Содержание

PostgreSQL

Версия PostgreSQL

Поддерживаются версии PostgreSQL начиная с 13 версии с поддержкой ICU.

Для однообразной сортировки в PostgreSQL на Windows и Linux, и для сортировки как в SQL Server, используется провайдер icu, который использует внешнюю библиотеку ICU. Локали ICU можно использовать, только если поддержка ICU была включена в конфигурации сборки PostgreSQL.

Для установки PostgreSQL требуется сборка PostgreSQL начиная с 13 версии с поддержкой ICU.

Ограничения PostgreSQL

Основные ограничения PostgreSQL, которые отличаются от SQL Server (https://postgrespro.ru/docs/postgresql/14/limits):

1. Максимальная длина идентификатора 63 байта (имена таблиц, ключей, индексов и других объектов базы данных (https://postgrespro.ru/docs/postgresql/14/sql-syntax-lexical#SQL-SYNTAX-IDENTIFIERS)).
2. Максимальное кол-во столбцов в индексе 32.
3. По умолчанию поиск по строке (https://postgrespro.ru/docs/postgresql/14/datatype-character) чувствителен к регистру. Одно из решений использовать тип citext (https://postgrespro.ru/docs/postgresql/14/citext).

Настройка PostgreSQL

Надо настроить PostgreSQL, по умолчанию PostgreSQL использует 256 МБ памяти.

Для генерации настроек по конфигурации сервера надо зайти на https://pgtune.leopard.in.ua/ и заполнить параметры.

В поле DB Type надо выбрать «Mixed type of application».

Надо взять скрипт с вкладки ALTER SYSTEM для изменения параметров PostgreSQL, запустить его на сервере PostgreSQL и перезапустить сервис PostgreSQL после выполнения скрипта.

После запуска скриптов надо перезапустить сервис PostgreSQL:

```
sudo service postgresql restart
```

=== Пример для PostgreSQL 13: ===
DB Version: 13
OS Type: linux

DB Type: mixed
Total Memory (RAM): 40 GB
CPUs num: 16
Connections num: 1000
Data Storage: ssd

```
SQL
ALTER SYSTEM SET max_connections = '1000';
ALTER SYSTEM SET shared_buffers = '10GB';
ALTER SYSTEM SET effective_cache_size = '30GB';
ALTER SYSTEM SET maintenance_work_mem = '2GB';
ALTER SYSTEM SET checkpoint_completion_target = '0.9';
ALTER SYSTEM SET wal_buffers = '16MB';
ALTER SYSTEM SET default_statistics_target = '100';
ALTER SYSTEM SET random_page_cost = '1.1';
ALTER SYSTEM SET effective_io_concurrency = '200';
ALTER SYSTEM SET work_mem = '1310kB';
ALTER SYSTEM SET min_wal_size = '1GB';
ALTER SYSTEM SET max_wal_size = '4GB';
ALTER SYSTEM SET max_worker_processes = '16';
ALTER SYSTEM SET max_parallel_workers_per_gather = '4';
ALTER SYSTEM SET max_parallel_workers = '16';
ALTER SYSTEM SET max_parallel_maintenance_workers = '4';
```

После запуска скрипта с установкой настроек нужно перезапустить сервис PostgreSQL:

```
sudo service postgresql restart
```

Конвертация базы из SQL Server в PostgreSQL и обратно

Настройка приложения

Статьи по настройкам быстродействию PostgreSQL через строку подключения:

https://www.roji.org/prepared-statements-in-npgsql-3-2

https://www.npgsql.org/doc/prepare.html

В файле client.config:

В разделе <connectionStrings> заменить

```
<add name="db" providerName="System.Data.SqlClient"
connectionString="Server=.\SQL;Database=sl_3_24;User
Id=user;Password=123;Connection Timeout=3000;Max Pool Size=2000;"/>
<add name="dbCubes" providerName="System.Data.SqlClient"
connectionString="Server=.\SQL;Database=sl_3_24;User
Id=user;Password=123;Connection Timeout=300;Max Pool Size=2000;"/>
```

```
<add name="busDb" providerName="System.Data.SqlClient"
connectionString="Server=.\SQL;Database=sl_3_24;User
Id=user;Password=123;Connection Timeout=300;Max Pool Size=2000;"/>
```

на

```
<add name="db" providerName="Npgsql"
connectionString="Server=localhost;Port=5432;Database=sl_3_24;User
Id=postgres;Password=123;Pooling=true;CommandTimeout=7200;Timeout=500;MaxPoo
lSize=500;Max Auto Prepare=100;Auto Prepare Min Usages=3;"/>
<add name="dbCubes" providerName="Npgsql"
connectionString="Server=localhost;Port=5432;Database=sl_3_24;User
Id=postgres;Password=123;Pooling=true;CommandTimeout=7200;Timeout=500;MaxPoo
lSize=500;Max Auto Prepare=100;Auto Prepare Min Usages=3;"/>
<add name="busDb" providerName="Npgsql"
connectionString="Server=localhost;Port=5432;Database=sl_3_24;User
Id=postgres;Password=123;Pooling=true;CommandTimeout=7200;Timeout=500;MaxPoo
lSize=500;Max Auto Prepare=100;Auto Prepare Min Usages=3;"/>
```

В разделе <appSettings> заменить

```
<add key="BusTransportType" value="SqlServer" />
```

на

```
<add key="BusTransportType" value="PostgreSql" />
```

В разделе

```
<unity xmlns="http://schemas.microsoft.com/practices/2010/unity">
    <container name="db">
```

заменить

```
<register type="SmartSuite.Data.Persistence.StorageProviderFactory,
SmartSuite.Data"
mapTo="SmartSuite.Data.Persistence.SQLServer.SqlStorageProviderFactory,
SmartSuite.Data" />
<register type="SL.App.ServerObjects.IServerObjectsProvider, SL.App"
mapTo="SL.App.ServerObjects.SqlServer.ServerObjectsProvider, SL.SqlServer"
/>
<register type="P2M.Cubes.Data.IDirectoryDataManager, smcorelib"
mapTo="P2M.Cubes.Data.SQL.DirectoryDataManager, smcorelib" />
<register type="P2M.Cubes.Data.IPivotDataManager, smcorelib"
mapTo="P2M.Cubes.Data.SQL.PivotDataManager, smcorelib" />
```

на

```
<register type="SmartSuite.Data.Persistence.StorageProviderFactory,
SmartSuite.Data" mapTo="SmartSuite.PostgreSQL.PgStorageProviderFactory,
```

```
SL.PostgreSQL" />
<register type="SL.App.ServerObjects.IServerObjectsProvider, SL.App"
mapTo="SL.App.ServerObjects.PostgreSQL.ServerObjectsProvider, SL.PostgreSQL"
/>
<register type="P2M.Cubes.Data.IDirectoryDataManager, smcorelib"
mapTo="SL.PostgreSQL.Data.DirectoryDataManager, SL.PostgreSQL" />
<register type="P2M.Cubes.Data.IPivotDataManager, smcorelib"
mapTo="SL.PostgreSQL.Data.PivotDataManager, SL.PostgreSQL" />
```

Настройка приложения для использования PostgreSQL с PgBouncer

При использовании PgBouncer надо добавить в строку подключения параметр (https://www.npgsql.org/doc/compatibility.html?q=pgbouncer):

```
No Reset On Close=true
```

или отключить пул соединений в приложении в строке подключения:

```
Pooling=false
```

Если используем PgBouncer в режиме *transaction pooling*, надо отключать prepared statements в приложение в строке подключения (надо убрать параметры «Max Auto Prepare» и «Auto Prepare Min Usages» или выставить «Max Auto Prepare=0» в строке подключения):

https://www.pgbouncer.org/faq.html#:~:text=to%20DEALLOCATE%20ALL%3B-,How%20to%20use%20prepared%20statements%20with%20transaction%20pooling%3F,prepared%20statements%20in%20the%20client

Установка в Ubuntu

Установка локали

https://www.postgresql.org/download/linux/ubuntu/

https://wiki.postgresql.org/wiki/Apt

Установка локали https://askubuntu.com/questions/76013/how-do-i-add-locale-to-ubuntu-server

В Linux в терминале выполнить:

```
locale -a
```

Проверить, что в списке есть ru_RU.UTF-8, если есть — запустить (если нет - надо устанавливать русскую локаль):

```
sudo locale-gen ru_RU
sudo locale-gen ru_RU.UTF-8
sudo update-locale
```

Если не сработало (зависит от дистрибутива Linux)

```
sudo locale-gen ru_RU
```

то запустить

```
sudo dpkg-reconfigure locales
```

и выбрать нужные локали.

На базе postgres выполнить:

```
SELECT pg_import_system_collations('pg_catalog');
```

Перезапустить сервис PostgreSQL:

```
sudo service postgresql restart
```

Настройка PostgreSQL

1. Пересоздание кластера PostgreSQL с другой локалью:

http://breys.ru/4379.html

```
sudo pg_dropcluster --stop 13 main
sudo pg_createcluster --locale ru_RU.utf8 --start 13 main
sudo pg_ctlcluster 13 main start
```

2. Как отредакировать файл от имени админа

https://vitux.com/how-to-open-and-edit-files-and-folders-in-ubuntu-desktop-as-an-administrator/

3. Доступ извне (только для разработки, не использовать на продакшене)

https://blog.rvalitov.ru/development/yii2/postgresql-v-virtualbox/

Добавить строчку в самый низ в файл /etc/postgresql//main/pg_hba.conf

```
host    all             all             0.0.0.0/0               trust
```

В файле /etc/postgresql//main/postgresql.conf добавляем или меняем строчку `listen_addresses = '*'` (убедитесь, что она раскомментирована)

Перезагружаем сервис

```
sudo service postgresql restart
```

4. Настройка VirtualBox для доступа к PostgreSQL с хоста

https://serverfault.com/questions/225155/virtualbox-how-to-set-up-networking-so-both-host-and-guest-can-access-internet

5. Запуск psql
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-ubuntu-18-04-ru

```
sudo -u postgres psql
```

Установка и развертывание на Astra Linux Special Edition x.7

Что нужно знать

Установка незащищенной версии выполняется с подключенным компонентом astra-ce расширенного (extended) репозитория, установка защищенной версии выполняется без подключения этого компонента.

На момент написания данной статьи в составе Astra Linux Special Edition РУСБ.10015-01 (очередное обновление 1.7) доступны следующие версии СУБД PostgreSQL:

- версия 11.10-astra.se5 - защищенная версия в основном репозитории (на установочном диске);
- версия 11.10-astra.se13 - защищенная версия в основном репозитории (оперативное обновление №1);
- версия 11.12-astra.ce5 - незащищенная версия в компоненте astra-ce расширенного репозитория до оперативного обновления 1.7.2;
- версия 14+240astra4 - - незащищенная версия в компоненте astra-ce расширенного репозитория начиная с оперативного обновления 1.7.2.

> При установке СУБД PostgreSQL из компонента astra-ce расширенного репозитория защищенная версия СУБД, доработанная для взаимодействия с КСЗ Astra Linux Special Edition будет заменена на стандартную версию СУБД.
>
> Подробнее про структуру и использование репозиториев см. Репозитории Astra Linux Special Edition x.7: структура, особенности подключения и использования.

> При установке незащищенной СУБД PostgreSQL версии 14 из компонента astra-ce расширенного репозитория 1.7.2 при наличии ранее установленной СУБД PostgreSQL версии 11 необходимо перед запуском нового кластера вручную удалить ранее

> установленные кластеры версии 11:
>
> ```
> pg_dropcluster 13 <имя_старого_кластера>
> ```

Установка пакетов

1. Подключить репозитории:

- основной репозиторий и актуальное оперативное обновление основного репозитория;
- только для установки незащищенной версии PostgreSQL - актуальное оперативное обновление расширенного репозитория, включая компонент astra-ce. Для установки защищенной версии подключение компонента astra-ce расширенного репозитория недопустимо.

2. Обновить список пакетов:

```
sudo apt update
```

3. Опционально: установить актуальные обновления:

```
sudo apt dist-upgrade
```

4. Проверить доступные версии:

```
apt policy postgresql-13
```

5. Установить пакет postgresql-13 старшей доступной версии:

```
sudo apt install postgresql-13
```

или установить пакет postgresql-13 указанной версии:

```
sudo apt install postgresql-13=<номер_версии>
```

6. Убедиться, что служба postgresql запустилась:

```
systemctl status postgresql
```

```
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor
preset: enabled)
   Active: active (exited) since Fri 2021-09-10 12:48:20 MSK; 1min 26s ago
Main PID: 4338 (code=exited, status=0/SUCCESS)
   Tasks: 0 (limit: 4637)
   Memory: 0B
   CGroup: /system.slice/postgresql.service
```

Первичная настройка СУБД PostgreSQL

1. Выполнить вход в сессию служебного пользователя postgres:

```
sudo su - postgres
```

Работая в сессии служебного пользователя postgres:

- Установить пароль администратора СУБД:

```
psql -c "alter user postgres with password '<указать_пароль>'"
```

- Вместо текста <пароль> указать устанавливаемый пароль;
- Пароль заключается в одинарные кавычки;
- Вся команда заключается в двойные кавычки.

- Завершить работу в сессии служебного пользователя postgres:

```
exit
```

2. Настроить удаленный доступ к СУБД, для чего в конфигурационном файле /etc/postgresql/13/main/postgresql.conf проверить и установить параметр listen_addresses:

- Значение по умолчанию — служба postgresql подключена ко всем сетевым интерфейсам:

```
listen_addresses = '*'
```

Допустимые значения:

- Служба postgresql подключена ко всем сетевым интерфейсам IPv4:

```
listen_addresses = '0.0.0.0'
```

- Служба postgresql подключена ко всем сетевым интерфейсам IPv6:

```
listen_addresses = '::'
```

- Разделенный запятыми список IP-адресов сетевых интерфейсов, к которым будет подключена служба:

```
listen_addresses = '192.168.1.2,10.0.0.2'
```

Допускается использовать пустой список, тогда подключение будет возможно только через сокеты UNIX (подробнее см. документацию СУБД).

3. Если в конфигурацию были внесены изменения, то для того, чтобы сделанные изменения вступили в силу перезапустить службу postgresql:

```
sudo systemctl restart postgresql
```

4. Проверить, к каким сетевым портам и интерфейсам подключена служба postgresql, можно

командой:

```
ss -tunelp | grep uid:`id -u postgres`
```

```
tcp     LISTEN   0        1024              0.0.0.0:5432           0.0.0.0:*
uid:107 ino:32947 sk:5 <->
tcp     LISTEN   0        1024                 [::]:5432              [::]:*
uid:107 ino:32948 sk:9 v6only:1 <->
```

Приведен вывод команды для службы, настроенной по умолчанию (параметр listen_addresses = '*', служба работает с портом 5432 на всех доступных сетевых интерфейсах IPv4 и IPv6);

5. Настроить активные сетевые экраны, разрешив доступ к сетевому порту postgresql (по умолчанию - порт 5432):

- Для сетевого экрана ufw:

```
sudo ufw allow 5432/tcp
```

- Для сетевого экрана firewalld:

```
sudo firewall-cmd --add-service=postgresql --zone=internal --permanent
```

Первичное тестирование работоспособности СУБД

1. Выполнить вход в сессию служебного пользователя postgres:

```
sudo su - postgres
```

Работая в сессии служебного пользователя postgres:

- Добавить тестового пользователя СУБД test_user1:

```
createuser test_user1
```

- Добавить тестовую базу данных test_db, указав в качестве её владельца тестового пользователя:

```
createdb test_db -O test_user1
```

- Установить пароль тестового пользователя:

```
psql -c "alter user test_user1 with password '<указать_пароль>'"
```

- Подключиться к созданной тестовой базе данных и войти в сессию СУБД:

```
psql test_db
```

```
psql (11.12 (Debian 11.12-astra.ce5))
Введите "help", чтобы получить справку.
test_db=#
```

Дальнейшие команды выполняются в сессии СУБД:

- Создать таблицу и добавить в нее данные:

```
CREATE TABLE test_table ( id INT,first_name text, last_name text );
```

```
CREATE TABLE
```

- Добавить в таблицу данные:

```
INSERT INTO test_table (id,first_name,last_name) VALUES (2,'Иван','Иванов');
```

```
INSERT 0 1
```

- Вывести табличные данные:

```
SELECT * FROM test_table;
```

```
 id | first_name | last_name
----+------------+-----------
  1 | Иван       | Иванов
(1 строка)
```

- Выйти из сессии СУБД:

```
exit
```

- Удалить тестовую базу данных:

```
dropdb test_db
```

- Завершить работу в сессии служебного пользователя postgres:

```
exit
```

Бэкап и восстановление базы

https://postgrespro.ru/docs/postgresql/13/backup

Бэкап базы делается утилитой pg_dump , нужно использовать pg_dump той же версии, которая установлена на сервере PostgreSQL:

```
  pg_dump.exe --file "D:\\DbBackups\\sl_3_24.bak" --host "sl_3_24" --port
"5432" --username "postgres" --password --verbose --format=c --blobs
"sl_3_24"
```

Для восстановления базы надо создать пустую базу скриптом:

```
  SQL
  CREATE DATABASE "sl_3_24"
  WITH
```

```
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'ru_RU.utf8' -- для Linux
  LC_CTYPE = 'ru_RU.utf8' -- для Linux
  --LC_COLLATE = 'Russian_Russia.1251' -- для Windows
  --LC_CTYPE = 'Russian_Russia.1251' -- для Windows
  --TEMPLATE template0 -- нужно раскомметировать, если COLLATION в базе postgres
(SHOW lc_collate;) отличается от значения в LC_COLLATE
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1;

  ALTER DATABASE "sl_3_24" SET search_path TO "$user", public, dbo;

  CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
```

И запустить утилиту `pg_restore` , в созданную пустую базу:

```
 pg_restore.exe --host "localhost" --port "5432" --username "postgres" --
password --dbname "sl_3_24" --verbose "D:\DbBackups\sl_3_24.bak"
```

Можно установить search_path на системном уровне, чтобы при каждом восстановление базы не надо было это настраивать отдельно:

```
  SQL
  ALTER SYSTEM SET search_path
  TO "$user", public, dbo;

  SELECT pg_reload_conf();
```

Если нужно выдать права пользователю на базу (https://postgrespro.ru/docs/postgresql/13/sql-grant):

```
  SQL
  ALTER DATABASE <base> OWNER TO <user>;
  GRANT postgres TO <user>;
```

Полезные ссылки по PostgreSQL

1. Документация по PostgreSQL: https://www.postgresql.org/docs/
2. Документация по PostgreSQL на русском: https://postgrespro.ru/docs/
3. Онлайн утилита генерации настроек для конфигурации сервера PostgreSQL: https://pgtune.leopard.in.ua/
4. PgBouncer FAQ: https://www.pgbouncer.org/faq.html