# MongoDB 特殊索引

李焕贞

河北师范大学软件学院

https://docs.mongodb.com/manual/reference/operator/query/ near/#op. S near

## 本章大纲

- ▶2dsphere 球面索引
- ▶2d平面索引
- >全文索引

## 2dsphere 球面索引-地理空间索引的概念

地理空间索引可以对地理位置数据建立索引条目, MongoDB原生支持地理位置索引,可以直接用于位置距离计算和查询。

MongoDB对原生支持的查询函数在性能上进行高度的优化,与其它的数据库相比,具有很高的查询效率。

MongoDB支持两种类型的地理空间索引:

- •球面 2dsphere 球面索引
- 平面 2d索引

## 2dsphere 球面索引-概念

2dsphere球面索引在一个类似于地球的球形表面上来计算地理数据,在进行数据查询时,MongoDB使用的是WGS84地心坐标系.

2dsphere索引支持两种类型的数据格式:

- GeoJSON对象
- 普通坐标(Legacy Coordinate Pairs)对象

## 2dsphere 球面索引-GeoJSON对象

GeoJSON是一种对各种地理数据结构进行编码d数据格式,使用基于 JavaScript对象的表示方法,可以用于地理空间信息的交换。

2dsphere索引使用GeoJSON对象来存储位置数据,GeoJSON对象可以表示地理特征,支持多种几何类型,具体包括:

- Point LineString Polygon
- MultiPoint MultiLineString MutiPolygon GeometryCollection

## 2dsphere 球面索引-GeoJSON对象

Geo.JSON对象包含一个type字段和一个coordinates字段,格式如下: {<location field>:{ type:"GeoJSON type", //几何图形类型 coordinates:<coordinate> //[经度,维度] }} 点 {loc: {type:"Point", coordinates:[40,5]}} 线段 {loc: {type:"LineString", coordinates:[[40,5],[41,6]]}} 面 {loc: {type:"Polygon", coordinates:[[[0,0],[3,6] [6,1],[0,0]]]}}

## 2dsphere 球面索引-创建

创建球面索引使用createIndex函数,将索引字段设置为2dsphere 即可。 在创建索引时,可以为其制定2dsphereIndexVersion选项。例如: db.test.createIndex({loc:"2dsphere",2dsphereIndexVersion:3})

db.test.createIndex({loc:"2dsphere"})

## 2dsphere 球面索引-创建

#### 创建球面索引时,需要注意的几个问题:

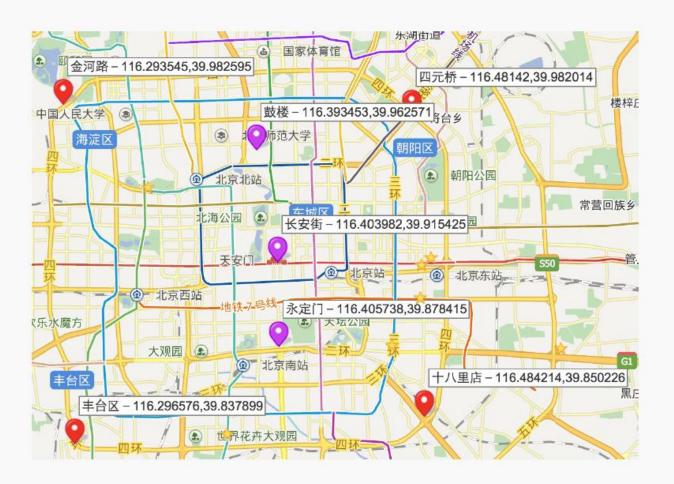
- 只能在拥有GeoJSON和普通坐标对对象的字段上创建球面索引,而且一旦在某个字段创建了球面索引,该字段只能用来存储以上两种格式的地理信息数据
- •默认情况下,球面索引是稀疏的,并且会自动忽略spare:true选项
- 当复合索引中包含球面索引字段时,只有球面索引可以决定是否引用一 条文档
- 在分片集合上,球面索引不能作为片键来使用

## 2dsphere 球面索引-查询

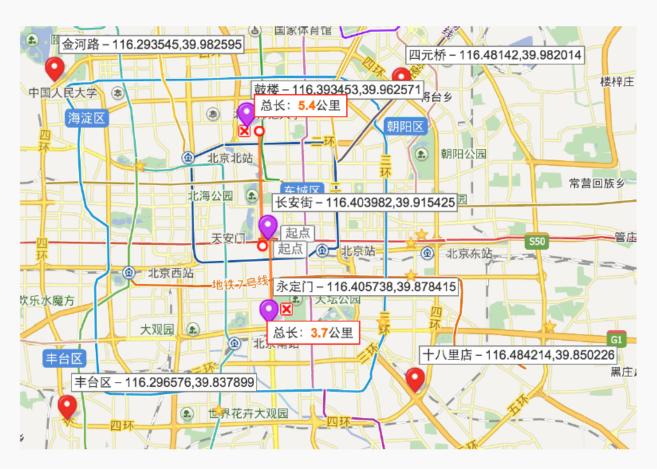
下面列出2dsphere索引查询有关的操作符:

操作符	作用	操作符	作用
\$geoWithin	包含关系	\$near	接近点判断
\$geoIntersects	相交关系	\$nearSphere	接近点判断(球面图形)

### 2dsphere 球面索引-使用



### 2dsphere 球面索引-使用



## 本章大纲

- ▶2dsphere 球面索引
- ▶2d平面索引
- >全文索引

## 2d平面索引-简介

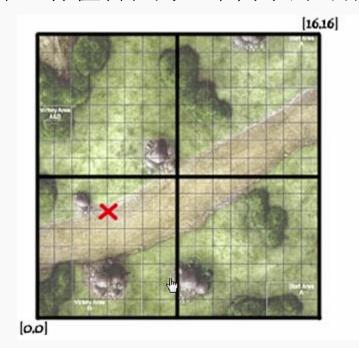
对于非球面的平面,可以使用2d索引代替2dsphere索引,2d索引用于扁平表面的二维坐标平面内。

2d索引支持在欧几里德空间内计算关系,存储的数据只能使用普通坐标对(Legacy Coordinate Pairs)对象。

普通坐标对对象的数据格式如下: {loc:[<longitude>,<latitude>]}

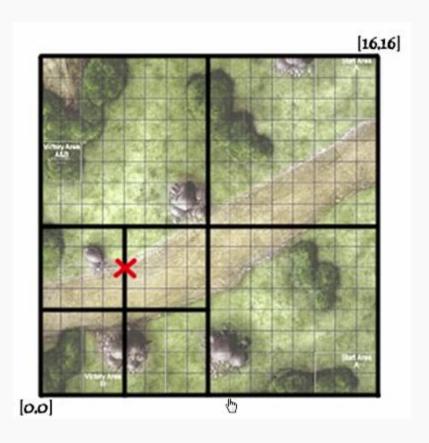
#### 2d平面索引-存储原理

MongoDB使用**geoHash**算法为地图位置数据建立索引,该算法将一个二维坐标值转化为一个简单的可排序可比较的字符串。(4,6)



01	11	
00	10	

## 2d平面索引-存储原理



## 2d平面索引-存储原理

最终得到[4,6]点的geohash值为: 00110100

mongoDB默认划分26次 [16.16] [16,16] [0,0] 0.0

#### 2d平面索引-创建

创建2d索引使用createIndex函数, 语法如下:

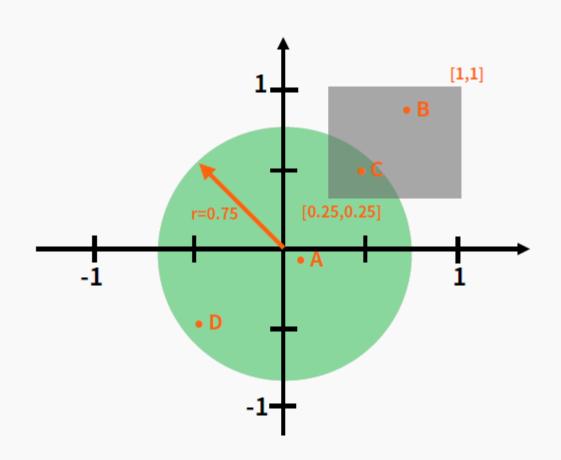
db.<collection>.createIndex(

{<location field>:"2d"},

{min:<lower bound>,max:<upper bound>,bits:<bit precision>})

- 1、第一个文档参数用来设置索引字段,将平面索引字段的值设置为2d
- 2、第二个可选参数可以设置三个选项:
  - min 设置最小的边界
  - max 设置最大的边界
  - bits 设置精确度,默认情况下, 2d索引的精度是26位

## 2d平面索引-使用



A [0.001, -0.002]

B [0.75, 0.75]

C [0.5, 0.5]

D [-0.5, -0.5]

#### 2d平面索引-使用

\$geoWithin 专门用来查询地理索引

例1: 盒子查找

\$box 参数为两个数值的数组,第一个左下角的点,第二个右上角的点

db. place. find({"loc.coor": {\$geoWithin: {\$box:[[-100, -100], [100, 100]]}}}))

例2: 圆形查找\$centor 第一个参数是x,y, 第二个参数是半径大小

db. place. find({"loc. coor": {\$geoWithin: {\$center: [[-73, 40], 10]}}})

例3:多边形查找\$polygon,每一个值都是一个点,最后一个值和第一个值连起来,

形成一个封闭的多边形

db. place. find({"loc. coor": {\$geoWithin: {\$polygon: [[0, 0], [3, 6], [6, 0]}}})

## 本章大纲

- ▶2dsphere 球面索引
- ▶2d平面索引
- >全文索引

## MongoDB全文索引-简介

文本索引,用于从大量的文本(文章、博客)中搜索出所需要的内容,它对每一个词建立一个索引,指明该词在整段文本中出现的位置和次数。当用户查询时,检索程序就根据事先建立的索引进行查找。这种技术主要用于搜索引擎,站内搜索等。

## MongoDB全文索引-原理

```
全文索引覆盖所有的字符串或者数组中的字符串,进行全文匹配
db.collection.createIndex( { field: "text" } )
使用通配符进行文本索引的创建
db.collection.createIndex( { "$**": "text" } )
对text index创建复合索引
db.collection.createIndex( { a: 1, "$**": "text" } )
一个集合只能有一个全文索引,可以对多个字段创建全文索引
text index默认是稀疏的,因此会忽略sparse:true
```

## MongoDB全文索引-原理

- 1. 一个集合只能有一个全文索引,但是可以对多个字段创建全文索引
- 2. text index默认是稀疏的,因此会忽略sparse:true
- 3. 如果一个文档缺少text index涉及到的field,那么查询的时候会跳过这个文档
- 4. 在插入时, mongoDB会将文档插入, 但是并不会建立text index
- 5. 无法通过hint方法进行强制索引
- 6. compound text index不能包含其他特殊的索引,例如: multikey index或者 geo index
- 7. 无法对伴有text index查询结果集进行text index排序

### MongoDB全文索引-原理

```
例1: 匹配一个单词
db. articles. find({$text: {$search: "coffee"}})
例2: 匹配多个单词,单词之间通过空格进行分离,任意一个即可
db. articles. find({\$text:\{\$search:\"coffee cake\"\}\})
例3: 匹配一个短句, 需要用\"\" 转义字符+双引号
db.articles.find({$text:{$search:"\"a cake \""}})
例4: 匹配包含一个单词,不包含另一个单词
db.articles.find({$text:{$search:"coffee -shop"}})
例5: 匹配大小写
db.articles.find({$text:{$search:"coffee", $caseSensitive:true}})
```