



MongoDB Aggregation 业务场景实战

Teambition - 孙祁

简介

一、MongoDB 聚合简介

二、MongoDB 聚合在 Teambition Tools 中的应用

三、MongoDB 聚合场景优化

四、MongoDB 聚合常见技巧

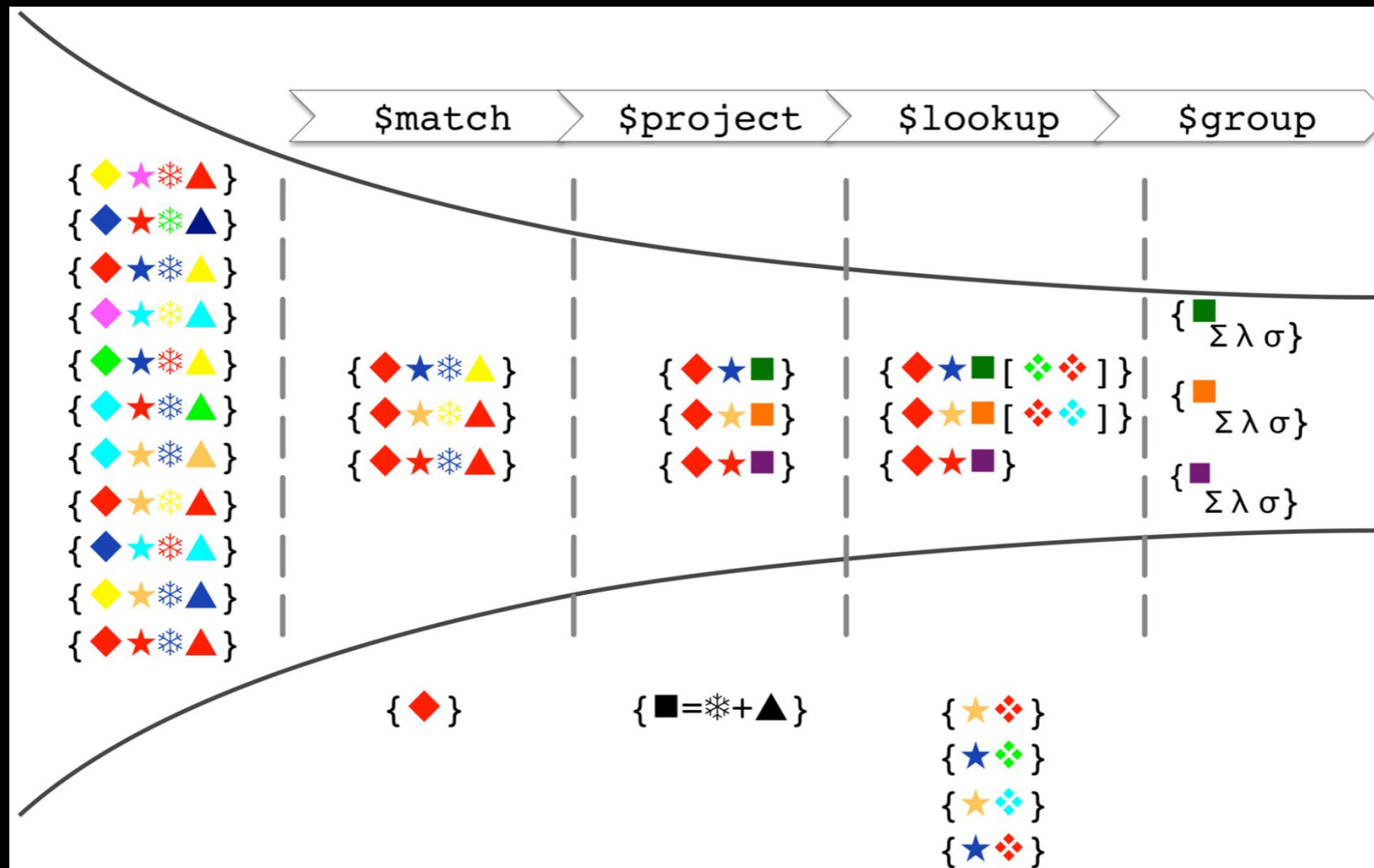
五、MongoDB 聚合常见注意事项

常用分类

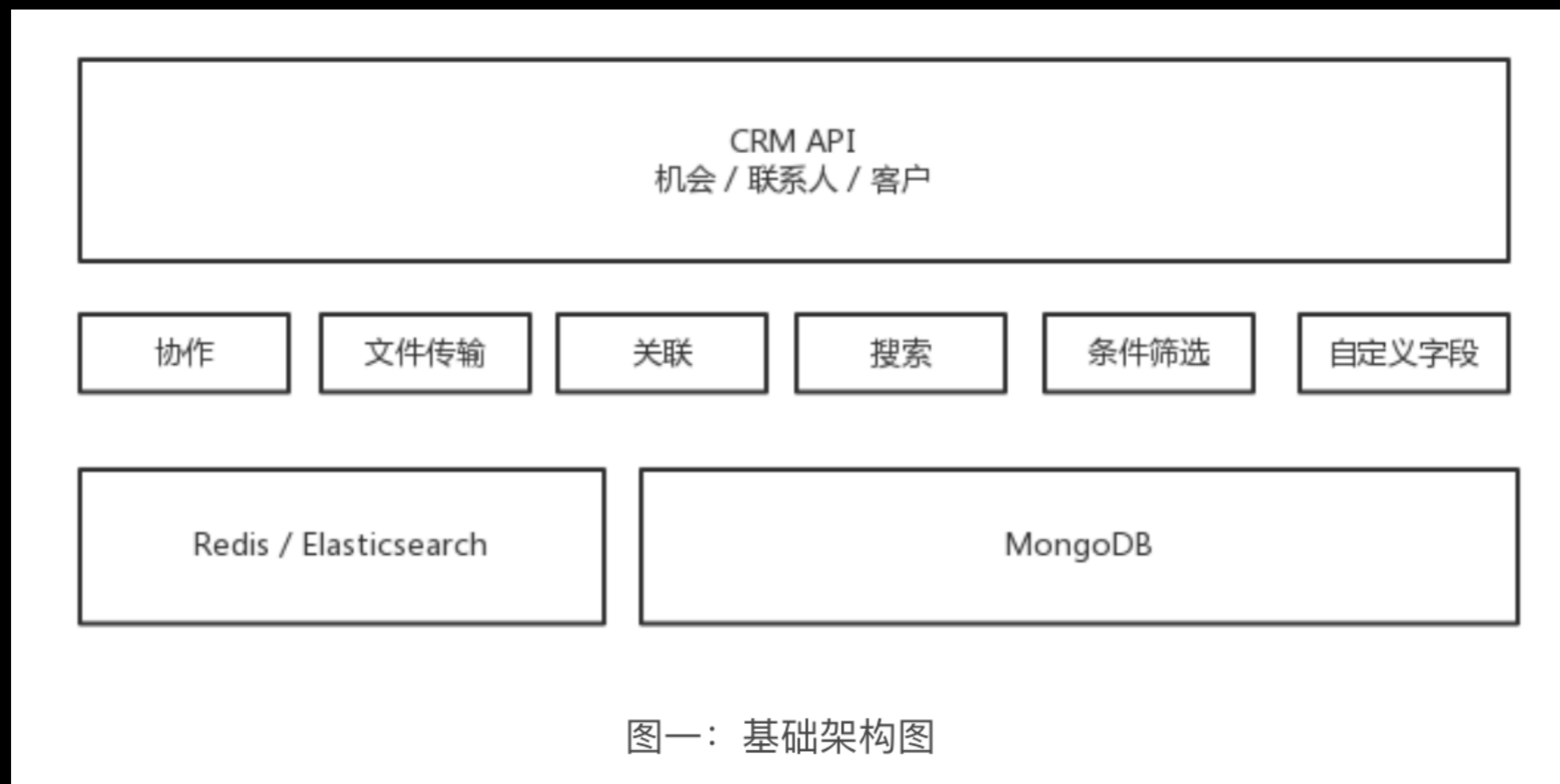
- 聚合管道 (aggregation pipeline)
- map-reduce function
- 单目聚合方法 (group, count, distinct)

定义

每个文档通过一个由多个节点组成的管道，每个节点都有自己的特殊的作用（分组、过滤等），文档经过由多个节点组成的管道后最终得到输出结果。



应用背景



应用场景

- 基础对象查询（客户查询）
- 表 join 查询（机会关联客户信息查询）
- 分类统计（统计机会赢输单信息）
- 嵌套对象排序（关联对象排序）

管道命令

- \$match
- \$project
- \$lookup
- \$group, \$unwind
- \$sort, \$skip, \$limit

管道组合

- `$match` -> `$project`
- `$match` -> `$group`
- `$match` -> `$group` -> `$sort`
- `$match` -> `$match` -> `$project` -> `$group`
- `$match` -> `$lookup` -> `$match` -> `$sort`
- `$match` -> `$project` -> `$sort` -> `$skip` -> `$limit`
- `$match` -> `$project` -> `$unwind` -> `$group` -> `$sort` -> `$skip` -> `$limit`

\$match -> \$project

```
db.deals.aggregate([
  { $match: { createTime: { '$gte': ISODate('2018-05-01') } } },
  { $project: { title: 1, owner: 1, status: 1, updateTime: 1 } }
])
```

\$match -> \$group

```
db.deals.aggregate([
  { $match: { status: 'won', owner: { '$ne': null } } },
  { $group: { _id: '$owner._id', num_deals: { $sum: 1 } } }
])
```

\$match -> \$lookup -> \$match -> \$sort

```
db.deals.aggregate([
  { $match: { 'owner.name': '张三' } },
  { $lookup: {
    from: 'organizations',
    localField: '_organizationId',
    foreignField: '_id',
    as: 'organization'
  } },
  { $match: { 'organization.address': {'$ne': ''} } },
  { $sort: { 'createdTime': -1 } }
])
```

\$match -> \$project -> \$sort -> \$skip ->
\$limit

```
db.deals.aggregate([
  { $match: { 'status': 'won', 'owner.name': '张三' } },
  { $project: { 'name': 1, 'status': 1, 'wonTime': 1, 'note': 1 },
  { $sort: { 'wonTime': -1 },
  { $skip: 0 },
  { $limit: 50 }
])
```

场景优化

- 管道序列优化
- 管道合并优化

场景优化 - 管道序列优化

- 管道序列
- `$sort + $match`
- `$skip + $limit`
- `$project + $skip` or `$project + $limit`

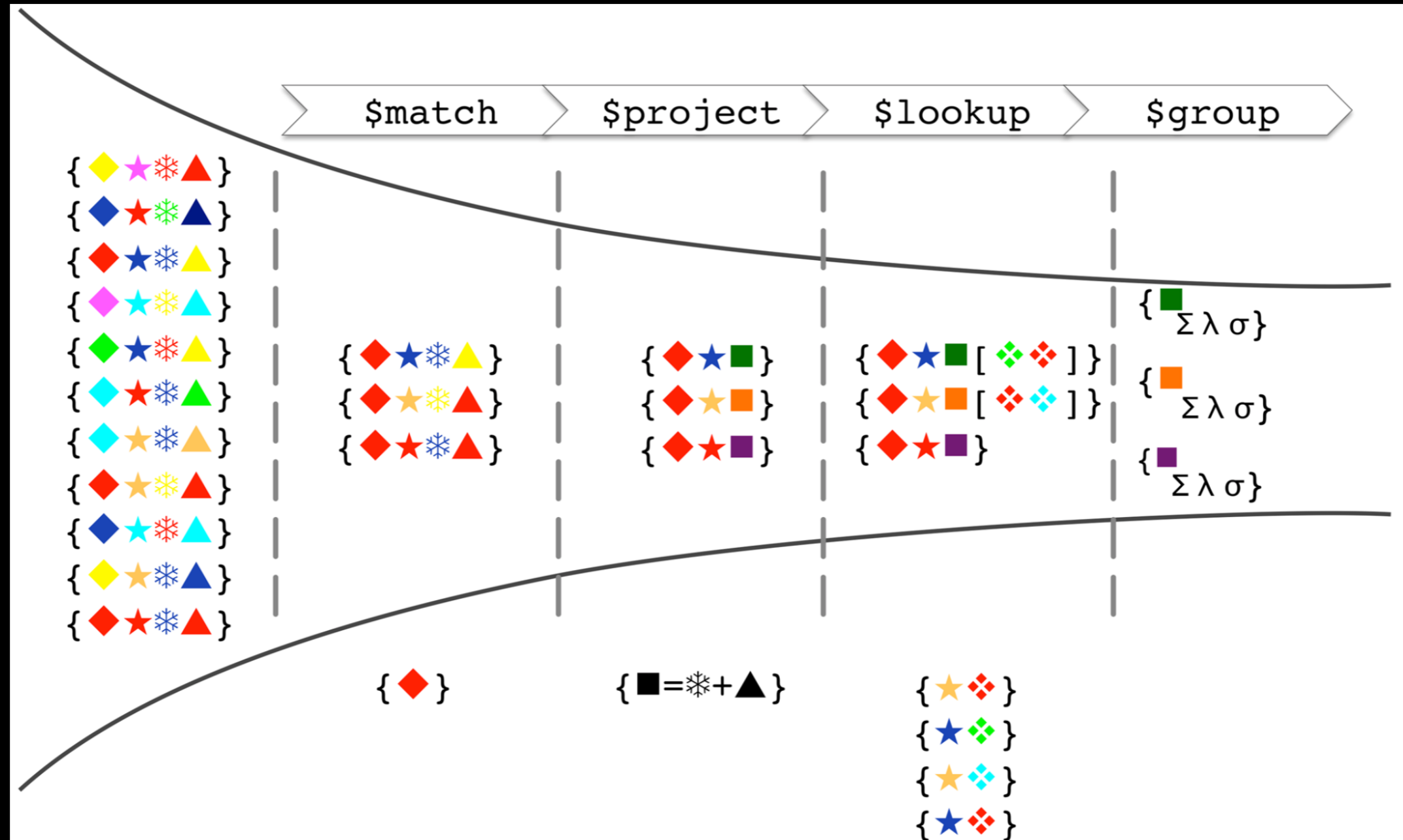
场景优化 - 管道合并优化

- `$sort + $limit`
- `$limit + $limit`
- `$skip + $skip`
- `$match + $match`
- `$lookup + $unwind`

实战技巧

- \$project
- \$ifNull
- \$cond
- ObjectId

实战技巧 - \$project



实战技巧 – \$ifNull

```
{ $ifNull: [ <expression>, <replacement-expression-if-null> ] }
```

- 机会对象查询：按照指定更新时间排序，空值排在最后
- 客户对象查询：按照跟进时间去排序，空值排到最后

实战技巧 – \$ifNull

When comparing values of different **BSON types**, MongoDB uses the following comparison order, from lowest to highest:

1. MinKey (internal type)
2. Null
3. Numbers (ints, longs, doubles, decimals)
4. Symbol, String
5. Object
6. Array
7. BinData
8. ObjectId
9. Boolean
10. Date
11. Timestamp
12. Regular Expression
13. MaxKey (internal type)

实战技巧 – \$cond

```
{ $cond: { if: <boolean-expression>, then: <true-case>, else: <false-case-> } }
```

Or:

```
{ $cond: [ <boolean-expression>, <true-case>, <false-case> ] }
```

实战技巧 – \$cond

```
db.deals.aggregate([
  { $match: { 'isArchive': false } },
  { $project: {
    title: 1,
    updateTime: { $cond: {
      if: {'$eq': ['$updateTime', null]} ,
      then: ISODate('9000-01-01')
      else '$updateTime'
    } }
  } }
  { $sort: { 'updateTime': -1}
})
```

实战技巧 – ObjectId

- aggregate 里比较的 id 格式为 ObjectId 时, mongoose 不会做 ObjectId() 转换

注意事项

- 返回结果大小
- 内存限制
- 聚合操作符使用

小结

今天我主要介绍了 MongoDB 聚合管道的应用实践、技巧、优化以及注意事项，解释了聚合管道操作符的基础概念，希望能够对大家有所帮助。

谢谢大家聆听