates, and 1–2 window function queries.