Database 101

1. RDBMS란?

김용담 강사

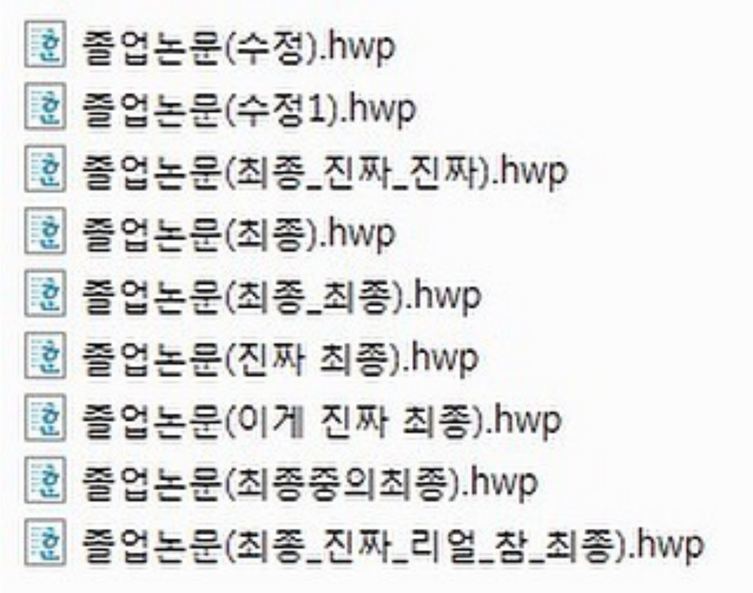
Contents

1

RDBMS란?

Intro

현실 세계에서 데이터를 관리하는 법.jpeg



1. 데이터베이스의 정의

데이터베이스의 정의

- 컴퓨터가 보급화 되면서 (1980's) 처리해야 할 데이터가 늘어남
- 여러 회사들에서 대규모 데이터를 관리하기 위해서 통합된 환경이 필요함
- 데이터들을 표현할 수 있는 다양한 모델이 등장
    - Relational Model, Network Model, Hierarchical Model, ...
- 데이터를 체계적으로 관리할 수 있는 공간을 "Database"로 정의함
- 데이터베이스를 관리하는 시스템을 "DataBase Management System(DBMS)"으로 정의함

1. 데이터베이스의 정의

데이터베이스를 사용하면 좋은 이유

- 데이터베이스라는 개념이 나타나기 이전엔, File 단위로 데이터를 관리
  - e.g. 윈도우 탐색기, macOS Finder
- 데이터를 효율적으로 관리하기 힘든 여러가지 문제들이 발생.
  - Redundancy, Inconsistency
  - Isolation, Integrity
  - Atomicity, Concurrency

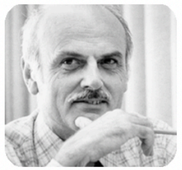
1. 데이터베이스의 정의

기존 시스템의 문제점

- 데이터가 여러 형태로 여기저기 저장되면서 통일성을 잃고, 통합되지 않음
- 데이터를 처리하기 위해서는 매번 application이 필요함
- 데이터의 상태를 파일 자체적으로 확인하기 힘듬 (열어봐야 알 수 있음)
- 여러 명의 사용자가 동시에 접근해서 사용하기 힘듬
- 파일의 일부가 수정되었을 때, 전체적인 데이터가 통일성을 유지하기 힘듬

1. 데이터베이스의 정의

Relational DataBase Management System



Ted Codd
Turing Award 1981

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

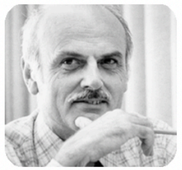
(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

1. 데이터베이스의 정의

테이블과 구성요소



Ted Codd
Turing Award 1981

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

1. 데이터베이스의 정의

Structured Query Language(SQL)

```
create table classroom
    (building       varchar(15),
     room_number        varchar(7),
     capacity       numeric(4,0),
     primary key (building, room_number)
    );

create table department
    (dept_name      varchar(20),
     building       varchar(15),
     budget             numeric(12,2) check (budget > 0),
     primary key (dept_name)
    );

create table course
    (course_id      varchar(8),
     title          varchar(50),
     dept_name      varchar(20),
     credits        numeric(2,0) check (credits > 0),
     primary key (course_id),
     foreign key (dept_name) references department (dept_name)
        on delete set null
    );
```

MySQL

```
# Write your MySQL query statement below
select name, population, area
from World as w
where (w.area >= 3000000) or (w.population >= 25000000)
```

MySQL

```
# Write your MySQL query statement below
select c.name as Customers
from Customers as c
where c.id not in (select customerId from Orders)
```

1. 데이터베이스의 정의

Cursor(커서)

- Python, Java, C++ 같은 다른 언어를 사용하여, 절차적(Procedural)으로 명령을 수행하고 싶을 때 사용하는 개념.
- SELECT 등에 의한 데이터베이스 검색 실행 결과를 한 줄씩 처리하기 위해서 데이터베이스 서버 측의 결과와 행 위치 등을 나타내는 개념
- 커서 사용 방법은 다음과 같습니다.

1. DECLARE CURSOR
2. OPEN
3. FETCH
4. UPDATE
5. DELETE
6. CLOSE

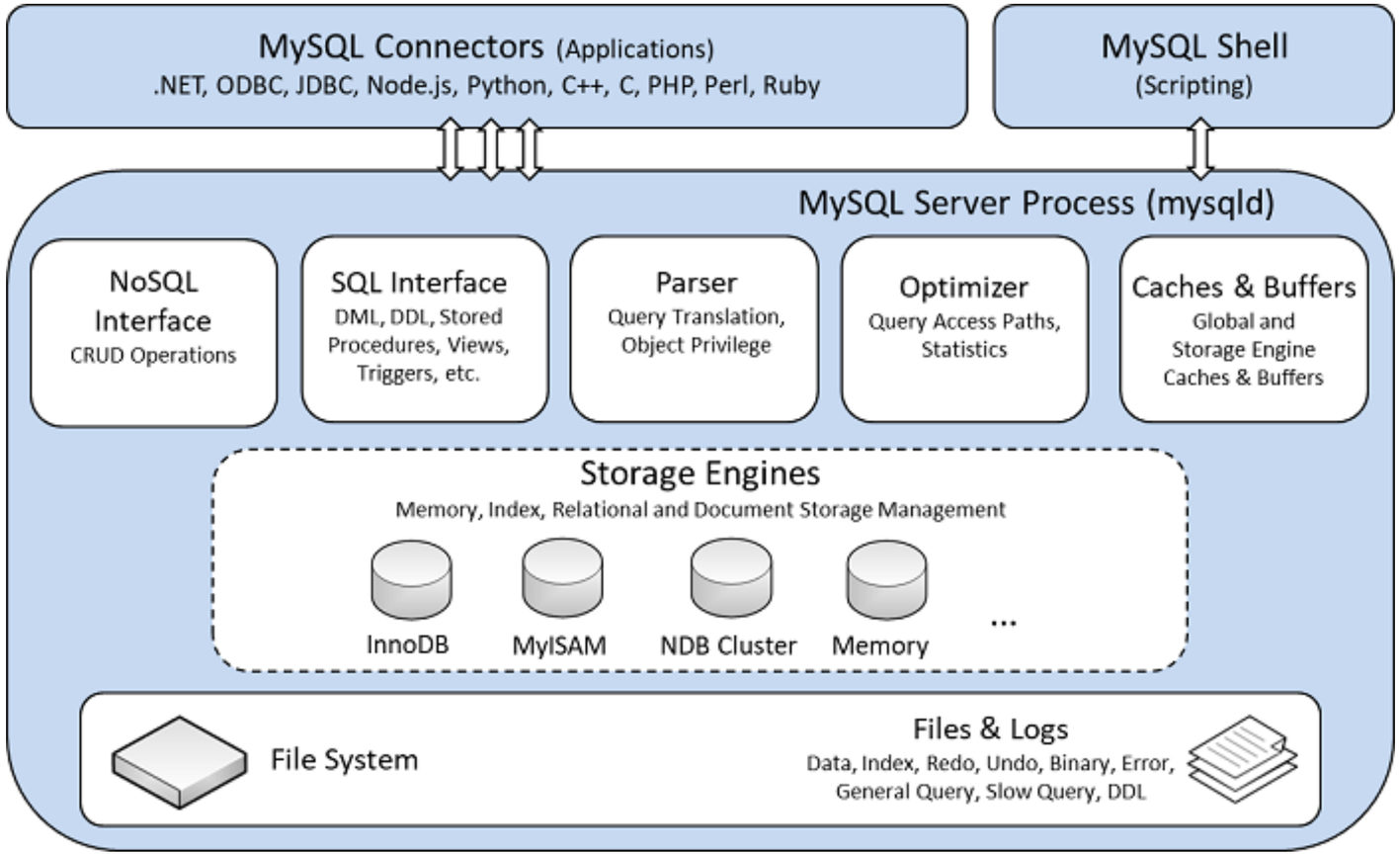
1. 데이터베이스의 정의

DBMS

- DBMS도 결국 컴퓨터 시스템.
- 사용자로부터 입력받은 query(질의)를 내부 시스템이 이해하고 처리함.
- SQL을 시스템적으로 이해하고 성능 개선을 하는 "튜닝" 작업도 있음.
- 데이터베이스의 연구는 DBMS 내부의 여러가지 개념들을 고도화하는데에 있음.
  e.g. index, query optimization, file system, ...
- DBMS마다 시스템의 정의와 목적이 다름.

1. 데이터베이스의 정의

e.g. MySQL DBMS

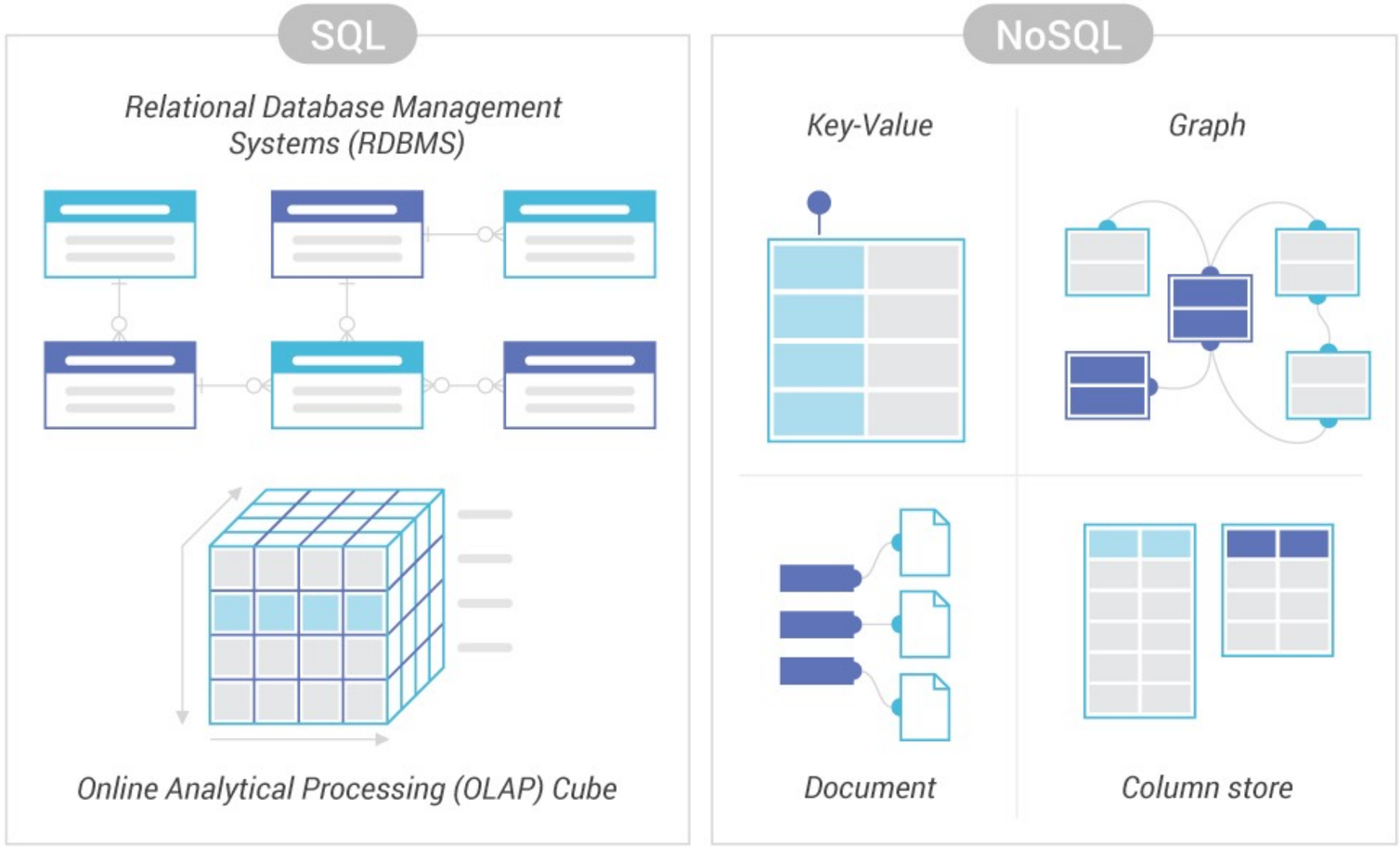


2

RDBMS(SQL) vs NoSQL

2. SQL vs NoSQL

데이터베이스의 종류 구분



Source : https://www.scylladb.com/learn/nosql/nosql-vs-sql/

2. SQL vs NoSQL

SQL(RDBMS)

- 정형 데이터(Structured Data, Panel Data, Tabular Data)를 처리하기 좋음
- attributes들로 정의되기 쉬운 데이터에 대해서 효과적으로 표현 가능
- 테이블 형태로 표현되면 처리 속도가 굉장히 빠름
- 대표적으로 Oracle DBMS, MySQL, PostgreSQL, MariaDB 등이 있음

2. SQL vs NoSQL

MySQL : 세계에서 가장 많이 사용되는 오픈소스 RDBMS

- Oracle 회사가 관리하고 있는 오픈소스 RDBMS
- 1995년에 처음 발표되었으며, 현재(2023년 9월 기준) 최신 버젼은 8.1.0
- C, C++로 개발되었으며 대부분의 운영체제를 지원함
- MySQL 자체를 설치해서 Shell 환경에서도 운영가능하나, 보통 MySQL Workbench를 사용하여 관리함
- PyMySQL 라이브러리를 사용하면 Python에서도 사용 가능

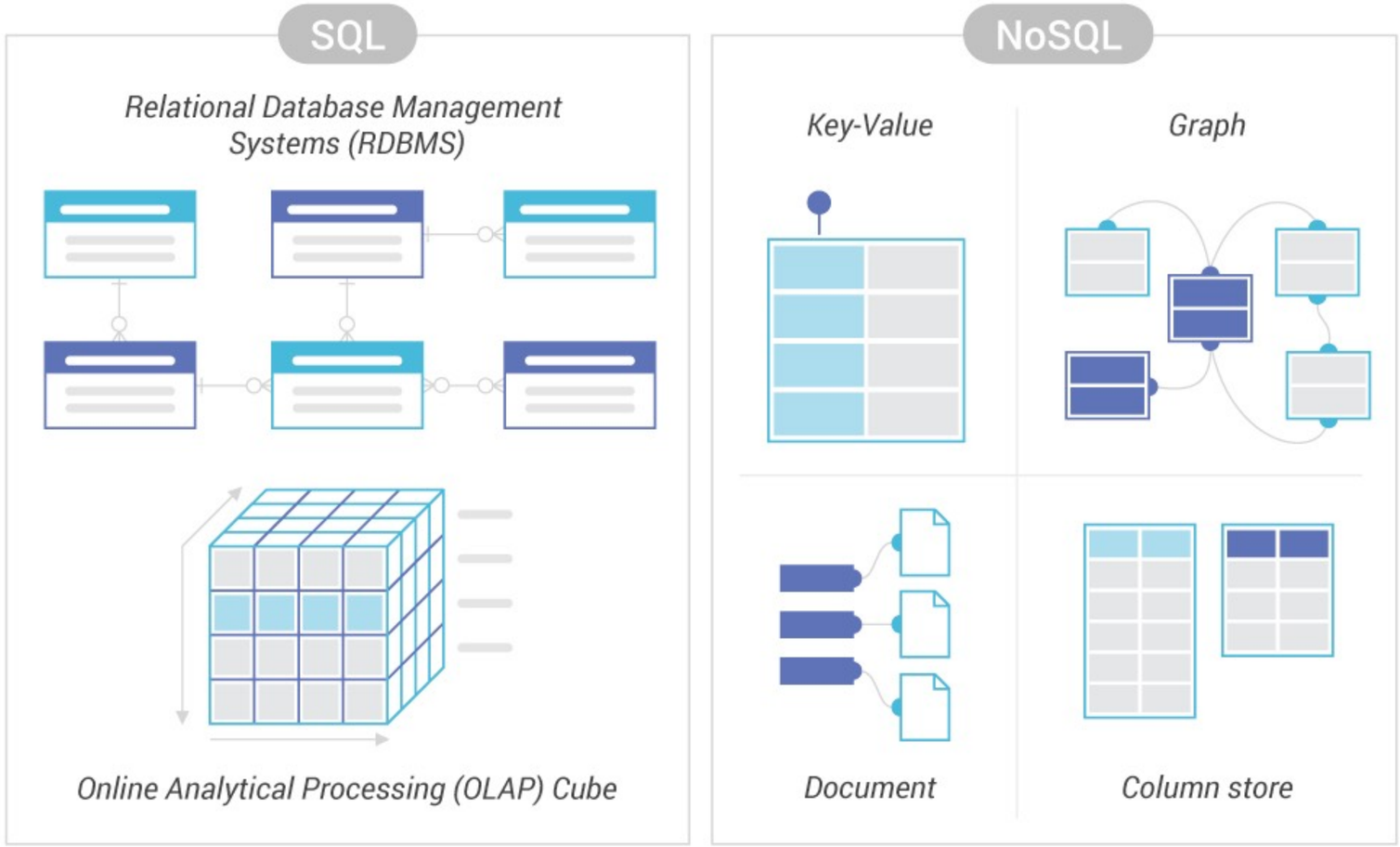
2. SQL vs NoSQL

MySQL을 이용한 enterprise 서비스들

- 1. TOSS : https://www.mysql.com/why-mysql/case-studies/toss-bank-financial-services-mysql-enterprise-edition.html
- 2. SSG : https://www.mysql.com/why-mysql/case-studies/ssg-com-online-shopping-mysql-enterprise-edition.html
- 3. LINE : https://www.mysql.com/why-mysql/case-studies/line-database-availability-scalability-security-mysql.html
- 4. 카카오뱅크 : https://www.bloter.net/newsView/blt201908300010

2. SQL vs NoSQL

데이터베이스의 종류 구분



Source : https://www.scylladb.com/learn/nosql/nosql-vs-sql/

2. SQL vs NoSQL

NoSQL(non-RDBMS)

- 비정형 데이터(Image, Text, Audio, Video, Graph 등)를 처리하기 좋음
- attributes들로 명확하게 정의되기 힘든 데이터들을 처리하기 좋음 (Schema-less)
- 주로 Key-Value, Document, Column-Family, Graph 방식을 지원함
- 대표적으로 Redis, MongoDB, Kafka, Cassandra 등이 있음

2. SQL vs NoSQL

redis : Key-value 방식의 효과적인 in-memory DBMS

- Redis-Lab에서 관리하고 있는 오픈소스 NoSQL DBMS
- 2009년에 처음 발표되었으며, AWS에서 엄청나게 많이 사용됨.
- 주로 Cache server나, Session 관리용도로 사용됨.
- 대표적인 In-memory DBMS로 엄청나게 빠른 속도로 데이터 처리가 가능함.

2. SQL vs NoSQL

redis use-cases

Use cases

Real-time data store

Redis' versatile in-memory data structures enable building data infrastructure for real-time applications that require low latency and high-throughput.

Caching & session storage

Redis' speed makes it ideal for caching database queries, complex computations, API calls, and session state.

Streaming & messaging

The stream data type enables high-rate data ingestion, messaging, event sourcing, and notifications.

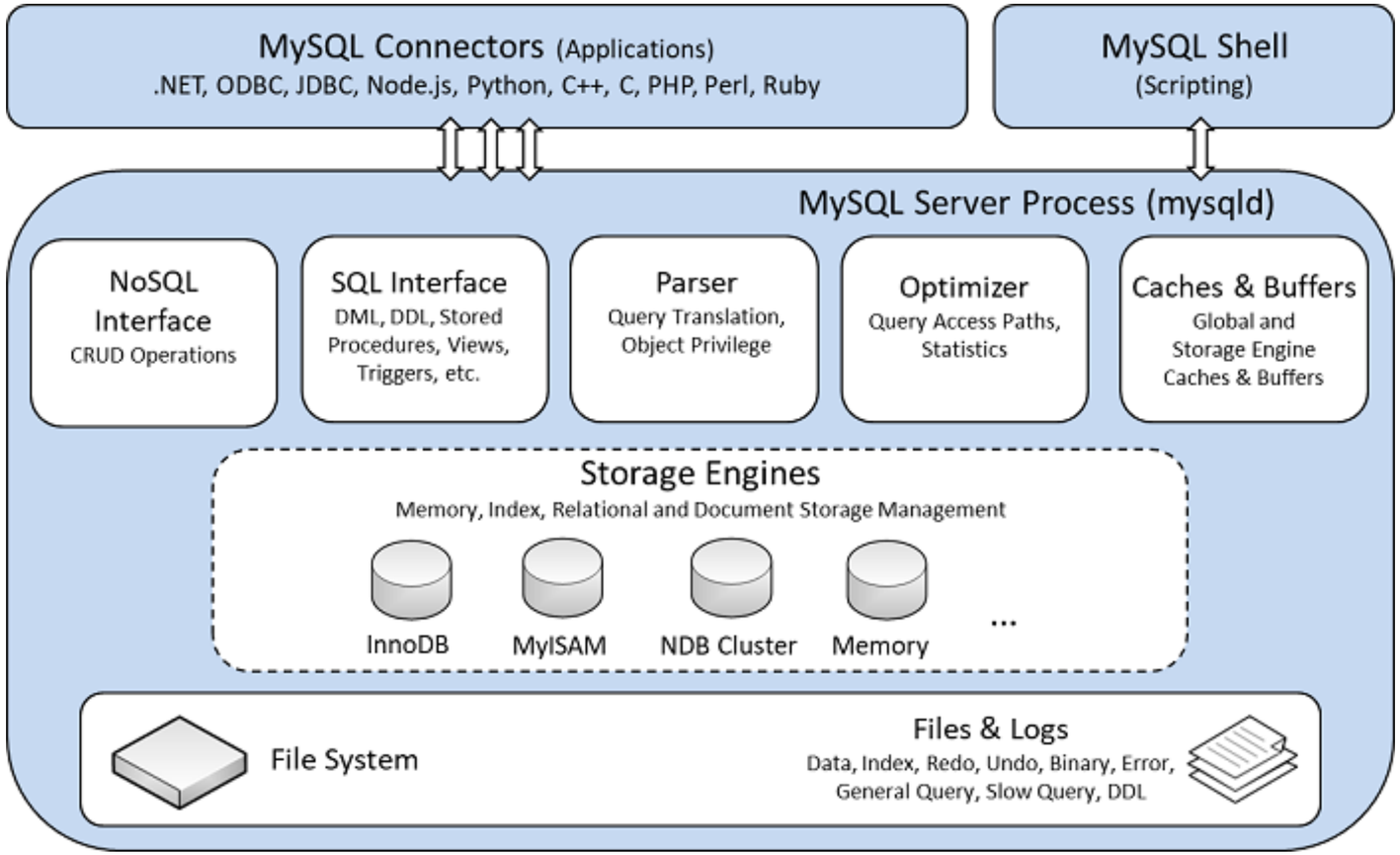
Source : https://redis.io/

3

RDBMS를 만들기 위한 요소들

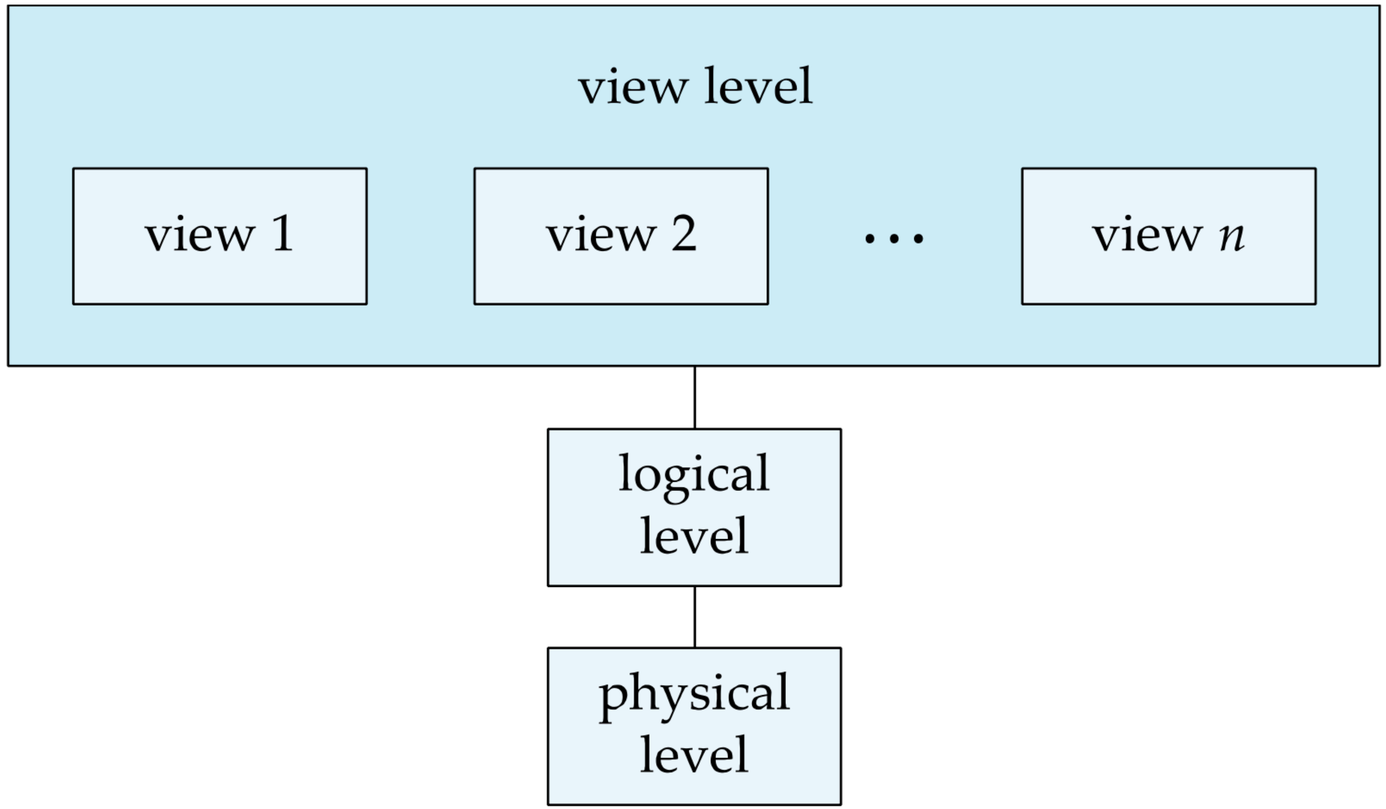
3. RDBMS를 만들기 위한 요소들

MySQL DBMS



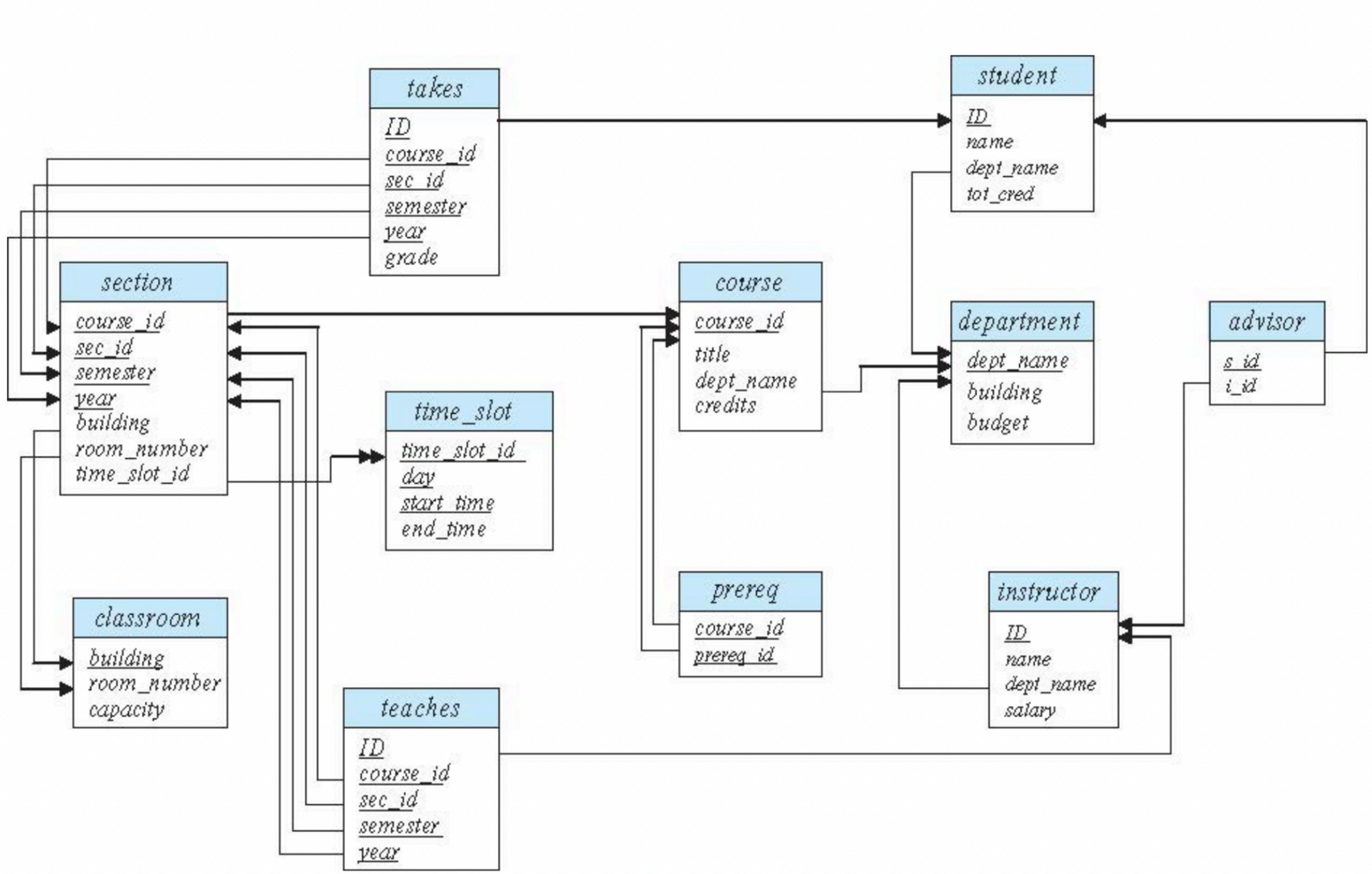
3. RDBMS를 만들기 위한 요소들

RDBMS Modeling



3. RDBMS를 만들기 위한 요소들

RDBMS Modeling



End of Slides