

Web Studio 2019

5. RDBMS

Contents

1. RDBMS

2. SQLAlchemy

- 실습: 지금까지의 API서버를 json에서 Sqlite3으로 바꾸기

RDBMS

RDBMS?

1. Relational DataBase Management System
2. 70년에 에드거 F. 커드가 도입한 관계형 모델을 기반으로 하는 데이터베이스 관리 시스템
3. Oracle, MSSQL, MySQL, PostgreSQL, SQLite3 등이 있음
4. 데이터베이스 > 테이블 > Column, row

| id | email | password |
|----|--------------------------|-------------|
| 1 | <u>sisobus@vuno.co</u> | Dksdkffiwba |
| 2 | <u>sisobus@naver.com</u> | Djffywba |

RDBMS

Why?

1. 데이터의 삽입, 수정, 삭제, 검색등의 성능이 빠름
(일반적으로 B-tree 계열을 이용)
2. 수십년 동안 입증된 안정성 (은행, kakaobank*, facebook, 도서관, 정부기관, etc ...)
3. 쓸만한 무료 오픈소스가 많음 (PostgreSQL, MariaDB, SQLite3)

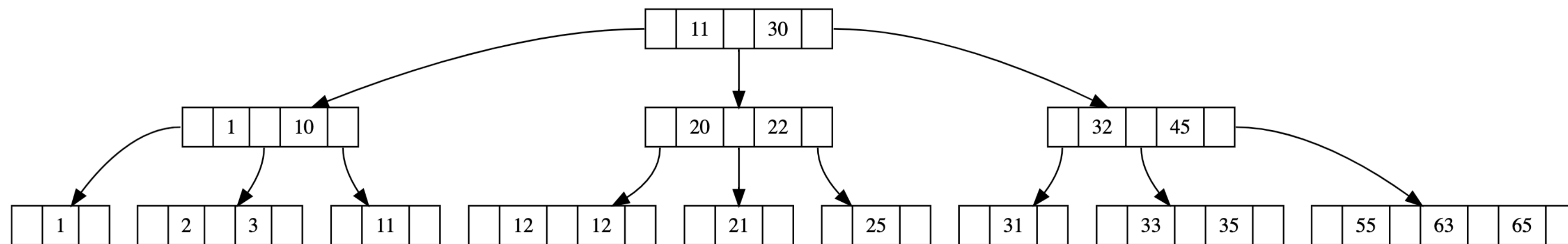


Figure 1.B-tree 예시

RDBMS

단점

1. 안되는건 안됨
(e.g. 70% 일치하는 문자열을 찾아줘!)
2. 설치하는 거 귀찮음
3. 무거운 RDBMS는 background에서 resource를 잡아먹음
4. 데이터를 관리하기 위해서는 SQL을 알아야함

RDBMS

SQL?

1. Structured Query Language
2. RDBMS의 데이터를 관리하기 위한 목적의 프로그래밍 언어
3. Query? 질의?
(e.g. 이름이 김상근인 유저를 찾아라
= select * from user where name="김상근";)
4. 몰라도 됨, 우린 ORM이란 걸 쓸거임 (물론 알면 좋습니다 ;))

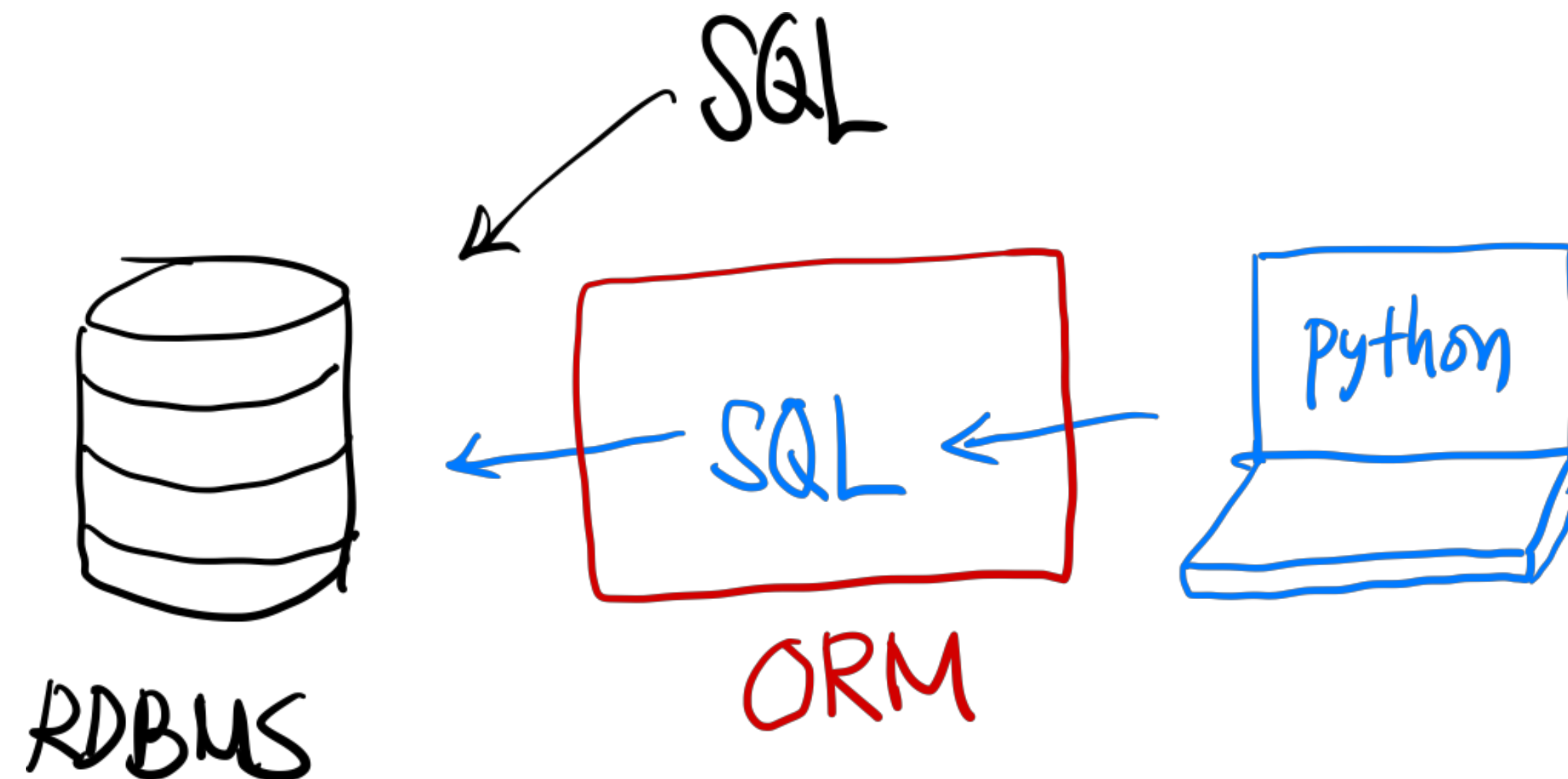
RDBMS

ORM?

1. Object Relational Mapping

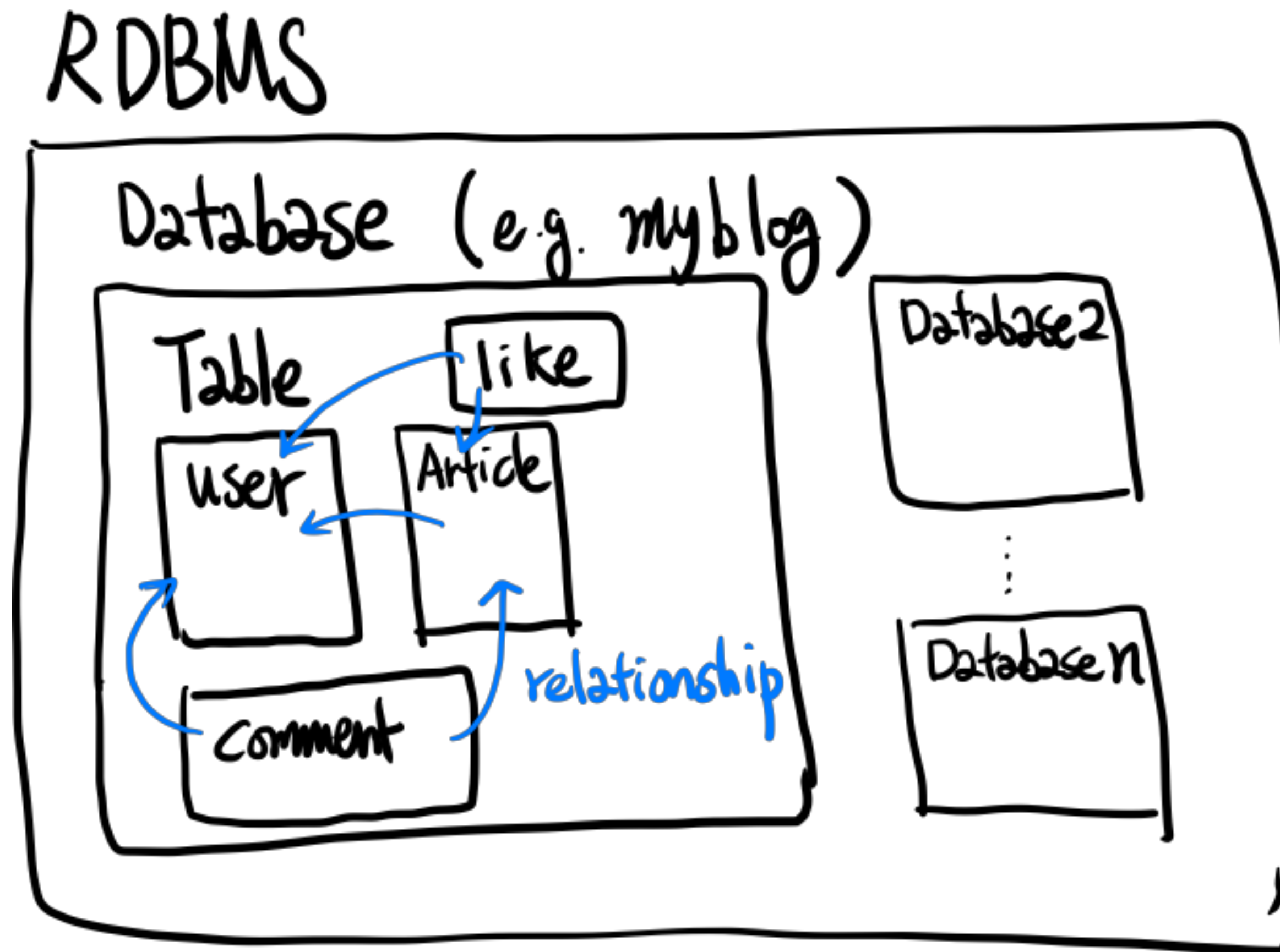
2. 데이터베이스와 프로그래밍 언어 간의 호환되지 않는 데이터를 변환하는 프로그래밍 기법

3. 우리는 python으로 코드를 작성하면 라이브러리가 SQL로 바뀌서 호출해 줌



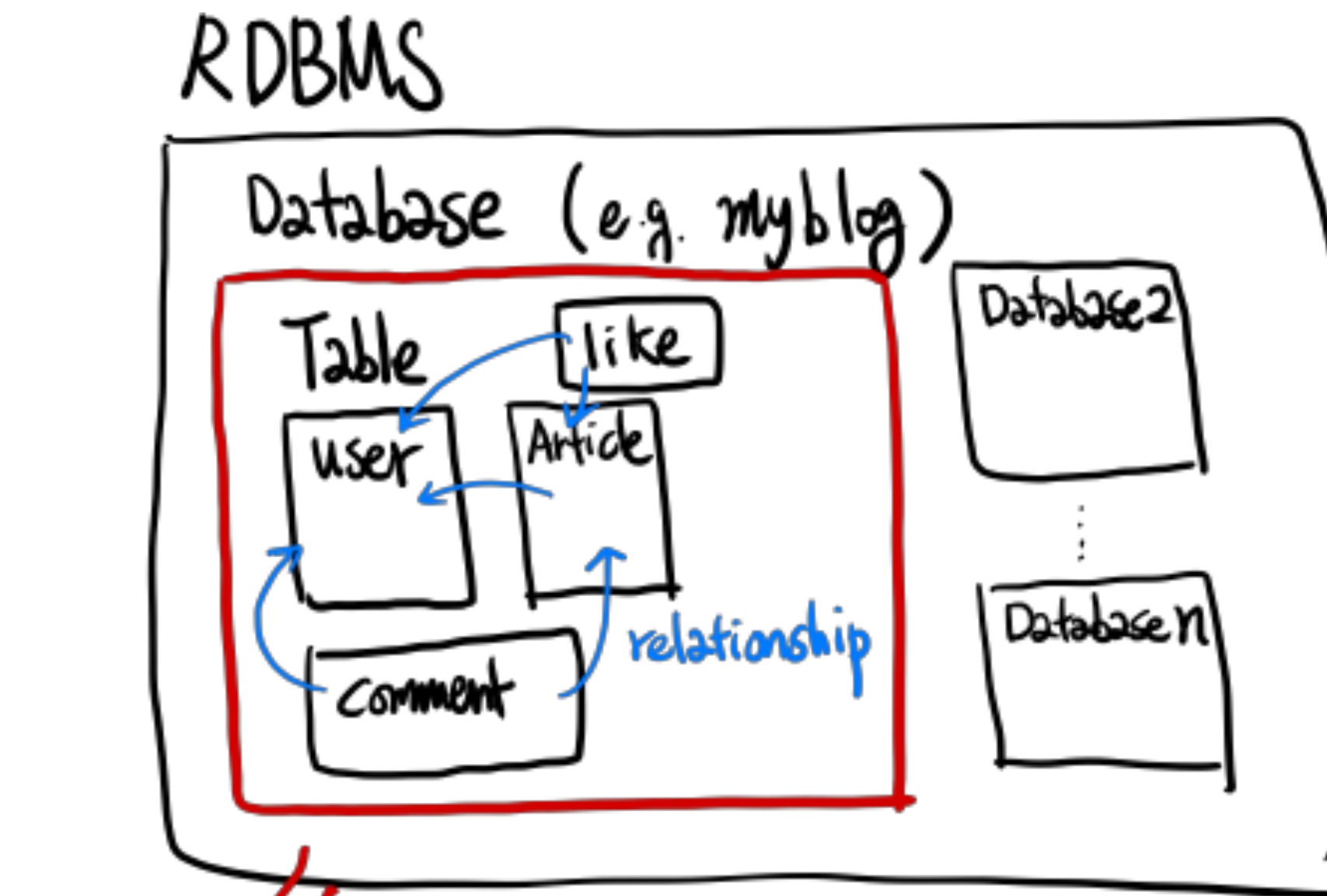
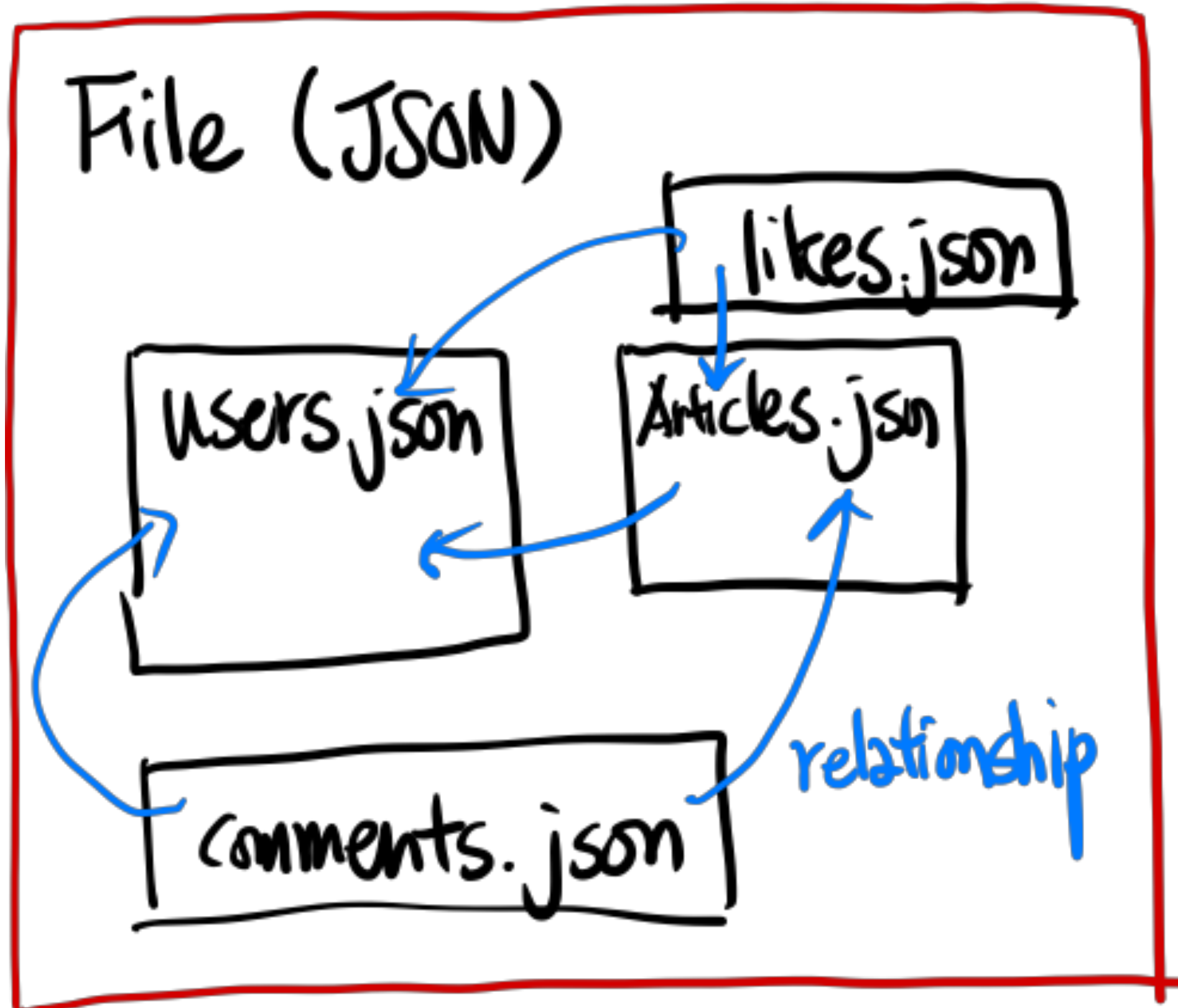
RDBMS

우리 API를 RDBMS로 적용해보면



RDBMS

우리 API를 RDBMS로 적용해보면



기존 RDBMS
file 한개 = table 하나

SQLAlchemy

Python ORM 라이브러리

1. flask에서 쓰기 편하게 Flask-sqlalchemy가 있어서 이를 설치하면 됨

\$ pip install flask-sqlalchemy

2. 오른쪽과 같이 table을 정의할 수 있음

3. 테이블에 어떤 column을 가질지 정의

4. 각각의 column에 어떤 데이터를 넣을지
타입도 정의

```
# models.py
from flask_sqlalchemy import SQLAlchemy
from werkzeug import generate_password_hash, check_password_hash
from sqlalchemy.orm import relationship, backref
```

```
db = SQLAlchemy()
```

```
class User(db.Model):
    __tablename__ = 'user'
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(300))
    password = db.Column(db.String(300))

    def __init__(self, email, password):
        self.email = email
        self.set_password(password)

    def set_password(self, password):
        self.password = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password, password)
```

SQLAlchemy

Python ORM 라이브러리

1. from models import db, User, Article로 models.py에서 정의한 class를 import해줌
2. basedir: 이 파일이 실행된 절대 주소
3. SQLALCHEMY_DATABASE_URI: database 파일 위치
4. db.init_app(app)으로 초기화
5. db.create_all() 로 테이블 생성

```
# boilerplate.py
from flask import Flask, request
from flask_restful import Api, Resource
import json
import os
from models import db, User, Article

basedir = os.path.dirname(os.path.abspath(__file__))
SQLALCHEMY_DATABASE_URI = 'sqlite:/// ' + os.path.join(basedir, 'app.db')
app = Flask(__name__)
app.config.update({
    'SQLALCHEMY_TRACK_MODIFICATIONS': True,
    "SQLALCHEMY_DATABASE_URI": SQLALCHEMY_DATABASE_URI,
})
api = Api(app)
db.init_app(app)

if __name__ == '__main__':
    with app.app_context():
        db.create_all()
    app.run(host='0.0.0.0', port=5000, debug=True)
```

RDBMS

실습: 지금까지의 API서버를 json에서 Sqlite3으로 바꾸기

1. 브랜치 새로 파고
2. 5_rdbms/practice/〈여러분영어이름〉 디렉토리 만들고
3. 5_rdbms/practice/Sangkeun에서 boilerplate.py, tests.sh 카피하고
4. ArticleList, CommentList, LikeList 수정하기

Q & A

Appendix

Bash 기본 명령어

1. ls : 현재 위치의 파일 리스트
 - ls -al : 숨겨진 파일까지 전부 보기
2. cd : 디렉토리 이동
 - .은 현재 디렉토리, ..은 바로 전 디렉토리를 의미함
 - 예를들어 cd .. 이 명령어는 바로 전 디렉토리로 이동
3. rm : 파일 지우기
 - rm -rf : 디렉토리도 지울 수 있음
4. vi <filename> : vim 에디터를 이용한 파일 작성하기
5. mkdir <directoryname> : 디렉토리(폴더) 만들기
6. pwd : 현재 위치 출력하기

Appendix

Vim

1. 최고의 Text editor
2. 이것만 잘써도 코딩 생산성이 상당히 많이 올라감
3. 대부분의 IDE에는 Vim Plugin이 존재함
4. Vim 쓰세요
5. 두번쓰세요
6. 세번쓰세요
7. 평생쓰세요

Appendix

Vim

1. 명령어 모드와 에디터 모드로 나뉨
2. 맨처음 들어갈 때에는 명령어 모드임
3. 에디터 모드에서 명령어 모드로 바꾸는 것은 esc로 함
4. 명령어 모드에서 에디터 모드로 바꾸는 것은 i, o, a로 함 (i만 알아도 상관 없습니다.)

Appendix

Vim 명령어 모드

1. i는 현재 커서에서 에디터모드로 전환
2. yy는 현재 줄 복사
3. p는 복사한 것을 붙여넣기
4. dd는 현재줄 삭제
5. u는 undo
6. ctrl+r는 redo
7. :wq 는 저장후 종료를 의미함
8. /<search string> 으로 검색할 수 있음
9. :%s/<stringA>/<stringB>/g 는 stringA를 stringB로 대체함
 - :%s/^/₩/₩//g : 맨 앞에 //를 붙임
 - :3,12s/ha/hi/g : 3번줄부터 12번째줄의 ha를 hi로 바꿈