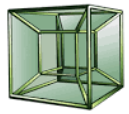
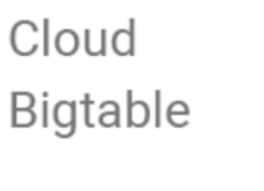
Cloud
Bigtable

Apache
CASSANDRA

SQL Database

NOSQL Database





SQL vs NOSQL

목차

SQL이란?

SQL(Structured Query Language) :
구조화된 쿼리 언어 , 관계형 데이터베이스

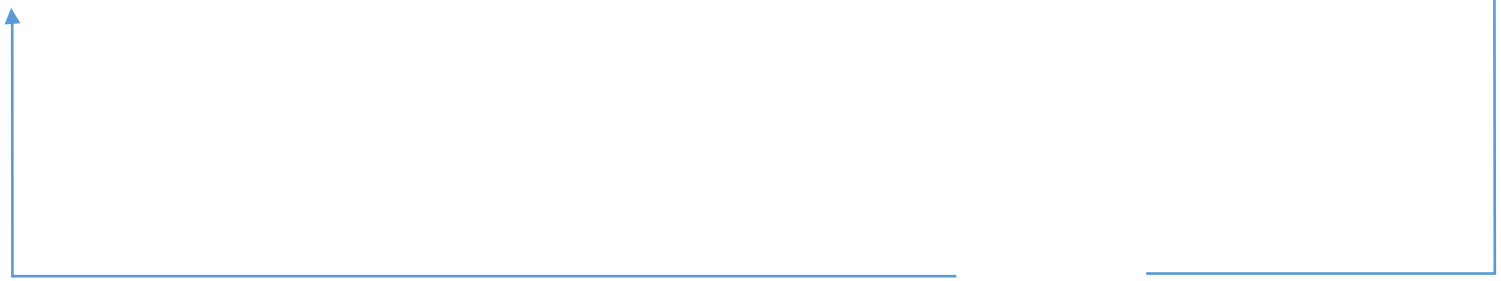
특징:
1. 데이터는 정해진 데이터 스키마에 따라 테이블에 저장됨
2. 데이터는 **관계**를 통해 여러 테이블에 분산됨.
3. 즉 스키마를 따르지 않는 레코드는 테이블에 추가할 수 없음.

Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Ya dong ill



id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5

Relation Table

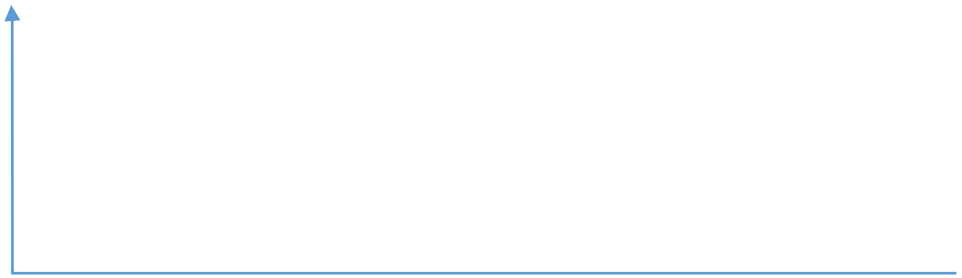
만약 김현욱이 수강신청에서 임베디드시스템을 추가하기 위해서 실행해야하는 쿼리는?

Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Ya dong ill



id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5

Relation Table

만약 김현욱이 수강신청에서 임베디드시스템을 추가하기 위해서 실행해야하는 쿼리는?

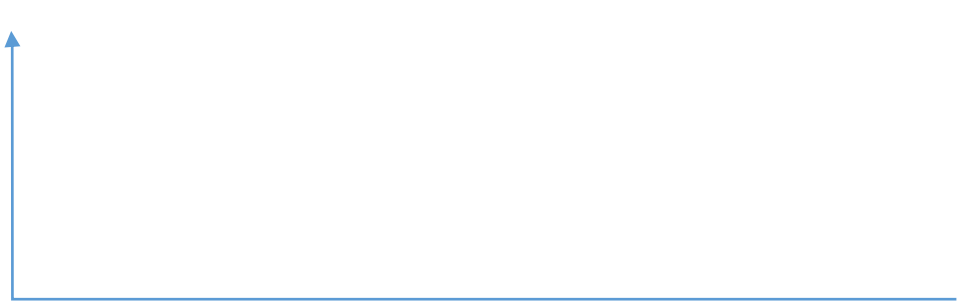
INSERT INTO Relation(id,Student_id,Class_id) VALUES(8,2,5);

Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Ya dong ill



id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5
8	2	5

Relation Table

만약 김현욱이 수강신청에서 임베디드시스템을 추가하기 위해서 실행해야하는 쿼리는?

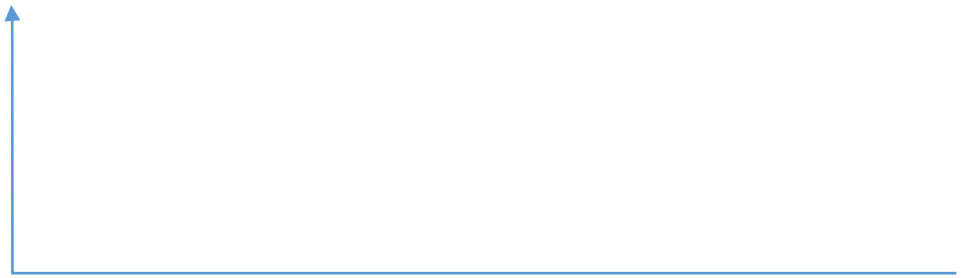
INSERT INTO Relation(id,Student_id,Class_id) VALUES(8,2,5);

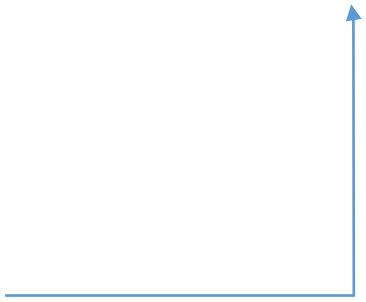
Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Ya dong ill





id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5
8	2	5

Relation Table

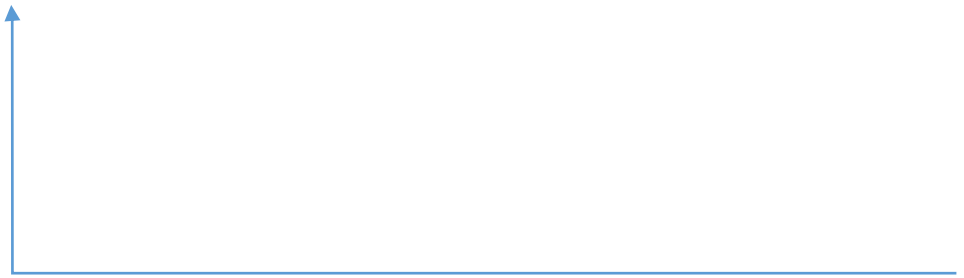
강구민이 수강신청한 수업들의 정보를 알고싶다면?

Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Ya dong ill



id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5
8	2	5

Relation Table

강구민이 수강신청한 수업들의 정보를 알고싶다면?

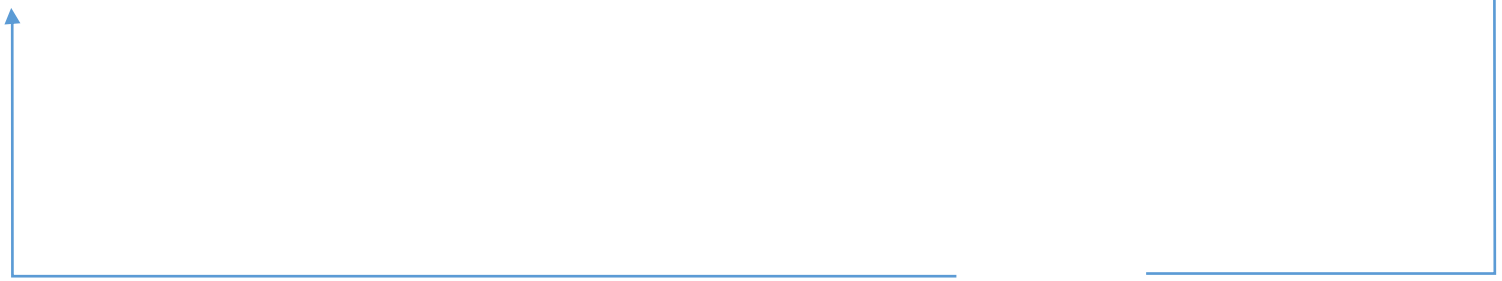
```
SELECT Class_id,Class_name,Grade_Professor Relation r LEFT JOIN Class c
ON r.Class_id = c.Class_id
WHERE r.Student_id = 1;
```

Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Ya dong ill



id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5
8	2	5

Relation Table

강구민이 수강신청한 수업들의 정보를 알고싶다면?

```
SELECT Class_id,Class_name,Grade_Professor Relation r LEFT JOIN Class c
ON r.Class_id = c.Class_id
WHERE r.Student_id = 1;
```

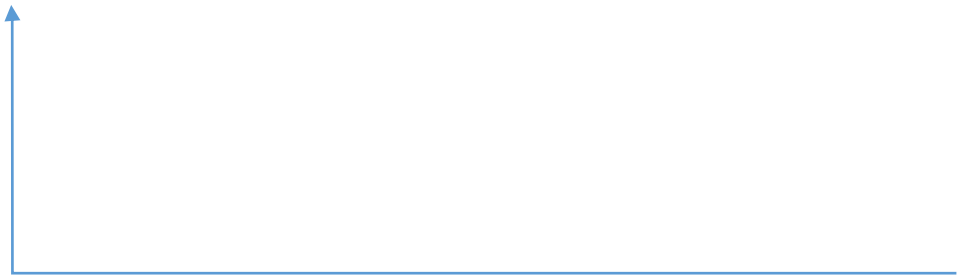
Class_id	Class_name	Grade	Professor
2	Android Studio	3	Kim
6	DataBase	3	Ya dong ill

Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Ya dong ill



id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5
8	2	5

Relation Table

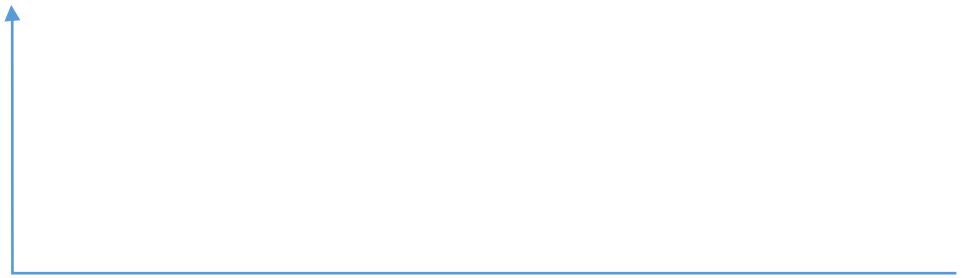
데이터베이스 교수가 Ya dong ill에서 Yong Hack으로 바뀌어야 한다면?

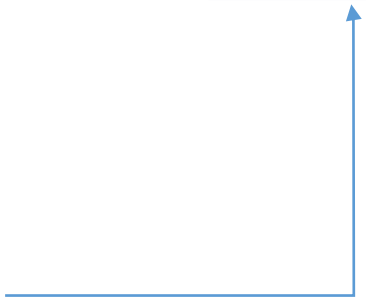
Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Ya dong ill





id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5
8	2	5

Relation Table

데이터베이스 교수가 Ya dong ill에서 Yong Hack으로 바꿔야 한다면?

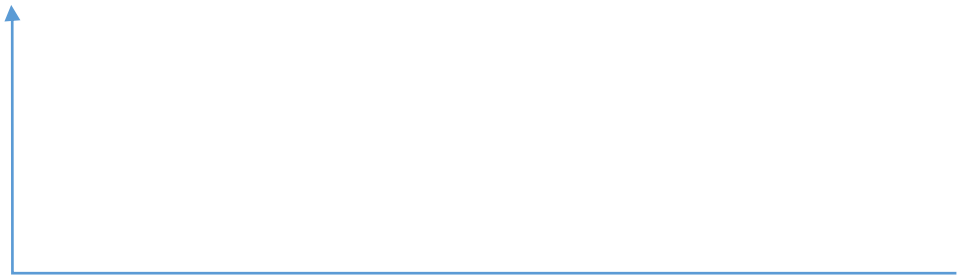
UPDATE Class SET Professor = "Yong Hack" WHERE Class_id = 6;

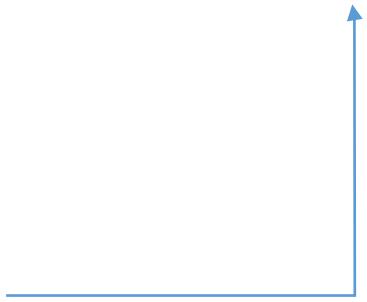
Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Yong Hack





id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5
8	2	5

Relation Table

데이터베이스 교수가 Ya dong ill에서 Yong Hack으로 바꿔야 한다면?

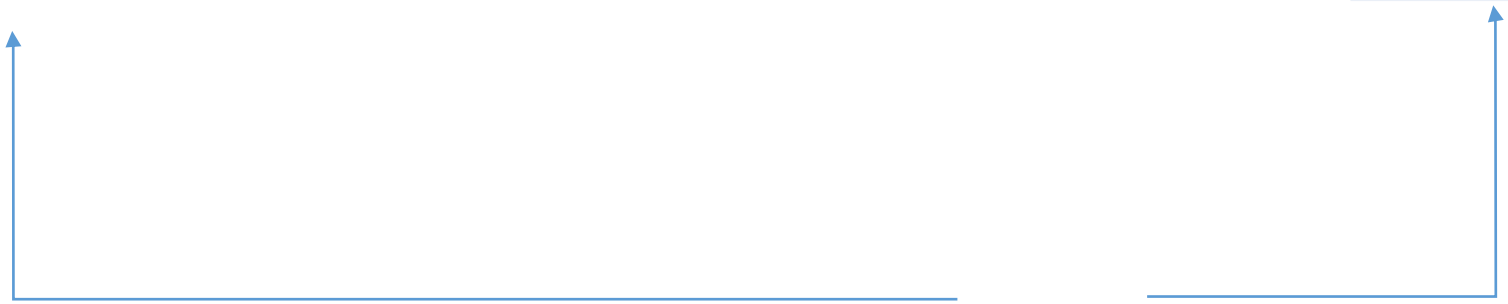
UPDATE Class SET Professor = "Yong Hack" WHERE Class_id = 6;

Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Yong Hack



id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5
8	2	5

Relation Table

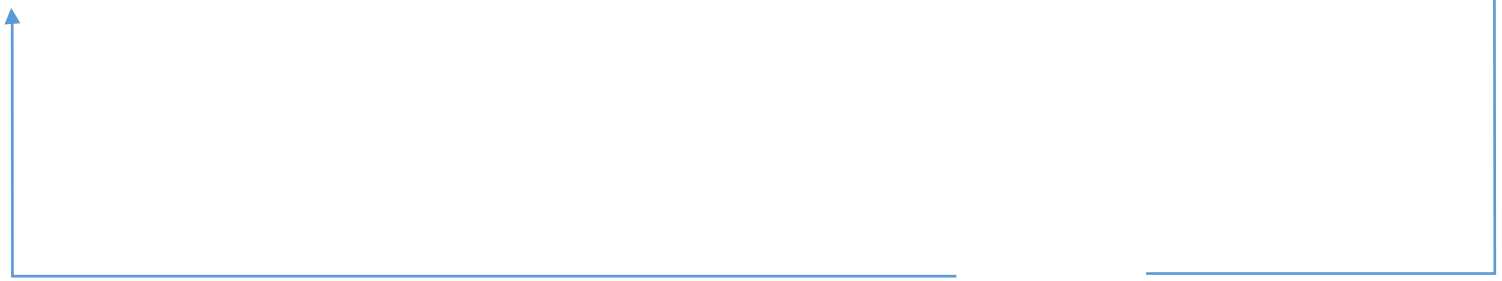
다시 구민이의 정보를 불러온다면?

Student_id	name	sex	age
1	강구민	남	25
2	김현욱	남	25
3	박태순	남	25
4	위성철	남	25
5	조찬민	남	25

Student Table

Class Table

Class_id	Class_name	Grade	Professor
1	SpringBoot	3	Park
2	Android Studio	3	Kim
3	Swift	3	Ahn
4	NodeJS	3	John
5	Embedded System	4	Giho park
6	DataBase	3	Yong Hack



id	Student_id	Class_id
1	1	2
2	1	6
3	2	1
4	3	1
5	4	3
6	5	4
7	5	5
8	2	5

Relation Table

다시 구민이의 정보를 불러온다면?

```
SELECT Class_id,Class_name,Grade_Professor Relation r LEFT JOIN Class c
ON r.Class_id = c.Class_id
WHERE r.Student_id = 1;
```

Class_id	Class_name	Grade	Professor
2	Android Studio	3	Kim
6	DataBase	3	Yong Hack

SQL 장단점

장점:

데이터의 일관성,무결성을 보증한다.

정규화에 따른 갱신 비용을 최소화한다.

단점:

데이터 스키마를 정확하게 설계해야함.

스키마를 따르지 않으면 데이터 추가 불가

관계가 복잡해지면 복잡한 Join 쿼리가 생길 수 있음.

NoSQL이란?

NoSQL(Not Only Structured Query Language) :
SQL의 정 반대. 비관계형 데이터베이스

특징:
1. 스키마와 관계가 없다.
2. SQL에서 레코드라고 불리는것이 NoSQL에서는 문서(documents) 라고 불린다.
3. 문서는 JSON데이터와 비슷한 형태를 가지고 있다.
4. 관련된 데이터들을 동일한 컬렉션에 저장한다.

Student Collection


```
_id: ObjectId("61ff749efc80c46c9a772111")
student_id: 1
name: "강구민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 2
      class_name: "Android Studio"
      grade: 3
      professor: "Park"
  1: Object
      class_id: 6
      class_name: "Database"
      grade: 3
      professor: "Ya dong ill"
```

```
_id: ObjectId("61ff7c04fc80c46c9a772112")
student_id: 2
name: "김현욱"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
```

```
_id: ObjectId("61ff7c39fc80c46c9a772113")
student_id: 3
name: "박태순"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
```

```
_id: ObjectId("61ff7c53fc80c46c9a772114")
student_id: 4
name: "위성철"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 3
      class_name: "Swift"
      grade: 3
      professor: "Ahn"
```

```
_id: ObjectId("61ff7cfefc80c46c9a772115")
student_id: 5
name: "조찬민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 4
      class_name: "NodeJS"
      grade: 3
      professor: "John"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```


만약 김현욱이 수강신청에서 임베디드시스템을 추가하기 위해서 실행해야하는 쿼리는?

Student Collection

```
_id: ObjectId("61ff749efc80c46c9a772111")
student_id: 1
name: "강구민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 2
      class_name: "Android Studio"
      grade: 3
      professor: "Park"
  1: Object
      class_id: 6
      class_name: "Database"
      grade: 3
      professor: "Ya dong ill"
```

```
_id: ObjectId("61ff7c04fc80c46c9a772112")
student_id: 2
name: "김현욱"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```

```
_id: ObjectId("61ff7c39fc80c46c9a772113")
student_id: 3
name: "박태순"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
```

```
_id: ObjectId("61ff7c53fc80c46c9a772114")
student_id: 4
name: "위성철"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 3
      class_name: "Swift"
      grade: 3
      professor: "Ahn"
```

```
_id: ObjectId("61ff7cfefc80c46c9a772115")
student_id: 5
name: "조찬민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 4
      class_name: "NodeJS"
      grade: 3
      professor: "John"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```

만약 김현욱이 수강신청에서 임베디드시스템을 추가하기 위해서 실행해야하는 쿼리는?

```
db.student.update(
 {student_id:2},
 {$push:{
   class : {
     class_id : 5,
     class_name : "Embedded System",
     grade : 4,
     professor : "Giho Park"
   }
 }
}
)
```

Student Collection


```
_id: ObjectId("61ff749efc80c46c9a772111")
student_id: 1
name: "강구민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 2
      class_name: "Android Studio"
      grade: 3
      professor: "Park"
  1: Object
      class_id: 6
      class_name: "Database"
      grade: 3
      professor: "Ya dong ill"
```

```
_id: ObjectId("61ff7c04fc80c46c9a772112")
student_id: 2
name: "김현욱"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```

```
_id: ObjectId("61ff7c39fc80c46c9a772113")
student_id: 3
name: "박태순"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
```

```
_id: ObjectId("61ff7c53fc80c46c9a772114")
student_id: 4
name: "위성철"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 3
      class_name: "Swift"
      grade: 3
      professor: "Ahn"
```

```
_id: ObjectId("61ff7cfefc80c46c9a772115")
student_id: 5
name: "조찬민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 4
      class_name: "NodeJS"
      grade: 3
      professor: "John"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```


강구민이 수강신청한 수업들의 정보를 알고싶다면?

Student Collection

```
_id: ObjectId("61ff749efc80c46c9a772111")
student_id: 1
name: "강구민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 2
      class_name: "Android Studio"
      grade: 3
      professor: "Park"
  1: Object
      class_id: 6
      class_name: "Database"
      grade: 3
      professor: "Ya dong ill"
```

```
_id: ObjectId("61ff7c04fc80c46c9a772112")
student_id: 2
name: "김현욱"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```

```
_id: ObjectId("61ff7c39fc80c46c9a772113")
student_id: 3
name: "박태준"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
```

```
_id: ObjectId("61ff7c53fc80c46c9a772114")
student_id: 4
name: "위성철"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 3
      class_name: "Swift"
      grade: 3
      professor: "Ahn"
```

```
_id: ObjectId("61ff7cfefc80c46c9a772115")
student_id: 5
name: "조찬민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 4
      class_name: "NodeJS"
      grade: 3
      professor: "John"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```

강구민이 수강신청한 수업들의 정보를 알고싶다면?
db.student.find({student_id:1},
{student_id : false, name : false, sex : false, age : false, class : true})

```
"class":[
  {
    "class_id":2,
    "class_name":"Android Studio",
    "grade":3,
    "professor":"Park"
  },
  {
    "class_id":6,
    "class_name":"DataBase",
    "grade":3,
    "professor":"Ya dong ill"
  }
]
```

Student Collection

```
_id: ObjectId("61ff749efc80c46c9a772111")
student_id: 1
name: "강구민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 2
      class_name: "Android Studio"
      grade: 3
      professor: "Park"
  1: Object
      class_id: 6
      class_name: "Database"
      grade: 3
      professor: "Ya dong ill"
```

```
_id: ObjectId("61ff7c04fc80c46c9a772112")
student_id: 2
name: "김현욱"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```

```
_id: ObjectId("61ff7c39fc80c46c9a772113")
student_id: 3
name: "박태순"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
```

```
_id: ObjectId("61ff7c53fc80c46c9a772114")
student_id: 4
name: "위성철"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 3
      class_name: "Swift"
      grade: 3
      professor: "Ahn"
```

```
_id: ObjectId("61ff7cfefc80c46c9a772115")
student_id: 5
name: "조찬민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 4
      class_name: "NodeJS"
      grade: 3
      professor: "John"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```

데이터베이스 교수가 Ya dong ill에서 Yong Hack으로 바꿔야 한다면?

Student Collection


```
_id: ObjectId("61ff749efc80c46c9a772111")
student_id: 1
name: "강구민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 2
      class_name: "Android Studio"
      grade: 3
      professor: "Park"
  1: Object
      class_id: 6
      class_name: "Database"
      grade: 3
      professor: "Yong Hack"
```

```
_id: ObjectId("61ff7c04fc80c46c9a772112")
student_id: 2
name: "김현욱"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```

```
_id: ObjectId("61ff7c39fc80c46c9a772113")
student_id: 3
name: "박태준"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 1
      class_name: "Spring Boot"
      grade: 3
      professor: "Kim"
```

```
_id: ObjectId("61ff7c53fc80c46c9a772114")
student_id: 4
name: "위성철"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 3
      class_name: "Swift"
      grade: 3
      professor: "Ahn"
```

```
_id: ObjectId("61ff7cfefc80c46c9a772115")
student_id: 5
name: "조찬민"
sex: "남"
age: 25
class: Array
  0: Object
      class_id: 4
      class_name: "NodeJS"
      grade: 3
      professor: "John"
  1: Object
      class_id: 5
      class_name: "Embedded System"
      grade: 4
      professor: "Giho Park"
```


데이터베이스 교수가 Ya dong ill에서 Yong Hack으로 바뀌야 한다면?
db.student.update({"class.class_name":"Database"},
{$set:{"class.professor":"Yong Hack"},{multi:true})

NoSQL 장단점

장점:

스키마와 관계가 없기때문에 유연한 데이터 모델링이 가능하다.

빠른 읽기/쓰기 속도를 자랑한다.

단점:

데이터가 여러 컬렉션에 중복되어 저장되었기 때문에, 수정하기위해서 데이터가 들어있는 모든 컬렉션을 수정해야한다.

SQL vs NoSQL

SQL :

1. 데이터의 변경이 잦은 서비스인경우
2. 스키마가 변경될일이 적고, 스키마가 사용자와 데이터에게 중요한 경우

NoSQL:

1. 데이터구조가 자주 변경되거나, 정확한 데이터구조를 알 수 없을경우
2. 데이터구조가 확장될 수 있는 경우
3. 읽기/쓰기는 자주하지만, 데이터변경이 자주 일어나지 않는경우
4. 막대한양의 데이터를 처리해야할 경우