

MongoDB 索引 和 explain 的使用

主讲教师：（大地）

合作网站：www.itying.com

目录

一、索引基础.....	1
二、唯一索引.....	2
三、索引的一些参数.....	4
四、使用 explain	4
五、explain executionStats 查询具体的执行时间.....	5

一、索引基础

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,可以让我们查询数据库变得更快。MongoDB 的索引几乎与传统的关系型数据库一模一样,这其中也包括一些基本的查询优化技巧。

下面是创建索引的命令:

```
db.user.ensureIndex({"username":1})
```

获取当前集合的索引:

```
db.user.getIndexes()
```

删除索引的命令是:

```
db.user.dropIndex({"username":1})
```

在 MongoDB 中，我们同样可以创建**复合索引**，如：

数字 1 表示 username 键的索引按升序存储，-1 表示 age 键的索引按照降序方式存储。

```
db.user.ensureIndex({"username":1, "age":-1})
```

该索引被创建后，基于 username 和 age 的查询将会用到该索引，或者是基于 username 的查询也会用到该索引，但是只是基于 age 的查询将不会用到该复合索引。因此可以说，如果想用到复合索引，**必须在查询条件中包含复合索引中的前 N 个索引列**。然而如果查询条件中的键值顺序和复合索引中的创建顺序不一致的话，MongoDB 可以智能的帮助我们调整该顺序，以便使复合索引可以为查询所用。如：

必须
从头
开始

```
db.user.find({"age": 30, "username": "stephen"})
```

对于上面示例中的查询条件，MongoDB 在检索之前将会动态的调整查询条件文档的顺序，以使该查询可以用到刚刚创建的复合索引。

对于上面创建的索引，MongoDB 都会根据索引的 keyname 和索引方向为新创建的索引自动分配一个索引名，下面的命令可以在创建索引时为其指定索引名，如：

```
db.user.ensureIndex({"username":1}, {"name": "userindex"})
```

随着集合的增长，需要针对查询中大量的排序做索引。如果没有对索引的键调用 sort，MongoDB 需要将所有数据提取到内存并排序。因此在做无索引排序时，如果数据量过大以致无法在内存中进行排序，此时 MongoDB 将会报错。

二、唯一索引

在缺省情况下创建的索引均不是唯一索引。下面的示例将创建唯一索引，如：

```
db.user.ensureIndex({"userid":1}, {"unique":true})
```

如果再次插入 userid 重复的文档时，MongoDB 将报错，以提示插入重复键，如：

```
db.user.insert({"userid":5})  
db.user.insert({"userid":5})
```

E11000 duplicate key error index: user.user.\$userid_1 dup key: { : 5.0 }

如果插入的文档中不包含 `userid` 键，那么该文档中该键的值为 `null`，如果多次插入类似的文档，MongoDB 将会报出同样的错误，如：

```
db.user.insert({"userid1":5})  
db.user.insert({"userid1":5})
```

E11000 duplicate key error index: user.user.\$userid_1 dup key: { : null }

如果在创建唯一索引时已经存在了重复项，我们可以通过下面的命令帮助我们在创建唯一索引时消除重复文档，仅保留发现的第一个文档，如：

先删除刚刚创建的唯一索引。

```
db.user.dropIndex({"userid":1})
```

插入测试数据，以保证集合中有重复键存在。

```
db.user.remove()  
db.user.insert({"userid":5})  
db.user.insert({"userid":5})
```

重新创建唯一索引

```
db.user.ensureIndex({"userid":1},{ "unique":true })
```

```
db.test.ensureIndex({"userid":1}, {"unique":true})

{"ok" : 0,
  "errmsg" : "E11000 duplicate key error collection: itying.test index: userid_1 dup key: { : 5.0 }",
  "code" : 11000,
  "codeName" : "DuplicateKey"}
```

我们同样可以创建复合唯一索引，即保证复合键值唯一即可。如：

```
db.user.ensureIndex({"userid":1,"age":1}, {"unique":true})
```

三、索引的一些参数

Parameter	Type	Description
background	Boolean	建索引过程会阻塞其它数据库操作，background可指定以后台方式创建索引，即增加"background" 可选参数。"background" 默认值为false。
unique	Boolean	建立的索引是否唯一。指定为true创建唯一索引。默认值为false。
name	string	索引的名称。如果未指定，MongoDB的通过连接索引的字段名和排序顺序生成一个索引名称。
dropDups	Boolean	在建立唯一索引时是否删除重复记录,指定 true 创建唯一索引。默认值为 false。

如果在为已有数据的文档创建索引时，可以执行下面的命令，以使 MongoDB 在后台创建索引，这样的创建时就不会阻塞其他操作。但是相比而言，以阻塞方式创建索引，会使整个创建过程效率更高，但是在创建时 MongoDB 将无法接收其他的操作。

```
db.user.ensureIndex({"username":1}, {"background":true})
```

四、使用 explain

explain 是非常有用的工具，会帮助你获得查询方面诸多有用的信息。只要对游标调用该方法，就可以得到查询细节。explain 会返回一个文档，而不是游标本身。如：

```
> db.user.find({"username":"zhangsan5000"}).explain();
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "news.user",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "username" : {
        "$eq" : "zhangsan5000"
      }
    },
    "winningPlan" : {
      "stage" : "COLLSCAN",
      "filter" : {
        "username" : {
          "$eq" : "zhangsan5000"
        }
      },
      "direction" : "forward"
    },
    "rejectedPlans" : [ ]
  },
  "serverInfo" : {
    "host" : "MICROSO-JR3EE06",
    "port" : 27017,
    "version" : "3.4.1",
    "gitVersion" : "5e103c4f5583e2566a45d740225dc250baacfb7"
  },
  "ok" : 1
}
```

explain 会返回查询使用的索引情况，耗时和扫描文档数的统计信息。

五、explain executionStats 查询具体的执行时间

```
db.tablename.find().explain( "executionStats" )
```

关注输出的如下数值: explain.executionStats.executionTimeMillis

