

Благодарность от редакции

Издательство благодарит Ивана Евгеньевича Панченко за помощь в подготовке этого издания.

Эта книга – хорошее пособие для близкого знакомства с PostgreSQL. С той или иной степенью полноты она охватывает практически все вопросы, с которыми встречается разработчик и администратор, начинающий профессионально работать с этой СУБД. А то, что в книге не нашлось, есть в документации на русском языке: <https://postgrespro.ru/docs>.

Приятного погружения в Postgres!

Содержание

Благодарность от редакции	5
Об авторах	13
О рецензентах	15
Предисловие	16
Глава 1. Реляционные базы данных	22
Системы управления базами данных	22
Историческая справка	22
Категории баз данных	23
Базы данных NoSQL	23
Реляционные и объектно-реляционные базы данных	25
Свойства ACID	26
Язык SQL	26
Понятия реляционной модели	27
Реляционная алгебра	33
Операции выборки и проекции	34
Операция переименования	36
Теоретико-множественные операции	36
Операция декартова произведения	36
Моделирование данных	38
Виды моделей данных	38
Модель сущность-связь	39
UML-диаграммы классов	44
Резюме	44
Глава 2. PostgreSQL в действии	46
Обзор PostgreSQL	46
История PostgreSQL	46
Преимущества PostgreSQL	47
Применения PostgreSQL	48
Истории успеха	49
Ответвления	49
Архитектура PostgreSQL	50
Сообщество PostgreSQL	52
Возможности PostgreSQL	52

Репликация	52
Безопасность.....	53
Расширения.....	54
Возможности NoSQL.....	55
Адаптеры внешних данных	56
Производительность	57
Установка PostgreSQL	58
Установка PostgreSQL с помощью менеджера пакетов APT	59
Установка PostgreSQL в Windows.....	63
Клиенты PostgreSQL	64
Резюме.....	69
Глава 3. Основные строительные блоки PostgreSQL.....	71
Кодирование базы данных.....	71
Соглашение об именовании объектов базы данных.....	71
Идентификаторы в PostgreSQL.....	72
Документация.....	73
Система управления версиями.....	73
Средство миграции базы данных	74
Иерархия объектов в PostgreSQL	74
Шаблонные базы данных	74
Пользовательские базы данных	75
Роли	76
Табличное пространство.....	77
Шаблонные процедурные языки.....	78
Параметры	78
Взаимодействия с объектами PostgreSQL верхнего уровня	80
Компоненты базы данных PostgreSQL.....	81
Схема	81
Применение схем	82
Таблица.....	83
Встроенные типы данных	84
База данных сайта торговли автомобилями	91
Резюме.....	94
Глава 4. Дополнительные строительные блоки PostgreSQL	96
Представления	96
Синтаксис определения представления	98
Категории представлений.....	99
Материализованные представления.....	99
Обновляемые представления	100
Индексы.....	102
Синтаксис создания индекса	103

Избирательность индекса	103
Типы индексов	106
Категории индексов	106
Рекомендации по работе с индексами	108
Функции	109
Встроенные языки программирования PostgreSQL	110
Создание функции на языке C	110
Применение функций	112
Зависимости между функциями	112
Категории функций в PostgreSQL	113
Анонимные функции в PostgreSQL	114
Пользовательские типы данных	114
Триггеры и правила	118
Правила	118
Триггеры	120
Резюме	127
Глава 5. Язык SQL	129
Основы SQL	129
Лексическая структура SQL	131
Запрос данных командой SELECT	134
Структура запроса SELECT	134
Список выборки	136
Фраза FROM	142
Фраза WHERE	148
Группировка и агрегирование	152
Упорядочение и ограничение количества результатов	155
Подзапросы	156
Теоретико-множественные операции – UNION, EXCEPT, INTERSECT	158
Значения NULL	159
Изменение данных в базе	162
Команда INSERT	162
Команда UPDATE	164
Команда DELETE	166
Команда TRUNCATE	167
Резюме	167
Глава 6. Дополнительные сведения о написании запросов	168
Общие табличные выражения	168
СТЕ как средство повторного использования SQL-кода	170
Рекурсивные и иерархические запросы	172
Изменение данных сразу в нескольких таблицах	176
Оконные функции	178

Определение окна	179
Фраза WINDOW	180
Использование оконных функций	181
Оконные функции с группировкой и агрегированием	183
Продвинутые методы работы с SQL	184
Выборка первых записей	184
Извлечение выборочных данных	185
Функции, возвращающие множества	186
Латеральные подзапросы	189
Дополнительные средства группировки	191
Дополнительные виды агрегирования	193
Резюме	195
Глава 7. Серверное программирование на PL/pgSQL	196
Сравнение языков SQL и PL/pgSQL	196
Параметры функций в PostgreSQL	197
Параметры функций, относящиеся к авторизации	197
Параметры функции, относящиеся к планировщику	199
Параметры функции, относящиеся к конфигурации	202
Команды управления в PL/pgSQL	203
Объявления	203
Присваивание	205
Условные команды	207
Итерирование	209
Возврат из функции	212
Предопределенные переменные в функциях	215
Обработка исключений	216
Динамический SQL	218
Динамическое выполнение команд DDL	218
Динамическое выполнение команд DML	219
Динамический SQL и кеширование	220
Рекомендации по использованию динамического SQL	220
Резюме	222
Глава 8. OLAP и хранилища данных	223
Оперативная аналитическая обработка	224
Извлечение, преобразование и загрузка	225
Моделирование данных для OLAP	228
Агрегирование	230
Секционирование	231
Параллельные запросы	235
Просмотр только индексов	236
Резюме	238

Глава 9. За пределами традиционных типов данных	239
Массивы.....	240
Функции и операторы массивов	243
Доступ к элементам массива и их модификация.....	244
Индексирование массивов.....	245
Хранилище ключей и значений	246
Индексирование hstore	248
Структура данных JSON.....	249
JSON и XML.....	249
Типы данных JSON в PostgreSQL	250
Доступ к объектам типа JSON и их модификация.....	250
Индексирование JSON-документов	252
Реализация REST-совместимого интерфейса к PostgreSQL	253
Полнотекстовый поиск в PostgreSQL	257
Типы данных tsquery и tsvector	257
Сопоставление с образцом	258
Полнотекстовые индексы	260
Резюме	261
Глава 10. Транзакции и управление параллельным доступом	262
Транзакции	262
Транзакции и свойства ACID	263
Транзакции и конкурентность	264
Уровни изоляции транзакций	267
Явная блокировка	272
Блокировка на уровне таблиц.....	273
Блокировка на уровне строк.....	276
Взаимоблокировки	277
Рекомендательные блокировки	278
Резюме	279
Глава 11. Безопасность в PostgreSQL	281
Аутентификация в PostgreSQL.....	281
Файл pg_hba.conf.....	283
Прслушиваемые адреса.....	284
Рекомендации по аутентификации	284
Привилегии доступа по умолчанию.....	285
Система ролей и прокси-аутентификация	286
Уровни безопасности в PostgreSQL	288
Безопасность на уровне базы данных.....	288
Безопасность на уровне схемы	289

Безопасность на уровне таблицы	289
Безопасность на уровне столбца	290
Безопасность на уровне строк	290
Шифрование данных	293
Шифрование паролей ролей в PostgreSQL.....	293
Расширение pgcrypto.....	293
Резюме	297
Глава 12. Каталог PostgreSQL.....	298
Системный каталог.....	298
Системный каталог для администраторов	301
Получение версии кластера баз данных и клиентских программ	301
Завершение и отмена пользовательского сеанса	301
Задание и получение параметров кластера баз данных.....	302
Получение размера базы данных и объекта базы данных	304
Очистка базы данных	305
Очистка данных в базе	308
Оптимизация производительности	310
Избирательная выгрузка	311
Резюме	314
Глава 13. Оптимизация производительности базы данных.....	315
Настройка конфигурационных параметров PostgreSQL	316
Максимальное количество подключений.....	316
Параметры памяти	316
Параметры жесткого диска.....	317
Параметры планировщика	317
Эталонное тестирование вам в помощь	318
Оптимизация производительности записи	318
Оптимизация производительности чтения	321
План выполнения и команда EXPLAIN	322
Обнаружение проблем в планах выполнения запросов	326
Типичные ошибки при написании запросов	329
Избыточные операции	329
Индексы отсутствуют или построены не так.....	329
Использование CTE без необходимости	333
Использование процедурного языка PL/pgSQL.....	333
Межстолбцовая корреляция	334
Секционирование таблиц	336
Недостатки механизма исключения в силу ограничений.....	336
Переписывание запросов.....	337
Резюме	338

Глава 14. Тестирование	339
Автономное тестирование	339
Специфика автономного тестирования в базе данных	340
Фреймворки юнит-тестирования	343
Различие схем	345
Интерфейсы абстрагирования базы данных	346
Отличия в данных	347
Тестирование производительности	350
Резюме	352
Глава 15. PostgreSQL в приложениях на Python	353
Python DB API 2.0	354
Низкоуровневый доступ к базе данных с помощью psycopg2	355
Соединение с базой данных	357
Пул соединений	358
Выполнение SQL-команд	359
Чтение данных из базы	361
Команда COPY	361
Асинхронный доступ	362
Альтернативные драйверы для PostgreSQL	363
pg8000	363
asyncpg	364
SQLAlchemy – библиотека объектно-реляционного отображения	366
Основные компоненты SQLAlchemy	367
Подключение к базе и выборка данных с помощью языка SQL Expression	367
ORM	369
Резюме	373
Глава 16. Масштабируемость	374
Проблема масштабируемости и теорема CAP	375
Репликация данных в PostgreSQL	377
Журнал транзакций	377
Физическая репликация	378
Логическая репликация	384
Применение репликации для масштабирования PostgreSQL	387
Масштабирование на большое количество запросов	388
Разделение данных	389
Масштабирование с ростом числа подключений	391
Резюме	392
Предметный указатель	394

Об авторах

Салахалдин Джуба более десяти лет работает в промышленности и академических учреждениях, уделяя основное внимание разработкам приложений баз данных в крупномасштабных и корпоративных системах. Имеет степень магистра по управлению природоохранной деятельностью, а также степень бакалавра по проектированию компьютерных систем. Является сертифицированным разработчиком программных решений на основе продуктов Microsoft (MCSD).

Работал преимущественно с базами данных SQL Server, PostgreSQL и Greenplum. Разрабатывал для научных сообществ приложения, связанные с распределенной обработкой географической информации, во время работы в академических учреждениях участвовал во многих международных проектах и разработке стандартов, имеющих отношение к обработке изображений.

Работая программистом, занимался в основном определением ETL-процессов обработки данных, полученных извне, постановкой программных задач, популяризацией передовых практик работы с SQL, проектированием приложений OLTP и OLAP, исследованием и оценкой новых технологий, преподавательской деятельностью и консультированием.

Выражаю глубочайшую благодарность своему коллеге Андрею Волкову, без которого эта книга не состоялась бы. Также благодарю всех, кто оказывал поддержку, а особенно коллектив издательства Packt за советы, корректуру и замечания по оформлению книги. Хочу также поблагодарить свою семью, которая не оставляла меня заботами, хотя я был вынужден уделять много внимания книге в ущерб близким. И наконец, самые теплые слова и глубокая признательность моему покойному отцу, Икраему Джубе, который всегда поддерживал меня, помогал и наставлял в жизни.

Андрей Волков изучал банковские информационные системы. Свою карьеру начал финансовым аналитиком в коммерческом банке. Базы данных были его основным рабочим инструментом, и скоро он понял, что прямые запросы к базе данных и мастерское владение SQL гораздо эффективнее визуальных отчетов, когда нужно произвести ситуативный анализ. Он перешел в группу хранилищ данных и спустя некоторое время возглавил ее, заняв должность архитектора хранилищ данных.

Работал в основном с Oracle и занимался разработкой логических и физических моделей финансовых и бухгалтерских данных, реализацией их с помощью СУБД, разработкой ETL-процессов и аналитикой. Отвечал за обучение пользователей работе с хранилищем данных и инструментами бизнес-аналитики. Преподавание SQL также входило в его обязанности.

Проработав 10 лет в финансовом секторе, он сменил поле деятельности и теперь работает старшим разработчиком баз данных в телекоммуникационной компании. Здесь он имеет дело в основном с СУБД PostgreSQL и отвечает за моделирование данных и реализацию физических структур, разработку хранимых процедур, интеграцию баз данных с другими программными компонентами и разработку хранилища данных.

Имея большой опыт работы с Oracle и PostgreSQL – ведущей коммерческой РСУБД и одной из самых технически продвинутых РСУБД с открытым исходным кодом, – он может сравнивать их и понимает, в чем преимущества той и другой. Опыт реализации различных типов приложений баз данных, а также работы в роли бизнес-аналитика, использующего базы данных как инструмент, научил его понимать, как лучше применять средства СУБД в различных ситуациях. Накопленным опытом он с удовольствием делится в этой книге.

Я благодарен жене и сыну, которые поддерживали меня и давали возможность работать по вечерам и в выходные. Огромное спасибо редакторам за советы и помощь в организации материала. Но больше всего я благодарен основному автору, Салахалдину Джубе, который предложил мне принять участие в работе над книгой, включил в команду и, по сути дела, сделал большую часть работы.

О рецензентах

Д-р Изабель М. Д. Роза – обладатель стипендии имени Марии Склодовской-Кюри, с мая 2016 г. работает в Немецком исследовательском центре интегративного биоразнообразия (iDiv). Родилась в Лиссабоне, Португалия, в 1986 году. Получила степень бакалавра по лесотехнике (2007) и степень магистра по управлению природными ресурсами (2009) в Лиссабонском университете, а также степень доктора по вычислительной экологии (2013) в Имперском колледже Лондона. Является руководителем исследовательского проекта «Применение моделей изменения растительного покрова для решения важных вопросов охраны природы», финансируемого в рамках гранта H2020-MSCA-IF-2015. С 2013 года принимала участие в двух международных проектах, включая финансируемый ЕС проект с бюджетом 1,5 миллиона евро (Terragenesis, ERC-2011-StG_20101109). Автор 15 публикаций в рецензируемых научных журналах, в т. ч. *Nature Ecology and Evolution*, *Current Biology* и *Global Change Biology*, на которые имеется 252 ссылки (Google citations, октябрь 2017), индекс Хирша равен 8. За время работы в академических учреждениях приобрела ряд навыков, в т. ч. в области статистического анализа, программирования (на языках R, C++ и Python), работы с геоинформационными системами (ArcGIS и QGIS) и создания баз данных (PostgreSQL/PostGIS и SQLServer). Рецензировала книгу «Learning PostgreSQL», также выпущенную издательством Packt.

Шелдон Штраух – ветеран с 23-летним опытом консультирования таких компаний, как IBM, Sears, Ernst & Young, Kraft Foods. Имеет степень бакалавра по организации управления, применяет свои знания, чтобы помочь компаниям самоопределиться. В сферу его интересов входят сбор, управление и глубокий анализ данных, карты и их построение, бизнес-аналитика и применение анализа данных в целях непрерывного улучшения. В настоящее время занимается разработкой сквозного управления данными и добычей данных в компании Enova, оказывающей финансовые услуги и расположенной в Чикаго. В свободное время увлекается искусством, особенно музыкой, и путешествует со своей женой Мэрилин.

Предисловие

Выбор правильной системы управления базами данных – трудная задача из-за большого количества предложений на рынке. В зависимости от бизнес-модели можно выбрать коммерческую СУБД или базу данных с открытым исходным кодом и коммерческой поддержкой. Следует также принимать во внимание ряд технических и нетехнических факторов. Когда речь заходит о реляционной СУБД, PostgreSQL оказывается на вершине рейтинга по нескольким причинам. Лозунг PostgreSQL – *самая передовая база данных с открытым исходным кодом* – отражает развитость технических средств и уверенность сообщества.

PostgreSQL – объектно-реляционная система управления базами данных с открытым исходным кодом. К ее сильным сторонам относится расширяемость, она успешно конкурирует с основными реляционными СУБД: Oracle, SQL Server и MySQL. Благодаря разнообразию расширений и открытой лицензии PostgreSQL часто применяется в исследовательских проектах, но ее код также лежит в основе многих открытых и коммерческих СУБД, например Greenplum и Vertica. К тому же стартапы часто отдают предпочтение PostgreSQL в силу условий лицензии и изобилия компаний, оказывающих коммерческую поддержку.

Существуют версии PostgreSQL для большинства современных операционных систем, включая Windows, Mac и различные дистрибутивы Linux. Имеется также несколько расширений для доступа к данным, управления и мониторинга работы кластеров PostgreSQL, например pgAdmin, OmniDB и psql. Установка и настройки PostgreSQL достаточно просты и поддерживаются большинством диспетчеров пакетов, в т. ч. yum и apt. Разработчики баз данных не испытывают трудностей в освоении PostgreSQL, поскольку она совместима со стандартами ANSI SQL. Да и помимо стандартов существует масса ресурсов, которые помогут изучить PostgreSQL, – СУБД отлично документирована и может похвастаться очень активным и хорошо организованным сообществом.

PostgreSQL пригодна как для OLTP, так и для OLAP-приложений. Она совместима с ACID-транзакциями и для использования в OLTP-приложениях не нуждается ни в каких дополнениях. Что касается OLAP-приложений, то PostgreSQL поддерживает оконные функции, адаптеры внешних данных (FDW – Foreign Data Wrapper) и наследование таблиц; кроме того, существует немало внешних расширений для этой цели.

Несмотря на совместимость с ACID-транзакциями, PostgreSQL демонстрирует отличную производительность, поскольку в ней применены самые современные алгоритмы и методы. Например, в PostgreSQL используется архитектура MVCC (MultiVersion Concurrency Control – управление параллельным доступом с помощью многоверсионности) для обеспечения параллельного доступа к данным. Вдобавок PostgreSQL поддерживает как пессимистическую,

так и оптимистическую конкурентность, а поведение механизма блокировок можно изменять в зависимости от ситуации. Кроме того, в PostgreSQL имеется великолепный анализатор и такие передовые средства, как секционирование данных с помощью наследования таблиц и исключение в силу ограничений, позволяющие ускорить обработку очень больших объемов данных. PostgreSQL поддерживает несколько типов индексов, в т. ч. **B-Tree**, **GiN**, **GiST** и **BRIN**. А начиная с версии PostgreSQL 9.6 поддерживается параллельное выполнение запросов. Наконец, репликация позволяет балансировать нагрузку на различные узлы кластера.

PostgreSQL допускает масштабирование благодаря многочисленным представленным на рынке методам репликации, например Slony и pgpool-II. Дополнительно PostgreSQL изначально поддерживает синхронную и асинхронную потоковую репликацию, а также логическую репликацию. Это делает PostgreSQL чрезвычайно привлекательным решением, поскольку ее можно использовать для создания высокодоступных и высокопроизводительных систем.

КРАТКОЕ СОДЕРЖАНИЕ КНИГИ

Глава 1 «Реляционные базы данных» содержит введение в концепции реляционных систем управления базами данных, в т. ч. реляционную алгебру и моделирование данных. Здесь же описываются различные типы СУБД: графовые, документные, столбцовые, а также хранилища ключей и значений.

В главе 2 «PostgreSQL в действии» мы расскажем, как установить сервер и клиенты PostgreSQL на различных платформах. Также мы познакомимся с некоторыми возможностями PostgreSQL, в т. ч. встроенной поддержкой репликации и очень богатым набором типов данных.

В главе 3 «Основные строительные блоки PostgreSQL» описываются рекомендации по кодированию, в т. ч. принятые соглашения об идентификаторах. Здесь рассматриваются основные структурные элементы и взаимодействие между ними: шаблонные базы данных, пользовательские базы данных, табличные пространства, роли и настройки. Описываются также основные типы данных и таблицы.

Глава 4 «Дополнительные строительные блоки PostgreSQL» посвящена представлениям, индексам, функциям, пользовательским типам данных, триггерам и правилам. Рассматриваются различные применения этих элементов и сравниваются различные элементы, применимые для решения задачи, например правила и триггеры.

Глава 5 «Язык SQL» – введение в структурированный язык запросов (SQL), используемый для взаимодействия с базой данных: создания и обслуживания структур данных, а также для ввода данных в базу, их изменения, выборки и удаления. В SQL есть команды, относящиеся к трем подязыкам: **языку определения данных (DDL)**, **языку манипулирования данными (DML)** и **языку управления данными (DCL)**. В этой главе описаны четыре команды

SQL, составляющие основу языка DML. Особое внимание уделено команде SELECT, на примере которой объясняются концепции группировки и фильтрации для демонстрации того, что такое выражения и условия SQL и как используются подзапросы. Здесь же рассматриваются некоторые вопросы реляционной алгебры в применении к соединению таблиц.

В главе 6 «Дополнительные сведения о написании запросов» описаны такие средства SQL, как общие табличные выражения и оконные функции. Они позволяют реализовать вещи, которые без них были бы невозможны, например рекурсивные запросы. Рассматриваются и другие конструкции, как то: фразы DISTINCT ON и FILTER, а также латеральные подзапросы. Без них, в принципе, можно обойтись, но с их помощью запросы получаются компактнее, проще и быстрее.

В главе 7 «Серверное программирование на PL/pgSQL» рассматриваются параметры функций, в т. ч. количество возвращенных строк и стоимость функции, которые нужны главным образом планировщику запросов. Здесь же представлены управляющие конструкции, например условные команды и циклы. Наконец, объясняется, что такое динамический SQL и как им лучше пользоваться.

В главе 8 «OLAP и хранилища данных» речь пойдет о применении реляционных баз данных для аналитической обработки. Обсуждаются различия между двумя типами рабочей нагрузки: OLTP и OLAP, а также вопросы моделирования в OLAP-приложениях. Кроме того, рассматриваются некоторые технические приемы выполнения ETL (extract, transform, load – извлечение, преобразование, загрузка), в т. ч. команда COPY. Также описываются средства PostgreSQL, предназначенные для повышения скорости выборки, в частности просмотр только индексов и секционирование таблиц.

Глава 9 «За пределами традиционных типов данных» посвящена некоторым нестандартным типам: массивам, хешам, JSON-документам и полнотекстовому поиску. Описываются операции и функции для каждого типа данных: инициализация, обновление, доступ и удаление. Наконец, продемонстрировано, как объединить PostgreSQL и Nginx для обслуживания REST-совместимых запросов чтения.

В главе 10 «Транзакции и управление параллельным доступом» подробно обсуждаются свойства ACID и их связь с управлением параллельным доступом. Здесь же рассмотрены уровни изоляции и их побочные эффекты – с демонстрацией этих эффектов на примерах SQL. Также уделено внимание различным методам блокировки, в т. ч. стратегиям пессимистической блокировки: блокировке на уровне строк и рекомендательным блокировкам.

Глава 11 «Безопасность в PostgreSQL» посвящена аутентификации и авторизации. Описываются методы аутентификации в PostgreSQL и объясняется структура конфигурационного файла аутентификации по адресу узла. Также обсуждаются разрешения на доступ к таким объектам базы данных, как схемы, таблицы, представления, индексы и столбцы. Наконец, описано, как защитить секретные данные, в т. ч. пароли, с помощью одностороннего и двустороннего шифрования.

В главе 12 «Каталоги PostgreSQL» приведено несколько рецептов обслуживания кластера базы данных, в т. ч. очистка данных, обслуживание пользовательских процессов, удаление индексов и неиспользуемых объектов базы данных, определение и добавление индексов по внешним ключам и т. д.

В главе 13 «Оптимизация производительности базы данных» обсуждается несколько подходов к оптимизации производительности. Описаны конфигурационные параметры кластера PostgreSQL, используемые для настройки производительности кластера в целом. Также разобраны типичные ошибки при написании запросов и предложено несколько способов повышения производительности, включая построение индексов, секционирование таблиц и исключение в силу ограничений.

В главе 14 «Тестирование» освещены некоторые аспекты процесса тестирования программного обеспечения и его особенности применительно к базам данных. Автономные тесты для баз данных можно писать в виде SQL-скриптов или хранимых функций. Существует несколько каркасов для написания автономных тестов и обработки результатов тестирования.

В главе 15 «PostgreSQL в приложениях на Python» обсуждаются различные продвинутые механизмы, в т. ч. организация пулов соединений, асинхронный доступ и **объектно-реляционное отображение (ORM)**. Приведен пример, демонстрирующий подключение к базе данных, отправку запроса и выполнение обновления из программы на языке Python. Наконец, дается введение в различные технологии взаимодействия с PostgreSQL, чтобы разработчик имел полную картину современного состояния дел.

В главе 16 «Масштабируемость» рассматриваются проблема масштабируемости и теорема CAP. Здесь же обсуждается репликация в PostgreSQL – физическая и логическая. Описаны различные сценарии масштабирования и их реализация в PostgreSQL.

Что необходимо для чтения книги

Вообще говоря, для сервера и клиентов PostgreSQL не нужно какое-то особое оборудование. PostgreSQL устанавливается на все современные платформы, включая Linux, Windows и Mac. Если необходима некая конкретная библиотека, то мы приводим инструкции по ее установке.

Вам понадобится версия PostgreSQL 10; впрочем, большинство примеров будет работать и с предыдущими версиями. Для выполнения примеров кода и скриптов необходим какой-нибудь клиент PostgreSQL на вашей машине, предпочтительно psql, и доступ к удаленному серверу PostgreSQL.

В среде Windows команда cmd.exe не очень удобна, лучше бы поставить Cygwin (<http://www.cygwin.com/>) или еще какую-нибудь альтернативу, например PowerShell.

При чтении некоторых глав понадобится дополнительное ПО. Например, в главе 15 нужно будет установить Python и библиотеки, необходимые для работы с PostgreSQL. А в главе 16 удобнее всего будет работать с Docker.

И еще – в главах 9 и 11 рекомендуется использовать Linux, поскольку работать с некоторыми программами, например Nginx и GnuPG, в Windows не очень комфортно. Для запуска Linux на машине под управлением Windows можете воспользоваться программой Virtual Box (<https://www.virtualbox.org/>).

НА КОГО РАССЧИТАНА ЭТА КНИГА

Если вам интересно узнать о PostgreSQL – одной из самых популярных в мире реляционных баз данных, – то вы попали по адресу. Те, кого интересует создание добротных приложений для работы с базой данных или хранилищем данных, тоже найдут в книге много полезного. Предварительного опыта программирования или администрирования баз данных для чтения книги не требуется.

ГРАФИЧЕСКИЕ ВЫДЕЛЕНИЯ

В этой книге тип информации обозначается шрифтом. Ниже приведено несколько примеров с пояснениями.

Фрагменты кода внутри абзаца, имена таблиц базы данных, папок и файлов, данные, которые вводит пользователь, и адреса в Твиттере выделяются следующим образом: «В следующих строчках читается ссылка и результат передается функции open».

Кусок кода выглядит так:

```
fin = open('data/fake_weather_data.csv', 'r', newline='')
reader = csv.reader(fin)
for row in reader:
    myData.append(row)
```

Входная и выходная информация командных утилит выглядит так:

```
$ mongoimport --file fake_weather_data.csv
```

Новые термины и важные фрагменты выделяются полужирным шрифтом. Например, элементы графического интерфейса в меню или диалоговых окнах выглядят в книге так: «Чтобы загрузить новые модули, выберите пункт меню **Files | Settings | Project Name | Project Interpreter**».



Предупреждения и важные примечания выглядят так.



Советы и рекомендации выглядят так.

ОТЗЫВЫ И ПОЖЕЛАНИЯ

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв прямо на нашем сайте www.dmkpress.com, зайдя на страницу книги, и оставить комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com, при этом напишите название книги в теме письма.

Если есть тема, в которой вы квалифицированы, и вы заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

СКАЧИВАНИЕ ИСХОДНОГО КОДА ПРИМЕРОВ

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте www.dmkpress.com на странице с описанием соответствующей книги.

СПИСОК ОПЕЧАТОК

Хотя мы приняли все возможные меры для того, чтобы удостовериться в качестве наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг — возможно, ошибку в тексте или в коде, — мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от расстройств и поможете нам улучшить последующие версии данной книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

НАРУШЕНИЕ АВТОРСКИХ ПРАВ

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Packt очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконно выполненной копией любой нашей книги, пожалуйста, сообщите нам адрес копии или веб-сайта, чтобы мы могли применить санкции.

Пожалуйста, свяжитесь с нами по адресу электронной почты dmkpress@gmail.com со ссылкой на подозрительные материалы.

Мы высоко ценим любую помощь по защите наших авторов, помогающую нам предоставлять вам качественные материалы.

Глава 1

Реляционные базы данных

Эта и следующая главы содержат общий обзор вопросов разработки приложений баз данных, в т. ч. теоретические аспекты реляционных баз данных. Знакомство с теорией поможет не только создавать качественные проекты, но и овладеть тонкостями работы с реляционными базами.

Эта глава касается не только PostgreSQL, но реляционных баз вообще. Будут рассмотрены следующие вопросы:

- **системы управления базами данных:** понимание классификации баз данных дает разработчику возможность использовать лучшее из имеющегося для решения конкретной задачи;
- **реляционная алгебра:** знакомство с реляционной алгеброй позволит овладеть языком SQL, особенно техникой переписывания запросов;
- **моделирование данных:** применяя методы моделирования, вы сможете лучше донести свои идеи до коллег.

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Разные системы управления базами данных (СУБД) поддерживают различные сценарии и требования. У СУБД долгая история. Сначала мы сделаем краткий обзор недавней истории, а затем расскажем о преобладающих на рынке категориях СУБД.

Историческая справка

Термином «база данных» обозначается много разных понятий. Более того, он наводит на мысли о других терминах: данные, информация, структура данных и управление. Базу данных можно определить как набор, или репозиторий данных, обладающий определенной структурой и управляемый **системой управления базами данных (СУБД)**. Данные могут быть структурированными таблицами, слабоструктурированными XML-документами или вовсе не иметь структуры, описываемой какой-то заранее определенной моделью.

На заре развития базы данных были ориентированы в основном на поддержку приложений в сфере бизнеса; это положило начало математически

точно определенной реляционной алгебре и реляционным системам баз данных. Кроме того, во многих отраслях бизнеса, а равно в научных приложениях используются массивы, изображения и пространственные данные, поэтому поддерживаются также новые модели, включая растровые изображения, карты и алгебраические операции с массивами. Графовые базы данных поддерживают запросы к графам, например о поиске кратчайшего пути между двумя вершинами. Они также обеспечивают простой обход графа.

С появлением веб-приложений, в частности социальных порталов, возникла необходимость поддерживать огромное количество запросов распределенным образом. Это привело к новой парадигме в области баз данных, получившей название **NoSQL (Not Only SQL)**, с другими требованиями, например: примат производительности над отказоустойчивостью и возможность горизонтального масштабирования. В целом на эволюцию баз данных оказывали влияние разнообразные факторы, в том числе:

- **функциональные требования:** сама природа приложений, в которых используются базы данных, потребовала разработки расширений реляционных баз, таких как PostGIS (пространственные данные), или даже специализированных СУБД типа SciDB (для анализа научных данных);
- **нефункциональные требования:** успех языков объектно-ориентированного программирования породил новые тенденции, в частности объектно-ориентированные базы данных. Появились объектно-реляционные системы управления базами данных, перебрасывающие мост между реляционными базами и объектно-ориентированными языками. Взрывной рост объема данных и необходимость обрабатывать терабайты данных на стандартном оборудовании привели к появлению столбцовых баз данных, которые легко масштабируются по горизонтали.

Категории баз данных

Рождались и уходили в небытие различные модели баз данных, включая сетевую и иерархическую модель. Сейчас на рынке преобладают реляционные, объектно-реляционные и NoSQL базы данных. Не следует считать базы данных NoSQL и SQL соперниками – они дополняют друг друга. Применяя ту или иную систему баз данных, можно преодолеть ограничения технологий и выбрать оптимальный вариант.

Базы данных NoSQL

На базы данных NoSQL оказывает влияние теорема CAP, известная также как теорема Брюера. В 2002 году С. Джильберт и Н. Линч опубликовали ее формальное доказательство в статье «Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services». В 2009 году зародилось движение NoSQL. Сегодня насчитывается свыше 150 баз NoSQL (nosql-database.org).

Теорема CAP

Теорема CAP утверждает, что распределенная вычислительная система не может одновременно обеспечить все три следующих свойства:

- **согласованность** – все клиенты сразу же видят последние данные после каждого обновления;
- **доступность** – любой клиент может найти копию нужных данных даже в случае отказа узла. Иными словами, даже после выхода из строя части системы клиенты все равно могут обратиться к данным;
- **устойчивость к разделению** – система продолжает работать даже в случае утраты любого сообщения или отказа части системы.

Характер системы определяется тем, от какого свойства мы отказываемся. Например, пожертвовав согласованностью, мы получим простую, масштабируемую, высокопроизводительную систему управления базами данных. Зачастую основное различие между реляционной и NoSQL базой данных как раз и состоит в согласованности. В реляционной базе гарантируется выполнение свойств **ACID** (**атомарность, согласованность, изолированность, долговечность**). Напротив, во многих базах данных NoSQL принята модель **BASE** (basically available soft-state, eventual consistency – **базовая доступность, неустойчивое состояние, согласованность в конечном счете**).

Мотивация NoSQL

База данных NoSQL предоставляет средства для хранения нереляционных данных, их выборки и манипулирования ими. Такие базы являются распределенными, горизонтально масштабируемыми и, как правило, поставляются с открытым исходным кодом. NoSQL часто основана на модели BASE, отдающей предпочтение доступности над согласованностью. Эта модель дает неформальные гарантии того, что в отсутствие новых обновлений обращение к любому элементу данных рано или поздно вернет его последнюю версию. У такого подхода имеются следующие преимущества:

- простота проектирования;
- горизонтальная масштабируемость и простота репликации;
- отсутствие схемы;
- поддержка гигантских объемов данных.

Теперь рассмотрим некоторые типы баз данных NoSQL.

Хранилища ключей и значений

Хранилище ключей и значений – самый простой тип базы данных. Как следует из названия, хранилище основано на хеш-таблицах. Некоторые хранилища допускают хранение составных типов данных – списков и словарей. В некоторых ситуациях хранилища ключей и значений обеспечивают исключительную скорость, но им недостает поддержки сложных запросов и агрегирования. К числу наиболее известных баз данных такого типа относятся Riak, Redis, Memebase и MemcacheDB.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru