Revised Data Engineering Curriculum

Month 1: Foundation Building

Week 1: Python for Data Engineering

1. Quick Wins with Python:
   - Learn Python basics (data types, loops, and conditionals) by building small programs, such as a CSV data analyzer.
   - Hands-on Challenge: Write a script to parse a log file and extract useful information.
2. Data Manipulation & Visualization:
   - Use pandas to clean and manipulate datasets.
   - Visualize results with matplotlib or seaborn (e.g., show the top 5 items sold from a dataset).
   - Mini-Project: Create a script to process sales data and output a report with visualizations.
3. Automate Simple Tasks with Python:
   - Write Python scripts to automate common tasks like renaming files or scraping data from websites.
   - Hands-on Challenge: Write a program that automatically downloads and organizes files from a given URL.

Week 2: SQL

1. Foundations of SQL:
   - Learn the basics of SQL by querying a mock e-commerce database.
   - Interactive Challenge: Retrieve top-selling products, identify the highest-paying customers, and calculate monthly revenue.
2. SQL for Data Transformation:
   - Perform data cleansing and transformation with SQL.
   - Hands-On Activity: Normalize a poorly designed database into a clean, efficient structure.
3. Mini-Project: Build Your First SQL Portfolio:
   - Design a small database for a real-world scenario (e.g., a library system or movie database). Write queries to extract meaningful insights.

Week 3: Linux/Bash Scripting

1. Master Linux Basics:
    - Learn the Linux file system and basic commands (ls, grep, find, etc.).
    - Hands-On Activity: Automate file organization using bash scripting.
2. Write Bash Scripts for Data Tasks:
    - Write scripts to automate file cleaning, data merging, and scheduling tasks with cron jobs.
    - Challenge: Process and clean a folder of messy log files using bash.
3. Mini-Project:
    - Automate a simple ETL pipeline using Bash (e.g., extract files from a folder, clean them, and move them to an organized directory).

Week 4: Data Engineering Basics

1. Introduction to Data Pipelines:
    - Learn the concept of ETL (Extract, Transform, Load) and its importance in data engineering.
    - Hands-On Activity: Write a Python script to perform a basic ETL process on a small dataset.
2. Data Formats and Storage Systems:
    - Work with data formats like CSV, JSON, and Parquet.
    - Challenge: Convert a dataset into multiple formats and compress it.
3. Mini-Project:
    - Build an end-to-end pipeline to clean a dataset and load it into an SQLite database.

Month 2: Advanced Tools and Concepts

Week 5: Big Data Tools

1. Apache Spark Introduction:
    - Learn Spark fundamentals, including RDDs and DataFrames.
    - Hands-On Challenge: Analyze a large dataset with PySpark and calculate insights like average sales per region.
2. Spark SQL for Big Data:
    - Write SQL queries in Spark to process structured data.

- Activity: Query and process data stored in a Parquet file.

3. Mini-Project:
    - Process a large dataset using PySpark and generate a summary report.

Week 6: Workflow Orchestration

1. Apache Airflow Basics:
    - Learn to create and deploy DAGs for task orchestration.
    - Hands-On Activity: Schedule a daily task to clean and archive logs.
2. Advanced Airflow Features:
    - Use XComs for data sharing between tasks, and integrate Airflow with cloud storage.
    - Challenge: Orchestrate a multi-step pipeline involving data extraction, cleaning, and storage.
3. Mini-Project:
    - Build and deploy an Airflow pipeline that ingests API data and loads it into a database.

Week 7: Data Modeling and Advanced SQL

1. Dimensional Data Modeling:
    - Design star and snowflake schemas for business scenarios.
    - Activity: Create a star schema for a sales reporting database.
2. Advanced SQL Techniques:
    - Learn window functions, CTEs, and recursive queries.
    - Challenge: Write complex queries to calculate rolling averages and rank data.
3. Mini-Project:
    - Design a data warehouse schema and populate it with mock data.

Week 8: Azure Cloud for Data Engineering

1. Azure Data Storage and Processing:
    - Learn Azure Blob Storage and Azure Data Factory basics.
    - Activity: Store and retrieve files in Azure Blob Storage.

2. ETL with Azure Data Factory:
    - Build pipelines to process data in Azure.
    - Challenge: Set up an Azure Data Factory pipeline to clean data and save it in Blob Storage.
3. Mini-Project:
    - Build a cloud-based ETL pipeline using Azure Data Factory and Blob Storage.

Month 3: Real-World Applications

Week 9: Databricks for Big Data Analytics

1. Introduction to Databricks:
    - Overview of Databricks and its role in big data processing.
    - Setting up a Databricks workspace and understanding clusters, notebooks, and jobs.
2. Mounting Azure Data Lake on Databricks:
    - Connect Databricks to Azure Data Lake and access large datasets.
    - Hands-on Activity: Mount an Azure Data Lake container to your Databricks workspace.
3. Analyzing Large Datasets:
    - Process and analyze large datasets using Spark DataFrames in Databricks.
    - Perform transformations, aggregations, and visualizations in Databricks notebooks.
    - Challenge: Analyze a 1GB+ dataset stored in Azure Data Lake and generate key business insights.
4. Mini-Project:
    - Build an end-to-end data analysis solution:
        - Mount an Azure Data Lake container on Databricks.
        - Extract, transform, and analyze data using Spark.
        - Visualize the results in Databricks notebooks.

Week 10: DevOps for Data Engineering

1. CI/CD Pipelines:
    - Set up a CI/CD pipeline for deploying data pipelines.

   - Activity: Use GitHub Actions to test and deploy a data pipeline.

2. Docker for Data Engineering:
   - Learn to containerize data engineering workflows.
   - Challenge: Deploy a pipeline in Docker and scale it.
3. Mini-Project:
   - Build and deploy a containerized ETL pipeline with a CI/CD process.

Week 11–12: Capstone Project

1. Design an End-to-End Solution:
   - Create a complete ETL pipeline:
     - Extract data from APIs, files, or databases.
     - Transform data with Python/Spark.
     - Load data into an Azure SQL database or a data warehouse.
2. Enhance with Streaming and Cloud:
   - Add a real-time streaming component with Kafka and Spark.
   - Deploy the solution to Azure with containerization and CI/CD.
3. Portfolio Presentation:
   - Document and present the solution as a portfolio project.