**Mongodb的个人理解和总结**

**Mongodb是一个基于分布式文件存储的数据库。**

支持字段索引，优势在于查询功能强大，能存储海量数据，只支持单文档事务。

它面向集合存储，意思是数据是被分组存储在数据集合中的，每个集合在数据库中有唯一的标识名，集合的概念类似于关系型数据的表。

在集合中存储的是文档，被存储为键-值对的形式，键是每个文档的唯一标识，为字符串类型，值可以是各种的文件类型，这种存储形式为bson，文档的概念类似于数据库表中的一行数据。

**特点是：性能高，使用方便，存储数据方便。**

使用场景，做服务器的日志记录，查找方便，导出也方便。

存储监控数据，增删字段不需要修改表结构，

存储大量的商家信息。

**与mysql的比较，**

mysql查询语句是传统的sql语句，拥有成熟的体系海量数据处理时效率显著变慢。

mongodb：非关系型数据库，属于文档型数据库。（可以存放xml,json,bson类型的数据。存储方式：虚拟内存+持久化。

数据类型：null,布尔，数值，字符串，日期，数组，内嵌文档，对象id。

每个文档必须有一个\_id键，可以是任意类型，默认是Objectid,在一个集合里有唯一的\_id。

数据库操作：

新增数据库：use db1(db1为数据库名) #有这个数据就使用这个，没有则创建（很方便简洁）。查询数据：show dbs。

删除数据库：db.dropDatabase()

**集合（表）的增删改：**

新增：db.db1.info #db1.info是表名 或者直接插入文档，集合也会被创建。 db.table1.insert({'a',1})

查询：show tables

删除集合：db.table1.drop()

文档（表中一行记录）的增删改，

user1={

"name":"lin",

"age":25,

'hobbies':['music','read'],

'addr':{

'country':'China',

'city':'GZ'

}

}

新增一条数据：db.table1.insert(user1)

新增多条记录：db.table1.insertMany([user1,user2])

查询一条记录：db.table1.findOne()

查询所有记录：db.table1.find()

带条件查询：db.table1.find({"age":25})

比较运算'!=' db.table1.find({"age":{"$en":25}})

运算符'<' {"age":{"$lt":25}},'>'{"age":{"$gt":25}},'<='{"age":{"$lte":25}},'>='{"age":{"gte":25}}。

逻辑and db.table1.find( {"age":{"$gte":24,"$lte":25},"name":"lin"})

逻辑or db.table1.find({"or":[{"age":{"$gte":24,"$lte":25}},{"name":"lin"}]})

修改表中字段：db.table1.update({'age':25},{"name":"linqz"})

db.table1.update({"\_id":1},obj)

**Mongodb在springboot中的使用，**

1.首先是在项目中导包，

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-mongodb</artifactId>

</dependency>

2配置文件中配置mongodb,

spring.data.mongodb.host=120.67.195.135 //主机ip

spring.data.mongodb.port=27017 //mongodb的端口号

spring.data.mongodb.database=linDatabase 数据库名

或者

spring.data.mongodb.uri=mongodb://120.67.195.135 :27017/linDatabase

3使用

@Autowired

private MongoTemplate mongoTemplate;

新增：mongoTemplate.save(user);

根据条件查询

Query query = new Query(Criteria.where("name").is(name));

User user = mongoTemplate.findOne(query,User.class);

根据条件更新

Query query = new Query(Criteria.where("id").is(user.getId()));

Update update = new Update().set("name",user.getName()).set("password",user.getPassword());

UpdateResult result = mongoTemplate.updateFirst(query,update,User.class);

根据条件删除

Query query = new Query(Criteria.where("id").is(id));

mongoTemplate.remove(query,User.class)；