G.AI.L

Tecnológico de Monterrey

COECYTJAL
Consejo Estatal de Ciencia y Tecnología de Jalisco

Machine Learning Operations Bootcamp

Module 3

MODULE 3: Storage Systems, Databases and Data warehouses

Session 1

About us

Grisell Reyes Rios

- I am a chemist who turned into a Data Scientist/ Data Engineer
- I like to travel at every opportunity that I have
- I speak German

Edgar Talledos

- I'm a physicist
- My favorite sentence is: "Algo tranqui."

Important Notes

Use your name and last name to identify yourself in Meet

Mute your microphone

Raise your hand to participate

Turn off your camera if you have connection issues

Academy Code of Conduct

Be respectful
All questions and ideas are valid and welcome

Be welcoming and patient

Be careful with the words you use

Session Goals

At the end, you will be able to:

- **Understand** the fundamentals of storage
- **Differentiate** between OLTP (databases) and OLAP (data warehouses)
- **Review** use cases oriented to MLOps

Table of Contents

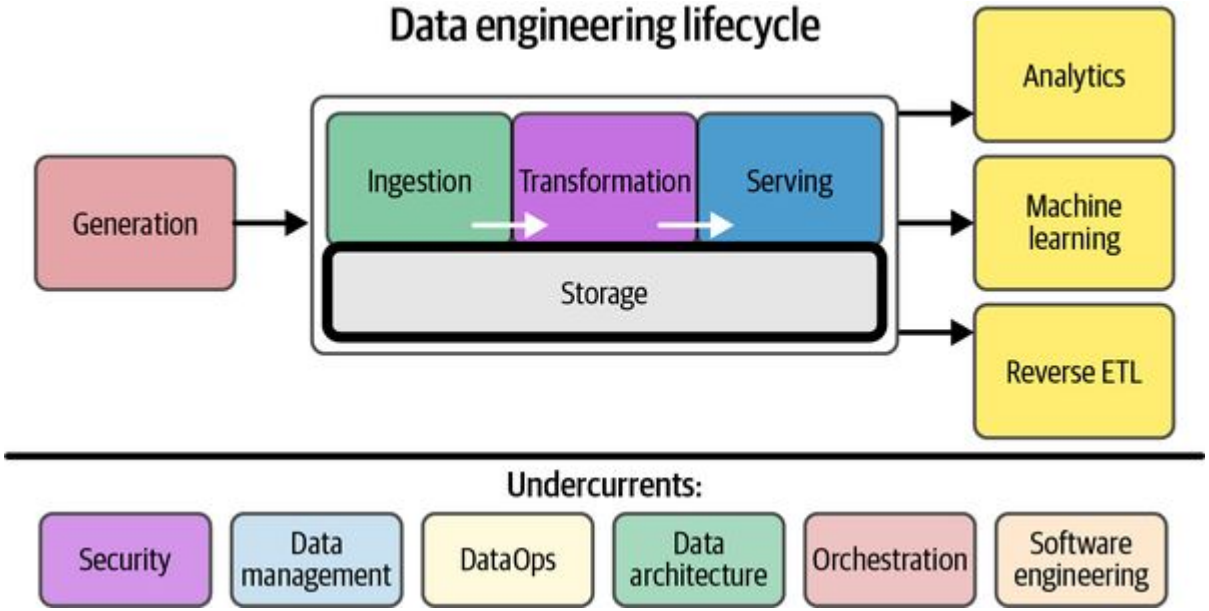
Storage Systems

Understand the fundamentals and use cases

What is Storage?

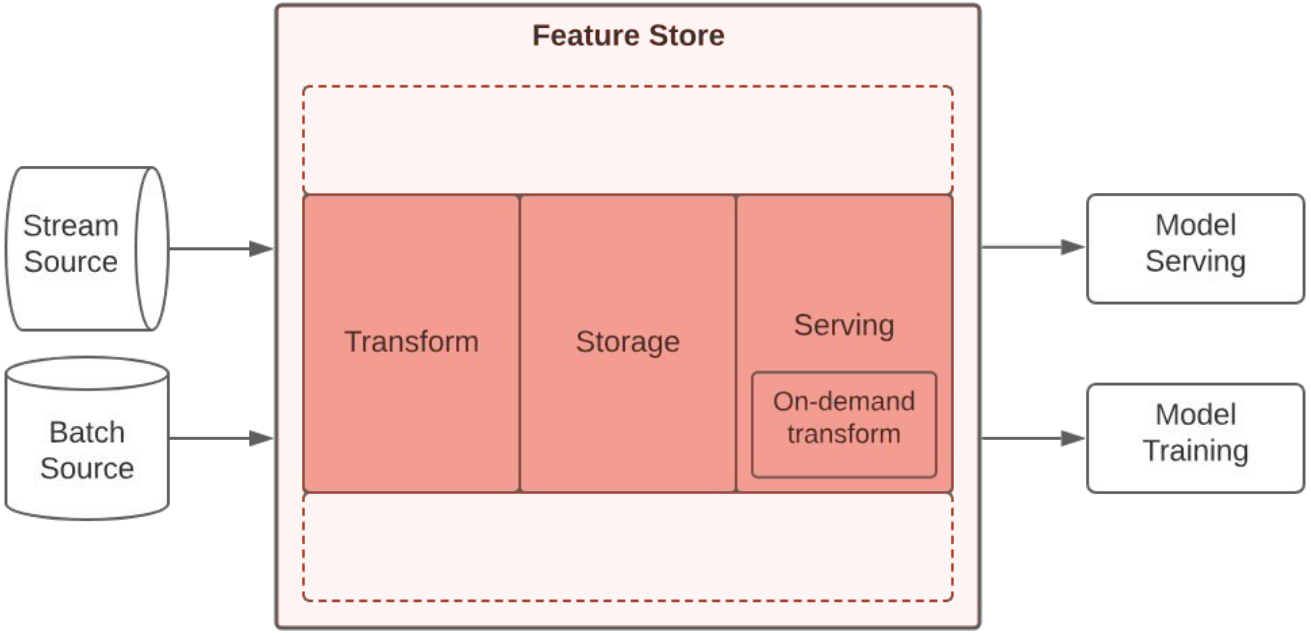
- Process through digital data is saved within a data storage device.
- Enables the computer to retain data either temporarily or permanently.

Storage in the Data Engineering Lifecycle



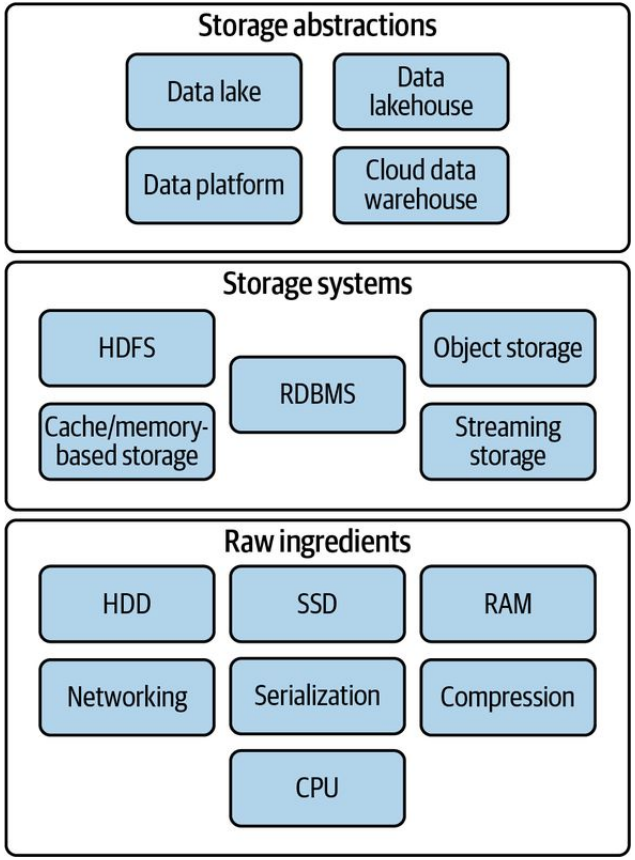
Source: Reis, Joe and Housley Matt. (2022). Fundamentals of Data Engineering. O´Reily Media Inc.

Storage in the Feature Store Lifecycle, part of MLOps



Source: https://mlops.community/mlops-is-mostly-data-engineering/

Storage Types



Cache hierarchy displaying storage types with approximate pricing and performance characteristics.

Storage type	Data fetch latency[a]	Bandwidth	Price
CPU cache	1 nanosecond	1 TB/s	N/A
RAM	0.1 microseconds	100 GB/s	$10/GB
SSD	0.1 milliseconds	4 GB/s	$0.20/GB
HDD	4 milliseconds	300 MB/s	$0.03/GB
Object storage	100 milliseconds	10 GB/s	$0.02/GB per month
Archival storage	12 hours	Same as object storage once data is available	$0.004/GB per month

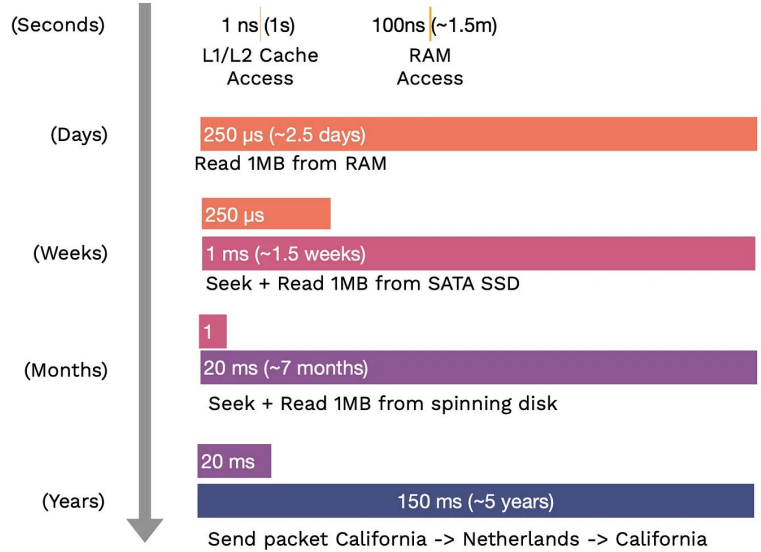
[a] A microsecond is 1,000 nanoseconds, and a millisecond is 1,000 microseconds.

Source: Reis, Joe and Housley Matt. (2022). Fundamentals of Data Engineering. O'Reily Media Inc.

Data Storage Systems Types

1. File Storage

The filesystem is a fundamental abstraction. Its fundamental unit is a file — which can be text or binary, is not versioned, and is easily overwritten. The filesystem is usually on a disk connected to your machine — physically connected on-prem, attached in the cloud, or even distributed.
The following diagram shows speed comparisons for some memory types:

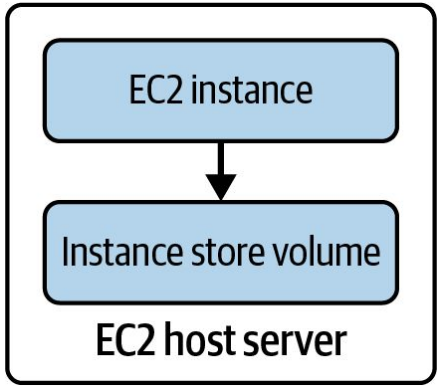


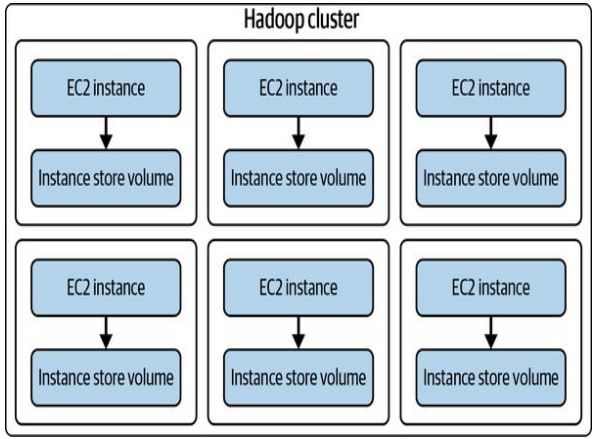
Source: https://medium.com/aiguys/mlops-data-management-34ed06558998

2. Cloud Virtualized Block Storage

Free engineers from dealing with Storage Area Network clusters and networking details. An example is Amazon Elastic Block Store (EBS); other public clouds have similar offerings. EBS is the default storage for Amazon EC2 virtual machines; other cloud providers also treat virtualized object storage as a key component of their Virtual Machine offerings.

Local Instance Volume over a network can construct a cluster like Hadoop Cluster.

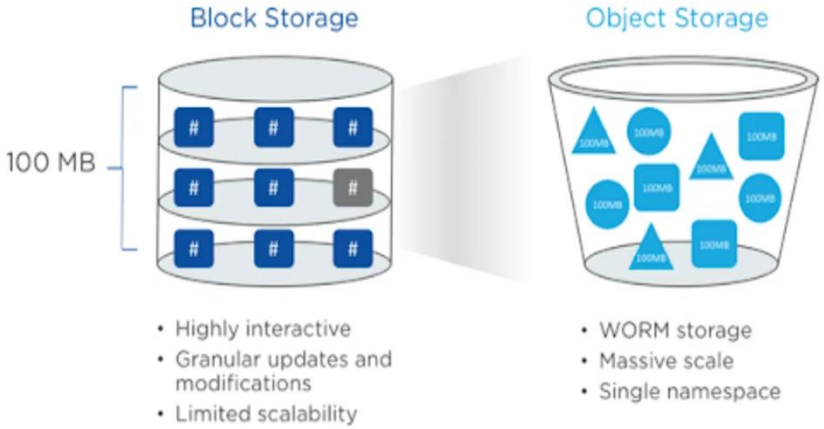




Source: Reis, Joe and Housley Matt. (2022). Fundamentals of Data Engineering. O'Reily Media Inc.

Data Storage Systems

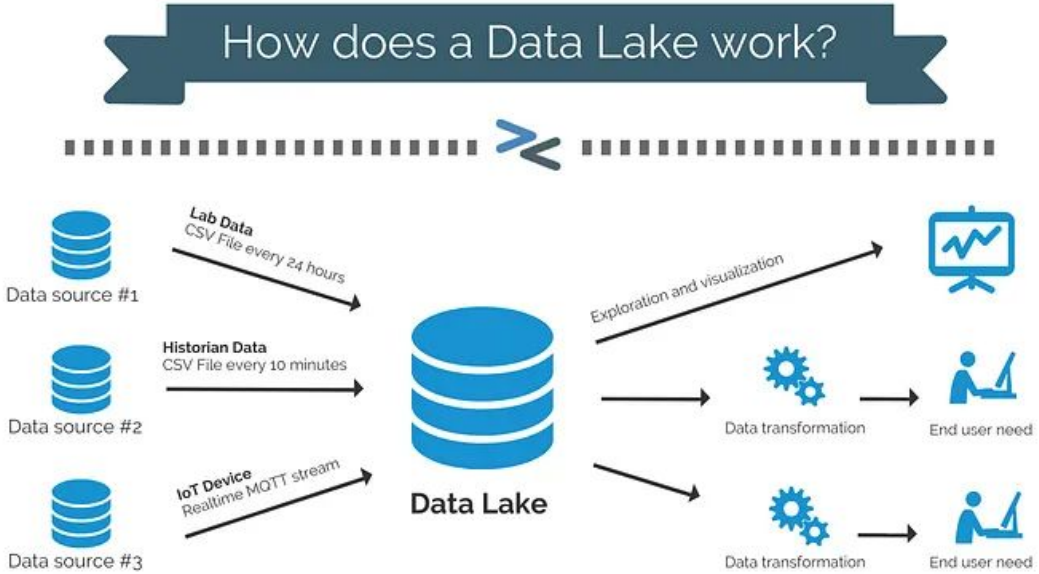
3. Object Storage: is a data storage architecture for storing structured, semistructured and unstructured data, which sections data into units—objects—and stores them in a structurally flat data environment. Each object includes the data, metadata, and a unique identifier that applications can use for easy access and retrieval. It is the fundamental brick of a Data lake.



Source: https://www.freecodecamp.org/news/how-to-use-object-storage-for-parallelization-and-experimentation/

Data Storage Systems

3. Object Storage: Datalake



Source: https://medium.com/@daamian.kalupa/data-lake-best-practices-throughout-implementation-b976e9be03c4

Break Time

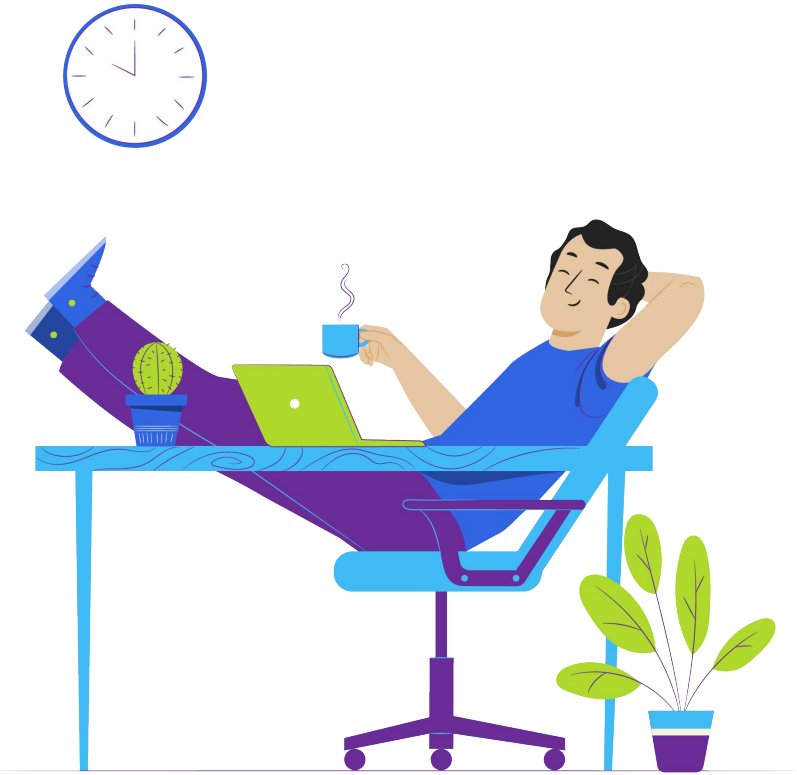


Table of Contents

OLAP -OLTP

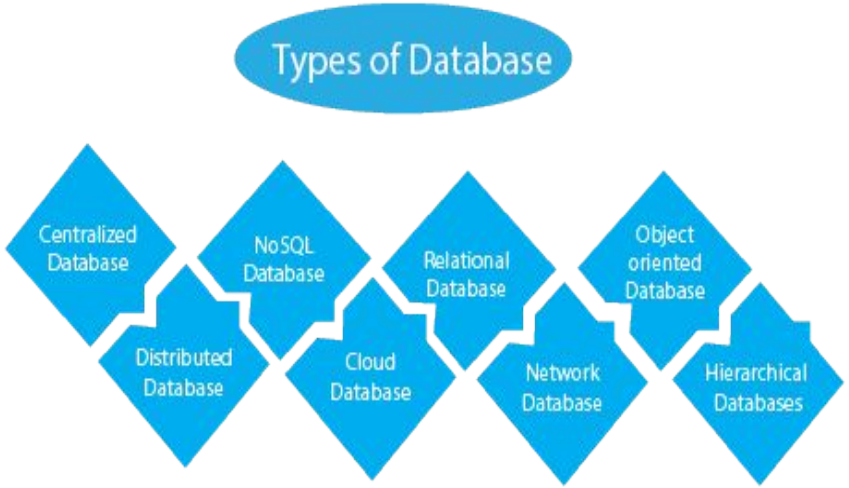
Differences between OLAP and OLTP

OLTP versus OLAP

	OLAP: Online analytical processing	OLTP: online transaction processing
Purpose	Analyze large volumes of data to support decision-making	Manage and process real-time transactions.
Data source	Uses historical and aggregated data from multiple sources.	Uses real-time and transactional data from a single source.
Data structure	Uses multidimensional (cubes) or relational databases.	Uses relational databases.
Data model	Uses star schema, snowflake schema, or other analytical models.	Uses normalized or denormalized models.
Volume of data	OLAP has large storage requirements. Think terabytes (TB) and petabytes (PB).	OLTP has comparatively smaller storage requirements. Think gigabytes (GB).
Response time	OLAP has longer response times, typically in seconds or minutes.	OLTP has shorter response times, typically in milliseconds
Example applications	OLAP is good for analyzing trends, predicting customer behavior, and identifying profitability.	OLTP is good for processing payments, customer data management, and order processing.

Source: https://aws.amazon.com/compare/the-difference-between-olap-and-oltp/

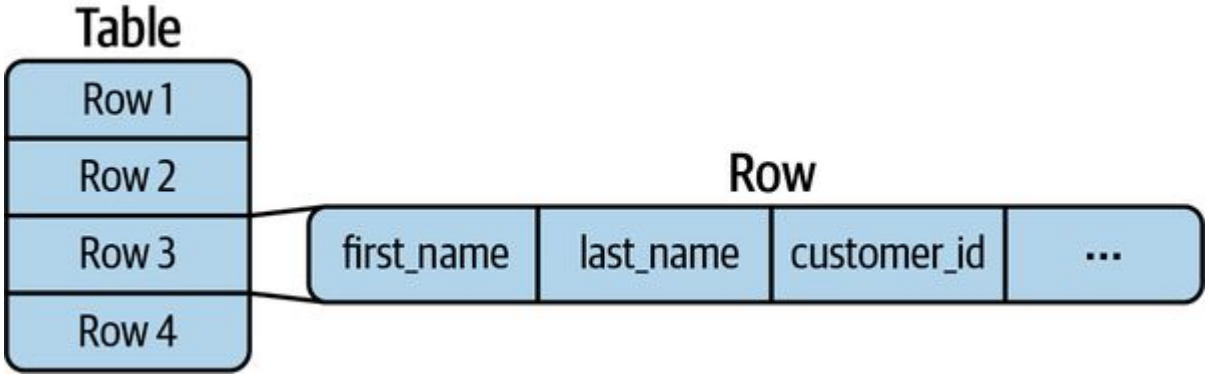
Databases Types



- Schema and record sizes
- Number of clients
- Types of queries and access patterns
- Rates of the read and write queries
- Expected changes in any of these variables

Relational Databases

A *relational database management system* (RDBMS) is one of the most common application backends. They were developed at IBM in the 1970s and popularized by Oracle in the 1980s. The growth of the internet saw the rise of the LAMP stack (Linux, Apache web server, MySQL, PHP) and an explosion of vendor and open-source RDBMS options. Data is stored in a table of *relations* (rows), and each relation contains multiple *fields* (columns). Each relation in the table has the same *schema* (a sequence of columns with assigned static types such as string, integer, or float). Rows are typically stored as a contiguous sequence of bytes on disk.

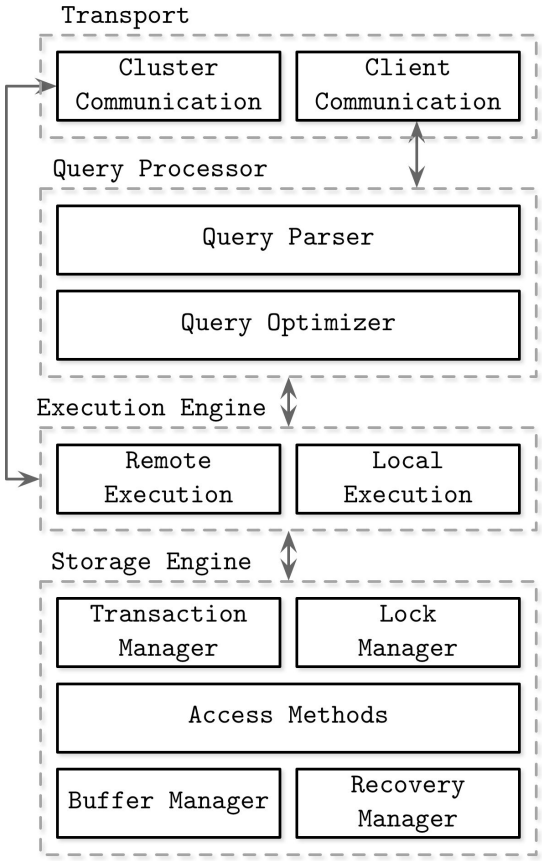


Source: Reis, Joe and Housley Matt. (2022). Fundamentals of Data Engineering. O'Reily Media Inc.

Structured Query Language

- SQL has been around for over 40 years.
- It is recognizable, documented, widely used, safe, and versatile.
- Its main drawback is that it restricts the user from working with a predefined tabular schema, and more care must be taken to organize and understand the data before it is used.

Database Management Systems



Components of a Database Management Systems

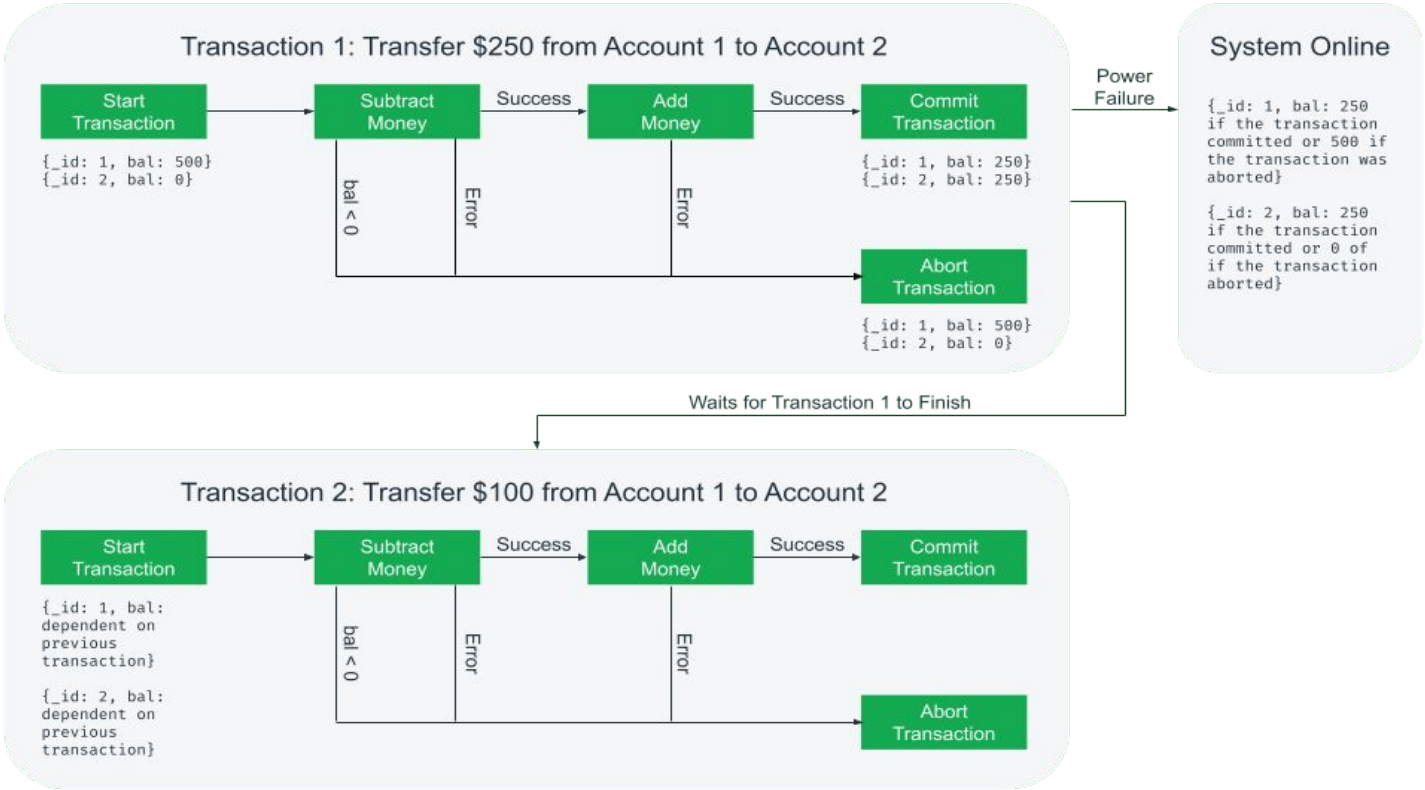
- Storage Engine
- Transaction Manager
- Lock Manager
- Access Methods
- Buffer
- Recovery Manager

Source: Petrov, A. (2019). Database Internals. O'Reily Media Inc.

ACID Principle

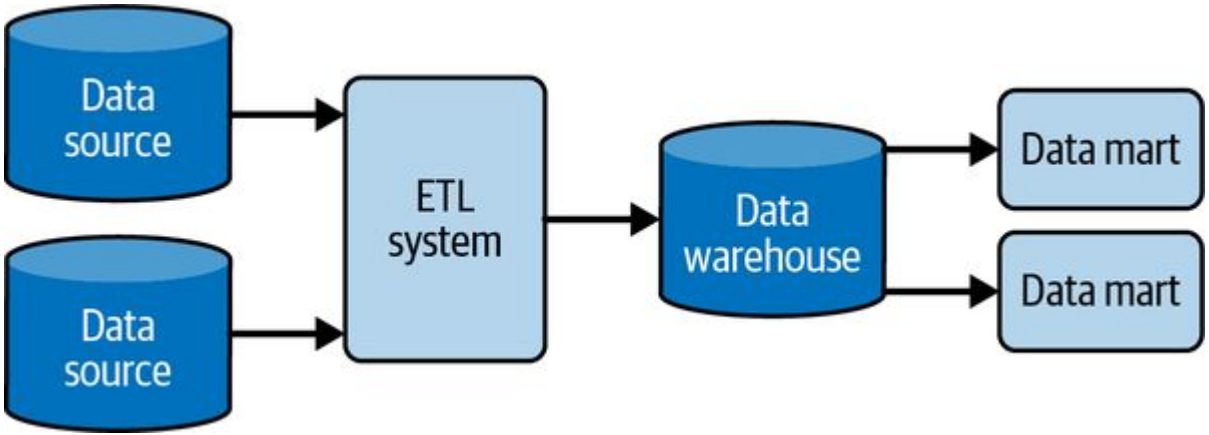
Relational databases must exhibit the four ACID properties:

- Atomicity
- Consistency
- Isolation
- Durability



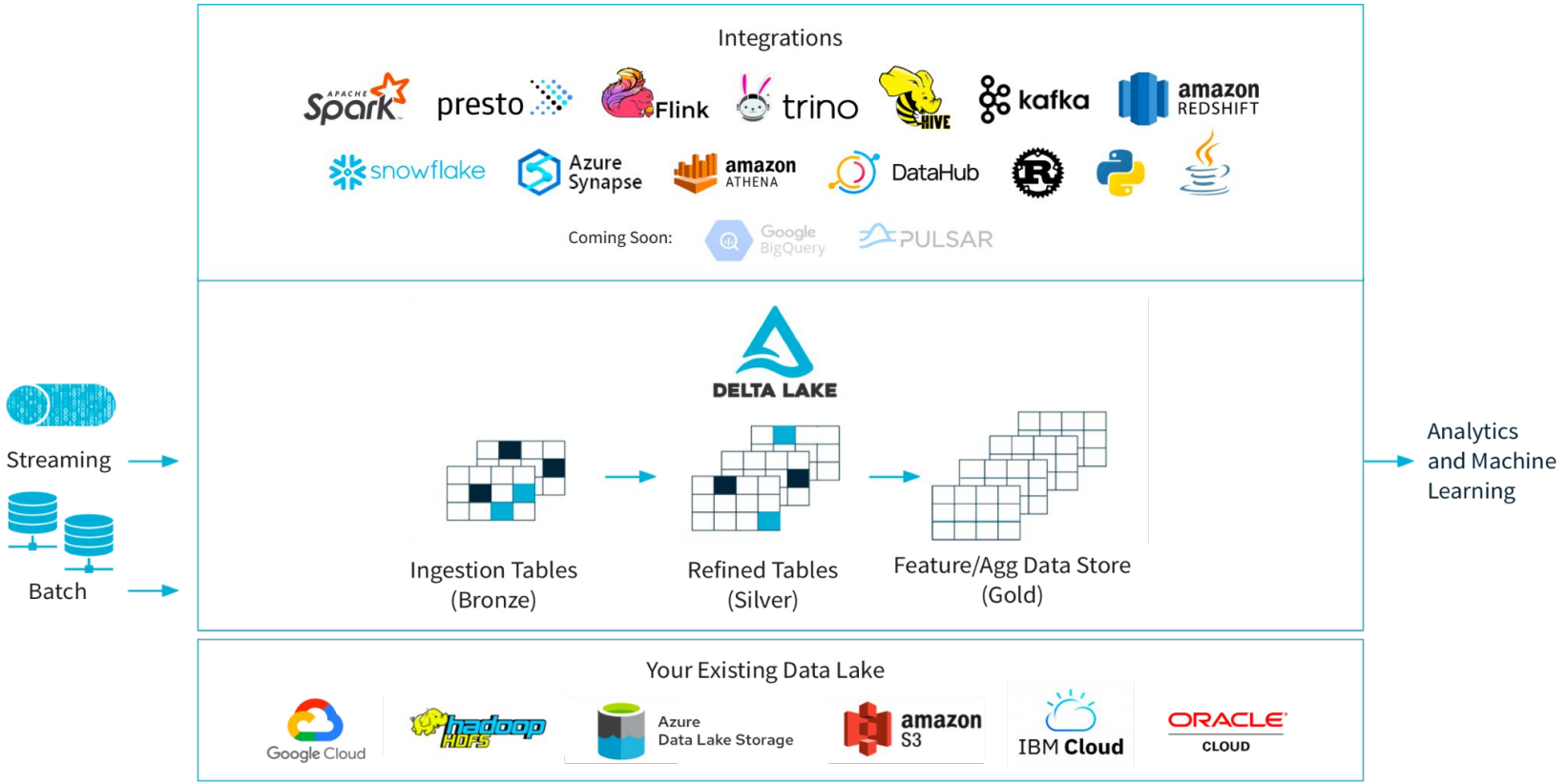
OLAP

In contrast to an OLTP system, an *online analytical processing* (OLAP) system is built to run large analytics queries and is typically inefficient at handling lookups of individual records. For example, modern column databases are optimized to scan large volumes of data, dispensing with indexes to improve scalability and scan performance. Any query typically involves scanning a minimal data block, often 100MB or more in size. Trying to look up thousands of individual items per second in such a system will bring it to its knees unless it is combined with a caching layer designed for this use case.



Source: Reis, Joe and Housley Matt. (2022). Fundamentals of Data Engineering. O'Reily Media Inc.

Delta Lake



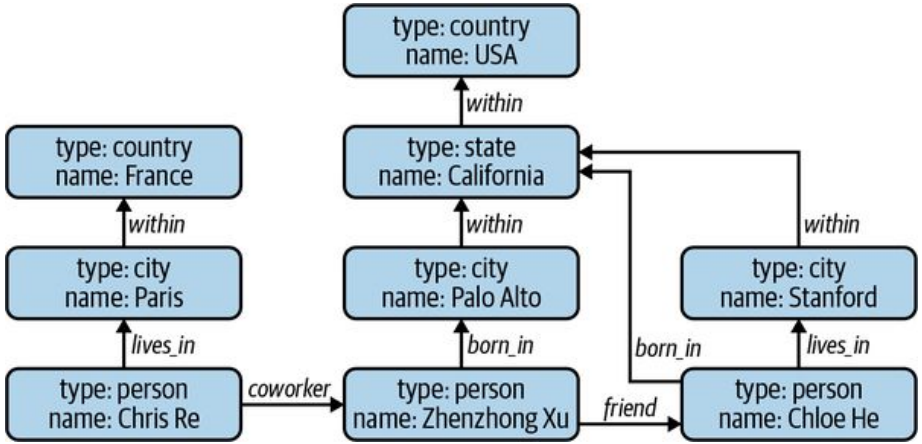
Source: Michael Armbrust, Ali Ghodsi , Reynold Xin , Matei Zaharia.(2021). Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics [white paper]. Databricks, Berkeley, Stanford University.
https://www.cidrdb.org/cidr2021/papers/cidr2021_paper17.pdf

NoSQL Databases

Originally started as a hashtag for a meetup to discuss non relational databases, NoSQL has been retroactively reinterpreted as Not Only SQL, as many NoSQL data systems also support relational models. Two major types of non relational models are the document model and the graph model.
The document model targets use cases where data comes in self-contained documents, and relationships between one document and another are rare.
The graph model goes in the opposite direction, targeting use cases where relationships between data items are common and important.


```
{
  "Title": "Harry Potter",
  "Author": "J .K. Rowling",
  "Publisher": "Banana Press",
  "Country": "UK",
  "Sold as": [
    {"Format": "Paperback", "Price": "$20"},
    {"Format": "E-book", "Price": "$10"}
  ]
}
```


Document Model



Graph Model

Source: Reis, Joe and Housley Matt. (2022). Fundamentals of Data Engineering. O'Reily Media Inc.

Graph Databases: Enhanced AI

There are at least four main areas where graphs provide context for AI, which we'll detail in the following sections throughout the rest of this white paper.

1. **Knowledge graphs** provide context for decision support (e.g., for call center staff or support engineers) and help ensure that answers are appropriate to the situation (e.g., autonomous vehicles in rainy driving conditions).
2. **Graphs** offer greater processing efficiency, and thus, graph-accelerated machine learning uses graphs to optimize models and speed up processes.
3. **Connected feature extraction** analyzes data to identify the most predictive elements within data. Basing a predictive model on strong characteristics found in the data improves accuracy.
4. **Graphs** offer a way to provide transparency into how AI makes decisions. This area is called AI explainability.

Source: https://neo4j.com/product/neo4j-graph-database/

Table of Contents

Use cases

Use cases applied to MLOps

How databases are used in different stages of the MLOps cycle:

Data Ingestion and Feature Engineering:

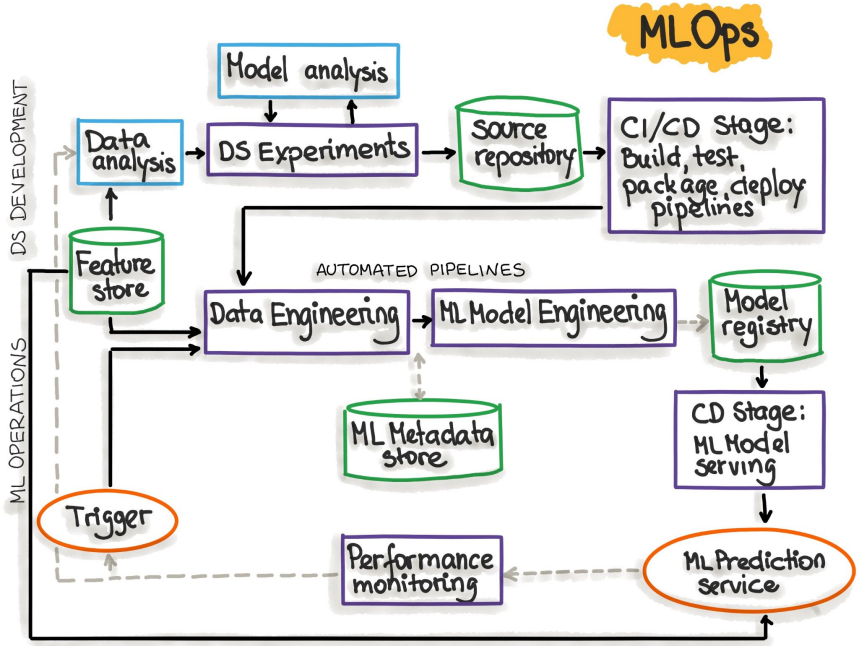
- Storing Raw Data
- Data Preprocessing and Feature Engineering
- Versioning and Lineage Tracking

Model Training and Evaluation:

- Training Data Management
- Feature Selection and Model Optimization
- Model Performance Evaluation

Model Deployment and Monitoring:

- Model Serving and Inference
- Model Monitoring and Logging
- Model Governance and Auditing



Source: https://ml-ops.org/content/mlops-principles

Use Cases

Basecamp Research is transforming biotech, using Neo4j's graph database to map Earth's biodiversity.
The team at Basecamp has built partnerships with nature parks across five continents to collect biological & chemical data across more than 50% of global biomes to address the fundamental knowledge gap of us knowing less than 0.001% of our planet's biodiversity.

With Neo4j at its core, Basecamp Research developed a multidimensional knowledge graph that maps three broad categories:

- Environmental, geological, and chemical conditions
- Microecology, metagenomics, and genomic context
- Deep learning-derived functional and structural protein characteristics

The application of deep learning models on biological sequence data, such as AlphaFold2. For example through graph embeddings that are part of Neo4J's graph data science library.

Source: https://neo4j.com/case-studies/basecamp-research/

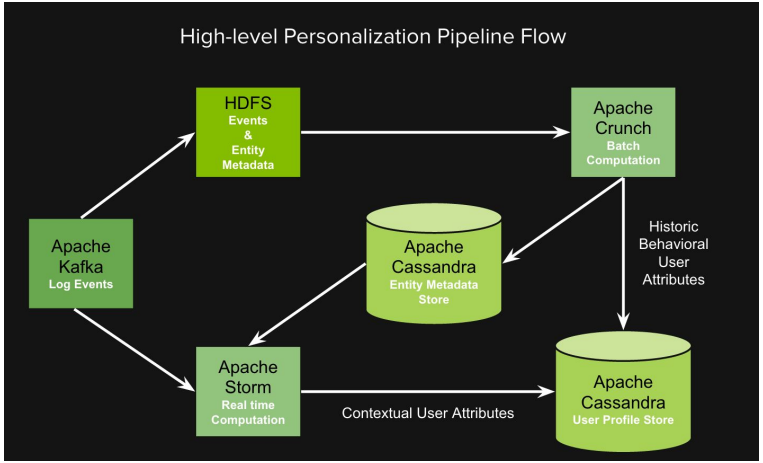
Use Cases

The personalization stack that we use is *Kafka* for log collection, *Storm* for real-time event processing, *Crunch* for running batch mapreduce jobs on *Hadoop*, and *Cassandra* to store user profile attributes and metadata about entities like playlists, artists, etc.

There are two sets of Kafka consumers, subscribed to different topics, for consuming events:

1. Kafka Consumers for Hadoop, which writes these events to HDFS. All the raw logs on HDFS are processed in Crunch to remove duplicate events, filter out unwanted fields, and convert records into Avro Format.
2. Kafka Consumer Spouts run within Storm topologies which stream the events for real-time computation.

The Storm pipelines process raw log events from Kafka, filter out unwanted events, decorate the entities with metadata fetched from EMS, group it per user, and determine user-level attributes by some algorithmic combination of aggregation and derivation. When combined, these user attributes represent a user's profile and are stored in a Cassandra Cluster, which we will call the User Profile Store (UPS).



Source: https://engineering.atspotify.com/2015/01/personalization-at-spotify-using-cassandra/

Menti!

10 minutes

Please, give us feedback!

Q&A

Thank you!