1. mongodb是什么？

是一个非关系型数据库，以bson格式存储数据，类似于json格式。

1. mongodb有哪些特点？
2. 你说的NoSQL数据库是什么意思?NoSQL与RDBMS直接有什么区别?为什么要使用和不使用NoSQL数据库?说一说NoSQL数据库的几个优点?

RDBMS：有严格的关系数据模型，R(A1，A2，…，An) 其中R为关系模式名，A表示属性名，通常以二维表的形式表示，

NoSql：没有严格的关系数据模型，一般是以json等键值格式存储数据，无需严格的数据存储结构

NoSql优点：

* 无需严格的数据存储结构，数据结构好扩展。
* 查询速度快，由于是存储结构好扩展，一边一条记录就能存储所需的所有信息，所以查询起来会很快，没有关系型数据库那样需要多个表连接。(mongodb有时候也要连接)

一般非关系型数据库是基于CAP模型(用在分布式)，而传统的关系型数据库是基于ACID模型的，在分布式中很难保持ACID特性。

1. 数据存储结构： 首先关系型数据库一般都有固定的表结构，并且需要通过DDL语句来修改表结构，不是很容易进行扩展，而非关系型数据库的存储机制就有很多了，比如基于文档的，K-V键值对的，还有基于图的等，对于数据的格式十分灵活没有固定的表结构，方便扩展，因此如果业务的数据结构并不是固定的或者经常变动比较大的，那么非关系型数据库是个好的选择

2. 可扩展性 传统的关系型数据库给人一种横向扩展难，不好对数据进行分片等，而一些非关系型数据库则原生就支持数据的水平扩展(比如mongodb的sharding机制)，并且这可能也是很多NoSQL的一大卖点，其实象Mysql这种关系型数据库的水平扩展也并不是难，即使NoSQL水平扩展容易但对于向跨分片进行joins这种场景都没有什么太好的解决办法，不管是关系型还是非关系型数据库，解决水平扩展或者跨分片Joins这种场景，在应用层和数据库层中间加一层中间件来做数据处理也许是个好的办法

3. 数据一致性 非关系型数据库一般强调的是数据最终一致性，而不没有像ACID一样强调数据的强一致性，从非关系型数据库中读到的有可能还是处于一个中间态的数据，因此如果你的业务对于数据的一致性要求很高，那么非关系型数据库并不一个很好的选择，非关系型数据库可能更多的偏向于OLAP场景，而关系型数据库更多偏向于OLTP场景

1. NoSQL数据库有哪些类型?

* 键值(key-value)存储数据库，redis
* 列存储数据库，HBase
* 文档型数据库，MongoDB
* 图形数据库，InfoGrid

1. MongoDB成为最好NoSQL数据库的原因是什么?

* 面向文件的，
* 高性能，以降低强一致性为代价，但是比其他nosql要慢
* 高可用性，集群分片
* 易扩展性，没有固定的数据存储结构
* 丰富的查询语言

1. 分析器在MongoDB中的作用是什么?

MongoDB中包括了一个可以显示数据库中每个操作性能特点的数据库分析器。通过这个分析器你可以找到比预期慢的查询(或写操作);利用这一信息，比如，可以确定是否需要添加索引。

1. journal回放在条目(entry)不完整时(比如恰巧有一个中途故障了)会遇到问题吗?

每个journal (group)的写操作都是一致的，除非它是完整的否则在恢复过程中它不会回放。

1. 名字空间(namespace)是什么?

MongoDB存储BSON对象在丛集(collection)中。数据库名字和丛集名字以句点连结起来叫做名字空间(namespace)。

1. 如果用户移除对象的属性，该属性是否从存储层中删除?

是的，用户移除属性然后对象会重新保存(re-save())。

1. 能否使用日志特征进行安全备份?

是的。

1. MongoDB中有几种日志？

* 系统日志
* Journal日志

journaling(日记) 日志功能则是 MongoDB 里面非常重要的一个功能 ， 它保证了数据库服务器在意外断电 、 自然灾害等情况下数据的完整性。它通过预写式的redo日志为MongoDB增加了额外的可靠性保障。开启该功能时,MongoDB会在进行写入时建立一条Journal日志,其中包含了此次写入操作具体更改的磁盘地址和字节。因此一旦服务器突然停机，可在启动时对日记进行重放，从而重新执行那些停机前没能够刷新到磁盘的写入操作。

MongoDB会批量地提交更改，即每次写入不会立即刷新到磁盘

* 固定集合(Capped Collection)
* oplog主从日志

Replica Sets复制集用于在多台服务器之间备份数据。MongoDB的复制功能是使用操作日志oplog实现的，操作日志包含了主节点的每一次写操作。oplog是主节点的local数据库中的一个固定集合。备份节点通过查询这个集合就可以知道需要进行复制的操作。

* 慢查询日志

MongoDB中使用系统分析器(system profiler)来查找耗时过长的操作

1. 允许空值null吗?

允许的，空值时没有具体的数据类型。

1. MongoDB中有几种数据类型

Array、Binary、Boolean、Code、Date、Decimal128、Double、Int32、Int64、MaxKey、MinKey、Null、Object、ObjectId、BsonRegexp、String、Symbol、Timestamp、Undefined

1. MongoDb查询

db.demo35.find({"Name": {$ne: null}});

$ne:不等于

$eq:等于

$and，$or 条件判断

db.example.find( {

$and: [

{ x: { $ne: 0 } },

{ $expr: { $eq: [ { $divide: [ 1, "$x" ] }, 3 ] } }

]

} )

1. 更新操作立刻fsync到磁盘?

不会，磁盘写操作默认是延迟执行的。写操作可能在两三秒(默认在60秒内)后到达磁盘，如果mongodb崩了，容易数据丢失，所以最好要开启Journal日志，确保数据写入日志才算写入完成。

1. 如何执行事务/加锁?

MongoDB没有使用传统的锁或者复杂的带回滚的事务，因为它设计的宗旨是轻量，快速以及可预计的高性能。可以把它类比成MySQLMylSAM的自动提交模式。通过精简对事务的支持，性能得到了提升，特别是在一个可能会穿过多个服务器的系统里。

更正：mongodb 4.2及之后的版本是可以使用事务的，前提是开启数据集集群。

1. 为什么我的数据文件如此庞大?

* MongoDB会积极的预分配预留空间来防止文件系统碎片
* MongoDB删除数据时候是不会回收空间的，新的数据依旧是放在后续空间中。

1. 如何回收MongoDB的空间

* db.collection.drop()

删除集合的物理文件，空间立即被回收

* Compact，

如果只是删除文档，db.collection.remove({}, {multi: true}) ，空间也是不会回收的，需要手动Compact来进行空间回收，compact 一个集合，会加集合所在DB的互斥写锁，会导致该DB上所有的读写请求都阻塞；因为 compact 执行的时间可能很长，跟集合的数据量相关，所以强烈建议在业务低峰期执行，避免影响业务。Compact 动作最终由存储引擎 WiredTiger 完成，WiredTiger 在执行 compact 时，会不断将集合文件后面的数据往前面空闲的空间写，然后逐步 truancate 文件回收物理空间。每一轮 compact 前，WT 都会先检查是否符合 comapact 条件。

* repairDatabase

是官方文档中认为唯一可以回收硬盘空间的方法。

repairDatabase is the appropriate and the only way to reclaim disk space.

虽然可以用 db.repairDatabase()修复数据。但这种方法有两个不好的地方。1.在生产上操作如果意外停止可能会造成数据无法恢复的危险。2.如果磁盘空间不足，小于现在这个db时间占有的空间，这种情况是用不了 db.repairDatabase()的。

注意，repairDatabase操作所需要磁盘的空余空间为当前数据总量再加上2G。如果当前磁盘分区空间不足，可以尝试用 –repairpath 参数指定一个空间足够的分区路径。

* 副本集

通过副本集的方式来释放内存，把从节点设置为主节点，清空/释放主节点的内存空间(可以删除原有副本再重新创建一个，也可以在副本进行前面compact操作)，然后再重新设置为主节点。

1. MongoDB存储引擎

WiredTiger

1. mongodb写入操作流程：

wiredTiger写操作先到cache，并持久化WAL

每60s或log文件达到2G，则执行一次Checkpoint持久化，产生一个新快照。

wiredTiger连接初始化时，先将数据恢复到最新的快照，再根据WAL恢复数据。

1. MongoDB 事务

* Write Concern

w:0 设置为 0 无需关注写入成功与否。

w:1 ～ 任意节点数 自定义节点数设置，复制集中不得大于最大节点数。默认情况下为 1，表示写入到 Primary 节点就开始往客户端发送确认写入成功。

w:"majority" 大多数节点成功原则，例如一个复制集 3 个节点，2 个节点成功就认为本次写入成功。

w:"all" 所以节点都成功，才认为写入成功。

j:true 默认情况 j:false，写操作到达内存算作完成。如果设置为 j:true，写操作只有到达 journal 文件才算成功。

wtimeout: 写入超时实践

设置示例：

writeConcern: {

w:"majority" // 大多数原则

j:true,

wtimeout: 5000,

}

* Read Preference

在一个事务操作中使用事务级别的 readPreference 来决定读取时从哪个节点读取。可方便的实现读写分离、就近读取策略。

primary 只从主节点读取，默认值。

primaryPreferred 优先选择主节点，不可用时选择从节点

secondary 只在从节点读取

secondaryPreferred 优先在从节点读取，从节点不可用时选择主节点。

nearest 选择附近节点

* Read Concern

MongoDB 3.2 引入了 readConcern 来决定读取的策略，但是与 readPreference 不同，readPreference 决定从哪个节点读取，readConcern 决定该节点的哪些数据是可读的。主要保证事务中的隔离性，避免脏读。

available：读取所有可用的数据。

local：仅读取当前分片的数据。

majority：读取在大多数节点上提交完成的数据。

snapshot：读取最近快照中的数据。

1. 启用备份故障恢复需要多久?

从备份数据库声明主数据库宕机到选出一个备份数据库作为新的主数据库将花费10到30秒时间。这期间在主数据库上的操作将会失败--包括

写入和强一致性读取(strong consistent read)操作。然而，你还能在第二数据库上执行最终一致性查询(eventually consistent query)(在slaveOk模式下)，即使在这段时间里。

1. 什么是master或primary?

它是当前备份集群(replica set)中负责处理所有写入操作的主要节点/成员。在一个备份集群中，当失效备援(failover)事件发生时，一个另外的成员会变成primary。

1. 什么是secondary或slave?

Seconday从当前的primary上复制相应的操作。它是通过跟踪复制oplog(local.oplog.rs) 做到的。

1. 我必须调用getLastError来确保写操作生效了么?

不用。不管你有没有调用getLastError(又叫"Safe Mode")服务器做的操作都一样。调用getLastError只是为了确认写操作成功提交了。当然，你经常想得到确认，但是写操作的安全性和是否生效不是由这个决定的。看业务需求吧。

1. 我应该启动一个集群分片(sharded)还是一个非集群分片的 MongoDB 环境?

建议非集群分片方式，除非一个服务器在性能上已经不足以支撑时，才需要分片。

1. 分片(sharding)和复制(replication)是怎样工作的?

分片是根据片键把数据存储到不同的服务器节点中，这样是为了提高数据吞吐量，提高读写性能。复制是把数据写入到主节点中，然后使用openlog复制到从节点中，复制是为了备份数据，解决节点宕机问题，能直接切换节点。

1. 如果在一个分片(shard)停止或者很慢的时候，我发起一个查询会怎样?

如果一个分片(shard)停止了，除非查询设置了“Partial”选项，否则查询会返回一个错误。如果一个分片(shard)响应很慢，MongoDB则会等待它的响应。

1. 我可以把moveChunk目录里的旧文件删除吗?

没问题，这些文件是在分片(shard)进行均衡操作(balancing)的时候产生的临时文件。一旦这些操作已经完成，相关的临时文件也应该被删除掉。但目前清理工作是需要手动的，所以请小心地考虑再释放这些文件的空间。

1. 如果块移动操作(moveChunk)失败了，我需要手动清除部分转移的文档吗?

不需要，移动操作是一致(consistent)并且是确定性的(deterministic);一次失败后，移动操作会不断重试;当完成后，数据只会出现在新的分片里(shard)。

1. 如果我在使用复制技术(replication)，可以一部分使用日志(journaling)而其他部分则不使用吗?

可以。

1. 当更新一个正在被迁移的块（Chunk）上的文档时会发生什么？

更新操作会立即发生在旧的块（Chunk）上，然后更改才会在所有权转移前复制到新的分片上。

1. MongoDB在A:{B,C}上建立索引，查询A:{B,C}和A:{C,B}都会使用索引吗？

不会，只会在A:{B,C}上使用索引。

1. 如何理解MongoDB中的GridFS机制，MongoDB为何使用GridFS来存储文件？

GridFS是为了解决bson对象的限制问题，MongoDB在collection中是可以存储binary对象的，但是在大小上有限制，所以不能存储过大文件。GridFS就是一个用来解决该问题的规范。GridFS使用两个集合来存储大文件，分别是fs.files和fs.chunks集合，fs.files存储的是文件的基本信息，包括文件名、md5、上传时间、chunk的大小(默认是261120/255kb)、文件大小和文件id(自动生成)等信息，而fs.chunks存储的就是文件的binary数据，根据chunks大小切分为好几条数据存储到chunks的collection中。Chunks中数据有一个files\_id，就是fs.files中的文件id，然后还有一个n字段进行排序，最后一个data字段就是binary数据，在获取文件的时候可以直接根据文件id，到chunks中查找文件数据块，根据n排序文件数据块，然后把data组合起来就是一个文件了。

1. 数据库的整体结构

Database->collection->document

1. MongoDB是由哪种语言写的

MongoDB用c++编写的,流行的开源数据库MySQL也是用C++开发的。C++1983年发行是一种使用广泛的计算机程序设计语言。它是一种痛用程序设计语言，支持 多种编程模式。

1. 什么是集合

集合就是一组 MongoDB 文档

1. 什么是文档

一条bson格式数据算是一个文档

1. 什么是”mongod“

mongod是处理MongoDB系统的主要进程。它处理数据请求，管理数据存储，和执行后台管理操作。当我们运行mongod命令意味着正在启动MongoDB进程,并且在后台运行。

1. 什么是"mongo"

它是一个命令行工具用于连接一个特定的mongod实例。当我们没有带参数运行mongo命令它将使用默认的端口号和localhost连接

1. MongoDB哪个命令可以切换数据库

MongoDB 用 use +数据库名称的方式来创建数据库。 use 会创建一个新的数据库，如果该数据库存在，则返回这个数据库。

1. 在哪些场景使用MongoDB

大数据、实时性要求不高的时候，因为mongodb是没有数据强一致性的，所以相对来说读写会比较快，但是写入不是马上就能够读取得到，所以对实时性比较高的会不太适合。

1. MongoDB中的命名空间是什么意思?
2. 在MongoDB集合操作

Show Collections

db.createCollection(name, options)

db.xxx.drop()

1. 什么是聚合

把数据进行组合处理之后返回结果。