عنوان: **MongoDB #13**
نویسنده: برات جوادی
تاریخ: ۱۳۹۳/۱۱/۱۹ ۲۲:۳۵
آدرس: www.dotnettips.info
گروه‌ها: NoSQL, Database, mongodb

توابع جمعی در MongoDB

عملگرهای جمعی، رکوردهای اطلاعات را پردازش می‌کنند و نتیجه‌های محاسبه شده را برمی‌گردانند. عملیات جمعی مقادیر چندین سند را باهم گروه بندی می‌کند و می‌تواند یک نوع از عملگرها را روی اطلاعات دسته بندی شده انجام دهد تا یک نتیجه‌ی واحد را برگرداند. در sql، دستور count(*) همراه Group by معادل یک تابع جمعی در MongoDB است.

متد () aggregate

برای توابع جمعی در MongoDB باید از متد ()aggregate استفاده کنید.

گرامر

گرامر پایه متد ()aggregate به صورت زیر است:

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

مثال

در این مجموعه، داده‌های زیر را دارید:

```
{
   _id: ObjectId(7df78ad8902c)
   title: 'MongoDB Overview',
   description: 'MongoDB is no sql database',
   by_user: 'user1',
   url: 'http://www.site.com',
   tags: ['mongodb', 'database', 'NoSQL'],
   likes: 100
},
{
   _id: ObjectId(7df78ad8902d)
   title: 'NoSQL Overview',
   description: 'No sql database is very fast',
   by_user: 'user1',
   url: 'http://www.site.com',
   tags: ['mongodb', 'database', 'NoSQL'],
   likes: 10
},
{
   _id: ObjectId(7df78ad8902e)
   title: 'Neo4j Overview',
   description: 'Neo4j is no sql database',
   by_user: 'Neo4j',
   url: 'http://www.neo4j.com',
   tags: ['neo4j', 'database', 'NoSQL'],
   likes: 750
},
```

حالا اگر بخواهید از مجموعه‌ی بالا یک لیست را که تعداد دوره‌های نوشته شده توسط هر کاربر را نمایش می‌دهد، استخراج کنید، باید ار متد () aggregate به صورت زیر استفاده نمائید:

```
> db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$sum : 1}}}])
{
   "result" : [
      {
         "_id" : "user1",
         "num_tutorial" : 2
      },
      {
         "_id" : "Neo4j",
         "num_tutorial" : 1
      }
   ],
   "ok" : 1
}
>
```

معادل کوئری بالا در sql بصورت زیر خواهد بود:

```
select by_user, count(*) from mycol group by by_user
```

در مثال بالا، سندهای گروه بندی شده‌ی توسط فیلد by_user را داریم و در هر اجرای by_user مقدار قبلی جمع کلی افزایش می‌یابد. در اینجا لیست عبارت‌های جمعی موجود، آمده است.

عبارت	توضیحات	مثال
$sum	مقدار تعیین شده از همه سندهای مجموعه را جمع می‌کند.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$sum : "$likes"}}}])
$avg	میانگین همه مقادیر بدست آمده از سندهای مجموعه را محاسبه می‌کند.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$avg : "$likes"}}}])
$min	کمترین مقادیر مشابه را از همه سندهای مجموعه، بر می‌گرداند.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$min : "$likes"}}}])
$max	بیشترین مقادیر مشابه را از همه سندهای مجموعه، بر می‌گرداند.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$max : "$likes"}}}])
$push	یک مقدار را در سند نتیجه، در یک آرایه درج می‌کند.	db.mycol.aggregate([{$group : {_id : "$by_user", url : {$push: "$url"}}}])
$addToSet	یک مقدار را در سند نتیجه در یک آرایه درج می‌کند، اما مقدار تکراری ایجاد نمی‌کند.	db.mycol.aggregate([{$group : {_id : "$by_user", url : {$addToSet : "$url"}}}])
$first	اولین سند از اسناد را برطبق گروه بندی بر می‌گرداند. معمولا این عبارت بعد از عبارت‌های مرتب سازی مرحله‌ای استفاده می‌شود.	db.mycol.aggregate([{$group : {_id : "$by_user", first_url : {$first : "$url"}}}])
$last	آخرین سند از اسناد را برطبق گروه بندی بر می‌گرداند. معمولا این عبارت بعد از عبارت‌های مرتب سازی مرحله‌ای استفاده می‌شود.	db.mycol.aggregate([{$group : {_id : "$by_user", last_url : {$last : "$url"}}}])

مفهوم Pipeline

در Command shell یونیکس، خط لوله (Pipeline) به معنی امکان اجرای یک عملیات روی چندین ورودی و استفاده از خروجی بعنوان ورودی برای دستور بعدی و ادامه‌ی آن است. MongoDB نیز این مفهوم را در چارچوب توابع جمعی پشتیبانی می‌کند. یک مجموعه از مراحل وجود دارند که هرکدام از آنها یک مجموعه از اسناد را بعنوان ورودی می‌گیرند و یک مجموعه از سند را بعنوان نتیجه (یا نتیجه را بعنوان سند JSON در پایان خط لوله) ارائه می‌دهند. این عمل به نوبه خود می‌تواند برای مرحله بعد و یا مراحل بعدی، استفاده شود.

مراحل ممکن در چارچوب توابع جمعی در زیر آمده اند:

$project : برای انتخاب چندین فیلد از یک مجموعه استفاده می‌شود.

$match : این یک عملگر فیلترگذاری است که می‌تواند میزان اسنادی را که بعنوان ورودی در مرحله بعد گرفته می‌شوند، کاهش

دهد.

$group : این همان تابع جمعی است که در بالا توضیح داده شد.

$skip : توسط این عبارت، در یک لیست بدست آمده (نتیجه)، می‌توانید از لیست اسناد بصورت روبه جلو صرفنظر کنید.

$limit : این عبارت تعداد اسناد را توسط عدد گرفته شده، از موقعیت فعلی برای نمایش محدود می‌کند.

$unwind : این عبارت برای باز کردن (unwind) سندی که از آرایه‌ها بهره گیری می‌کند استفاده می‌شود. وقتی از آرایه استفاده می‌کنید، داده از نوع پیش پیوست (Pre-joined) است و با این نوع داده، این عمل برای داشتن سندهای اختصاصی نا تمام خواهد ماند. بنابراین با این مرحله می‌توانید میزان اسناد را برای مرحله بعد افزایش دهید.