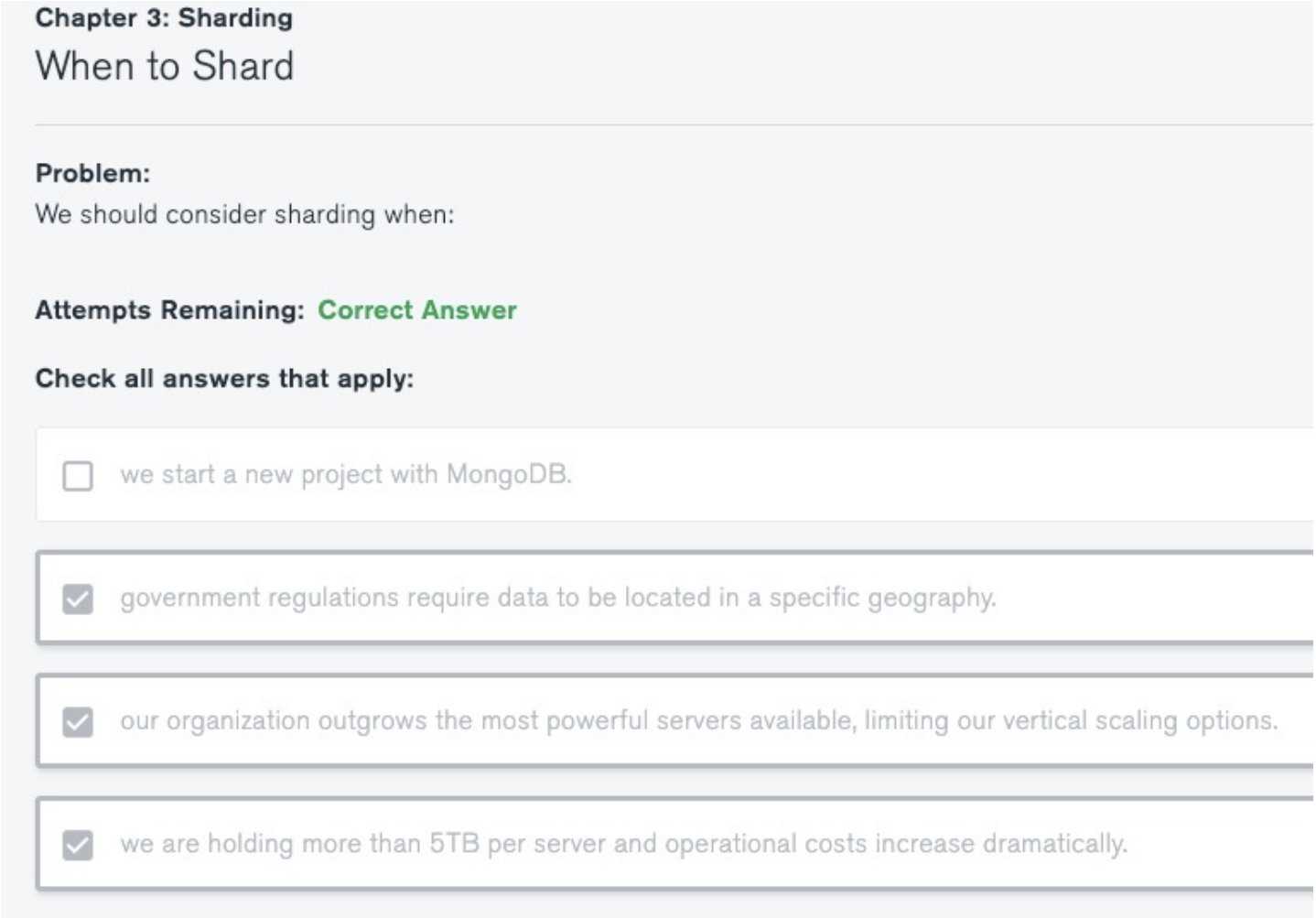
MongoDB DBA 认证知识 Review

1. When to shard



2. 和 RDBMS 的不同

在 functionality 和 scalability 之间的 trade off。

it enables them to **build applications faster**, **handle highly diverse data types**, and **manage applications more efficiently at scale**.

- json 天然适配 OOP 编程范式，避免 orm，业务开发友好
- 文档模型，schema-free 适应业务逻辑发展
- sharding 提供更优秀的横向扩展能力

3. Shard key 选择

- cardinality
- frequency
- rate of change of a potential shard key.

4. JSON **不支持**的类型

4. JSON 不支持的类型

- function
- date
- timestamp
- undefined

支持的类型：

- number
- string
- object
- array
- boolean
- null

5. BSON 中的 ObjectID

- 12 Bytes = 4 Bytes unix timestamp + 5 Bytes random string + 3 Bytes counter
- 秒间可以保证递增，秒内不能

6. db.coll.save()

db.collection.save() will cause a document to be inserted if the _id is not included, or the _id doesn't match any documents.

7. arrayFilter

An array of filter documents that determine which array elements to modify for an update operation on an array field.

8. 关于 upsert 之后包含哪些的 field

The query document also contributes fields to the document that gets created.
但是如果 query document 中的new field 不能被决定，不会被包含仅新插入的 document 中。

9. 关于_id field

- _id 总是第一个 field
- _id 不能是 array
- immutable
- 如果 insert 不提供，client 或者 mongod 会提供一个

10. 关于 projection

- _id 默认显示

- 不能既包含需要显示的 field，又包含不需要显示的 field，_id 除外

11. 关于 array field $all:[xxx,,,]

查找包含 $all 中所有的 element 的 doc，其实等价于 $and
直接等值查找一个 array，既包含 exactly matched doc ，也包含 array field 中 nested array 相等的 doc

```
db.foo.insert({ay:[1, 3]})
db.foo.insert({ay:[[1, 3]]})
db.foo.insert({ay:[[1, 3], [4,5]]})

MongoDB Enterprise m312RS:PRIMARY> db.foo.find({ay:[1, 3]})
{ "_id" : ObjectId("5f1c3011ef4b422d89175cb3"), "ay" : [ 1, 3 ] }
{ "_id" : ObjectId("5f1c3017ef4b422d89175cb4"), "ay" : [ [ 1, 3 ] ] }
{ "_id" : ObjectId("5f1c30d0ef4b422d89175cb6"), "ay" : [ [ 1, 3 ], [ 4, 5 ] ] }
```

12. 关于$type

$type 的值既可以是一个 scalar value，也可以是一个 array

13. 关于$regex

- 不能用在$in query operator中，但是可用JavaScript regular expression objects (**i.e. /pattern/**).
- 如果是对 string 的前缀查询，可以走索引，但是不能利用 case insensitive 索引，
- 关于走索引的性能：Additionally, while /^a/, /^a.*/, and /^a.*$/ match equivalent strings, they have different performance characteristics. All of these expressions use an index if an appropriate index exists; however, **/^a.*/, and /^a.*$/ are slower. /^a/ can stop scanning after matching the prefix**.

14. 关于 count()

- 如果不带 query predicate，count 直接利用 collection metadata，可能不准（unclean shutdown，sharding orphaned docs（可以用 aggregate 解决））
- count 可以走索引，原则由 predicate 决定

15. 关于 distinct()

- 返回结果是一个数组
- If the value of the specified field is an array, db.collection.distinct() **considers each element of the array as a separate value**.

16. 关于 skip()

需要每次重头开始扫描文档，建议使用 range query 来优化

17. 关于 findAndModify()

17. 关于 findAndModify()

- 只能修改单条文档
- 默认返回修改前的文档，可以指定 new option
- 对于分片集合，query 必须是包含 shard key 的等值查询

18. 关于 db.collection.update()

- update 可以直接 replace document，通过在 update 参数中只指定 field:value
- upsert 和 multi option 可以结合使用，先考虑 match 情况下的 multi，再考虑 upsert 一条文档
- upsert 的 query 中如果使用_id，必须是对_id field 的完全 match，{_id.a :1, _id.b:2}是不行的。
- 对于 sharded collection，如果 multi 为 false，query document必须包含_id 或 shard key 的等值查询

19. 关于$unset update operator

- $unset array element 时，会填充一个 null，保持 element position 和 array size 不变

20. 关于 $rename

- 作用于 array element 中的 field 无效
- old name 和 new name 不能相同
- 先对 old name 和 new name 执行一个$unset，再$set new name，所以field 顺序可能会变
- 可以把 field 从 embedded document 中移出/移进

21. $inc 在 field 不存在时，创建 field 并设置值

22. 关于 $mul

- field 不存在时，创建并把 value 置0 （number 类型为 multiplier 的类型）
- 会做类型提升

23. 关于 array update operator $

- 更新第一个 match 的element，用于无法确定 ele position 的情况
- query doc 中必须包含 array field
- 不能和 upsert 一起用

24. 关于$not query operator

- 逻辑操作符，不能直接作用于 field value
- $not: { $gt: 1.99 } 和 { $lte: 1.99 }的区别：前者会返回 field 不存在的 doc

25. 关于 $or operator

25. 关于 array $pull operator

- 作用是对 array 中的每个 element，考察是否 match 指定的 value 或者条件，如果 match，则移除
- 如果指定的是 array value，则必须完全 match
- 如果指定的是 doc value，每个 array ele 会被视为 top level 的 doc 一样去 match

26. 关于 array $pullAll operator

- 和$pull 指定 query 不同，$pullAll 指定一个 value list，doc 和 array value 必须都精确匹配

27. 关于 $addToSet

- 添加不存在的 ele 到 array 中
- 不存在的 array field 会创建

28. 关于 array $push operator

- 默认添加在尾部
- 其他 modifier 需要和 $each 配合使用，modifier 执行顺序：先添加，再排序，再 slice
- $position 是指定本次添加的所有元素的起始位置，可以为负值

29. 关于 case insensitive index

- 创建时指定 collation 或者继承 collection 的collation
- collation.strength 为 1 或者 2 时 是 case insensitive
- query 和 sort 必须使用和 index 相同的collation 才能走索引
- 不是必须走 case insensitive index 才能进行 case insensitive query

30. 关于 sparse index

- 3.2 中支持了 partial index，是 sparse index 的超集，不建议再去使用 sparse index
- geo index 和 text index 默认是 sparse 的
- sparse index 可能导致返回结果不全（强制使用了 hint，默认如果可能导致结果不全，不会走 sparse index）
- sparse 和 unique 联合使用，避免 field 不存在的 doc，报 duplicate key error

31. 关于 multi key index

- 限制：只能包含一个array field，否则会造成 index key 的暴涨（笛卡尔积），后面的 insert 如果违反现有的 multi key index，也会报错
- multi key index bound 的特殊之处：
  - 因为非 $elemMatch 的查询，并不要求 array 中的elem 同时满足多个 predicate，所以最多只能 bound 一个 array field
  - 即使在$elemMatch 查询的情况下，也需要同时 bound 的 array field 的**前缀路径一致**，否则不

能同时 bound，原因同上

The following query, however, uses an `$elemMatch` but not on the required path:

```
db.survey3.find( { ratings: { $elemMatch: { 'scores.q1': 2, 'scores.q2': 8 } } } )
```

As such, MongoDB **cannot** compound the bounds, and the `"ratings.scores.q2"` field will be unconstrained during the index scan. To compound the bounds, the query must use `$elemMatch` on the path `"ratings.scores"`:

```
db.survey3.find( { 'ratings.scores': { $elemMatch: { 'q1': 2, 'q2': 8 } } } )
```

- 同一个 doc 中的 array elem 相同不影响 uniqueness
- 不能走 index sort，不能作为 shard key，不能 cover query（3.6 开始的优化，识别 compound index 是由哪些 field 导致的 multi key），不能做hash index。这些都由 multi key index 的特点决定。

32. 可以使用 explain()的命令

Starting in MongoDB 3.0	Starting in MongoDB 3.2
• aggregate() • count() • find() • remove() • update()	• distinct() • findAndModify()

33. plan cache 淘汰的几种情况

- LRU
- 重启
- 索引重建
- 索引创建和删除
- query 执行的时间超过了winning plan的 10 倍

34. 关于 unique index

- 对于 compound multikey index，同一个 doc 中的 array element 存在相同是允许的，不同的 doc 之间校验所有的 array element key 是否有重复
- 已经有重复 value 的 field 上建立unique索引会失败
- hash 索引不能指定 unique（hash 碰撞决定）
- 对 range sharded collection 来说，除了_id 索引，唯一索引key 必须把 shard key 作前缀（避免不全局唯一）

35. 关于 ttl index

- ttl index 只能 single field，可以在 array field 上建立，但是 array field 需要包含date value，删除以最老的 date 为准
- 不能是_id field，不能在 capped collection 上建立
- 只能 primary 删除，后台线程 60 秒唤醒一次，可能有删除延迟（应该没做访问时删除优化）
- 不能把已有的 non ttl index 直接 modify 为 ttl index

36. wildcard index use case

- unpredictable query shape
- implement attribute pattern

37. wildcard index 特性

- 对 embedded doc 或者 doc in array field 递归的创建索引，而不是对 doc 或 array field 本身
- 但是，对于 array 中的 array，不再继续递归，直接对整个 embedded array 建立索引
- 只创建 single field index，且是 sparse index
- path-specific wildcard index syntax 和 wildcardProjection option 不兼容
- covered query 要求：exclude _id; single field predicate（not array）; single field projection
- 不能参与 index intersection；可以作为 prefix支持 multi predicate query

38. 关于 $regex query operator 使用索引

- not collation-aware，所以对于 case insensitive regular expression，并不能利用case insensitive index

39. 关于 text index

- text index 天然是 sparse 的
- 一个集合只能有一个 text index
- text index 类似于 multi key index，但是允许 compound index 包含多个 text index key，preceding index key 需要是等值查询才能走compound text index
- 基于score 排序，使用 $meta projection operator

```
db.articles.find(
   { $text: { $search: "coffee" } },
   { score: { $meta: "textScore" } }
).sort( { score: { $meta: "textScore" } } ).limit(2)
```

40. 关于 geoSpacial index

- 不支持 covered query
- Geo JSON Object

- GeoJSON Object
  - { type: "Point", coordinates: [40, 5] }
  - { type: "LineString", coordinates: [[40, 5], [41, 6]] }
  - { type: "Polygon", coordinates: [[[0 , 0] , [3 , 6] , [6 , 1] , [0 , 0]]]}
- query：
  - within a polygon（using new style）： **$geoWithin: { $geometry:** { type : "Polygon" , coordinates: [[[0, 0], [3, 6], [6, 1], [0, 0]]] }}
  - within a circle（only 2d index support）：{ **$geoWithin: { $center**: [[-74, 40.74], **10**] } } }
  - near a point （new style）：{ **$near : { $geometry**: { type: "Point", coordinates: [-73.9667, 40.78] }, $minDistance: 1000, $maxDistance: 5000 }}

41. 关于 different type of members

- **delayed secondary**，hidden：true，voting：1，priority：0
- **non-voting members**，voting：0， priority：0

42. 关于 rollback data

- createRollbackDataFiles控制是否把 rollback 的数据写到 bson 文件中，默认为 true
- db.adminCommand({ setParameter: 1, createRollbackDataFiles: false })

44. chunk split 的条件

- 超过 max chunk size（default 64MB）
- 超过 Maximum Number of Documents Per Chunk to Migrate（MongoDB cannot move a chunk if the number of documents in the chunk is greater than **1.3 times the result of dividing the configured chunk size by the average document size**）

45. 关于 pre-split chunk

- hash sharding 可以在 shard empty collection 时指定 chunk 数，默认每个 shard 两个，range sharding 不可以
- pre-split chunk 的目的是为了解决在一些特殊情况下，比如灌数据，MongoDB 本身的 sharding 机制会导致 write load 不平衡
- pre-split chunk 只建议在 empty collection 上做，split 提供 **splitFind** 和 **splitAt** 两个命令
- chunk merge：db.adminCommand({mergeChunks: "test.members",bounds: [{ "username" : "user69816" },{ "username" : "user96401" }]})

46. query in sharding

- shard key 可以是某个 compound index 的前缀，满足 index 需求
- 如果是 compound shard key ， prefix in query 也可以产生 targeted query
- aggregation 是 broadcast 还是 target，取决于具体的 pipeline，aggregate 的结果合并，可能在 primary shard 上，也可能在 mongos 上

primary shard 上，也可能在mongos 上。

47. primary shard

- 选择逻辑：根据 listDatabases 命令中的totalSize 判断数据量最少的 shard 作为new db 的 primary shard
- 如果aggregate pipeline以$match 开头，且包含 shard key，那么只会在单一 shard 上执行
- 否则，The **$out stage and the $lookup stage** require running on the database's primary shard. 其他情况，会随机选择一个 shard。

48. 关于 journaling

- 从 4.0 开始，副本集启动不支持关闭journaling
- 写操作引发的索引更新在同一条 log record 中
- journal 刷盘的因素：
  - 有操作在等待oplog，比如复制扫 oplog，因果一致读等
  - write concern majority，默认 j 为 true
  - At every 100 milliseconds (See storage.journal.commitIntervalMs).
  - 超过单个 journal 的 100MB 大小，创建新的前，刷老的

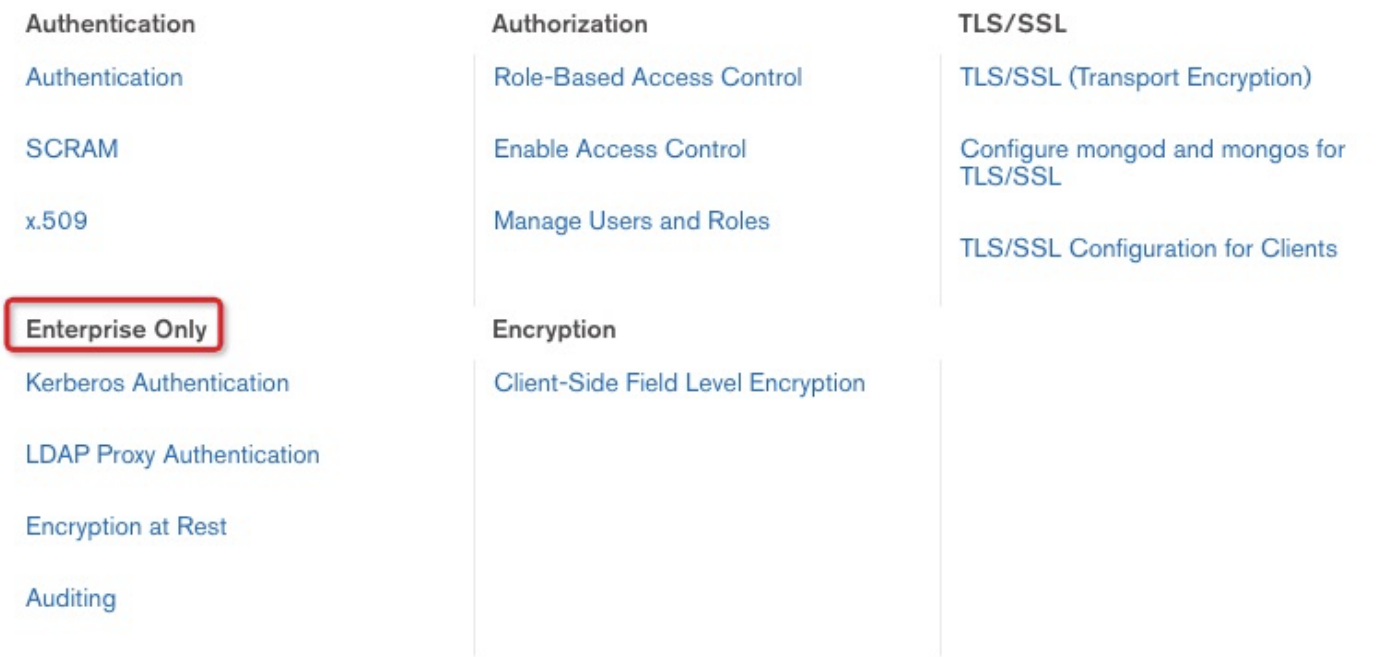
49. 关于server log rotate

- rotate的方法：kill -SIGUSR1；logRotate 命令；基于 syslog
- --logRotate rename （default）：old 的重命名，当前的名字不变
- --logRotate reopen：依靠外部对log 重命名，然后 reopen新的

50. 关于 Security

- replset 内部成员之间的认证：x509，keyfile
- x509，ldap，kerberos都属于外部认证机制，需要创建在'$external' db 上的 user
- security checklist：
  - 启用认证和授权
  - 为不同用户创建合适的账号
  - TLS 链路加密
  - 磁盘数据加密
  - bind ip /vpc/vpn
  - audit
  - mongodb 进程不要使用 root 账号启动
  - 关闭 server side js 支持
  - 考虑安全规章：zone sharding，HIPPA，PCI-DSS

MongoDB provides various features, such as authentication, access control, encryption, to secure your MongoDB deployments. Some key security features include:



51. backup

- filesystem snapshot backup 需要开启 journaling，为了获取一个 point in time 的 snapshot；
- fsyncLock 可以避免RAID 情况下，多块盘不提供全局一致性 snapshot 的问题
- sharding 备份，先关闭 balancer
- 对于 4.2+的 sharded cluster，如果有分布式事务在执行，mongodump 不保证分布式事务的原子性

52. 关于 mongodb tools

- mongoreplay 用于捕获源端流量，包含 Request，reply，client metadata
  - 对源端无需实例权限，需要系统 root 权限，类似于 tcpdump
  - record，生成 bson 格式的流量记录文件
  - play，回放 bson 格式的流量记录文件，对目的端实例需要相应权限，可以生成报告
  - monitor，根据 record 文件或实时输出报告，包括请求的 latency
- bsondump：把 bson 文件（比如 mongodump 生成的）转化为 json 文件
- mongofiles：
  - 用于操作GridFS 中文件的工具
  - GridFS 用于存储大于 16MB 的文件，文件会被切分为多个 chunk 存储在 collection 中；对于音视频文件，经常需要跳跃读取的情况，也适合用 GridFS
  - 支持的功能：list、search、get；put、delete

53. storage engine

- in-memory storage engine 在企业版才支持
  - 基本上是定位一个 cache 场景：Other than some metadata and diagnostic data, the in-memory storage engine does not maintain any on-disk data, including configuration data, indexes, user credentials, etc.
  - 文档级锁

- wt
  - 压缩：
    - block & journal：snappy（default），zlib，zstd
    - index：prefix

54. rs.serverStatus()['repl'] 输出

类似于 isMaster，除了包含 rbid

```
db.serverStatus().repl
{
        "hosts" : [
                "e18c04345.et15sqa:9791",
                "100.81.164.181:9792",
                "100.81.164.181:9793"
        ],
        "setName" : "xdjmgset-19702683",
        "setVersion" : 5,
        "ismaster" : false,
        "secondary" : true,
        "primary" : "e18c04345.et15sqa:9791",
        "me" : "100.81.164.181:9792",
        "lastWrite" : {
                "opTime" : {
                        "ts" : Timestamp(1596027558, 1),
                        "t" : NumberLong(14)
                },
                "lastWriteDate" : ISODate("2020-07-29T12:59:18Z"),
                "majorityOpTime" : {
                        "ts" : Timestamp(1596027558, 1),
                        "t" : NumberLong(14)
                },
                "majorityWriteDate" : ISODate("2020-07-29T12:59:18Z")
        },
        "rbid" : 1
}
```

55. 相同 field，不同 value type 的排序规则

- 整体上是先 value type，然后value 值本身比较
- 特殊情况：
  - 对于 array：如果是递增排序，array field 选用最小的元素比较，反之亦反；空 array 比 null 小
  - non-exist field 比 null 大，小于 numerical field
- index 排序规则是：先 type，后 value，这也导致了为什么说 multi key index 不能走 index sort（sort 规则和 index 排序规则不一样）

```
db.sorttest.find({},{_id:0, a:1}).sort({a:1})
{ "a" : [ ] }
{ "a" : null }
{ }
{ "a" : [ 1 ] }
```

```
{ "a" : 2 }
{ "a" : {  } }
{ "a" : { "b" : 1 } }
```

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)

56. 关于 audit log

- audit log 只记录成功完成的操作，比如删除一条不存在的文档，不存在的 collection，是不会记录的。
- DML 的 filter 配置方式：