Chapter 1: Introduction to Data Engineering & Data Lake (Week 1)

Welcome, and thank you for joining DSI310: Data Exploration and Preprocessing. Throughout this course, we will embark on a comprehensive journey that bridges the gap between raw information and actionable business insights. Our primary goal is to empower you with the skills to design, build, and maintain the data systems that form the backbone of a data-driven enterprise.

Today, we lay the foundational groundwork by exploring the core principles of data engineering and the architectural paradigms that define a modern data platform.

1. The Strategic Role of a Data Engineer

The modern data ecosystem is complex and dynamic. It is no longer sufficient to simply collect data; data must be organized, processed, and made accessible in a reliable, scalable, and secure manner. This is the central mission of the data engineer.[1] A data engineer is the architect and builder of the data infrastructure that transforms raw data into a valuable, analytical-ready asset.[1]

The DSI310 curriculum is designed around the full data lifecycle.[1] From discovering initial data sources and creating a formal data catalog to building a continuous data pipeline, a data engineer oversees the entire journey of data, ensuring its quality and utility.[1] This process is what allows a business to move "From Data to Decisions".[2]

- **Defining the Role and Responsibilities:** Data engineers are responsible for designing, constructing, and maintaining architectures, including databases and systems for large-scale processing.[4] Their work includes the development and maintenance of data management systems.[4] They are the critical link between various source systems (e.g., transactional databases, APIs, log files) and the end-users who consume data (e.g., business analysts, data scientists, machine learning engineers).[5]
- **Essential Skills for Success:** The role demands a diverse skill set, blending technical expertise with a strong business acumen.[5]
  - **Technical Skills:** Proficiency in programming languages like Python or Java is crucial for automating processes and building data pipelines.[4] A deep understanding of both relational (SQL) and non-relational (NoSQL) databases is essential.[4] Data engineers must also be familiar with big data technologies like Apache Hadoop and Apache Spark for distributed systems, as well as cloud services from providers like Amazon

    Web Services (AWS) or Azure.[4]
  - **Non-Technical Skills:** Effective communication and collaboration are vital.[4] A data engineer must be able to understand the goals and needs of both technical and non-technical colleagues and then translate complex technical concepts into clear, understandable terms for business stakeholders.[4]
- **Enabling Intelligence:** The work of a data engineer directly supports the application of advanced technologies like Artificial Intelligence (AI) and Large Language Models (LLMs).[2] Without a clean, well-structured, and continuously updated data foundation, these sophisticated models cannot function effectively.[2] The course materials highlight how a real-time data platform is essential for powering real-time analytics and intelligent applications.[2]

2. Understanding the Modern Data Platform Architecture

A modern data platform is a unified, end-to-end ecosystem for managing all of an organization's data assets.[2] While the specific components can vary, a common and highly effective architectural pattern is the data lake, which is often segmented into distinct zones based on the maturity and usability of the data.

2.1 The Data Lake Paradigm

A data lake is a centralized repository that allows you to store all of your structured and unstructured data at any scale.[2] Unlike a traditional data warehouse, which requires a pre-defined schema, a data lake stores data in its raw, native format.[3] This flexibility is a significant advantage, but it also necessitates a disciplined approach to organization to prevent the data lake from becoming a "data swamp." This is where the concept of "Data Lake Zones" becomes critical.[2] The data lake serves as a foundation and landing zone for raw data, ensuring that no information is lost by prematurely transforming or aggregating it.[6]

2.2 The Layered Architecture: Data Lake Zones

The DSI310 course introduces a three-tiered data lake architecture, a proven methodology for managing the data pipeline from ingestion to analysis. Each zone serves a unique purpose in refining data quality and preparing it for consumption.[1]

- **Landing Zone (Raw Zone):**
  - **Purpose:** This is the initial entry point for all data.[1] Data is ingested "as-is" from its source systems (e.g., relational databases, APIs, IoT sensors) and stored without any

transformations.[6]

- **Key Characteristics:**
  - **Immutability:** The raw data in this zone is never changed.[1] This provides a complete historical record and allows for reprocessing if a new business requirement or a data quality issue is discovered downstream.[6] The only exception is the removal of personally identifiable information (PII) to ensure privacy and regulatory compliance before the data is saved.[6]
  - **Schema-on-Read:** The schema is not enforced at the time of writing; instead, the data's structure is inferred when it is read by a downstream process.[1]
- **Example:** A daily feed of sales transactions from the OmniCorp Chinook and Northwind databases would be stored in the Landing Zone in their original format, with no modifications to the column names or data types.[3]

- **Staging Zone (Cleansed Zone):**
  - **Purpose:** The Staging Zone is where the initial data cleansing and preparation takes place.[1] Data is read from the Landing Zone, and preliminary transformations are applied to address data quality issues.[1]
  - **Key Characteristics:**
    - **Data Cleansing:** Common tasks include handling missing values, removing duplicates, and correcting inconsistent data types.[1] This is where data profiling is performed to understand the data's characteristics and potential anomalies.[1]
    - **Standardization:** Data is standardized into a consistent format. For instance, dates are converted to a standard format (e.g., YYYY-MM-DD), and text fields are cleaned.[1]
  - **Example:** In the OmniCorp project, the Staging Zone would be used to address the different CustomerID formats (integer in Chinook vs. text in Northwind) and to concatenate FirstName and LastName into a single CustomerName.[3]
- **Integration Zone (Curated/Analytics Zone):**
  - **Purpose:** The Integration Zone is the final destination for data before it is made available for analysis.[1] This is where data from different sources is combined and transformed into a structured format that is optimized for analytical querying.[1] This zone often takes the form of a data warehouse or data mart.[2]
  - **Key Characteristics:**
    - **Aggregation and Denormalization:** This zone often involves complex transformations, such as aggregating data and creating denormalized tables (e.g., star schemas) to improve query performance.[1]
    - **Business Logic:** Business rules and logic are applied here.[1] For example, calculating a new metric like
SalesAmount from UnitPrice and Quantity.[3]
    - **Optimized for Performance:** Data in this zone is typically stored in a columnar format (e.g., Parquet), which is highly efficient for analytical queries.[1]
  - **Example:** The OmniCorp project’s final output—the unified FactSales table and the

Dim tables—would reside in the Integration Zone, ready to be consumed by business intelligence tools.[3]

3. ETL vs. ELT: A Fundamental Choice in Data Integration

The movement of data through the data platform is managed by a process known as ETL (Extract, Transform, Load) or its modern alternative, ELT (Extract, Load, Transform).[1] While both processes aim to move data from a source to a destination, they differ fundamentally in the sequence and location of the transformation step.

Aspect	ETL (Extract, Transform, Load)	ELT (Extract, Load, Transform)
Process Flow	Data is extracted, transformed on a separate processing server, and then loaded into the target system.[7]	Data is extracted, loaded directly into the target system in its raw format, and then transformed within the destination.[7]
Transformation Location	A separate, intermediate staging server is used to perform transformations.[7]	Transformations are performed directly within the target data warehouse.[7]
Data Types	Best suited for structured data that can be represented in tables with rows and columns.[8]	Handles all types of data, including structured, semi-structured, and unstructured data like images or documents.[8]
Speed & Scalability	Slower, as the transformation step is a bottleneck that is difficult to scale as data volume increases.[7]	Faster, as data is loaded directly into the target and transformed in parallel. Leverages the scalable processing power of modern cloud data warehouses.[7]
Cost	Can be more costly due to the need for a separate server and the initial time-intensive setup	Often more cost-efficient as all transformations occur within a single system, reducing

Aspect	ETL (Extract, Transform, Load)	ELT (Extract, Load, Transform)
	to define data structures.[8]	maintenance and setup overhead.[8]

The DSI310 curriculum is structured to teach a practical, methodical pipeline that aligns with the principles of both ETL and ELT.[1] We will focus on the step-by-step process of managing data in the Landing Zone, performing cleansing and transformations in the Staging Zone, and finally loading aggregated data into the Integration Zone.[1] This approach provides a robust and methodical understanding of data quality and transformation.[1]

4. Real-World Applications of Data Platforms

The concepts we discuss today are not abstract theories; they are the backbone of real-world solutions that drive business value across diverse industries.

4.1 Smart Manufacturing and Predictive Maintenance

Data platforms are transforming smart manufacturing by turning raw sensor data from machines into "actionable insights".[2]

- **Predictive Maintenance:** By applying AI to analyze machine performance and sensor data, companies can forecast potential equipment failure and perform proactive maintenance, thereby minimizing costly downtime.[9] This strategy can cut unplanned downtime by up to 50% and slash maintenance costs by 10-40%.[9]
- **How it works:** IoT devices with sensors continuously monitor metrics like temperature, pressure, and vibration levels in real time.[9] This data is streamed to a central system where machine learning algorithms analyze it to detect subtle changes or anomalies that may signal wear and tear.[9]
- **Examples in Action:** A cloud-based data platform allowed an automobile manufacturer to analyze its global fleet performance without significant infrastructure costs.[9] In the manufacturing sector, data platforms help identify production bottlenecks and allow for real-time adjustments to optimize inventory and efficiency.[10] The **Hitachi Lumada** platform is a prime example of this, using data and AI to optimize industrial operations.[12]

4.2 Data Platforms in Other Industries

The power of a robust data platform extends far beyond manufacturing.

- **Healthcare and Life Sciences:** In healthcare, data analytics tools can be used to detect fraudulent billing patterns, such as "phantom bills" or "upcoding".[14] They also play a crucial role in patient data security by monitoring network traffic and assigning real-time risk scores to transactions to prevent cyberattacks.[14] These platforms enable the creation of a "patient 360" view by unifying fragmented data from various sources like clinical, claims, and consumer data.[15]
- **Finance:** Data platforms are critical for fraud detection and risk management.[14] Real-time analytics can be used to detect stock market manipulation, for example, by using technologies like Generative Adversarial Networks (GANs) to distinguish between real and manipulated stock prices.[10]
- **International Trade:** The course materials highlight a successful project with the Department of International Trade Promotion (DITP).[2] A data warehouse was built to support small and medium-sized exporters, enabling the tracking of trade achievements, assessment of SMEs, and the provision of targeted trade recommendations.[2]

5. The DSI310 Course Project: OmniCorp's Data Unification Challenge

The central project for this course, "Data Lake Modeling," is a hands-on simulation of a real-world data engineering challenge.[3] We will be working with a fictional company, OmniCorp, which has acquired two businesses: a music store (Chinook) and a food and beverage distributor (Northwind).[3]

The core challenge is to take the data from these two disparate legacy systems and unify them into a single, cohesive data warehouse.[3] This is not merely a technical exercise; it's a strategic imperative to enable OmniCorp to perform "cross-business analytics" and make informed, data-driven decisions.[3]

The project is structured around four key tasks [3]:

- **Task 1: Data Understanding & Source-to-Target Mapping:** You will conduct an Exploratory Data Analysis (EDA) on the two source databases to identify common entities like customers, employees, and products.[3] The goal is to create a formal mapping document that serves as a blueprint for data integration.[3]
- **Task 2: Dimensional Modeling & Schema Design:** You will design a **Kimball Star Schema** to unify the sales data from both sources into a central FactSales table and several dimension tables (DimCustomer, DimProduct, DimEmployee, DimTime, DimSourceSystem).[3] This task requires careful justification for how you will handle

different data types and naming conventions.[3]

- **Task 3: Data Expression & Business Value:** This task focuses on demonstrating the business value of your data model.[3] You will express what a single row in your fact table represents in a simple, business-friendly sentence and mock up a report to show how the schema supports key business questions.[3]
- **Task 4: Critical Thinking & Data Engineering Challenges:** You will analyze and discuss the real-world challenges of data integration, such as managing primary key conflicts between the two source systems.[3] You will also explain the distinction between a data lake and a data warehouse and how the star schema's design allows for future growth.[3]

6. From Data Models to Knowledge Graphs: A Deeper Connection

The principles of dimensional modeling are not just practical; they are also grounded in a formal theoretical framework for knowledge representation. The DSI310 course materials draw a direct parallel between the components of a star schema and the ontological concepts of a knowledge graph.[16]

- **The Terminological Box (T-Box):** The T-Box represents the schema, concepts, and hierarchical structure of permanent knowledge.[16] This aligns perfectly with the function of
  dimension tables.[16] A
  DimCustomer table, for example, contains the permanent, descriptive attributes of customers that provide context to transactions.[3] The T-Box is a master list of entities and their properties, mirroring the role of a dimension table.[16]
- **The Assertion Box (A-Box):** The A-Box represents the transactional, factual data, or "events".[16] This is the conceptual equivalent of a
  fact table.[16] A row in
  FactSales is a single, specific event—a sales line item—that occurred at a particular time, to a specific customer, and for a particular product.[3] The A-Box records these individual facts and the high degree of relationships between them, which is precisely the purpose of a fact table.[16]

This conceptual alignment demonstrates that a star schema is not just a performance hack for databases; it is a logically sound and robust method for representing business knowledge. By separating the static, descriptive master data (dimensions/T-Box) from the dynamic, event-driven transactional data (facts/A-Box), the model provides a clear, powerful, and theoretically consistent framework for analysis.

7. Key Theories and Keywords

To conclude our first lecture, let's establish a common vocabulary that we will use throughout the course. A strong understanding of these core concepts is essential for success.

- **Data Platform:** A centralized, end-to-end system that encompasses the infrastructure, tools, and processes for the collection, processing, and management of data to support business intelligence and analytics.[2]
- **Data Engineering:** The professional discipline focused on the design, construction, and maintenance of the data platforms and pipelines that enable organizations to turn data into strategic insights.[5]
- **Data Lake:** A massive, centralized repository that holds all forms of data in its native format, regardless of its structure.[2] It is a key component of a modern data platform.[2]
- **Landing Zone:** The first layer of a data lake, where raw, immutable data is stored exactly as it was ingested from the source system.[6]
- **Staging Zone:** The second layer of a data lake, where preliminary data cleansing, standardization, and quality checks are performed to prepare the data for further transformations.[1]
- **Integration Zone:** The third layer of a data lake, where data from various sources is combined, aggregated, and transformed into a structured, analytics-ready format, such as a star schema.[1]
- **ETL (Extract, Transform, Load):** A traditional data integration process where data is first extracted from a source, then transformed in a separate staging area, and finally loaded into a target data store.[7]
- **ELT (Extract, Load, Transform):** A modern data integration process where data is first extracted and loaded into a target data store, with transformations occurring within the target system itself.[7]
- **Predictive Maintenance:** An AI-driven strategy that uses real-time data from machinery to predict potential failures, allowing for proactive maintenance and minimizing downtime.[9]
- **Dimensional Modeling:** A data design philosophy that organizes data into a central fact table and surrounding dimension tables to optimize for analytical queries.[3]
- **Star Schema:** A simple but highly effective dimensional model consisting of a central fact table surrounded by multiple dimension tables, resembling a star.[3]
- **Fact Table:** A central table in a star schema that contains the quantitative metrics, or "facts," of a business process (e.g., SalesAmount, SalesQuantity).[2] It is the conceptual equivalent of a Knowledge Graph's Assertion Box (A-Box).[16]
- **Dimension Table:** A table in a star schema that provides the descriptive context for the facts (the "who, what, where, when, and how").[2] It is the conceptual equivalent of a Knowledge Graph's Terminological Box (T-Box).[16]

Works cited

1. dsi310_แบบฟอร์มเค้าโครงการบรรยาย 2025
2. dsi310 week01: Data Engineering & Data Lake
3. dsi310: Data Lake Modeling
4. Learning Data Engineer Skills: Career Paths and Courses | Coursera, accessed September 10, 2025, https://www.coursera.org/articles/data-engineer-skills
5. What is a Data Engineer? - Splunk, accessed September 10, 2025, https://www.splunk.com/en_us/blog/learn/data-engineer-role-responsibilities.html
6. Data lake best practices | Databricks, accessed September 10, 2025, https://www.databricks.com/discover/data-lakes/best-practices
7. ETL vs ELT: Key Differences, Comparisons, & Use Cases - Rivery, accessed September 10, 2025, https://rivery.io/blog/etl-vs-elt/
8. ETL vs ELT - Difference Between Data-Processing Approaches - AWS, accessed September 10, 2025, https://aws.amazon.com/compare/the-difference-between-etl-and-elt/
9. Predictive Maintenance Case Studies: How Companies Are Saving Millions with AI-Powered Solutions - ProValet, accessed September 10, 2025, https://www.provalet.io/guides-posts/predictive-maintenance-case-studies
10. Real-Time Analytics Use Cases and Examples - Striim, accessed September 10, 2025, https://www.striim.com/blog/real-time-analytics-use-cases-and-examples/
11. 12 Key Manufacturing Analytics Use Cases - NetSuite, accessed September 10, 2025, https://www.netsuite.com/portal/resource/articles/erp/manufacturing-analytics-use-cases.shtml
12. Lumada: Hitachi Global, accessed September 10, 2025, https://www.hitachi.com/products/it/lumada/global/en/
13. List of Lumada Ready Products - Hitachi Global, accessed September 10, 2025, https://www.hitachi.com/products/it/lumada/global/en/about/lumada_ready/index.html
14. Top 10 Healthcare Analytics Use Cases with Examples - Research AIMultiple, accessed September 10, 2025, https://research.aimultiple.com/healthcare-analytics-examples/
15. AI Data Cloud for Healthcare & Life Sciences | Snowflake, accessed September 10, 2025, https://www.snowflake.com/en/solutions/industries/healthcare-and-life-sciences/
16. 2025 dsi310 week02