

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ  
БАЗАМИ ДАННЫХ «ЈАТОВА»

Руководство по настройке. Часть 24.  
Реорганизация таблиц.  
Компонент «pg\_repack»

643.72410666.00067-07 98 01-24

Листов 22

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

## АННОТАЦИЯ

В документе приведены сведения, необходимые для установки и эксплуатации компонента «pg\_rerack» (далее по тексту – «компонент» или pg\_rerack), предназначенного для реорганизации таблиц СУБД «Jatoba».

Настоящее руководство предназначено для администраторов СУБД.



Все примеры в данном документе приведены для СУБД «Jatoba» версии ядра 6.x, для других версий все шаги выполняются аналогично, разница состоит в именах директорий.

Например, СУБД «Jatoba» версии 6.x по умолчанию устанавливается в директорию:

- ОС Windows – «C:\Program Files\GIS\Jatoba\6\bin»;
- ОС Linux – «/usr/jatoba-6/bin».

Версия компонента — 1.5.1

Степени важности примечаний, применяемые в документе:



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



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



**Важная информация**

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

## СОДЕРЖАНИЕ

1. Назначение компонента.....	4
1.1. Условия применения.....	4
2. Установка и настройка.....	6
2.1. Установка пакета «pg_gesack» в ОС GNU/Linux.....	6
2.2. Установка расширения.....	7
3. Функциональные возможности компонента.....	9
3.1. Параметры реорганизации.....	10
3.2. Параметры подключения утилиты .....	11
4. Пример использования .....	13
4.1. Кластеризация online.....	13
4.2. Полная вакуумизация online конкретных таблиц .....	14
4.3. Перемещение отдельного индекса.....	16
4.4. Перемещение всех индексов конкретной таблицы .....	18
5. Удаление расширения.....	20
Перечень сокращений.....	21

## 1. НАЗНАЧЕНИЕ КОМПОНЕНТА

Компонент «pg\_gerack» — представляет собой утилиту и расширение, которое предотвращает раздувание таблиц и индексов. В отличие от команд CLUSTER и VACUUM FULL, компонент «pg\_gerack» осуществляет операции без полной блокировки таблиц. Также, это расширение позволяет восстановить физический порядок кластеризованных индексов.

Такая функциональная возможность снижает объем занимаемого дискового пространства и повышает производительность сервера СУБД.

### 1.1. Условия применения

Компонент «pg\_gerack» может использоваться с СУБД «Jatoba» версий 5.x и выше, под управлением операционных систем GNU/Linux. Непосредственно в составе СУБД «Jatoba» компонент используется для поддержки мониторинга СУБД в части анализа запросов, как описано в документах

- Руководство по настройке. Часть 24. Поддержка мониторинга СУБД в части анализа запросов;
- Руководство по настройке. Часть 7. Пользовательский веб-интерфейс для администраторов. Компонент «Jatoba data safe». Подраздел «Анализ запросов» (Query analysis).

Компонент выполнен в форме внешней утилиты и расширения СУБД и не имеет ограничений по совместимости с другими компонентами.

Имеет ряд особенностей по функциональным возможностям:

- Отсутствие ограничений на работу с жестким диском и возможности распределить нагрузку приводит к значительной нагрузке на жесткий диск и падению производительности при работе с большими таблицами (падение сравнимо с вызовом полного копирования таблицы);
- Во время работы переполняется журнал предзаписи (WAL, Write-Ahead Logging), что приводит к задержке между лидером кластера и репликой. Результатом этого может стать падение СУБД;



При обработке больших таблиц рекомендуется отключить слот репликации на время работы утилиты

— Для реорганизации таблиц требуется дополнительное место на жестком диске, равное сумме размера оригинальной таблицы, размера индексов и размера таблицы с журналом аудита. При небольшой нагрузке размером журнала аудита можно пренебречь.

— Не может реорганизовывать временные таблицы;

— Не может кластеризовать таблицу согласно индексам GiST и RUM;

— Во время работы утилиты, нельзя выполнять DDL команды: CREATE, ALTER и DROP.

## 2. УСТАНОВКА И НАСТРОЙКА

Установка компонента должна производиться от имени пользователя, обладающего административными привилегиями в системе.

Установка компонента под управлением ОС Windows и ОС GNU/Linux приведено ниже.

### 2.1. Установка пакета «pg\_repack» в ОС GNU/Linux

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

Установку пакета 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 jatoba<ver>-pg-repack
```

— 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 jatoba<ver>-pg_repack
```

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 jatoba<ver>-pg_repack
```

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

```
zypper install jatoba<ver>-pg_repack
```

Установка компонента в составе других версий СУБД «Jatoba» осуществляется аналогично. Отличие будет только в номере версии СУБД, в составе которой он распространяется.

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.

В результате установки пакета в директории:

- `/usr/jatoba-<ver>/bin` будет создана утилита `*pg_repack`;
- `/usr/jatoba-6/share/extension/` будут созданы одноименные файлы, а именно:
  - управляющий файл расширения, с суффиксом. `control`;
  - SQL-скрипт для расширения.

## 2.2. Установка расширения

После перезагрузки СУБД и загрузки расширения станет доступной установка расширения «`pg_repack`» SQL-командой:

```
CREATE EXTENSION pg_repack;
```

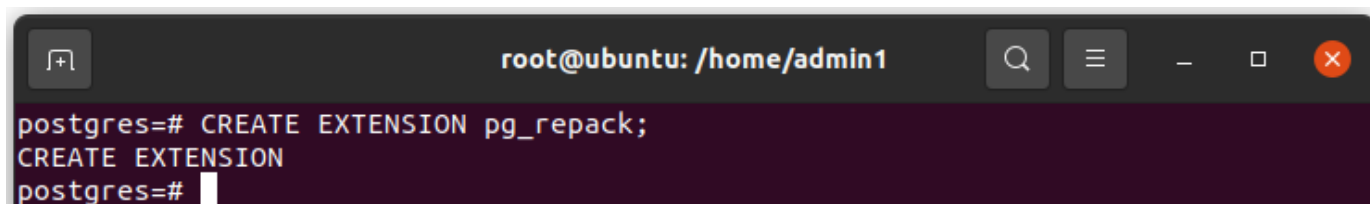


Рисунок 2.1 – Установка расширения

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

Установка расширения выполняется от имени и с правами привилегированного пользователя.



### 3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ КОМПОНЕНТА

Можно выбрать один из четырех способов использования:

- неблокирующая кластеризация (CLUSTER) (с упорядочиванием по кластеризующему индексу);
- упорядочивание по указанным столбцам;
- неблокирующая полная очистка (VACUUM FULL) (только упаковка строк);
- перестроение или перемещение только индексов таблицы.

Применить «pg\_gerack» можно только к таблицам, в которых должен быть первичный ключ (PRIMARY KEY) или уникальный не нулевой индекс (UNIQUE NOT NULL).

При этом алгоритм работы будет следующим:

- Расширение pg\_gerack создаёт лог-таблицу, чтобы хранить данные обо всех изменениях во время работы. Триггер будет реплицировать эти изменения на каждую операцию INSERT, UPDATE и DELETE.
- Создаётся таблица, аналогичная исходной по структуре, но без индексов и ограничений, чтобы не замедлять процесс вставки данных.
- Компонент «pg\_gerack» переносит в новую таблицу данные из старой, автоматически фильтруя все неактуальные строки, а затем создаёт индексы для новой таблицы. За время выполнения всех этих операций в лог-таблице накапливаются изменения.
- Переносятся изменения в новую таблицу. Перенос выполняется в несколько итераций, и когда в лог-таблице остаётся менее 20 записей, pg\_gerack захватывает строгую блокировку, переносит последние данные и подменяет старую таблицу на новую в системных таблицах Jatoba.
- Старая таблица и таблица с логами удаляются и в файловой системе освобождается место.

### 3.1. Параметры реорганизации

Параметры реорганизации таблиц приведены в таблице 3.1.

Таблица 3.1 – Параметры реорганизации таблиц

Параметры	Описание
-a --all	Попытаться перепаковать все базы данных в кластере. Базы данных, в которых расширение pg_gerack не установлено, будут пропущены.
-t таблица, --table=таблица	Реорганизовать только указанную таблицу (таблицы). Реорганизовать несколько таблиц можно, добавив несколько ключей -t. По умолчанию реорганизуются все подходящие таблицы в целевых базах данных.
-I таблица --parent-table=таблица	Реорганизовать как указанную таблицу (таблицы), так и дочерние. Реорганизовать несколько иерархий таблиц можно, добавив несколько ключей -I.
-с схема --schema=схема	Перепаковать таблицы только в указанной схеме (схемах). Перепаковать несколько схем можно, добавив несколько ключей -с. Можно использовать в сочетании с --tablespace для перемещения таблиц в другое табличное пространство.
-о столбец[, ...] --order-by=столбец[, ...]	Выполнить неблокирующую кластеризацию (CLUSTER), упорядочивая данные по указанным столбцам.
-n --no-order	Выполнить неблокирующую полную очистку (VACUUM FULL).
-N --dry-run	Только показать, какие таблицы будут перепакованы, и завершиться.
-j num_jobs --jobs=num_jobs	Установить заданное количество дополнительных соединений к ятоба и использовать эти дополнительные подключения для перестраивания индексов таблиц в параллельном режиме. Параллельное перестроение индексов поддерживается только при реорганизации таблиц полностью, без ключей --index или --only-indexes. Если ваш сервер имеет несколько процессорных ядер и быструю дисковую подсистему, параллельный режим может быть полезен для ускорения pg_gerack.
-s тбл_пр-во --tablespace= тбл_пр-во	Перенести перепакованные таблицы в заданное табличное пространство: по сути это неблокирующая версия команды ALTER TABLE ... SET TABLESPACE. Индексы таблиц остаются в исходном табличном пространстве, если только дополнительно не указан ключ --moveidx.
-S --moveidx	Также перенести индексы перепакованных таблиц в табличное пространство, заданное ключом --tablespace.
-i индекс --index=индекс	Перепаковать только указанный индекс. Перепаковать несколько индексов можно, добавив несколько ключей -i. Можно использовать в сочетании с --tablespace для перемещения индексов в другое табличное пространство.
-x --only-indexes	Перепаковать только индексы таблиц, заданных ключами --table.
-Т сек --wait-timeout=сек	Расширению pg_gerack необходимо получить исключительную блокировку в конце реорганизации. Этот параметр определяет,

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

Параметры	Описание
	сколько секунд pg_repack будет ждать получения этой блокировки. Если за это время блокировка не будет получена и параметр --no-kill-backend не указан, pg_repack принудительно отменит конфликтующие запросы. Если же вы используете СУБД «Jatoba» с версией ядра 5 и новее, pg_repack прибегнет к вызову pg_terminate_backend(), чтобы отключить все оставшиеся фоновые процессы, после того, как это время истечёт дважды. Значение по умолчанию: 60 сек.
-D --no-kill-backend	Пропускать перепакровку таблицы вместо отмены конфликтующих запросов, если блокировка не может быть получена в течение времени, указанного в --wait-timeout. Значение по умолчанию — false.
-Z --no-analyze	Не выполнять ANALYZE после полной реорганизации таблицы. В отсутствие этого ключа после реорганизации ANALYZE выполняется.
-k --no-superuser-check	Пропускать проверку суперпользователя клиентом. Этот параметр полезен при использовании pg_repack на платформах, которые поддерживают запуск расширения не от имени суперпользователя.
-C --exclude-extension	Пропускать таблицы, которые принадлежат указанному расширению (расширениям). Время планирования некоторых расширений может сильно зависеть от таких таблиц
--switch-threshold	Переключать таблицы, когда в таблице журнала остаётся заданное число кортежей. Этот параметр можно использовать, чтобы иметь возможность нагнать таблицы с большим объёмом записи.

### 3.2. Параметры подключения утилиты

Для подключения утилиты к СУБД можно использовать переменные окружения: PGDATABASE, PGHOST, PGPORT и PGUSER

Параметры подключения утилиты к СУБД приведены в таблице 3.2.

Таблица 3.2 - Параметры подключения утилиты к СУБД

Параметры	Описание
-a --all	Реорганизовать все базы данных. Параметр --all нельзя использовать вместе с --dbname, --table или --parent-table.
-d имя_бд --dbname=имя_бд	Указывает имя базы данных для реорганизации. Если оно не указано, и параметр -a (или --all) не используется, то имя базы берётся из переменного окружения PGDATABASE. Если и эта переменная не задана, выбирается имя подключающегося пользователя.
-h сервер --host=сервер	Указывает имя компьютера, на котором работает сервер. Если значение начинается с косой черты, оно определяет каталог Unix-сокета.
-p порт --port=порт	Порт подключения
-U имя_пользователя	Имя пользователя для подключения.

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

Параметры	Описание
--username=имя_пользователя	
-w --no-password	Не выдавать запрос на ввод пароля. Если сервер требует аутентификацию по паролю и пароль не доступен с помощью других средств, таких как файл .rgrpass, попытка соединения не удастся. Этот параметр может быть полезен в пакетных заданиях и скриптах, где нет пользователя, который вводит пароль.
-W --password	Принудительно запрашивать пароль перед подключением к базе данных. Это несущественный параметр, так как pg_gerack запрашивает пароль автоматически, если сервер проверяет подлинность по паролю. Однако, чтобы понять это, pg_gerack лишний раз подключается к серверу. Поэтому иногда имеет смысл ввести -W, чтобы исключить эту ненужную попытку подключения.

## 4. ПРИМЕР ИСПОЛЬЗОВАНИЯ

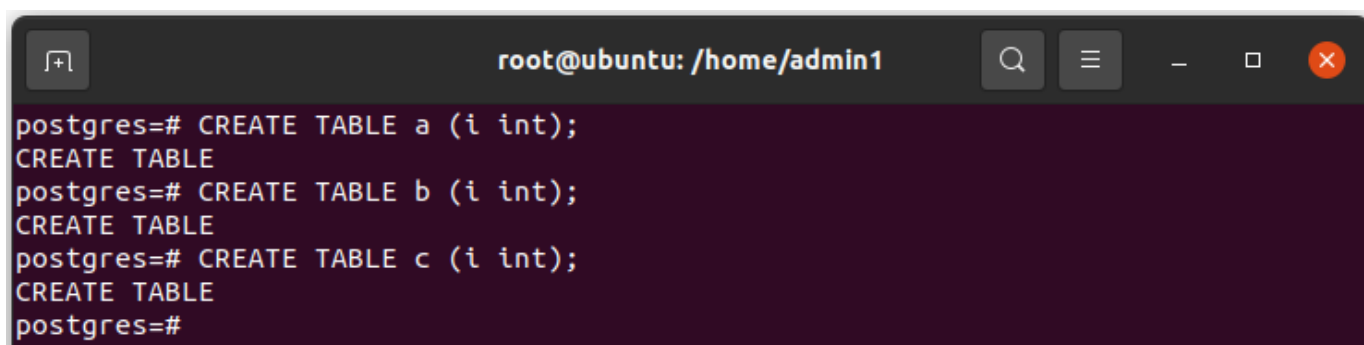
### 4.1. Кластеризация online

Компонент обладает функциональной возможностью выполнения неблокирующей кластеризации (CLUSTER) всех кластеризованных таблиц и полной очистки (VACUUM FULL) всех некластеризованных таблиц.

#### Например

Создать таблицы:

```
CREATE TABLE a (i int);  
CREATE TABLE b (i int);  
CREATE TABLE c (i int);
```

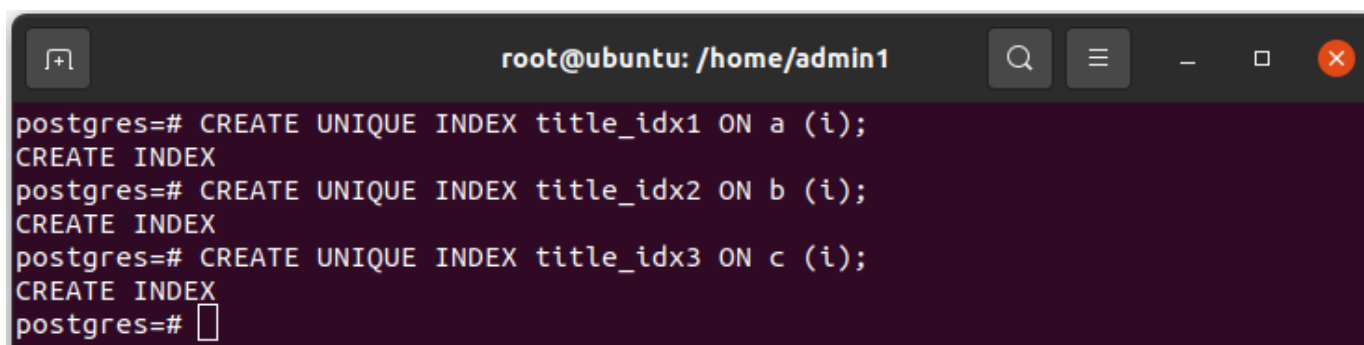


```
root@ubuntu: /home/admin1  
postgres=# CREATE TABLE a (i int);  
CREATE TABLE  
postgres=# CREATE TABLE b (i int);  
CREATE TABLE  
postgres=# CREATE TABLE c (i int);  
CREATE TABLE  
postgres=#
```

Рисунок 4.1 – SQL-команда создания таблиц

Создать индексы:

```
CREATE UNIQUE INDEX title_idx1 ON a (i);  
CREATE UNIQUE INDEX title_idx2 ON b (i);  
CREATE UNIQUE INDEX title_idx3 ON c (i);
```



```
root@ubuntu: /home/admin1  
postgres=# CREATE UNIQUE INDEX title_idx1 ON a (i);  
CREATE INDEX  
postgres=# CREATE UNIQUE INDEX title_idx2 ON b (i);  
CREATE INDEX  
postgres=# CREATE UNIQUE INDEX title_idx3 ON c (i);  
CREATE INDEX  
postgres=#
```

Рисунок 4.2 - SQL-команда создания индексов

Кластеризовать первую таблицу:

```
CLUSTER a using title_idx1;
```

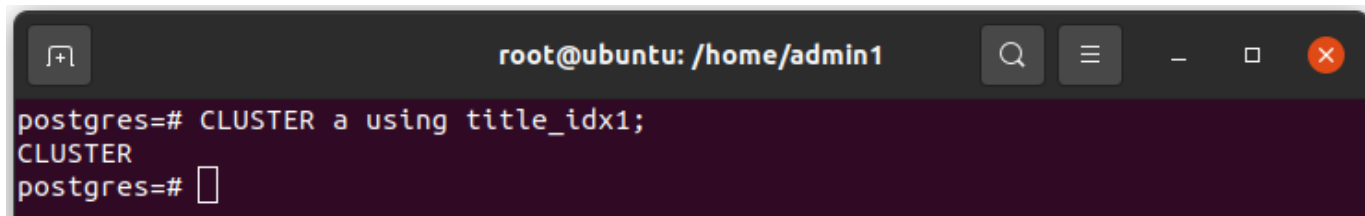


Рисунок 4.3 - SQL-команда создания индекса таблицы

Выполнить кластеризацию online таблицы a и полный вакуум online таблиц b и c утилитой «pg\_repack» в терминале ОС:

```
./pg_repack postgres
```

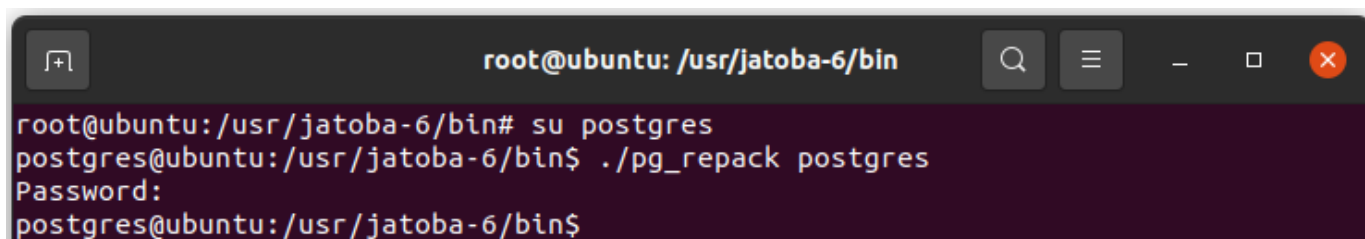


Рисунок 4.4 – Команда кластеризации таблицы

В выводе команды не должно быть сообщений об ошибках.

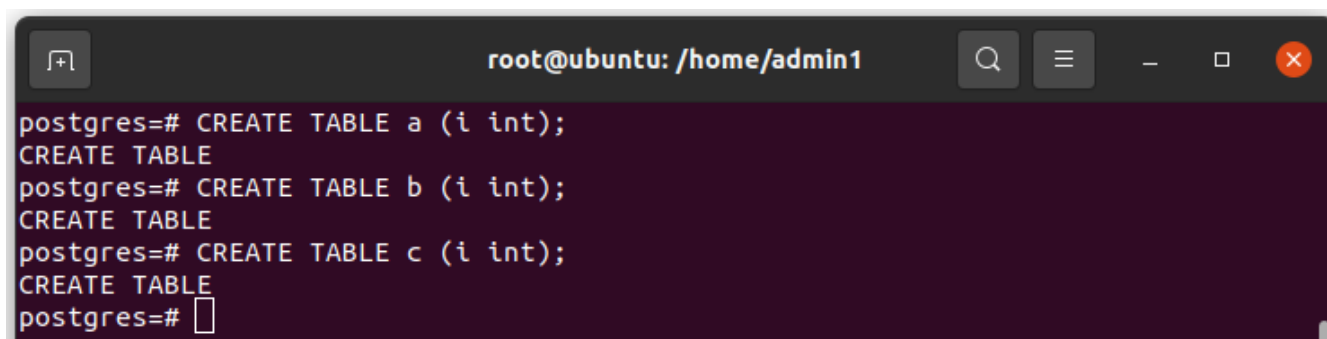
#### 4.2. Полная вакуумизация online конкретных таблиц

Компонент обладает функциональной возможностью полной вакуумизации «online» конкретных таблиц

##### Например

Создать таблицы:

```
CREATE TABLE a (i int);  
CREATE TABLE b (i int);  
CREATE TABLE c (i int);
```

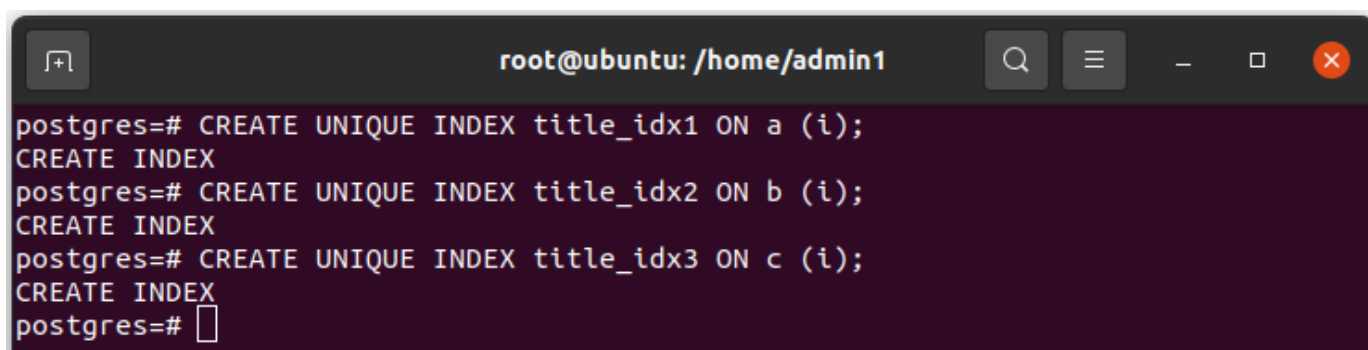


```
root@ubuntu: /home/admin1
postgres=# CREATE TABLE a (i int);
CREATE TABLE
postgres=# CREATE TABLE b (i int);
CREATE TABLE
postgres=# CREATE TABLE c (i int);
CREATE TABLE
postgres=#
```

Рисунок 4.5– SQL-команда создания таблиц

Создать индексы:

```
CREATE UNIQUE INDEX title_idx1 ON a (i);
CREATE UNIQUE INDEX title_idx2 ON b (i);
CREATE UNIQUE INDEX title_idx3 ON c (i);
```

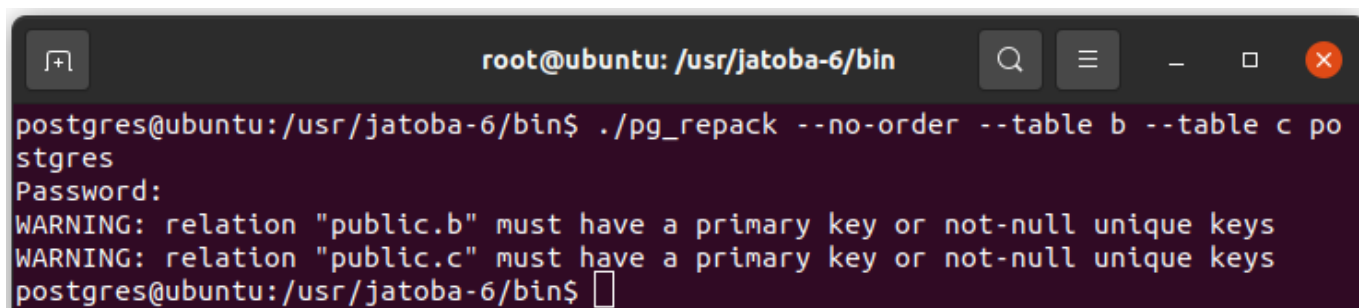


```
root@ubuntu: /home/admin1
postgres=# CREATE UNIQUE INDEX title_idx1 ON a (i);
CREATE INDEX
postgres=# CREATE UNIQUE INDEX title_idx2 ON b (i);
CREATE INDEX
postgres=# CREATE UNIQUE INDEX title_idx3 ON c (i);
CREATE INDEX
postgres=#
```

Рисунок 4.6 - SQL-команда создания индексов

Выполнить полный вакуум online таблиц b и c утилитой «pg\_repack» в терминале ОС:

```
./pg_repack --no-order --table b --table c postgres
```



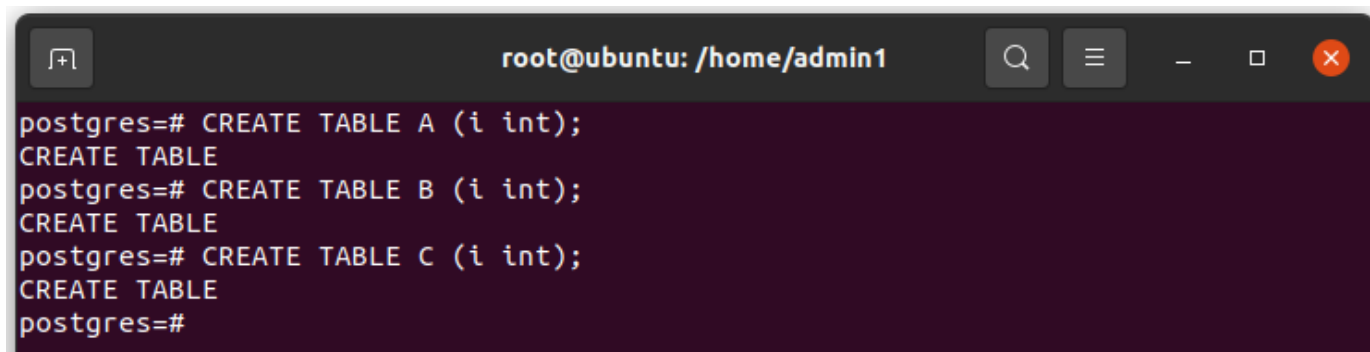
```
root@ubuntu: /usr/jatoba-6/bin
postgres@ubuntu:/usr/jatoba-6/bin$ ./pg_repack --no-order --table b --table c po
stgres
Password:
WARNING: relation "public.b" must have a primary key or not-null unique keys
WARNING: relation "public.c" must have a primary key or not-null unique keys
postgres@ubuntu:/usr/jatoba-6/bin$
```

Рисунок 4.7 – Команда кластеризации таблиц

### 4.3. Перемещение отдельного индекса

Создать таблицы:

```
CREATE TABLE A (i int);  
CREATE TABLE B (i int);  
CREATE TABLE C (i int);
```



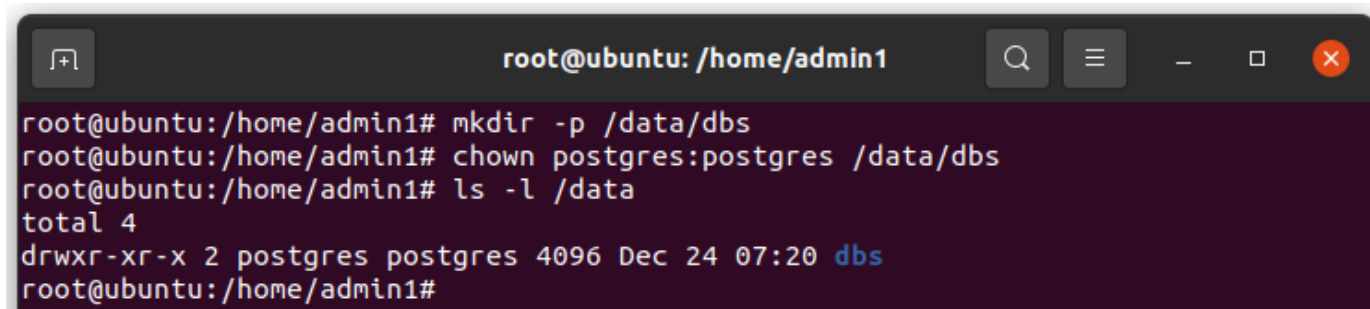
The screenshot shows a terminal window with the title bar 'root@ubuntu: /home/admin1'. The terminal content is as follows:

```
postgres=# CREATE TABLE A (i int);  
CREATE TABLE  
postgres=# CREATE TABLE B (i int);  
CREATE TABLE  
postgres=# CREATE TABLE C (i int);  
CREATE TABLE  
postgres=#
```

Рисунок 4.8 – SQL-команда создания таблиц

Создать каталог и назначить права в терминале ОС:

```
# mkdir -p /data/dbs  
# chown postgres:postgres /data/dbs
```



The screenshot shows a terminal window with the title bar 'root@ubuntu: /home/admin1'. The terminal content is as follows:

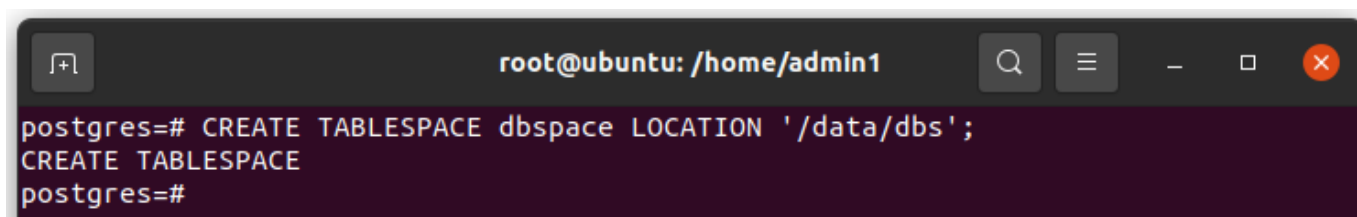
```
root@ubuntu:/home/admin1# mkdir -p /data/dbs  
root@ubuntu:/home/admin1# chown postgres:postgres /data/dbs  
root@ubuntu:/home/admin1# ls -l /data  
total 4  
drwxr-xr-x 2 postgres postgres 4096 Dec 24 07:20 dbs  
root@ubuntu:/home/admin1#
```

Рисунок 4.9 – Создание каталогов для хранения индекса и табличного пространства

Создать табличное пространство в отдельном каталоге:

```
CREATE TABLESPACE dbspace LOCATION '/data/dbs';
```



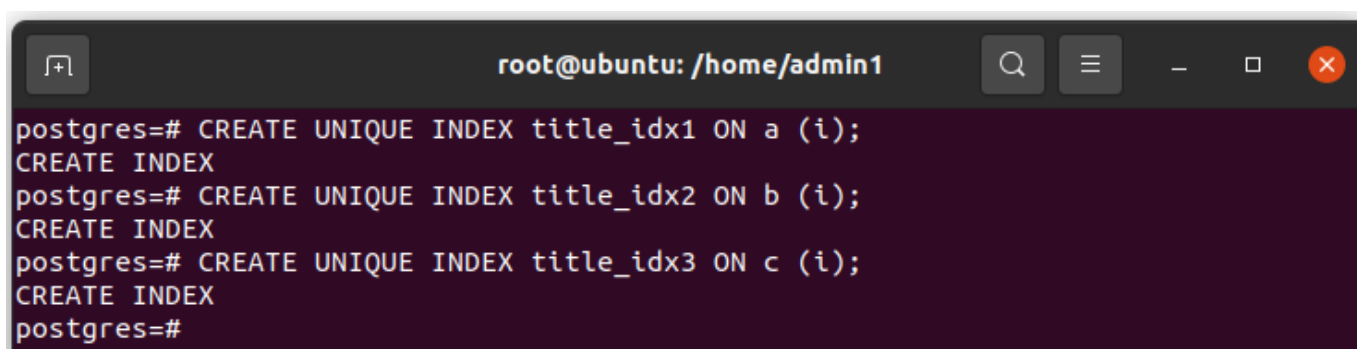


```
root@ubuntu: /home/admin1
postgres=# CREATE TABLESPACE dbspace LOCATION '/data/dbs';
CREATE TABLESPACE
postgres=#
```

Рисунок 4.10 - SQL-команда создания табличного пространства

Создать индексы:

```
CREATE UNIQUE INDEX title_idx1 ON a (i);
CREATE UNIQUE INDEX title_idx2 ON b (i);
CREATE UNIQUE INDEX title_idx3 ON c (i);
```

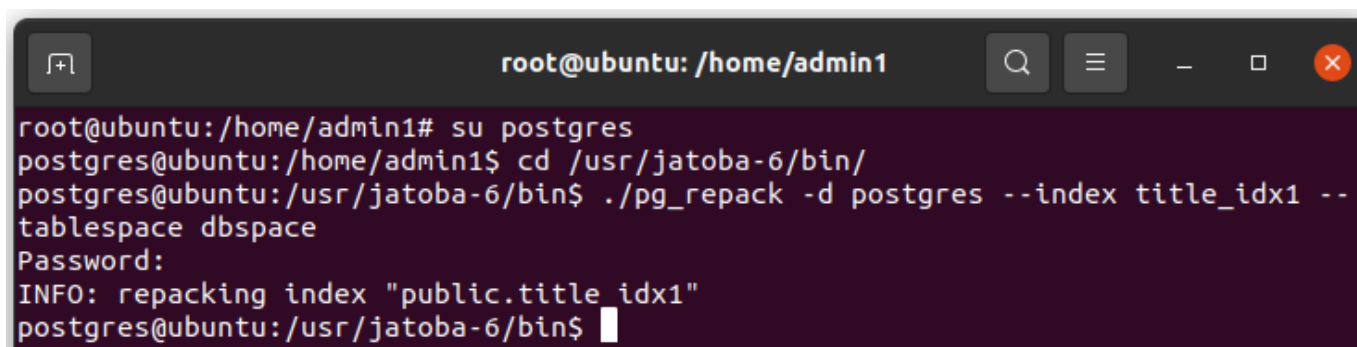


```
root@ubuntu: /home/admin1
postgres=# CREATE UNIQUE INDEX title_idx1 ON a (i);
CREATE INDEX
postgres=# CREATE UNIQUE INDEX title_idx2 ON b (i);
CREATE INDEX
postgres=# CREATE UNIQUE INDEX title_idx3 ON c (i);
CREATE INDEX
postgres=#
```

Рисунок 4.11 - SQL-команда создания индексов

Переместить указанный индекс в табличное пространство утилитой «pg\_repack» в терминале ОС:

```
./pg_repack -d postgres --index title_idx1 --tablespace dbspace
```



```
root@ubuntu: /home/admin1# su postgres
postgres@ubuntu: /home/admin1$ cd /usr/jatoba-6/bin/
postgres@ubuntu: /usr/jatoba-6/bin$ ./pg_repack -d postgres --index title_idx1 --
tablespace dbspace
Password:
INFO: repacking index "public.title_idx1"
postgres@ubuntu: /usr/jatoba-6/bin$
```

Рисунок 4.12 – Команда перемещения индекса в табличное пространство

#### 4.4. Перемещение всех индексов конкретной таблицы

Компонент обладает функциональной возможностью перемещения всех индексов таблицы.

##### Например

Создать таблицы:

```
CREATE TABLE a (i int);  
CREATE TABLE b (i int);  
CREATE TABLE c (i int);
```

Создать каталог и назначить права:

```
# mkdir -p /data/dbs  
# chown postgres:postgres /data/dbs
```

Создать табличное пространство:

```
CREATE TABLESPACE dbspace LOCATION '/data/dbs';
```

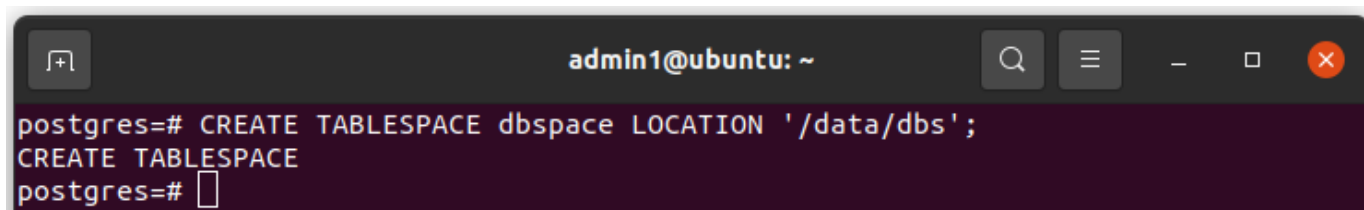
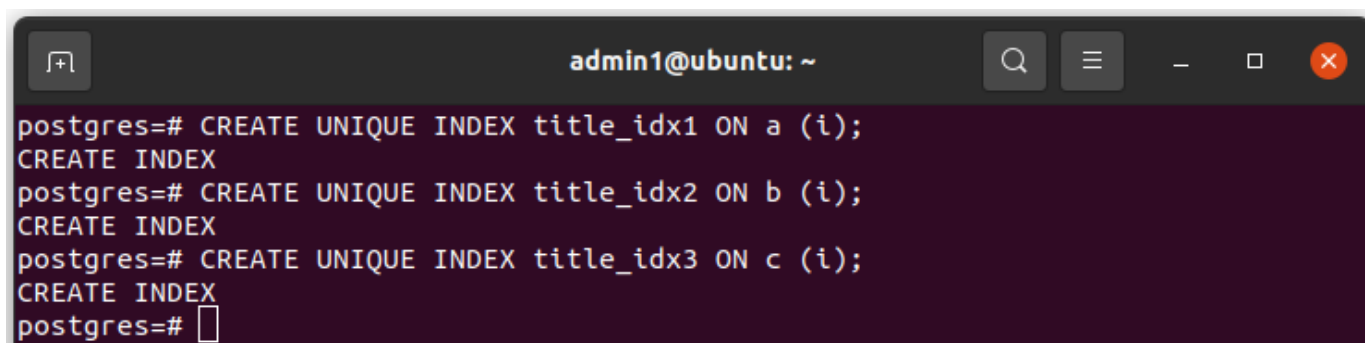


Рисунок 4.13 - SQL-команда создания табличного пространства

Создать индексы:

```
CREATE UNIQUE INDEX title_idx1 ON a (i);  
CREATE UNIQUE INDEX title_idx2 ON b (i);  
CREATE UNIQUE INDEX title_idx3 ON c (i);
```

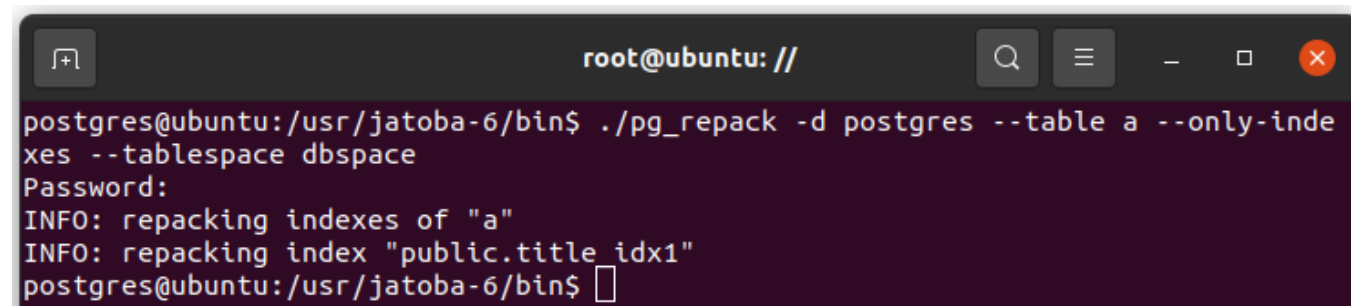


```
admin1@ubuntu: ~  
postgres=# CREATE UNIQUE INDEX title_idx1 ON a (i);  
CREATE INDEX  
postgres=# CREATE UNIQUE INDEX title_idx2 ON b (i);  
CREATE INDEX  
postgres=# CREATE UNIQUE INDEX title_idx3 ON c (i);  
CREATE INDEX  
postgres=#
```

Рисунок 4.14 - SQL-команда создания индексов

Переместить индексы таблицы a в табличное пространство утилитой «pg\_repack» в терминале ОС:

```
./pg_repack -d postgres --table a --only-indexes --tablespace  
dbspace
```



```
root@ubuntu: //  
postgres@ubuntu:/usr/jatoba-6/bin$ ./pg_repack -d postgres --table a --only-indexes --tablespace dbspace  
Password:  
INFO: repacking indexes of "a"  
INFO: repacking index "public.title_idx1"  
postgres@ubuntu:/usr/jatoba-6/bin$
```

Рисунок 4.15 – Команда перемещения индексов в табличное пространство

В выводе команды не должно быть сообщений об ошибках.

## 5. УДАЛЕНИЕ РАСШИРЕНИЯ

Расширение компонента удаляется SQL-командой:

```
DROP EXTENSION pg_repack CASCADE;
```

## **ПЕРЕЧЕНЬ СОКРАЩЕНИЙ**

SQL	–	Structured Query Language
БД	–	База данных
КС	–	Контрольные суммы
ОС	–	Операционная система
СУБД	–	Система управления базами данных

## Лист регистрации изменений

[illegible]

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