COMPLETE MASTER PROGRAM IN

DATA ENGINEERING

Python • SQL • ETL/ELT • PySpark • Data Warehousing
Data Quality • File Processing • End-to-End Pipelines

3 MONTHS | 120 GUIDED HOURS

Beginner-friendly • Guided learning • Hands-on projects

9 structured modules • Real datasets • Career preparation

Learn. Practice. Get Hired.

TOOLS & TECHNOLOGIES

Practical tools and concepts covered throughout the program

PYTHON | SQL | PYSPARK | PANDAS | NUMPY | ETL / ELT

PROGRAM AT A GLANCE

Duration: 3 months

Guided learning: 120 hours

Suggested pace: 10 guided hours per week over 12 weeks

Who can join: Students, graduates and working professionals; beginner-friendly

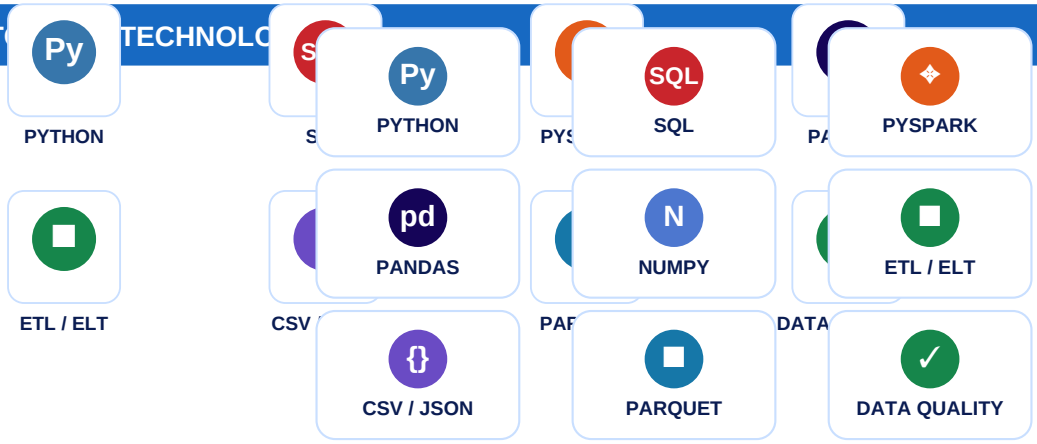
Core stack: Python, SQL, ETL/ELT, PySpark, data modeling and file processing

LEARNING PATH

- Python foundations and file handling
- SQL querying and relational data modeling
- ETL/ELT concepts and pipeline development
- PySpark distributed processing and transformations
- Data warehousing, file formats and quality checks
- End-to-end capstone project and interview preparation

TOOLS & TECHNOLOGIES

Each tool is shown in its own tile for clear, readable identification.



MODULE 01

Python Programming for Data Engineering

Build reliable Python skills for reading, transforming and validating data in repeatable workflows.

TOPICS COVERED

- Python syntax, variables, data types and operators
- Strings, lists, tuples, sets and dictionaries
- Conditions, loops, functions and reusable modules
- Working with files, folders, paths and exceptions
- Reading and writing CSV, JSON and text files
- Using datetime, collections and standard-library utilities
- Logging, debugging and writing maintainable scripts

REAL-WORLD EXAMPLE

Read daily sales files, validate required fields and produce a clean consolidated output file.

PRACTICE PROJECT

Build a Python ingestion script that processes multiple files, records errors and generates a processing summary.

MODULE 02

SQL Programming and Advanced Querying

Query relational data confidently and create reliable transformations using SQL.

TOPICS COVERED

- SELECT, filtering, sorting, aliases and expressions
- Aggregate functions, GROUP BY and HAVING
- INNER, LEFT, RIGHT and FULL joins
- Subqueries, correlated subqueries and common table expressions
- CASE expressions, NULL handling and date/string functions
- Window functions: ROW_NUMBER, RANK, LAG and LEAD
- INSERT, UPDATE, DELETE, constraints and transactions
- Query plans, indexes and practical query optimization

REAL-WORLD EXAMPLE

Combine customer, order and product tables to calculate daily sales and identify repeat customers.

PRACTICE PROJECT

Create a set of SQL transformations and reporting queries for a sample retail database.

MODULE 03
Database Design and Data Modeling

Design structured data models that support dependable operational and analytical workloads.

TOPICS COVERED

- Relational database concepts, tables, rows and keys
- Primary keys, foreign keys and integrity constraints
- Normalization and reducing unwanted duplication
- Entity-relationship diagrams and relationship cardinality
- Fact tables, dimension tables and star schemas
- Choosing data types and handling timestamps
- Basic slowly changing dimension concepts

REAL-WORLD EXAMPLE

Model customers, products, orders and order items for an e-commerce business.

PRACTICE PROJECT

Design an ER diagram and create normalized tables plus an analytical star schema.

MODULE 04
ETL and ELT Pipeline Development

Understand how raw data is extracted, transformed, validated and delivered to target systems.

TOPICS COVERED

- ETL vs. ELT and common pipeline patterns
- Batch processing, source extraction and target loading
- Full loads and incremental loads
- Data mapping, type conversion and transformation rules
- Idempotency, retries and safe reprocessing
- Handling duplicate records and late-arriving data
- Pipeline logging, checkpoints and error handling
- Building reusable pipeline components

REAL-WORLD EXAMPLE

Load daily order extracts, standardize records, reject invalid rows and update an analytics table.

PRACTICE PROJECT

Build a small end-to-end pipeline with extraction, transformation, validation and load steps.

MODULE 05

PySpark and Distributed Data Processing

Use PySpark to process large datasets with distributed transformations and Spark SQL.

TOPICS COVERED

- Big data concepts and distributed processing fundamentals
- Spark architecture: driver, executors, jobs, stages and tasks
- SparkSession and DataFrame creation
- Reading and writing CSV, JSON and Parquet data
- Selecting, filtering, sorting and transforming columns
- Handling nulls, duplicates, casts and date/time fields
- Joins, aggregations, groupBy and window functions
- Built-in functions, UDFs and avoiding unnecessary UDF use
- Lazy evaluation, transformations and actions
- Partitions, shuffles, caching and performance basics

REAL-WORLD EXAMPLE

Process several million sales records to calculate regional revenue, monthly trends and product rankings.

PRACTICE PROJECT

Develop a PySpark batch-processing workflow that cleans, enriches and aggregates a large sales dataset.

MODULE 06
Data Warehousing and Analytical SQL

Organize curated data for efficient business reporting and analytical consumption.

TOPICS COVERED

- Operational databases vs. analytical data warehouses
- Warehouse layers: raw, cleaned and curated data
- Fact and dimension table design
- Star schema and snowflake schema comparison
- Surrogate keys and basic dimension history handling
- Aggregations, summary tables and reusable views
- Data marts, business metrics and reporting readiness

REAL-WORLD EXAMPLE

Prepare a curated sales model for revenue, margin, product and regional performance reporting.

PRACTICE PROJECT

Build a warehouse-style data model with fact/dimension tables and analytical SQL queries.

MODULE 07

Data Formats, Storage and File Processing

Select appropriate data formats and manage files efficiently for batch data workflows.

TOPICS COVERED

- CSV, JSON, line-delimited JSON and Parquet formats
- Row-based vs. columnar storage concepts
- Compression basics and file-size trade-offs
- Schema inference vs. explicit schema definition
- Reading multiple files and processing folder-based inputs
- Partitioned output folders and date-based organization
- Handling malformed files, encoding and schema drift

REAL-WORLD EXAMPLE

Compare CSV and Parquet for a date-partitioned dataset used by recurring reports.

PRACTICE PROJECT

Create a file-processing utility that validates schemas and writes cleaned output in an efficient format.

MODULE 08

Data Quality, Troubleshooting and Optimization

Improve trust in data by detecting defects, tracing failures and tuning processing workflows.

TOPICS COVERED

- Completeness, validity, consistency, uniqueness and accuracy
- Null checks, range checks, referential and business-rule checks
- Duplicate detection and reconciliation of record counts
- Bad-record handling, quarantine and error reports
- Pipeline monitoring through logs and processing metrics
- Troubleshooting failed transformations and incorrect results
- SQL and PySpark performance troubleshooting
- Documentation, lineage concepts and data governance basics

REAL-WORLD EXAMPLE

Identify missing customer IDs, invalid dates and duplicate orders before publishing a daily dataset.

PRACTICE PROJECT

Create a reusable data-quality validation layer with a report of passed and failed checks.

MODULE 09

Capstone Projects and Interview Preparation

Bring the core skills together in portfolio-ready projects and practice explaining engineering decisions.

TOPICS COVERED

- Capstone planning, source profiling and requirements
- Designing a complete batch data pipeline
- Python ingestion, SQL transformations and PySpark processing
- Data validation, exception handling and audit summaries
- Creating curated analytical tables and business metrics
- Testing edge cases and documenting pipeline behavior
- SQL, Python and PySpark coding practice
- Data engineering architecture and troubleshooting questions
- Project walkthrough, resume points and interview communication

REAL-WORLD EXAMPLE

Build a retail pipeline that ingests raw files, cleans data, calculates KPIs and prepares curated tables.

PRACTICE PROJECT

Deliver an end-to-end data engineering capstone with code, data model, quality checks and project documentation.

KEY LEARNING OUTCOMES

- Write Python scripts for data ingestion, transformation and validation.
- Query, join and transform relational data using SQL.
- Build ETL/ELT workflows with error handling and safe reprocessing.
- Use PySpark DataFrames to clean, join and aggregate large datasets.
- Design warehouse-style models and prepare curated analytical tables.
- Apply data-quality checks and troubleshoot common pipeline failures.
- Complete a practical project and explain engineering decisions clearly.

CAREER OPPORTUNITIES

- **Junior Data Engineer**
- **ETL Developer**
- **PySpark Developer**
- **Data Integration Developer**
- **SQL Developer**
- **Data Operations Analyst**
- **Data Quality Analyst**
- **Analytics Engineer (Junior)**