

УТВЕРЖДЕН  
643.72410666.00067-05 98 01-ЛУ

**ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ  
БАЗАМИ ДАННЫХ «ЈАТОВА»**

**Руководство по настройке. Часть 13.  
Поддержка платформы 1С**

**643.72410666.00067-05 98 01-13**

Листов 27

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2023

Литера О<sub>1</sub>

## АННОТАЦИЯ

В документе приведены сведения, необходимые для установки и эксплуатации компонентов совместимости с 1С, таких как:

- fasttrun. Версия компонента – 2.0;
- fulleq. Версия компонента – 2.0;
- mchar. Версия компонента – 2.2.1;
- online\_analyze. Версия компонента – 1.0;
- plantuner. Версия компонента – 1.0.

Данные компоненты входят в пакет 1С\_support, который необходим для поддержки платформы 1С:Предприятие.

Настоящее руководство предназначено для администраторов СУБД «Jatoba».



Все примеры в данном документе приведены для СУБД «Jatoba» версии ядра 4.x, для других версий все шаги выполняются аналогично, разница состоит в именах директорий.

Например, СУБД «Jatoba» версии 5.x по умолчанию устанавливается в директорию:

- ОС Windows – «C:\Program Files\GIS\Jatoba\5\bin»;
- ОС Linux – «/usr/jatoba-5/bin».



### Важная информация

Для сертифицированной версии СУБД «Jatoba» поддерживается работа только на ОС, указанных в формуляре на поставку!

Степени важности примечаний, применяемые в документе:



**Важная информация** – указания, требующие особого внимания



**Дополнительная информация** – указания, позволяющие упростить работу с изделием

## СОДЕРЖАНИЕ

1. Назначение компонента.....	5
1.1. Условия применения.....	5
2. Установка и настройка.....	6
2.1. Установка компонента «1С_support» .....	6
2.2. Компонент «fasttran».....	7
2.3. Компонент «fulleq» .....	10
2.4. Компонент «mchar».....	12
2.5. Компонент «planter».....	13
2.6. Компонент «online_analyze» .....	16
3. Настройки для связки 1С и СУБД «Jatoba» .....	18
3.1. Настройки при нехватке памяти или остановке процесса ООМ-киллером .....	18
3.2. Настройка конфигурационного файла СУБД postgresql.conf .....	18
3.3. Безусловные параметры конфигурационного файла СУБД postgresql.conf.....	20
4. Процедура создания новой базы 1С в связке с СУБД «Jatoba».....	23
Перечень сокращений.....	26

## 1. НАЗНАЧЕНИЕ КОМПОНЕНТА

В пакете «1С\_support» устанавливаются компоненты совместимости с 1С:

- `fasttrun` – предоставляет транзакционную функцию для усечения временных таблиц, предотвращающую разрастание каталога `pg_class`;
- `fulleq` – предоставляет дополнительный оператор равенства для совместимости с Microsoft SQL Server (MS SQL);
- `mchar` – предоставляет дополнительные типы данных для совместимости с MS SQL;
- `online_analyze` – добавляет поддержку указаний для планировщика, позволяющих отключать или подключать определенные индексы при выполнении запроса;
- `plantuner` – предоставляет набор функций, которые немедленно обновляют статистику после операций INSERT, UPDATE, DELETE или SELECT INTO в целевых таблицах.

### 1.1. Условия применения

Пакет «1С\_support» может использоваться совместно с СУБД «Jatoba» версий 4.x и выше, под управлением ОС GNU/Linux.

Отдельная установка расширений не требуется, при установке продукта 1С они установятся автоматически.

Ограничений по совместимости с другими компонентами СУБД нет.

## 2. УСТАНОВКА И НАСТРОЙКА

Установка компонентов должна производиться от имени пользователя, обладающего административными привилегиями в системе. Данный модуль штатным образом может быть установлен только с СУБД «Jatoba» (см. документ «Защищенная система управления базами данных «Jatoba». Руководство по установке).

### 2.1. Установка компонента «1С\_support»

Компонент устанавливается в составе СУБД «Jatoba». Его возможно установить при первичной установке либо доустановить.

Установку компонента возможно провести двумя способами:

- 1) установка из локального репозитория (CDROM) – производится из файлов, записанных на компакт-диск или скопированных с него;
- 2) установка непосредственно из deb/rpm-файлов – производится опционально, по усмотрению пользователя.

Компонент выполнен в виде отдельного deb или rpm-пакета. Установка компонента осуществляется средствами пакетного менеджера ОС. Для разных типов пакетных менеджеров команда установки немного отличается. Ниже приведены основные типы:

– для систем на основе пакетного менеджера APT (к таким системам относятся все ОС семейства Debian, использующие deb-пакеты) команда установки следующая:

```
apt-get install jatoba4-1csupport
```

– для систем на основе пакетных менеджеров YUM/DNF (к таким системам относятся все ОС семейства RedHat и вышедшие из нее, использующие rpm-пакеты) команда установки следующая:

```
yum install jatoba4-1csupport
```

Отдельного уточнения требуют операционные системы ALT Linux и openSUSE.

– ALT Linux использует пакетный менеджер APT, но распространяется в виде rpm-пакетов и для нее команда установки выглядит аналогично Debian:

```
apt-get install jatoba4-1csupport
```

– openSUSE также распространяется в виде rpm-пакетов, но использует собственный пакетный менеджер zypper, для нее команда установки выглядит следующим образом:

```
zypper install jatoba4-1csupport
```

Установка компонента в составе других версий СУБД «Jatoba» осуществляется аналогично. Отличие будет только в номере версии СУБД, в составе которой он распространяется. Например, jatoba5-1csupport и т.п.

Удаление модуля также осуществляется средствами пакетного менеджера ОС. Вместо команды install нужно использовать соответствующую данному пакетному менеджеру команду удаления (remove, purge, erase и т.п.).

Для получения детальной информации по пакетному менеджеру рекомендуется обратиться к документации по ОС.

Компоненты выполнены в виде расширений СУБД.

## 2.2. Компонент «fasttran»

Устанавливается расширение SQL-командой:

```
CREATE EXTENSION fasttrun;
```

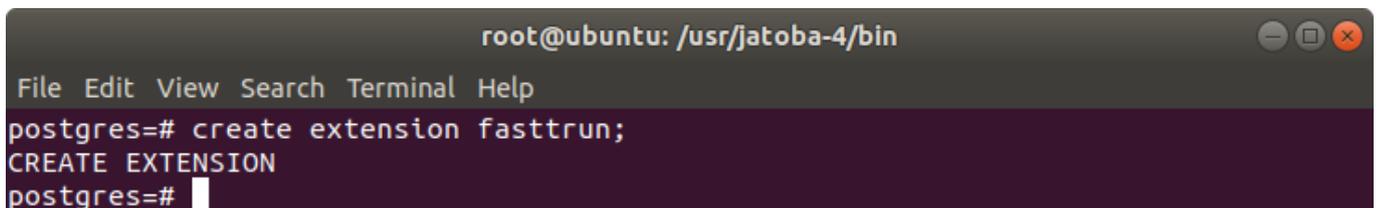
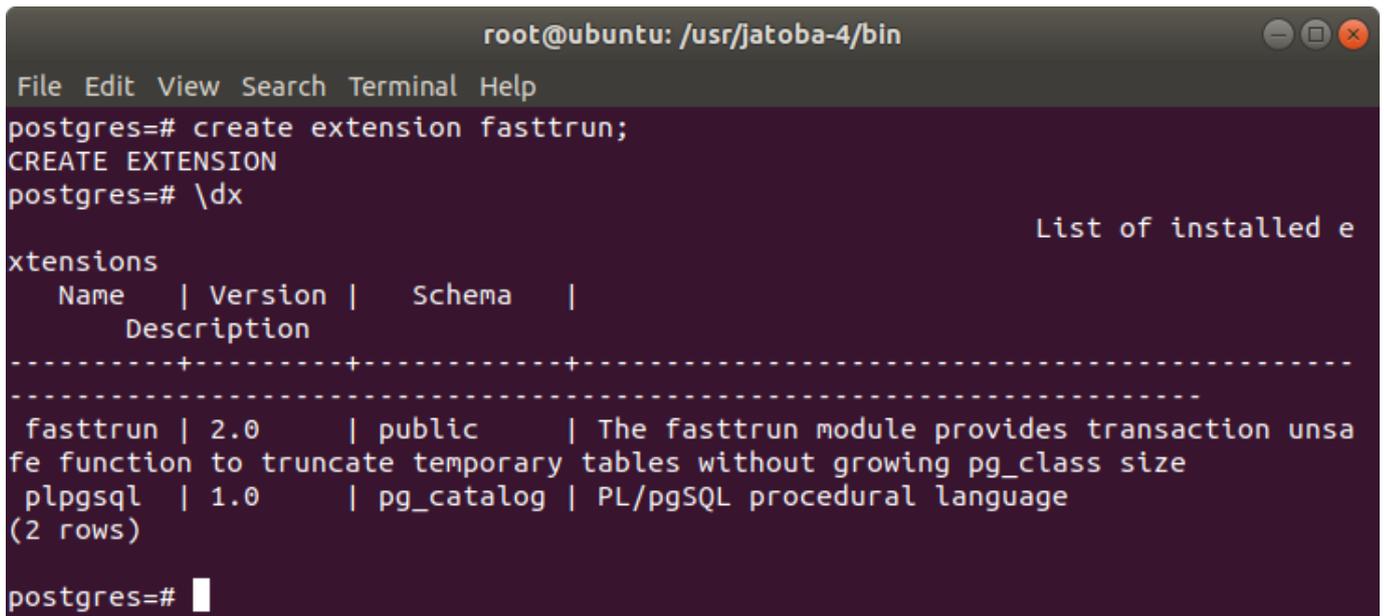


Рисунок 2.1 – Создание расширения

Проверить установленные расширения возможно SQL-командой:

```
\dx
```



```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# create extension fasttrun;
CREATE EXTENSION
postgres=# \dx
List of installed e
xtensions
  Name   | Version | Schema |
  Description
-----+-----+-----+-----
fasttrun | 2.0     | public | The fasttrun module provides transaction unsa
fe function to truncate temporary tables without growing pg_class size
plpgsql  | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)
postgres=#
```

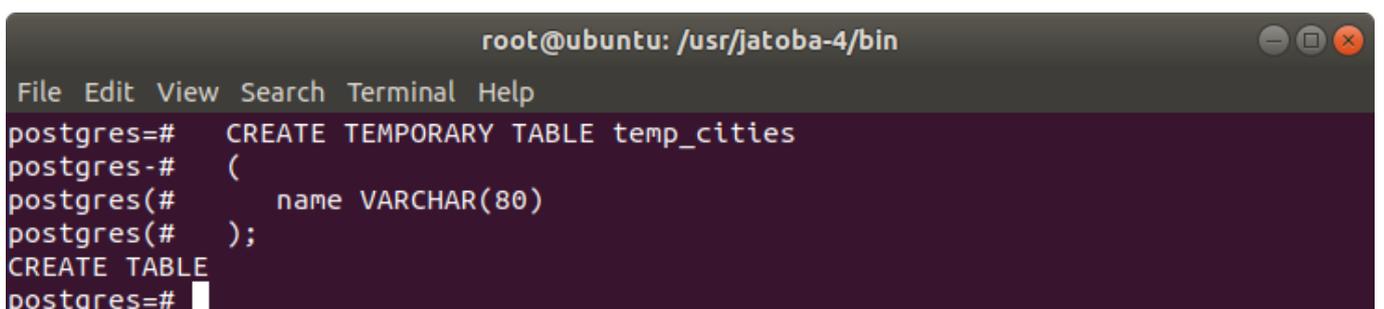
Рисунок 2.2 – Проверка установленных расширений

Для примера и проверки работы компонента необходимо:

- создать временную таблицу;
- внести в нее данные;
- проверить содержимое таблицы;
- вызвать функцию;
- проверить содержимое таблицы.

Создать временную таблицу SQL-командой:

```
CREATE TEMPORARY TABLE temp_cities
(
  name VARCHAR(80)
);
```

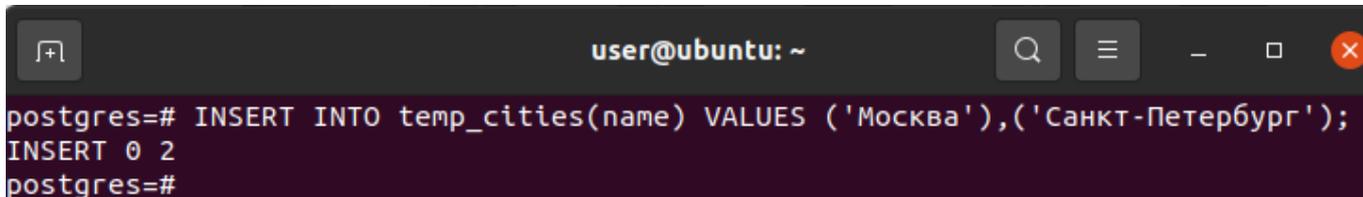


```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# CREATE TEMPORARY TABLE temp_cities
postgres=# (
postgres=# name VARCHAR(80)
postgres=# );
CREATE TABLE
postgres=#
```

Рисунок 2.3 – Создание временной таблицы

Внести в нее данные SQL-командой:

```
INSERT INTO temp_cities(name) VALUES ('Москва'), ('Санкт-Петербург');
```

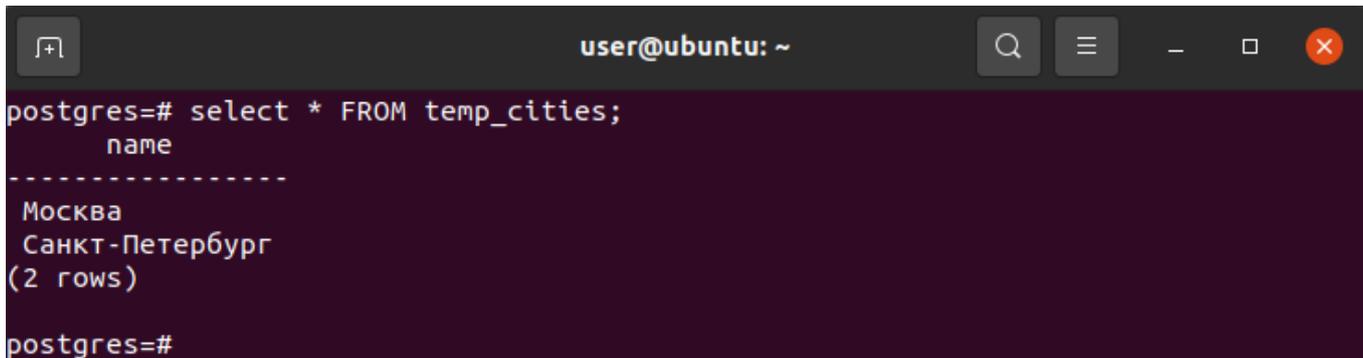


```
user@ubuntu: ~  
postgres=# INSERT INTO temp_cities(name) VALUES ('Москва'),('Санкт-Петербург');  
INSERT 0 2  
postgres=#
```

Рисунок 2.4 – Внесение данных в временную таблицу

Просмотреть содержание временной таблицы SQL-командой:

```
SELECT * FROM temp_cities;
```

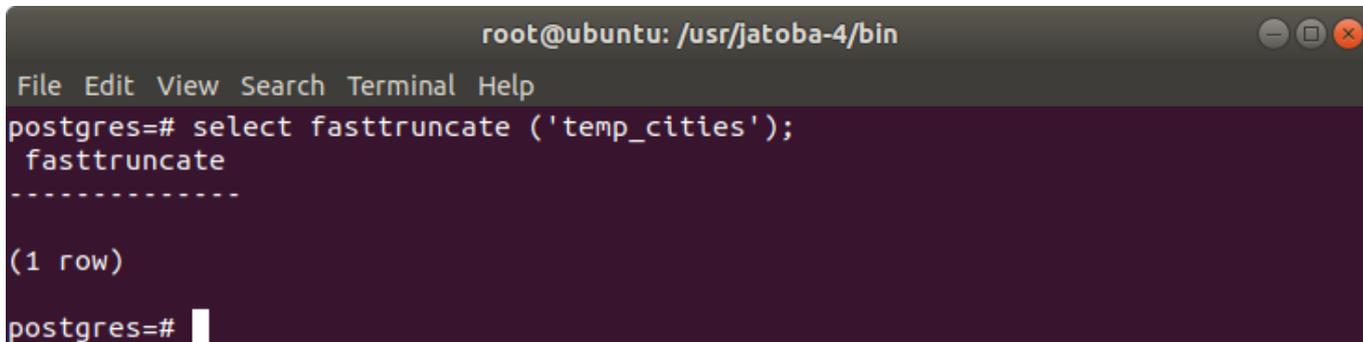


```
user@ubuntu: ~  
postgres=# select * FROM temp_cities;  
      name  
-----  
Москва  
Санкт-Петербург  
(2 rows)  
postgres=#
```

Рисунок 2.5 – Просмотр содержания таблицы

Применить функцию к временной таблице 'temp\_cities' SQL-командой:

```
SELECT fasttruncate ('temp_cities');
```



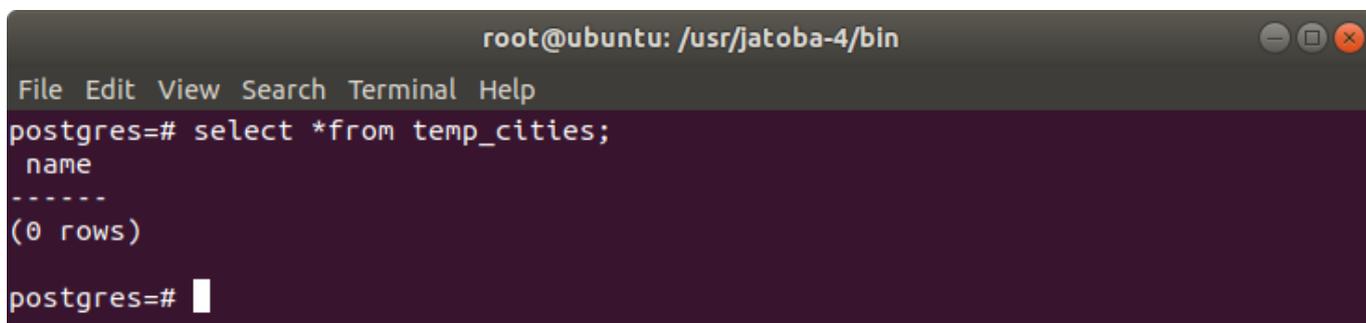
```
root@ubuntu: /usr/jatoba-4/bin  
File Edit View Search Terminal Help  
postgres=# select fasttruncate ('temp_cities');  
fasttruncate  
-----  
(1 row)  
postgres=#
```

Рисунок 2.6 – Применение функции fasttruncate

После применения функции данные в таблице обнуляются.

Просмотреть содержание временной таблицы SQL-командой:

```
SELECT *from temp_cities;
```



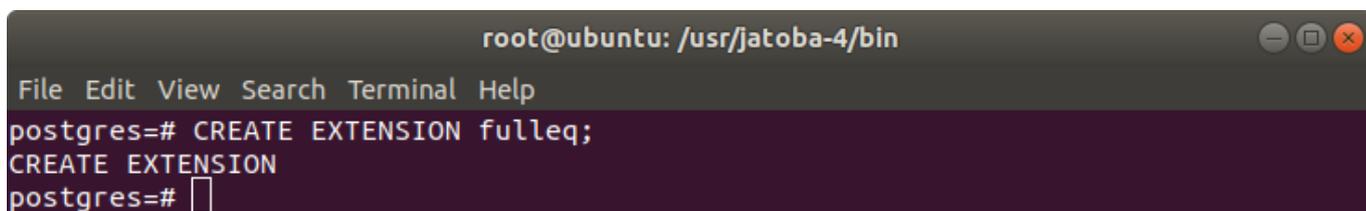
```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# select *from temp_cities;
 name
-----
(0 rows)
postgres=#
```

Рисунок 2.7 – Проверка содержания временной таблицы после применения функции `fasttruncate`

### 2.3. Компонент «fulleq»

Устанавливается расширение SQL-командой:

```
CREATE EXTENSION fulleq;
```



```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# CREATE EXTENSION fulleq;
CREATE EXTENSION
postgres=#
```

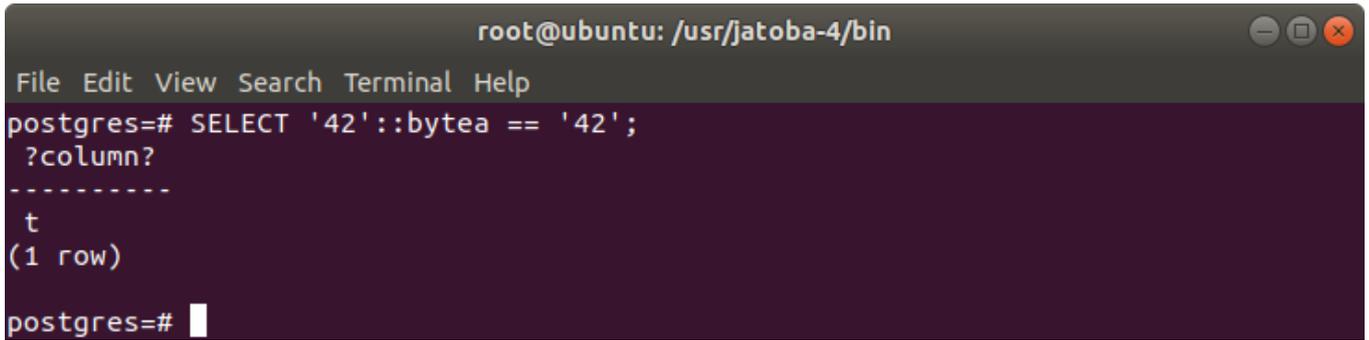
Рисунок 2.8 – Установка расширения `fulleq`

Примеры тестирования работы компонента приведен ниже.

Проверка типа равенства байтового и символьного типов.

```
SELECT '42'::bytea == '42';
```

В результате будет выведено значение «TRUE», что означает равенство.

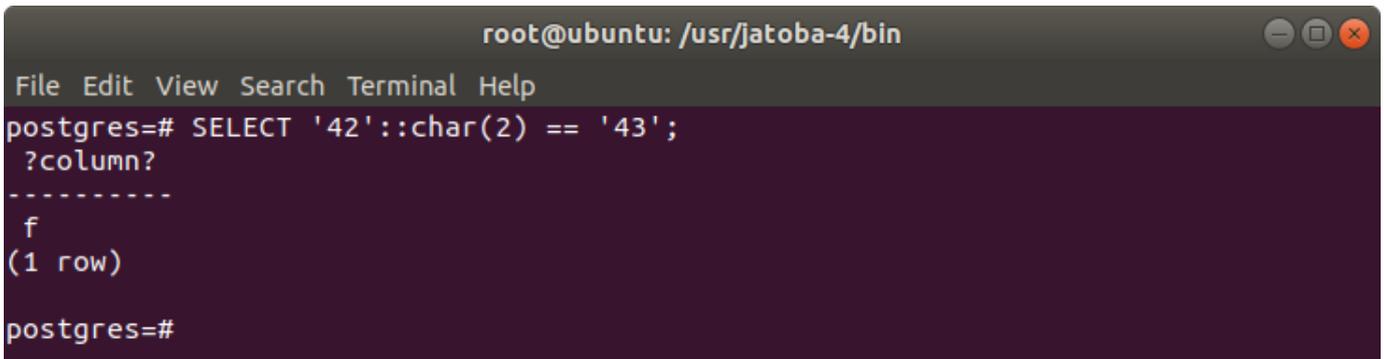


```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# SELECT '42'::bytea == '42';
?column?
-----
t
(1 row)
postgres=#
```

Рисунок 2.9 – Проверка типа равенства байтового и символьного типов  
Проверка типа равенства символьных типов.

```
SELECT '42'::char(2) == '43';
```

В результате будет выведено значение «FALSE».

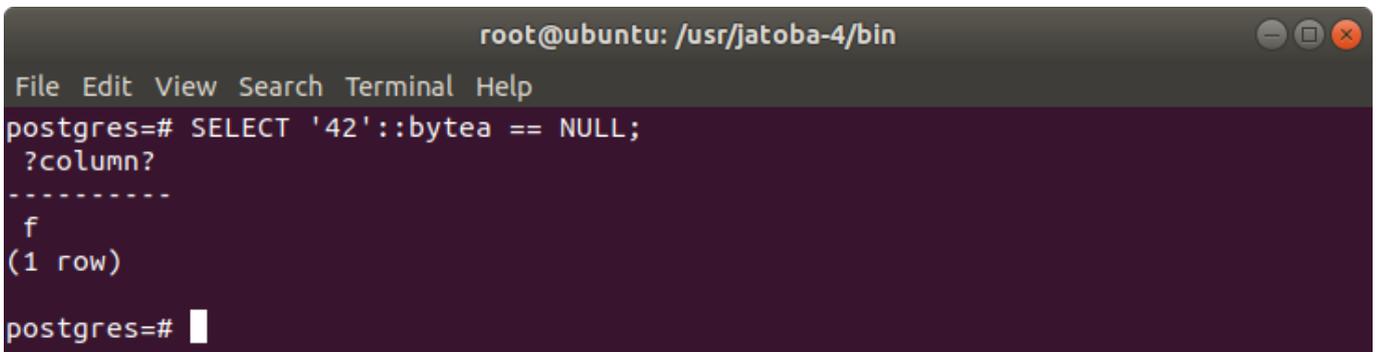


```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# SELECT '42'::char(2) == '43';
?column?
-----
f
(1 row)
postgres=#
```

Рисунок 2.10 – Проверка типа равенства символьных типов  
Проверка типа равенства с «NULL» (абстрактной величиной).

```
SELECT '42'::bytea == NULL;
```

В результате будет выведено значение «FALSE», т.к. это не пустое значение.



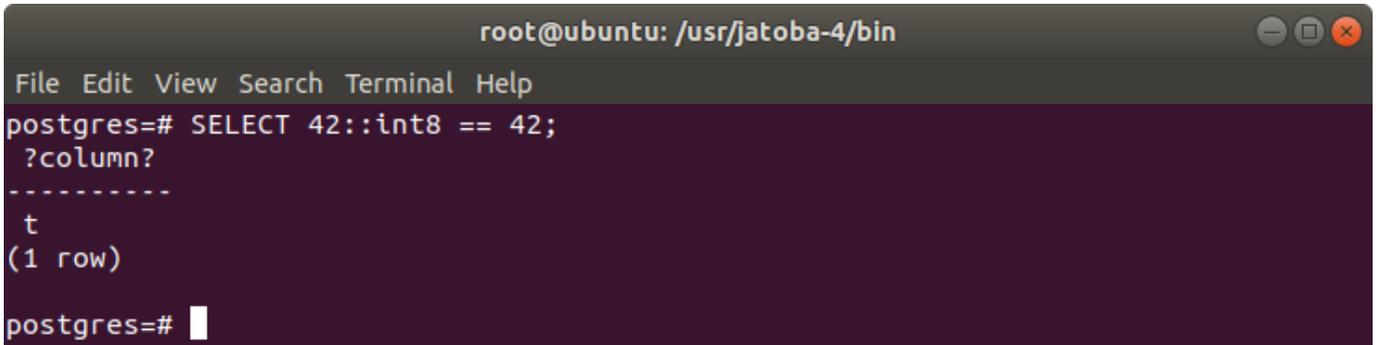
```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# SELECT '42'::bytea == NULL;
?column?
-----
f
(1 row)
postgres=#
```

Рисунок 2.11 – Проверка типа равенства с NULL

Проверка целых типов.

```
SELECT 42::int8 == 42;
```

В результате будет выведено значение «TRUE», что означает равенство.



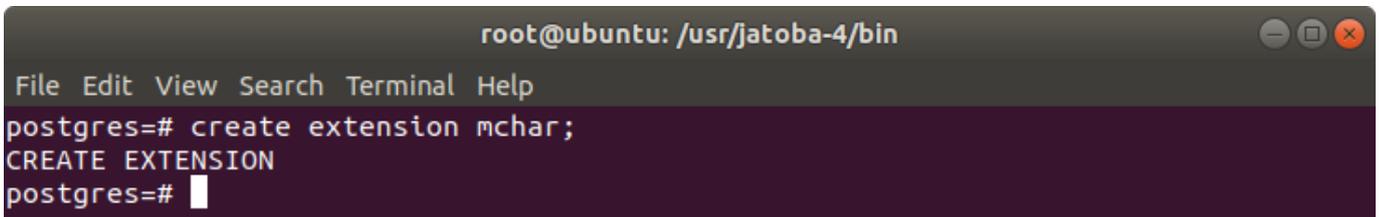
```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# SELECT 42::int8 == 42;
?column?
-----
 t
(1 row)
postgres=#
```

Рисунок 2.12 – Проверка целых типов

## 2.4. Компонент «mchar»

Устанавливается расширение SQL-командой:

```
CREATE EXTENSION mchar;
```



```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# create extension mchar;
CREATE EXTENSION
postgres=#
```

Рисунок 2.13 – Установка расширения mchar

Компонент содержит типы:

- mchar (аналог типа char в MS SQL);
- mvarchar (аналог типа varchar в MS SQL).

Для типов mchar и mvarchar определяются:

- функция length(str);
- функция substr(str, pos[, length]);
- оператор ||, применяемый для конкатенации любых аргументов mchar и mvarchar;
- набор операторов: <, <=, =, >= и > для сравнения символов без учета регистра;

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

- набор операторов: &<, &<=, &=, &>= и &> для сравнения с учетом регистра.

Также типы поддерживают:

- неявное приведение между типами mchar и mvvarchar;
- В-деревья и хеш-индексы;
- оператор LIKE [ESCAPE];
- оператор SIMILAR TO [ESCAPE];
- оператор ~ (регулярные выражения POSIX);
- индексы для оператора LIKE.

## 2.5. Компонент «planter»

Для активации компонента загрузить разделяемую библиотеку «plantuner» в сеансе psql, как показано на рисунке 2.14.

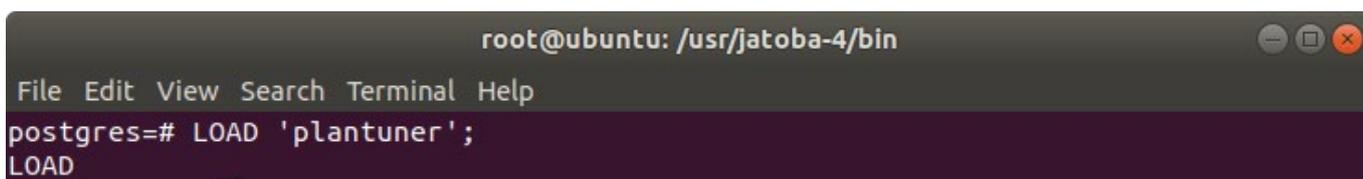


Рисунок 2.14 – Загрузка разделяемой библиотеки

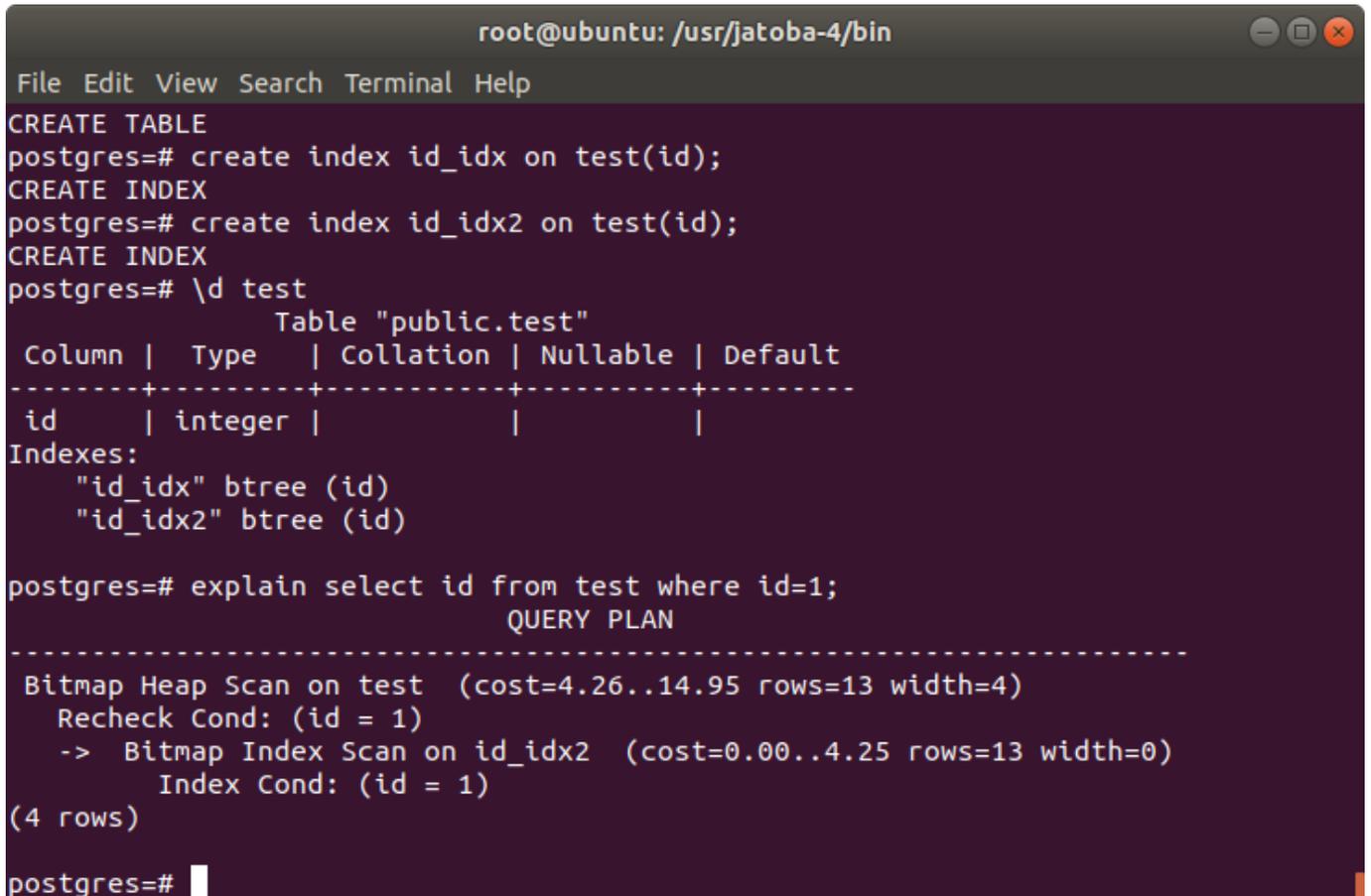
Либо указать ее в параметре shared\_preload\_libraries в postgresql.conf.

Представленный пример состоит из нескольких частей.

В первой части примера, загружается разделяемая библиотека «plantuner», создается тестовая таблица и индексы для нее. После чего создается план запроса.

```
=# LOAD 'plantuner';  
=# create table test(id int);  
=# create index id_idx on test(id);  
=# create index id_idx2 on test(id);  
=# \d test  
      Table "public.test"  
  Column | Type      | Modifiers  
-----+-----+-----  
   id    | integer  |  
Indexes:  
   "id_idx" btree (id)  
   "id_idx2" btree (id)
```

```
=# explain select id from test where id=1;
          QUERY PLAN
-----
Bitmap Heap Scan on test  (cost=4.34..15.03 rows=12 width=4)
  Recheck Cond: (id = 1)
    -> Bitmap Index Scan on id_idx2  (cost=0.00..4.34 rows=12
width=0)
      Index Cond: (id = 1)
(4 rows)
```



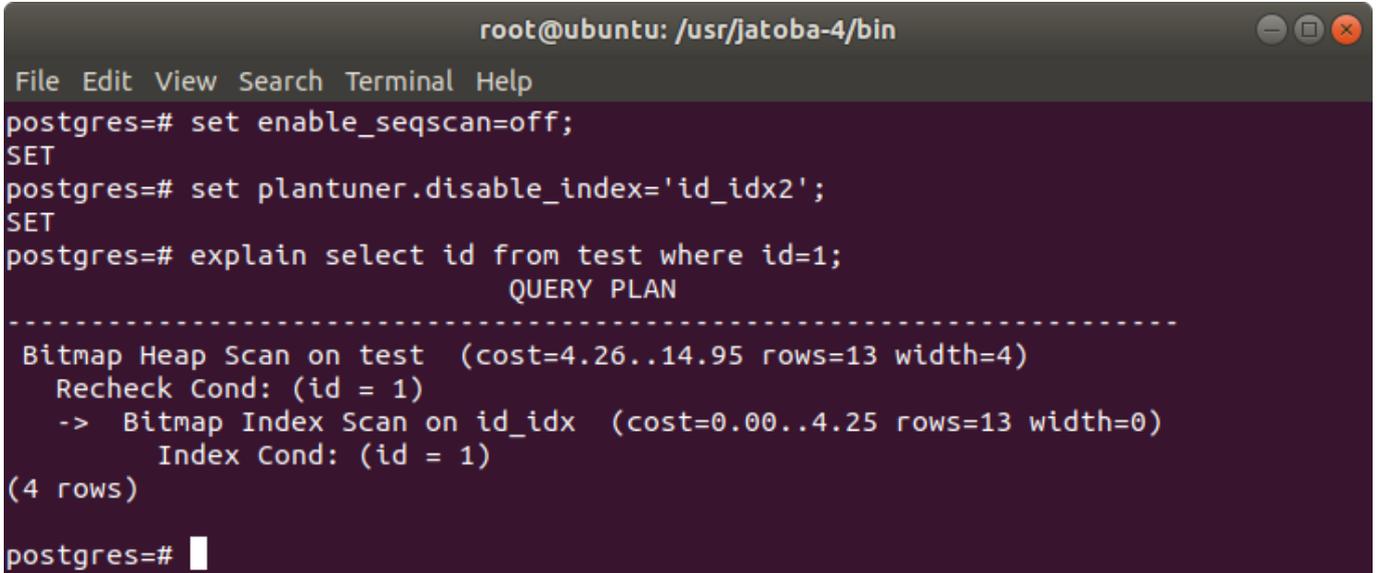
```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
CREATE TABLE
postgres=# create index id_idx on test(id);
CREATE INDEX
postgres=# create index id_idx2 on test(id);
CREATE INDEX
postgres=# \d test
          Table "public.test"
  Column | Type   | Collation | Nullable | Default
-----+-----+-----+-----+-----
 id      | integer |           |          |
Indexes:
 "id_idx" btree (id)
 "id_idx2" btree (id)
postgres=# explain select id from test where id=1;
          QUERY PLAN
-----
Bitmap Heap Scan on test  (cost=4.26..14.95 rows=13 width=4)
  Recheck Cond: (id = 1)
    -> Bitmap Index Scan on id_idx2  (cost=0.00..4.25 rows=13 width=0)
      Index Cond: (id = 1)
(4 rows)
postgres=#
```

Рисунок 2.15 – Создание плана

Во второй части примера отключается индекс 'id\_idx2' из плана запросов.

```
=# set enable_seqscan=off;
=# set plantuner.disable_index='id_idx2';
=# explain select id from test where id=1;
          QUERY PLAN
-----
Bitmap Heap Scan on test  (cost=4.34..15.03 rows=12
width=4)
  Recheck Cond: (id = 1)
```

```
-> Bitmap Index Scan on id_idx (cost=0.00..4.34
rows=12 width=0)
      Index Cond: (id = 1)
(4 rows)
```



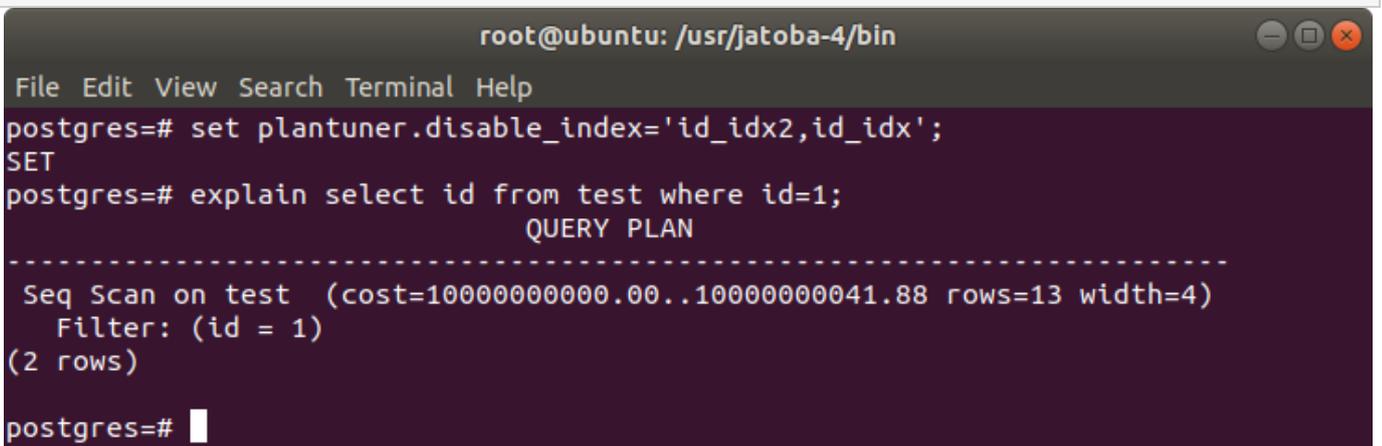
```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# set enable_seqscan=off;
SET
postgres=# set plantuner.disable_index='id_idx2';
SET
postgres=# explain select id from test where id=1;
          QUERY PLAN
-----
Bitmap Heap Scan on test (cost=4.26..14.95 rows=13 width=4)
  Recheck Cond: (id = 1)
    -> Bitmap Index Scan on id_idx (cost=0.00..4.25 rows=13 width=0)
        Index Cond: (id = 1)
(4 rows)

postgres=#
```

Рисунок 2.16 – Отключение индекса 'id\_idx2' из плана запросов

В третьей части примера исключаются из плана запросов индексы id\_idx2,id\_idx

```
=# set plantuner.disable_index='id_idx2,id_idx';
=# explain select id from test where id=1;
          QUERY PLAN
-----
Seq Scan on test (cost=10000000000.00..100000000040.00
rows=12 width=4)
  Filter: (id = 1)
(2 rows)
```



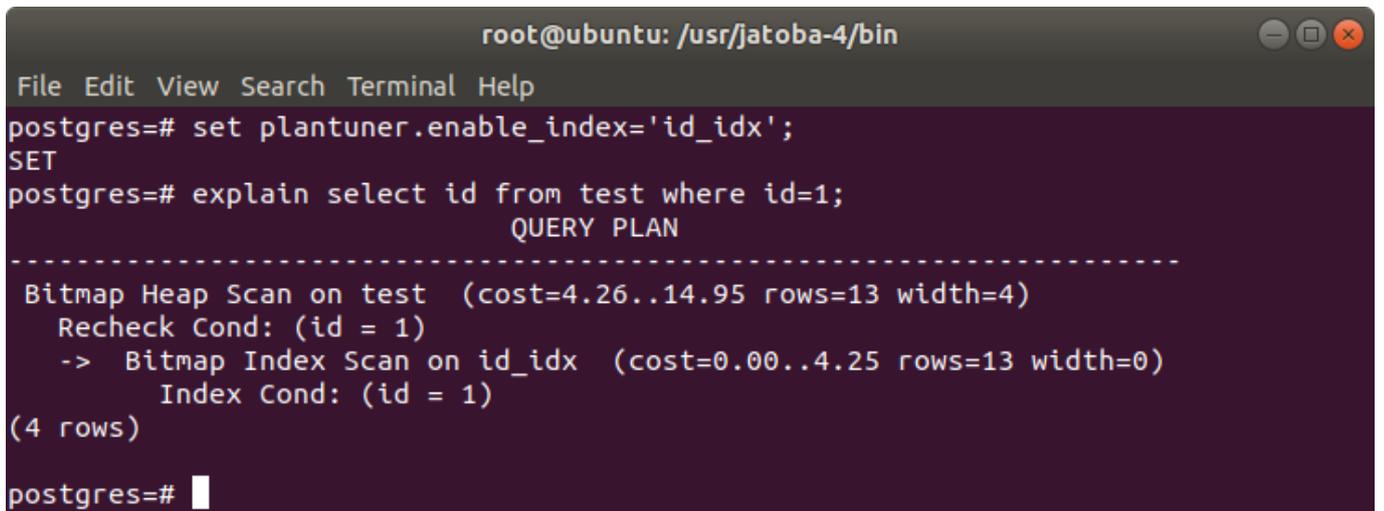
```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# set plantuner.disable_index='id_idx2,id_idx';
SET
postgres=# explain select id from test where id=1;
          QUERY PLAN
-----
Seq Scan on test (cost=10000000000.00..100000000041.88 rows=13 width=4)
  Filter: (id = 1)
(2 rows)

postgres=#
```

Рисунок 2.17 – Исключение из плана запросов индексы 'id\_idx2,id\_idx'

В четвертой части примера индекс 'id\_idx' включается в план запросов.

```
=# set plantuner.enable_index='id_idx';
=# explain select id from test where id=1;
          QUERY PLAN
-----
Bitmap Heap Scan on test  (cost=4.34..15.03 rows=12
width=4)
  Recheck Cond: (id = 1)
    -> Bitmap Index Scan on id_idx  (cost=0.00..4.34
rows=12 width=0)
      Index Cond: (id = 1)
(4 rows)
```



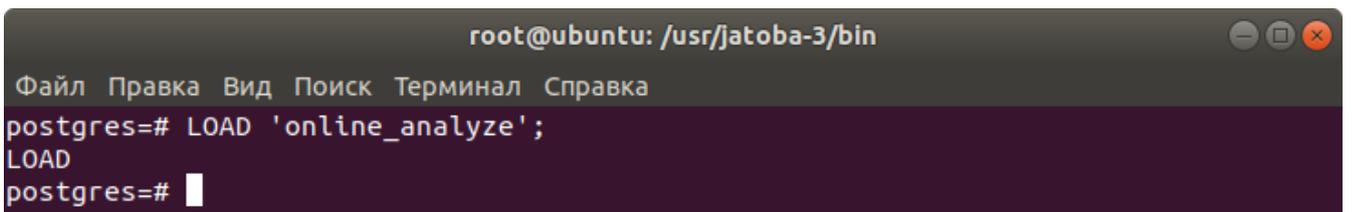
```
root@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=# set plantuner.enable_index='id_idx';
SET
postgres=# explain select id from test where id=1;
          QUERY PLAN
-----
Bitmap Heap Scan on test  (cost=4.26..14.95 rows=13 width=4)
  Recheck Cond: (id = 1)
    -> Bitmap Index Scan on id_idx  (cost=0.00..4.25 rows=13 width=0)
      Index Cond: (id = 1)
(4 rows)
postgres=#
```

Рисунок 2.18 – Включение индекса 'id\_idx' в план запросов

## 2.6. Компонент «online\_analyze»

После установки пакета 1C\_support использование компонента «online\_analyze» доступно после загрузки разделяемой библиотеки. Библиотека загружается SQL-командой:

```
LOAD 'online_analyze';
```

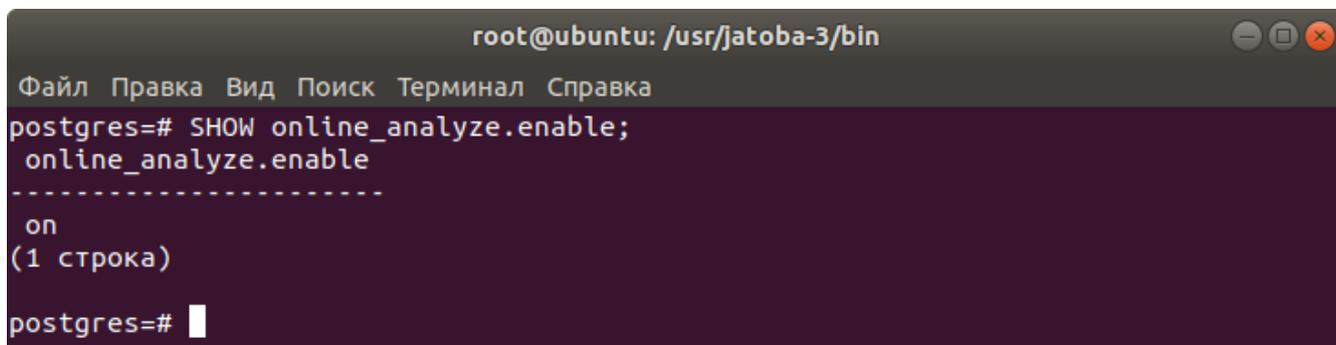


```
root@ubuntu: /usr/jatoba-3/bin
Файл Правка Вид Поиск Терминал Справка
postgres=# LOAD 'online_analyze';
LOAD
postgres=#
```

Рисунок 2.19 – Загрузка разделяемой библиотеки

По умолчанию компонент находится в состоянии «on» и проверить его состояние возможно SQL-командой:

```
SHOW online_analyze.enable;
```

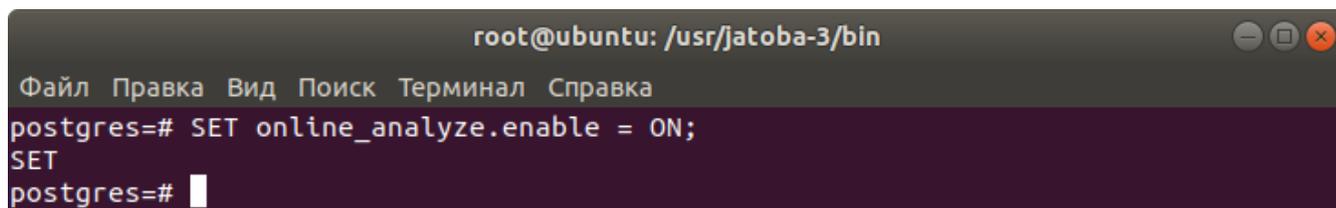


```
root@ubuntu: /usr/jatoba-3/bin
Файл Правка Вид Поиск Терминал Справка
postgres=# SHOW online_analyze.enable;
online_analyze.enable
-----
on
(1 строка)
postgres=#
```

Рисунок 2.20 – Проверка состояния компонента

Инициализация компонента проводится SQL-командой:

```
SET online_analyze.enable = ON;
```

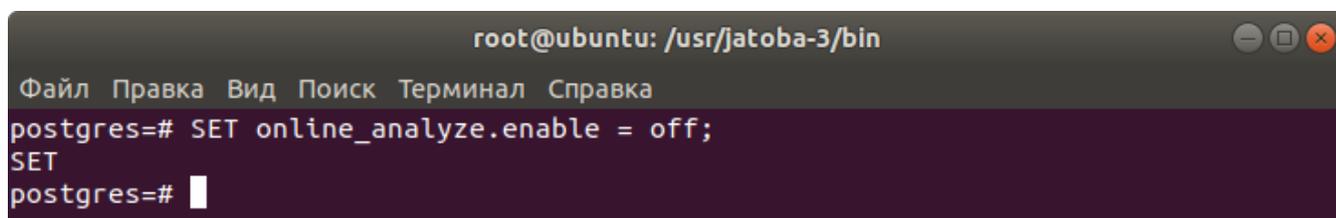


```
root@ubuntu: /usr/jatoba-3/bin
Файл Правка Вид Поиск Терминал Справка
postgres=# SET online_analyze.enable = ON;
SET
postgres=#
```

Рисунок 2.21 – Команда инициализации компонента

Изменить установку компонента возможно SQL-командой:

```
SET online_analyze.enable = off;
```



```
root@ubuntu: /usr/jatoba-3/bin
Файл Правка Вид Поиск Терминал Справка
postgres=# SET online_analyze.enable = off;
SET
postgres=#
```

Рисунок 2.22 – Команда отключения компонента

### 3. НАСТРОЙКИ ДЛЯ СВЯЗКИ 1С И СУБД «ЯТОВА»

Перед первой инициализацией СУБД «Jatoba», в консоли ОС выполнить команду:

```
#dpkg-reconfigure locales (выбрать ru_RU.UTF-8)  
#export PGSETUP_INITDB_OPTIONS='--encoding=UTF8 --lc-  
collate="ru_RU.UTF-8" --lc-ctype="ru_RU.UTF-8"'
```

#### 3.1. Настройки при нехватке памяти или остановке процесса ООМ-киллером

В конфигурационном файле /etc/sysctl.conf установить параметры:

```
vm.overcommit_memory = 2
```

Для включения следующего параметра.

```
vm.overcommit_ratio = число
```

Число обозначающее процент RAM, не больше которого будет выделено на один процесс. Должно быть больше shared\_buffers.

В консоли ОС выполнить команду обновления конфигурации ОС:

```
#sudo sysctl -system
```

В свойствах службы jatoba в systemd проверить параметры в разделе раздел Service:

```
OOMScoreAdjust=-1000  
Environment=PG_OOM_ADJUST_FILE=/proc/self/oom_score_adj  
Environment=PG_OOM_ADJUST_VALUE=0
```

#### 3.2. Настройка конфигурационного файла СУБД postgresql.conf

В конфигурационном файле СУБД postgresql.conf проверить условные параметры:

```
shared_buffers = RAM/4  
work_mem = 256MB
```

Параметры можно проверить SQL-запросом в СУБД:

```
SELECT sum(temp_files) as temp_files,  
pg_size_pretty(sum(temp_bytes)) as temp_size from  
pg_stat_database;
```

Параметры возможно уменьшить при необходимости.

```
wal_level = minimal
```

Устанавливается значение «minimal» при отсутствии репликации или «replica», если она обязательна.

```
synchronous_commit = off
```

Значение устанавливается только при резервировании питания.

```
commit_delay = 1000  
commit_siblings = 5
```

Параметры целесообразны, если интенсивность транзакций превосходит 1000 TPS.

```
max_wal_senders = 0
```

Значение «0» устанавливается, если параметр wal\_level = minimal или количество реплик более одной.

```
random_page_cost = 1.1
```

Устанавливаются параметры для:

- SSD – 1.1-1.3;
- RAID – 1.5-2.0;
- единичного SATA – 4.0.

```
effective_cache_size =
```

Параметр устанавливается в размере 75% от RAM.

```
from_collapse_limit = 8
```

```
join_collapse_limit = 8
```

Как правило значение устанавливается больше 8.

Например. Значение устанавливается 20, если частота CPU ~4ГГц.

```
autovacuum_max_workers =
```

Значение устанавливается в размере 50% от количества ядер.

```
vacuum_cost_limit =
```

Значение высчитывается по формуле `autovacuum_max_workers * 100`.

### 3.3. Безусловные параметры конфигурационного файла СУБД `postgresql.conf`

В конфигурационном проверьте следующие безусловные параметры:

- `temp_buffers = 256MB;`
- `maintenance_work_mem = 1024MB;`
- `max_files_per_process = 10000;`
- `max_parallel_workers_per_gather = 0;`
- `max_wal_size = 8GB;`
- `min_wal_size = 4GB;`
- `checkpoint_timeout = 15min;`
- `enable_bitmapscan = on;`
- `enable_hashagg = on;`
- `enable_hashjoin = on;`
- `enable_indexscan = on;`
- `enable_indexonlyscan = on;`
- `enable_material = on;`
- `enable_mergejoin = off;`
- `enable_nestloop = on;`

- enable\_seqscan = on;
- enable\_sort = on;
- enable\_tidscan = on;
- seq\_page\_cost = 1.0;
- log\_min\_duration\_statement = 5000;
- autovacuum = on;
- log\_autovacuum\_min\_duration = 5000;
- autovacuum\_naptime = 20s;
- autovacuum\_vacuum\_scale\_factor = 0.01;
- autovacuum\_analyze\_scale\_factor = 0.005;
- max\_locks\_per\_transaction = 256;
- escape\_string\_warning = off;
- standard\_conforming\_strings = off;
- online\_analyze.threshold = 50;
- online\_analyze.scale\_factor = 0.1;
- online\_analyze.enable = on;
- online\_analyze.verbose = off;
- online\_analyze.min\_interval = 10000;
- online\_analyze.table\_type = 'temporary';
- plantuner.fix\_empty\_table = on;
- shared\_preload\_libraries = 'online\_analyze, plantuner';
- ssl = 'off';
- row\_security = off;
- listen\_addresses = '\*' или сетевой интерфейс, куда придут за базой

- отключить Transparent Huge Pages (смотреть для своей системы);
- `backslash_quote = safe_encoding`;
- `temp_file_limit` = не нулевое значение;
- `default_statistics_target=100`;

1С с 8.3.13 сама умеет выполнять функцию ANALYZE, поэтому высокое значение нежелательно.

#### 4. ПРОЦЕДУРА СОЗДАНИЯ НОВОЙ БАЗЫ 1С В СВЯЗКЕ С СУБД «ЯТОВА»

Для выполнения процедуры создания новой базы 1С в связке с СУБД «ЯТОВА» должны быть выполнены действия описанные

- в разделе 3 «Настройки для связки 1С и СУБД «Jatoba»;
- в п. 2.1 «Установка компонента «1С\_support».

Процесс создания БД состоит из следующих шагов:

- 1) В клиенте «1С:Предприятие» нажать кнопку добавить;

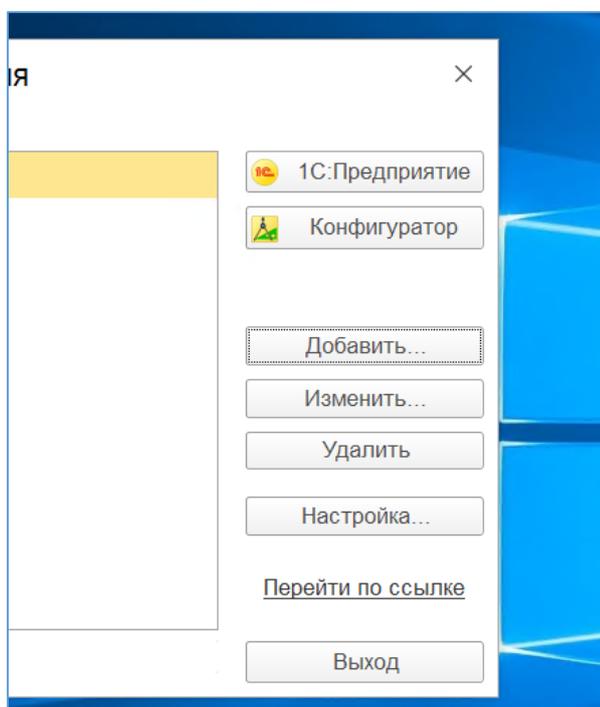


Рисунок 4.1 – Кнопка «Добавить» в клиенте «1С:Предприятие»

- 2) «Создание новой информационной базы» на основе шаблона или без него;

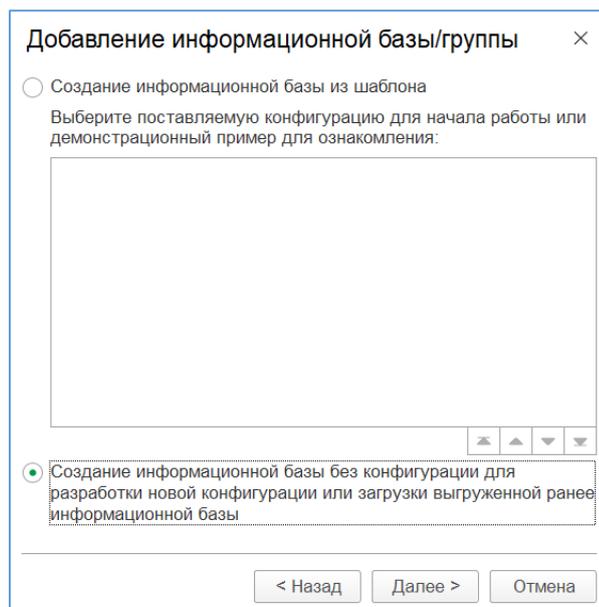


Рисунок 4.2 – Окно «Добавление информационной базы/группы»

3) Задать название для базы и выбрать «На сервере 1С:Предприятия»;

4) Заполнить параметры:

- кластер серверов – имя или адрес сервера, на котором размещается сервер 1С:Предприятия. “localhost”, если настройка происходит на этом же сервере;
- тип СУБД – PostgreSQL;
- сервер баз данных – имя или адрес сервера (кластера), где установлена СУБД «Jatoba»;
- имя базы данных – имя создаваемой БД;
- пользователь и пароль базы данных – существующий пользователь в СУБД Jatoba, имеющий право на создание БД;
- оставить включенным пункт «Создать базу данных в случае ее отсутствия»;
- если планируется загружать в БД файл конфигурации готовой БД 1С в качестве копии, можно включить пункт «Установить блокировку регламентных заданий»;

**Добавление информационной базы/группы** ×

Укажите параметры информационной базы:

Кластер серверов 1С:Предприятия: localhost

Имя информационной базы в кластере: jatoba\_db

Защищенное соединение: Выключено ▾

Тип СУБД: PostgreSQL ▾

Сервер баз данных: jatoba\_server

Имя базы данных: jatoba\_db

Пользователь базы данных: postgres

Пароль пользователя: ●●●●●●

Смещение дат: 0 ▾

Создать базу данных в случае ее отсутствия

Язык (Страна): русский (Россия) ▾

Установить блокировку регламентных заданий

< Назад    Далее >    Отмена

Рисунок 4.3 – Окно «Добавление информационной базы/группы»

5) Нажать кнопку далее «Далее» и «Готово».

Если параметры заполнены верно, СУБД создаст базу, заполнит ее таблицами для 1С по умолчанию, и после недолгой загрузки откроется первое окно со списком информационных баз.

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

APT		Advanced Packaging Tool
MS SQL		Microsoft SQL Server
SQL	–	Structured Query Language
БД	–	База данных
ОС	–	Операционная система
СУБД	–	Система управления базами данных

