

● RDB - NoSQL를 비교 설명해라.

▶ 관계형 데이터베이스(RDB, Relational DataBase)는 사전에 엄격하게 정의된 DB schema를 요구하는 table 기반 데이터 구조를 갖습니다. RDB는 엄격한 schema로 인해서 데이터 중복이 없기 때문에 데이터 update가 많을 때 유리합니다. 즉, RDB는 데이터 구조가 명확하여 변경될 여지가 없는 경우, 또 데이터 중복이 없으므로 데이터 update가 잦은 시스템에서 사용하면 좋습니다. 반면, NoSQL(비관계형 데이터베이스)은 table 형식이 아닌 비정형 데이터를 저장할 수 있도록 지원합니다. NoSQL의 경우 데이터 중복으로 인해 데이터 update 시 모든 컬렉션에서 수정이 필요하기 때문에 update가 적고 조회가 많을 때 유리합니다. 즉, 정확한 데이터 구조가 정해지지 않은 경우, 데이터 update가 자주 이루어지지 않고 조회가 많은 경우, 또 scale out(수평적 확장)이 가능하므로 데이터 양이 매우 많은 경우에 NoSQL을 사용하면 좋습니다.

● RDB vs NoSQL

	RDB (SQL)	NoSQL
데이터 저장 모델	table	json document / key-value / 그래프 등으로 이루어져 있다
개발 목적	데이터 중복 감소	애자일 / 확장가능성 / 수정가능성
예시	Oracle, MySQL, PostgreSQL 등	MongoDB, DynamoDB 등
Schema	엄격한 데이터 구조를 요구한다	유연한 데이터 구조를 갖는다
★장점★	<ul style="list-style-type: none"> - 명확한 데이터구조 보장 - 데이터 중복 없이 한 번만 저장 (무결성 보장) - 데이터 중복이 없어서 데이터 update 용이 	<ul style="list-style-type: none"> - 유연하고 자유로운 데이터 구조 - 새로운 필드 추가 자유로움 - 수평적 확장(scale out) 용이 (더 많은 서버가 추가되어 데이터베이스를 분산할 수 있다)
★단점★	<ul style="list-style-type: none"> - 시스템이 커지면 Join문이 많은 복잡한 query가 필요하다 - 수평적 확장이 까다로워 비용이 큰 <u>소직적 확장(Scale up)</u>이 주로 사용됨. (수직적 확장이란 데이터베이스 서버의 성능 향상 (ex CPU 성능 업그레이드, 메모리 증량)이다) - 데이터 구조가 유연하지 못함 	<ul style="list-style-type: none"> - 데이터 중복 발생 가능 - 중복 데이터가 많기 때문에 데이터 변경 시 모든 컬렉션에서 수정이 필요함 - 명확한 데이터구조 보장 X
★사용★	<ul style="list-style-type: none"> - 데이터 구조가 변경될 여지가 없이 명확한 경우 - 데이터 update가 잦은 시스템 (중복 데이터가 없으므로 변경에 유리) 	<ul style="list-style-type: none"> - 정확한 데이터 구조가 정해지지 않은 경우 - Update가 자주 이루어지지 않는 경우 (조회가 많은 경우) - 데이터 양이 매우 많은 경우 (scale out 가능)

● 수직적 확장Scale-up vs 수평적 확장 Scale-out

▶ RDB와 비교하여 NoSQL의 특징은 ACID(원자성, 일관성, 고립성, 지속성), Transaction을 지원하지 않는다는 것입니다. RDB는 ACID와 Transaction을 보장하기 위해 수평적 확장이 쉽지가 않습니다. 또한 RDB 같은 경우에는 multiple server로 수평적 확장을 하게 되면 join을 하기 위해 굉장히 복잡한 과정이 필요합니다. 때문에 RDB도 수평적 확장이 가능하지만 NoSQL에 비해 훨씬 복잡합니다. RDB를 수평적 확장하려면 샤딩(sharding)(데이터가 수

평적으로 분할되고 기기의 모음 전반에 걸쳐 분산되는 경우)가 필요합니다. ACID 준수를 유지하면서 RDB를 샅당하는 것은 매우 까다로운 작업입니다.

● Key-value storage system (NoSQL의 데이터 저장 모델)

▶ NoSQL, Not only SQL은, key-value storage system을 일컫는 말입니다. 기존의 관계형 database의 경우에는 단일 기업의 데이터를 다루는데 최적화 되어 있습니다. 하지만 최신 데이터들은 꼭 관계형으로 처리될 필요가 없는 경우도 많고, 다뤄야 하는 데이터의 양도 훨씬 커졌습니다. 즉, Big data라고 일컬어 지는 많은 양의 데이터를 처리하기 위한 방법으로 다양한 해결책이 나왔는데, 그 중 하나의 방법이 Key-value storage system 입니다. Key-value stores는 NoSQL system이라고도 불리는데, 그 이유는 SQL을 보통 지원하지 않고 transaction을 지원하지 않는 등 SQL을 사용하는 기존의 RDB와의 차이점 때문입니다. NoSQL의 종류는 MongoDB, Bigtable, DynamoDB, Cassandra등이 있습니다.

● MongoDB (NoSQL의 종류)

▶ NoSQL의 종류는 Bigtable, DynamoDB, Cassandra, MongoDB등이 있습니다. 예를 들면 MongoDB의 경우, 테이블을 생성한 후에 key-value 쌍으로 된, json-format으로 데이터를 작성을 해서 테이블에 insert를 하면 됩니다. 그런 다음, find() 함수로는 json format으로 모든 row를 가져올 수 있고, findOne() 함수로는 해당 key에 일치하는 row 하나만 가져올 수 있고, remove()함수로는 key에 해당하는 모든 row를 삭제할 수 있고, drop() 함수로는 전체 컬렉션을 삭제할 수 있습니다.

```
db.createCollection("student")

db.student.insert({"id": 2022394, "name": "Nossi", "class": ["Math", "Eng"]})
db.student.insert({"id": 2021921, "name": "Bob", "class": ["Eng"]})

db.student.find() // Fetch all students in JSON format
db.student.findOne({"id": 2022394}) // Find one matching student

db.student.remove({"name": "Nossi"}) // Delete matching students
db.student.drop() // Drops the entire collection
```

→ SQL : Structured Query Language의 약자로 데이터베이스에서 데이터를 조회하고 입력하고 수정하고 삭제하고 저장하는데 사용되는 질의 언어이다.

→ 쿼리(query): 데이터베이스에 정보를 요청하는 것이다. 데이터베이스나 파일의 내용 중 원하는 내용을 검색하기 위하여 몇 개의 코드(code)나 키(Key)를 기초로 질의하는 것을 말한다. 대개 데이터베이스로부터 특정한 주제어나 어귀를 찾기 위해 사용된다. 주제가가 검색엔진의 검색필드 내에 입력된 다음, 그 내용이 웹 서버로 넘겨진다.

데이터베이스 스키마(Database Schema): 데이터베이스에서 자료의 구조, 자료의 표현 방법, 자료 간의 관계를 형식 언어로 정의한 구조.

→ transaction: 데이터베이스 내에서 수행되는 작업의 최소 단위로, 데이터베이스의 무결성(데이터베이스에 저장된 데이터의 일관성과 정확성을 지키는 것)을 유지하며 DB의 상태를 변화시키는 기능을 수행한다. transaction은 하나 이상의 query를 포함해야 하고, ACID라고 칭해지는 원자성, 일관성, 고립성, 지속성의 4가지 규칙을 만족해야 한다.

Resources: inflearn 개발남 노씨