**MongoDB的高级查询aggregate聚合管道**

**主讲教师：（大地）**

**合作网站：[www.itying.com](http://www.itying.com) （IT营）**

**我的专栏：<https://www.itying.com/category-79-b0.html>**

[一、 MongoDB 聚合管道（Aggregation Pipeline）](#_Toc1829_WPSOffice_Level1) [1](#_Toc1829_WPSOffice_Level1)

[二、 MongoDB Aggregation 管道操作符与表达式](#_Toc24375_WPSOffice_Level1) [2](#_Toc24375_WPSOffice_Level1)

[三、  数据模拟](#_Toc14394_WPSOffice_Level1) [4](#_Toc14394_WPSOffice_Level1)

[四、  $project](#_Toc29073_WPSOffice_Level1) [4](#_Toc29073_WPSOffice_Level1)

[五、  $match](#_Toc12121_WPSOffice_Level1) [5](#_Toc12121_WPSOffice_Level1)

[六、  $group](#_Toc21450_WPSOffice_Level1) [6](#_Toc21450_WPSOffice_Level1)

[七、  $sort](#_Toc15243_WPSOffice_Level1) [6](#_Toc15243_WPSOffice_Level1)

[八、  $limit](#_Toc31161_WPSOffice_Level1) [7](#_Toc31161_WPSOffice_Level1)

[九、  $skip](#_Toc16593_WPSOffice_Level1) [8](#_Toc16593_WPSOffice_Level1)

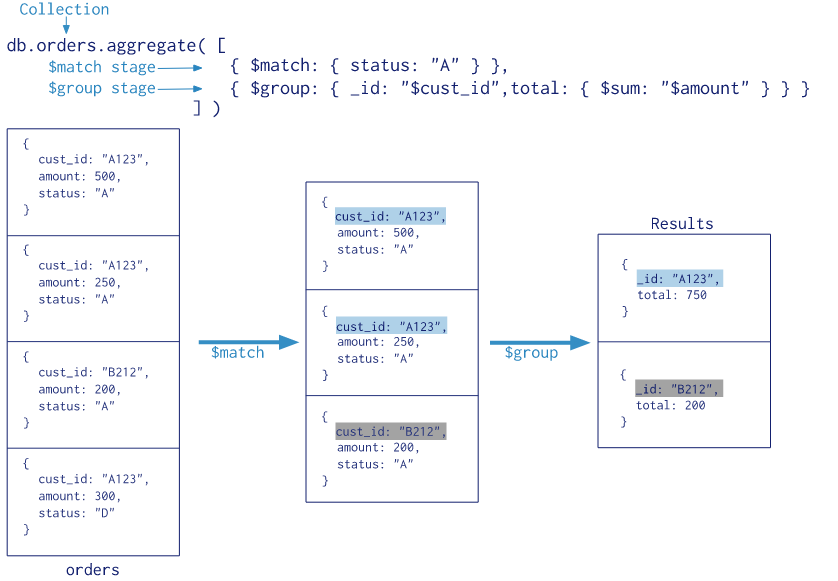
[十、  $lookup 表关联](#_Toc213_WPSOffice_Level1) [9](#_Toc213_WPSOffice_Level1)

### 一、MongoDB 聚合管道（Aggregation Pipeline）

使用聚合管道可以对集合中的文档进行变换和组合。

**实际项目：**表关联查询、数据的统计。

MongoDB 中使用 db.COLLECTION\_NAME.**aggregate([{<stage>},...])** 方法来构建和使用聚合管道。先看下官网给的实例，感受一下聚合管道的用法。



### MongoDB Aggregation 管道操作符与表达式

|  |  |  |
| --- | --- | --- |
| **管道操作符** | **Description** |  |
| $project | 增加、删除、重命名字段 |  |
| $match | 条件匹配。只满足条件的文档才能进入下一阶段 |  |
| $limit | 限制结果的数量 |  |
| $skip | 跳过文档的数量 |  |
| $sort | 条件排序。 |  |
| $group | 条件组合结果 统计 |  |
| $lookup | [$lookup 操作符](https://jira.mongodb.org/browse/SERVER-19095" \t "_blank) 用以引入其它集合的数据 （表关联查询） |  |

**SQL和NOSQL对比:**

|  |  |
| --- | --- |
| WHERE | $match |
| GROUP BY | $group |
| HAVING | $match |
| SELECT | $project |
| ORDER BY | $sort |
| LIMIT | $limit |
| SUM() | $sum |
| COUNT() | $sum |
| join | $lookup |

**管道表达式:**

管道操作符作为“键”,所对应的“值”叫做管道表达式。

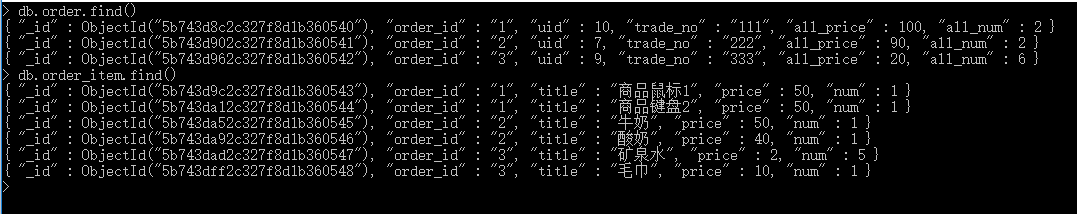
例如{$match:{status:"A"}}，$match称为管道操作符，而status:"A"称为管道表达式，是管道操作符的操作数(Operand)。

每个管道表达式是一个文档结构，它是由字段名、字段值、和一些表达式操作符组成的。

|  |  |
| --- | --- |
| **常用表达式操作符** | **Description** |
| $addToSet | 将文档指定字段的值去重 |
| $max | 文档指定字段的最大值 |
| $min | 文档指定字段的最小值 |
| $sum | 文档指定字段求和 |
| $avg | 文档指定字段求平均 |
| $gt | 大于给定值 |
| $lt | 小于给定值 |
| $eq | 等于给定值 |

### 数据模拟

|  |
| --- |
| db.order.insert({"order\_id":"1","uid":10,"trade\_no":"111","all\_price":100,"all\_num":2})  db.order.insert({"order\_id":"2","uid":7,"trade\_no":"222","all\_price":90,"all\_num":2})  db.order.insert({"order\_id":"3","uid":9,"trade\_no":"333","all\_price":20,"all\_num":6})  db.order\_item.insert({"order\_id":"1","title":"商品鼠标1","price":50,num:1})  db.order\_item.insert({"order\_id":"1","title":"商品键盘2","price":50,num:1})  db.order\_item.insert({"order\_id":"1","title":"商品键盘3","price":0,num:1})  db.order\_item.insert({"order\_id":"2","title":"牛奶","price":50,num:1})  db.order\_item.insert({"order\_id":"2","title":"酸奶","price":40,num:1})  db.order\_item.insert({"order\_id":"3","title":"矿泉水","price":2,num:5})  db.order\_item.insert({"order\_id":"3","title":"毛巾","price":10,num:1}) |

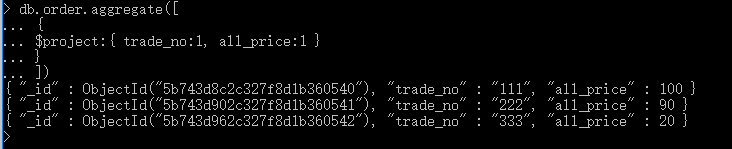


### $project

修改文档的结构，可以用来重命名、增加或删除文档中的字段。

要求查找order只返回文档中 trade\_no和 all\_price字段

|  |
| --- |
| db.order.aggregate([  {  $project:{ trade\_no:1, all\_price:1 }  }  ]) |

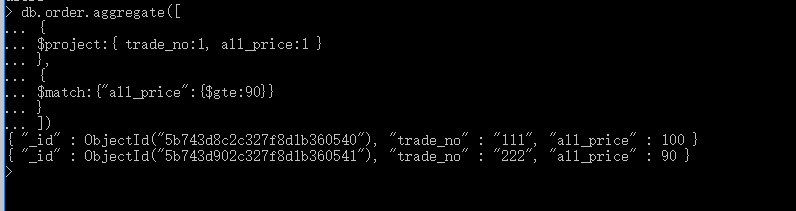


### $match

**作用**

用于过滤文档。用法类似于 find() 方法中的参数。

|  |
| --- |
| db.order.aggregate([  {  $project:{ trade\_no:1, all\_price:1 }  },  {  $match:{"all\_price":{$gte:90}}  }  ]) |

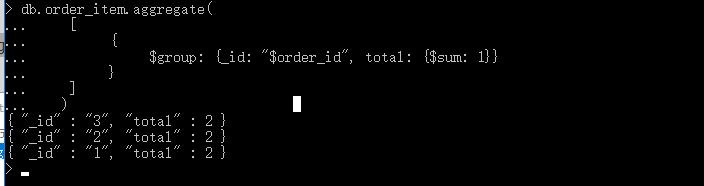


### $group

将集合中的文档进行分组，可用于统计结果。

统计每个订单的订单数量，按照订单号分组

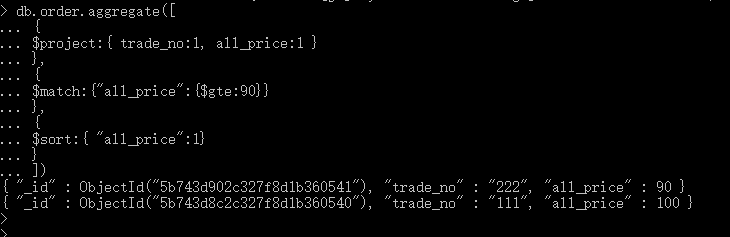
|  |
| --- |
| db.order\_item.aggregate(  [  {  $group: {\_id: "$order\_id", total: {$sum: "$num"}}  }  ]  ) |

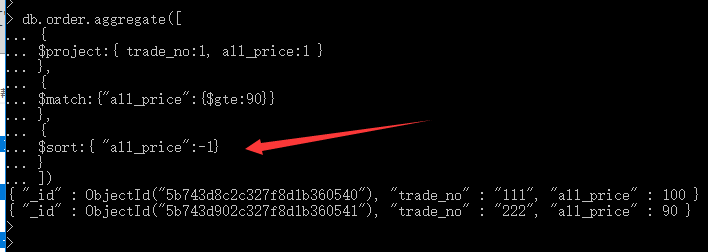


### $sort

将集合中的文档进行排序。

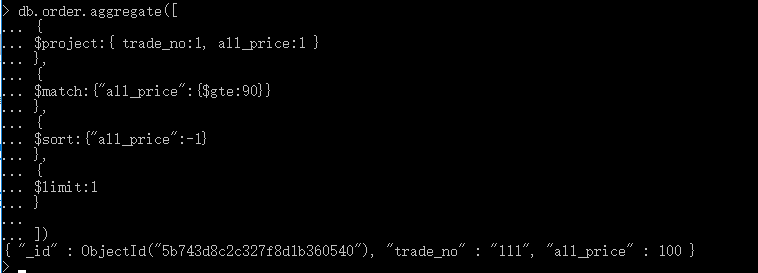
|  |
| --- |
| db.order.aggregate([  {  $project:{ trade\_no:1, all\_price:1 }  },  {  $match:{"all\_price":{$gte:90}}  },  {  $sort:{"all\_price":-1}  }  ]) |





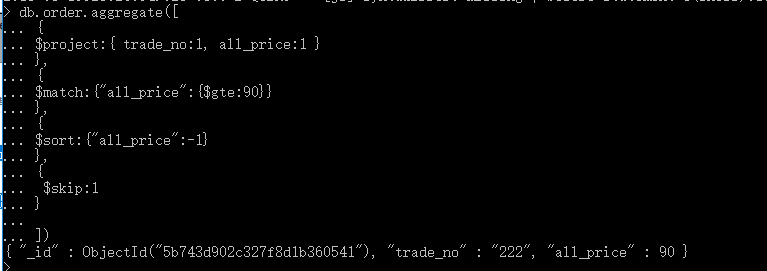
### $limit

|  |
| --- |
| db.order.aggregate([  {  $project:{ trade\_no:1, all\_price:1 }  },  {  $match:{"all\_price":{$gte:90}}  },  {  $sort:{"all\_price":-1}  },  {  $limit:1  }  ]) |



### $skip

|  |
| --- |
| db.order.aggregate([  {  $project:{ trade\_no:1, all\_price:1 }  },  {  $match:{"all\_price":{$gte:90}}  },  {  $sort:{"all\_price":-1}  },  {  $skip:1  }  ]) |



### $lookup 表关联

|  |
| --- |
| db.order.aggregate([  {  $lookup:  {  from: "order\_item",  localField: "order\_id",  foreignField: "order\_id",  as: "items"  }  }  ])  {  "\_id": ObjectId("5b743d8c2c327f8d1b360540"),  "order\_id": "1",  "uid": 10,  "trade\_no": "111",  "all\_price": 100,  "all\_num": 2,  "items": [{  "\_id": ObjectId("5b743d9c2c327f8d1b360543"),  "order\_id": "1",  "title": "商品鼠标1",  "price": 50,  "num": 1  }, {  "\_id": ObjectId("5b743da12c327f8d1b360544"),  "order\_id": "1",  "title": "商品键盘2",  "price": 50,  "num": 1  }, {  "\_id": ObjectId("5b74f457089f78dc8f0a4f3b"),  "order\_id": "1",  "title": "商品键盘3",  "price": 0,  "num": 1  }]  } {  "\_id": ObjectId("5b743d902c327f8d1b360541"),  "order\_id": "2",  "uid": 7,  "trade\_no": "222",  "all\_price": 90,  "all\_num": 2,  "items": [{  "\_id": ObjectId("5b743da52c327f8d1b360545"),  "order\_id": "2",  "title": "牛奶",  "price": 50,  "num": 1  }, {  "\_id": ObjectId("5b743da92c327f8d1b360546"),  "order\_id": "2",  "title": "酸奶",  "price": 40,  "num": 1  }]  } {  "\_id": ObjectId("5b743d962c327f8d1b360542"),  "order\_id": "3",  "uid": 9,  "trade\_no": "333",  "all\_price": 20,  "all\_num": 6,  "items": [{  "\_id": ObjectId("5b743dad2c327f8d1b360547"),  "order\_id": "3",  "title": "矿泉水",  "price": 2,  "num": 5  }, {  "\_id": ObjectId("5b743dff2c327f8d1b360548"),  "order\_id": "3",  "title": "毛巾",  "price": 10,  "num": 1  }]  } |

|  |
| --- |
| db.order.aggregate([  {  $lookup:  {  from: "order\_item",  localField: "order\_id",  foreignField: "order\_id",  as: "items"  }  },  {  $match:{"all\_price":{$gte:90}}  }  ])  {  "\_id": ObjectId("5b743d8c2c327f8d1b360540"),  "order\_id": "1",  "uid": 10,  "trade\_no": "111",  "all\_price": 100,  "all\_num": 2,  "items": [{  "\_id": ObjectId("5b743d9c2c327f8d1b360543"),  "order\_id": "1",  "title": "商品鼠标1",  "price": 50,  "num": 1  }, {  "\_id": ObjectId("5b743da12c327f8d1b360544"),  "order\_id": "1",  "title": "商品键盘2",  "price": 50,  "num": 1  }, {  "\_id": ObjectId("5b74f457089f78dc8f0a4f3b"),  "order\_id": "1",  "title": "商品键盘3",  "price": 0,  "num": 1  }]  } {  "\_id": ObjectId("5b743d902c327f8d1b360541"),  "order\_id": "2",  "uid": 7,  "trade\_no": "222",  "all\_price": 90,  "all\_num": 2,  "items": [{  "\_id": ObjectId("5b743da52c327f8d1b360545"),  "order\_id": "2",  "title": "牛奶",  "price": 50,  "num": 1  }, {  "\_id": ObjectId("5b743da92c327f8d1b360546"),  "order\_id": "2",  "title": "酸奶",  "price": 40,  "num": 1  }]  } |

|  |
| --- |
| db.order.aggregate([  {  $lookup:  {  from: "order\_item",  localField: "order\_id",  foreignField: "order\_id",  as: "items"  }  },  {  $project:{ trade\_no:1, all\_price:1,items:1 }  },  {  $match:{"all\_price":{$gte:90}}  },  {  $sort:{"all\_price":-1}  },  ])  {  "\_id": ObjectId("5b743d8c2c327f8d1b360540"),  "trade\_no": "111",  "all\_price": 100,  "items": [{  "\_id": ObjectId("5b743d9c2c327f8d1b360543"),  "order\_id": "1",  "title": "商品鼠标1",  "price": 50,  "num": 1  }, {  "\_id": ObjectId("5b743da12c327f8d1b360544"),  "order\_id": "1",  "title": "商品键盘2",  "price": 50,  "num": 1  }, {  "\_id": ObjectId("5b74f457089f78dc8f0a4f3b"),  "order\_id": "1",  "title": "商 品键盘3",  "price": 0,  "num": 1  }]  } {  "\_id": ObjectId("5b743d902c327f8d1b360541"),  "trade\_no": "222",  "all\_price": 90,  "items": [{  "\_id": ObjectId("5b743da52c327f8d1b360545"),  "order\_id": "2",  "title": "牛奶",  "price": 50,  "num": 1  }, {  "\_id": ObjectId("5b743da92c327f8d1b360546"),  "order\_id": "2",  "title": "酸奶",  "price": 40,  "num": 1  }]  } |

