

Managing Big Data with Spark: Comprehensive Beginner Foundations

Build SQL and Python foundations before distributed data-processing exercises

FA-0714 · 7 days

Data & Analytics · Software Development

Introduction

Develop the SQL and Python foundations needed to approach Apache Spark, then build small batch-processing and introductory streaming examples. The course starts with no assumed programming or big-data experience and progresses through guided, bounded exercises.

Spark distributes processing across tasks and executors and can use memory and disk. Performance depends on workload, partitioning, storage and resources; the course does not promise petabyte-scale labs or fixed speed improvements over other systems.

Learning outcomes

- Explain relational data, big-data pipelines and structured or semi-structured inputs.
- Write foundational SQL queries and recognise database-specific procedural SQL.
- Use Python collections, control flow, functions and basic file or pandas operations.
- Describe Spark drivers, executors, jobs, stages and cluster-manager roles.
- Read, transform, join and write sample data using PySpark DataFrames and Spark SQL.
- Inspect shuffles and query plans, and introduce Structured Streaming and a small machine-learning example without claiming production mastery.

Prerequisites

- Basic computer, browser and file-management skills; no prior SQL, Python or Spark knowledge required.
- Access to the supplied database and compatible Python/Spark lab environment.
- For remote delivery, reliable internet and supported conferencing tools; a second screen is helpful, not mandatory.
- Any remote connection or file-transfer tools are specified for the supplied lab; arbitrary external-IP access and a particular client are not universal requirements.

Training outline

1. Day 1 — Data pipelines and relational SQL

- Big Data
- Tools
- Structure
- Pipeline and flow
- Data Source

Relational tables, keys and the role of a database in a data pipeline.

- SELECT, WHERE, LIKE, ordering, IN and BETWEEN; aggregate functions and grouping.
- Joins and subqueries; recognise insert, update, delete and schema changes in an isolated relational database.
- Introduce database functions and stored procedures as database-specific concepts; SQL Server T-SQL is not interchangeable with Spark SQL.
- Inspect sample datetime, JSON and XML data and distinguish formats from structured, semi-structured or unstructured content.

2. Day 2 — Python fundamentals

Basics

- IDEs
- Environment Variables
- Python Documentation
- Hello world with python
- Modes of Programming

Variables & Collections

- Numbers
- Python Lists
- Python Tuples
- Python Dictionaries
- Python Sets
- Copying
- Python Strings
- String formatting
- Regular Expressions

Operators

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Membership Operators
- Operators Precedence

Decisions and Loops

- Decision Making
- The if Statement
- The if else Statement
- For Loop
- While Loop
- Break And Continue

3. Day 3 — Functions and practical data handling

Functions and Subroutines

- Defining Your Own Functions
- Parameters
- Function Documentation
- Passing Collections to a Function
- Variable Number of Arguments
- Scope
- Map
- Filter
- Lambda

Data Handling

- Pandas
- Recursion
- File Access

Introduce modules, external libraries, basic objects and error handling through the small data exercise; recursion is awareness rather than a prerequisite for Spark.

4. Day 4 — Spark architecture and DataFrames

- Driver, executors, tasks, jobs and stages; compare standalone, Hadoop YARN and Kubernetes deployment contexts.
- Start a Spark session and distinguish the structured DataFrame API from lower-level RDDs.
- Read sample supported sources, define or inspect schemas and apply basic transformations.
- Use DataFrame operations and Spark SQL together; inspect lazy execution and trigger an action.

5. Day 5 — Transform, aggregate and write

- Clean selected columns, handle nulls and apply suitable types and date operations.
- Group, aggregate and join sample datasets; compare built-in functions with user-defined functions.
- Read JSON or other supported formats, examine malformed input and write a bounded output dataset.
- Discuss connectors and source/sink compatibility rather than assuming Spark reads every Hadoop data store automatically.

6. Day 6 — Shuffles, plans and streaming foundations

- Inspect partitions, shuffles and execution plans; compare a selected optimisation against a measured baseline.
- Understand memory, disk and caching trade-offs; more caching is not always beneficial.
- Introduce a small Structured Streaming source, transformation and sink with checkpoint and restart considerations.
- Review the role of cluster resources and YARN as one deployment option, not a mandatory Spark dependency.

7. Day 7 — Introductory machine learning and a pipeline review

- Introduce clustering and a selected MLlib pipeline using a small prepared dataset.
- Separate data preparation, model fitting and evaluation; avoid treating a demonstration model as a validated business solution.

- Complete and review a small end-to-end data pipeline, document assumptions and inspect its output.
- Summarise next steps for deeper Spark operations, streaming correctness and model evaluation.