

```
//1.
```

```
db.student.drop()
```

```
db.student.insert({"name":"zhangsan","score":{"English":69,{"Math":86,{"Computer":77}}}})
```

```
db.student.insert({"name":"lisi","score":{"English":55,{"Math":100,{"Computer":88}}}})
```

```
//2
```

```
db.student.find()
```

```
//3
```

```
db.student.find({"name":"zhangsan"}, {"score":1})
```

```
//4
```

```
db.student.update({"name":"lisi", {"score":{"$set":{"Math":95}}}}
```

```
//5
```

```
a={  
    "name":"scofield",  
    "score":{"English":55,"Math":100,"Computer":88}  
}
```

```
collection.insert(a)
```

```
//6
```

```
cursor = db["student"].find({"name":"scofield"}, {"name":0})
```

```
//7
```

```
db.getCollection('users').find()
```

```
db.users.find({"age":17})
```

```
//8
```

```
db.users.find({"age":{"$gte":17,"$lte":20}})
```

```
//9
```

```
db.users.find({"$or":[{"age":17},{"age":20}]})
```

```
//10
```

```
db.users.find({"name":/0/})
```

```
//11
```

```
db.users.find({"age":{"$lt":21}},{"name":1,"age":1})
```

```
//12
```

```
db.users.find().sort({"age":-1})
```

```
//13
```

```
db.users.find().limit(3)
```

```
//14
```

```
db.users.find().skip(3).limit(3)
```

```
//15
```

```
15 db.users.aggregate([{"group":"_id":"$name"}])
```

```
db.users.find({"like":{"$and":[{"movies":"长津湖"}, {"movies":"战狼"}]}})
```

```
db.users.find({"$and":[{"like":{"movies":"长津湖"}}, {"like":{"movies":"战狼"}}]})
```

```
//16
```

```
db.users.find({"name":"WuKong"}, {"com":1}).limit(3)
```

```
17 db.users.find({"name":"WuKong"},"com":{"$slice":3}))
```

```
//17
```

```
db.users.find({"com":{"$exists":true}})
```

```
//18
```

```
db.users.updateOne({"name":"zhangsan5"},"$addToSet":{"like":"sing"})
```

```
//19
```

```
db.users.find({"com":{"uid":"001"},"content":"很好"})
```

```
//20?
```

```
20 db.users.find({"com.uid":"001"},"com.content":"很好"))
```

```
//21?
```

```
db.users.update({"name":""},"$inc":{"age":1}))
```

```
//22?
```

```
db.users.updateOne({"name":"zhangsan5"},"$pop":{"like":-1}))
```

```
//23
```

```
db.users.aggregate([{"$group":{"_id":"$sex","sum":{"$sum":1}}}))
```

```
//24.
```

```
db.users.aggregate([{"$group":{"_id":"$age","人数":{"$sum":1},"学生": {"$push":"$name"}}, {"$match":{"人数":{"$gt":3}}})
```

```
//25.
```

```
db.users.aggregate([{"$group":{"_id":"$name","总分":{"$sum":{"$sum":["score.chinese","score.math","score.english"]}}})
```

//26

索引，使用索引可快速访问数据库表中的特定信息。索引是对数据库表中一列或多列的值进行排序的一种结构，例如 employee 表的姓名（name）列。如果要按姓查找特定职员，与必须搜索表中的所有行相比，索引会帮助您更快地获得该信息。

索引的优点：

不需要做全表扫描，只需要扫描索引索引只存储了这个表的数据的一小部分，这小部分可以帮我们实现快速查询，因此扫描的时候只扫描这一小部分即可，如果将这小部分装载入内存中的话，速度会更快

//27

NoSQL 是非关系型数据库，NoSQL = Not Only SQL。

关系型数据库采用的结构化的数据，NoSQL 采用的是键值对的方式存储数据。在处理非结构化/半结构化的大数据时；在水平方向上进行扩展时；随时应对动态增加的数据项时可以优先考虑

使用 NoSQL 数据库。在考虑数据库的成熟度；支持；分析和商业智能；管理及专业性等问题时，应优先考虑关系型数据库。

//28

一、文档[document]

- 文档是一组键值对。
- 文档是 MongoDB 中数据的基本单元，非常类似关系型数据库中的行。多个键及其关联的值有序地放置在一起便是文档。

二、集合[collection]

- 在 MongoDB 中集合是一组文档。相当于关系型数据库中的表。
- 集合是无模式的。意味着集合中的文档可以是各式各样的。【关系数据库中，每一行都有同名的键，有对应相同类型的键值】