План:

1. Термины

2. История

3. Определение

4. Пример документа

5. Достоинства

6. Недостатки

7. Область применения

На сегодняшней паре я расскажу вам про такую БД, как MongoDB. Для удобства усвоения всей информации, что я вам расскажу, доклад разбит на пункты, которые представлены в плане на слайде.

Первоначально, будет рассмотрена история MongoDB. Тут мы узнаем, что способствовало появлению данной БД.

Далее поймем, что подразумевается под данной БД, дав определение. Кроме того, разберемся в некоторых терминах, используемых в MongoDB.

После этого рассмотрим пример, демонстрирующий использование БД.

Далее выявим достоинства и недостатки БД. И в конце рассмотрим области применения данной БД.

1. Термины



2. История

В 1995 году была создана рекламная компания DoubleClick. Вскоре DoubleClick взлетел и в течение нескольких лет обслуживал до 400 000 объявлений в секунду. Такой крупномасштабный трафик не был предусмотрен технологиями реляционных баз данных, доступными в то время. Настройка реляционных баз данных для такого масштаба также требовала значительных денежных и аппаратных ресурсов.

Давайте сформулируем задачи, которые должна решать новая БД.

1. Поддержка простого горизонтального масштабирования.

Так как нашей основной проблемой была сложность в масштабировании реляционной БД, то разрабатываемая БД должна позволять делать это намного проще.

2. Только необходимые функции для более быстрой разработки и легкого масштабирования.

Понятно, что масштабирование было не плохо делать при помощи функции. Ну грубо говоря, написал функцию create передал нужные параметры и новый сервер подсоединен.

Но как мы знаем, где достоинства, там и недостатки. Основным недостатком является: нет соединения между и транзакций между коллекциями. Почему этого нет? Все просто: настройка связей в значительной степени усложняет процесс масштабирования.

3. Документы без схем для гибкой итеративной разработки

В реляционной бд был еще один недостаток – сложность изменения созданной таблицы. Хочешь добавить или убавить строку в таблицу: ну придется потрудиться. А как мы с вами знаем, или не знаем, все хорошие принципы нужно закладывать на этапе проектирования.

4. Использование привычного языка и формата – JavaScript/JSON

Тут все логично, зачем придумывать что-то новое, когда большая часть людей уже умеет работать хорошо с каким-то одним инструментом.

5. Быстрая и простая модель данных для более быстрого программирования - модель документа с CRUD.

Тоже самое про хороший принцип программирования.

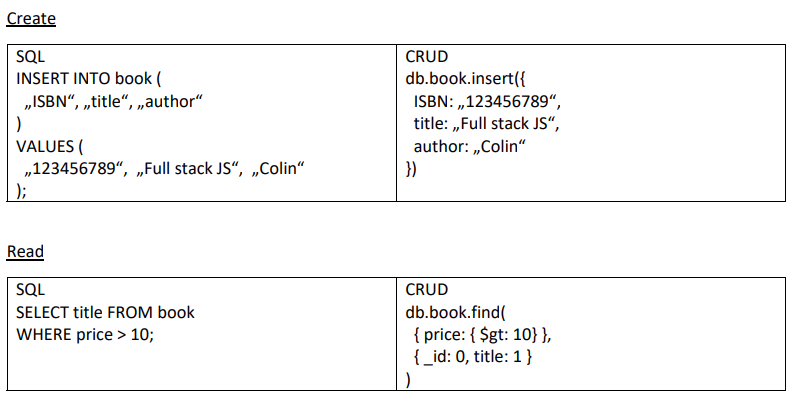
Ну вот разобрались с тем, что хотели сделать. Теперь посмотрим, что получилось, дав определение mongoDB

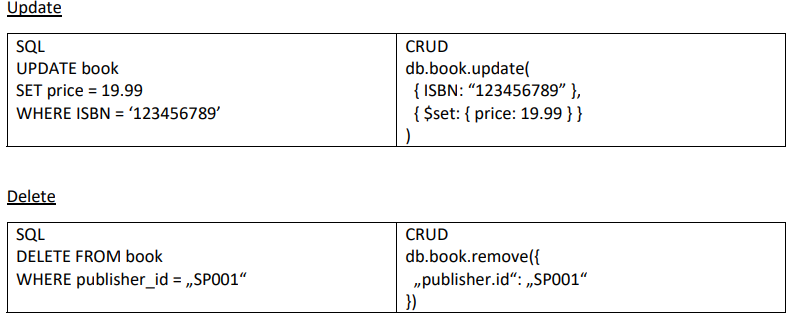
3. Определение

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.

4. Пример документа

Что касается доступа: там, где мы к реляционным данным используем язык SQL, в MongoDB и многих других NoSQL-базах данных используется такой стандарт, как CRUD. Этот стандарт говорит, что есть операции для создания, чтения, удаления и обновления документов.





5. Достоинства

Нет строгой схемы документов.

Надо что-то поменять – меняем Массивы, вложенные документы и т.д. Язык запросов JSON.

Горизонтальное масштабирование шардингом (отдельные части данных хранятся на разных серверах). Пользователь выбирает ключ шарда, который определяет как данные в коллекции будут распределены. Данные разделятся на диапазоны (в зависимости от ключа шарда) и распределятся по шардам.

Может хранить неструктурную информацию (двоичные данные)

Файловое хранилище GridFS. MongoDB может быть использован в качестве файлового хранилища с балансировкой нагрузки и репликацией данных.

Grid File System поставляется вместе с драйверами MongoDB. MongoDB предлагает разработчикам функции для работы с файлами и их содержимым. GridFS разделяет файл на части и хранит каждую часть как отдельный документ.

6. Недостатки

MongoDB гарантирует ACID в том же документе. Неспособность реализовать свойства ACID означает, что база данных не обеспечивает долговечность, целостность, согласованность и изоляцию, необходимые для транзакций. Возможно, что в будущих версиях это будет решено.

Отсутствие понятия изоляции

Блокировка БД происходит на уровне документа, т.е. при изменении одного документа, другие документы также можно изменять.

Отсутствие понятия транзакции

Атомарность гарантируется только на уровне документа. Проблемы с масштабируемостью Проблемы с производительностью появляются тогда, когда объем данных превышает 100 ТБ

Нет оператора JOIN

Обычно данные могут быть организованы более денормализованным способом, но на разработчиков ложится дополнительная нагрузка по обеспечению непротиворечивости данных.

Максимальный размер документа – 16 МБ

7. Область применения

Очень быстрая разработка.

Потому что всё можно постоянно менять, не нужно постоянно заботиться о строгом формате документа.

С MongoDB хорошо делать приложения, у которых очень ограниченный цикл жизни. То есть если мы делаем приложение, которое живёт недолго, например, сайт для запуска фильма или олимпиады. Мы пожили несколько месяцев после этого, и это приложение практически не используется. Если приложение живёт дольше, то тут уже другой вопрос.

Обработка больших объемов данных

Хранение и регистрация событий

Мобильные приложения

Проекты с гибкими методологиями разработки (Agile, Scrum, etc)