**Week 1: -**

* Overview of Data Engineering
* ETL vs ELT
* Batch vs Streaming
* Data Pipeline Components
* Common Tools in the industry

1. **Overview of Data Engineering Field: -**

* **Data Engineering** is the discipline of designing, building, and maintaining systems that allow for the collection, storage, processing, and access of data at scale. It enables organizations to use data for analytics, decision-making, and machine learning.
* **Key Goals:**
* Build data pipelines to ingest, clean, and transform raw data into usable formats.
* Ensure data availability, quality, and reliability.
* Support data analysts, data scientists, and business users with scalable infrastructure.
* **Data Engineers handle:**
  + **Data ingestion** (from APIs, files, databases, etc.)
  + **Data transformation** (cleaning, normalization, enrichment)
  + **Data storage** (data warehouses, lakes, etc.)
  + **Workflow orchestration** (scheduling & monitoring jobs)
  + **Scalability & performance optimization**

1. **ETL vs ELT**

* **ETL – Extract, Transform, Load**
* Workflow:
* **Extract**: Pull data from source systems (e.g., databases, APIs).
* **Transform**: Clean, filter, format, and reshape the data using tools or scripts.
* **Load**: Push the **transformed data** into a target system (like a data warehouse).
  + Best for Traditional Warehouses and When data needs to be cleaned and modelled **before** storage.
  + Apache spark, apache nifi, informetica etc
* **ELT – Extract, Load, Transform**
* **Workflow**
  + **Extract:** Pull raw data from source systems (e.g., databases, APIs, logs).
  + **Load:** Load the raw data directly into the data warehouse or data lake.
  + **Transform:** Clean, join, and process the data **inside the warehouse** using SQL or transformation tools.
* **Best for Cloud Data Warehouses** like **Snowflake, BigQuery**, and **Redshift**, where you store large volumes of raw data and transform it later using the warehouse’s compute power.
* **Tools:** dbt (Data Build Tool), Snowflake SQL, BigQuery SQL, Azure Data Factory, Fivetran, Stitch

1. **Batch vs Streaming**

### **Batch Processing**

• Processing large volumes of data **at once** in scheduled intervals (e.g., hourly, daily, weekly).

• **Workflow:**

* Collect data over a period (e.g., a day).
* Process it in **batches** (e.g., run ETL jobs at midnight).
* Store results for reporting or analytics.

• **Best for:**  
Periodic reporting, historical trend analysis, data warehouse updates.

• **Tools:**  
Apache Spark (batch mode), Apache Hive, AWS Glue, Apache Airflow, SQL-based ETL jobs.

• **Example:**  
Run a job every night to calculate daily sales totals and load them into a data warehouse.

### • **Streaming Processing**

• **Definition:**  
Processing data **in real-time or near real-time** as it is generated or received.

• **Workflow:**

* Continuously capture incoming data (e.g., user clicks, sensor data).
* Process each event/message immediately.
* Push results to storage, dashboards, or trigger alerts.

• **Best for:**  
Real-time analytics, fraud detection, live dashboards, recommendation systems.

• **Tools:**  
Apache Kafka, Apache Flink, Apache Spark Streaming, AWS Kinesis, Google Dataflow.

• **Example:**  
Detect suspicious login activity instantly and alert the security team in real-time.

### **Key Differences: -**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Batch** | **Streaming** |
| **Processing** | Periodic, scheduled | Continuous, real-time |
| **Latency** | Minutes to hours | Seconds to milliseconds |
| **Data Volume** | Large chunks | Continuous flow |
| **Use Case** | Reports, historical insights | Alerts, live analytics |

1. Data Pipelines Components: -

* A data pipeline refers to the end-to-end architecture that moves and processes data from various sources to a destination system like a data warehouse or data lake, making it ready for analytics, reporting, or machine learning.

#### 1. Data Sources

* This is where the data originates. It can include transactional databases (like MySQL, PostgreSQL), APIs (e.g., REST, GraphQL), flat files (CSV, JSON), log files, mobile apps, sensors (IoT), or SaaS platforms (like Salesforce or Shopify).

#### 2. Ingestion Layer

* This layer is responsible for bringing raw data into the pipeline. Ingestion can be real-time (streaming) or periodic (batch). Tools like Apache Kafka, Apache NiFi, AWS Kinesis, Fivetran, and Stitch are commonly used to collect and ingest data from diverse sources into staging areas or directly into data lakes/warehouses.

#### 3. Storage Layer

Once ingested, the data needs to be stored before or after processing.

* **Data Lakes** (e.g., S3, HDFS, Azure Data Lake) store raw, semi-structured, or unstructured data.
* **Data Warehouses** (e.g., Snowflake, BigQuery, Redshift) store cleaned, structured, and analytical data that supports fast querying.

#### 4. Processing Layer

* This is where raw data is cleaned, transformed, joined, and aggregated. Processing can be done in batch or in real-time depending on the use case. Batch tools include Apache Spark and AWS Glue, while real-time tools include Apache Flink, Kafka Streams, and Spark Streaming.

#### 5. Orchestration Layer

* This layer manages workflow execution and dependencies. It schedules tasks, handles retries on failure, and ensures proper sequencing of jobs. Common orchestration tools include Apache Airflow, Prefect, and Dagster.

#### 6. Transformation and Modeling Layer

* In this step, data is further refined into analytics-ready formats. This includes applying business rules, normalization, joining datasets, and creating data models (like star or snowflake schemas). Tools like dbt, SQL, and PySpark are widely used here.

#### 7. Serving Layer

* This is the final destination of the processed data where it is made accessible to users and systems. The data can be served to BI tools (like Tableau, Power BI, Looker), dashboards, data science models, or external APIs.

#### 8. Monitoring & Logging Layer

* This component ensures that the pipeline is reliable, observable, and error-free. It includes monitoring job statuses, data quality checks, latency tracking, and alerts for failures or anomalies. Tools include Great Expectations, Monte Carlo, and DataDog.

1. **Tools Used in Industry**

#### Data Ingestion:

* **Apache Kafka** – Event streaming platform
* **Apache NiFi** – Flow-based programming
* **AWS Glue** – Serverless ingestion & transformation

#### Storage:

* **Amazon S3**, **Google Cloud Storage** – Object storage (for Data Lakes)
* **Snowflake**, **BigQuery**, **Redshift** – Cloud Data Warehouses

#### Data Processing:

* **Apache Spark** – Scalable data processing engine (batch/stream)
* **Apache Flink** – Real-time streaming engine
* **dbt (Data Build Tool)** – SQL-based transformation framework

#### Workflow Orchestration:

* **Apache Airflow** – DAG-based scheduler
* **Prefect** – Pythonic modern orchestration
* **Dagster** – Type-safe, modern data orchestrator

#### Data Quality:

* **Great Expectations** – Data validation framework
* **Monte Carlo**, **Soda.io** – Data observability platforms

#### BI & Visualization:

* **Tableau**, **Power BI**, **Looker** – Data dashboards and reporting tools

#### Infrastructure & Deployment:

* **Docker**, **Kubernetes** – Containerization & orchestration
* **Terraform**, **Pulumi** – Infrastructure as Code (IaC)