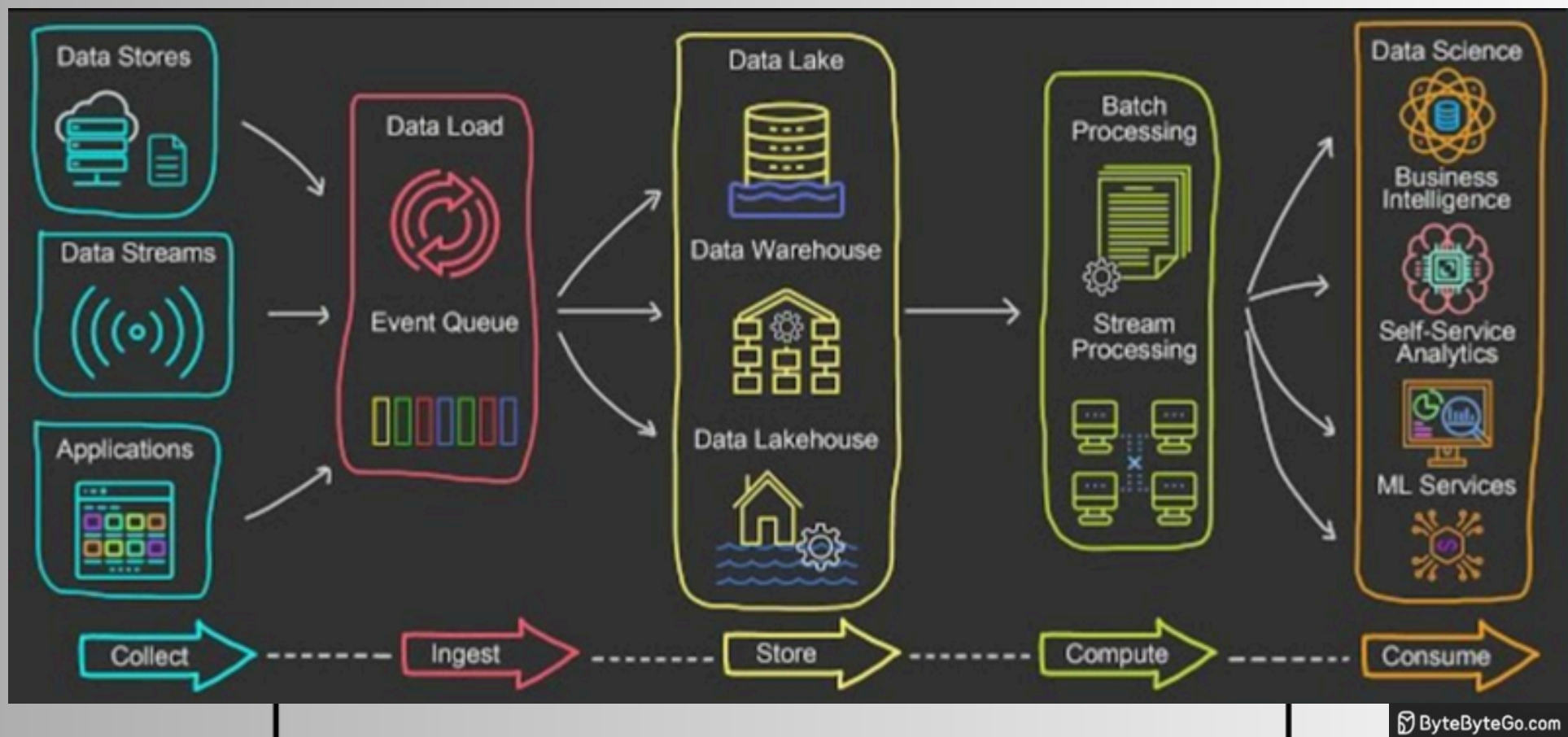


DATA PIPELINE



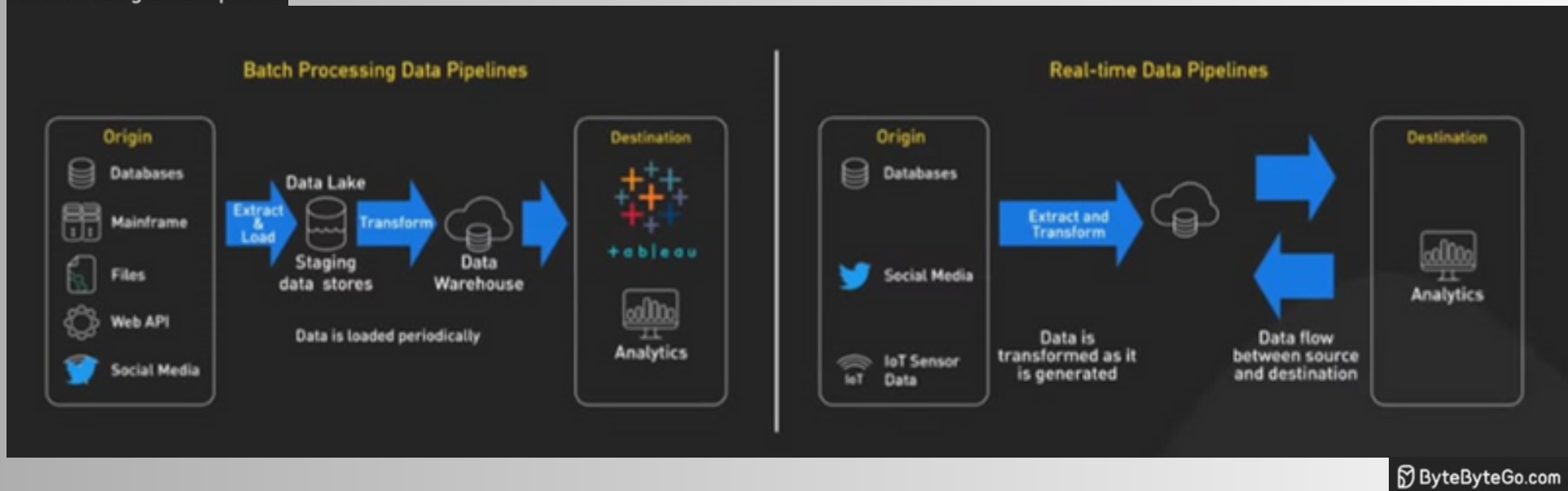
LET'S EXPLORE:

- What is a data pipeline?
- Why are data pipelines important?
- Why SQL for building data pipelines?
- What are the key stages of a data pipeline?

💡 What is a Data Pipeline?

A data pipeline is a series of steps that moves data from one system to another, often transforming it along the way. It's like an assembly line for data.

The Amazing Data Pipeline



🔧 Why are Data Pipelines important?

Because raw data is pretty messy!

Pipelines help:

- **Extract** data from different sources (databases, APIs, files)
- **Transform** it (clean, filter, enrich, deduplicate)
- **Load** it into a destination (like a data warehouse, dashboard, or report)

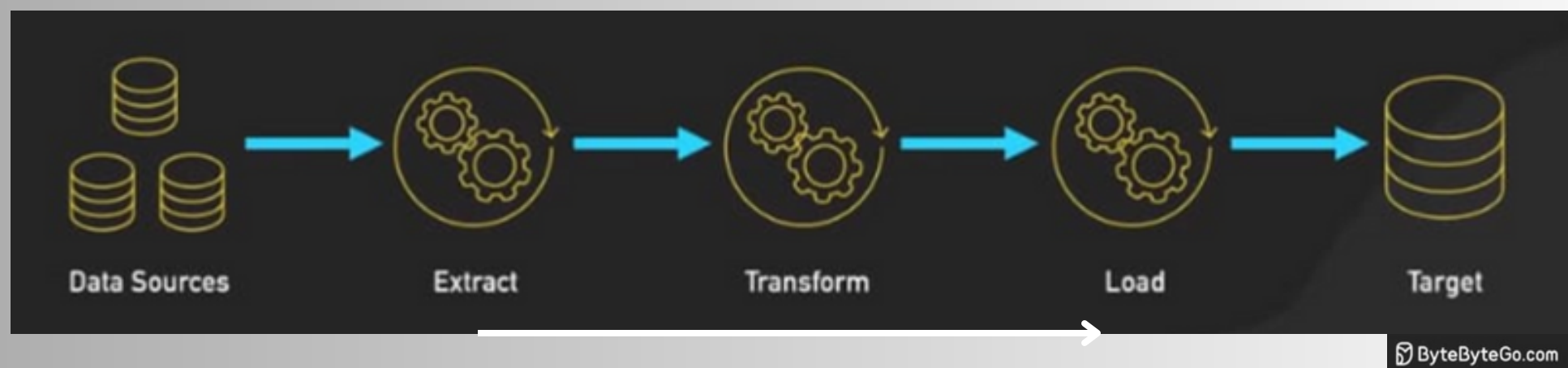
This is often called ETL (Extract, Transform, Load).

A data pipeline is the invisible engine that turns raw data into real-time decisions quickly, reliably, and at scale.

Why SQL for building Data Pipelines?

SQL is perfect for:

- Extracting data using **SELECT**
- Transforming with filters, joins, aggregations
- Loading into tables using **INSERT**, **MERGE**, **CREATE TABLE AS**

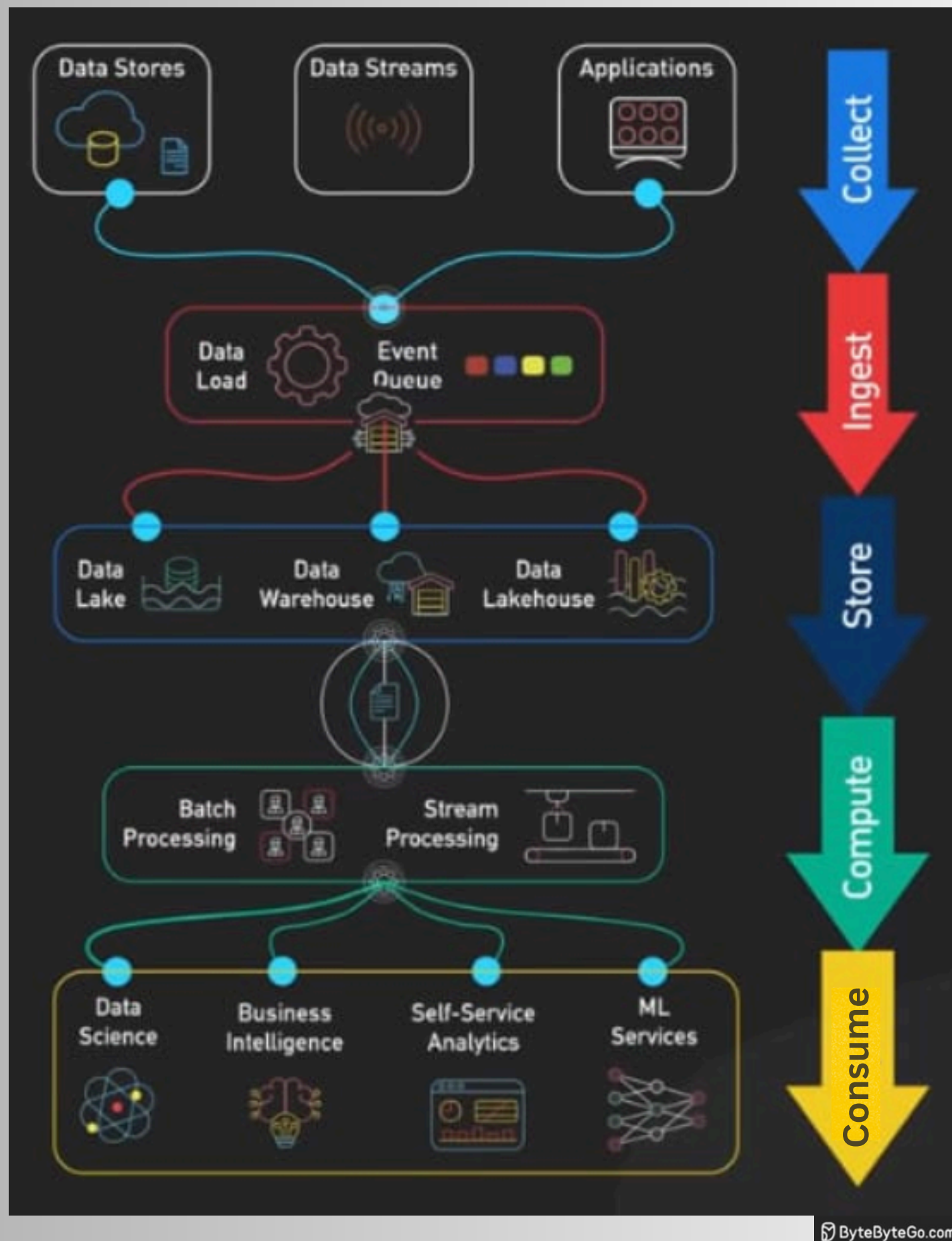


In many companies, SQL is the backbone of lightweight pipelines, especially in tools like:

- Airflow (with SQL operators)
- dbt (which is SQL-based!)
- Stored procedures or scheduled SQL jobs

SQL powers every stage of the data pipeline, from cleaning and transforming to aggregating and delivering insights that drive decisions.

What are the key stages of a Data Pipeline?



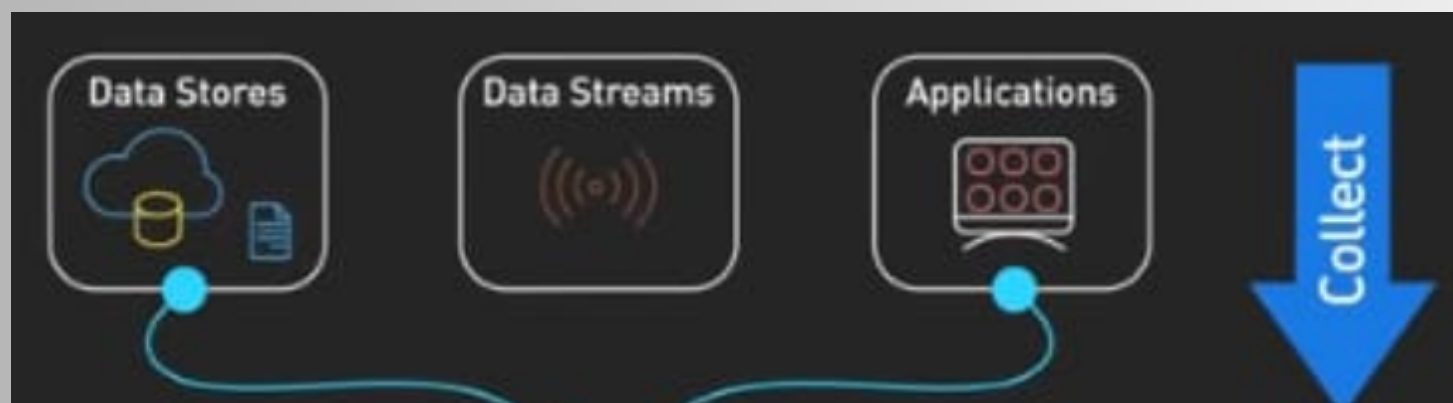
1. Collect: Where Data Begins

Before data can do anything, it needs to be captured.

This happens through:

- **Data Stores** like databases (MySQL, MongoDB) that log structured data
- **Data Streams** like Kafka or IoT devices sending real-time signals
- **Apps & Web Platforms** that generate user activity, payments, and events

Think of this as: “Where is the data coming from?”

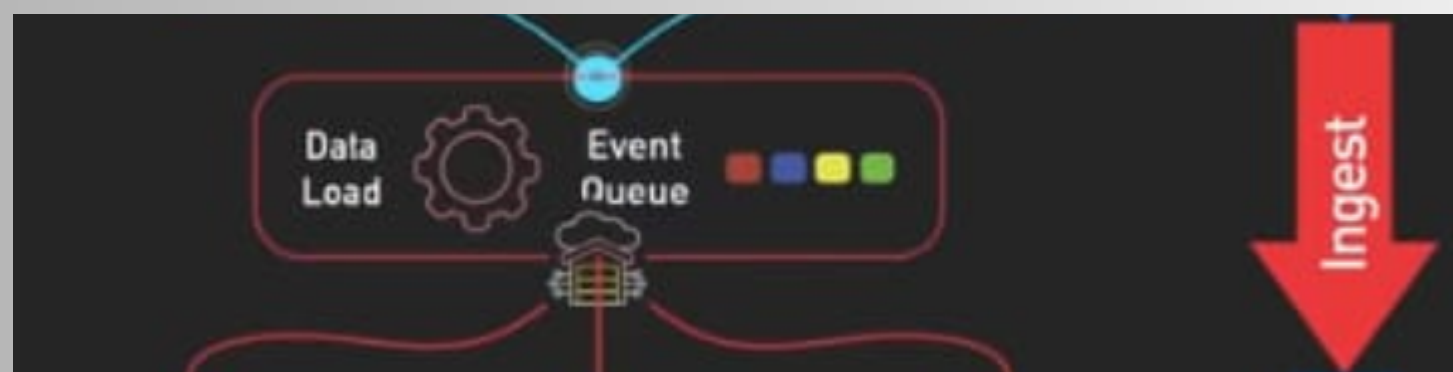


2. Ingest: Getting Data into the System

Once data is collected, it needs to be moved efficiently into your pipeline.

- **Data Load** brings in batches of data (e.g., a daily CSV from your CRM)
- **Event Queues** send data in real time; messages move through a system like Kafka, staying reliable and ordered

This step ensures data flows continuously, whether hourly or per second.

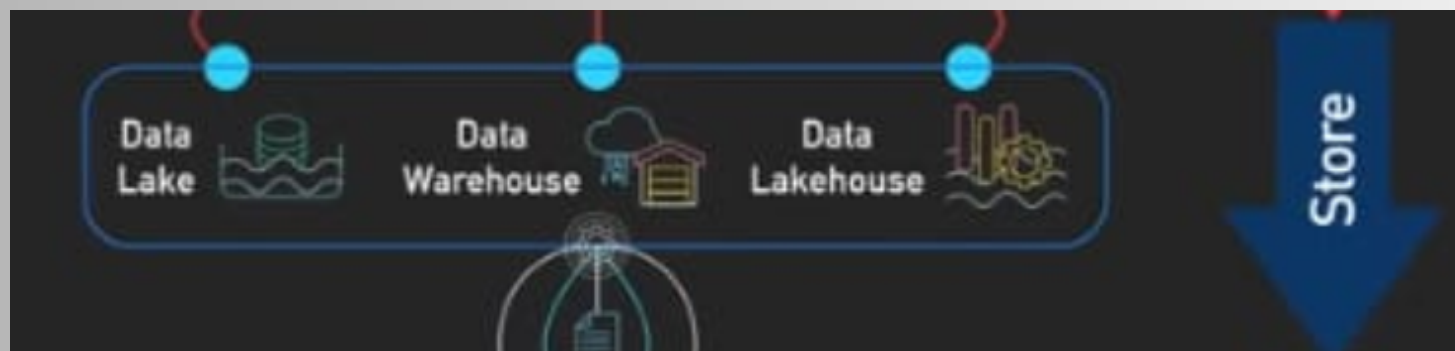


3. Store: Organizing the Raw Material

Where should this data live once it's in the system?

- **Data Lake:** Raw, flexible storage for logs, images, text - cheap and vast
- **Data Warehouse:** Structured, clean, analytics-ready - perfect for dashboards
- **Data Lakehouse:** Combines both, flexibility of lakes + speed of warehouses

The goal: make data findable, usable, and fast, depending on the use case.



4. Compute: Making Data Useful

This is where data gets cleaned, enriched, and modeled.

- **Batch Processing:** Large volumes processed on a schedule (e.g., every night with dbt or Spark)
- **Stream Processing:** Real-time insights for use cases like fraud detection or live dashboards

This is the “T” in ETL, where raw becomes ready.

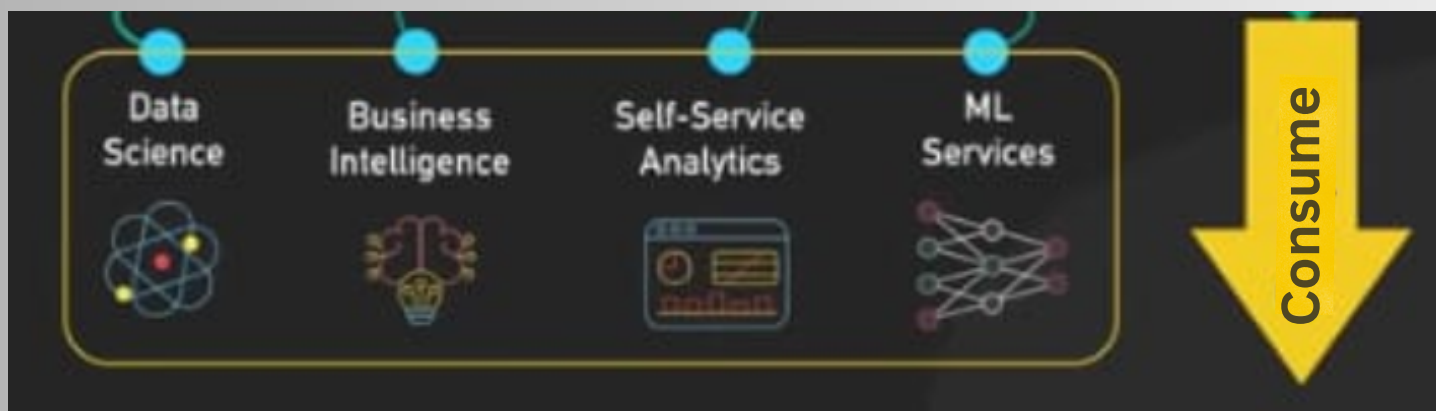


5. Consume: Delivering Insights & Impact

Data is finally ready to power decisions and experiences.

- **Business Intelligence:** Dashboards, reports, visualizations (Looker, Power BI)
- **Self-Service Analytics:** Teams explore data without engineering help
- **Data Science & ML:** Predictive models, A/B tests, personalizations

From CEOs to product teams, this is where data drives action.



🚀 Key Takeaways

- Every data-driven decision is powered by this data pipeline.
- A data pipeline is a journey, moving data from raw input to business value in structured stages.
- No matter the tool or company, these 5 steps are always in play:

Collect → Ingest → Store → Compute → Consume

⚡ Bottom Line

- Great pipelines are reliable, scalable, and aligned to business use cases.
- SQL plays a central role in making all this happen.



Now that we've got the data pipeline basics down, it's time to build one (SQL-style). Stay tuned!

