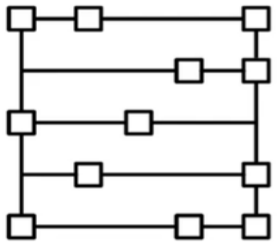
The Overview of Data Engineering Ecosystem

from-database-to-ai-the-evolution-of-data-platforms

Data Type

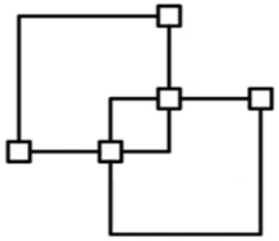
정형 데이터

비정형 데이터



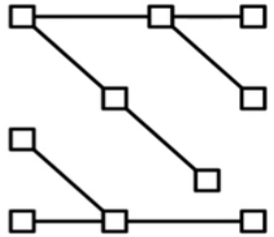
Structured

Data that follows a rigid format and can be organized into rows and columns.



Semi-structured

Mix of data that has consistent characteristics and data that does not conform to a rigid structure.



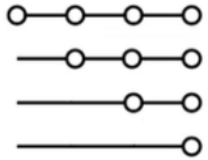
Unstructured

Data that is complex and mostly qualitative information that cannot be structured into rows and columns.

Data Sources

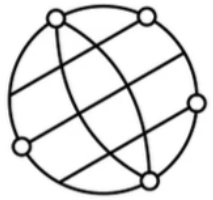
Data also comes in a wide-ranging variety of file formats being collected from a variety of data sources,

Relational Database



Non-Relational Database

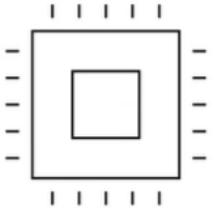
APIs



Web Services

Data Streams

Social Platforms



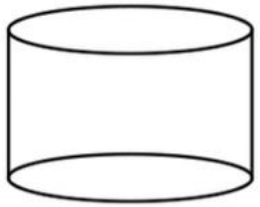
Sensor Devices

Data Repository(자료 저장소)

운영계 분석계

거래(운영) 데이터 분석 데이터

Transactional (Operational)

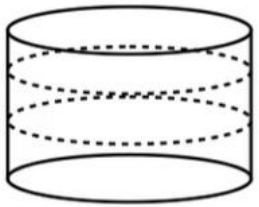


Transactional
or
Online Transaction
Processing (OLTP)
System

Designed to store high volume day-to-day operational data

Typically relational, but can also be non-relational

Analytical



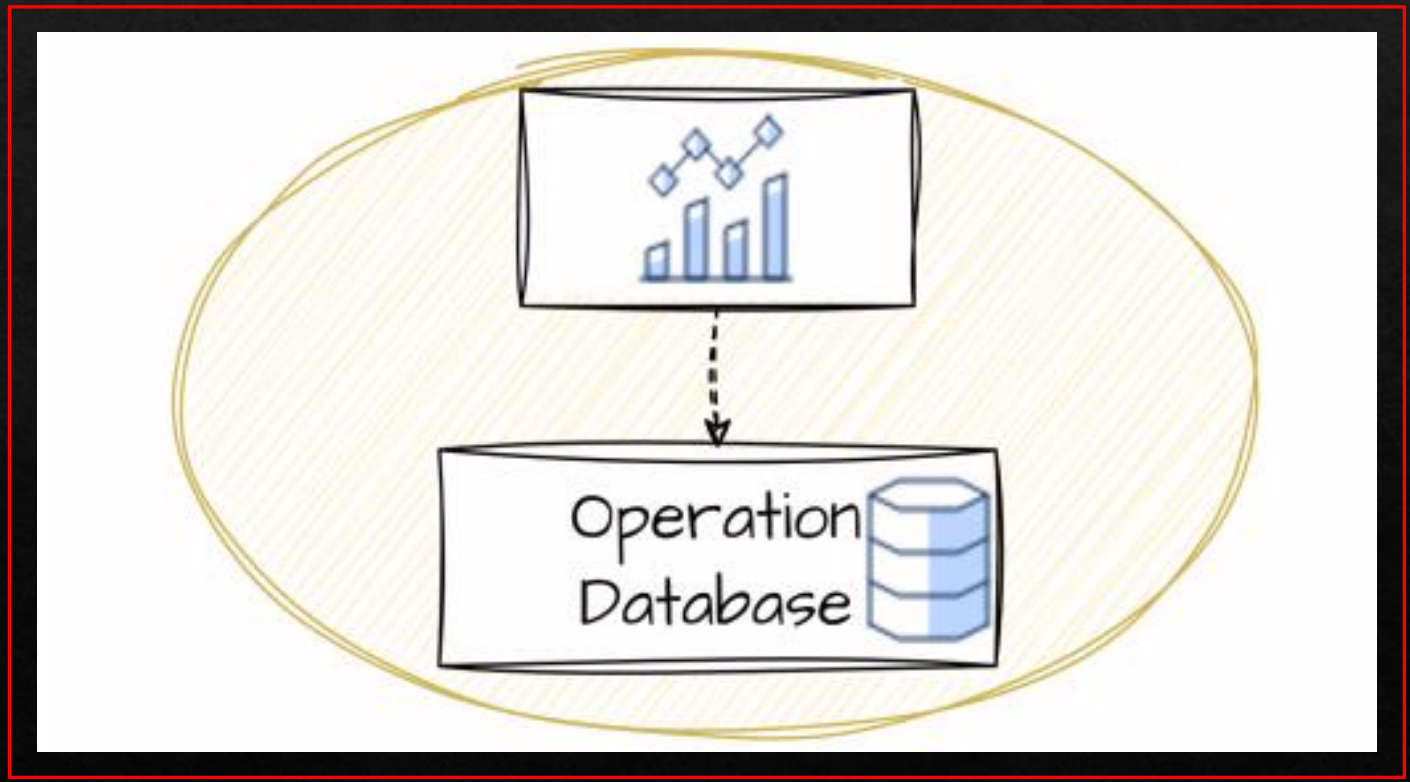
Analytical
or
Online Analytical
Processing (OLAP)
Systems

Optimized for conducting complex data analytics

Include relational and non-relational databases, data warehouses, data marts, data lakes, and big data stores

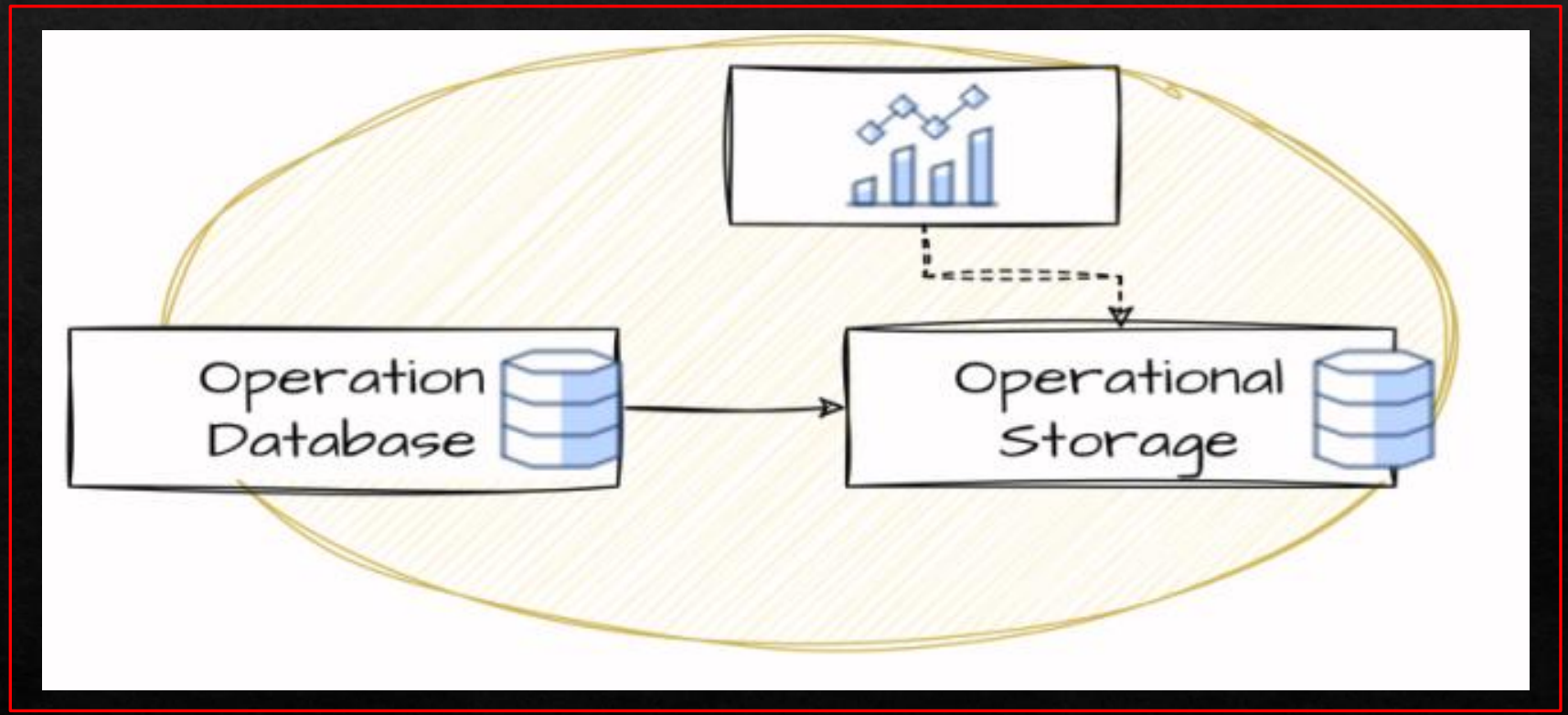
Level 0

- The reports are generated by directly accessing **transaction systems** and querying data from them.
- Tools such as **Excel, Power BI,** or similar applications are employed to collect and present data in the form of tables or charts.



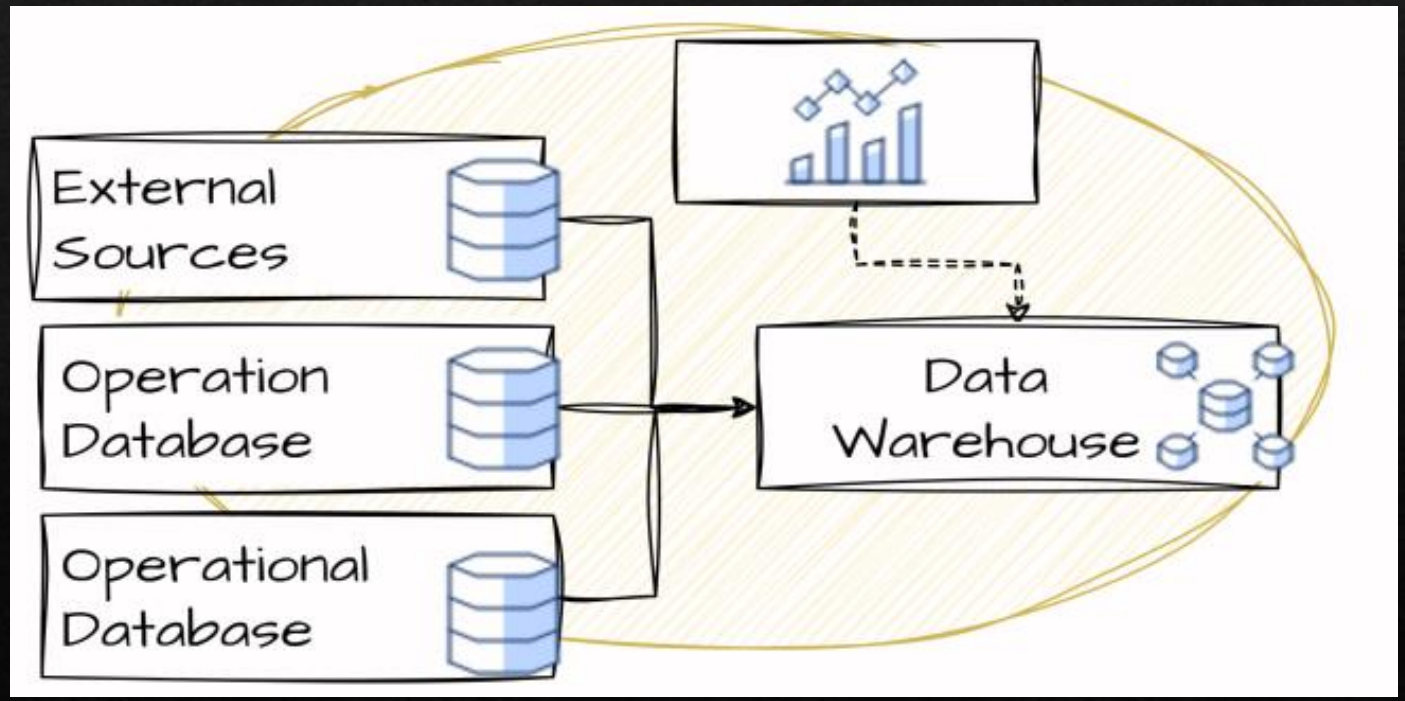
Level 1

- At level 0 stage, both your reports and applications are struggling with performance issues tied to heavy analytical queries consuming data from your transaction system daily.
- Timing is key; **data extraction** by IT department is typically carried out during periods of low activity in the transaction system.



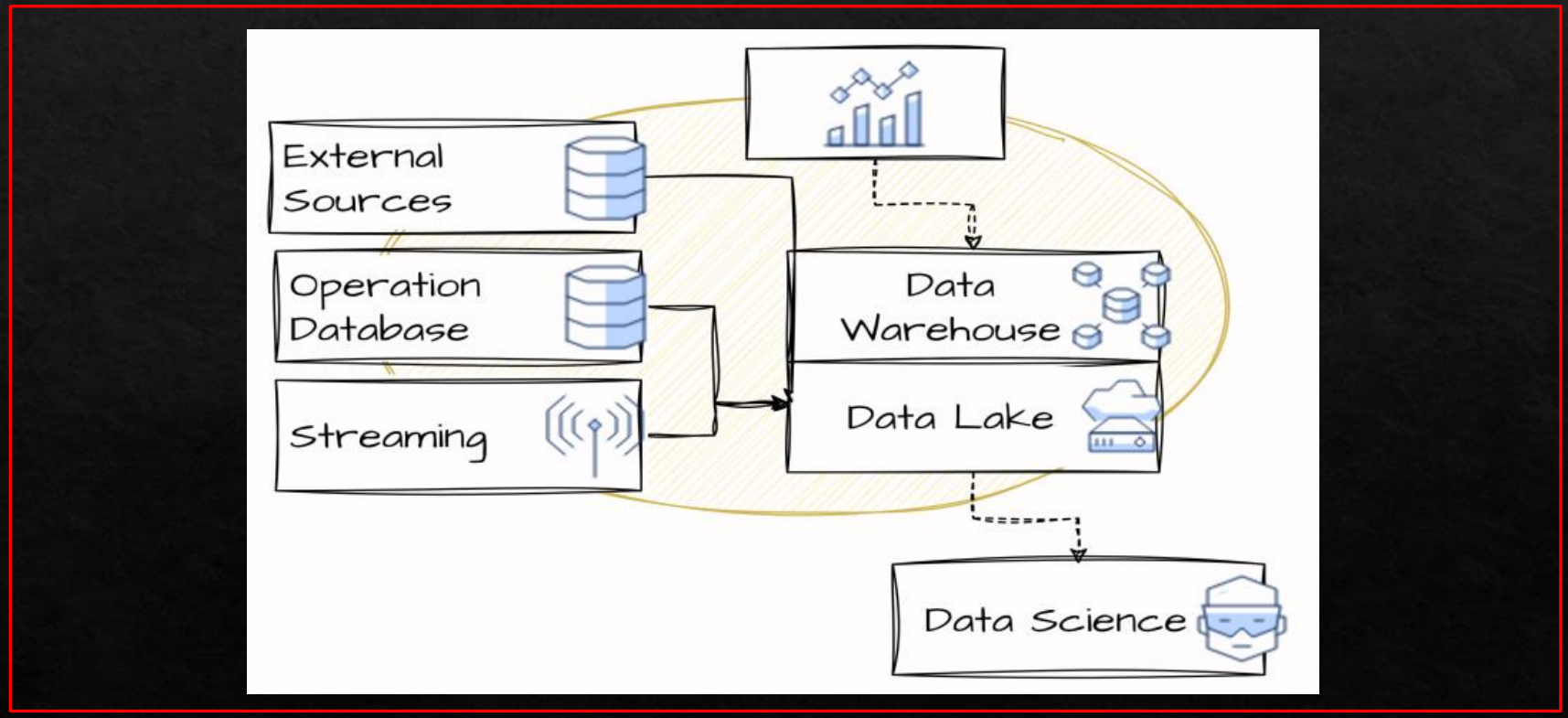
Level 2

- The data warehouse by **ETL** serves as the foundation for decision-making, ensuring that it contains the right data to support informed choices.
- This transformation requires to select between a public cloud or on-premise server and choosing an appropriate data warehouse engine

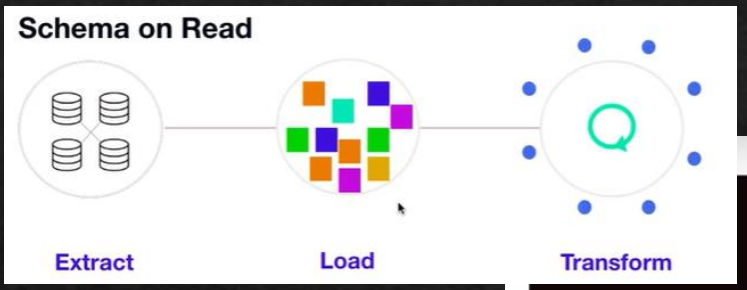


Level 3

- At level 2, Data warehouses typically require time-consuming data modeling, a **data lake** adopts a schema-on-read approach, allowing data to be stored without predefined schema definitions.
- To process this data, **Apache Spark** is often employed, working on a compute cluster to handle large-scale data processing.



Schema on Read vs Schema on Write



Hadoop, Apache Spark

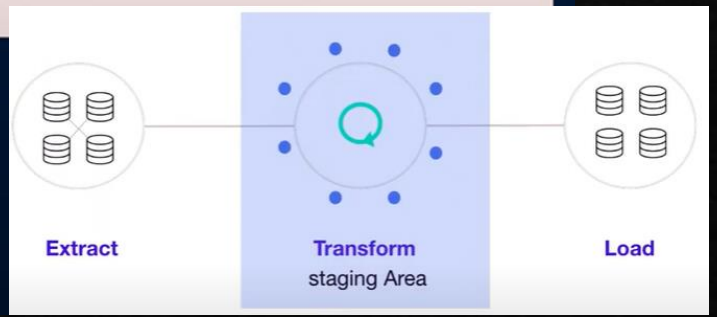
Schema on Read

SSN
Friends
Country
Movies
Hobbies
Food
Phone

```
{
    "emp_details" : [
        {
            "emp_name" : "Mark" ,
            "email" : mark@company.com",
            "role : "BA"
        }
    ]
}
```

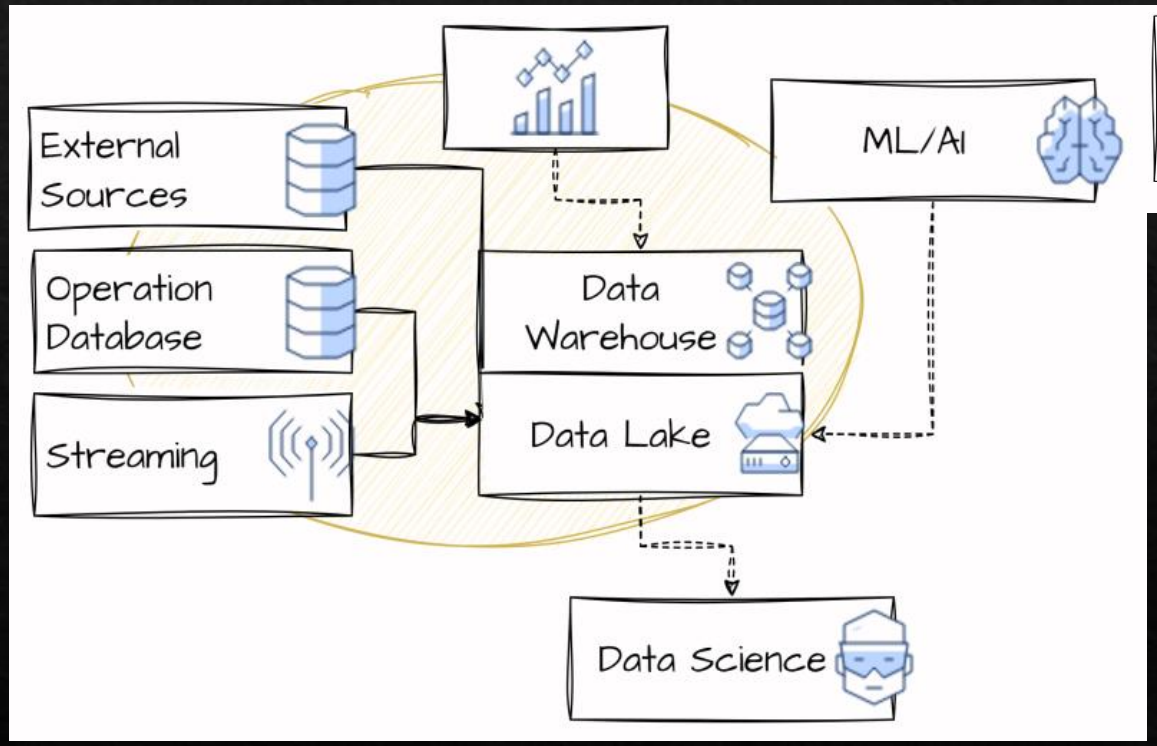
Schema on Write

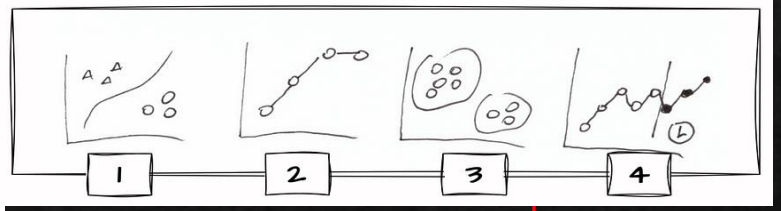
SQL RDBMS



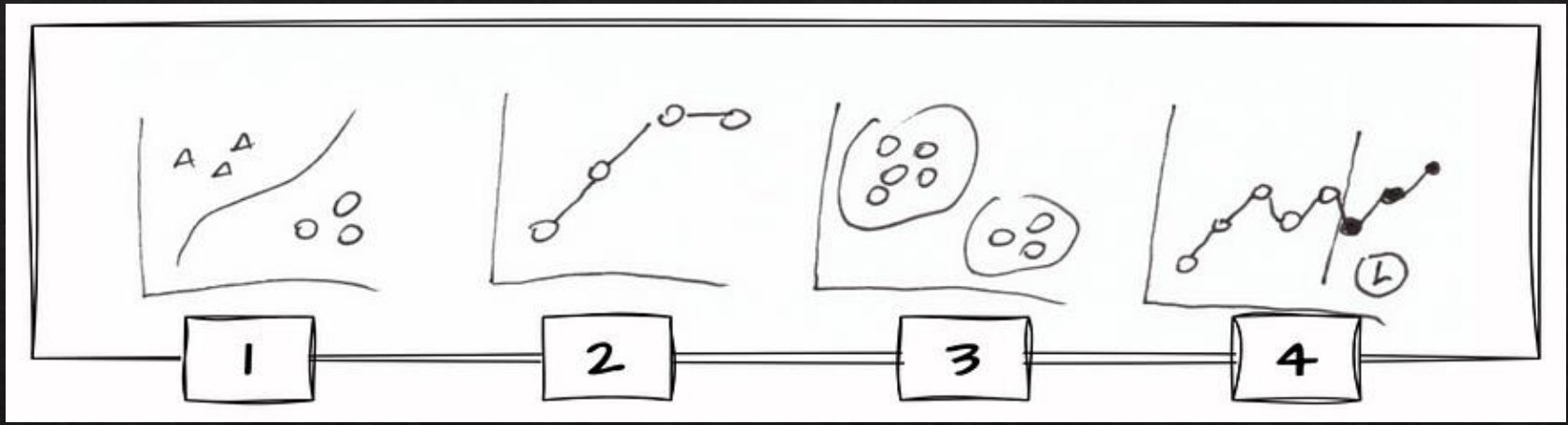
Level 4

- At this advanced level, decision-making goes beyond relying solely on current numbers; we integrate forecasts derived from machine learning models. This transformative approach enables us to not only respond to present circumstances but also proactively plan based on predictive analytics.



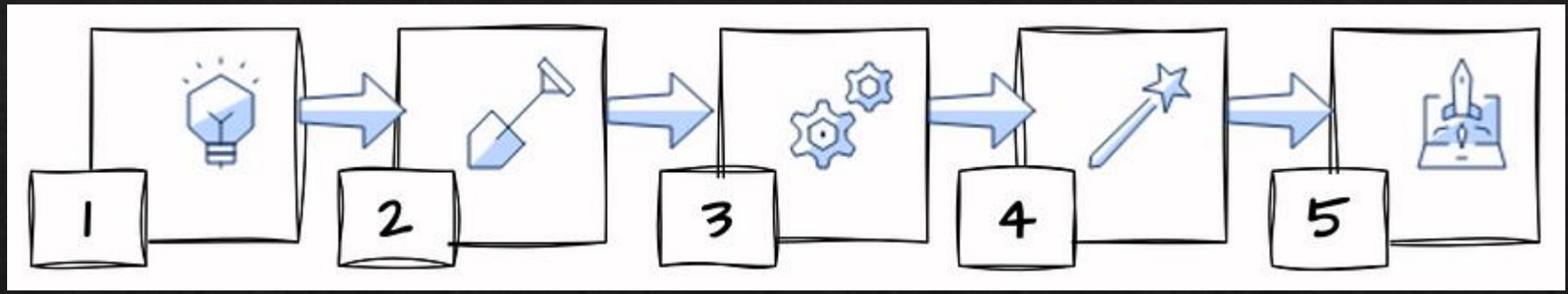


Level 4_Machine Learning Models



1. Classification: Used for pattern recognition, it can label entities like client types, images, and document categories.
2. Regression: Investigates the relationship between independent variables (features) and a dependent variable (outcome), commonly employed for flat price prediction, sales forecasting, and similar applications.
3. Clustering: Groups similar data points into clusters or unlabeled groups, beneficial for tasks like anomaly detection and market segmentation.
4. Time-series: Predicts future numerical values based on time-series data, applicable to scenarios such as forecasting sales, demand, calls, or web traffic.

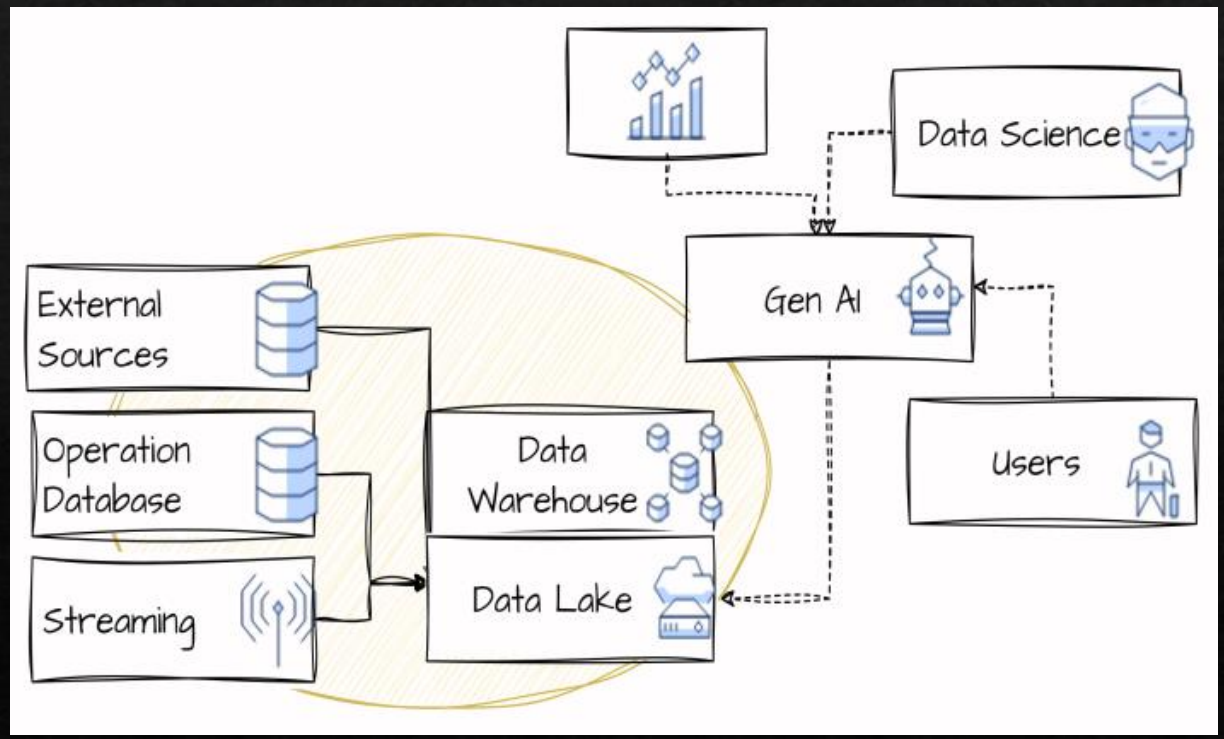
Level 4_To deploy a machine learning process



1. 목적 정의(Define the use case) : Collaborate with business users to determine what the model should predict or recognize.
2. 자료 수집(Get the data) : Identify data sources and ingest the data into a data lake or lakehouse.
3. 자료 정제/분석(Prepare & Anlaysis the data) : Explore and clean the data, transforming it according to the model's requirements.
4. 모델 평가 및 분석(Train the model) : Choose an algorithm and hyperparameters, then evaluate the results.
5. 모델 배포(Deploy the model) : Utilize the trained model to generate the requested results

Level 5

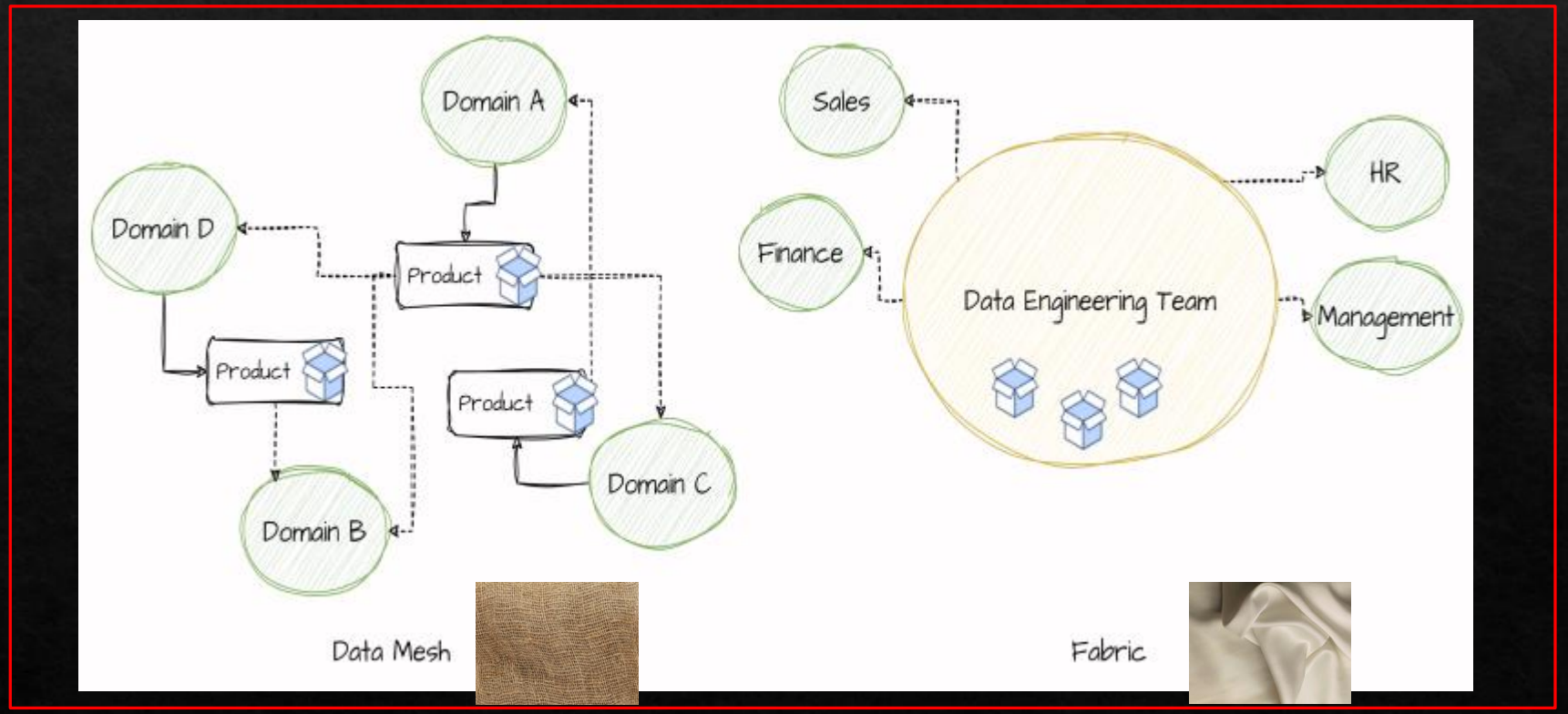
- While we may not have reached Level 5 yet, the ongoing innovation in the AI domain, especially in **Generative AI**, is positioned to influence the evolution of data platforms.
- Users might raise questions about intriguing data, receiving prompt and insightful answers
- This evolution holds the promise of vastly improving access to information within organizations.



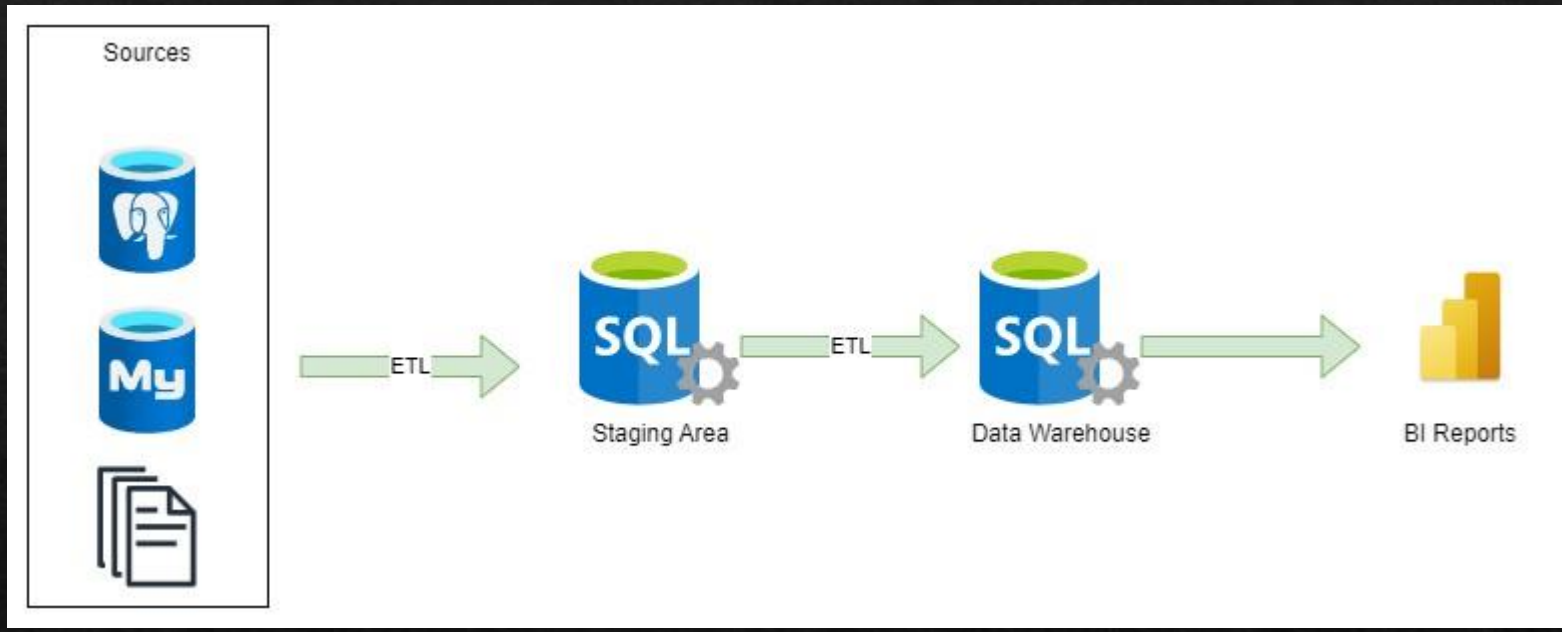
Appendix

Data Mesh vs Fabric

- Data Mesh is to **decentralize** data platform model where domain-oriented teams are responsible for delivering, maintaining, and ensuring the quality of their respective data products.
- Fabric emphasizes the **centralization** of activities associated with a data platform, encompassing data ingestion, ETL, reporting, and data governance.



Data warehouse



Advantages:

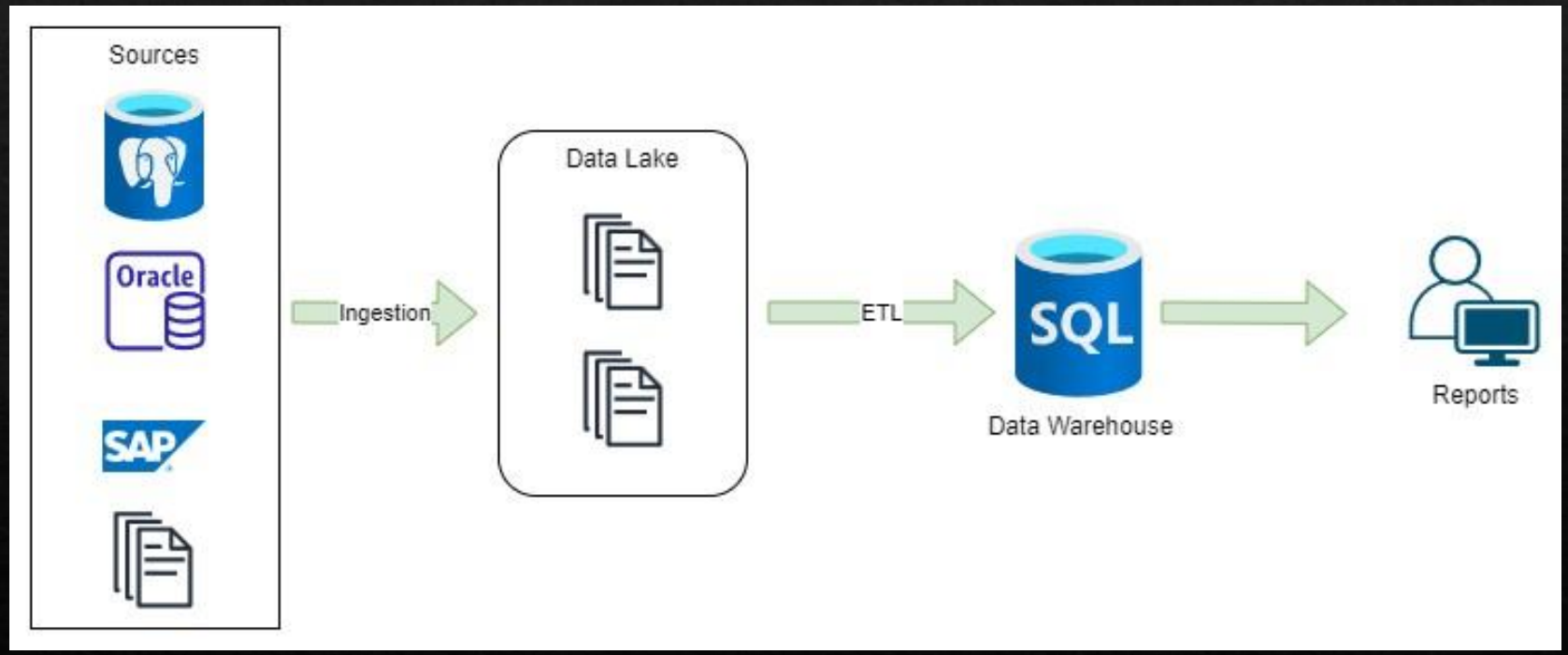
- Data is structured, modeled, cleaned, and prepared.
- Easy access to data.
- Optimized for reporting purposes.
- Column and row-level security, data masking.
- ACID transaction support.

Disadvantages:

- Complexity and time-consuming for changes in implementation in the data model and the ETL process.
- Schema must be defined.
- Costs of the platform (depends on the database provider and type).
- Database vendor dependence (Complex in a migration from, for instance, Oracle to SQL Server).

Data Lakes

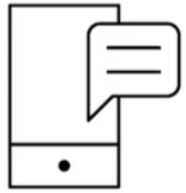
- The emergence of the data lake concept can be traced back to the early 2000s, evolving as a response to the limitations of traditional data management systems and **the advent of big data**.
- **Schema-on-Read**: Unlike data warehouses, data lakes use a schema-on-read approach, where data is stored in its raw form and only processed and structured when it is read, offering more flexibility.



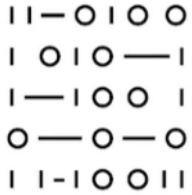
https://medium.com/@mariusz_kujawski/data-warehouse-data-lake-and-data-lakehouse-comparison-of-data-platforms-842f0288b71

Data Engineering Skills

Languages available in the Data Analyst Ecosystem:



Query languages
For example, SQL for querying and manipulating data



Programming languages
For example, Python for developing data applications

Shell and Scripting languages
For repetitive operational tasks

실습하기


PRODUCTION
DIRECTOR
CAMERA
DATE
SCENE
TAKE
transactional vs analytic DB

Which one is better for data analysis? long format vs wide format

- Wide Format
  - 각 행이 여러 관측치(거래, transactional events)를 나타내며 일반적으로 상관관계와 같은 각 변수의 관계를 분석
- Long Format
  - 각 행이 단일 관측치를 나타내며 일반적으로 시계열 데이터나 그룹화된 데이터 분석(analytics)에 적합

Wide Format

Each value is unique in first column

Team	Points	Assists	Rebounds
A	88	12	22
B	91	17	28
C	99	24	30
D	94	28	31

Long Format

The values in the first column repeat

Team	Variable	Value
A	Points	88
A	Assists	12
A	Rebounds	22
B	Points	91
B	Assists	17
B	Rebounds	28
C	Points	99
C	Assists	24
C	Rebounds	30
D	Points	94
D	Assists	28
D	Rebounds	31

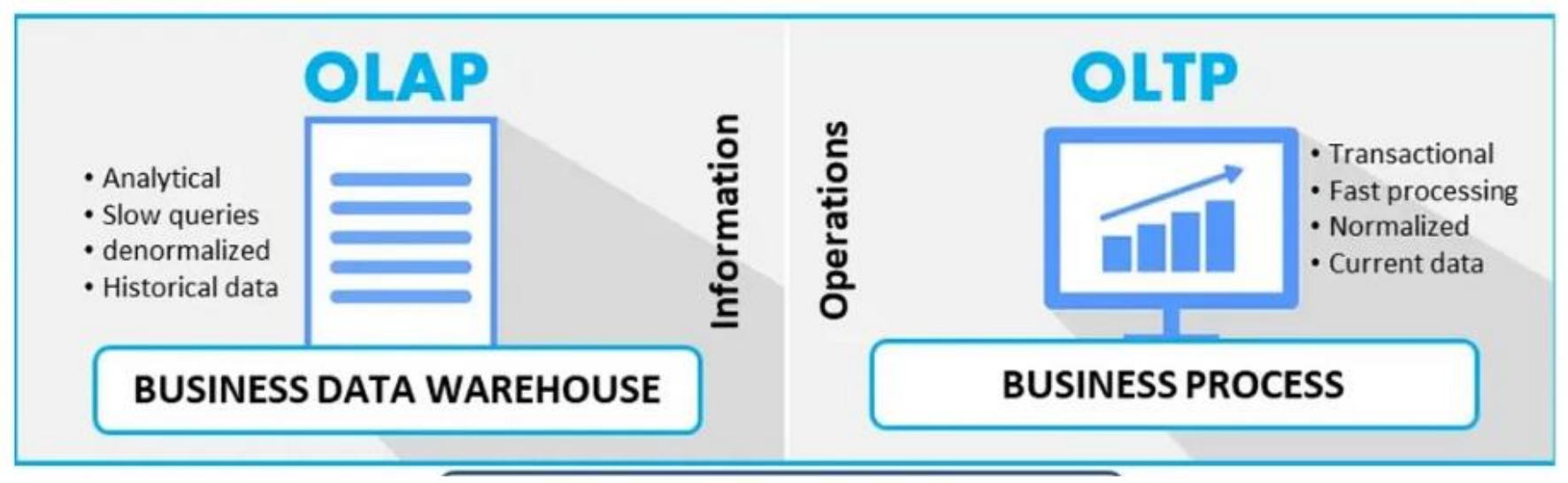
https://pandas.pydata.org/docs/reference/api/pandas.wide_to_long.html

OLTP vs OLAP : SW

- OLTP (Online Transaction Processing)
  - OLTP는 주로 트랜잭션 지향 시스템에서 사용하며, 은행, 슈퍼마켓, 호텔 예약 시스템 등 일상 업무에 사용되며 데이터베이스의 일부 정보를 빠르고 효율적으로 처리하는 데 초점
  - 주로 데이터 입력 및 수정에 최적화되어 있으며, 단일 레코드에 대한 간단한 조회 및 업데이트를 빠르게 처리
  - 일반적으로 많은 사용자에게 높은 처리량을 제공하며, 각 트랜잭션은 일반적으로 짧은 시간 안에 완료됩니다.
  - DB는 정규화된 형태(Normalized form)로 유지되어 데이터 중복을 최소화하고 데이터 무결성을 보장
- OLAP (Online Analytical Processing)
  - 복잡한 분석, 다차원 분석, 대규모 데이터셋에 대한 질의(querying) 및 보고에 사용되며 비즈니스 인텔리전스(BI), 데이터 마이닝, 보고서 작성 등에 초점
  - OLAP 시스템은 대량의 데이터에 대해 복잡한 쿼리를 실행하여 비즈니스 인사이트를 제공하는 데 최적화
  - 데이터를 종합, 집계, 계산하여 다양한 관점에서 데이터를 분석할 수 있게 합니다.
  - OLAP 데이터베이스는 종종 비정규화되거나 다차원 데이터 모델(예: 스타 스키마)을 사용하여 설계되어, 분석 쿼리의 성능을 최적화

OLTP vs OLAP

- OLAP(Online Analytical Processing) is used for data analysis and not data processing.
- OLTP(Online transaction processing) is used for data processing and not for data analysis.



https://ashutosh-bitmesra.medium.com/oltp-and-olap-what-are-the-differences-a6e21f25bfe0

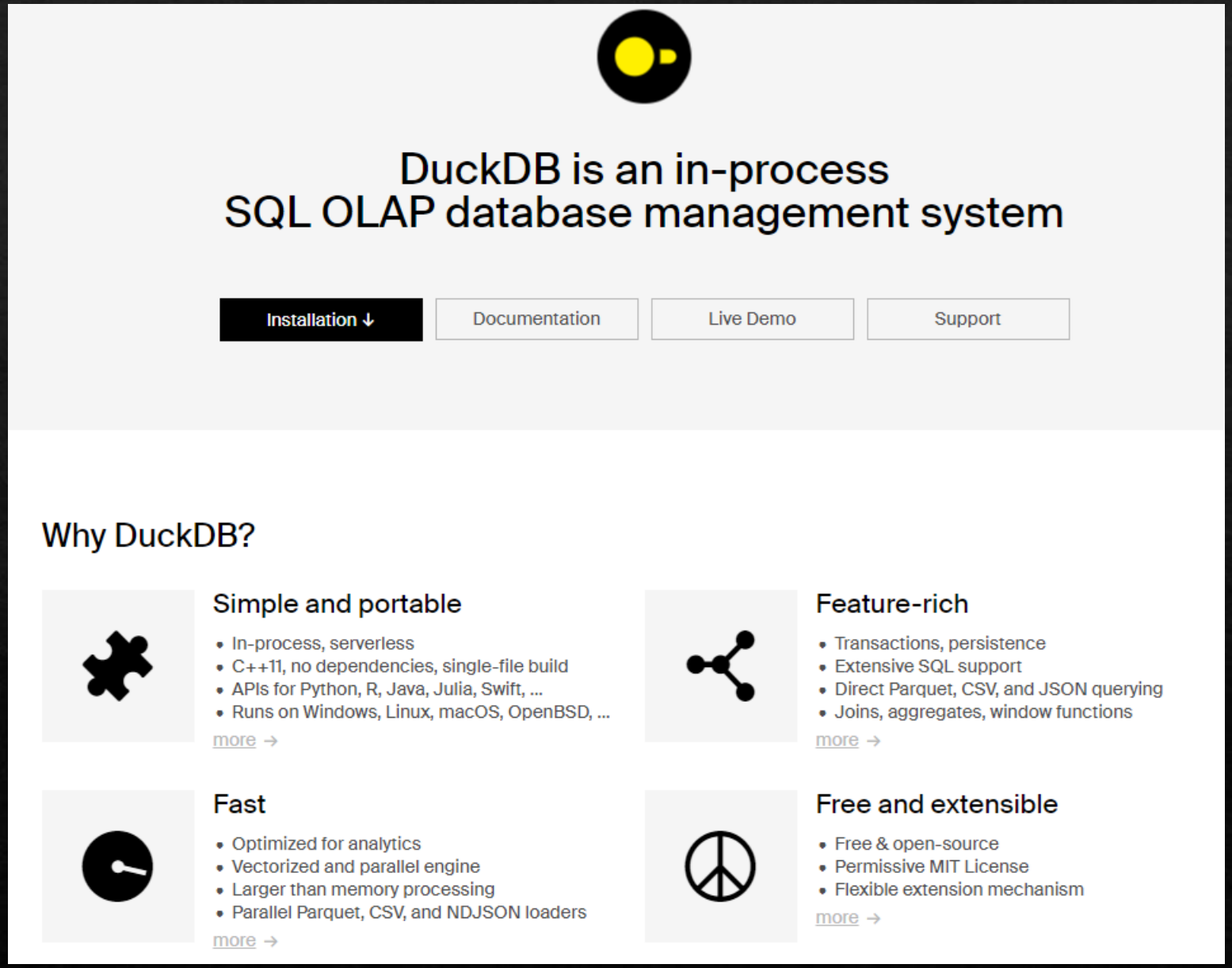
SQLite

- SQLite is a database engine written in the C programming language.
  - It is not a standalone app; rather, it is a library that software developers embed in their apps.
    - Python과 같이 사용하게 되면 파이썬 실행코드(주체)와 같은 process에서 작동(In-process)
  - As such, it belongs to the family of embedded databases.
    - MySQL, Postgres와 같은 DB의 Disk I/O와 통신하기 위해 Network I/O가 필요 없음(In-Memory query)
  - It is the most widely deployed database engine, as it is used by several of the top web browsers, operating systems, mobile phones, and other embedded systems.

SQLite는 MySQL나 PostgreSQL와 같은 데이터베이스 관리 시스템이지만, 서버가 아니라 응용 프로그램에 넣어 사용하는 비교적 가벼운 데이터베이스이다. 영어권에서는 '에스큐엘라이트'또는 '시퀄라이트'라고 읽는다. 위키백과

Duck DB

https://duckdb.org/

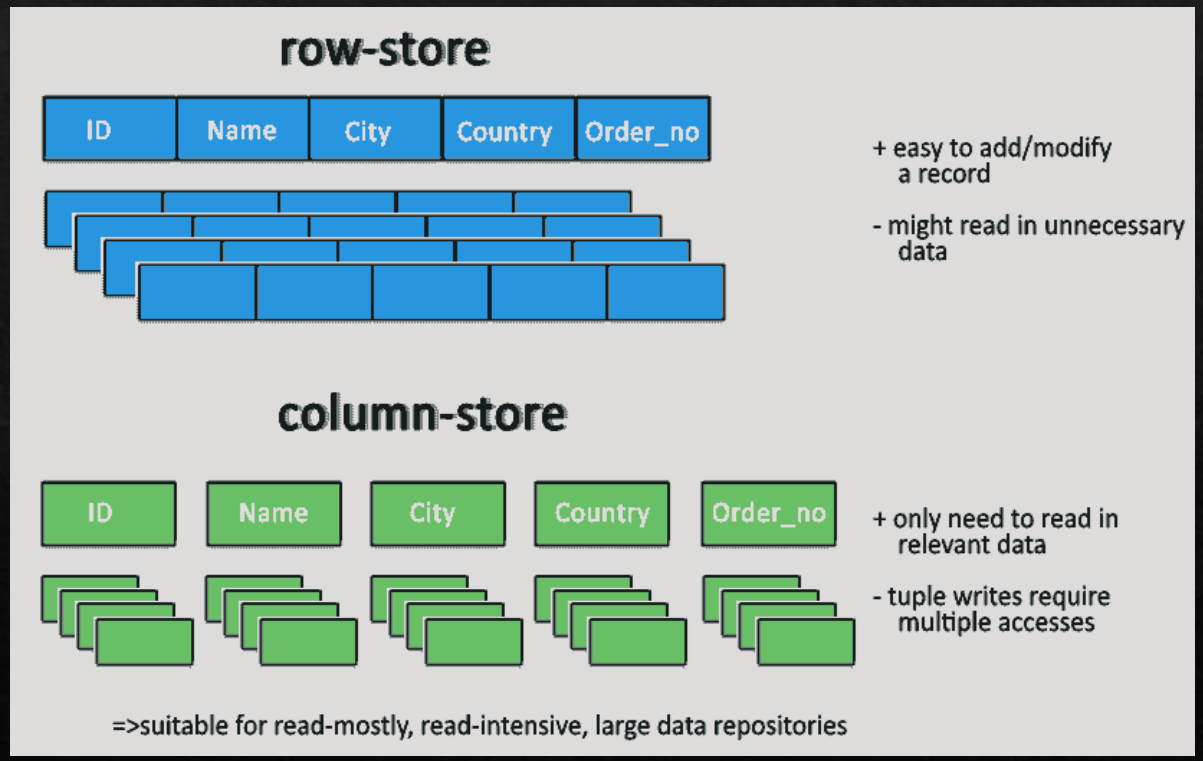


실습하기


PRODUCTION
DIRECTOR
CAMERA
DATE
SCENE
TAKE
Forget about SQLite, Use
DuckDB Instead

Is Duck DB a row-store or column-store?

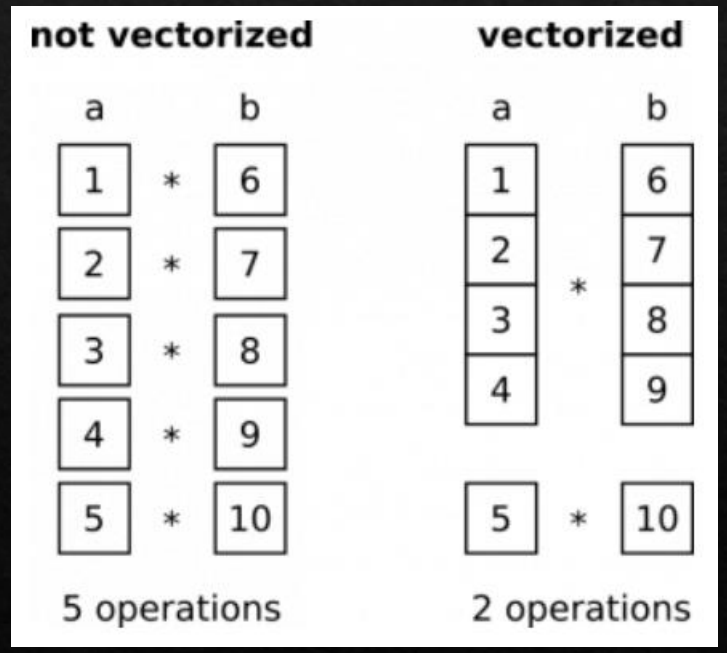
- Duck DB is a columnar database.
  - Duck DB stores data in columns rather than the rows used by traditional databases.
  - Columnar databases are designed to read data more efficiently and return queries with greater speed.

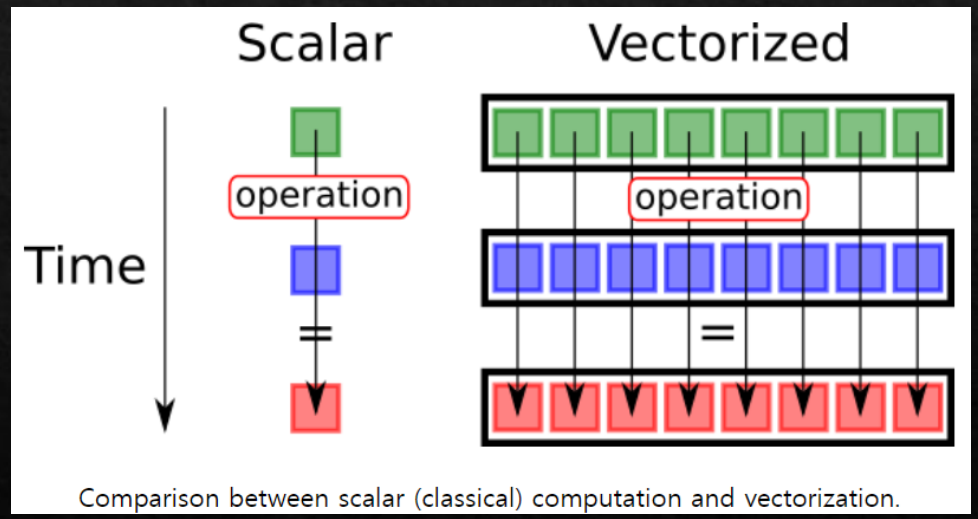


https://www.heavy.ai/technical-glossary/columnar-database

Is Duck DB a Vectorized processing?

- DuckDB is an embeddable In-process OLAP DBMS which runs on a single machine with no external dependency.
- DuckDB contains columnar-vectorized query execution engine for faster data processing.
  - by vectorizing query executions (columnar-oriented), while other DBMSs (SQLite, PostgreSQL…) process each row sequentially
- DuckDB doesn't have to be a Pandas substitute.

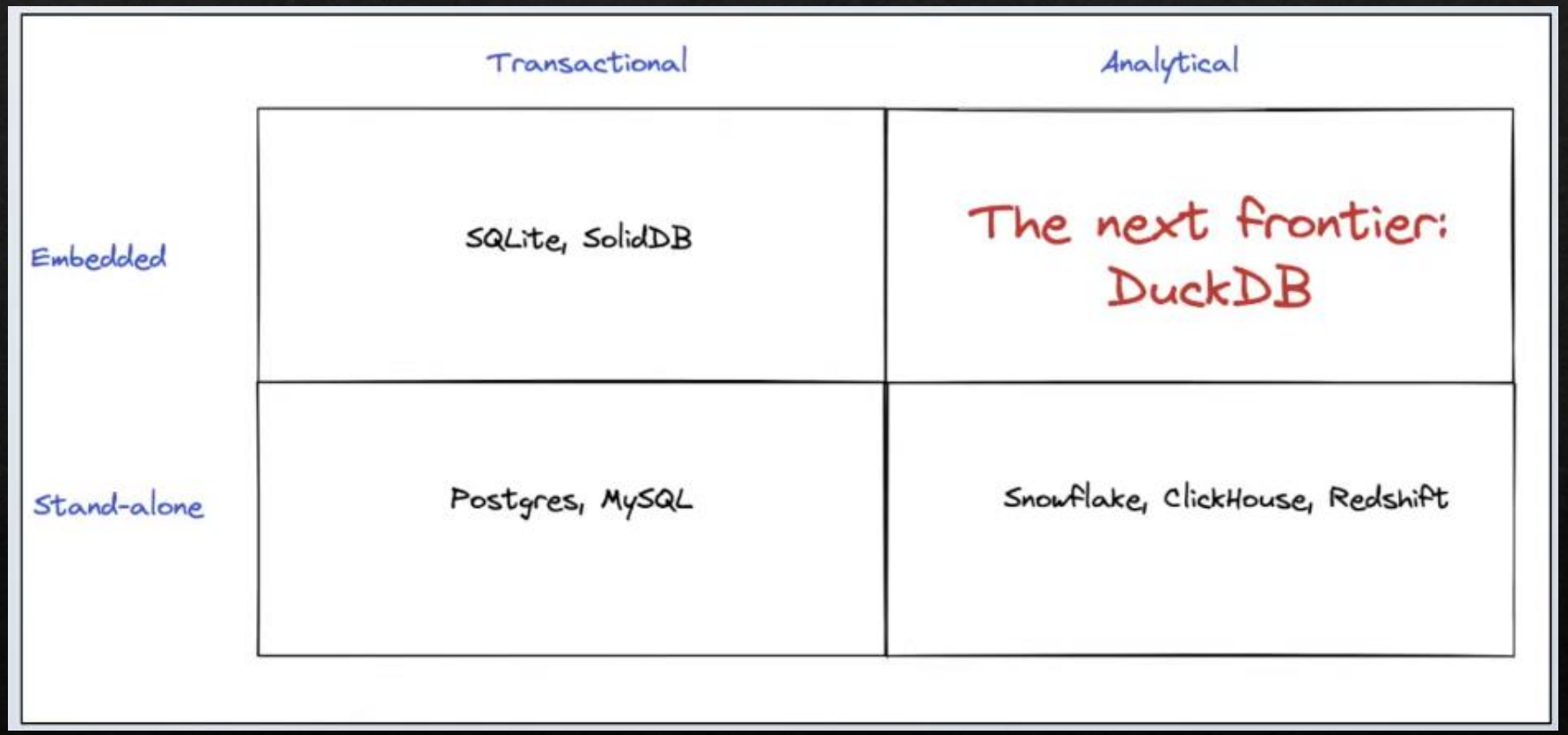




It is a copy-paste of the computing part of a CPU (ALU) to enable several computing at the same time.

https://lappweb.in2p3.fr/~paubert/ASTERICS.HPC/6-6-1-985.html

Duck DB


Transactional
Analytical
Embedded
Stand-alone
SQLite, SolidDB
The next frontier: DuckDB
Postgres, MySQL
Snowflake, ClickHouse, Redshift