Masters of Science in Data Engineering

African Digital Academy

Open Learning Campus

Course outline:

Day 1: Overview of Data Engineering Concepts and Tools

Introduction to Data Engineering:

Data engineering as a discipline focuses on the acquisition, storage, processing, and delivery of data.

- ✓ Discuss the importance of data engineering in enabling data-driven decision making and supporting data science and analytics initiatives.
- ✓ Highlight the role of data engineers in the data lifecycle and their responsibilities in managing data pipelines.

Data Engineering Lifecycle:

i. **Data Ingestion:** Explain the process of acquiring data from various sources, including databases, APIs, files, and streaming platforms.

ii. **Data Transformation:** Discuss the need to clean, validate, and transform raw data into a usable format for analysis.

iii. **Data Storage:** Explain different storage options such as databases, data warehouses, and data lakes, and their pros and cons.

iv. **Data Processing:** Highlight the importance of scalable and efficient data processing techniques for handling large volumes of data.

v. **Data Delivery:** Discuss the methods of delivering processed data to downstream systems, applications, or end-users.

Data Engineering Tools and Technologies:

a) Data Integration Tools:

Apache Kafka: Explain how Kafka enables high-throughput, fault-tolerant, and real-time data streaming and messaging.

Apache Nifi: Discuss the capabilities of Nifi for data ingestion, routing, transformation, and integration.

AWS Glue: Highlight the managed ETL service provided by AWS for extracting, transforming, and loading data.

b) Data Processing Frameworks:

Apache Spark: Discuss Spark's distributed computing capabilities for processing large-scale data and its support for various programming languages.

Apache Flink: Explain Flink's stream processing and batch processing capabilities and its integration with popular data storage systems.

Apache Beam: Highlight Beam's unified programming model for batch and streaming data processing across different execution engines.

c) ETL Tools:

Apache Airflow: Introduce Airflow as a platform for orchestrating and scheduling data pipelines, including support for dependency management and monitoring.

Informatica: Discuss Informatica's comprehensive ETL capabilities, including data profiling, data quality, and metadata management.

Talend: Explain Talend's open-source and cloud-based ETL platform, which offers a wide range of connectors and transformation components.

d) Cloud Platforms:

AWS: Discuss Amazon Web Services (AWS) as a cloud platform offering various services for data storage (S3), computing (EC2), and analytics (Redshift, Athena).

Azure: Highlight Microsoft Azure's data services, such as Azure Data Lake Storage, Azure Databricks, and Azure Synapse Analytics.

GCP: Introduce Google Cloud Platform (GCP) and its services like BigQuery, Cloud Storage, and Dataflow for data processing and analytics.

e) Data Storage Systems:

Relational Databases: Explain the use of relational databases for structured data storage, including popular options like MySQL, PostgreSQL, and Oracle.

NoSQL Databases: Discuss the advantages of NoSQL databases for handling unstructured and semi-structured data, mentioning databases like MongoDB and Cassandra.

Data Warehouses: Highlight the purpose and benefits of data warehouses for storing and analyzing structured data at scale, including solutions like Amazon Redshift and Google BigQuery.

f) Data Visualization and Reporting Tools:

Tableau: Introduce Tableau as a widely used data visualization tool for creating interactive dashboards and reports.

Power BI: Discuss Microsoft Power BI's capabilities for data visualization, self-service analytics, and collaboration.

Looker: Highlight Looker as a platform for building data exploration and analytics applications.

g) Data Engineering Best Practices:

Data Governance: Explain the importance of establishing data governance frameworks to ensure data quality, privacy, and compliance

h) Case Studies

Provide real-world case studies or examples of successful data engineering projects, highlighting the challenges faced and the solutions implemented.