

Hello,

KDT 웹 개발자 양성 프로젝트

5기!

with







DataBase



데이터 들의 집합



그래서 그거

왜 쓰나요?



```
public class Scanner7 { new *  
    public static void main(String[] args) { new *  
        Scanner scanner = new Scanner(System.in);  
  
        String id = null;  
        String pw = null;  
        String name = null;  
        String addr = null;  
  
        System.out.println("id = " + id);  
        System.out.println("pw = " + pw);  
        System.out.println("name = " + name);  
        System.out.println("addr = " + addr);
```



```
System.out.println("=== 회원 가입을 시작 합니다 ===");  
System.out.print("ID를 입력하세요 : ");  
id = scanner.nextLine();  
System.out.print("비밀번호를 입력하세요 : ");  
pw = scanner.nextLine();  
System.out.print("이름을 입력하세요 : ");  
name = scanner.nextLine();  
System.out.print("주소를 입력하세요 : ");  
addr = scanner.nextLine();  
System.out.println("회원 가입을 축하합니다!");  
  
System.out.println("id = " + id);  
System.out.println("pw = " + pw);  
System.out.println("name = " + name);  
System.out.println("addr = " + addr);
```

```
id = null
pw = null
name = null
addr = null
=== 회원 가입을 시작 합니다 ===
ID를 입력하세요 : xenosign
비밀번호를 입력하세요 : 1234
이름을 입력하세요 : 이효석
주소를 입력하세요 : 서울 서대문구 북아현동
회원 가입을 축하합니다!
id = xenosign
pw = 1234
name = 이효석
addr = 서울 서대문구 북아현동
```

하지만 프로그램을
재실행 하면!?




```
id = null
```

```
pw = null
```

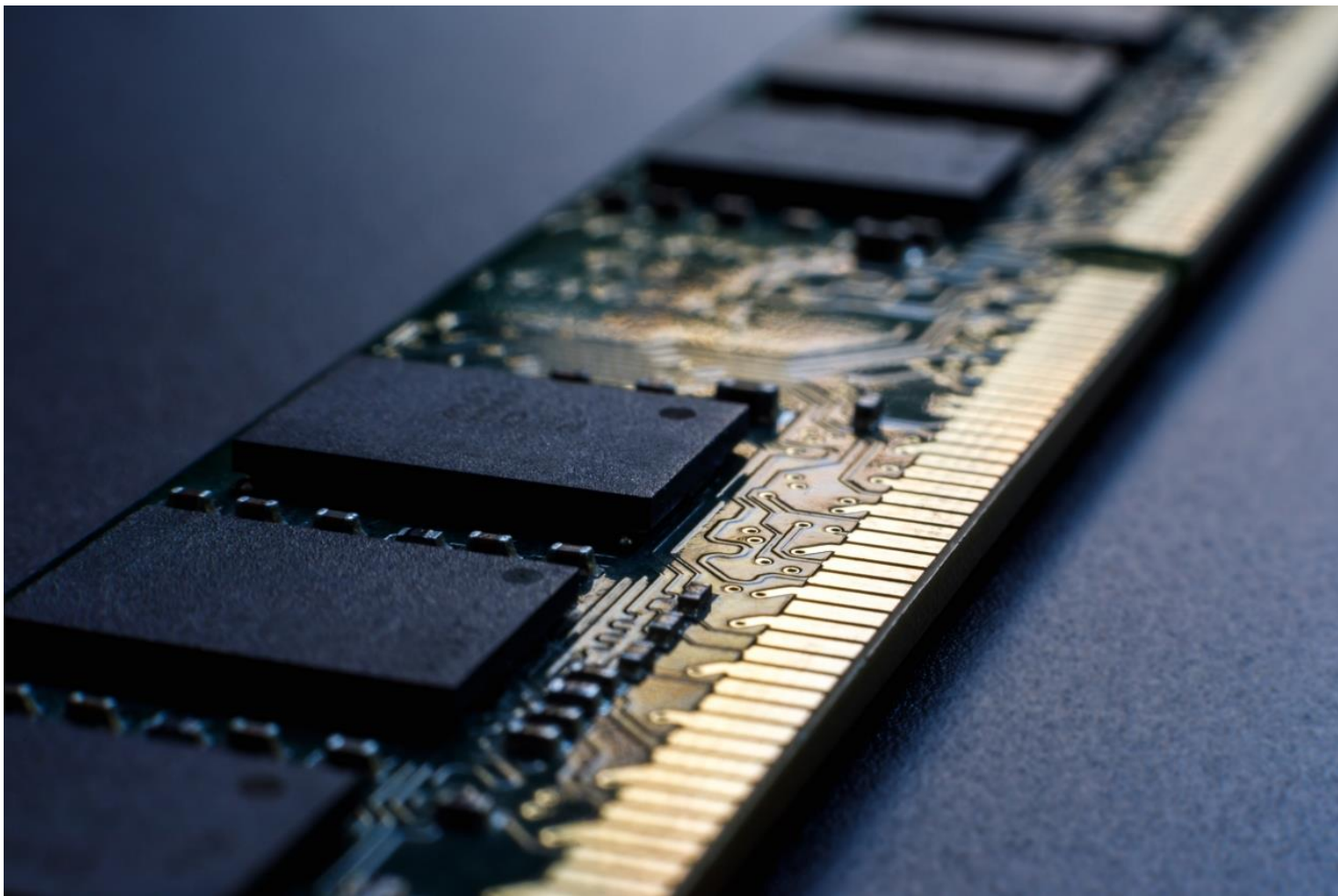
```
name = null
```

```
addr = null
```

```
=== 회원 가입을 시작 합니다 ===
```

```
ID를 입력하세요 :
```









하지만

“BUT”







symbian
OS



AMIGA OS



OS/2 **WARP**

 **BlackBerry**



Mac OS


solaris™



TIZEN®

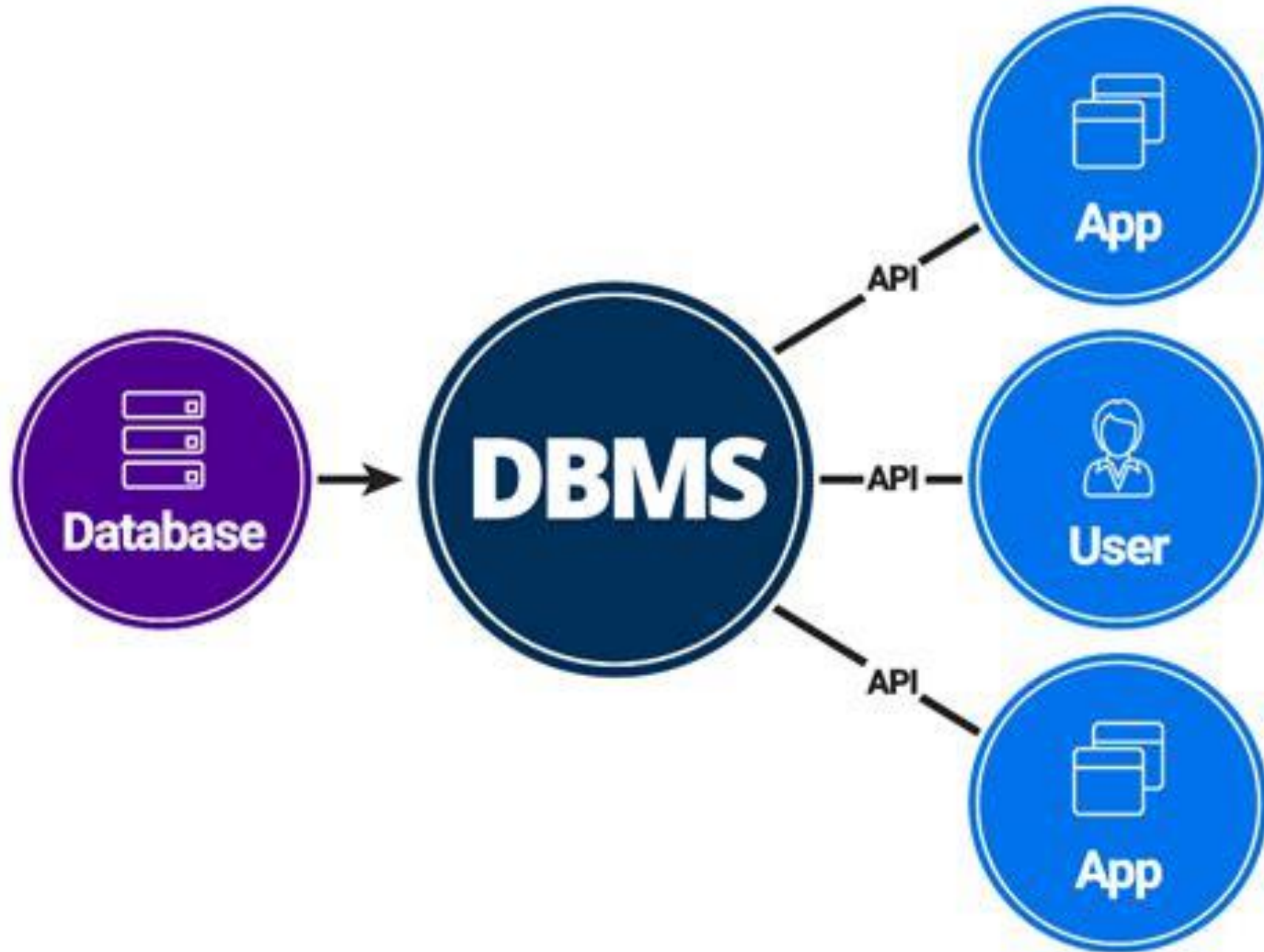
UNIX®



일단 노력은 해 본다









DBMS

(DataBase Management System)

DBMS



- 데이터 베이스를 관리하고 운영하는 SW
- 다양한 형태, 서비스가 존재 합니다!





SQL

(Structured Query Language)

SQL



- 구조가 있는 질문용 언어라는 뜻
- SELECT, INSERT, UPDATE, DELETE 같은 언어를 통해 데이터 베이스의 데이터를 다루는 언어
- 다수의 DBMS가 SQL 방식을 따르기 때문에 SQL 을 배우면, 많은 DBMS를 비교적 빠르게 습득이 가능
- MySQL, SQLite, ORACLE 등이 SQL 구문을 사용



DB의 종류



관계형(SQL)

VS

비관계형(NoSQL)



관계형 DB



관계형, Relational DBMS(RDBMS)

- RDBMS 는 SQL 을 사용하는 DB 입니다
- 키와 값의 관계를 테이블화 시킨 원칙을 토대로 DB 를 구성합니다

열 이름

↓

아이디	이름	연락처
flower	화사	010-1111-1111
finetree	솔라	010-2222-2222
moon	문별	010-3333-3333
whee	휘인	010-4444-4444

행(row) →

↑

열(column)


관계형, Relational DBMS(RDBMS)



- 즉, RDBMS 는 테이블로 구성이 됩니다.
- 먼저 테이블이 구성되고 테이블의 구조에 맞추어 데이터가 들어가기 때문에 DB를 구성하기 전에 스키마라 불리는 DB의 구조, 관계, 제약 사항에 대한 정의가 필요합니다



너무 제약이 많아!



하지만 빨랐죠

관계형, Relational DBMS(RDBMS)



- 장점

- 구조화가 명확하게 되어 있어서 예외가 없음
- 데이터 입, 출력 속도가 매우 빠릅니다
- 신뢰성이 매우 높음

- 단점

- DB의 구조 변경이 매우 어려움 → 빅데이터 등에는 사용이 어려움(새로운 키가 추가 되면 전체 스키마 변경이 필요)



비관계형 DB

비관계형, Non Relational DBMS(NoSQL)



- SQL 을 사용하지 않는 모든 DB를 통칭합니다
 - ex) 한국어 vs 비한국어(= 영어, 프랑스어, 스페인어 등등)
- 대표적으로 문서형, 그래프형, 키밸류형, 와이드 컬럼형 등이 있습니다!
- 약간 컴퓨터 언어와 일맥 상통하는 부분으로 특정 목적에 맞는 DB가 존재 합니다 (ex. Html → 웹 개발 / 파이썬 → 인공지능 / Swift → 앱)



비관계형, Non Relational DBMS(NoSQL)



- 장점

- 보통 대용량 데이터 처리에 효율적
- DB의 구조 변경이 쉽고, 확장성이 뛰어남
- 복잡한 데이터 구조의 표현이 가능

- 단점

- 데이터 자체가 크면 전체 데이터를 일부 읽어서 처리해야 하므로 데이터가 크면 속도가 저하되는 문제 발생

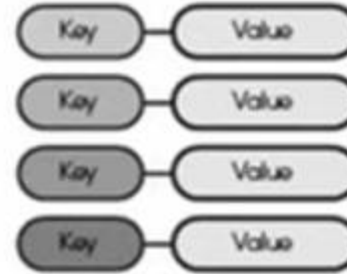
Document



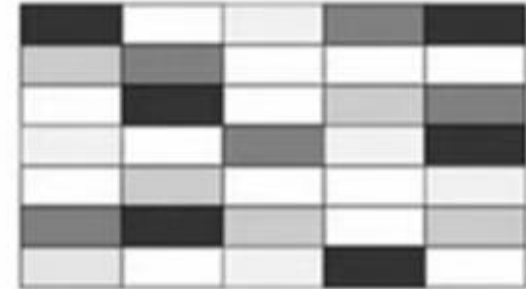
Graph



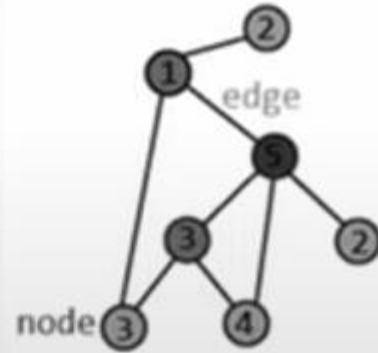
Key-Value



Wide-Column



```
{
  "user": {
    "id": "143",
    "name": "improgrammer",
    "city": "New York"
  }
}
```



1	Fruit	A Foo	B Baz	
2	City	E DC	D PLA	G FLD
3	State	A NZ	C CL	





https://youtu.be/Q_9cFgzZr8Q