MongoDB — документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных. Применяется в веб-разработке, в частности, в рамках JavaScript-ориентированного стека MEAN.

**Возможности**

Система **поддерживает ad-hoc-запросы**: они могут возвращать конкретные поля документов и пользовательские JavaScript-функции. Поддерживается поиск по регулярным выражениям. Также можно настроить запрос на возвращение случайного набора результатов.

Имеется поддержка индексов.

Система **может работать с набором реплик**, то есть содержать две или более копии данных на различных узлах. Каждый экземпляр набора реплик может в любой момент выступать в роли основной или вспомогательной реплики. Все операции записи и чтения по умолчанию осуществляются с основной репликой. Вспомогательные реплики поддерживают в актуальном состоянии копии данных. В случае, когда основная реплика дает сбой, набор реплик проводит выбор, которая из реплик должна стать основной. Второстепенные реплики могут дополнительно являться источником для операций чтения.

Система **масштабируется горизонтально**, используя технику сегментирования (англ. sharding) объектов баз данных — распределение их частей по различным узлам кластера. Администратор выбирает ключ сегментирования, который определяет, по какому критерию данные будут разнесены по узлам (в зависимости от значений хэша ключа сегментирования). Благодаря тому, что каждый узел кластера может принимать запросы, обеспечивается балансировка нагрузки.

Есть официальные драйверы для основных языков программирования (Си, C++, C#, Go, Java, Node.js, Perl, PHP, Python, Ruby, Rust, Scala, Swift). Существует также большое количество неофициальных или поддерживаемых сообществом драйверов для других языков программирования и фреймворков.

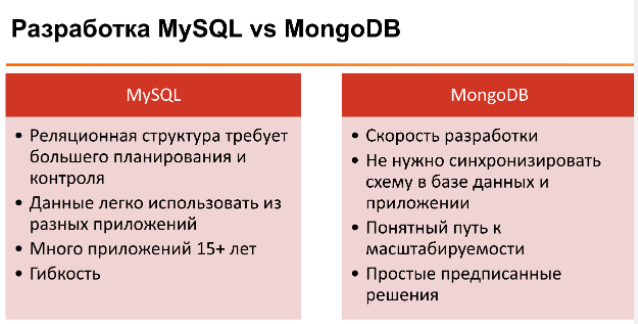
Основным интерфейсом к базе данных была командная оболочка «mongo». С версии MongoDB 3.2 в качестве графической оболочки поставляется «MongoDB Compass». Существуют продукты и сторонние проекты, которые предлагают инструменты с GUI для администрирования и просмотра данных.

**Преимущества:**

Здесь преимущество то, что у нас гибкий JSON-формат документов. Для некоторых задач и каким-то разработчикам это удобнее, чем мучиться с добавлением колонок в SQL-базах данных. Не нужно учить SQL — для некоторых это сложно. Простые запросы реже создают проблемы. Если посмотреть на проблемы производительности, в основном они возникают, когда люди пишут сложные запросы с JOIN в кучу таблиц и GROUP BY. Если такой функциональности в системе нет, то создать сложный запрос получается сложнее.

В MongoDB встроена достаточно простая масштабируемость с использованием технологии шардинга. Сложные запросы если возникают, мы их обычно решаем на стороне приложения. То есть, если нам нужно сделать что-то вроде JOIN, мы можем сходить выбрать данные, потом сходить выбрать данные по ссылкам и затем их обработать на стороне приложения. Для людей, которые знают язык SQL, это выглядит как-то убого и ненатурально. Но на самом деле для многих разработка application-серверов такое куда проще, чем разбираться с JOIN.

Если говорить про распределение преимуществ и недостатков MySQL и MongoDB с точки зрения цикла разработки приложения, то их можно представить так:



У MongoDB структура данных основана на документах. Данные многих веб-приложений отображать очень просто. Потому что если храним структуру — что-то вроде ассоциированного массива приложения, то это очень просто и понятно для разработчика сериализуется в JSON-документ. Раскладывать такое в реляционной базе данных по разным табличкам — задача более нетривиальная.

Результаты как список документов, у которых может быть совершенно разная структура — более гибкое решение.

MongoDB не поддерживает транзакции, но он поддерживает атомарные операции над документом. Это значит, что с точки зрения одного документа операция у нас будет атомарна. Если у нас операция изменяет несколько документов, и во время этой операции произойдет какой-то сбой внутри, то какие-то из этих документов могут быть изменены, а какие-то — не изменены.

**Администрирование**

В MongoDB всё больше ориентированно на то, что оно работает каким-то одним стандартным образом — есть минимизация администрирования. Но понятно, что это происходит при потере гибкости. Коммьюнити open source решений для MongoDB значительно меньше. Многие вещи в MongoDB с точки зрения рекомендаций достаточно жестко привязаны к Ops Manager — коммерческой разработке MongoDB.