

Содержание

Об авторах.....	12
О рецензентах	13
Предисловие	14
О чем эта книга.....	14
Что вам понадобится	17
Для кого написана книга.....	17
Условные обозначения	18
Сообщение об ошибках	18
Незаконное воспроизведение содержимого книги.....	19
Вопросы.....	19
ГЛАВА 1.	
Первые шаги.....	20
Введение.....	20
Представляем PostgreSQL 9	20
Где взять PostgreSQL	23
Соединение с сервером PostgreSQL	25
Открытие доступа для сетевых/удаленных пользователей.....	28
Использование графических инструментов администрирования	30
Использование инструмента для написания запросов и скриптов psql	34
Безопасная смена пароля	38
Как избежать хранения пароля в коде.....	39
Использование сервисного файла соединений	40
Если установить соединение не удастся.....	41
ГЛАВА 2.	
Исследование базы данных.....	44
Введение.....	44
Как узнать версию сервера.....	44
Определение времени работы сервера	46
Где находятся файлы сервера БД	47

Где находится журнал сообщений сервера	49
Где находится системный идентификатор БД	52
Список баз данных на сервере	53
Сколько таблиц в базе данных	56
Сколько места на диске занимает база данных.....	59
Сколько места на диске занимает таблица	59
Какие таблицы являются самыми большими	60
Сколько строк в таблице	61
Быстрый подсчет строк в таблице.....	62
Понятие объектных зависимостей	67

ГЛАВА 3.

Конфигурация..... 70

Введение.....	70
«Правильная» документация (RTFM).....	70
Планирование новой базы данных	72
Изменение параметров в программах	73
Текущие конфигурационные настройки	75
Параметры, не определяемые по умолчанию	76
Обновление файла параметров.....	78
Настройка параметров для определенных групп пользователей	79
Основной список параметров для проверки конфигурации сервера	80
Добавление внешних модулей в PostgreSQL.....	82
Запуск сервера в энергосберегающем режиме	84

ГЛАВА 4.

Контроль над сервером 86

Введение.....	86
Запуск сервера БД вручную.....	87
Быстрая и безопасная остановка сервера	88
Экстренная остановка сервера.....	89
Перезагрузка конфигурационных файлов.....	89
Быстрый перезапуск сервера	91
Запрет на создание новых соединений.....	93
Создание ограничения по одной сессии на пользователя.....	94
Принудительное отключение пользователей	96
Проектирование с учетом multi-tenancy	97
Использование множественных схем	99
Выделение для пользователя собственной базы данных	101
Запуск нескольких серверов на одной системе	103
Настройка пула соединений	104

ГЛАВА 5.**Таблицы и данные 108**

Введение.....	108
Выбор имени для объекта БД	108
Управление объектами с именами, заключенными в кавычки	110
Принудительная установка единого имени, единого определения столбцов	112
Обнаружение и удаление дубликатов	116
Предотвращение появления строк-дубликатов	119
Поиск уникального ключа для выборки данных	124
Генерирование тестовых данных	126
Случайные данные по примеру.....	129
Подгрузка данных из электронных таблиц	132
Подгрузка данных из файлов	135

ГЛАВА 6.**Безопасность..... 139**

Введение.....	139
Как забрать у пользователя права доступа к таблице.....	140
Как предоставить пользователю доступ к таблице.....	142
Создание нового пользователя	144
Временный запрет соединений для пользователя.....	145
Удаление пользователя без потери его данных.....	146
Проверка безопасности паролей пользователей.....	148
Назначение определенным пользователям ограниченных прав суперпользователя	150
Аудит изменений DDL.....	153
Аудит изменений данных	154
Интеграция с LDAP	158
Соединение с использованием SSL.....	159
Шифрование секретных данных	162

ГЛАВА 7.**Администрирование баз данных 166**

Введение.....	166
Написание скрипта, который срабатывает полностью или не работает вообще	167
Написание скрипта rsq , который прерывается после первой ошибки	170
Действия с несколькими таблицами	171
Добавление и удаление столбца	176

Изменение типа данных столбца	178
Добавление и удаление схем.....	181
Перемещение объектов между схемами.....	183
Добавление и удаление табличных пространств.....	184
Перемещение объектов между табличными пространствами....	188
Доступ к объектам, находящимся в других БД PostgreSQL.....	191
Обновление представления	196

ГЛАВА 8.

Мониторинг и диагностика 202

Введение.....	202
Подключен ли пользователь	205
Что выполняет пользователь	207
Активен или заблокирован запрос	209
Почему запрос заблокирован.....	211
Как прервать сессию	212
Принятие решения о сомнительной подготовленной транзакции	215
Использует ли кто-либо определенную таблицу	215
Когда в последний раз использовали таблицу	217
Сколько места на диске будут занимать временные данные	220
Почему запросы выполняются медленно	222
Обнаружение ошибок и сообщение о них	225
Подведение итога записей в журнале об ошибках за день.....	227

ГЛАВА 9.

Регулярное обслуживание 230

Введение.....	230
Управление автоматическим обслуживанием базы данных ...	231
Предотвращение «автозаморозки» и повреждения страниц.....	236
Предупреждение «свертывания» счетчика транзакций	238
Удаление ранее подготовленных транзакций.....	240
Для любителей временных таблиц.....	243
Обнаружение и исправление раздутых таблиц и индексов.....	244
Обслуживание индексов.....	249
Обнаружение неиспользуемых индексов	252
Безопасное удаление ненужных индексов	254
Планирование обслуживания	255

ГЛАВА 10.

Производительность и согласованность 258

Введение.....	258
Как найти медленные запросы SQL	259

Как получить регулярную статистику из представлений <code>pg_stat*</code>	262
Как установить причины замедления работы SQL	263
Как уменьшить число возвращаемых строк	268
Как упростить сложный SQL	269
Как ускорить выполнение запросов, не переписывая их	275
Почему запрос не использует индекс	279
Как заставить запрос использовать индекс	280
Использование оптимистичных блокировок	282
Сообщения о проблемах с производительностью	284

ГЛАВА 11.

Резервное копирование и восстановление 285

Введение	285
Что такое восстановление после сбоя и как его контролировать	286
Планирование резервного копирования	288
Горячее логическое резервное копирование одной базы данных	291
Горячее логическое резервное копирование всех баз данных	293
Горячее логическое резервное копирование всех таблиц из табличного пространства	294
Резервное копирование определений объектов базы данных	296
Автономное горячее физическое резервное копирование базы данных	297
Горячее физическое резервное копирование и непрерывное архивирование	300
Восстановление всех баз данных	303
Восстановление до определенной временной точки	307
Восстановление удаленной или поврежденной таблицы	310
Восстановление удаленного или поврежденного табличного пространства	312
Восстановление удаленной или поврежденной базы данных	314
Улучшение производительности резервного копирования и восстановления	315
Инкрементальное и дифференциальное резервное копирование и восстановление	318

ГЛАВА 12.

Репликация и обновления	321
Репликация: основные термины	322
Лучшие способы репликации.....	326
Файловая репликация с передачей журналов	328
Настройка потоковой журнальной репликации	332
Управление репликацией с передачей журналов	337
Управление режимом горячего резервирования	341
Выборочная репликация с помощью Londiste 3.0.....	346
Выборочная репликация с помощью Slony 2.0	351
Балансировка нагрузки с помощью pgpool-II 3.0.....	356
Минорное обновление.....	359
Мажорные обновления на месте.....	360
Мажорные онлайн-обновления с помощью инструментов репликации	362

Предисловие

PostgreSQL – это продвинутый сервер баз данных, который доступен на множестве платформ. Заработав отличную репутацию благодаря высокой производительности, стабильности и широким возможностям, PostgreSQL быстро набирает популярность, становясь одной из самых широко используемых СУБД в мире. Один из первых проектов с открытым кодом, в котором участвует множество разработчиков со всего мира, является абсолютно бесплатным. И это работает!

Преимущества свободного программного обеспечения с открытым кодом очевидны: вы вольны устанавливать, использовать и распространять PostgreSQL, и за это не нужно никому платить. Главное же заключается в том, что СУБД PostgreSQL способна долго и бесперебойно работать при минимальной поддержке со стороны администратора. В общем, стоимость использования PostgreSQL довольно мала.

Эта книга поможет вам управлять рабочими базами данных PostgreSQL. Здесь вы найдете рекомендации ведущего разработчика функционала репликации и восстановления PostgreSQL и системного архитектора Skype – самого успешного стартапа, созданного с помощью PostgreSQL. Данное руководство должно стать настольным для разработчиков, имеющих дело с «боевыми» базами данных и использующих Java, Python, Ruby, .Net в любой среде разработки.

В книге представлены ответы на общие вопросы и решения проблем, почерпнутые авторами из личного опыта работы с СУБД PostgreSQL – в качестве пользователей, разработчиков и преподавателей.

Каждая задача разбивается на простые части, или рецепты, проиллюстрированные кодом и сопровождаемые подробными объяснениями, как и почему это работает. Таким образом, руководство может быть полезным как новичкам, так и специалистам. Кроме того, здесь отражены все новые возможности PostgreSQL 9.

О чем эта книга

В первой главе вы познакомитесь с PostgreSQL 9, узнаете, как скачать и установить PostgreSQL, как подключиться к серверу PostgreSQL, открыть доступ для сетевых/удаленных пользователей, как использовать визуальные инструменты администрирования, с помощью каких инструментов писать скрипты и запросы `psql`, как безопасно

менять свой пароль, использовать служебный файл соединений и выяснить причины потери соединения.

Вторая глава поможет вам определить версию сервера БД и время его работы. Вы поймете, где находятся файлы сервера БД, журнал сообщений сервера и системный идентификатор СУБД. Вы сможете просмотреть список баз данных на сервере и узнаете, сколько таблиц в базе данных, сколько места на диске занимают БД и отдельные таблицы, какие таблицы являются самыми большими, сколько строк в таблице и как определить зависимости объектов.

В третьей главе рассматривается, как читать документацию (RTFM), планировать новую базу данных, изменять параметры в программах, как узнавать текущие конфигурационные настройки, параметры, которые не являются значениями по умолчанию, как обновлять файл параметров, как устанавливать параметры для определенных групп пользователей. Также описываются простой список для проверки конфигурации сервера, добавление на сервер PostgreSQL внешних модулей и запуск сервера в режиме экономии энергии.

В четвертой главе вы найдете информацию о ручном запуске сервера БД, быстрой и безопасной остановке сервера, остановке сервера в экстренных случаях, перезагрузке конфигурационных файлов сервера, быстром перезапуске сервера, запрете на установку новых соединений, установке ограничения по одной сессии на каждого пользователя и отключении пользователей от системы. Простые рецепты помогут вам организовать работу в команде, применять множественные схемы, выделять для пользователей приватные базы данных, запускать несколько серверов БД на одной системе и настраивать пул соединений.

Из пятой главы вы узнаете, как выбирать имена для объектов баз данных, как управлять объектами с квотированными именами, присваивать одно имя, одно определение для разных столбцов, обнаруживать и удалять строки-дубликаты, предотвращать появление строк-дубликатов, как находить уникальный ключ для набора данных, генерировать тестовые данные, случайные данные, подгружать данные из динамических таблиц и из неструктурированных файлов.

В шестой главе содержатся рецепты восстановления доступа пользователя к таблице, предоставления пользователю прав доступа к таблице, создания нового пользователя, временного ограничения соединения для пользователя, удаления пользователя без потери его данных, проверки безопасности пароля пользователей, делегирования ограниченных прав суперпользователя для определенных

пользователей, аудита изменений DDL, аудита изменений данных, интеграции с LDAP, установки соединения с использованием SSL и шифровании важных данных.

В седьмой главе рассказывается, как писать скрипт, в котором все строки должны выполняться или, наоборот, не выполняться, как создать скрипт в `psql`, который вызывается при первой ошибке, как выполнять действия сразу с несколькими таблицами, добавлять/удалять столбцы, изменять тип данных столбца, добавлять/удалять схемы, перемещать объекты между схемами, добавлять/удалять табличные пространства, перемещать объекты между табличными пространствами, получать доступ к объектам из других баз данных PostgreSQL и делать представления обновляемыми.

Рецепты, содержащиеся в восьмой главе, позволяют понять, подключен ли пользователь, что он запускает, является ли он активным или его заблокировали, кто его заблокировал, использует ли кто-либо определенную таблицу и когда он в последний раз это делал, сколько места на диске занимают временные данные, почему ваши запросы замедляются. Кроме того, вы узнаете, как обнаружить ошибку и сообщить о ней, как создавать ежедневный итоговый отчет на основе журнализации ошибок, как прервать определенную сессию и что делать с сомнительной подготовленной транзакцией.

Из девятой главы вы узнаете, как контролировать автоматическую поддержку базы данных, как избежать «автозаморозки» и повреждения страниц, обертывания транзакций, как удалять старые подготовленные транзакции, действия для «тяжелых»¹ пользователей со временными таблицами, как обнаруживать и исправлять раздутые таблицы и индексы, как поддерживать индексы, находить неиспользуемые индексы, аккуратно удалять ненужные индексы и планировать обслуживание.

В десятой главе описано, как находить медленные SQL-параметры, получать регулярную статистику из представлений `pg_stat*`, выявлять причины замедления SQL, уменьшать количество возвращаемых строк, упрощать сложный SQL, ускорять запросы без переписывания. Вы узнаете, почему запрос не использует индекс и как заставить запрос его задействовать. Кроме того, рассматривается использование оптимистичных блокировок и сообщение о проблемах с производительностью.

В одиннадцатой главе описываются резервное копирование и восстановление. Многие признают важность резервного копирования, но мало кто думает об этом всерьез. Из этой главы вы получите важную

¹ Речь идет о пользователях, генерирующих «тяжелые» (сильно нагружающие систему) запросы и/или о пользователях с плохим каналом связи. – *Прим. ред.*

информацию о том, как создавать резервную копию и как восстанавливать из нее базу данных PostgreSQL, узнаете о планировании резервного копирования, горячем логическом резервном копировании как одной, так и всех баз данных, горячем логическом резервном копировании всех таблиц в табличном пространстве, резервном копировании определений объектов БД, отдельном горячем физическом резервном копировании БД, горячем физическом резервном копировании и непрерывном архивировании. Кроме того, в этой главе рассматривается восстановление всех баз данных, восстановление к какому-либо моменту времени, восстановление удаленной или поврежденной таблицы, табличного пространства, улучшение производительности резервного копирования/восстановления и инкрементальное/дифференциальное резервное копирование и восстановление.

Двенадцатая глава посвящена репликации и обновлениям. Репликация – это не волшебство, но вещь удивительная (еще удивительнее, когда она работает). В этой главе дается определение репликации, описываются лучшие приемы, настройка репликации, основанной на трансфере файлов журнала, управление режимом hot standby, выборочная репликация с использованием Londiste 3.0, выборочная репликация с использованием Slony 2.0, балансирование загрузки с помощью pgpool II 3.0, обновление до нового минорного релиза (например, с 9.0.0 до 9.0.1), мажорные обновления на месте (например, с 8.4 на 9.0 или с 9.0 на 9.1) и мажорные обновления онлайн с помощью инструментов репликации.

Что вам понадобится

Для работы с книгой вам понадобится следующее программное обеспечение:

- PostgreSQL 9.0 Server Software;
- psql client utility (part of 9.0);
- pgAdmin3 1.12.

Для кого написана книга

Руководство предназначено системным администраторам, администраторам баз данных, архитекторам, разработчикам, а также всем, кто планирует работать с базами данных или уже занимается этим. В первую очередь она подойдет тем, у кого уже есть какой-либо технический опыт.

Условные обозначения

Кодовые слова в тексте выделены моноширинным шрифтом. Например: «В PostgreSQL 9.0 вам больше не понадобится утилита `pg_standby`, поскольку многие ее функции теперь выполняет непосредственно сервер».

Блоки кода выглядят так:

```
CREATE USER repuser
SUPERUSER
LOGIN
CONNECTION LIMIT 1
ENCRYPTED PASSWORD 'changeme';
```

Фрагменты кода, на которые требуется обратить особенное внимание, выделены полужирным моноширинным шрифтом:

```
SELECT *
FROM mytable
WHERE (col1, col2, ... ,colN) IN(SELECT col1, col2, ... ,colN
FROM mytable
GROUP BY
col1, col2, ... ,colN HAVING count(*) > 1);
```

Все, что вводится или выводится в командной строке, также записывается полужирным моноширинным шрифтом:

```
$ postgres --single -D /full/path/to/datadir postgres
```

Новые термины и важные слова выделены *курсивом*.

Слова, которые вы можете видеть на экране, например в меню или диалоговых окнах, в тексте выделены следующим образом: «В Query есть замечательная опция **Visual Explain** и **Graphical Query Builder**».

Предупреждения, важные замечания, советы и подсказки вынесены в такие блоки.

Сообщение об ошибках

Хотя мы делаем все возможное для того, чтобы не допустить ошибок в наших изданиях, вы можете встретить в тексте какие-либо неточности. В таком случае мы будем рады, если вы сообщите нам об этом. Пожалуйста, зайдите на веб-страницу <http://www.packtpub.com/support>, выберите название книги, в которой вам встретилась ошибка, и опишите ее, заполнив соответствующую форму. Как только эта информация будет проверена, мы опубликуем ее в разделе «Errata» (Ошибки) на сайте. Уже найденные ошибки можно просмотреть,

выбрав название интересующей вас книги в разделе <http://www.packtpub.com/support>.

Сообщения об ошибках в русскоязычном издании книги можно оставить на сайте издательства «ДМК Пресс»:

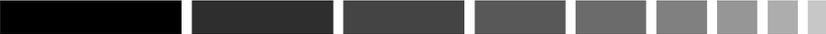
<http://www.dmk-press.ru/contacts1/contacts/>

Незаконное воспроизведение содержимого книги

Пиратские копии – повсеместная проблема. Если вам встретились незаконным образом растиражированные экземпляры данной книги (в любом формате), пожалуйста, сообщите источник публикации, написав письмо по адресу dm@dmk-press.ru или copyright@packtpub.com.

Вопросы

Вы можете присылать любые вопросы, касающиеся данной книги, по адресу dm@dmk-press.ru и questions@packtpub.com. Мы постараемся решить возникшие проблемы.



ГЛАВА 1.

Первые шаги

Рассматриваемые темы:

- Представляем PostgreSQL 9
- Где взять PostgreSQL
- Соединение с сервером PostgreSQL
- Открытие доступа для сетевых/удаленных пользователей
- Использование графических инструментов администрирования
- Использование инструмента для написания запросов и скриптов `psql`
- Безопасная смена пароля
- Как избежать хранения пароля в коде
- Использование сервисного файла соединений
- Если установить соединение не удастся

Введение

PostgreSQL – это мощная система управления базами данных (СУБД), используемая для выполнения самых разных задач. Путь к овладению ею непрост, но любое путешествие должно начинаться с первого шага.

Для начала рассмотрим самое первое соединение с базой. Для многих оно оказывается непреодолимым препятствием, поэтому разберем его более подробно: сначала остановимся на доступе для удаленных пользователей, а оттуда доберемся до инструментов администрирования с графическим интерфейсом.

Также проанализируем инструмент запросов `psql`, который будет использоваться для подгрузки нашей базы данных.

Кроме того, здесь представлено несколько полезных рецептов, которые пригодятся вам в будущем.

Представляем PostgreSQL 9

PostgreSQL – это продвинутый сервер для баз данных SQL, доступный на множестве платформ.

Одно из очевидных преимуществ СУБД PostgreSQL заключается в ее открытости: вы можете совершенно бесплатно устанавливать, использовать и распространять ее. Но самое главное – PostgreSQL славится своей стабильностью, и во многих случаях необходима лишь минимальная поддержка. Эта система очень недорога.

Возможности PostgreSQL огромны, ведь она разрабатывается и улучшается на протяжении вот уже 20 лет – начиная с рождения в Группе исследований проблем баз данных Калифорнийского университета Беркли. Сегодня свой вклад вносит огромная армия разработчиков кода, архитекторов, администраторов БД и преподавателей – многие из них используют PostgreSQL на своем основном месте работы. Некоторые, в том числе и авторы этой книги, работают в компаниях, специализирующихся на создании сервисов для PostgreSQL. Но нет такой компании, которая могла бы считать себя владельцем этой СУБД, и никто не требует и не предполагает, что вы где-то будете указывать, что используете PostgreSQL.

Ключевые особенности PostgreSQL:

- высокий уровень соответствия стандартам языка SQL вплоть до стандарта SQL:2008;
- архитектура «клиент-сервер»;
- рассчитанная на большое количество параллельных операций архитектура: чтение и запись в базу данных не мешают друг другу;
- большие возможности конфигурирования и работы с различными типами приложений;
- отличная масштабируемость и производительность в сочетании с широкими возможностями настройки.

Чем отличается PostgreSQL от других СУБД

При разработке PostgreSQL ставились следующие задачи:

- создание надежного высококачественного программного кода, сопровождаемого подробными комментариями;
- сведение к минимуму необходимости в обслуживании системы;
- соответствие стандартам SQL, интероперабельность, обратная совместимость кода;
- высокая производительность, безопасность и доступность.

Многих удивляет, что по функциональности PostgreSQL ближе к Oracle или Microsoft SQL Server, чем к MySQL. Да и связывает MySQL с PostgreSQL лишь то, что оба эти проекта обладают открытым исходным кодом, в то время как основные свойства и философия различаются в корне.

Одной из ключевых особенностей Oracle – начиная с Oracle 7 – была «изоляция снимков» (snapshot isolation), позволявшая читающим и пишущим процессам не блокировать друг друга, а работать независимо. Как ни удивительно, PostgreSQL – первая СУБД, в которой эта особенность заложена изначально. В PostgreSQL она называется MVCC (Multi-Version Concurrency Control – мультиверсионная система управления параллельным выполнением операций). Далее вы сможете ознакомиться с ней подробнее.

PostgreSQL – это СУБД для широкого использования. Вы настраиваете базу данных так, чтобы вам было удобно с ней работать. PostgreSQL представляет для этого огромное количество возможностей. Вы можете использовать как модель нормализованной базы данных, так и расширения для работы с массивами и record-подтипами, или же положиться на полностью динамическую схему, используя расширение `hstore`. Кроме того, PostgreSQL позволяет создавать серверные функции на одном из десятка различных языков программирования.

Одно из полезных свойств PostgreSQL – расширяемость. Вы можете добавлять свои типы данных, операторы, типы индексов и функциональные языки. Например, можно переопределить поведение части системы и изменить выполнение команд, используя систему плагинов, или же добавить новый оптимизатор.

Все эти свойства открывают для архитекторов программного обеспечения огромные возможности – без лишних проблем создавать и поддерживать приложения на протяжении длительного времени. К сожалению, нельзя уместить в одну книгу все, что могло бы пригодиться разработчикам. Данное руководство посвящено администрированию, техническому обслуживанию и резервному копированию.

В самом начале, когда PostgreSQL была только системой для исследования в академических кругах, основные разработки сводились к добавлению новых впечатляющих функций. За последние 15 лет было переписано и усовершенствовано огромное количество строк кода, что сделало PostgreSQL одним из самых больших и стабильных серверов, готовых для оперативного использования.

Возможно, вы где-то читали, что PostgreSQL работает медленнее, чем Ваша Любимая СУБД (какой бы она ни была). Мы, авторы

этой книги, можем сообщить, что на протяжении последних шести лет трудились над улучшением производительности и наша команда добилась больших успехов в создании высокопроизводительного и масштабируемого сервера. Это дает проекту PostgreSQL большее пространство для развития.

Кто использует PostgreSQL? Среди крупных компаний можно назвать Apple, BASF, Genentech, IMDb.com, Skype, NTT, Yahoo и The National Weather Service. По данным Европейской комиссии, число загрузок PostgreSQL за год доходит до миллиона, и многие пользователи БД считают PostgreSQL достойной альтернативой их привычной СУБД.

Осталось упомянуть еще одну вещь. При рождении PostgreSQL получила имя Postgres, поэтому в некоторых ключевых частях проекта вы можете встретить именно это слово. Например, база данных по умолчанию называется postgres, и при установке программного обеспечения это слово часто выступает в качестве имени пользователя. Именно поэтому многие сокращают PostgreSQL до Postgres и считают эти два названия взаимозаменяемыми.

PostgreSQL читается «пост-грес-ку-эл», а Postgres – «пост-грес».

Некоторые используют еще и название Postgre, но это только внесит дополнительную путаницу. Двух названий вполне достаточно – пожалуйста, используйте только их!

Где взять PostgreSQL

PostgreSQL на 100 % является свободным программным обеспечением с открытым кодом.

Вы можете свободно использовать, изменять и распространять PostgreSQL любым удобным для вас способом. Лицензия на применение PostgreSQL является улучшенной версией лицензии *BSD (Berkeley Distribution Software)* и в настоящее время называется *TPL (The PostgreSQL License)*.

Как это сделать

PostgreSQL уже используется во многих пакетах приложений, так что, возможно, она уже установлена на ваших серверах. Многие дистрибутивы Linux включают в себя PostgreSQL как часть основной системы, или же PostgreSQL содержится на диске с этими дистрибутивами.

Единственная проблема в том, что, скорее всего, прилагаемая версия PostgreSQL не окажется самой последней. Точнее, она будет

самой свежей на момент выхода дистрибутива операционной системы. Как правило, нет смысла держаться именно за эту версию – такой подход не дает какой-либо дополнительной стабильности, а более новые версии СУБД так же хорошо поддерживаются различными Linux-дистрибутивами.

Если у вас до сих пор нет копии дистрибутива PostgreSQL или имеющаяся версия уже не нова, вы можете скачать исходники или бинарные пакеты для различных операционных систем на сайте <http://www.postgresql.org/download/>.

Конкретный способ установки зависит от используемой вами платформы, и универсального рецепта здесь нет. Просто следуйте инструкциям, прилагаемым к вашей версии дистрибутива – этого будет достаточно. Мы сознательно не описываем здесь процесс установки, чтобы не исказить информацию, приведенную специально в помощь вам.

Если вы хотите получать сообщения об обновлениях и последних новостях по электронной почте, можно подписаться на рассылку, описывающую обновления от всех производителей, которые поддерживают PostgreSQL. Ежемесячно вам будет приходить несколько писем о новых версиях ядра PostgreSQL, приложениях, конференциях, а также и другая информация для пользователей. Поддерживать связь с разработчиками порой бесполезно.

Подробнее о данной рассылке можно узнать по адресу <http://archives.postgresql.org/pgsql-announce/>.

Как это работает

Многие удивляются: неужели все действительно бесплатно? Не придется ли в итоге платить? Кто стал бы трудиться даром?

Программные продукты с открытым кодом, такие как PostgreSQL, создаются сообществом разработчиков и пользователей. Из множества маленьких песчинок вырастает большая гора. Среди этих людей есть и любители, и профессионалы, ведь проектирование приложений часто бывает полезным не только разработчикам, но и их работодателям.

Хотите верьте, хотите нет – это просто работает!

И еще

Не забудьте, что PostgreSQL – это не только ядро. Множество веб-сайтов предлагают дополнения, расширения и инструменты для PostgreSQL. В блогах можно обнаружить массу полезной информации

и подсказок, которые могут помочь в работе. И, наконец, существуют компании профессионалов, которые в любой момент придут вам на помощь.

Соединение с сервером PostgreSQL

Как подключиться к PostgreSQL? Для многих с подключения начинается первое знакомство с PostgreSQL, так что постараемся сделать его приятным. Попутно разберем проблемы, которые могут возникнуть, и методы их решения. Помните, что соединение должно быть безопасным, так что постараемся защитить данные, к которым хотим получить доступ.

Прежде чем вы сможете работать с базой данных, необходимо подключиться к серверу, который откроет *сессию*.

Сессия рассчитана на длительный период времени, поэтому можно один раз подключиться, выполнить все необходимые запросы (которых может быть много) и, наконец, отключиться. Существуют некоторые накладные расходы в процессе подключения. Это можно заметить, если несколько раз подряд приходится подключаться/отключаться, и в таком случае есть смысл изучить пулы соединений (*connection pools*). Они позволяют быстро разбираться с предыдущими сессиями при каждом последующем подключении.

Подготовка

Итак, возьмем базу данных. Если вы не знаете, где она находится, это сильно затруднит работу. Баз данных может быть несколько; следует знать, какая именно вам нужна в данный момент, и иметь к ней соответствующие права доступа.

Как это сделать

Для подключения к PostgreSQL нужно определить следующие пять параметров:

- узел (*host*) или сетевой адрес;
- порт (*port*);
- имя базы данных;
- пользователь (*user*);
- пароль (или другие средства аутентификации).

Для соединения необходимо, чтобы по адресу `host` располагался сервер PostgreSQL, слушающий порт номер `port`. На этом сервере должны быть база данных `dbname` и пользователь `user`. Узел должен принимать соединения от вашего клиентского приложения (об этом читайте в разделе «Открытие доступа для сетевых/удаленных пользователей»), и необходимо пройти аутентификацию с использованием того метода, который требуется для данного сервера. Например, бесполезно вводить пароль на сервере, где используется другой механизм аутентификации.

Практически все интерфейсы PostgreSQL используют библиотеку `libpq`. При использовании `libpq` большая часть параметров соединения идентичны, поэтому обсудим их здесь и более не будем возвращаться к данному вопросу.

Если вы не определили эти параметры, ищем значения, заданные с помощью следующих переменных окружения:

- `PGHOST` или `PGHOSTADDR`;
- `PGPORT` (если ничего не указано, установите значение 5432);
- `PGDATABASE`;
- `PGUSER`;
- `PGPASSWORD` (вот это вообще-то не рекомендуем делать).

Если вы как-либо определили первые четыре параметра, но не определили пароль, будем искать файл с паролем (см. раздел ниже).

В некоторых интерфейсах PostgreSQL протокол «клиент-сервер» используется напрямую, так что значения по умолчанию могут быть другими. Существенных различий быть не должно, но лучше проверить точный синтаксис для вашего интерфейса.

Как это работает

PostgreSQL – это клиент-серверная база данных. Система, на которой она запущена, считается узлом (`host`). Можно получить удаленный доступ к серверу PostgreSQL по сети, но нужно указать имя узла (`hostname`) или IP-адрес (`hostaddr`). Например, если вы хотите подключить TCP/IP к своей системе, в качестве узла указывайте `localhost`. Как правило, лучше использовать соединение через Unix-сокеты, которое выполняется, если имя узла начинается с наклонной черты (`/`). Предполагается, что имя соответствует имени директории (по умолчанию `/tmp`).

В любой системе может быть несколько серверов баз данных. Каждый сервер слушает определенный сетевой порт (для каждого сервера в системе порт свой). По умолчанию для PostgreSQL используется порт 5432. Он зарезервирован для PostgreSQL, что официально за-

регистрировано в IANA. (Во многих *nix-серверах можно видеть его использование в файле `/etc/services`.) Если в системе несколько серверов, номер порта может служить уникальным идентификатором.

Иногда сервер баз данных называют кластером баз данных, так как PostgreSQL позволяет определить несколько баз данных на любом сервере. Каждый запрос на соединение должен содержать имя базы данных, и после установления соединения вы будете иметь дело только с объектами указанной базы данных.

Имя пользователя применяется для идентификации соединения. По умолчанию количество соединений для одного пользователя неограничено, но есть способ ввести такое ограничение; он будет описан в одном из рецептов. В более поздних версиях СУБД понятия «пользователь» (`user`) и «группа» (`group`) объединили в понятие «роль» (`role`). Пользователи в прежнем понимании – это своего рода «логин-роли». Можно считать, что «логин-роль» – такая роль, которой можно назначить привилегию `CONNECT`. Несмотря на объединение понятий, во многих случаях можно встретить прежние термины («пользователь» и «группа»).

При каждом соединении должна проводиться аутентификация. Способ определяется сервером.

Для каждого установленного соединения в любое время возможна одна активная транзакция и одна полностью активная инструкция.

Каждый сервер может одновременно поддерживать ограниченное количество соединений, поэтому если сервер перегружен, то запрос на соединение может быть отклонен.

Проверка информации о соединении

Чтобы убедиться, что вы подключились к нужному серверу нужным способом, выполните все или некоторые из перечисленных команд:

```
SELECT inet_server_port();
```

(Порт, который слушает сервер.)

```
SELECT current_database();
```

(Текущая база данных.)

```
SELECT current_user;
```

(Идентификатор текущего пользователя.)

```
SELECT inet_server_addr();
```

(IP-адрес сервера, принявшего соединение.)

SQL-запроса, показывающего пароль, по понятным причинам не существует.

Еще может пригодиться

```
SELECT version();
```

Смотрите также

В процессе соединения есть еще много тонкостей. Часть из них рассматривается в шестой главе, часть описана в этой. Самую подробную информацию вы найдете в документации по серверу PostgreSQL.

Открытие доступа для сетевых/удаленных пользователей

PostgreSQL поставляется во многих сборках и пакетах, и в некоторых случаях из соображений безопасности удаленные соединения по умолчанию оказываются заблокированными.

Как это сделать

- Добавьте в `postgresql.conf` следующую строку (или отредактируйте соответствующую):
`listen_addresses = '*'`
- Чтобы разрешить доступ ко всем базам данных всем пользователям по паролю, запишите в файл `pg_hba.conf` первой строкой:

```
# TYPE      DATABASE   USER       CIDR-ADDRESS   METHOD
host                all        all           0.0.0.0/0      md5
```

Как это работает

Параметр `listen_addresses` определяет IP-адрес, соединения с которым будет ожидать сервер. Это позволяет иметь в системе несколько сетевых карт. Как правило, вы ждете соединения по всем доступным адресам, потому используете значение `*`.

В файле `pg_hba.conf` содержится набор правил аутентификации для данного узла. Правила рассматриваются последовательно, пока одно из них не сработает или пока попытка соединения не будет отклонена методом `reject`.

Правило из примера означает, что для соединения любого пользователя с любой базой данных на любом IP-адресе для аутентификации потребуется md5-шифрованный пароль.

`Type = host` означает удаленное соединение.

`Database = all` значит «для всех баз данных». Другие имена воспринимаются в точности так, как записаны, за исключением случаев, когда перед именем стоит плюс (+), что означает групповую роль.

Кроме того, можно перечислить имена пользователей через запятую или дать ссылку на файл со списком пользователей с помощью символа @. Также можно задать значение `sameuser`; тогда правило будет считаться выполненным, если имена базы данных и пользователя будут совпадать.

`User = all` значит «для всех пользователей». Другие имена воспринимаются в точности так, как записаны, за исключением случаев, когда перед именем стоит плюс (+), что означает групповую роль. Кроме того, можно перечислить имена пользователей через запятую или дать ссылку на файл со списком пользователей с помощью символа @.

`CIDR-ADDRESS` состоит из двух частей: IP-адрес/маска подсети. Маска подсети задается числом первых битов IP-адреса. Таким образом, /0 означает 0 битов IP-адреса, то есть доступ будет открыт для всех IP-адресов. Например, если указано `192.168.0.0/24`, то берем первые 24 бита, то есть для соединения подойдут все IP-адреса вида `192.168.0.x`. Кроме того, можно указать значение `samenet` или `samehost`.

Не используйте значение `password`, которое разрешает пароли в виде простого текста.

`Method = trust` значит «никакой аутентификации». Среди других методов аутентификации назовем GSSAPI, SSPI, LDAP, RADIUS и PAM. Можно также создавать соединения с использованием SSL – тогда аутентификация будет проводиться по SSL-сертификатам клиента.

И еще

В ранних версиях PostgreSQL сетевой доступ разрешался с помощью ключа `-i` в командной строке при запуске сервера. Эта опция доступна и сейчас, но теперь она записывается по-другому:

```
listen_addresses = '*'
```

Так что, если в инструкции по настройке вам встретилось упоминание этой опции, возможно, ваше руководство уже устарело, и не исключено, что многие другие рекомендации тоже не соответствуют действительности.

Смотрите также

Точнее стандартное расположение файлов можно узнать в документации по вашему установщику и/или операционной системе.

Использование графических инструментов администрирования

Системным администраторам часто требуются графические инструменты администрирования.

В PostgreSQL много таких инструментов, но самыми популярными являются следующие:

- pgAdmin3;
- phpPgAdmin.

Первый по умолчанию входит в установщик PostgreSQL в Windows, что, вероятно, делает его самым популярным интерфейсом даже несмотря на то, что чаще на сервере стоит Linux или другая операционная система. Поэтому в книге подробнее рассматривается pgAdmin3.

Как это сделать

pgAdmin3 – это клиентское приложение, которое отправляет и получает SQL с сервера PostgreSQL, отображая для вас результат своей работы. Один клиент pgAdmin может иметь доступ к нескольким серверам PostgreSQL, так же как и один сервер PostgreSQL может принимать несколько клиентов pgAdmin.

Обычно pgAdmin3 сокращают до pgAdmin. Появление тройки в названии – долгая история, но никакого практического смысла здесь нет. Это не номер версии. На момент написания книги самой свежей является версия 1.12.

При первом запуске pgAdmin предложит вам зарегистрировать новый сервер, как показано на рис. 1.1.



Рис. 1.1