Data Engineering Notes

Chuks Okoli

02 November, 2023

Introduction to Data Engineering

In the simplest possible terms, the field of Data Engineering concerns itself with the mechanics for the flow and access of data. And its goal is to make quality data available for fact-finding and data-driven decision making.

The field of Data Engineering involves:

1. Collecting source data
    - Extracting, integrating, and organizing data from disparate sources
        - Develop tools, workflows, and processes that help data acquisition from multiple sources
        - Design, build, and maintain scalable data architecture for storing source data
2. Processing data
    - Cleaning, transforming, and preparing data to make it usable
        - Implement and maintain distributed systems for large-scale data processing
        - Design pipelines for the extraction, transformation, and loading of data into data repositories
        - Design solutions for validating and safeguarding quality, privacy, and security of data
        - Performance optimization
        - Adherence to compliance guidelines
3. Storing data
    - Storing data for reliability and easy availability of data
        - Data stores for storage of processed data
        - Scalable systems
        - Ensuring data privacy, security, compliance, monitoring, backup, and recovery
4. Making data available to users securely
    - APIs, services, and programs for retrieving data for end-users
    - User access through interfaces and dashboards
    - Checks and balances to ensure data security

Modern data ecosystem includes a network of interconnected and continually evolving entities that include:

- Data, that is available in a host of different formats, structures, and sources.
- Enterprise Data Environment, in which raw data is staged so it can be organized, cleaned, and optimized for use by end-users.
- End-users, such as business stakeholders, analysts, and programmers who consume data for various purposes.

Emerging technologies such as Cloud Computing, Machine Learning, and Big Data, are continually reshaping the data ecosystem and the possibilities it offers.

Data Engineers, Data Analysts, Data Scientists, Business Analysts, and Business Intelligence Analysts, all play a vital role in the ecosystem for deriving insights and business results from data.

Responsibility of a Data Engineer

- Extract, organize, and integrate data from disparate sources
- Prepare data for analysis and reporting by transforming and cleansing it
- Design and manage data pipelines that encompasses the journey of data from source to destination systems
- Setup and manage infrastructure for ingestion, processing and storage of data including data platforms, data stores, distributed systems, and data repositories.

Skillset of a Data Engineer
Some of the technical skills includes knowledge of:

- Operating Systems
  - UNIX
  - Linus
  - Windows Administrative Tools
  - System Utilities & Commands
- Infrastructure Components
  - Virtual Machines
  - Networking
  - Application Services
  - Cloud-based Services like AWS, Google Cloud Platform (GCP), IBM Cloud, Microsoft Azure
- Databases and Data Warehouses
  - RDBMS - IBM DB2, MySQL, Oracle Database, PostgreSQL
  - NoSQL - Redis, MongoDB, Cassandra, Neo4J
  - Data Warehouses - Oracle Exadata, IBM DB2 Warehouse on cloud, IBM Netezza Performance Server, Amazon Redshift
- Data Pipelines
  - Apache Beam
  - Airflow
  - DataFlow
- ETL Tools
  - IBM Infosphere Information Server
  - AWS Glue
  - Improvado
- Proficiency in languages for querying, manipulating, and processing data
  - SQL for relational databases, SQL-like query languages for NoSQL databases
  - Python, R, and Java
  - Shell and Scripting languages such as Unix/Linux Shell and PowerShell
- Big Data processing tools
  - Hadoop
  - Hive
  - Spark

Role of a Data Engineer
The goal of Data Engineering is to make quality data available for analytics and decision-making. And it does this by collecting raw source data, processing data so it becomes usable, storing data, and making quality data available to users securely.

The role of a Data Engineer includes:

- Gathering data from disparate sources.
- Integrating data into a unified view for data consumers.

- Preparing data for analytics and reporting.
- Managing data pipelines for a continuous flow of data from source to destination systems.
- Managing the complete infrastructure for the collection, processing, and storage of data.

To be successful in their role, Data Engineers need a mix of technical, functional, and soft skills.

- Technical Skills include working with different operating systems and infrastructure components such as virtual machines, networks, and application services. It also includes working with databases and data warehouses, data pipelines, ETL tools, big data processing tools, and languages for querying, manipulating, and processing data.
- An understanding of the potential application of data in business is an important skill for a data engineer. Other functional skills include the ability to convert business requirements into technical specifications, an understanding of the software development lifecycle, and the areas of data quality, privacy, security, and governance.
- Soft Skills include interpersonal skills, the ability to work collaboratively, teamwork, and effective communication.

Data Ecosystems

A Data Engineer's ecosystem includes the infrastructure, tools, frameworks, and processes for extracting data, architecting and managing data pipelines and data repositories, managing workflows, developing applications, and managing BI and Reporting tools.

There are two main types of data repositories - Transactional and analytical.

- **Transactional systems, also known as Online Transaction Processing (or OLTP) systems**, are designed to store high-volume day-to-day operational data. Such as online banking transactions, ATM transactions, and airline bookings. While OLTP databases are typically relational, they can also be non-relational.
- **Analytical systems, also known as Online Analytical Processing (OLAP) systems**, are optimized for conducting complex data analytics. These include relational and non-relational databases, data warehouses, data marts, data lakes, and big data stores. The type, format, sources of data, and context of use influence which data repository is ideal.

Based on how well-defined the structure of the data is, data can be categorized as

- Structured data, that is data which is well organized in formats that can be stored in databases.
  - SQL Databases
  - Online Transaction Processing
  - Spreadsheet
  - Online forms
  - Sensors GPS and RFID
  - Network and Web server logs
- Semi-structured data, that is data which is partially organized and partially free-form.
  - E-mails
  - XML and other markup language
  - Binary executables
  - TCP/IP packets
  - Zipped files
  - Integration of data
- Unstructured data, that is data which can not be organized conventionally into rows and columns.
  - Web pages
  - Social media feeds
  - Images in varied file formats
  - Video and Audio files
  - Documents and PDF files
  - Powerpoint presentation

  - Media logs
  - Surveys

Data comes in a wide-ranging variety of file formats, such as, delimited text files, spreadsheets, XML, PDF, and JSON, each with its own list of benefits and limitations of use.

Data is extracted from multiple data sources, ranging from relational and non-relational databases, to APIs, web services, data streams, social platforms, and sensor devices.

Once the data is identified and gathered from different sources, it needs to be staged in a data repository so that it can be prepared for analysis. The type, format, and sources of data influence the type of data repository that can be used.

Data professionals need a host of languages that can help them extract, prepare, and analyse data. These can be classified as:

- Querying languages, such as SQL, used for accessing and manipulating data from databases.
- Programming languages such as Python, R, and Java, for developing applications and controlling application behavior.
- Shell and Scripting languages, such as Unix/Linux Shell, and PowerShell, for automating repetitive operational tasks.

Data Repository

A **Data Repository** is a general term that refers to data that has been collected, organized, and isolated so that it can be used for reporting, analytics, and also for archival purposes.

The different types of Data Repositories include:

- Databases, which can be relational or non-relational, each following a set of organizational principles, the types of data they can store, and the tools that can be used to query, organize, and retrieve data.
- Data Warehouses, that consolidate incoming data into one comprehensive store house.
- Data Marts, that are essentially sub-sections of a data warehouse, built to isolate data for a particular business function or use case.
- Data Lakes, that serve as storage repositories for large amounts of structured, semi-structured, and unstructured data in their native format.
- Big Data Stores, that provide distributed computational and storage infrastructure to store, scale, and process very large data sets.

The ETL, or Extract Transform and Load, Process is an automated process that converts raw data into analysis-ready data by:

- Extracting data from source locations.
- Transforming raw data by cleaning, enriching, standardizing, and validating it.
- Loading the processed data into a destination system or data repository.

The ELT, or Extract Load and Transfer, Process is a variation of the ETL Process. In this process, extracted data is loaded into the target system before the transformations are applied. This process is ideal for Data Lakes and working with Big Data.

Data Pipeline, sometimes used interchangeably with ETL and ELT, encompasses the entire journey of moving data from its source to a destination data lake or application, using the ETL or ELT process.

Data Integration Platforms combine disparate sources of data, physically or logically, to provide a unified view of the data for analytics purposes.

Big Data Platforms

Big Data refers to the vast amounts of data that is being produced each moment of every day, by people, tools, and machines. The sheer velocity, volume, and variety of data challenged the tools and systems used for conventional data, leading to the emergence of processing tools and platforms designed specifically for Big Data.

V's of Big Data

- Velocity - speed at which data accumulates. Data is being generated extremely fast, in a process that never stops. Near or real-time streaming, local, and cloud-based technologies can process information very quickly.
- Volume - scale of the data, or the increase in the amount of data stored.
- Variety - the diversity of the data. Structured data fits neatly into rows and columns, in relational databases while unstructured data is not organized in a pre-defined way, like Tweets, blog posts, pictures, numbers, and video. Variety also reflects that data comes from different sources, machines, people, and processes, both internal and external to organizations. Drivers are mobile technologies, social media, wearable technologies, geo technologies, video, and many, many more.
- Veracity - the quality and origin of data, and its conformity to facts and accuracy. Attributes include consistency, completeness, integrity, and ambiguity.
- Value - our ability and need to turn data into value.

Big Data processing technologies help derive value from big data. These include NoSQL databases and Data Lakes and open-source technologies such as Apache Hadoop, Apache Hive, and Apache Spark.

- Apache Hadoop provides distributed storage and processing of large datasets across clusters of computers. One of its main components, the Hadoop File Distribution System, or HDFS, is a storage system for big data.
- Apache Hive is a data warehouse software for reading, writing, and managing large datasets files that are stored directly in either HDFS or other data storage systems such as Apache HBase.
- Apache Spark is a general-purpose data processing engine designed to extract and process large volumes of data. It is used to perform complex analytics in real-time.

Data Platform Layer

The architecture of a data platform can be seen as a set of layers, or functional components, each one performing a set of specific tasks. These layers include:

- **Data Ingestion Layer:** Responsible for connecting to source systems and bringing data into the platform. It transfers data in streaming or batch modes and maintains metadata about the collected data.
- **Data Storage and Integration Layer:** Stores data for processing, transforms and merges data logically or physically, and makes data available for processing in streaming and batch modes. Relational and non-relational databases are commonly used in this layer.
- **Data Processing Layer:** Processes data by performing validations, transformations, and applying business logic. It reads data from storage, supports querying tools and programming languages, and allows analysts and data scientists to work with the data.
- **Analysis and User Interface Layer:** Delivers processed data to various users, including business intelligence analysts, data scientists, and other applications. It supports querying tools, programming languages, APIs, and visualization tools like dashboards and business intelligence applications.
- **Data Pipeline Layer:** Overlaying other layers, it implements and maintains a continuously flowing data pipeline. It uses Extract, Transform, and Load (ETL) tools such as Apache Airflow and DataFlow to ensure a smooth data flow across the platform.

A well-designed data repository is essential for building a system that is scalable and capable of performing during high workloads.

The choice or design of a data store is influenced by the type and volume of data that needs to be stored, the intended use of data, and storage considerations. The privacy, security, and governance needs of your organization also influence this choice.

The CIA, or Confidentiality, Integrity, and Availability triad are three key components of an effective strategy for information security. The CIA triad is applicable to all facets of security, be it infrastructure, network, application, or data security.

Data Collection and Wrangling

Depending on where the data must be sourced from, there are a number of methods and tools available for gathering data. These include query languages for extracting data from databases, APIs, Web Scraping, Data Streams, RSS Feeds, and Data Exchanges.

Once the data you need has been gathered and imported, your next step is to make it analytics-ready. This is where the process of Data Wrangling, or Data Munging, comes in.

Data Wrangling involves a whole range of transformations and cleansing activities performed on the data. Transformation of raw data includes the tasks you undertake to:

- Structurally manipulate and combine data using Joins and Unions.
- Normalize data, that is, clean the database of unused and redundant data.
- Denormalize data, that is, combine data from multiple tables into a single table so that it can be queried faster.

Cleansing activities include:

- Profiling data to uncover anomalies and quality issues.
- Visualizing data using statistical methods in order to spot outliers.
- Fixing issues such as missing values, duplicate data, irrelevant data, inconsistent formats, syntax errors, and outliers.

A variety of software and tools are available for the data wrangling process. Some of the popularly used ones include Excel Power Query, Spreadsheets, OpenRefine, Google DataPrep, Watson Studio Refinery, Trifacta Wrangler, Python, and R, each with their own set of features, strengths, limitations, and applications.

Querying Data, Performance Tuning and Troubleshooting

- In order for raw data to become analytics-ready, a number of transformation and cleansing tasks need to be performed on raw data. And that requires you to understand your dataset from multiple perspectives. One of the ways in which you can explore your dataset is to query it.
- Basic querying techniques can help you explore your data, such as, counting and aggregating a dataset, identifying extreme values, slicing data, sorting data, filtering patterns, and grouping data.
- In a data engineering lifecycle, the performance of data pipelines, platforms, databases, applications, tools, queries, and scheduled jobs, need to be constantly monitored for performance and availability.
- The performance of a data pipeline can get impacted if the workload increases significantly, or there are application failures, or a scheduled job does not work as expected, or some of the tools in the pipeline run into compatibility issues.
- Databases are susceptible to outages, capacity overutilization, application slowdown, and conflicting activities and queries being executed simultaneously.
- Monitoring and alerting systems collect quantitative data in real time to give visibility into the performance of data pipelines, platforms, databases, applications, tools, queries, scheduled jobs, and more.
- Time-based and condition-based maintenance schedules generate data that helps identify systems and procedures responsible for faults and low availability.

Career Opportunities in Data Engineering

Data Engineering is reported to be one of the top ten jobs experiencing tremendous growth in the U.S. today. It is also reported to be one of the fastest growing tech occupations with year-over-year growth of around 50

Currently, the demand for skilled data engineers far outweighs the supply, which means companies are willing to pay a premium to hire skilled data engineers.

Data engineering roles in organizations tend to break the specialization up into Data Architecture, Database Design and Architecture, Data Platforms, Data Pipelines and ETL, Data Warehouses, and Big Data.

Regardless of the niche you choose to specialize in, knowledge of operating systems, languages, databases, and infrastructure components, is essential.

To work your way up from a Junior Data Engineer to a Lead or Principal Data Engineer, you need to continually advance your technical, functional, and soft skills from a foundational level to an expert level. You need to not only expand your skills in your niche area, but also into other areas of data engineering at the same time.

Big Data Engineers and Machine Learning Engineers are some of the emerging roles in this field and they require specialized skills in addition to basic data engineering.

There are several paths you can consider in order to gain entry into the data engineering field.

- An academic degree in Computer Science or engineering qualifies you for an entry-level job.
- If you are not a graduate, or a graduate in a non-relate stream, you can earn professional certifications from online multi-course specializations offered by learning platforms such as Coursera, edX, and Udacity.
- If you have a coding background, or you are an IT Support Specialist, a Software Tester, a Programmer, or a data professional such as a Statistician, Data Analyst, or BI Analyst, you can upskill with the help of online courses to become a Data Engineer.