

MongoDB 聚合函数的使用

聚合函数的使用

MongoDB中聚合(aggregate)主要用于处理数据(诸如统计平均值, 求和等), 并返回计算后的数据结果。

aggregate() 方法的基本语法格式如下所示:

```
>db.COLLECTION_NAME.aggregate(AGGREGATE_OPERATION)
```

求和: 求男女人数的总数

```
> db.user.aggregate(
... [
...   {
...     "$group": {
...       "_id": "$gender",
...       "total": {
...         "$sum": 1
...       }
...     }
...   }
... ]
... )
{ "_id" : 2, "total" : 2 }
{ "_id" : 1, "total" : 7 }
```

求平均值: 求男女年龄的平均值

```
> db.user.aggregate(
... [
...   {
...     "$group": {
...       "_id": "$gender",
...       "age": {
...         "$avg": "$age"
...       }
...     }
...   }
... ]
... )
{ "_id" : 2, "age" : 55 }
{ "_id" : 1, "age" : 60.285714285714285 }
```

求最大值 & 最小值 : 求男女中年龄最小(大)的年龄

```
> db.user.aggregate(
... [
...   {
...     "$group": {
...       "_id": "$gender",
...       "age": {
...         "$min": "$age"
...       }
...     }
...   }
... ]
... )
{ "_id" : 2, "age" : 25 }
{ "_id" : 1, "age" : 25 }
> db.user.aggregate(
... [
...   {
...     "$group": {
...       "_id": "$gender",
...       "age": {
...         "$max": "$age"
...       }
...     }
...   }
... ]
... )
{ "_id" : 2, "age" : 85 }
{ "_id" : 1, "age" : 117 }
```

在结果文档中插入值到一个数组中

```

> db.user.aggregate(
... [
...   {
...     "$group": {
...       "_id": "$gender",
...     "ages": {
...       "$push": "$age"
...     },
...     "avg": {
...       "$avg": "$age"
...     }
...   }
... ]
... )
{ "_id" : 2, "ages" : [ 85, 25 ], "avg" : 55 }
{ "_id" : 1, "ages" : [ 26, 25, 25, 88, 116, 117, 25 ], "avg" : 60.285714285714285 }
/

```

管道

管道在Unix和Linux中一般用于将当前命令的输出结果作为下一个命令的参数。

MongoDB的聚合管道将MongoDB文档在一个管道处理完后将结果传递给下一个管道处理。管道操作是可以重复的。

表达式：处理输入文档并输出。表达式是无状态的，只能用于计算当前聚合管道的文档，不能处理其它的文档。

聚合框架中常用的几个操作：

- \$project：修改输入文档的结构。可以用来重命名、增加或删除域，也可以用于创建计算结果以及嵌套文档。
- \$match：用于过滤数据，只输出符合条件的文档。\$match使用MongoDB的标准查询操作。
- \$limit：用来限制MongoDB聚合管道返回的文档数。
- \$skip：在聚合管道中跳过指定数量的文档，并返回余下的文档。
- \$unwind：将文档中的某一个数组类型字段拆分成多条，每条包含数组中的一个值。
- \$group：将集合中的文档分组，可用于统计结果。
- \$sort：将输入文档排序后输出。
- \$geoNear：输出接近某一地理位置的有序文档。

多管道使用示例：

```

db.user.aggregate(
[
  {
    $match: {age: { $gt: 0, $lte: 100 }}
  },
  {
    "$group": {
      id: "$gender",
      "ages": {
        "$addToSet": "$age"
      },
      "avg": {
        "$avg": "$age"
      }
    }
  }
]
)
> db.user.aggregate(
... [
...   {
...     $match: {
...       age: {
...         $gt: 0, $lte: 100
...       }
...     },
...     {
...       "$group": {
...         _id: "$gender",
...       "ages": {
...         "$addToSet": "$age"
...       },
...       "avg": {
...         "$avg": "$age"
...       }
...     }
...   }
... ]
... )
{ "_id" : 2, "ages" : [ 25, 85 ], "avg" : 55 }
{ "_id" : 1, "ages" : [ 88, 25, 26 ], "avg" : 37.8 }
/

```

