Data Engineering Professional Program

Data Engineering Professional Program

3-Month Comprehensive Curriculum

Month 1: Database Architecture & Linux Fundamentals

Goal: Design robust database schemas, understand data storage, and navigate server environments.

Week 1: Advanced SQL & Database Design

- **Data Modeling:** Normalization (1NF to 3NF), Entity-Relationship (ER) diagrams, and designing scalable schemas.
- **Query Optimization:** Understanding execution plans, indexing strategies, and partitioning for handling large datasets.
- **Advanced RDBMS:** Utilizing Stored Procedures, Triggers, and managing ACID transactions in PostgreSQL/MySQL.

Week 2: Data Warehousing Concepts

- **Architecture:** Differences between OLTP (Transactional) and OLAP (Analytical) systems, and understanding Data Lake vs. Data Warehouse vs. Data Lakehouse.
- **Schema Design:** Implementing Star and Snowflake schemas specifically tailored for analytical workloads.
- **Cloud Warehouses:** Introduction to cloud-native warehousing concepts and architecture using Snowflake or Google BigQuery.

Week 3: Python for Data Engineering

- **Python Fundamentals:** Object-oriented programming (OOP), robust error handling, and writing modular, maintainable code.
- **Data Connectors:** Using Python libraries (e.g., psycopg2, SQLAlchemy) to connect, query, and load data into databases programmatically.
- **File Formats:** Reading, writing, and optimizing large datasets in various Big Data formats (CSV, JSON, Parquet, Avro).

Week 4: Linux & Bash Scripting

- **OS Fundamentals:** Navigating the Linux file system, managing user permissions, and mastering essential CLI commands.
- **Bash Scripting:** Writing shell scripts to automate file transfers, log rotations, and system monitoring tasks.
- **Task Scheduling:** Setting up and managing Cron jobs for automated, time-based script execution on remote servers.

[Project 1: Automated Data Extraction & Storage]

- **Scope:** Write a Python script executed via a Bash Cron job on a Linux server to extract data from a public REST API, perform basic transformations, and load it into a normalized PostgreSQL database.

Month 2: Big Data Processing & ETL/ELT Pipelines

Goal: Build scalable automated data pipelines using modern distributed computing frameworks and orchestration tools.

Week 5: Introduction to Big Data & Hadoop Ecosystem

- **Big Data Concepts:** The 5 V's of Big Data and understanding the shift from centralized to distributed storage and compute.
- **Hadoop Ecosystem:** Overview of HDFS (Hadoop Distributed File System) and the foundational MapReduce programming model.
- **NoSQL Databases:** Introduction to columnar and document databases (e.g., Cassandra, MongoDB) for managing unstructured data.

Week 6: Distributed Data Processing (Apache Spark)

- **Spark Architecture:** Understanding the Spark core, Resilient Distributed Datasets (RDDs), and cluster managers.
- **PySpark DataFrames:** Processing, filtering, and joining massive datasets using the PySpark API.
- **Performance Tuning:** Managing partitions, caching, broadcasting, and handling data skewness in large-scale Spark jobs.

Week 7: Building ETL/ELT Pipelines

- **Data Integration:** Designing robust Extract, Transform, and Load (ETL) vs. Extract, Load, and Transform (ELT) workflows.
- **API Integration:** Handling rate limits, pagination, and secure authentication when ingesting data from external APIs.
- **Data Transformation:** Cleaning, standardizing, and enriching complex data streams before loading them into a data warehouse.

Week 8: Data Orchestration with Apache Airflow

- **Airflow Fundamentals:** Understanding DAGs (Directed Acyclic Graphs), Tasks, Operators, and scheduling logic.
- **Workflow Automation:** Building and deploying Python-based Airflow pipelines to orchestrate complex data dependencies.
- **Monitoring & Alerting:** Configuring task retries, email/Slack alerts on failure, and monitoring pipeline health via the Airflow UI.

[Project 2: Scalable Batch ETL Pipeline]

- **Scope:** Design and orchestrate a batch ETL pipeline using Apache Airflow that extracts daily financial data, processes and cleans it via PySpark, and loads it into a Snowflake Data Warehouse.

Month 3: Cloud Data Engineering & Real-Time Streaming

Goal: Master cloud data services, real-time data ingestion, and enterprise data governance practices.

Week 9: Cloud Data Services (AWS)

- **Cloud Storage:** Configuring AWS S3 buckets for Data Lakes, managing lifecycle policies, and setting up IAM access controls.

- **Serverless Data Integration:** Utilizing AWS Glue for managed ETL jobs, data crawlers, and data cataloging.
- **Serverless Querying:** Using Amazon Athena to run interactive SQL queries directly against flat files stored in S3.

Week 10: Real-Time Data Streaming (Apache Kafka)

- **Streaming Architecture:** Differences between batch processing and real-time event streaming architectures.
- **Kafka Fundamentals:** Setting up clusters, creating Topics, and understanding the roles of Producers, Consumers, and Brokers.
- **Stream Processing:** Integrating Kafka with Spark Structured Streaming to perform real-time data transformations.

Week 11: Data Governance & Quality

- **Data Quality:** Implementing automated data validation checks (e.g., Great Expectations), handling anomalies, and ensuring data accuracy.
- **Security & Privacy:** Applying data masking, encryption at rest/transit, and understanding compliance standards (GDPR/CCPA).
- **Data Catalogs:** Utilizing metadata management tools to enable easier data discovery for analysts and data scientists.

Week 12: Capstone Project & Career Readiness

- **System Design:** Practicing data architecture whiteboard sessions and evaluating trade-offs in pipeline design.
- **Portfolio Building:** Documenting end-to-end architectures on GitHub with clear architectural diagrams and clean code.
- **Interview Prep:** Technical preparation focusing on SQL optimizations, Python coding, and behavioral interviews specific to Data Engineering roles.

[Project 3: Real-Time E-commerce Streaming (Capstone)]

- **Scope:** Build a fault-tolerant streaming pipeline using Apache Kafka to ingest live e-commerce clickstream data, process it in real-time using Spark Streaming, and store aggregated metrics in AWS S3 for downstream visualization.