# SCHEMA & CONNECTION POOL

DATABASE STUDY

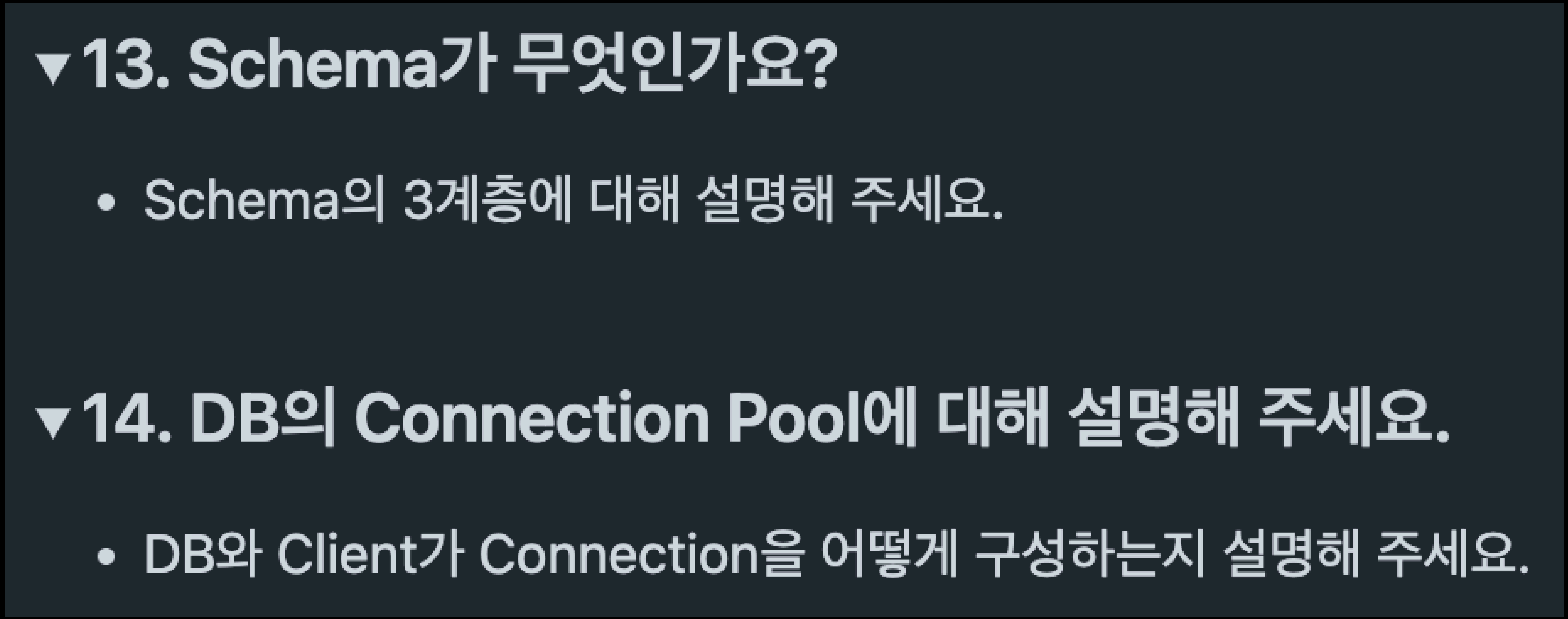
DongMin Kim

# CONTENTS

- Schema
    - Three schema architecture
    - Data independent

- Connection Pool
    - HikariCP
    - Formula

# GOALS



### ▼13. Schema가 무엇인가요?

- Schema의 3계층에 대해 설명해 주세요.

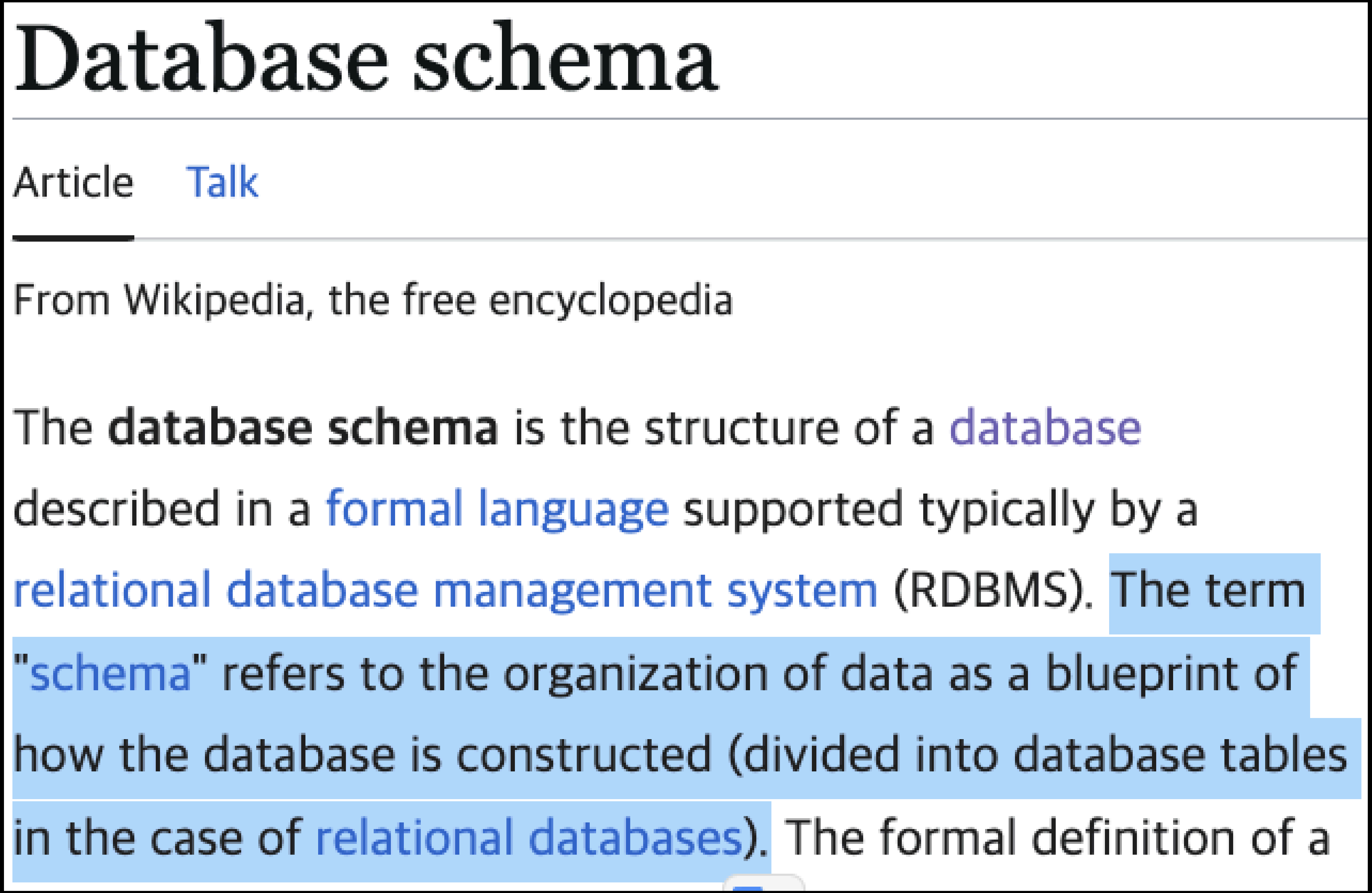
### ▼14. DB의 Connection Pool에 대해 설명해 주세요.

- DB와 Client가 Connection을 어떻게 구성하는지 설명해 주세요.

# Schema

σχήμα, schēma

# Schema



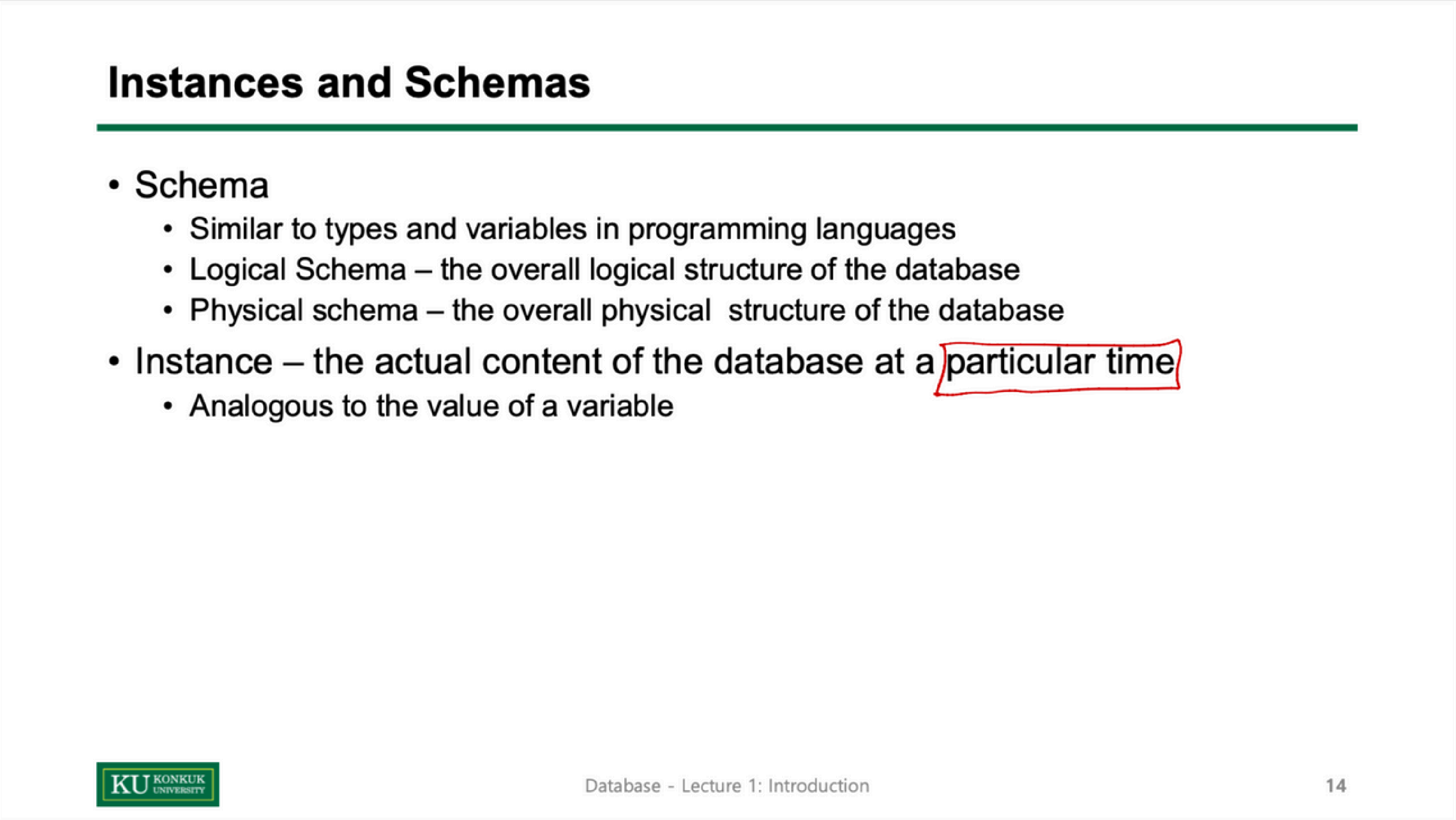
# Database schema

Article Talk

From Wikipedia, the free encyclopedia

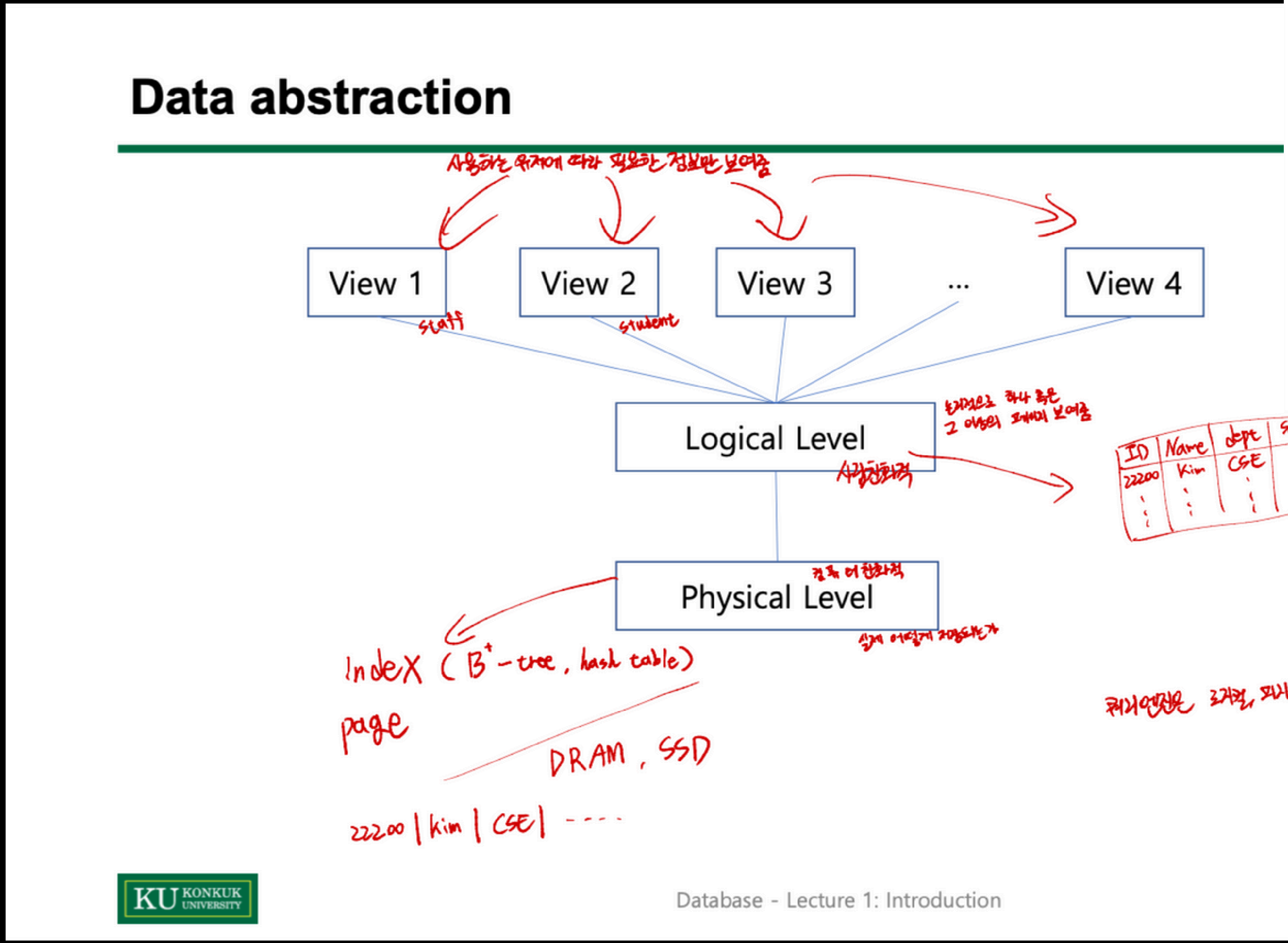
The **database schema** is the structure of a database described in a formal language supported typically by a relational database management system (RDBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a

# Schema



## Instances and Schemas

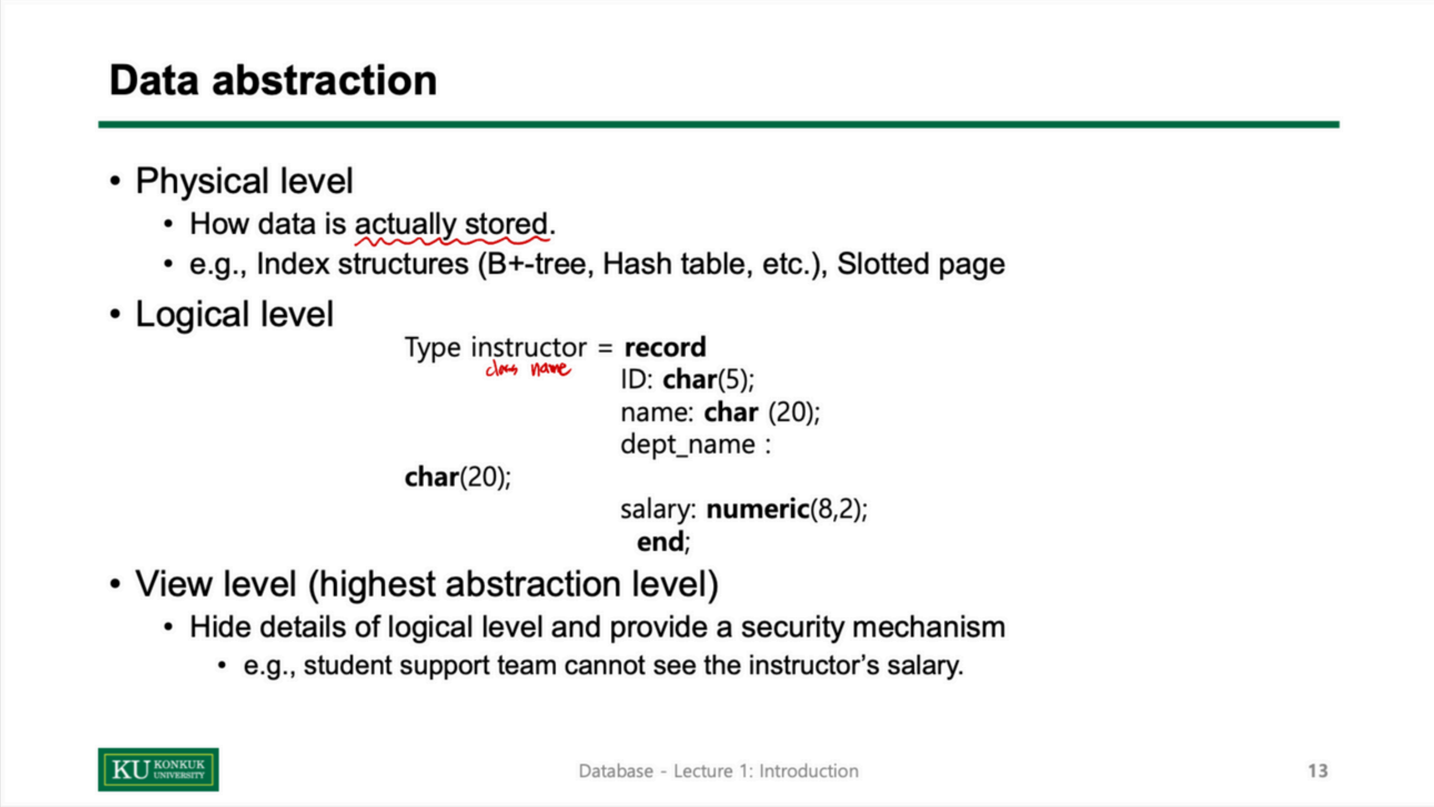
- Schema
  - Similar to types and variables in programming languages
  - Logical Schema – the overall logical structure of the database
  - Physical schema – the overall physical structure of the database
- Instance – the actual content of the database at a particular time
  - Analogous to the value of a variable



## Data abstraction

사용하는 유저에 따라 필요한 정보만 보여줌
View 1
View 2
View 3
…
View 4
staff
student
Logical Level
논리적으로 하나 혹은 그 이상의 테이블 보여줌
사람친화적
Physical Level
컴퓨터 친화적
실제 어떻게 저장되는가
Index ( B⁺-tree , hash table)
page
DRAM , SSD
22200 | kim | CSE | - - - -
쿼리언어는 로지컬, 피지컬 둘다 관여

| ID | Name | dept | Salary |
|---|---|---|---|
| 22200 | Kim | CSE | ⋮ |
| ⋮ | ⋮ | ⋮ | ⋮ |



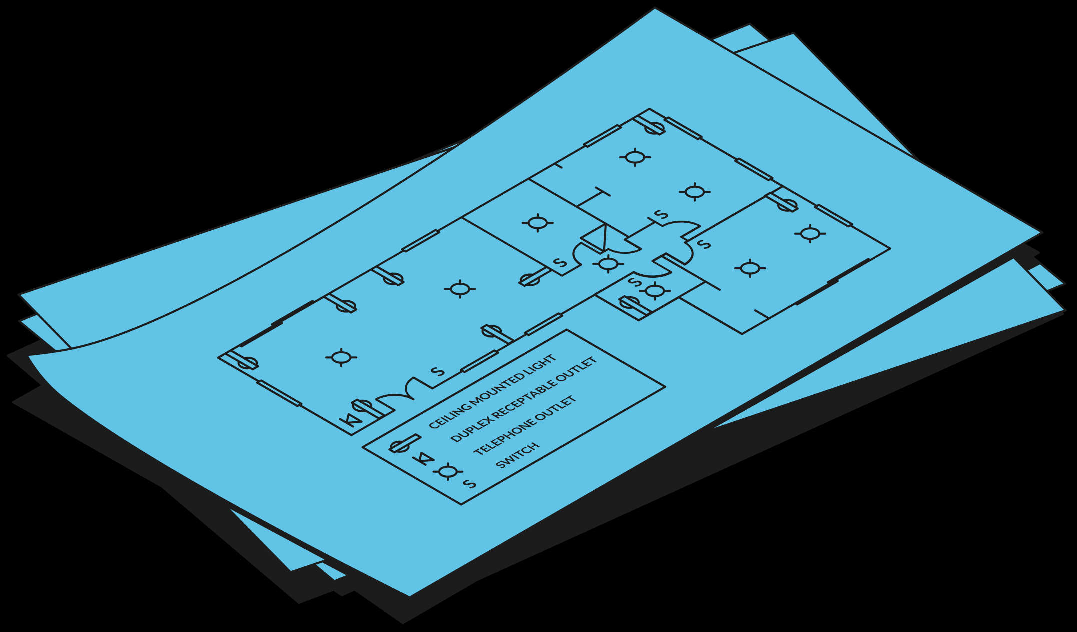
## Data abstraction

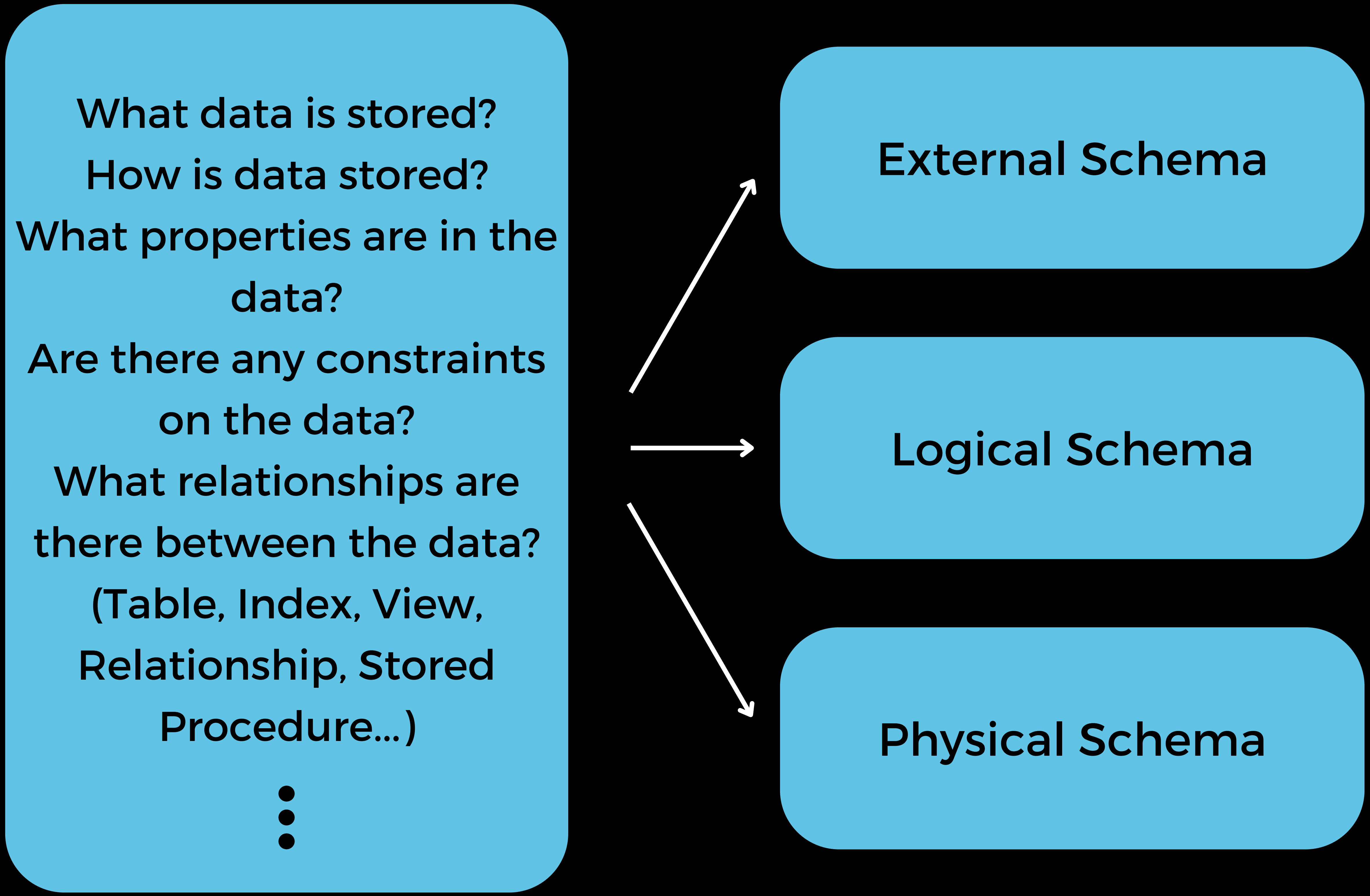
- Physical level
  - How data is actually stored.
  - e.g., Index structures (B+-tree, Hash table, etc.), Slotted page
- Logical level

```
Type instructor = record
  class name
                    ID: char(5);
                    name: char (20);
                    dept_name :
char(20);
                    salary: numeric(8,2);
                     end;
```

- View level (highest abstraction level)
  - Hide details of logical level and provide a security mechanism
    - e.g., student support team cannot see the instructor's salary.

# Schema





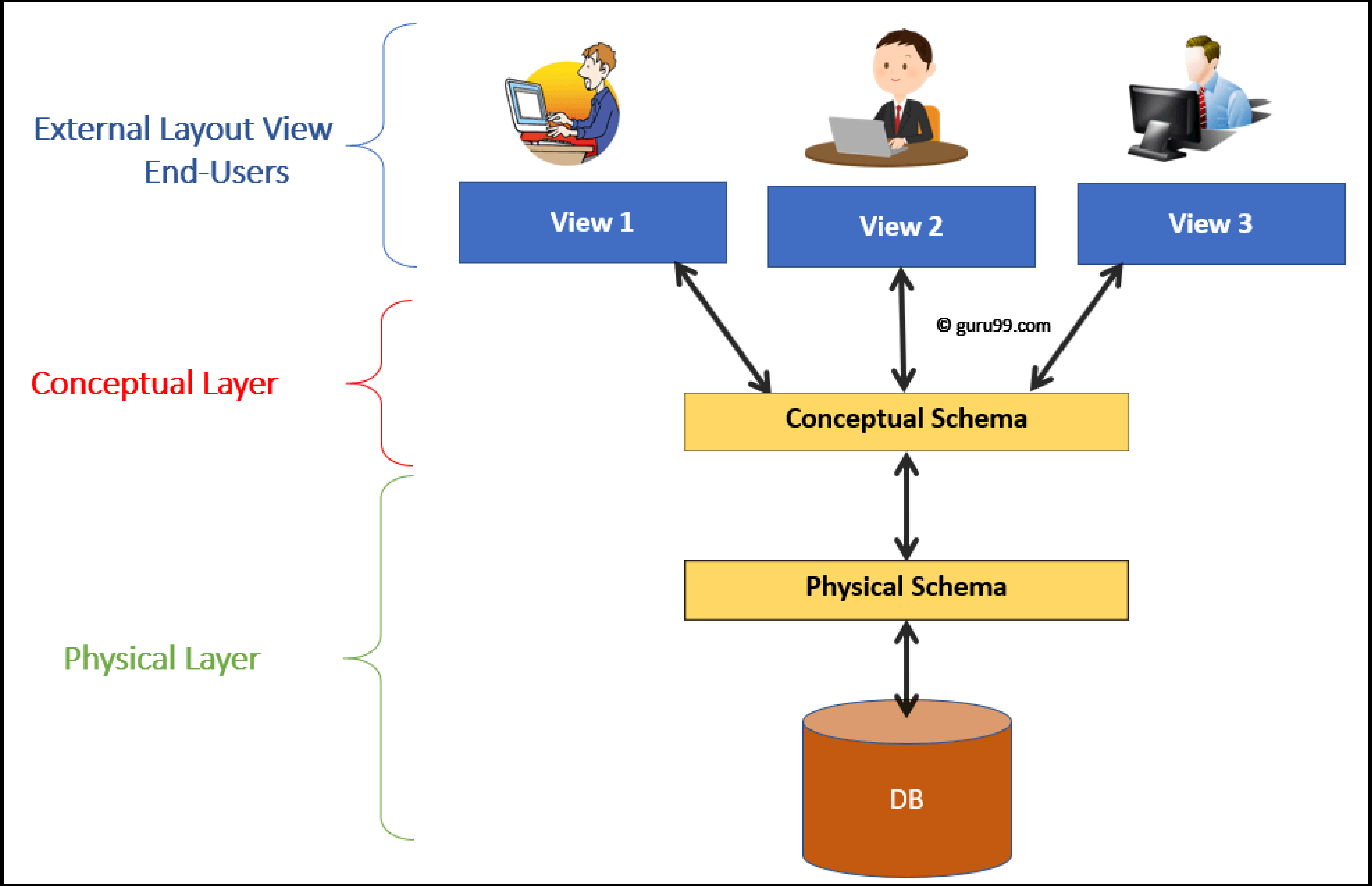
What data is stored?
How is data stored?
What properties are in the data?
Are there any constraints on the data?
What relationships are there between the data?
(Table, Index, View, Relationship, Stored Procedure...)
⋮

External Schema

Logical Schema

Physical Schema

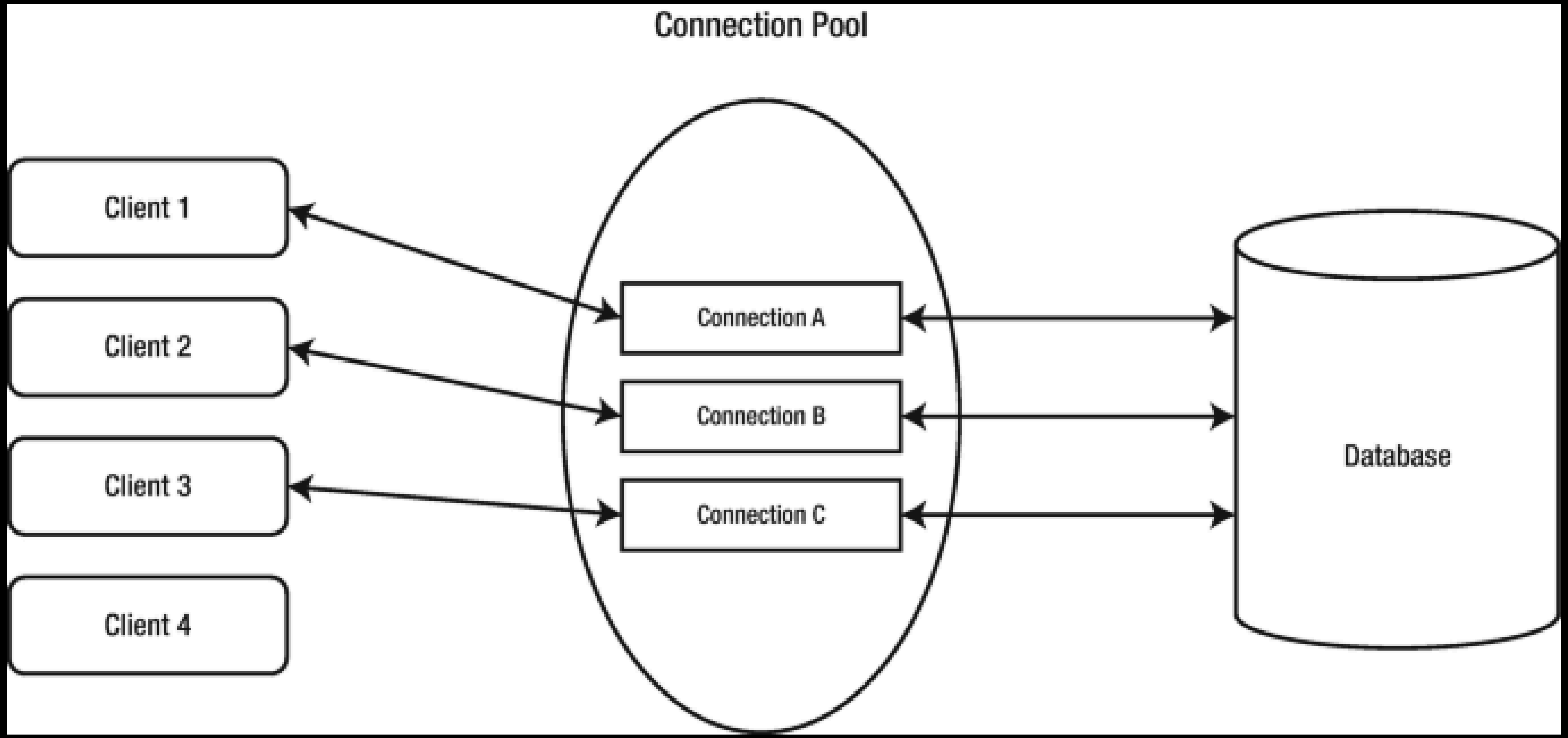
# Schema



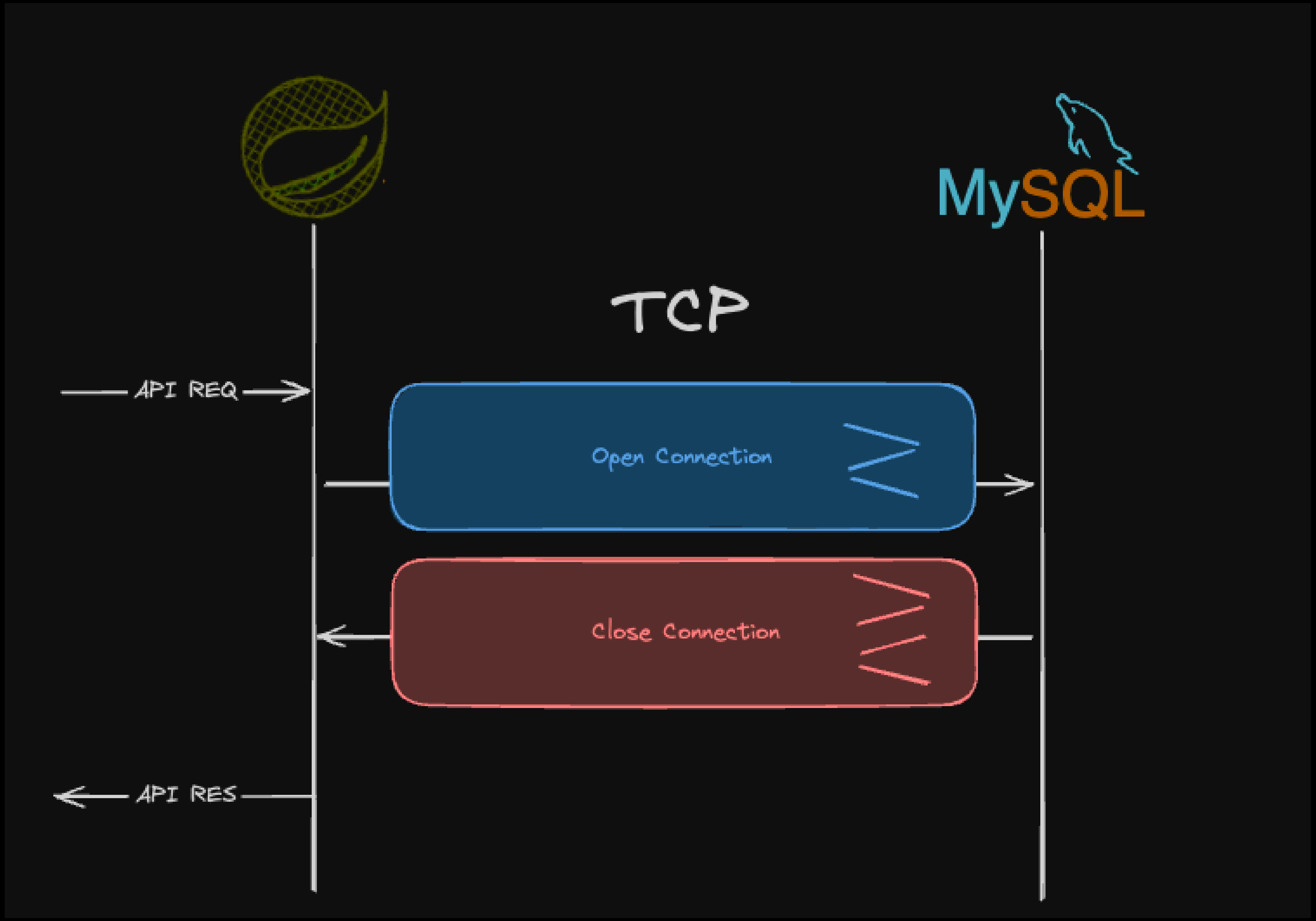
© guru99.com

# Connection Pool

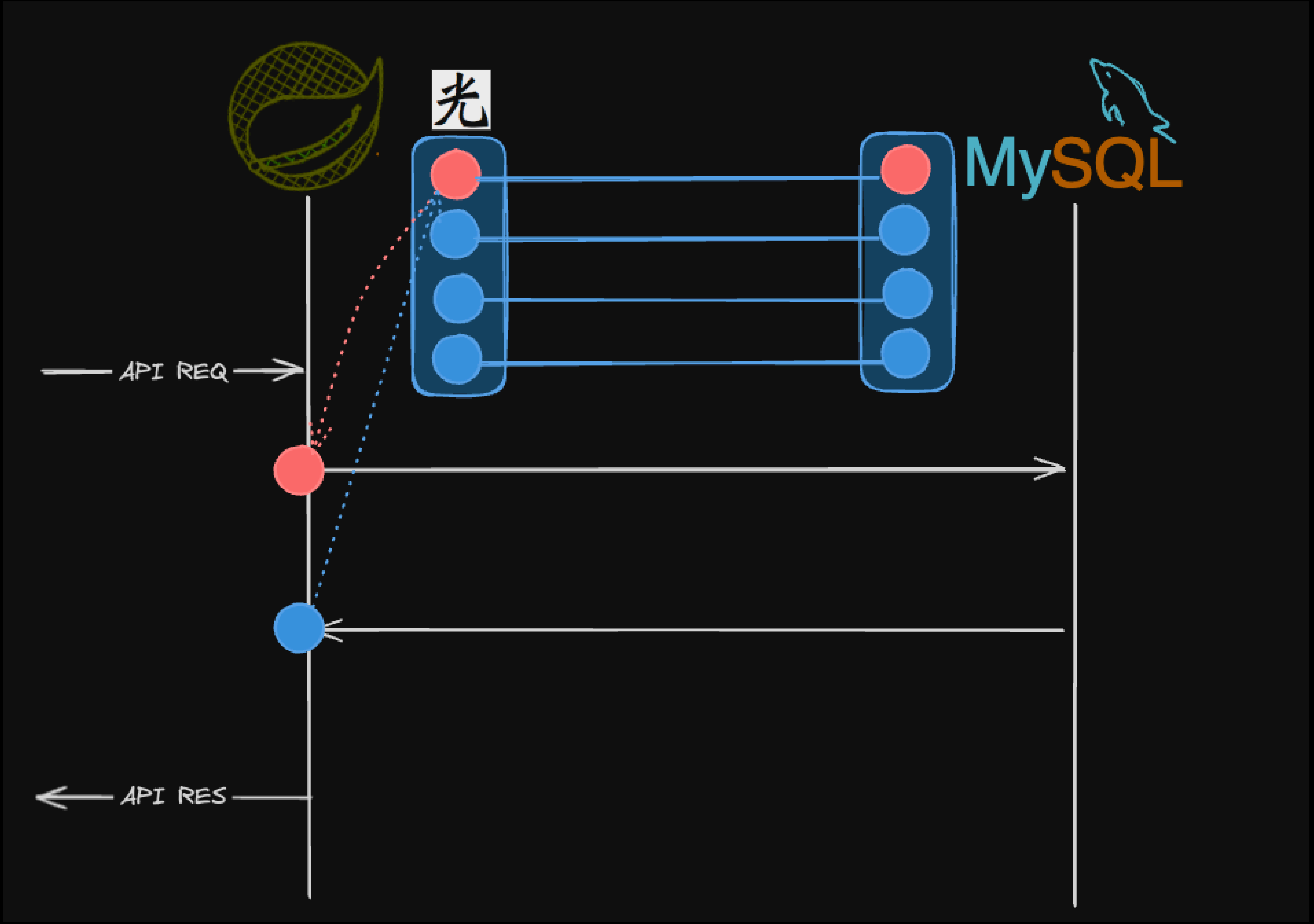
Efficiency



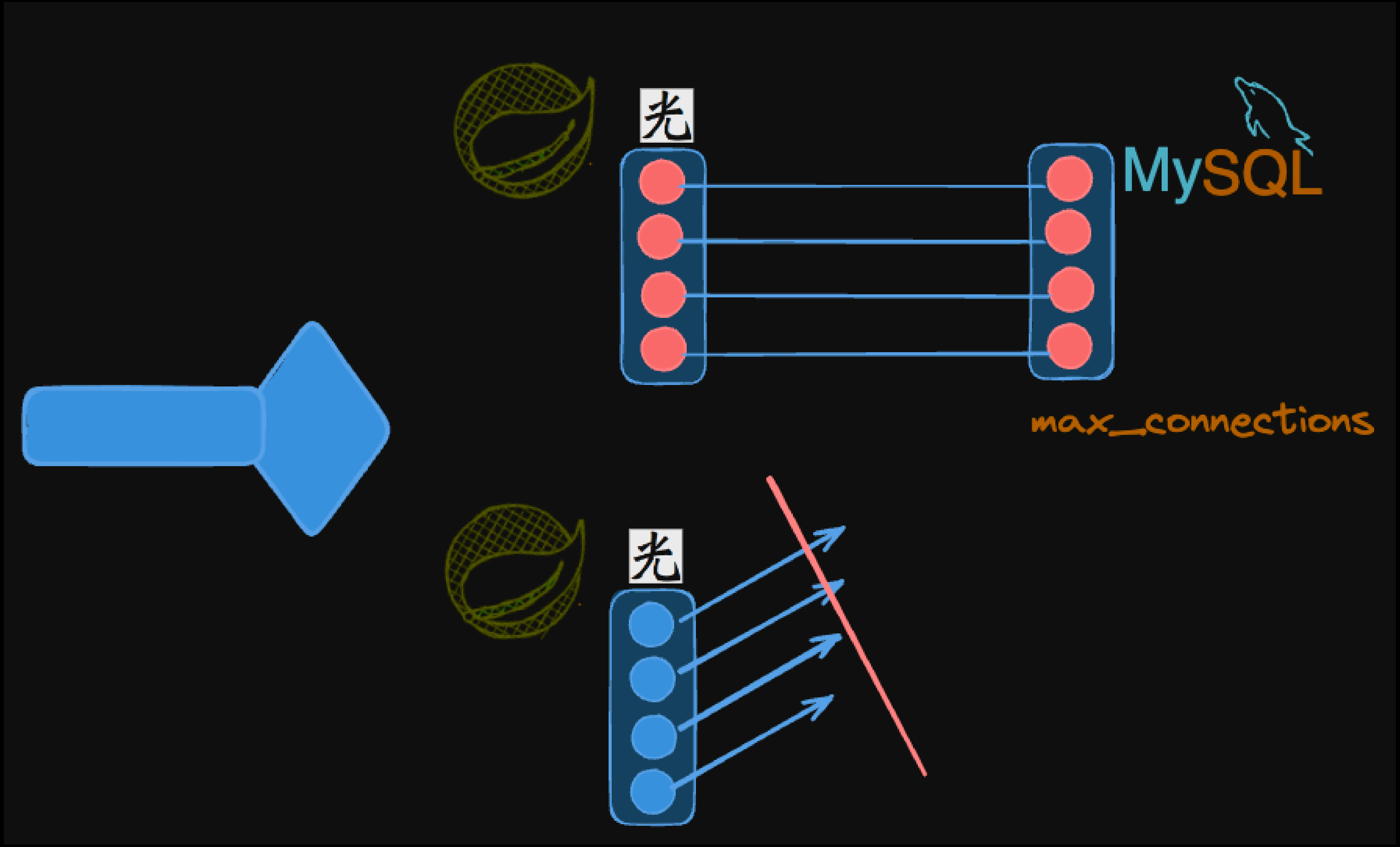
# Connection Pool



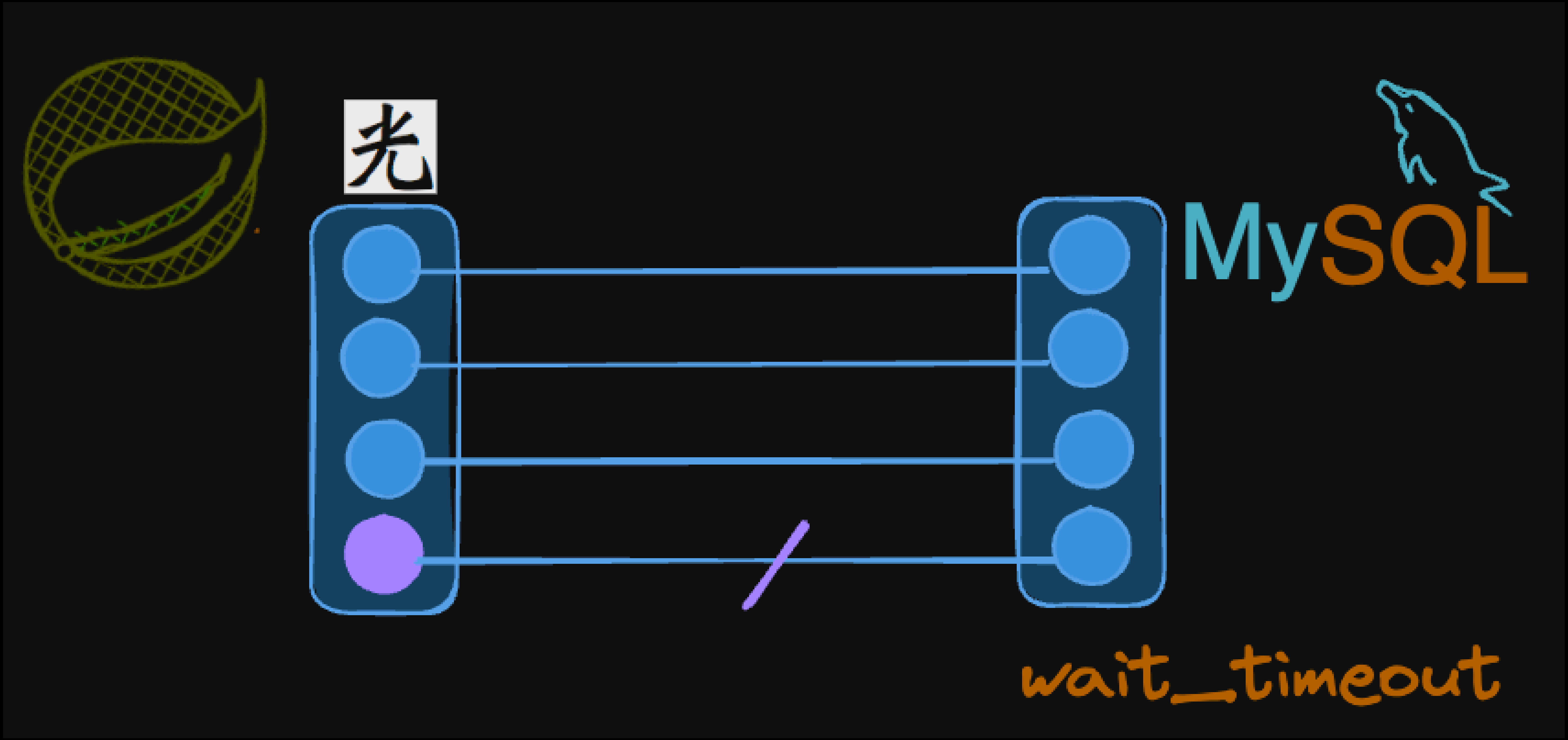
# Connection Pool



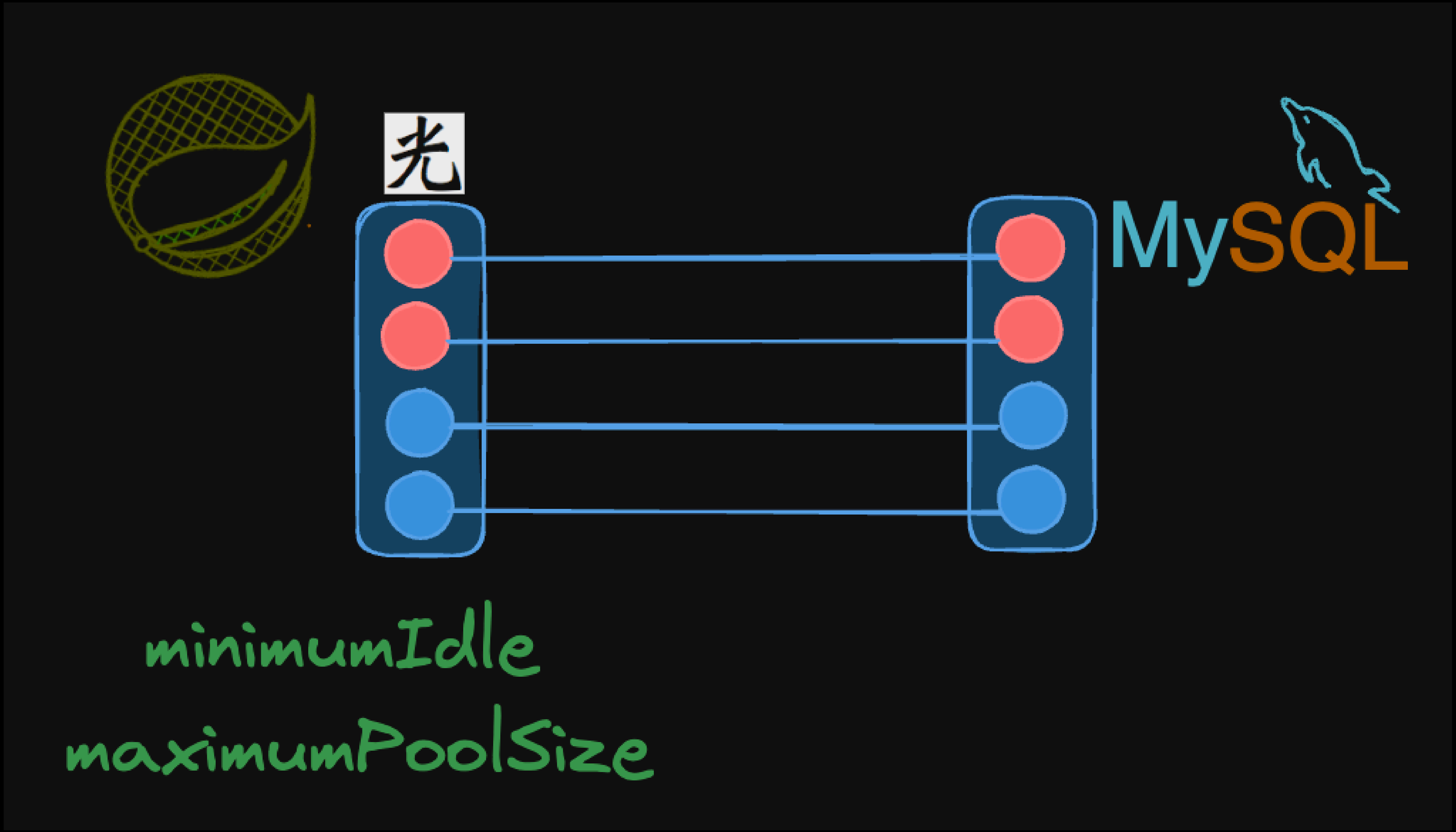
# Connection Pool



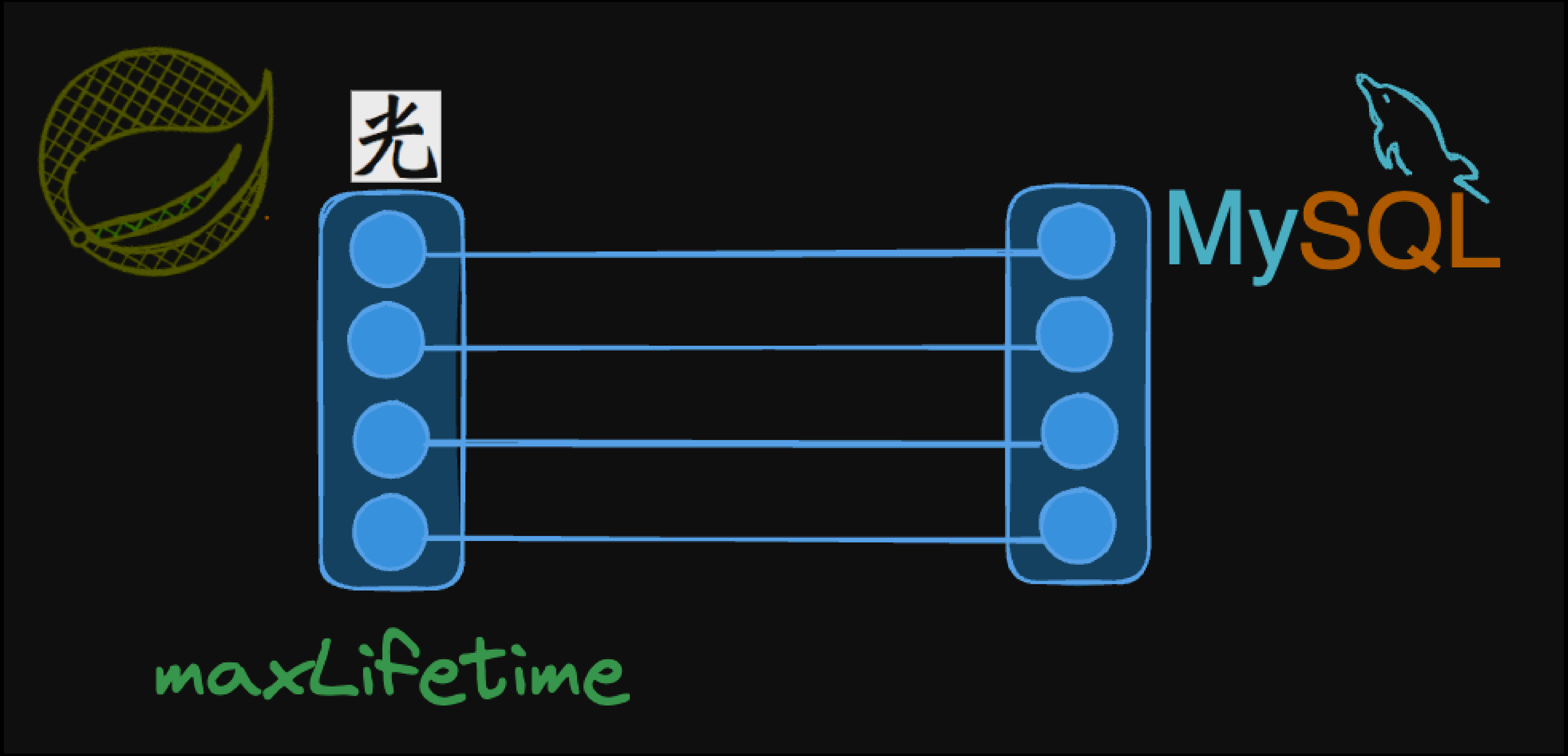
# Connection Pool



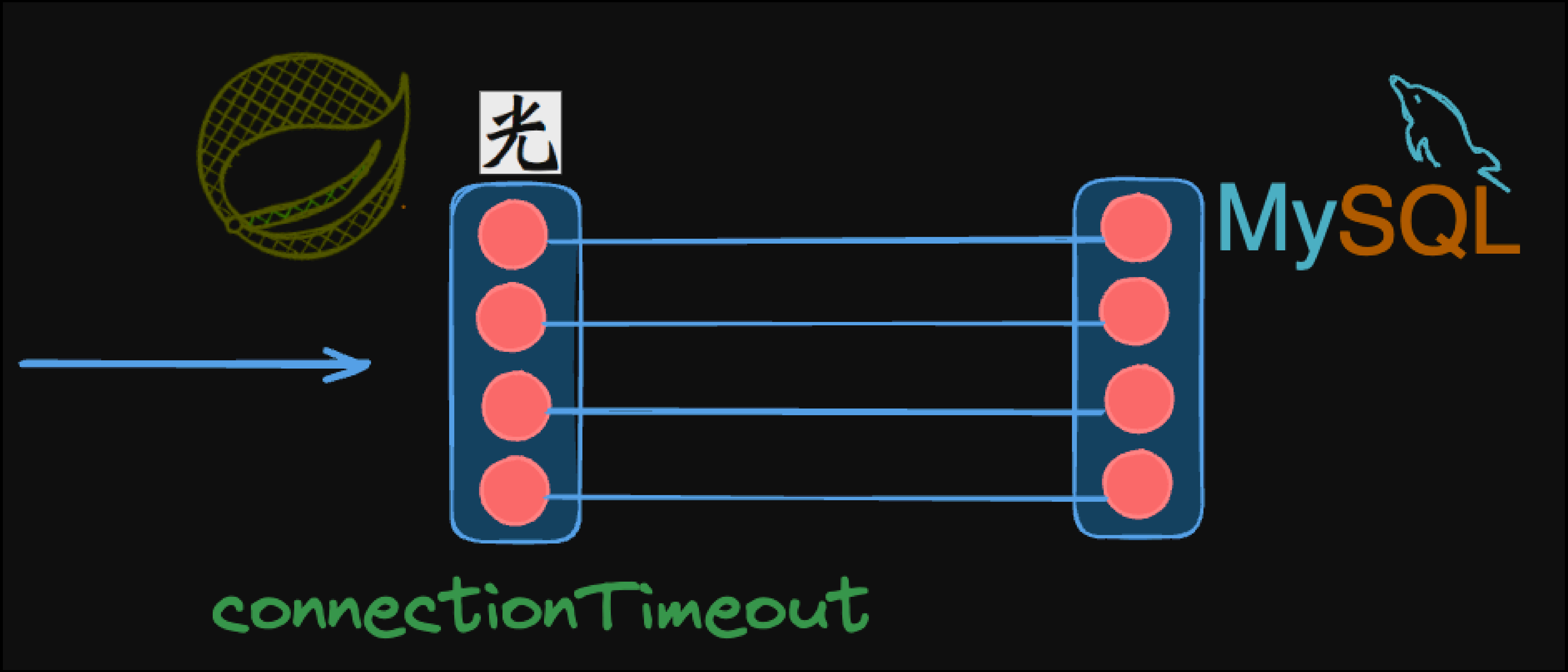
# Connection Pool



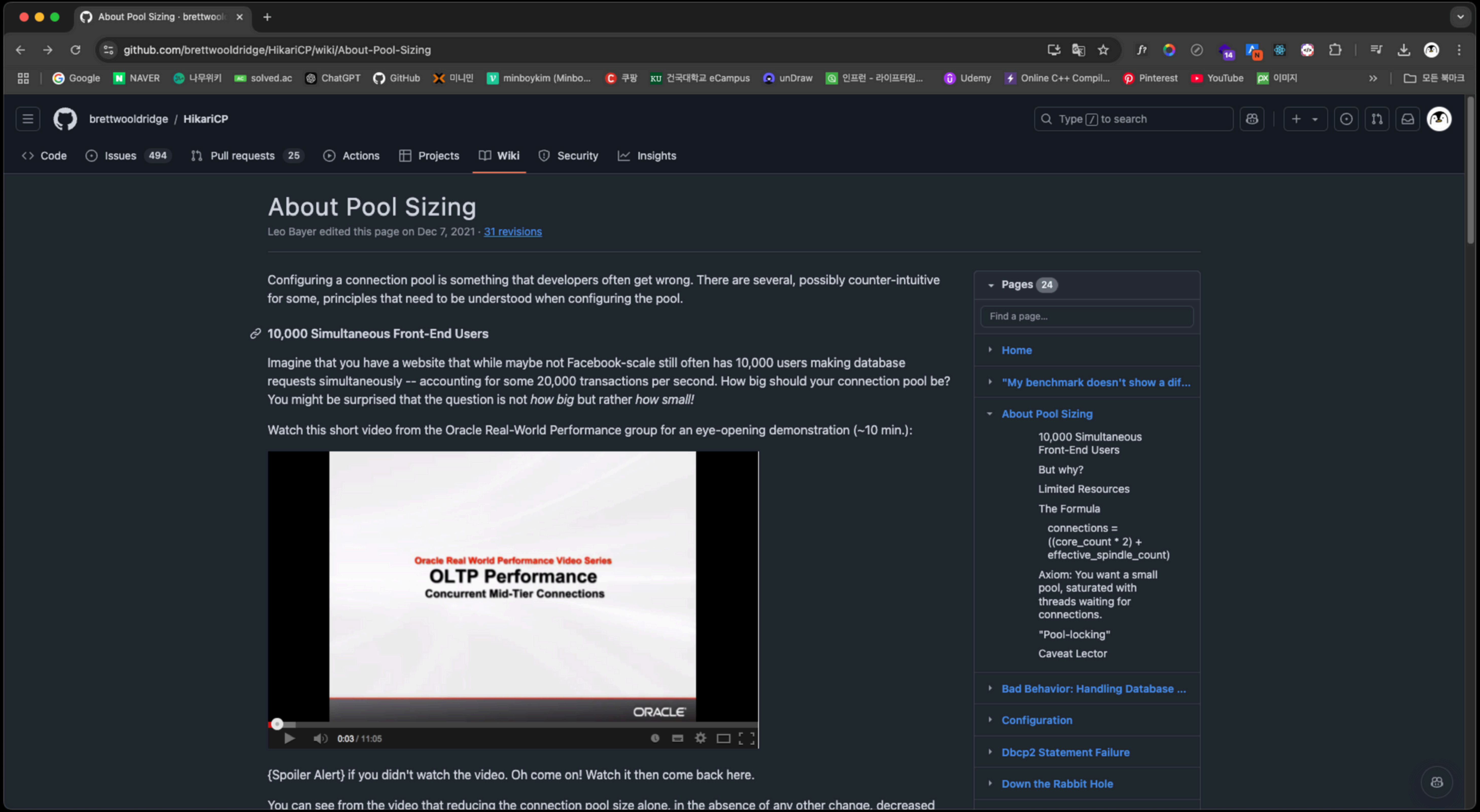
# Connection Pool



# Connection Pool



# Connection Pool



## About Pool Sizing

Leo Bayer edited this page on Dec 7, 2021 · 31 revisions

Configuring a connection pool is something that developers often get wrong. There are several, possibly counter-intuitive for some, principles that need to be understood when configuring the pool.

### 10,000 Simultaneous Front-End Users

Imagine that you have a website that while maybe not Facebook-scale still often has 10,000 users making database requests simultaneously -- accounting for some 20,000 transactions per second. How big should your connection pool be? You might be surprised that the question is not *how big* but rather *how small!*

Watch this short video from the Oracle Real-World Performance group for an eye-opening demonstration (~10 min.):

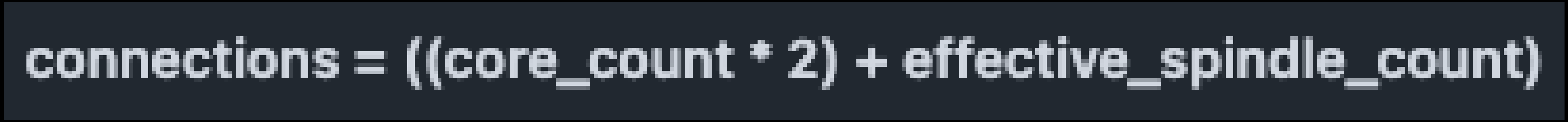
Oracle Real World Performance Video Series
OLTP Performance
Concurrent Mid-Tier Connections
ORACLE
0:03 / 11:05

{Spoiler Alert} if you didn't watch the video. Oh come on! Watch it then come back here.

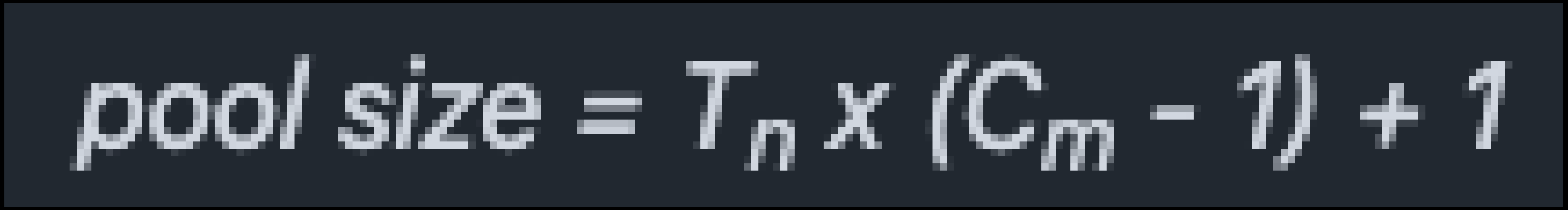
You can see from the video that reducing the connection pool size alone, in the absence of any other change, decreased

Pages 24
Find a page...
Home
"My benchmark doesn't show a dif...
About Pool Sizing
10,000 Simultaneous Front-End Users
But why?
Limited Resources
The Formula
connections = ((core_count * 2) + effective_spindle_count)
Axiom: You want a small pool, saturated with threads waiting for connections.
"Pool-locking"
Caveat Lector
Bad Behavior: Handling Database ...
Configuration
Dbcp2 Statement Failure
Down the Rabbit Hole

# Connection Pool



```
connections = ((core_count * 2) + effective_spindle_count)
```



$$pool\ size = T_n \times (C_m - 1) + 1$$

# QnA & Discussion

# THANK YOU