mongoDB®

8. MongoDB활용

1. R과 mongoDB연결
2. python과 Mongodb 연결
3. Java JDBC로 MongoDB 연결

1. R과 MongoDB 연동

MongoDB에 접속하여 아래 데이터 입력

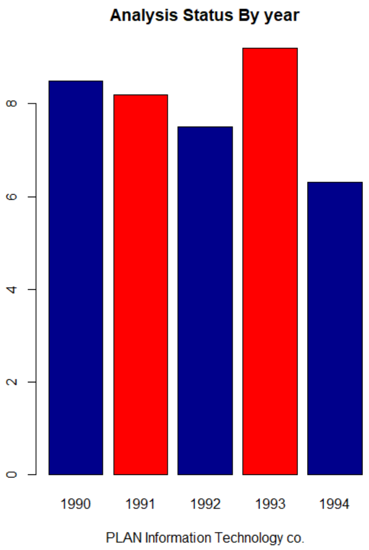
- 데이터베이스명 : test
- table명 :year_h

```
{ "_id" : 1990,"value" : 8.5 }
{ "_id" : 1991,"value" : 7.8 }
{ "_id" : 1992,"value" : 7.2 }
{ "_id" : 1993,"value" : 5.8 }
{ "_id" : 1994,"value" : 7.0 }
{ "_id" : 1995,"value" : 6.2 }
{ "_id" : 1996,"value" : 6.4 }
{ "_id" : 1997,"value" : 6.3 }
{ "_id" : 1998,"value" : 5.2 }
{ "_id" : 1999,"value" : 5.6 }
```

□ R을 실행한 후 MongoDB 관련 패키지를 설치

```
> install.packages(“mongolite")
> library(mongolite)
> year_h<-mongo(collection=“year_h“, db=“test")
> year_h<-data.frame(year=c(1990, 1991, 1992, 1993, 1994),
                        value=c(8.5, 8.2, 7.5, 9.2, 6.3))
> r_year$insert(year_h)
> result<-year$find()
```

```
barplot(result$value, main='Analysis Status By year',
        col=c("darkblue", "red"),
        names.arg=result.year,
        beside=TRUE, xlab="PLAN Information Technology co.")
```



- R을 통해 MongoDB 내의 컬렉션에 데이터를 조작
  - Update & Delete

```
r_year$update('{"year":1990}','{"$set":{"value":9.0}}')
r_year$find()

r_year$remove('{"year":1990}')
R_year$find()
```

Python과 MongoDB 연결

□ pymongo 설치

```
pip install pymongo
```

□ 몽고디비 접속하기

```
from pymongo import MongoClient
client = MongoClient() # 클래스 객체 할당
client = MongoClient('localhost', 27017)
# localhost: ip주소
# 27017: port 번호
```

```
#계정이 있는 경우
DB_HOST = 'XXX.XX.XX.XXX:27017'
DB_ID = 'root'
DB_PW = 'PW'
client = MongoClient('mongodb://%s:%s@%s' % (DB_ID, DB_PW, DB_HOST
```

Python과 MongoDB 연결

□ db 객체 할당받기

```
db = client["DB_이름"]
```

□ collection 객체 할당받기

```
collection = db["coll_이름"]
# collection = db.coll_이름
```

□ 도큐먼트 생성

```
import datetime
post = {
"author" : "Mike", "text" : "My first blog post!",
"tags" : ["mongodb", "python", "pymongo"], "date": datetime.datetime.utcnow()
}
```

□ 도큐먼트 insert하기

```
coll = db.collection
coll.insert(post)
# post_id = coll.insert(post)
```

```
coll = db.collection
coll.insertmany(post_list)
# post_id = coll.insert(post)
```

□ 콜렉션 목록 보기

```
coll_list = db.collection_names()
# [u'system.indexes', u'collection']
```

□ 도큐먼트 하나 가져오기

```
coll.find_one()
```

도큐먼트 가져오기

```
for post in coll.find():
        #######
```

도큐먼트 개수 세기

```
posts.count()
```

□ 예:

```
import pymongo
connection = pymongo.MongoClient("10.0.0.10", 27017)
db = connection.test_database
collection = db.emp
docs = collection.find()
for i in docs:
      print(i)
```

2. Java JDBC로 MongoDB 연결

□ Make a Connection

```
MongoClient mongoClient = new MongoClient();
MongoClient mongoClient = new MongoClient( "localhost" );
MongoClient mongoClient = new MongoClient( "localhost" , 27017 );
MongoClientURI connectionString =
                     new MongoClientURI("mongodb://localhost:27017");
MongoClient mongoClient = new MongoClient(connectionString);
```

□ Access a Database

```
MongoDatabase database = mongoClient.getDatabase("mydb");
```

□ Access a Collection

```
MongoCollection<Document> collection = database.getCollection("test");
```

Insert

- □ Create a Document

```
{
    "name" : "MongoDB",
    "type" : "database",
    "count" : 1,
    "versions": [ "v3.2", "v3.0", "v2.6" ],     "info" : { x : 203, y :
102 }
}
```

```
Document doc = new Document("name", "MongoDB")
    .append("type", "database") .append("count", 1)
    .append("versions", Arrays.asList("v3.2", "v3.0", "v2.6"))
    .append("info", new Document("x", 203).append("y", 102));
```

Insert a Document

```
collection.insertOne(doc);
```

Insert Multiple Documents

```
{ "i" : value }

List<Document> documents = new ArrayList<Document>();
for (int i = 0; i < 100; i++) {
    documents.add(new Document("i", i));
}

collection.insertMany(documents);
```

Count Documents in A Collection

```
System.out.println(collection.count());
```

Query the Collection

- **Find the First Document in a Collection**

```
Document myDoc = collection.find().first();
System.out.println(myDoc.toJson());
```

```
{
     "_id" : { "$oid" : "551582c558c7b4fbacf16735" },
     "name" : "MongoDB",
     "type" : "database",
     "count" : 1,
     "info" : { "x" : 203, "y" : 102 }
 }
```

□ Find All Documents in a Collection

```
MongoCursor<Document> cursor = collection.find().iterator(); try {
    while (cursor.hasNext()){
        System.out.println(cursor.next().toJson()); }
} finally {
    cursor.close();
}
```

while 대체 가능

```
for (Document cur : collection.find()) {
        System.out.println(cur.toJson());
}
```

□ Specify a Query Filter

```
BasicDBObject inQuery=new BasicDBObject();
inQuery.put("empno", empno);

FindIterable<Document> iterate=collection.find(inQuery);
```

Update_Delete

□ **update**

```
BasicDBObject inQuery=new BasicDBObject();
inQuery.put("empno", emp.empno);

BasicDBObject newDoc=new BasicDBObject()
.append("$set", new BasicDBObject("ename", emp.ename));
collection.updateOne(inQuery, newDoc);
```

□ **delete**

```
BasicDBObject inQuery=new BasicDBObject();
inQuery.put("empno", empno);
collection.deleteOne(inQuery);
```