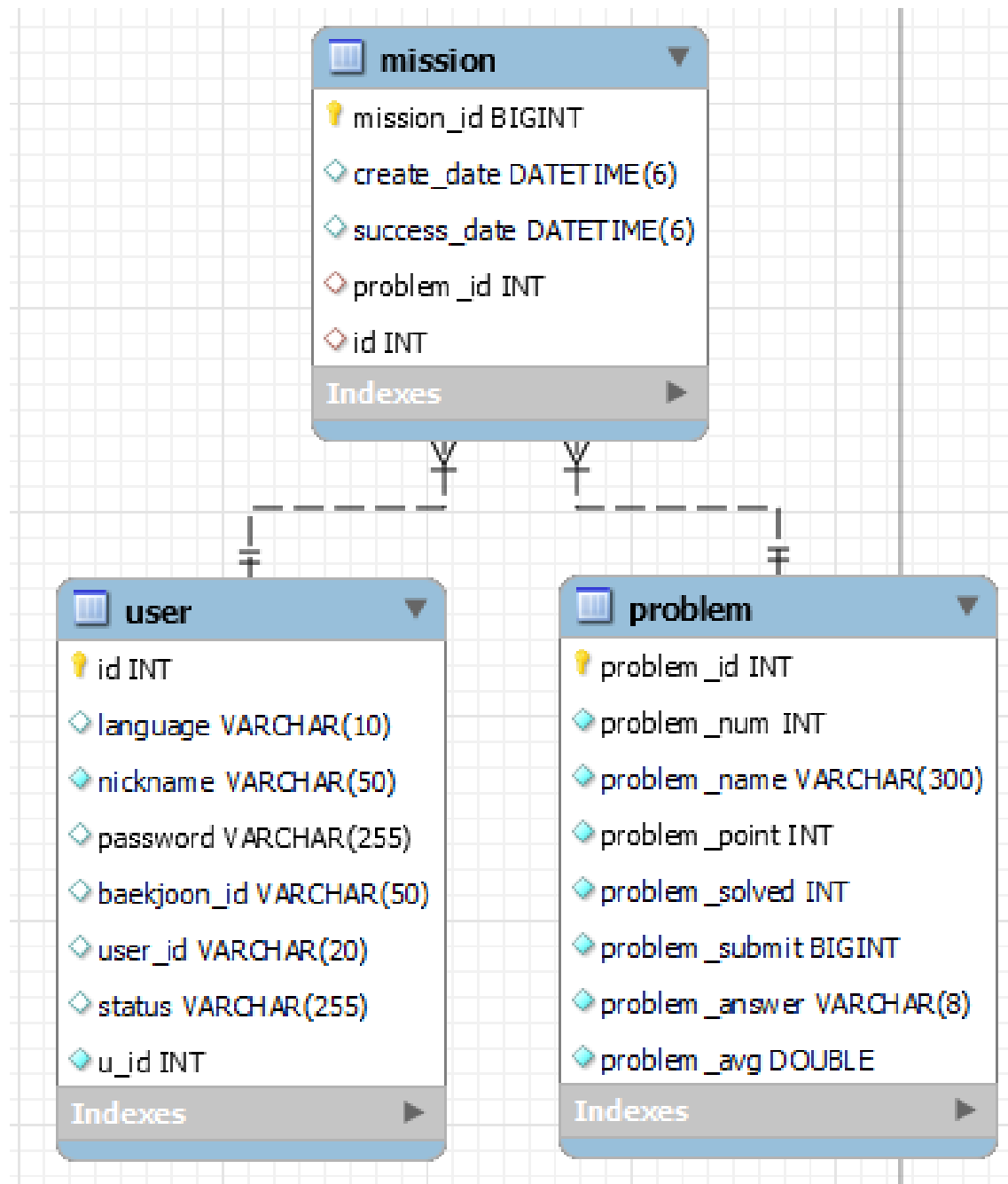


SQL과 NOSQL

장정훈

SQL(관계형 DB)



SQL(관계형 DB)

- SQL을 사용하면 RDBMS에서 데이터를 CRUD 작업을 할 수 있음.
- 관계형 DB의 2가지 특징
 - 데이터는 정해진 데이터 스키마에 따라 테이블에 저장.
 - 데이터는 관계를 통해 여러 테이블에 분산.
- 데이터는 테이블 레코드로 저장되는데, 각 테이블마다 명확하게 정의된 구조가 있음.
- 따라서 스키마를 준수하지 않은 레코드는 테이블에 추가할 수 없음.
- 데이터의 중복을 피하기 위해 '관계'를 이용.

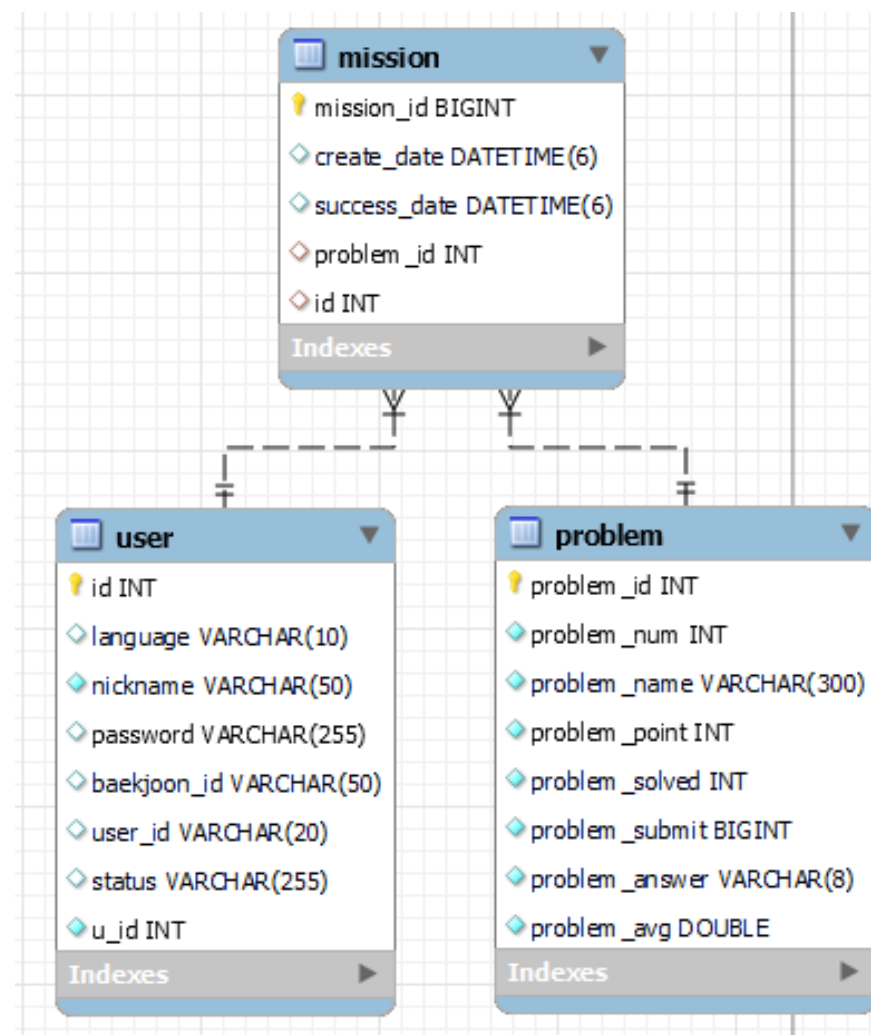
NoSQL(비관계형 DB)

- **스키마도 없고, 관계도 없다!**
- NoSQL에서는 다른 구조의 데이터를 같은 컬렉션에 추가 가능.
- 문서는 Json과 비슷한 형태로 데이터를 가지고 있음.
- 관련 데이터를 동일한 컬렉션에 넣음.
- 여러 테이블에 join할 필요없이 이미 필요한 모든 것을 갖춘 문서를 작성하는 것이 NoSQL.
 - NoSQL에는 join이라는 개념이 없음.

NoSQL

```
1 {
2   "problemDto": {
3     "problemId": 5,
4     "problemNum": 1004,
5     "problemName": "어린 왕자",
6     "problemSolved": 8244,
7     "problemSubmit": 24420,
8     "problemAnswer": "42.569%",
9     "problemAvg": 0.0
10  },
11  "createDate": [
12    2022,
13    3,
14    20,
15    14,
16    6,
17    49
18  ],
19  "successDate": [
20    2022,
21    3,
22    20,
23    14,
24    6,
25    49
26  ]
27 }
```

SQL



NoSQL에서 join하고 싶을 때?

- 컬렉션을 통해 데이터를 복제하여,
- 각 컬렉션 일부분에 속하는 데이터를 정확하게 산출.
- 하지만, 이러면 데이터가 중복되어 서로 영향을 줄 위험이 있음.
- 즉, join을 잘 사용하지 않고 자주 변경되지 않는 데이터일 때 NoSQL을 쓰면 상당히 효율적이다.

확장

- 수직적 확장 :
단순히 DB 서버의 성능을 향상시키는 것. (ex. CPU 업그레이드)
- 수평적 확장 :
더 많은 서버가 추가되고 DB가 전체적으로 분산됨을 의미.
(하나의 DB에서 작동하지만 여러 호스트에서 작동)
- 데이터 저장 방식으로 인해 SQL DB는 일반적으로 수직적 확장만 지원.
- 수평적 확장은 NoSQL DB에서만 가능.

장점

SQL	NoSQL
명확하게 정의된 스키마	스키마가 없어서 유연함
데이터 무결성 보장	언제든지 저장된 데이터를 조정하고 새로운 필드 추가 가능
관계는 각 데이터를 중복없이 한번만 저장	데이터는 필요로 하는 형식으로 저장
	데이터 읽어오는 속도 빨라짐
	수직 및 수평 확장이 가능
	애플리케이션이 발생시키는 모든 읽기/쓰기 요청 처리 가능

단점

SQL	NoSQL
덜 유연함	유연성으로 인해 데이터 구조 결정을 미루게 될 수 있음
데이터 스키마를 사전에 계획하고 알려야 함	데이터 중복을 계속 업데이트 해야 함
복잡한 쿼리가 만들어질 수 있음	수정 시 모든 컬렉션에서 수행해야 함
대체로 수직적 확장만 가능함	

그럼 둘 중에 뭘 선택?

- **SQL DB 사용이 더 좋을 때**
- 관계를 맺고 있는 데이터가 자주 변경되는 애플리케이션의 경우
 - NoSQL에서는 여러 컬렉션을 모두 수정해야 하기 때문에 비효율적
- 변경될 여지가 없고, 명확한 스키마가 사용자와 데이터에게 중요한 경우

그럼 둘 중에 뭘 선택?

- **NoSQL DB 사용이 더 좋을 때**
- 정확한 데이터 구조를 알 수 없거나 변경/확장 될 수 있는 경우
- 읽기를 자주 하지만, 데이터 변경은 자주 없는 경우
- DB를 수평으로 확장해야 하는 경우
(막대한 양의 데이터를 다뤄야 하는 경우)