



# ОГЛАВЛЕНИЕ

<b>Предисловие .....</b>	<b>13</b>
<b>Благодарности.....</b>	<b>15</b>
<b>Об этой книге.....</b>	<b>16</b>
<b>Об иллюстрации на обложке.....</b>	<b>20</b>

## **ЧАСТЬ 1.**

<b>Приступая к работе .....</b>	<b>21</b>
---------------------------------	-----------

<b>Глава 1. База данных для современной веб.....</b>	<b>22</b>
------------------------------------------------------	-----------

1.1. Рожденная в облаке .....	24
1.2. Основные особенности MongoDB .....	25
1.2.1. Документная модель данных.....	25
1.2.2. Произвольные запросы .....	28
1.2.3. Вторичные индексы .....	31
1.2.4. Репликация.....	32
1.2.5. Быстродействие и долговечность .....	32
1.2.6. Масштабирование .....	34
1.3. Сервер и инструментальные средства MongoDB .....	36
1.3.1. Сервер.....	37
1.3.2. JavaScript-оболочка .....	38
1.3.3. Языковые драйверы.....	39
1.3.4. Командные утилиты .....	40
1.4. Почему именно MongoDB? .....	41
1.4.1. Сравнение MongoDB с другими СУБД.....	41
1.4.2. Сценарии использования и примеры реального развертывания .....	46
1.5. Советы и ограничения.....	48
1.6. Резюме .....	49

## **Глава 2. MongoDB сквозь призму**

<b>JavaScript-оболочки.....</b>	<b>51</b>
---------------------------------	-----------

2.1. Первое знакомство с оболочкой MongoDB.....	52
-------------------------------------------------	----

2.1.1. Запуск оболочки .....	52
2.1.2. Вставка и выборка .....	52
2.1.3. Обновление документов .....	55
2.1.4. Удаление данных .....	57
2.2. Создание индексов и применение их в запросах .....	58
2.2.1. Создание большой коллекции .....	58
2.2.2. Индексирование и команда explain() .....	60
2.3. Основы администрирования .....	63
2.3.1. Получение информации о базе данных .....	63
2.3.2. Как работают команды .....	65
2.4. Получение справки .....	66
2.5. Резюме .....	67

## **Глава 3. Разработка программ для MongoDB... 69**

3.1. MongoDB сквозь призму Ruby .....	70
3.1.1. Установка и подключение к базе .....	70
3.1.2. Вставка документов на Ruby .....	71
3.1.3. Запросы и курсоры .....	73
3.1.4. Обновление и удаление .....	74
3.1.5. Команды базы данных .....	75
3.2. Как работают драйверы .....	76
3.2.1. Генерация идентификатора объекта .....	76
3.2.2. Формат BSON .....	78
3.2.3. Передача по сети .....	80
3.3. Разработка простого приложения .....	81
3.3.1. Подготовка .....	82
3.3.2. Сбор данных .....	83
3.3.3. Визуализация архива .....	85
3.4. Резюме .....	88

## **ЧАСТЬ 2.**

## **Разработка приложения для MongoDB ..... 89**

## **Глава 4. Документо-ориентированные данные ..... 91**

4.1. Принципы проектирования схемы .....	92
4.2. Проектирование модели данных для интернет-магазина ..	94
4.2.1. Товары и категории .....	95
4.2.2. Пользователи и заказы .....	100
4.2.3. Отзывы .....	102
4.3. Технические детали: о базах данных, коллекциях и документах .....	104
4.3.1. Базы данных .....	104

4.3.2. Коллекции .....	107
4.3.3. Документы и вставка.....	111
4.4. Резюме .....	117

## **Глава 5. Запросы и агрегирование ..... 118**

5.1. Запросы в приложении для интернет-магазина .....	119
5.1.1. Товары, категории и обзоры .....	119
5.1.2. Пользователи и заказы .....	122
5.2. Язык запросов MongoDB.....	124
5.2.1. Селекторы запроса.....	124
5.2.2. Дополнительные средства.....	136
5.3. Агрегирование заказов .....	139
5.3.1. Группировка отзывов по пользователям .....	139
5.3.2. Распределение-редукция для заказов по регионам .....	141
5.4. Агрегирование в деталях.....	143
5.4.1. Максимум и минимум .....	143
5.4.2. Команда distinct .....	144
5.4.3. Команда group .....	144
5.4.4. Map-reduce.....	146
5.5. Резюме .....	149

## **Глава 6. Обновление, атомарные операции и удаление ..... 150**

6.1. Краткий экскурс в обновление документов .....	151
6.2. Обновление данных интернет-магазина .....	154
6.2.1. Товары и категории .....	154
6.2.2. Отзывы .....	159
6.2.3. Заказы .....	160
6.3. Атомарная обработка документа .....	164
6.3.1. Переходы состояний заказа.....	164
6.3.2. Управление запасами .....	166
6.4. Технические детали: обновление и удаление в MongoDB.....	171
6.4.1. Типы и параметры операций обновления.....	171
6.4.2. Операторы обновления .....	173
6.4.3. Команда findAndModify .....	178
6.4.4. Операции удаления .....	179
6.4.5. Параллелизм, атомарность и изолированность .....	179
6.4.6. Замечания о производительности обновления .....	180
6.5. Резюме .....	182

## **ЧАСТЬ 3.**

## **MongoDB – постижение мастерства..... 183**

<b>Глава 7. Индексирование и оптимизация запросов .....</b>	<b>185</b>
7.1. Теория индексирования .....	186
7.1.1. Мысленный эксперимент .....	186
7.1.2. Основные понятия индексирования .....	190
7.1.3. В-деревья .....	194
7.2. Индексирование на практике .....	196
7.2.1. Типы индексов .....	196
7.2.2. Администрирование индексов .....	199
7.3. Оптимизация запросов .....	204
7.3.1. Выявление медленных запросов .....	205
7.3.2. Исследование медленных запросов .....	208
7.3.3. Образцы запросов .....	216
7.4. Резюме .....	218
<b>Глава 8. Репликация .....</b>	<b>219</b>
8.1. Обзор репликации .....	219
8.1.1. Почему так важна репликация .....	220
8.1.2. Сценарии репликации .....	221
8.2. Наборы реплик .....	223
8.2.1. Настройка .....	223
8.2.2. Как работает репликация .....	229
8.2.3. Администрирование .....	236
8.3. Репликация типа главный-подчиненный .....	247
8.4. Драйверы и репликация .....	247
8.4.1. Подключение и отработка отказов .....	248
8.4.2. Гарантии записи .....	250
8.4.3. Масштабирование чтения .....	252
8.4.4. Тегирование .....	254
8.5. Резюме .....	256
<b>Глава 9. Сегментирование .....</b>	<b>257</b>
9.1. Обзор сегментирования .....	258
9.1.1. Что такое сегментирование .....	258
9.1.2. Как работает сегментирование .....	260
9.2. Тестовый сегментированный кластер .....	266
9.2.1. Настройка .....	266
9.2.2. Запись в сегментированный кластер .....	272
9.3. Индексирование сегментированного кластера и запросы к нему .....	278
9.3.1. Типы сегментированных запросов .....	278
9.3.2. Индексирование .....	283
9.4. Выбор сегментного ключа .....	284

9.4.1. Неэффективные сегментные ключи .....	284
9.4.2. Идеальные сегментные ключи .....	287
9.5. Сегментирование в производственных системах .....	288
9.5.1. Развертывание и конфигурирование .....	288
9.5.2. Администрирование .....	294
9.6. Резюме .....	300

## **Глава 10. Развертывание и администрирование ..... 301**

10.1. Развертывание.....	302
10.1.1. Среда развертывания .....	302
10.1.2. Конфигурирование сервера.....	308
10.1.3. Импорт и экспорт данных.....	310
10.1.4. Безопасность.....	312
10.2. Мониторинг и диагностика .....	315
10.2.1. Протоколирование.....	315
10.2.2. Средства мониторинга .....	316
10.2.3. Внешние приложения для мониторинга .....	320
10.2.4. Диагностические средства (mongosniff, bsondump) ..	<b>321</b>
10.3. Обслуживание.....	322
10.3.1. Резервное копирование и восстановление .....	322
10.3.2. Сжатие и ремонт .....	324
10.3.3. Модернизация .....	326
10.4. Разрешение проблем, связанных с производительностью.....	326
10.4.1. Проверка эффективности индексов и запросов .....	327
10.4.2. Добавление памяти .....	328
10.4.3. Повышение производительности дисков .....	329
10.4.4. Горизонтальное масштабирование.....	330
10.4.5. Обращение к профессионалам .....	330
10.5. Резюме .....	330

## **ПРИЛОЖЕНИЕ А.**

### **Установка..... 332**

A.1. Установка .....	332
A.1.1. MongoDB в Linux .....	332
A.1.2. MongoDB в Mac OS X .....	334
A.1.3. MongoDB в Windows.....	336
A.1.4. Компилирование MongoDB из исходного кода.....	337
A.1.5. Поиск и устранение неполадок .....	337
A.2. Основные конфигурационные параметры.....	339
A.3. Установка Ruby .....	340

A.3.1. Linux и Mac OS X .....	340
A.3.2. Windows.....	341

## ПРИЛОЖЕНИЕ В.

### Паттерны проектирования ..... 342

V.1. Паттерны .....	342
V.1.1. Вложение или ссылка .....	342
V.1.2. Связь один-ко-многим .....	343
V.1.3. Связь многие-ко-многим.....	344
V.1.4. Деревья.....	345
V.1.5. Очереди .....	348
V.1.6. Динамические атрибуты.....	349
V.1.7. Транзакции.....	351
V.1.8. Локальность и предвычисления.....	352
V.2. Антипаттерны .....	353
V.2.1. Непродуманное индексирование .....	353
V.2.2. Смешанные типы.....	354
V.2.3. Коллекции-свалки .....	354
V.2.4. Большие документы с глубокой вложенностью.....	354
V.2.5. Одна коллекция на каждого пользователя .....	355
V.2.6. Несегментируемые коллекции .....	355

## ПРИЛОЖЕНИЕ С.

### Двоичные данные и GridFS ..... 356

C.1. Хранение простых двоичных объектов.....	357
C.1.1. Хранение миниатюр .....	357
C.1.2. Хранение MD5-свертки .....	358
C.2. GridFS.....	359
C.2.1. GridFS в Ruby.....	359
C.2.2. Доступ к GridFS с помощью mongofiles .....	362

## ПРИЛОЖЕНИЕ D.

### MongoDB на PHP, Java и C++ ..... 364

D.1. PHP .....	365
D.1.1. Документы .....	365
D.1.2. Подключение.....	365
D.1.3. Пример программы .....	366
D.2. Java .....	367
D.2.1. Документы .....	367
D.2.2. Подключение.....	368
D.2.3. Пример программы.....	368
D.3. C++.....	370

D.3.1. Документы .....	370
D.3.2. Подключение.....	371
D.3.3. Пример программы .....	372

## **ПРИЛОЖЕНИЕ Е.**

### **Пространственные индексы ..... 374**

E.1. Основы пространственного индексирования .....	375
E.2. Более сложные запросы .....	377
E.3. Составные пространственные индексы.....	378
E.4. Сферическая геометрия .....	379

### **Предметный указатель ..... 381**



# ПРЕДИСЛОВИЕ

Базы данных – рабочие лошадки информационной эпохи. Как атланты, они незримо подпирают цифровой мир, в котором мы обитаем. Легко позабыть, что всё наше цифровое общение, будь то комментарии в форумах, сообщения в Твиттере, поиск или сортировка, – это по существу взаимодействие с базой данных. Размышляя об этой краеугольной и в то же время скрытой функции, я испытываю чувство благоговейного трепета, сродни тому, которое возникает, когда идешь по подвесному месту, предназначенному в основном для автомобилей.

Базы данных принимают самые разные формы. Предметные указатели в книгах, каталожные карточки, когда-то стоявшие в библиотеках, – это тоже базы данных, как и специально структурированные текстовые файлы, излюбленные программистами былых времен, работавшими на языке Perl. Но, наверное, чаще всего базы данных ассоциируются с изощренными реляционными СУБД, лежащими в основе огромного количества современных программных систем, теми самыми СУБД, на которых делаются состояния. Реляционные базы, с их идеализированной третьей нормальной формой и выразительным SQL-интерфейсом, все еще вызывают уважение старой гвардии – и вполне заслуженно.

Но несколько лет назад, разрабатывая веб-приложения, я мечтал опробовать только появляющиеся на свет альтернативы правящим в мире реляционным базам. Открыв для себя MongoDB, я сразу понял – это как раз то, что мне надо. Мне понравилась идея использовать JSON-подобные структуры для представления данных. Язык JSON прост, интуитивно понятен и удобен для человека. И тот факт, что язык запросов в MongoDB также основан на JSON вселяет в пользователя этой новой СУБД ощущение комфорта и гармонии. На первом месте стоит интерфейс. А наличие таких привлекательных средств, как простая репликация и сегментирование, только подстегивает интерес к пакету. Но окончательно мое обращение состоялось, когда я написал несколько приложений на основе MongoDB и в полной мере осознал, как легко дается разработка.



По совершенно невероятному стечению обстоятельств я перешел на работу в компанию 10gen, которая была зачинателем разработки этой СУБД с открытым кодом. В течение двух лет у меня была возможность совершенствовать различные клиентские драйверы и общаться с многочисленными заказчиками, помогая им развертывать системы на основе MongoDB. Накопленный опыт и вылился в книгу, которую вы держите в руках. MongoDB активно развивается и пока еще далека от совершенства. Но она уже поддерживает тысячи приложений, использующих крупные и мелкие кластеры базы данных, и с каждым днем становится все более зрелой. Многие разработчики признавались, что при работе с ней испытывают ощущение чуда и даже счастья. Надеюсь, то же самое произойдет и с вами.



# ОБ ЭТОЙ КНИГЕ

Эта книга предназначена разработчикам приложений и администраторам баз данных, которые хотели бы узнать всё о MongoDB, начиная с самых основ. Если вы раньше не работали с MongoDB, то найдете здесь учебное пособие, не слишком стремительное. Если вы уже пользуетесь этой СУБД, то вам пригодятся более детальные справочные разделы, которые помогут восполнить пробелы в знаниях. Что касается глубины изложения, то материал должен удовлетворить всех, кроме самых продвинутых пользователей.

Примеры написаны на JavaScript, языке оболочки MongoDB, и на Ruby, популярном скриптовом языке. Я стремился, чтобы примеры были простыми и в то же время полезными, и не использовал сколько-нибудь сложных языковых средств JavaScript и Ruby. Основная цель – представить MongoDB API в максимально доступном виде. Если вы знакомы с другими языками программирования, то легко разберетесь в примерах.

И еще одно замечание о языках. Если у вас возникает вопрос «Почему бы не использовать в этой книге язык X?», можете расслабиться. Все официально поддерживаемые драйверы MongoDB предоставляют единообразный API. Это означает, что изучив основы API для какого-нибудь одного драйвера, вы легко освоите и все остальные. А чтобы помочь вам в этом, в приложении D приведен обзор драйверов для языков PHP, Java и C++.

## Как пользоваться этой книгой

Эта книга представляет собой одновременно учебное пособие и справочное руководство. Если вы ничего не знаете о MongoDB, то имеет смысл читать ее по порядку. В тексте много примеров, которые вы можете выполнять для закрепления пройденного. Как минимум, необходимо установить саму MongoDB; драйвер Ruby желателен, но не обязателен. Инструкции по установке приведены в приложении А.

Если вы уже работали с MongoDB, то, возможно, вас будут интересовать какие-то конкретные вопросы. Главы 7 – 10 и все приложения не зависят друг от друга, читать их можно в любом порядке.

Кроме того, главы 4 – 6 содержат «основные моменты» – фундаментальные принципы системы. К ним также можно обращаться из любого места.

## Организация материала

Книга состоит из трех частей.

Часть 1 представляет собой сквозное введение в MongoDB. В главе 1 приводится обзор истории MongoDB, основных возможностей и сценариев использования. В главе 2 основные идеи СУБД излагаются в виде учебного пособия с использованием командной оболочки MongoDB. В главе 3 приводится пример проектирования простого приложения на основе MongoDB.

Часть 2 содержит более подробное описание MongoDB API, представленного в части 1. В трех входящих в нее главах рассматривается разработка приложения для электронной торговли, начиная со схемы базы данных и заканчивая различными операциями. В главе 4 речь пойдет о документах, наименьшей единице хранения данных в MongoDB, здесь же описывается проектирование простой схемы. В главах 5 и 6 мы будем работать с этой схемой и познакомимся с запросами для выборки и обновления данных. Дополнительно в каждой из глав второй части содержатся углубленные комментарии к излагаемому материалу.

Часть 3 посвящена вопросам производительности и промышленной эксплуатации. В главе 7 подробно рассматривается индексирование и оптимизация запросов. Тема главы 8 – репликация, с упором на стратегию развертывания MongoDB для обеспечения высокой доступности и масштабируемости операций чтения. В главе 9 описывается сегментирование – ответ MongoDB на проблему горизонтальной масштабируемости. В главе 10 приведены рекомендации по развертыванию, администрированию и отладке систем на основе MongoDB.

В книге имеется пять приложений. В приложении А рассматривается установка MongoDB и Ruby (как пример драйвера) в Linux, Mac OS X и Windows. В приложении В приведен ряд паттернов проектирования схемы и приложения, а также несколько антипаттернов. В приложении С рассказано о работе с двоичными данными в MongoDB и об использовании GridFS – реализованной во всех драйверах спецификации хранения в базе очень больших файлов. Приложение D содержит сравнительный анализ драйверов для языков PHP, Java и C++. В приложении E показано, как использовать пространственные индексы для запроса по географическим координатам.

## Графические выделения и загрузка кода

Исходный код в листингах и в тексте набран моноширинным шрифтом, выделяющим его на фоне обычного текста.

Некоторые листинги сопровождаются аннотациями, в которых обсуждаются важные идеи. Иногда фрагменты кода помечаются числами, на которые потом даются ссылки в пояснениях. Поскольку MongoDB – проект с открытым кодом, компания 10gen поддерживает систему сбора информации об ошибках, открытую для всего сообщества. В нескольких местах, по преимуществу в сносках, вы встретите ссылки на извещения об ошибках и о планируемых усовершенствованиях. Например, заявка на добавление средств полнотекстового поиска имеет номер SERVER-380. Чтобы узнать о состоянии любой заявки, перейдите в браузере по адресу <http://jira.mongodb.org> и введите ее идентификатор в поисковое поле.

Исходный код приведенных в книге примеров, а также тестовые данные, можно скачать с сайта книги по адресу <http://mongodb-book.com>, а также с сайта издательства по адресу <http://manning.com/MongoDBinAction>.

## Требования к программному обеспечению

Чтобы получить от книги максимальную пользу, необходимо установить MongoDB на свой компьютер. Инструкции по установке приведены в приложении А и на официальном сайте MongoDB (<http://mongodb.org>).

Если вы собираетесь выполнять примеры, написанные на Ruby, то надо будет установить Ruby. Инструкции также имеются в приложении А.

## Автор в Сети

Приобретение книги «MongoDB в действии» открывает бесплатный доступ к закрытому форуму, организованному издательством Manning Publications, где вы можете оставить свои комментарии к книге, задать технические вопросы и получить помощь от автора и других пользователей. Получить доступ к форуму и подписаться на список рассылки можно на странице [www.manning.com/MongoDBinAction](http://www.manning.com/MongoDBinAction). Там же написано, как зайти на форум после регистрации, на какую помощь можно рассчитывать, и изложены правила поведения в форуме.

Издательство Manning обязуется предоставлять читателям площадку для общения с другими читателями и автором. Однако это

не означает, что автор обязан как-то участвовать в обсуждениях; его присутствие на форуме остается чисто добровольным (и не оплачивается). Мы советуем задавать автору хитроумные вопросы, чтобы его интерес к форуму не угасал!

Форум автора в сети и архивы будут доступны на сайте издательства до тех пор, пока книга не перестанет печататься.



# Часть 1

## ПРИСТУПАЯ К РАБОТЕ

Эта часть книги представляет собой общее практическое введение в MongoDB. Здесь же мы познакомимся с оболочкой, в которой применяется язык JavaScript, и с драйвером для Ruby. Примеры в книге написаны именно на этих языках.

В главе 1 мы поговорим об истории СУБД MongoDB, целях, поставленных при ее проектировании, и о сценариях использования. Мы также сравним MongoDB с другими СУБД класса «NoSQL» и отметим, в чем ее уникальность.

В главе 2 вы заговорите на языке оболочки MongoDB — познакомьтесь с основами языка запросов MongoDB и потренируетесь в создании, выборке, обновлении и удалении документов. Заканчивается глава рассмотрением нескольких продвинутых приемов работы с оболочкой и дополнительных команд MongoDB.

Глава 3 содержит введение в драйверы MongoDB и внутренний формат данных — BSON. Здесь мы узнаем, как обращаться к базе данных из программ на языке Ruby, и напишем на Ruby простенькое приложение, демонстрирующее гибкость и мощь языка запросов MongoDB.



# ГЛАВА 1.

## База данных для современной веб

### *В этой главе:*

- История, проектные цели и основные особенности MongoDB.
- Краткое введение в оболочку и драйверы.
- Сценарии использования и ограничения.

Если за последние несколько лет вам доводилось писать веб-приложения, то, вероятно, в качестве основного хранилища данных вы пользовались какой-нибудь реляционной базой и, скорее всего, она демонстрировала прекрасную производительность. Большинство разработчиков знакомы с языком SQL и могут оценить красоту хорошо нормализованной модели данных, необходимость транзакций и те гарантии долговечности, которые дает СУБД. И даже если вам не нравится работать с реляционной базой данных напрямую, существует масса инструментов – от административных консолей до систем объектно-реляционного отображения, которые помогают справиться со сложностью и громоздкостью. Проще говоря, реляционные СУБД хорошо известны и достигли зрелости. Поэтому, когда небольшая, но громогласная группа разработчиков начинает агитировать за альтернативные хранилища данных, возникает вопрос: насколько полезны и жизнеспособны эти новые технологии? Смогут ли они стать заменой реляционным СУБД? Кто использует их в производственных системах и почему? На какие компромиссы придется пойти, если переходить к нереляционной базе данных? Все эти вопросы сводятся к одному: почему разработчики проявляют интерес к MongoDB?

MongoDB – это система управления базами данных, «заточенная» под веб-приложения и инфраструктуру Интернета. Модель данных и стратегия их постоянного хранения спроектированы для достижения высокой пропускной способности чтения и записи и обеспечивает простую масштабируемость с автоматическим переходом на резервный ресурс в случае отказа. Сколько бы узлов ни требовалось приложению – один или десятки, – MongoDB сумеет обеспечить поразительно высокую производительность. Того, кто раньше мучился с масштабированием реляционных баз, эта новость обрадует. Но не всем приложениям необходим крупный масштаб. Быть может, вам всегда хватало одного сервера базы данных. Зачем тогда использовать MongoDB?

Как выясняется, привлекательность MongoDB объясняется в первую очередь не стратегией масштабирования, а интуитивно понятной моделью данных. Учитывая, что с помощью документо-ориентированной модели можно представить развитые иерархически организованные структуры данных, часто оказывается возможно обойтись без присущих реляционным СУБД сложностей, связанных с соединением нескольких таблиц. Пусть, например, вы моделируете товары для сайта интернет-магазина. В полностью нормализованной базе данных информация об одном товаре может быть разбросана по десяткам таблиц. Чтобы получить его представление в интерактивной оболочке СУБД, придется написать сложный SQL-запрос с кучей соединений. Поэтому разработчики, как правило, обращаются к дополнительным программам, когда хотят собрать разрозненные данные в нечто осмысленное.

С другой стороны, в документной модели большая часть информации о товаре может быть представлена в виде одного документа. В оболочке MongoDB, построенной на основе языка JavaScript, нетрудно получить полное представление о товаре в виде иерархически организованной JSON-подобной структуры<sup>1</sup>. К ней можно предъявлять запросы, ей можно манипулировать. Средства составления запросов в MongoDB специально спроектированы для работы со структурированными документами, но так, чтобы пользователь, имеющий опыт работы с реляционными базами, располагал сравнимой выразительной мощностью. К тому же, сегодня большинство разработчиков пишут на объектно-ориентированных языках, поэтому им нужно такое хра-

---

<sup>1</sup> JSON – это сокращение от *JavaScript Object Notation*. Как мы вскоре увидим, JSON-структуры состоят из ключей и значений и допускают произвольную глубину вложенности. Они аналогичны словарям и хешам в других языках программирования.



нилище, которое было бы проще отобразить на объекты. В случае MongoDB объект, определенный на языке программирования, сохраняется «как есть» – без дополнительной сложности, привносимой системами объектно-реляционного отображения.

Если различие между табличным и объектным представлением данных для вас новость, то, наверное, у вас возникла куча вопросов. Но потерпите немного – после того как в этой главе я расскажу о проектных целях и возможностях MongoDB, вам станет ясно, почему такие компании, как Geek.net (SourceForge.net) и газета The New York Times стали использовать в своих проектах MongoDB. Мы познакомимся с историей MongoDB и совершим экскурсию по ее основным средствам. Затем мы поговорим о некоторых альтернативах и о так называемом движении NoSQL<sup>2</sup>, показав место MongoDB в общей картине. Наконец, я расскажу, в каких ситуациях MongoDB работает прекрасно, а когда лучше предпочесть другое хранилище данных.

## 1.1. Рожденная в облаке

История MongoDB, хоть и коротка, но заслуживает упоминания, потому что родилась эта СУБД в ходе работы над гораздо более амбициозным проектом. В середине 2007 года только что образованная компания 10gen приступила к разработке проекта «программная платформа как услуга». Идея была в том, чтобы создать сервер приложений и базу данных, которые могли бы служить хостингом для веб-приложений, обеспечивая масштабирование по мере необходимости. Как и система AppEngine, созданная Google, платформа компании 10gen проектировалась с расчетом на автоматическое масштабирование и управление программной и аппаратной инфраструктурой. В итоге 10gen обнаружила, что большинство разработчиков не готовы отдать в чужие руки управление своим технологическим хозяйством, но зато новая технология баз данных оказалась востребованной. Поэтому компания решила сосредоточить усилия исключительно на этой СУБД, которая получила название MongoDB.

Теперь, когда признание MongoDB ширится, а количество промышленных установок – крупных и не очень – растет, компания 10gen продолжает спонсировать разработку СУБД в качестве проекта с открытым кодом. Код общедоступен, открыт для модификации

---

<sup>2</sup> Собираемый термин NoSQL был введен в обращение в 2009 году для обозначения многочисленных нереляционных СУБД, которые в то время набирали популярность.

и использования на условиях лицензии. Всё сообщество призывают сообщать об ошибках и предлагать исправления. Тем не менее, разработкой ядра занимаются исключительно основатели или работники компании 10gen, а планы развития проекта по-прежнему определяются нуждами пользователей и главной целью – создать такую СУБД, которая объединяла бы в себе лучшие черты реляционных баз данных и распределенных хранилищ ключей и значений. Таким образом, бизнес-модель 10gen в основном такая же, как в других хорошо известных компаниях, занимающихся ПО с открытым кодом: поддерживать разработку продукта и предлагать абонентское обслуживание конечным пользователям.

В этой истории заключено два важных урока. Во-первых, MongoDB изначально разрабатывалась для платформы, которая по определению нуждалась в базе данных, допускающей автоматическое масштабирование на несколько машин. Во-вторых, MongoDB проектировалась как хранилище данных для веб-приложений. Как мы увидим, именно тот факт, что MongoDB задумана как горизонтально масштабируемое первичное хранилище данных, и выделяет ее на фоне других современных СУБД.

## 1.2. Основные особенности MongoDB

База данных в значительной мере определяется своей моделью данных. В этом разделе мы рассмотрим документную модель данных, а затем покажем, какие особенности MongoDB позволяют эффективно работать с этой моделью. Мы также обсудим вопросы промышленной эксплуатации MongoDB, уделив особое внимание средствам репликации и стратегии горизонтального масштабирования.

### 1.2.1. Документная модель данных

Модель данных MongoDB является документо-ориентированной. Для тех, кто не знаком с идеей документа в контексте баз данных, продемонстрировать ее проще всего на примере.

**Листинг 1.1.** Документ, представляющий одну статью на социальном новостном сайте

```
{  _id: ObjectID('4bd9e8e17cefd644108961bb'),  
  title: 'Adventures in Databases',
```



Поле `_id` – первичный ключ

```

url: 'http://example.com/databases.txt',
author: 'msmith',
vote_count: 20,

tags: ['databases', 'mongodb', 'indexing'],

image: {
  url: 'http://example.com/db.jpg',
  caption: '',
  type: 'jpg',
  size: 75381,
  data: "Binary"
},

comments: [
  { user: 'bjones',
    text: 'Interesting article!'
  },
  { user: 'blogger',
    text: 'Another related article is at http://example.com/db/db.txt'
  }
]
}

```

1 Теги хранятся в виде массива строк

2 Атрибут указывает на другой документ

3 Комментарии хранятся в виде массива объектов, представляющих один комментарий

В листинге 1.1 приведен пример документа, представляющего статью на социальном новостном сайте (вспомните о Digg). Как видите, документ – это по существу набор, состоящий из имен и значений свойств. Значение может быть представлено простым типом, например: строки, числа и даты. Но может быть также массивом и даже другим документом 2. С помощью таких конструкций можно представлять весьма сложные структуры данных. Так, в нашем примере имеется свойство `tags` 1 – массив, в котором хранятся ассоциированные со статьей теги. Но еще интереснее свойство `comments` 3, которое ссылается на массив документов, содержащих комментарии.

Сделаем небольшую паузу и сравним это с представлением тех же данных в стандартной реляционной базе. На рис. 1.1 изображен типичный реляционный аналог. Поскольку таблицы по сути своей плоские, для представления связей типа один-ко-многим необходимо несколько таблиц. Мы начинаем с таблицы `posts`, в которой хранится основная информация о каждой статье. Затем создаем еще три таблицы, каждая из которых содержит поле `post_id`, ссылающееся на исходную статью. Эта техника разнесения данных объекта по нескольким таблицам называется *нормализацией*. Среди прочего, нормализованный набор данных характеризуется тем, что каждый элемент данных хранится только в одном месте.