



# СОДЕРЖАНИЕ

<b>ПРЕДИСЛОВИЕ .....</b>	<b>13</b>
<b>БЛАГОДАРНОСТИ .....</b>	<b>15</b>
<b>ВСТУПЛЕНИЕ .....</b>	<b>17</b>
Почему именно семь баз данных? .....	17
Что есть в этой книге? .....	17
Чего нет в этой книге? .....	18
Это не руководство по установке .....	18
Руководство администратора? Пожалуй, нет .....	19
Замечание для пользователей Windows .....	19
Примеры кода и соглашения .....	19
Сетевые ресурсы .....	20
<b>Глава 1. Введение .....</b>	<b>21</b>
1.1. Все начинается с вопроса .....	21
1.2. Жанры .....	23
Реляционные СУБД .....	24
Хранилища ключей и значений .....	25
Столбцовые базы данных .....	26
Документо-ориентированные базы данных .....	27
Графовые базы данных .....	28
Многостороннее хранение .....	28
1.3. Вперед и вверх .....	29
<b>Глава 2. PostgreSQL .....</b>	<b>30</b>
2.1. Произносится Post-greS-Q-L .....	30
2.2. День 1: отношения, операции CRUD и соединения .....	32
Введение в SQL .....	33

Быстрый поиск с применением индексов .....	41
День 1: итоги .....	43
День 1: домашнее задание .....	44
<b>2.3. День 2: более сложные запросы, код и правила .....</b>	<b>45</b>
Агрегатные функции .....	45
Оконные функции .....	48
Транзакции .....	49
Хранимые процедуры .....	50
Триггеры .....	52
Представление о мире .....	54
Правила .....	55
Создание сводных таблиц с помощью crosstab() .....	57
День 2: итоги .....	59
День 2: домашнее задание .....	59
<b>2.4. День 3: полнотекстовый поиск и многомерные кубы .....</b>	<b>60</b>
Нечеткий поиск .....	62
Полнотекстовый поиск .....	65
День 3: итоги .....	75
День 3: домашнее задание .....	75
<b>2.5. Резюме .....</b>	<b>75</b>
Сильные стороны PostgreSQL .....	76
Слабые стороны PostgreSQL .....	77
Перед расставанием .....	77
<b>Глава 3. Riak .....</b>	<b>78</b>
<b>3.1. Riak дружит с веб .....</b>	<b>79</b>
<b>3.2. День 1: CRUD, ссылки и типы MIME .....</b>	<b>80</b>
Лучше REST может быть только REST (или как завивать локоны) .....	82
Ссылки .....	85
Типы MIME в Riak .....	89
День 1: итоги .....	90
День 1: домашнее задание .....	90
<b>3.3. День 2: mapreduce и кластеры серверов .....</b>	<b>91</b>
Скрипт для загрузки данных .....	91
Введение в Mapreduce .....	92
Mapreduce в Riak .....	95
О согласованности и долговечности .....	101
День 2: итоги .....	109
День 2: домашнее задание .....	109

3.4. День 3: разрешение конфликтов и расширение Riak ..	110
Разрешение конфликтов с помощью векторных часов.....	110
Расширение Riak .....	117
День 3: итоги.....	121
День 3: домашнее задание.....	122
3.5. Резюме .....	122
Сильные стороны Riak.....	123
Слабые стороны Riak.....	123
Riak и теорема CAP .....	123
Перед расставанием.....	124

## **Глава 4. HBase..... 125**

4.1. Введение в HBase .....	126
4.2. День 1: операции CRUD и администрирование таблиц .....	127
Конфигурирование HBase .....	128
Оболочка HBase .....	129
Создание таблицы .....	129
Вставка, обновление и выборка данных .....	131
Добавление данных из программы.....	136
День 1: итоги.....	137
День 1: домашнее задание.....	138
4.3. День 2: работа с «большими данными».....	139
Импорт данных, выполнение скриптов .....	139
Потоковая загрузка XML.....	140
Загрузка википедии .....	141
Сжатие и фильтры Блума.....	143
Контакт? Есть контакт!.....	143
Знакомство с регионами и мониторингом места на диске.....	145
Опрос регионов .....	146
Сканирование одной таблицы для построения другой.....	149
Построение сканера.....	150
Запуск скрипта.....	152
Исследование результатов.....	153
День 2: итоги.....	154
День 2: домашнее задание.....	155
4.4. День 3: переходим в облако .....	156
Разработка «бережливого» приложения для HBase .....	156
Введение в Whirr .....	160
Подготовка к работе с EC2 .....	160
Подготовка Whirr .....	161

Настройка кластера .....	162
Запуск кластера .....	163
Подключение к кластеру.....	163
Уничтожение кластера .....	164
День 3: итоги.....	164
День 3: домашнее задание .....	165
4.5. Резюме .....	166
Сильные стороны HBase.....	166
Слабые стороны HBase .....	167
HBase и теорема CAP .....	167
Перед расставанием .....	168
<b>Глава 5. MongoDB .....</b>	<b>169</b>
5.1. Монстр.....	169
5.2. День 1: операции CRUD и вложенность .....	171
Поработаем с командной строкой .....	171
JavaScript .....	173
Чтение: продолжаем изучать Mongo.....	175
Копнем глубже .....	177
Обновление .....	181
Ссылки .....	183
Удаление .....	184
Функциональные критерии.....	185
День 1: итоги.....	186
День 1: домашнее задание .....	186
5.3. День 2: индексирование, группировка, mapreduce .....	187
Индексирование: когда быстродействия не хватает .....	187
Агрегированные запросы .....	191
Команды на стороне сервера .....	194
Mapreduce (и Finalize) .....	197
День 2: итоги.....	201
День 2: домашнее задание .....	201
5.4. День 3: наборы реплик, сегментирование, пространственные данные и GridFS .....	201
Наборы реплик.....	202
Сегментирование.....	206
Пространственные запросы .....	208
GridFS .....	210
День 3: итоги.....	211
День 3: домашнее задание .....	211
5.5. Резюме .....	212

Сильные стороны Mongo .....	212
Слабые стороны Mongo .....	212
Перед расставанием .....	213
<b>Глава 6. CouchDB .....</b>	<b>214</b>
6.1. Располагайтесь на кушетке .....	214
Сравнение CouchDB с MongoDB .....	215
6.2. День 1: операции CRUD, Futon и снова cURL .....	215
Знакомство с Futon .....	216
Выполнение операций CRUD с помощью REST-интерфейса и cURL .....	219
Чтение документа с помощью GET .....	220
Создание документа с помощью POST .....	221
Обновление документа с помощью PUT .....	222
Удаление документа с помощью DELETE .....	223
День 1: итоги .....	223
День 1: домашнее задание .....	223
6.3. День 2: создание и опрос представлений .....	224
Доступ к документам через представления .....	224
Создание первого представления .....	226
Сохранение представления в виде проектного документа .....	229
Поиск исполнителей по имени .....	229
Поиск альбомов по названию .....	230
Опрос представлений исполнителей и альбомов .....	231
Импорт данных в CouchDB с помощью программы на Ruby .....	233
День 2: итоги .....	238
День 2: домашнее задание .....	238
6.4. День 3: более сложные представления, Changes API и репликация данных .....	239
Создание более сложных представлений с помощью редукторов .....	239
Отслеживание изменений в CouchDB .....	243
Непрерывное отслеживание изменений .....	249
Фильтрация изменений .....	250
Репликация данных в CouchDB .....	252
День 3: итоги .....	256
День 3: домашнее задание .....	256
6.5. Резюме .....	257
Сильные стороны CouchDB .....	257
Слабые стороны CouchDB .....	258
Перед расставанием .....	258

<b>Глава 7. Neo4J.....</b>	<b>259</b>
7.1. Neo4J дружит с доской.....	259
7.2. День 1: графы, Groovy и операции CRUD .....	261
Веб-интерфейс Neo4j.....	262
Neo4j и Gremlin .....	264
Конвейеры .....	267
Конвейер и вершина .....	269
Бессхемная социальная сеть.....	270
Дорога меряется шагами .....	271
Обновляем, удаляем, стираем .....	278
День 1: итоги.....	279
День 1: домашнее задание.....	279
7.3. День 2: REST, индексы и алгоритмы.....	279
REST-интерфейс.....	279
Интересные алгоритмы.....	286
День 2: итоги.....	292
День 2: домашнее задание.....	292
7.4. День 3: распределенность и высокая доступность .....	293
Транзакции .....	293
Высокая доступность .....	294
HA-кластер.....	295
Резервное копирование .....	301
День 3: итоги.....	302
День 3: домашнее задание.....	302
7.5. Резюме .....	302
Сильные стороны Neo4j.....	303
Слабые стороны Neo4j .....	303
Neo4j и теорема CAP .....	304
Перед расставанием.....	304
<b>Глава 8. Redis.....</b>	<b>305</b>
8.1. Хранилище сервера структур данных .....	305
8.2. День 1: операции CRUD и типы данных.....	306
Приступая к работе .....	307
Транзакции .....	309
Составные типы данных .....	309
Блокирующие списки .....	313
Диапазоны .....	316
Пространства имен .....	319
И это еще не всё.....	320

День 1: итоги.....	321
День 1: домашнее задание.....	321
8.3. День 2: более сложные применения, распределенные вычисления.....	321
Простой интерфейс .....	322
Информация о сервере .....	325
Настройка Redis .....	325
Репликация главный-подчиненный.....	330
Загрузка данных.....	330
Кластер Redis.....	333
Фильтры Блума .....	334
SETBIT и GETBIT .....	337
День 2: итоги.....	338
День 2: домашнее задание.....	338
8.4. День 3: комбинирование с другими базами данных....	339
Служба многостороннего хранения.....	339
Заполнение данными .....	341
Фаза 1: трансформация данных .....	342
Фаза 2: вставка в каноническую систему.....	344
Хранилище связей .....	347
Веб-служба .....	349
Развитие веб-службы.....	351
День 3: итоги.....	352
День 3: домашнее задание.....	353
8.5. Резюме .....	353
Сильные стороны Redis.....	353
Слабые стороны Redis.....	354
Перед расставанием.....	354
<b>Глава 9. Подводя итоги .....</b>	<b>356</b>
9.1. Снова о жанрах .....	356
Реляционные базы данных .....	356
Хранилища ключей и значений .....	357
Столбцовые базы данных .....	358
Документные базы данных .....	359
Графовые базы данных .....	360
9.2. Как сделать выбор?.....	361
9.3. В каком направлении двигаться дальше? .....	362
<b>ПРИЛОЖЕНИЕ 1. Сравнительный обзор баз данных .....</b>	<b>363</b>

<b>ПРИЛОЖЕНИЕ 2. Теорема CAP .....</b>	<b>367</b>
A2.1. Согласованность в конечном счете.....	368
A2.2. CAP на практике .....	369
A2.3. Компромиссный выбор задержки.....	370
<b>СПИСОК ЛИТЕРАТУРЫ .....</b>	<b>371</b>
<b>ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ .....</b>	<b>372</b>





# ПРЕДИСЛОВИЕ

Сидя в креслах подъемника Бивер Ран Суперчер в Брикенридже, штат Колорадо, мы недоумевали, где же свежий снег. Склоны горы содержались в безукоризненном порядке, но везде условия огорчали неизменным однообразием. Без свежевывающего снега на настоящее удовольствие рассчитывать было нечего.

В 1994, работая в лаборатории IBM по разработке баз данных в Остине, я испытывал примерно такие же чувства. В Техасском университете в Остине я изучал объектно-ориентированные базы данных, так как полагал, что после десятилетнего доминирования реляционных СУБД у них появился реальный шанс пустить корни. Но и следующее десятилетие не принесло ничего, кроме все той же реляционной модели. Я удрученно наблюдал, как Oracle, IBM, а позднее продукты с открытым исходным кодом и, прежде всего, MySQL раскидывали ветви, полностью лишая солнечного света любые всходы, пытающиеся взрасти на плодородной почве внизу.

Пользовательские интерфейсы эволюционировали – зеленые буквы на черном экране сменились клиент-серверными, а потом и Интернет-приложениями, но код работы с реляционным уровнем был отмечен печатью унылого постоянства, пусть и доведенного до совершенства. Все ждали, когда же выпадет свежий снег.

И наконец начался снегопад. Поначалу снега не хватало даже для того, чтобы запорошить вчерашнюю лыжню, но постепенно пурга усиливалась, меняла ландшафт, неся с собой разнообразие и радость катания, которого мы так жаждали. Как раз в тот год я осознал, что мир баз данных тоже накрыло покрывалом свежевывающего снега. Разумеется, реляционные СУБД никуда не делись, и продукты с открытым исходным кодом по-прежнему способны решать поразительно разнообразные задачи. В них появилась кластеризация, полнотекстовый и даже нечеткий поиск. Но этот подход перестал быть единственным. За год я не создал ни одного решения, которое было бы целиком реляционным. В тот период я использовал документо-ориентированную базу данных и два хранилища ключей и значений.

Реляционные базы данных лишились монополии на гибкость и даже на масштабируемость. Для приложений, над которыми мы работали, существовали более подходящие модели – проще, быстрее и надежнее. Для человека, который десять лет занимался в Остинском подразделении IBM базами данных в интересах как самой лаборатории, так и заказчиков такое развитие событий было просто ошеломительным. В этой книге мы на примерах рассмотрим представительную выборку наиболее важных прорывов в области баз данных, цель которых – поддержать разработку Интернет-приложений. Вы узнаете о хранилищах ключей и значений – потрясающей масштабируемости и надежности Riak и красивом механизме запросов в Redis. Мы поговорим о достижениях сообщества столбцовых баз данных – мощи HBase, близкого родственника реляционных СУБД. Документо-ориентированные базы данных мы рассмотрим на примере прекрасно масштабируемой системы MongoDB, позволяющей элегантно хранить документы с глубокой вложенностью. Вы познакомитесь также с подходом Neo4J к графовым базам данных, которые позволяют быстро обходить связи.

Чтобы повысить свою квалификацию как программиста или администратора баз данных, необязательно использовать все эти продукты. Эрик Редмонд и Джим Уилсон станут вашими проводниками в этом удивительном путешествии, каждый этап которого сделает вас образованнее и принесет бесценные знания, так необходимые современному профессиональному разработчику программного обеспечения. Вы узнаете о сильных сторонах и ограничениях каждой из рассматриваемых платформ. Вы увидите, в каком направлении движется индустрия, и поймете, какие побудительные мотивы стоят за этим движением.

Приятного путешествия.

*Брюс Тейт,  
Автор книги «Seven Languages  
in Seven Weeks»  
Остин, Техас, май 2012*



# БЛАГОДАРНОСТИ

Чтобы справиться с книгой такого объема и широты тематики, двух авторов недостаточно. Требуются усилия многих одаренных людей с острым взглядом, способных заметить многочисленные ошибки и внести ценный вклад, поведав о тонких деталях технологий.

Мы хотели бы поблагодарить всех, кто жертвовал своим временем и делился опытом (порядок перечисления произвольный): Ян Деес (Ian Dees), Марк Филлипс (Mark Phillips), Ян Ленхардт (Jan Lenhardt), Роберт Стэм (Robert Stam), Олег Бартунов, Дэйв Пэррингтон (Dave Purrington), Дэниэл Бретюа (Daniel Bretoi), Мэтт Адамс (Matt Adams), Шон Копенхэйвер (Sean Copenhaver), Лоран Сэндс-Рэмшоу (Loren Sands-Ramshaw), Эмиль Эйфрем (Emil Eifrem), Андреас Колледжер (Andreas Kollegger).

И, наконец, спасибо Брюсу Тейту (Bruce Tate) за его опыт и рекомендации.

Мы также искренне благодарим весь коллектив издательства Pragmatic Bookshelf – за то, что они взялись за этот дерзкий проект и помогли нам в его реализации. Отдельное спасибо нашему редактору, Джеки Картер (Jackie Carter). Благодаря вашим терпеливым замечаниям книга приобрела свой настоящий вид. Спасибо всем, кто упорно работал, чтобы отшлифовать текст и исправить все наши огрехи.

И не в последнюю очередь мы хотим выразить благодарность Фредерику Дюмону (Frederic Dumont), Мэттью Флауэру (Matthew Flower), Ребекке Скиннер (Rebecca Skinner) и всем нашим требовательным читателям. Если бы не ваше неутолимое желание учиться, у нас не было бы шанса быть вам полезными.

Приносим извинения всем, кого мы забыли упомянуть, – разумеется, не по злому умыслу.

От Эрика. Дорогая Ноэль, ты не просто особенная, ты неповторима, и этим все сказано. Спасибо тебе, что сумела пережить еще одну книгу. Спасибо также создателям баз данных и их добровольным помощникам за то, что у нас есть, о чем писать и тем зарабатывать себе на жизнь.

От Джима. Прежде всего, я обязан поблагодарить свою семью – Рути за безграничное терпение и поддержку, которые согревали мне сердце; Эмму и Джимми – двухмышленных проказников, которые всегда могут рассчитывать на папину любовь. И отдельная благодарность всем невоспетым героям, которые читают IRC-каналы, форумы, списки рассылки и системы регистрации ошибки – всегда готовые помочь тем, кто в этом нуждается. Только благодаря вашей преданности движению за открытость исходного кода эти проекты и могут существовать.



# ВСТУПЛЕНИЕ

Кто-то сравнил данные с сырой нефтью. Раз так, то базы данных – это месторождения, буровые установки, насосы и нефтеочистительные заводы. Данные хранятся в базах и, если вы хотите добраться до них, то должны для начала познакомиться с современным оборудованием.

Базы данных – это инструменты, средства достижения конечного результата. У каждой базы есть своя история и свой взгляд на мир. Чем глубже вы их понимаете, тем лучше оснащены для обуздания мощи, скрытой в постоянно растущем корпусе доступных данных.

## Почему именно семь баз данных?

Еще в марте 2010 года мы захотели написать книгу о NoSQL-технологиях. Тогда этот термин только входил в обиход, о нем много говорили, но общего понимания еще не было. Что на самом деле означает слово *NoSQL*? Какие типы систем включать в эту категорию? Какое влияние новые технологии окажут на практику создания программ? Именно на эти вопросы мы и собирались ответить – как для самих себя, так и для других.

Прочитав книгу Брюса Тейта «Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages»<sup>1</sup> [Tat10], мы поняли, что он нашел правильный подход. Прогрессивная методика введения в языки нашла отклик в наших сердцах. Мы решили, что подобный способ можно применить и к базам данных, и это позволит доходчиво ответить на поставленные вопросы о технологиях NoSQL.

## Что есть в этой книге?

Эта книга адресована опытным разработчикам, которые хотели бы получить общее представление о положении дел в современных тех-

1 Семь языков за семь недель. Прагматическое руководство по изучению языков программирования. *Прим. перев.*

нологиях баз данных. Опыт работы с базами данных приветствуется, хотя и необязателен.

После краткого введения следуют семь глав, посвященных семи базам данных, представляющим пять разных жанров, или стилей, упомянутых во введении: PostgreSQL, Riak, Apache HBase, MongoDB, Apache CouchDB, Neo4J и Redis.

Предполагается, что на проработку каждой главы потребуется полный выходной – три дня. Каждый день завершается упражнениями, в которых развиваются затронутые идеи и темы, а в конце каждой главы приведено резюме, где обсуждаются сильные и слабые стороны базы данных. Вы можете продвигаться быстрее или медленнее – как будет удобнее, но перед тем как продолжить чтение, важно как следует усвоить изложенные концепции. Мы старались подобрать примеры, наглядно высвечивающие характерные особенности каждой базы данных. Чтобы по-настоящему понять, что может предложить та или иная база данных, необходимо некоторое время поработать с ней, то есть засучить рукава и попрактиковаться.

Возможно, у вас возникнет искушение пропустить некоторые главы, однако имейте в виду, что книга рассчитана на последовательное чтение. Некоторые концепции, например распределение-редукция (mapreduce), подробно рассматриваются в начальных главах и лишь бегло – в последующих. Цель книги – дать полное представление о положении дел в современной индустрии баз данных, поэтому мы рекомендуем изучить все главы.

## Чего нет в этой книге?

Прежде чем приступать к чтению книги, вы, наверное, захотите узнать, чего в ней нет.

### ***Это не руководство по установке***

Установка описанных в книге баз данных иногда осуществляется просто, иногда – с некоторыми сложностями, а иногда – откровенно безобразно. Для одних баз имеются готовые пакеты, другие придется компилировать из исходного кода. Кое-где мы будем давать полезные советы, но, вообще говоря, вы предоставлены сами себе. Опустив описание процедуры установки, мы смогли включить больше полезных примеров и пояснений, а ведь именно это вам и нужно, не так ли?

## **Руководство администратора?**

### **Пожалуй, нет**

В этой книге вы не найдете и того, что обычно входит в состав руководства по администрированию. У каждой базы данных бесчисленное множество параметров, флагов и тонкостей настройки; по большей части, все это хорошо документировано в Сети. Нас больше интересует обучение полезным идеям и полное погружение, нежели рутинные повседневные операции. Хотя характеристики работы базы данных могут зависеть от параметров – и в некоторых местах эти характеристики обсуждаются, – мы не собираемся вдаваться в тонкости всех возможных конфигураций. На это просто нет места!

## **Замечание для пользователей Windows**

Эта книга принципиально посвящена разнообразию выбора, преимущественно из различного ПО с открытым исходным кодом на платформах \*nix. Корпорация Microsoft стремится к созданию интегрированных сред, что ограничивает возможность выбора более узким набором предопределенных компонентов. Поэтому базы данных с открытым исходным кодом, которые мы здесь рассматриваем, разрабатываются пользователями (и преимущественно *для* пользователей) \*nix-систем. Это не наше собственное предубеждение, а отражение текущего положения дел. Поэтому предполагается, что все учебные примеры будут запускаться из оболочки \*nix. Если вы работаете в Windows, но всё же хотите поэкспериментировать, рекомендуем настроить среду Cygwin<sup>2</sup>. Можно также запустить виртуальную Linux-машину.

## **Примеры кода и соглашения**

В этой книге встречается код на различных языках. Отчасти это диктуется конкретной рассматриваемой базой данных. Мы старались ограничиться только языками Ruby/JRuby и JavaScript. Мы предпочитаем командные утилиты скриптам, но, когда это нужно для решения поставленной задачи, применяем и другие языки, например PL/pgSQL (Postgres) или Gremlin/Groovy (Neo4J). Мы также рассмотрим программирование некоторых серверных приложений на языке JavaScript с применением Node.js.

---

<sup>2</sup> <http://www.cygwin.com/>

Если явно не оговорено противное, листинги содержат законченный код, который можно выполнить на досуге. В примерах и фрагментах синтаксические конструкции графически выделены в соответствии с правилами языка. Команды оболочки начинаются со знака `$`.

## Сетевые ресурсы

Ценным ресурсом является посвященная этой книге страница на сайте издательства Pragmatic Bookshelf<sup>3</sup>. Там вы найдете весь представленный в книге исходный код, а также средства обратной связи – форум сообщества и форму для отправки сообщений о найденных ошибках, где заодно можно внести предложения по изменениям в будущих изданиях этой книги.

Благодарим всех, кто готов сопровождать нас в путешествии по миру современных баз данных.

*Эрик Редмонд и Джим Р. Уилсон*

---

<sup>3</sup> <http://pragprog.com/book/rwdata/seven-databases-in-seven-weeks>





# ГЛАВА 1.

## Введение

Мы являемся свидетелями поворотного момента в мире баз данных. В течение многих лет реляционная модель была стандартом де факто для решения любых задач – больших и малых. Мы не думаем, что в обозримом будущем реляционные СУБД полностью сойдут со сцены, но многие люди оглядываются по сторонам в поисках альтернатив, в частности: структур данных без схемы, нетрадиционных структур данных, простой репликации, высокой доступности, горизонтального масштабирования и новых способов формулирования запросов. Все эти технологии получили собирательное название *NoSQL*, и именно о них пойдет речь в этой книге.

Мы рассмотрим семь баз данных, представляющих широкий спектр технологий. Вы узнаете о функциональности разных СУБД и принятых при их проектировании компромиссах – долговечность данных или быстродействие, абсолютная согласованность или согласованность в конечном счете и т. д., а также поймете, как принимать оптимальное решение о выборе базы в каждом конкретном случае.

### 1.1. Все начинается с вопроса

Главный вопрос настоящей книги таков: какая база данных или их комбинация решает задачу оптимальным способом? Если после ее прочтения вы будете понимать, как сделать выбор, принимая во внимание конкретные требования и располагаемые ресурсы, то мы будем довольны.

Но чтобы ответить на этот вопрос, надо знать, какие имеются варианты. Для этого мы собираемся глубоко погрузиться в каждую из семи баз данных, показав ее сильные и слабые стороны. Вам предстоит попрактиковаться в операциях CRUD<sup>1</sup>, освежить знания о схемах и найти ответы на следующие вопросы.

---

1 Create, Read, Update, Delete (создание, чтение, обновление, удаление). *Прим. перев.*

- *К какому типу относится это хранилище данных?* Существует много «жанров» баз данных: реляционные, ключи и значения, столбцовые, документо-ориентированные, графовые. Популярные СУБД, в том числе рассматриваемые в этой книге, обычно можно отнести к одной из этих категорий. Вы узнаете о том, для каких задач предназначена каждая категория. Мы специально выбрали базы, покрывающие все категории: одну реляционную (Postgres), два хранилища ключей и значений (Riak, Redis), столбцовую (HBase), две документо-ориентированных (MongoDB, CouchDB) и одну графовую (Neo4J).
- *Что было побудительным мотивом?* Базы данных создаются не в вакууме. Они предназначаются для решения конкретных практических задач. РСУБД появились, когда гибкость запросов считалась важнее, чем гибкость схемы. С другой стороны, столбцовые базы данных прекрасно приспособлены для хранения больших объемов данных на нескольких машинах; связи между данными при этом отходят на второй план. Мы рассмотрим сценарии применения каждой базы данных и соответствующие примеры.
- *Как обращаться к базе?* Существуют различные способы подключения к базе данных. Если предлагается интерактивная командная утилита, то мы начинаем изучение с нее и только потом переходим к другим вариантам. Если требуется программирование, то мы стараемся ограничиться языками Ruby и JavaScript, хотя временами прибегаем и к другим, например PL/pgSQL (Postgres) или Gremlin (Neo4J). На более низком уровне обсуждаются протоколы типа REST (CouchDB, Riak) и Thrift (HBase). В последней главе рассматривается более сложная конфигурация баз данных, объединенных серверным скриптом для Node.js, написанным на JavaScript.
- *В чем состоит уникальность?* Любое хранилище данных поддерживает запись и обратное считывание данных. Все остальное изменяется в широких пределах. Иногда поддерживаются запросы по произвольным полям. Иногда – индексирование для ускорения поиска. Некоторые базы поддерживают произвольные запросы, для других запросы следует планировать заранее. Схема может быть жесткой и контролироваться СУБД, а может быть просто набором рекомендаций, которые переосматриваются по желанию. Понимание возможностей и огра-

ничений станет существенным подспорьем при выборе СУБД, которая в наибольшей степени отвечает решаемой задаче.

- *Как обстоит дело с производительностью?* Как функционирует база данных и во что это обходится? Поддерживается ли сегментирование? Как насчет репликации? Равномерно ли распределены данные за счет хороших функций хеширования или все данные оказываются в одном узле? Для чего оптимизирована база: для чтения, для записи или для какой-то другой операции? Есть ли какие-нибудь средства настройки и насколько они развиты?
- *Как масштабируется база данных?* Масштабируемость тесно связана с производительностью. Говорить о масштабируемости вне контекста – не указывая, что именно вы хотите *масштабировать*, в общем случае бессмысленно. В этой книге мы расскажем, как задавать правильные вопросы, устанавливающие такой контекст. Вопрос о том, *как* масштабируются разные базы, намеренно не акцентируется, но мы тем не менее поясним, к какому виду масштабирования наиболее приспособлена та или иная база: горизонтальному (MongoDB, HBase, Riak), традиционному вертикальному (Postgres, Neo4J, Redis) или чему-то среднему.

Наша цель заключается не в том, чтобы довести начинающего до уровня эксперта по всем рассматриваемым СУБД. Для детального рассмотрения любой из них потребовалась бы целая книга – и не одна (и такие книги есть). Однако к концу этой книги вы будете ясно представлять себе сильные стороны каждой базы данных и понимать, чем они отличаются друг от друга.

## 1.2. Жанры

Как и музыкальные произведения, базы данных можно отнести к одному или нескольким стилям. В отдельно взятой композиции используются те же самые ноты, что и в любой другой, но предназначены они для разных слушателей. Мало кто станет слушать мессу симинор Баха в открытом кабриолете, мчащемся по скоростной трассе 405 в Южной Калифорнии. Вот так и с базами данных – одни лучше приспособлены для решения одних задач, другие – других. Поэтому задавать следует не вопрос «Можно ли использовать эту СУБД для хранения и повышения качества данных?», а вопрос «Нужно ли это делать?».

В этом разделе мы изучим пять основных жанров баз данных, а также скажем, какие базы будем использовать для иллюстрации каждого жанра.

Важно помнить, что большая часть задач, с которыми вы сталкиваетесь на практике, может быть решена с помощью большинства, если не всех, рассматриваемых в этой книге, а также многих других СУБД. Вопрос не столько в том, можно ли приспособить СУБД конкретного стиля к моделированию данных, сколько в том, насколько эффективно будет ее применение в данной предметной области при данных сценариях использования и располагаемых ресурсах. Вы овладеете искусством предсказывать, насколько полезной окажется для вас конкретная база данных.

## **Реляционные СУБД**

Реляционная модель первой приходит на ум большинству разработчиков с опытом в области баз данных. Реляционные системы управления базами данных (РСУБД) основаны на теории множеств, в основе их реализации лежат двумерные таблицы, состоящие из строк и столбцов. Канонический способ взаимодействия с РСУБД – написание запросов на языке Structured Query Language (SQL). Значения данных типизированы, это могут быть числа, строки, даты, неструктурированные двоичные объекты (BLOB) и т. п. Тип данных контролируется системой. Существенно, что благодаря математическим основаниям реляционной модели (теории множеств) исходные таблицы можно соединять и трансформировать в новые, более сложные.

Существует немало реляционных СУБД с открытым исходным кодом – MySQL, H2, HSQLDB, SQLite и многие другие, так что выбирать есть из чего. В этой книге мы остановимся на СУБД PostgreSQL (глава 2).

## **PostgreSQL**

Испытанная в боях СУБД PostgreSQL – самая старая и надежная из всех рассматриваемых в этой книге. Совместимая со стандартом SQL, она покажется знакомой любому, кто раньше работал с реляционными базами данных, и послужит эталоном для сравнения с другими базами. Мы также рассмотрим некоторые малоизвестные средства SQL и уникальные достоинства Postgres. Здесь каждый – от новичка до эксперта – найдет себе что-то по душе.

Конец ознакомительного фрагмента.  
Приобрести книгу можно  
в интернет-магазине  
«Электронный универс»  
[e-Univers.ru](http://e-Univers.ru)