

PostgreSQL

ДЛЯ НАЧИНАЮЩИХ



О PostgreSQL

PostgreSQL — наиболее полнофункциональная, свободно распространяемая СУБД с открытым кодом. Разработанная в академической среде, за долгую историю сплотившая вокруг себя широкое сообщество разработчиков, эта СУБД обладает всеми возможностями, необходимыми большинству заказчиков. PostgreSQL активно применяется по всему миру для создания критичных бизнес-систем, работающих под большой нагрузкой.

Немного истории

Современный проект PostgreSQL ведет происхождение от проекта POSTGRES, который разрабатывался под руководством Майкла Стоунбрейкера (Michael Stonebraker), профессора Калифорнийского университета в Беркли. До этого Майкл Стоунбрейкер уже возглавлял разработку INGRES — одной из первых реляционных СУБД, — и POSTGRES возник как результат осмысления предыдущей работы и совершенных ошибок и желания преодолеть ограниченность системы типов за счет возможности определения новых типов данных.

Работа над проектом началась в 1985 году и до 1988 года был опубликован ряд научных статей, описывающих модель данных, язык запросов POSTQUEL (в то время SQL еще не был общепризнанным стандартом) и устройство хранилища данных.

POSTGRES иногда еще относят к так называемым постреляционным СУБД. Ограниченность реляционной модели всегда была предметом критики, хотя и являлась обратной стороной ее простоты и строгости. Однако проникновение компьютерных технологий во все сферы жизни требовали новых приложений, а от баз данных — поддержки новых типов данных и возможностей, например, поддержка наследования, создание сложных объектов и управление ими.

Первая версия СУБД была выпущена в 1989 году. База данных совершенствовалась на протяжении нескольких лет, а в 1993 году, когда вышла версия **4.2**, проект был закрыт. Но, несмотря на официальное

закрытие, открытый код и BSD-лицензия позволили выпускникам Беркли Эндрю Ю и Джоли Чену в 1994 году взяться за его дальнейшее развитие. В 1994 году они заменили язык запросов POSTQUEL на ставший общепринятым SQL, а проект получил название **Postgres95**.

К 1996 году стало ясно, что название «**Postgres95**» не выдержит испытание временем и было выбрано новое имя — **PostgreSQL**, которое отражает связь с оригинальным проектом POSTGRES и приобретением SQL. Именно поэтому PostgreSQL произносится «постгрес-ку-эль» или просто «постгрес», но не «постгре-эс-ку-эль».

Новая версия стартовала как **6.0**, продолжая исходную нумерацию. Проект вырос и управление на себя взяла небольшая вначале группа инициативных пользователей и разработчиков, которая получила название Глобальной группы разработки PostgreSQL (PostgreSQL Global Development Group).



Задний ряд: **Marc Fournier, Jan Wieck, Jolly Chen, Thomas Lokhart, Tom Lane.**

Первый ряд: **Andrew Yu, Vadim Mikheev**

Развитие

Все основные решения о планах развития и выпусках новых версий принимаются Управляющим комитетом (Core team) проекта. В настоящий момент он состоит из шести человек.

Помимо обычных разработчиков, вносящих посильную лепту в развитие системы, выделяется группа основных разработчиков, сделавших существенный вклад в развитие PostgreSQL, а также группа разработчиков, имеющих право записи в репозиторий исходного кода. Состав группы разработчиков со временем меняется, появляются новые члены, кто-то уходит от проекта. Актуальный список разработчиков поддерживается на официальном сайте PostgreSQL:

www.postgresql.org

Весьма значителен вклад российских разработчиков в PostgreSQL. Это, пожалуй, самый крупный глобальный проект с открытым исходным кодом с таким широким российским представительством.

Большую роль в становлении и развитии PostgreSQL сыграл программист из Красноярска Вадим Михеев, входивший в Управляющий комитет. Он является автором таких важнейших частей системы, как многоверсионное управление одновременным доступом (MVCC), система очистки (vacuum), журнал транзакций (WAL), вложенные запросы, триггеры. Сейчас Вадим уже не занимается проектом.

В настоящий момент в число основных разработчиков проекта входят трое представителей из России — Олег Бартунов, Федор Сигаев и Александр Коротков, — основавших в 2015 году компанию Postgres Professional. Среди крупных направлений выполненных ими работ можно выделить локализацию PostgreSQL (поддержка национальных кодировок и Unicode), систему полнотекстового поиска, работу с массивами и слабо-структурированными данными (hstore, json, jsonb), новые методы индексации (GiST, GIN, SP-GiST). Они являются авторами большого числа популярных расширений.

Цикл работы над очередной версией PostgreSQL обычно составляет около года. За это время от всех желающих принимаются на

рассмотрение патчи с исправлениями, изменениями и новым функционалом. Для обсуждения патчей по традиции используется список рассылки `pgsql-hackers`. Если сообщество признает идею полезной, а реализацию — правильной, и код проходит обязательную проверку другими разработчиками, то он включается в новый релиз.

В некоторый момент объявляется этап стабилизации кода — новый функционал откладывается до следующей версии, а принимаются только исправления или улучшения уже включенных в релиз патчей. Иногда, в процессе работы над новой версией вскрываются и исправляются старые ошибки, а наиболее критические из них исправляются и в предыдущих версиях. По мере накопления таких исправлений принимается решение о выпуске новой стабильной версии, совместимой со старой. Например, версия 9.5.3 содержит только исправления ошибок, найденных в 9.5.

Несколько раз в течении релизного цикла выпускаются бета-версии, ближе к концу цикла появляется релиз-кандидат, а вскоре выходит и новая основная версия PostgreSQL с увеличенным вторым числом номера (например, 9.6 после 9.5). Когда Управляющий комитет решает, что количество изменений переросло в качество, принимается решение об увеличении первого числа.

Поддержка

Глобальная группа разработки выполняет поддержку основных версий системы в течении пяти лет с момента выпуска. Эта поддержка, как и координация разработки, осуществляется через списки рассылки. Корректно оформленное сообщение об ошибке имеет все шансы на скорейшее решение: нередко случаи, когда исправления ошибок выпускаются в течении суток.

Помимо поддержки сообществом разработчиков, ряд компаний по всему миру осуществляет коммерческую поддержку PostgreSQL. В России такой компанией является Postgres Professional, предоставляя услуги по поддержке в режиме 24x7.

Современное состояние

PostgreSQL является одной из самых популярных баз данных. За более чем 20-летнюю историю развития на прочном фундаменте, заложенном академической разработкой, PostgreSQL выросла в полноценную СУБД уровня предприятия и составляет реальную альтернативу коммерческим базам данных. Чтобы убедиться в этом, достаточно посмотреть на важнейшие характеристики новейшей на сегодняшний день версии **PostgreSQL 9.5.3**.

Надежность и устойчивость

Вопросы обеспечения надежности особенно важны в приложениях уровня предприятия для работы с критически важными данными. С этой целью PostgreSQL позволяет настраивать горячее резервирование, восстановление на заданный момент времени в прошлом, различные виды репликации (синхронную, асинхронную, каскадную).

Безопасность

PostgreSQL позволяет работать по защищенному SSL-соединению и предоставляет большое количество методов аутентификации, включая аутентификация по паролю, по клиентским сертификатам, с помощью внешних сервисов (LDAP, RADIUS, PAM, Kerberos).

При управлении пользователями и доступом к объектам БД предоставляются следующие возможности:

- создание и управление пользователями и групповыми ролями;
- разграничение доступа к объектам БД на уровне как отдельных пользователей, так и групп;
- детальное управление доступом на уровне отдельных столбцов и строк;
- поддержка SELinux через встроенную функциональность SE-PostgreSQL (мандатное управление доступом).

В компании **Postgres Professional** ведется работа по государственной сертификации PostgreSQL для использования в системах обработки конфиденциальной информации и персональных данных.

Соответствие стандартам

По мере развития стандарта ANSI SQL, его поддержка постоянно добавлялась в PostgreSQL. Это относится ко всем версиям стандарта: SQL-92, SQL:1999, SQL:2003, SQL:2008, SQL:2011. В настоящий момент PostgreSQL 9.5 обеспечивает высокий уровень соответствия стандарту SQL:2011 и поддерживает 160 из 179 обязательных возможностей, а также большое количество необязательных.

Поддержка транзакционности

PostgreSQL обеспечивает полную поддержку свойств транзакций ACID и обеспечивает эффективную изоляцию транзакций. Для этого в PostgreSQL используется механизм многоверсионного управления одновременным доступом (MVCC). Он позволяет обходиться без блокировок во всех случаях, кроме одновременного изменения одной и той же строки данных в нескольких процессах. При этом читающие транзакции вообще никогда не блокируют пишущие транзакции, а пишущие — читающих. Это справедливо и для самого строгого уровня изоляции `serializable`, который, используя инновационную систему `Serializable Snapshot Isolation`, обеспечивает полное отсутствие аномалий сериализации и гарантирует, что при одновременном выполнении транзакций результат будет таким же, как и при последовательном выполнении.

Возможности для разработчиков приложений

Разработчики приложений получают в свое распоряжение богатый инструментарий, позволяющий реализовать приложения любого типа:

- Возможность использования различных языков для серверного программирования, в том числе встроенного PL/pgSQL, удобного тесной интеграцией с SQL, C для критичных по производительности задач, а также Perl, Python, Javascript и Java;
- Программные интерфейсы для обращения к СУБД из приложений на любом языке, включая стандартные интерфейсы ODBC и JDBC;
- Богатый набор объектов баз данных, позволяющий эффективно реализовать логику любой сложности на стороне сервера: таблицы и индексы, ограничения целостности, представления и материализованные представления, последовательности, наследование и секционирование, вложенные подзапросы и with-запросы (в том числе рекурсивные), агрегатные и оконные функции, хранимые функции, триггеры и т. д.;
- Встроенная гибкая система полнотекстового поиска с поддержкой русского и всех европейских языков, поддерживаемая специально разработанным эффективным индексным доступом;
- Поддержка слабо-структурированных данных в духе NoSQL: хранилище пар «ключ-значение» hstore, xml, json (включая эффективное двоичное представление jsonb);
- Подключение внешних источников данных (включая все основные СУБД) в качестве таблиц с возможностью их полноценного использования, в том числе записи и распределенного выполнения запросов.

Масштабируемость и производительность

PostgreSQL эффективно использует современную архитектуру многоядерных процессоров — его производительность растет практически линейно с увеличением количества ядер.

В версии 9.6, которая должна появиться в 2016 году, PostgreSQL получит возможность параллельного чтения данных и выполнения

запросов, что еще больше повысит возможности использования аппаратных средств для ускорения операций.

Планировщик запросов

В PostgreSQL используется планировщик запросов, основанный на стоимости. Используя собираемую статистику и учитывая в своих математических моделях как дисковые операции, так и время процессора, планировщик позволяет оптимизировать самые сложные запросы. В его распоряжении находятся все методы доступа к данным и способы выполнения соединений, характерные для передовых коммерческих СУБД.

Возможности индексирования

В PostgreSQL реализованы различные методы индексирования. Помимо традиционных B- и R-деревьев, также доступны:

- **GiST** — обобщенное сбалансированное дерево поиска, которое может применяться для данных, не допускающих упорядочения. Примерами могут служить R-деревья для индексирования точек на плоскости с возможностью поиска ближайших соседей (k-NN search) и индексирование операции пересечения для интервалов;
- **SP-GiST** — обобщенное несбалансированное дерево поиска, примером которого может служить дерево квадрантов или префиксное дерево;
- **GIN** — обобщенный инвертированный индекс. Основной областью применения является полнотекстовый поиск, где требуется находить, в каких документах встречается указанное в поисковом запросе слово. Другим примером является поиск значений в массивах данных;
- **BRIN** — индекс небольшого размера, позволяющий найти компромисс между размером индекса и скоростью поиска. Эффективен на больших кластеризованных таблицах.

За счет расширяемости набор методов индексного доступа постоянно расширяется.

Независимо от типа, индексы могут строиться как по одному, так и по нескольким столбцам таблицы, а также по выражениям. Можно создать частичный индекс только для определенных строк таблицы. Покрывающие индексы позволяют ускорить запросы за счет того, что все необходимые данные извлекаются из самого индекса без обращения к таблице.

Несколько индексов могут автоматически объединяться с помощью битовой карты для ускорения доступа к таблице.

Кроссплатформенность

PostgreSQL работает на операционных системах семейства Unix, включая серверные и клиентские разновидности Linux, FreeBSD, Solaris, Mac OS X, а также на Windows.

За счет открытого и переносимого кода на языке C PostgreSQL можно собрать на самых разных платформах, даже если для нее отсутствует поддерживаемая сообществом сборка.

Расширяемость

Расширяемость — одно из фундаментальных преимуществ системы, лежащее в основе архитектуры PostgreSQL. Пользователи могут самостоятельно, не меняя базовый код системы, добавлять:

- Типы данных;
- Функции и операторы для работы с новыми типами;
- Индексные методы доступа;
- Языки серверного программирования;
- Подключения к внешним источникам (Foreign Data Wrappers);
- Загружаемые расширения.

Полноценная поддержка расширений позволяет реализовать функционал любой сложности, не внося изменений в ядро PostgreSQL, и допуская подключение по мере необходимости. Например, именно в виде расширений построены такие сложные системы, как:

- **CitusDB** — возможность распределения данных по разным экземплярам PostgreSQL (шардинг) и массивно-параллельного выполнения запросов;
- **PostGIS** — система обработки геоинформационных данных.

Только стандартный комплект, входящий в сборку PostgreSQL 9.5, содержит больше полусотни расширений, доказавших свою надежность и полезность.

Доступность

Лицензия PostgreSQL разрешает неограниченное использование СУБД, модификацию кода, а также включение в состав других продуктов, в том числе закрытых и коммерческих.

Независимость

PostgreSQL не принадлежит ни одной компании и развивается международным сообществом, в том числе и российскими разработчиками. Это означает, что системы, использующие PostgreSQL, не зависят от конкретного вендора, тем самым в любой ситуации сохраняя вложенные в них инвестиции.

Установка и начало работы

Как же начать работать с PostgreSQL? Нет ничего проще: если вы располагаете получасом свободного времени и готовы попробовать, следуйте за нами. В этой главе мы объясним, как установить службу PostgreSQL и управлять ей, а потом создадим простую базу данных и покажем, как создать в ней таблицы и выполнять запросы. Будет неплохо, если вы сразу начнете пробовать команды по мере чтения.

Мы будем использовать дистрибутив **Postgres Pro**, разработанный в нашей компании Postgres Professional. Этот дистрибутив полностью совместим с обычной СУБД PostgreSQL и дополнительно включает в себя некоторые разработки, выполненные в нашей компании, а также ряд возможностей, которые будут доступны только в следующей версии PostgreSQL.

Итак, приступим. Установка и запуск сервера PostgreSQL зависит от того, какую операционную систему вы используете. Если у вас Windows, читайте дальше; если Linux семейства Debian или Ubuntu — переходите сразу к с. 16.

Инструкцию по установке для других операционных систем вы найдете на сайте нашей компании:

<http://postgrespro.ru/products/postgrespro/download/latest>

Если нужной вам версии там не оказалось — воспользуйтесь обычным дистрибутивом PostgreSQL: инструкции находятся по адресу:

www.postgresql.org/download

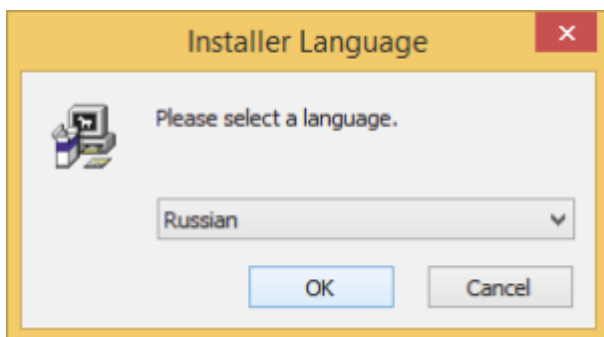
Windows

Установка

Скачайте установщик последней версии PostgreSQL с нашего сайта: www.postgrespro.ru/windows

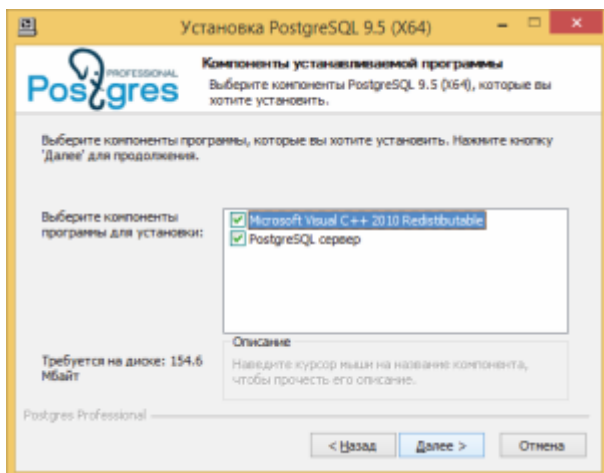
Выберите 32- или 64-разрядную версию в зависимости от того, какая у вас версия Windows.

Запустите скачанный файл и выберите язык установки:



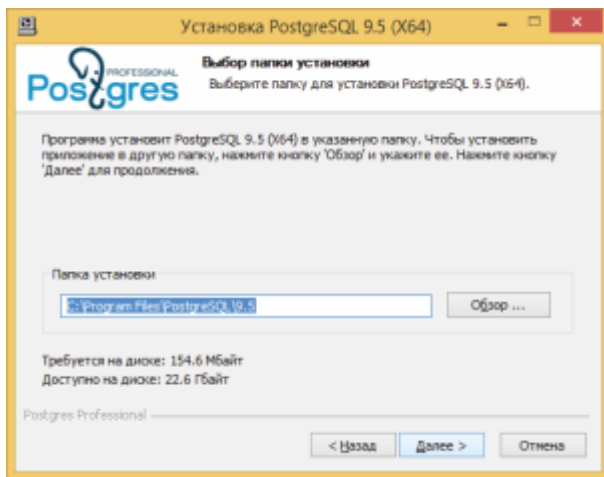
Установщик построен в традиционном стиле «мастера»: вы можете просто нажимать на кнопку «Далее», если вас устраивают предложенные варианты. Остановимся подробнее на основных шагах.

Компоненты устанавливаемой программы:

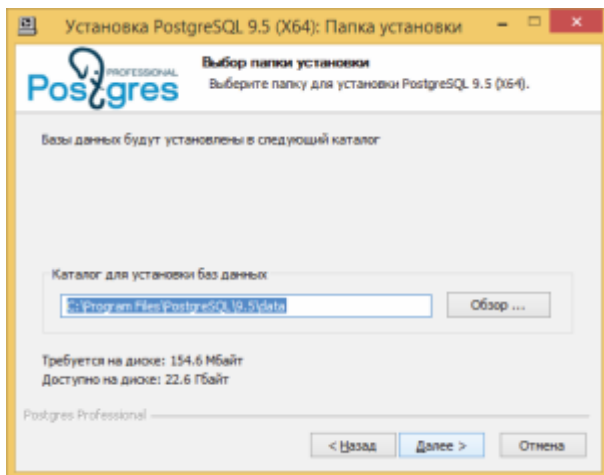


Оставьте оба флажка отмеченными, если не уверены, какие выбрать.

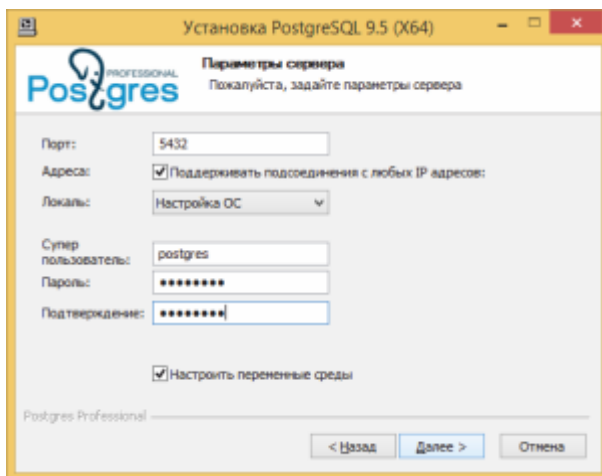
Выбор папки установки:



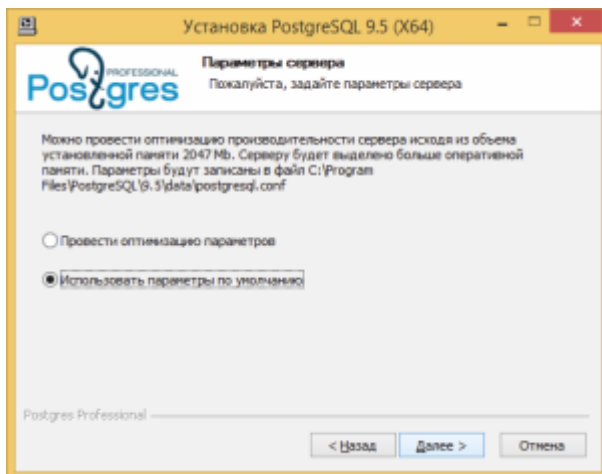
По умолчанию PostgreSQL устанавливается в каталог `C:\Program Files\PostgreSQL\9.5` или в каталог `C:\Program Files (x86)\PostgreSQL\9.5` — для 32-разрядной версии на 64-разрядной системе.



Отдельно можно выбрать расположение каталога для баз данных. Именно здесь будет находиться хранимая в СУБД информация, так что убедитесь, что на диске достаточно места, если вы планируете хранить много данных. Параметры сервера:



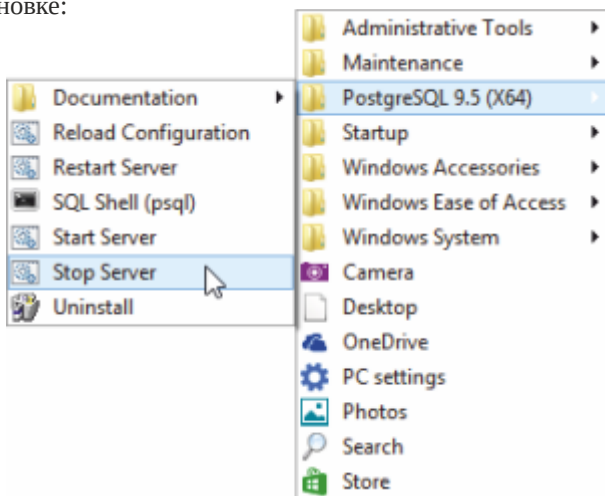
Введите (и подтвердите повторным вводом) пароль пользователя СУБД postgres. Также отметьте флажок «Настроить переменные среды», чтобы подключаться к серверу PostgreSQL под текущим пользователем ОС. Остальные поля можно оставить со значениями по умолчанию.



Если вы планируете установить PostgreSQL только для ознакомительных целей, можно отметить вариант «Использовать параметры по умолчанию», чтобы СУБД не занимала много оперативной памяти.

Управление службой, основные файлы и каталоги

При установке PostgreSQL в вашей системе регистрируется служба «postgresql-X64-9.5» (или «postgresql-X86-9.5» для 32-разрядной версии). Она запускается автоматически при старте компьютера под учетной записью Network Service. При необходимости вы можете изменить параметры службы с помощью стандартных средств Windows. Чтобы временно остановить службу сервера баз данных, выполните программу «Stop Server» из папки в меню «Пуск», которую вы указали при установке:



Для запуска службы там же есть программа «**Start Server**».

Если при запуске службы произошла ошибка, для поиска причины следует заглянуть в журнал сообщений сервера. Он находится в подкаталоге `pg_log` того каталога, который был выбран при установке для баз данных. Обычно это будет:

`C:\Program Files\PostgreSQL\9.5\pg_log.`

Журнал настроен таким образом, чтобы периодически переключаться на запись в новый файл. Найдите актуальный файл либо по дате последнего изменения, либо по имени, которое содержит дату и время переключения.

Есть несколько важных конфигурационных файлов, которые определяют настройки сервера. Они располагаются в каталоге баз данных. Вам не нужно их изменять, чтобы начать знакомство с PostgreSQL, но в реальной работе они непременно потребуются:

- `postgresql.conf` — основной конфигурационный файл, содержащий значения параметров сервера;
- `pg_hba.conf` — файл, определяющий настройки доступа. В целях безопасности по умолчанию доступ должен быть подтвержден паролем и допускается только с локального компьютера.

Загляните в эти файлы — они прекрасно документированы. Теперь мы готовы подключиться к базе данных и попробовать некоторые команды. Переходите к разделу «Пробуем SQL» на с. 19.

Linux

Установка

Если вы используете Linux, то для установки необходимо подключить пакетный репозиторий нашей компании.

Для ОС Debian (в настоящее время поддерживаются версии 6 «Squeeze», 7 «Wheezy» и 8 «Jessie») выполните в терминале следующие команды:

```
$ sudo apt-get install lsb-release  
$ sudo sh -c 'echo "deb http://repo.postgrespro.ru/pgpro-  
9.5/debian $(lsb_release -cs) main" >  
/etc/apt/sources.list.d/postgrespro.list'
```

Для ОС Ubuntu (в настоящее время поддерживаются версии 12.04 «Precise», 14.04 «Trusty», 15.04 «Vivid» и 15.10 «Wily») команды немного отличаются:

```
$ sudo sh -c 'echo "deb http://repo.postgrespro.ru/pgpro-9.5/ubuntu $(lsb_release -cs) main" > /etc/apt/sources.list.d/postgrespro.list'
```

Дальше все одинаково для обеих систем:

```
$ wget --quiet -O - http://repo.postgrespro.ru/pgpro-9.5/keys/GPG-KEY-POSTGRESPRO-95 | sudo apt-key add -  
$ sudo apt-get update
```

Теперь можно приступить к установке. Дистрибутив включает много разных пакетов, но для того, чтобы начать работу, достаточно одного основного:

```
$ sudo apt-get install postgrespro-9.5
```

Как только эта команда выполнится, СУБД PostgreSQL будет установлена, запущена и готова к работе. Чтобы проверить это, выполните команду:

```
$ sudo -u postgres psql -c 'select now();'
```

Если все сделано успешно, в ответ вы должны получить текущее время.

Управление службой, основные файлы и каталоги

При установке PostgreSQL на вашей системе автоматически был создан специальный пользователь postgres, под именем которого работают процессы, обслуживающие сервер, и которому принадлежат все файлы, относящиеся к СУБД. PostgreSQL будет автоматически запускаться при перезагрузке операционной системы. С настройками по умолчанию это не проблема: если вы не работаете с сервером базы

данных, он потребляет совсем немного ресурсов вашей системы. Если вы все-таки захотите отключить автозапуск, вам придется отредактировать файл:

```
/etc/postgresql/9.5/main/start.conf
```

Чтобы временно остановить службу сервера баз данных, выполните команду:

```
$ service postgresql stop 9.5
```

Запустить службу сервера можно командой:

```
$ service postgresql start 9.5
```

Если при запуске службы произошла ошибка, для поиска причины следует просмотреть журнал сообщений сервера, который находится в файле:

```
/var/log/postgresql/postgresql-9.5-main.log
```

Полный список команд управления службой можно получить, выполнив:

```
$ service postgresql
```

Вся информация, которая попадает в базу данных, располагается в файловой системе в каталоге:

```
/var/lib/pgstgresql/9.5/main/ — убедитесь, что у вас достаточно свободного места, если собираетесь хранить много данных.
```

Есть несколько важных конфигурационных файлов, которые определяют настройки сервера. Для начала работы вам не придется их изменять, но в них имеет смысл заглянуть (они прекрасно документированы), потому что в дальнейшем они вам непременно понадобятся:

- `/etc/postgresql/9.5/main/postgresql.conf` — основной конфигурационный файл, содержащий значения параметров сервера;
- `/etc/postgresql/9.5/main/pg_hba.conf` — файл, определяющий настройки доступа. В целях безопасности по умолчанию доступ разрешен только с локального компьютера и только для пользователя ОС postgres.

Самое время подключиться к базе данных и попробовать SQL в деле.

Пробуем SQL

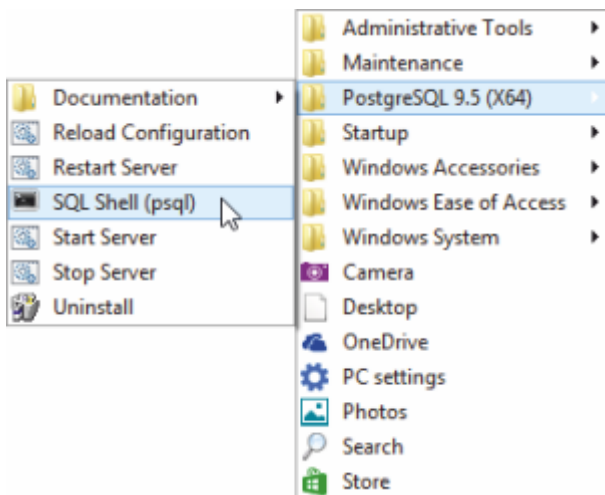
Чтобы подключиться к серверу СУБД и выполнить какие-либо команды, требуется программа-клиент. Мы будем говорить о терминальном клиенте `psql`, работа с которых происходит в режиме командной строки.

`Psql` — стандартный клиент, входящий в любую сборку PostgreSQL, и поэтому он всегда под рукой. `Psql` удобен для повседневных задач по администрированию баз данных, для написания небольших запросов и автоматизации установки программного кода на сервер СУБД. (Если вы привыкли работать с графическими пользовательскими интерфейсами, попробуйте `pgAdmin3` или другие аналогичные программы.)

Чтобы запустить `psql`, в системе Linux выполните команду:

```
| $ sudo -u postgres psql
```


В Windows запустите программу «SQL Shell (psql)» из папки меню «Пуск», которую вы указали при установке:



В ответ на запрос введите пароль пользователя postgres, который вы указали при установке PostgreSQL.

Пользователи Windows могут столкнуться с неправильным отображением русских букв в терминале. Если вместо букв вы видите закорючки, сделайте две вещи:

1. Закройте окно терминала и отредактируйте файл «SQL Shell (psql)» — добавьте в него первой строкой команду:
`chcp 1251`
2. Снова запустите psql и в свойствах окна измените шрифт на «Lucida Console».

В итоге и в одной, и в другой операционной системе вы увидите одинаковое приглашение `postgres=#`. «Postgres» здесь — имя базы данных, к которой вы сейчас подключены. Один сервер PostgreSQL может одновременно обслуживать несколько баз данных, но

одновременно вы работаете только с одной из них.

Дальше мы будем приводить некоторые команды. Вводите только то, что выделено **жирным шрифтом**; приглашение и ответ системы на команду приведены исключительно для удобства.

Давайте создадим новую базу данных с именем test. Выполните:

```
postgres=# create database test;  
CREATE DATABASE
```

Не забудьте про точку с запятой в конце команды — пока PostgreSQL не увидит этот символ, он будет считать, что вы продолжаете ввод (так что команду можно разбить на несколько строк). Теперь переключимся на созданную базу:

```
postgres=# \c test  
You are now connected to database "test" as user  
"postgres".  
test=#
```

Как вы видите, приглашение сменилось на `test=#`.

Команда, которую мы только что ввели, не похожа на SQL — она начинается с обратной косой черты. Так выглядят специальные команды, которые понимает только `psql` (поэтому, если у вас открыт `pgAdmin3` или другое графическое средство, пропускайте все, что начинается на косую черту, или попытайтесь найти аналог). Команд `psql` довольно много и с некоторыми из них мы познакомимся чуть позже, а полный список с кратким описанием можно получить прямо сейчас:

```
test=# \?
```

Поскольку справочная информация довольно объемна, она будет показана с помощью настроенной в операционной системе команды-

пейджера; обычно это `more` или `less`.

Теперь создадим простую таблицу. Введите:

```
test=# create table students(  
test(#   id integer,  
test(#   name text,  
test(#   date_entered date  
test(# );  
CREATE TABLE
```

Обратите внимание, как меняется приглашение `psql`: это подсказка, что ввод команды продолжается на новой строке.

Итак, мы создали таблицу студентов с несколькими столбцами:

- `id` — целочисленный идентификатор студента.
- `name` — имя студента: текст без ограничения длины.
- `date_entered` — дата поступления.

В PostgreSQL встроено большое число различных типов данных, среди которых всегда можно выбрать подходящий.

Что, если мы забудем точный синтаксис команды `create table`? Можно либо заглянуть в документацию, либо попросить справку прямо в `psql`:

```
test=# \help create table
```

Такая справка есть по каждой команде SQL, а полный список команд легко получить с помощью `\help` без параметров.

Таблица должна хранить какие-то значения; давайте вставим в нее несколько строк:

```
test=# insert into students(id, name, date_entered)
values
test=# (1, 'Петр Трофимов', date '1903-09-26'),
test=# (2, 'Евгений Базаров', date '1862-05-20'),
test=# (3, 'Родион Раскольников', date '1866-01-
01');
INSERT 0 3
```

Строки, находящиеся в таблице, можно запросить с помощью команды выборки (символ звездочки обозначает все поля, чтобы не перечислять их явно):

```
test=# select * from students;
```

id	name	date_entered
1	Петр Трофимов	1903-09-26
2	Евгений Базаров	1862-05-20
3	Родион Раскольников	1866-01-01

(3 rows)

Конечно, имеющиеся данные можно изменять. Например, сделаем дату поступления более современной:

```
test=# update students set date_entered = date
'2016-09-01';
UPDATE 3
test=# select * from students;
```

id	name	date_entered
1	Петр Трофимов	2016-09-01
2	Евгений Базаров	2016-09-01
3	Родион Раскольников	2016-09-01

(3 rows)

Другое дело! Теперь добавим еще одну таблицу — список групп:

```
test=# create table groups(  
test(#   id integer,  
test(#   no varchar(10)  
test(# );  
CREATE TABLE
```

Здесь два столбца:

- id — идентификатор группы (целое число),
- no — номер группы (строка длиной до десяти символов).

Нам будет достаточно двух групп:

```
test=# insert into groups(id, no) values  
test-#   (1, 'A-101'),  
test-#   (2, 'A-102');  
INSERT 0 2
```

Сейчас студенты никак не связаны с группами, что, конечно, неправильно. В таблице студентов нам необходимо еще одно поле, о которым мы не подумали — идентификатор группы. К счастью, в уже существующую таблицу можно добавить новый столбец:

```
test=# alter table students add group_id integer;  
ALTER TABLE
```

С помощью команды `psql` всегда можно посмотреть, какие поля определены в таблицах:

```
test=# \d students
          Table «public.students»
   Column      |  Type   | Modifiers
-----+-----+-----
 id             | integer |
 name           | text    |
 date_entered   | date    |
 group_id       | integer |
```

Также можно вспомнить, какие таблицы мы уже создали:

```
test=# \d
          List of relations
 Schema | Name      | Type  | Owner
-----+-----+-----+-----
 public | groups    | table | postgres
 public | students  | table | postgres
(2 rows)
```

Как теперь добавить студентов в группы? Распределим первых двух в группу A-101, а последнего — в A-102:

```
test=# update students set group_id = 1 where id <= 2;
UPDATE 2
test=# update students set group_id = 2 where id > 2;
UPDATE 1
```

Здесь мы использовали ограничение выборки с помощью условия `where`.

Теперь можно формулировать более сложные и интересные запросы, соединяя обе таблицы в одном операторе. Например, выведем имена всех студентов вместе с номерами групп:

```
test=# select s.name, g.no
test-# from students s, groups g
test-# where g.id = s.group_id;
      name      | no
-----+-----
Петр Трофимов   | A-101
Евгений Базаров | A-101
Родион Раскольников | A-102
(3 rows)
```

Или так: сколько студентов учатся в каждой группе?

```
test=# select g.no, count(*)
test-# from students s, groups g
test-# where g.id = s.group_id
test-# group by g.no;
   no   | count
-----+-----
A-101   |      2
A-102   |      1
(2 rows)
```

Язык SQL позволяет легко формулировать самые разные запросы. PostgreSQL качественно поддерживает стандарт SQL, так что, если вы знакомы с каким-либо диалектом этого языка, вам не составит труда использовать его и в этой СУБД. Пробуйте, экспериментируйте! Для того, чтобы завершить сеанс работы, наберите:

```
test=# \q
```

Конечно, мы успели осветить только малую толику того, что необходимо знать о СУБД, но надеемся, что вы убедились: начать использовать PostgreSQL совсем нетрудно. Куда же двигаться дальше?

Обучение и документация

Документация

Для серьезной работы с PostgreSQL не обойтись без чтения документации. Это не только описание всех возможностей СУБД, но и исчерпывающее справочное руководство, которое всегда должно быть под рукой. Читая документацию, вы получаете емкую и точную информацию из первых рук — она написана самими разработчиками и всегда аккуратно поддерживается в актуальном состоянии.

Мы в Postgres Professional перевели на русский язык весь комплект документации — он доступен на нашем сайте:

www.postgrespro.ru/doc

Предпочитающие оригинальную документацию на английском языке найдут ее по адресу:

www.postgresql.org/docs

Учебные курсы

Помимо документации, мы занимаемся и разработкой учебных курсов для администраторов баз данных, разработчиков приложений и тех, кто хочет присоединиться к разработке самой СУБД:

- **DBA1:** Администрирование PostgreSQL. Базовый курс
- **DBA2:** Администрирование PostgreSQL. Расширенный курс
- **DEV1:** Базовый курс для разработчиков PostgreSQL
- **DEV2:** Расширенный курс для разработчиков PostgreSQL
- **Hacking PostgreSQL**

Деление курсов на базовые и расширенные вызвано большим объемом информации, который невозможно изложить и усвоить за несколько дней. Не стоит считать, что базовый курс предназначен только для новичков, а расширенный — только для опытных администраторов или разработчиков. Хотя между курсами есть пересечения по темам, но их не очень много.

Например, базовый трехдневный курс **DBA1** знакомит с PostgreSQL и подробно объясняет базовые понятия администрирования, включая резервное копирование, а в пятидневный **DBA2** охватывает подробности внутреннего устройства СУБД, репликацию, оптимизацию запросов и ряд других тем. Расширенный курс предполагает, что к темам базового курса возвращаться не нужно. Те же принципы лежат в основе деления на две части курса для разработчиков.

Каждый курс представляет собой связанный набор тем, последовательно раскрывающих его содержание (в отличие от документации, где необходимые сведения могут быть разбросаны по разным главам). Каждая тема состоит из теоретической части и практики. Теория — это не только презентация, но в большинстве случаев еще и демонстрация работы на «живой» системе. На практике слушателям предлагается выполнить ряд заданий для закрепления пройденного материала.

Темы поделены таким образом, чтобы теоретическая часть не превышала часа, так как большее время значительно усложняет восприятие материала. Практика, как правило, не превышает 30 минут.

В качестве материалов слушателям выдаются презентации с подробными комментариями к каждому слайду, результат работы демонстрационных скриптов и решения практических заданий.

Для некоммерческого использования материалы курсов доступны на нашем сайте всем желающим.

Курсы для администраторов

DBA1. Администрирование PostgreSQL. Базовый курс

Продолжительность: 3 дня

Предварительные знания

Владение Unix

Минимальные сведения о базах данных и SQL

Какие навыки будут получены

Общее представление о PostgreSQL и его архитектуре

Установка базовая настройка, запуск СУБД

Управление пользователями, данными, доступом

Базовые задачи сопровождения и мониторинга

Резервное копирование и восстановление

Темы

01. Введение в PostgreSQL
02. Архитектура PostgreSQL
03. Установка PostgreSQL
04. Использование psql
05. Базы данных
06. Табличные пространства
07. Системный каталог
08. Основные объекты БД
09. Пользователи и роли
10. Схемы
11. Привилегии
12. Конфигурирование сервера
13. Подключение и аутентификация
14. Мониторинг работы системы
15. Сопровождение PostgreSQL
16. Логическое резервирование
17. Физическое резервирование

Материалы курса **DBA1** (презентации, демонстрации, практические задания, видеозапись лекций) доступны для самостоятельного изучения по адресу: www.postgrespro.ru/education/courses/DBA1

DBA2. Администрирование PostgreSQL. Расширенный курс

Продолжительность: 5 дней

Предварительные знания

Владение Unix

Базовые знания об архитектуре, установке, настройке, обслуживании СУБД

Какие навыки будут получены

Понимание архитектуры PostgreSQL

Мониторинг и настройка базы, решение задач оптимизации производительности

Выполнение задач сопровождения

Резервирование, репликация, клонирование

Темы

Введение

01. Архитектура PostgreSQL

Изоляция и многоверсионность

02. Изоляция транзакций

03. Страницы и версии строк

04. Снимки и блокировки

05. Очистка

06. Автоочистка и заморозка

Журналирование

07. Буферный кэш

08. Упреждающий журнал

09. Контрольная точка

Репликация

10. Файловая репликация

11. Поточковая репликация

12. Переключение на реплику

13. Репликация: варианты

Основы оптимизации

14. Обработка запроса

15. Методы доступа

16. Способы соединения

17. Статистика

18. Использование памяти

19. Оптимизация запросов

Разные темы

20. Локализация

21. Обновление сервера

22. Управление расширениями

23. Внешние данные

24. Секционирование

Материалы курса **DBA2** (презентации, демонстрации, практические задания, видеозапись лекций) доступны для самостоятельного изучения по адресу:

www.postgrespro.ru/education/courses/DBA2

Hacking PostgreSQL

Курс «**Hacking PostgreSQL**» собран из личного опыта разработчиков нашей компании, материалов конференций, статей и вдумчивого чтения документации и исходных кодов. В первую очередь он адресован начинающим разработчикам ядра PostgreSQL, но будет интересен и администраторам, которым иногда приходится обращаться к коду, и просто всем неравнодушным к архитектуре большой системы и желающим узнать, «как это работает на самом деле?».

Предварительные знания

- Знания основ языка SQL, функционала транзакций, индексов и т. п.
- Знание языка Си в объеме, достаточном как минимум для чтения исходных кодов (лучше иметь практические навыки)

- Знакомство с базовыми структурами и алгоритмами

Темы

01. Обзор архитектуры
02. Сообщество PostgreSQL и инструменты разработчика
03. Расширяемость
04. Обзор исходного кода
05. Особенности кода
06. Системный каталог
07. Физическое представление данных
08. Работа с памятью
09. Разделяемая память и блокировки
10. Nodes & Trees
11. Патч ядра PostgreSQL
12. Отладка. Тестирование функциональности и производительности
13. Транзакции. MVCC и VACUUM
14. WAL. И восстановление, и репликация
15. Индексы
16. Тенденции развития СУБД в целом и PostgreSQL в частности

Материалы курса **Hacking PostgreSQL** доступны для самостоятельного изучения по адресу:

www.postgrespro.ru/education/courses/hacking

Курсы для разработчиков

Курсы для разработчиков находятся в процессе подготовки и должны появиться в самое ближайшее время.

Где пройти обучение

Если вы хотите пройти обучение по перечисленным курсам в специализированном учебном центре, под руководством опытного преподавателя и с получением сертификата, то есть и такая возможность.

Мы авторизовали несколько известных учебных центров, которые читают наши курсы. Их список можно посмотреть здесь:

www.postgrespro.ru/education

О компании

Компания **Postgres Professional** была основана в 2015 году и объединила всех ключевых российских разработчиков, вклад которых в развитие PostgreSQL существенен и признан мировым сообществом. Компания является российским вендором PostgreSQL и выполняет разработки на уровне ядра СУБД и расширений, оказывает услуги по проектированию и поддержке прикладных систем, миграции на PostgreSQL.

Компания уделяет большое внимание образовательной деятельности, организует крупнейшую международную конференцию PgConf.Russia в Москве и принимает участие в конференциях по всему миру.

Услуги

Промышленные решения на основе PostgreSQL

- Проектирование и участие в создании критичных высоконагруженных систем.
- Оптимизация конфигурации СУБД.
- Консультирование по вопросам использования СУБД в промышленных системах.
- Аудит систем заказчика по вопросам использования СУБД, проектирования баз данных, высокопроизводительных и отказоустойчивых архитектур.
- Внедрение СУБД PostgreSQL.

Вендорская техническая поддержка

- Вторая и третья линии техподдержки СУБД PostgreSQL в режиме 24x7.

- Круглосуточная поддержка опытными администраторами: мониторинг, восстановление работоспособности, анализ непредвиденных обстоятельств, обеспечение производительности.
- Исправление ошибок, обнаруженных в СУБД и ее расширениях.

Миграция прикладных систем

- Анализ имеющихся прикладных систем и определение сложности их миграции с других СУБД на PostgreSQL.
- Определение архитектуры нового решения и требований к доработкам прикладных систем.
- Миграция систем на СУБД PostgreSQL, в том числе действующих систем под нагрузкой.
- Поддержка прикладных разработчиков в процессе миграции.

Разработка на уровне ядра и расширений

- Заказные разработки на уровне ядра СУБД и ее модулей расширения.
- Создание специальных модулей расширения для решения прикладных и системных задач заказчика.
- Создание кастомизированных версий СУБД в интересах заказчика.
- Публикация изменений в основную версию кода СУБД.

Организация обучения

- Обучение администраторов баз данных работе с СУБД PostgreSQL.
- Обучение разработчиков и архитекторов прикладных систем особенностям СУБД PostgreSQL и эффективному использованию ее достоинств.
- Информирование о новой функциональности и важных изменениях в новых версиях.
- Проведение семинаров по разбору проектов заказчиков.

Postgres Professional – российский вендор PostgreSQL. Компания является членом международного сообщества PostgreSQL и приглашает разработчиков к участию в Open Source проектах.

В **Postgres Professional** работают ведущие российские разработчики PostgreSQL (major contributor), признанные международным сообществом.

На сегодняшний день **Postgres Professional** оказывает следующие услуги: поддержка, миграция, разработка, консалтинг, обучение. Компания работает с крупными заказчиками различных отраслей.

opensource@postgrespro.ru

www.POSTGRESPRO.ru

+7 (495) 150-06-91

«PostgreSQL для начинающих»
П. Лузанов, Е. Рогов.
вёрстка А. Климковский
(с) Postgres Professional 2016

Иллюстрация на обложке: WIND

