Университет ИТМО

Факультет ФПИ и КТ

P33131

**Отчет**

**по лабраторной работе №4**

# «Распределенные системы хранения данных» Вариант 20

Студент:

Ляо Ихун

Гр.P33131

Предподаватель:

Шешуков Дмитрий Михайлович

## Задание:

Лабораторная работа включает настройку резервного копирования данных с основного узла на резервный, а также несколько сценариев восстановления. Узел из предыдущей лабораторной работы используется в качестве основного; новый узел используется в качестве резервного. В сценариях восстановления необходимо использовать копию данных, полученную на первом этапе данной лабораторной работы.

Требования к отчёту

Отчет должен быть самостоятельным документом (без ссылок на внешние ресурсы), содержать всю последовательность команд, содержимое скриптов по каждому пункту задания. Для демонстрации результатов приводить команду вместе с выводом (самой наглядной частью вывода, при необходимости).

1. Резервное копирование

1.1 Настроить резервное копирование с основного узла на резервный следующим образом:

Периодические полные копии с помощью SQL Dump

По расписанию (cron) раз в сутки, методом SQL Dump с сжатием. Созданные архивы должны сразу перемещаться на резервных хост, они не должны храниться на основной системе. Срок хранения архивов на резервной системе - 4 недели. По истечении срока хранения, старые архивы должны автоматически уничтожаться.

1.2 Подсчитать, каков будет объем резервных копий спустя месяц работы системы,исходя из следующих условий:

Средний объем измененных данных за сутки: ~750 МБ.

1.3 Проанализировать результаты.

2. Потеря основного узла

Этот сценарий подразумевает полную недоступность основного узла. Необходимо восстановить работу СУБД на резервном узле, продемонстрировать успешный запуск СУБД и доступность данных.

3. Повреждение файлов БД

Этот сценарий подразумевает потерю данных (например, в результате сбоя диска или файловой системы) при сохранении доступности основного узла. Необходимо выполнить полное восстановление данных из резервной копии и перезапустить СУБД на основном узле.

Ход работы:

3.1 Симулировать сбой:

удалить с диска директорию любого табличного пространства со всем содержимым.

3.2 Проверить работу СУБД, доступность данных, перезапустить СУБД, проанализировать результаты.

3.3 Выполнить восстановление данных из резервной копии, учитывая следующее условие:

Исходное расположение дополнительных табличных пространств недоступно - разместить в другой директории и скорректировать конфигурацию.

3.4 Запустить СУБД, проверить работу и доступность данных, проанализировать результаты.

4. Логическое повреждение данных

Этот сценарий подразумевает частичную потерю данных (в результате нежелательной или ошибочной операции) при сохранении доступности основного узла. Необходимо выполнить восстановление данных на основном узле следующим способом:

Восстановление с использованием архивных WAL файлов.(СУБД должна работать в режиме архивирования WAL, потребуется задать параметры восстановления).

Ход работы:

4.1 В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.

4.2 Зафиксировать время и симулировать ошибку:

Перезаписать строки любой таблицы “мусором” (INSERT, UPDATE)

4.3 Продемонстрировать результат.

4.4 Выполнить восстановление данных указанным способом.

4.5 Продемонстрировать и проанализировать результат.

## Резервное копирование

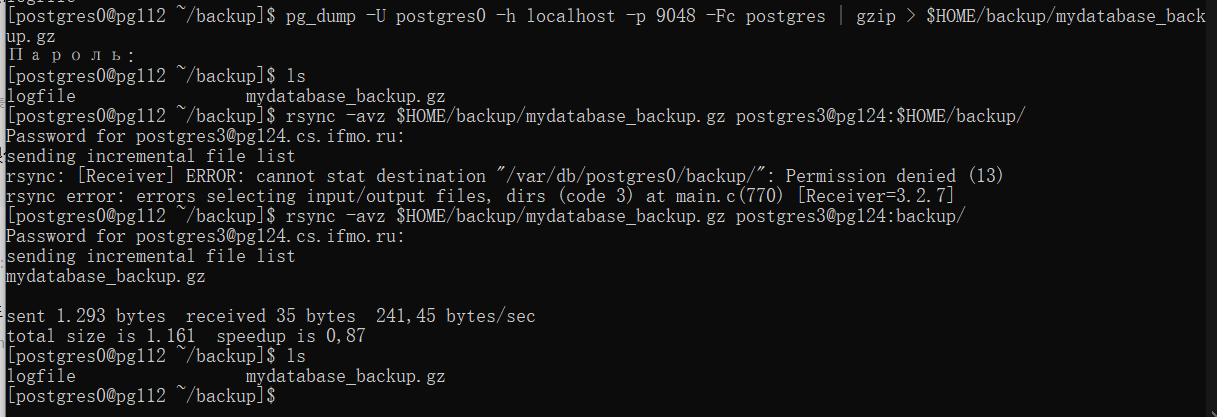
Средний объем измененных данных за сутки: ~750 МБ. Считаю это измнение файла sql.

объем резервных копий спустя месяц работы системы:

700МБ\*30 + исходной размер = 21ГБ + исходной размер

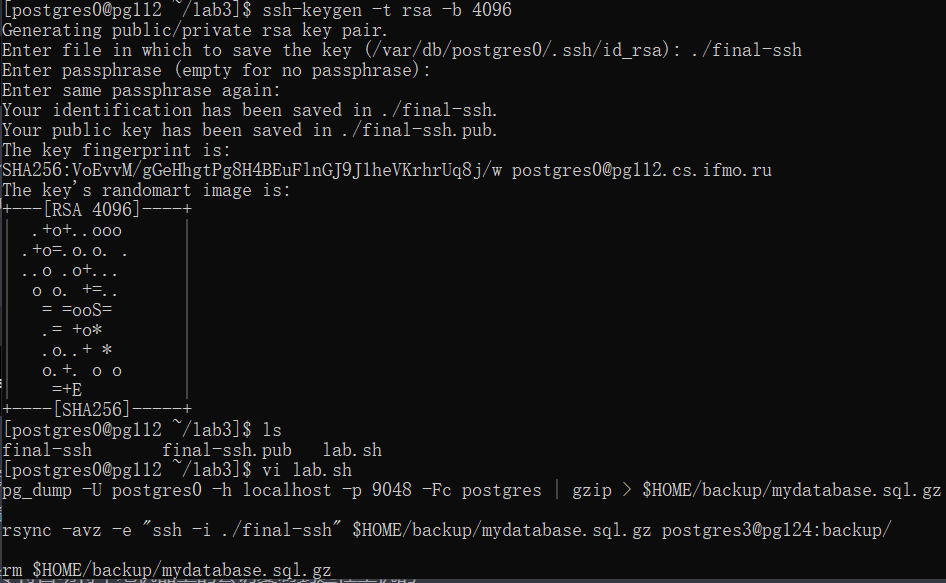
Здесь сохраняем его архив, поэтому объем меньше выше.

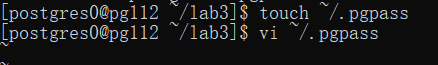
### Создание копии



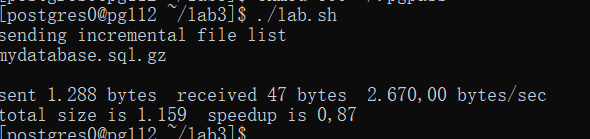


### Cоздание скрипта выполнения создания копии и её отправления





12

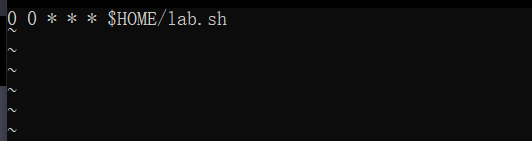


### Cоздание crontab

7

6

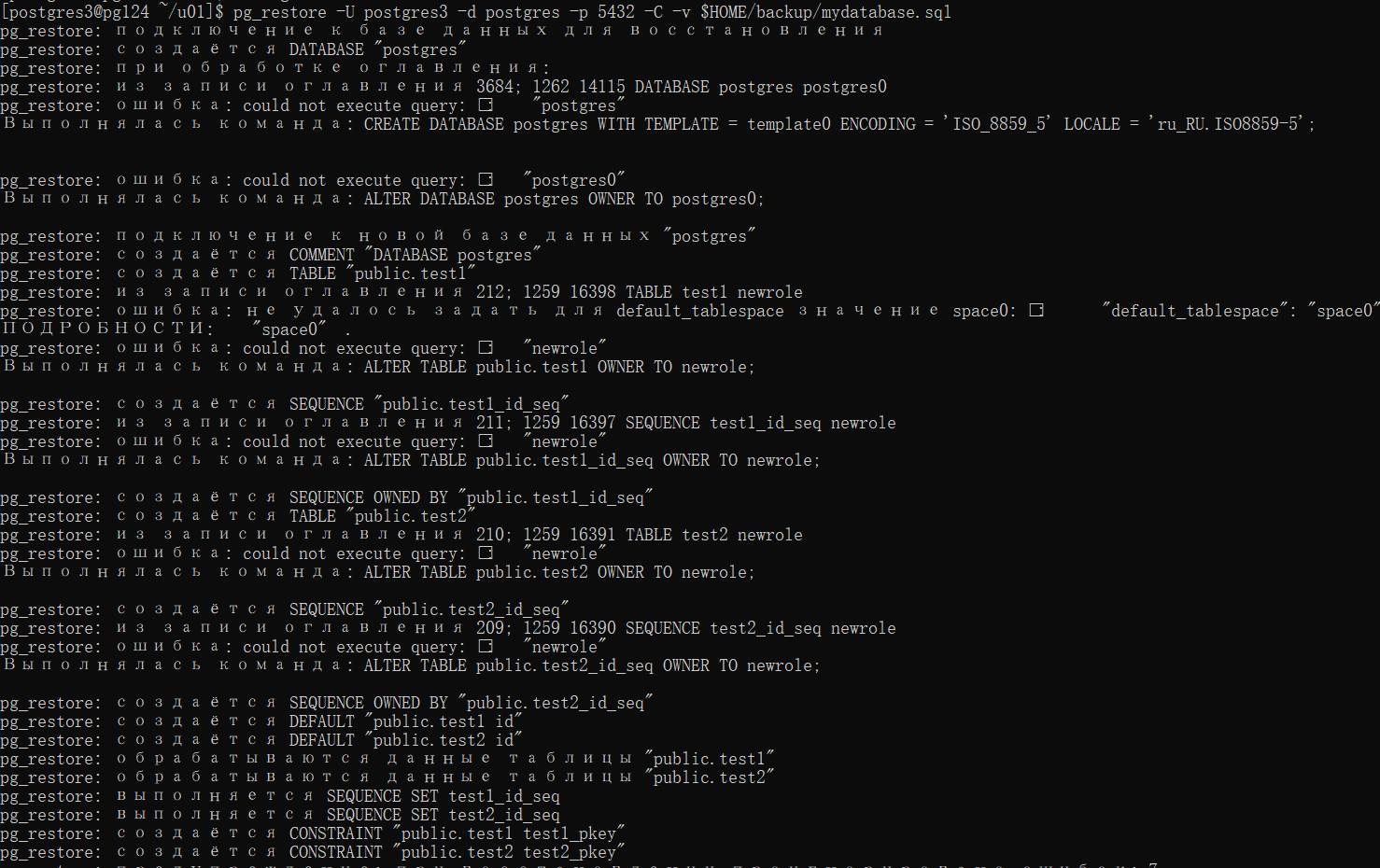
### Cоздание crontab и скрипта для удаления просрочной копии

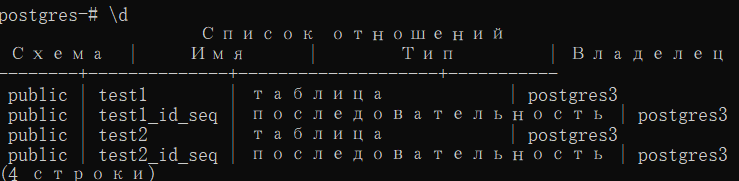
1315

## Потеря основного узла

### Имитация полной потери узлов

Мы можем делаем восстановление базы данных на резервном узле чтобы делать имитацию полной потери узлов.





2-5

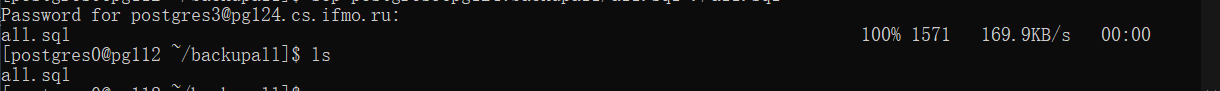
Видимо что там все информации о содержимом и структуре данных сохраняются. Но информации о tablespace и роли нет. Это из-за того что мы использовал pg\_dump. pg\_dumpall решает эту проблему и мы будем его использовать в следующих этапах.

## Повреждение файлов БД

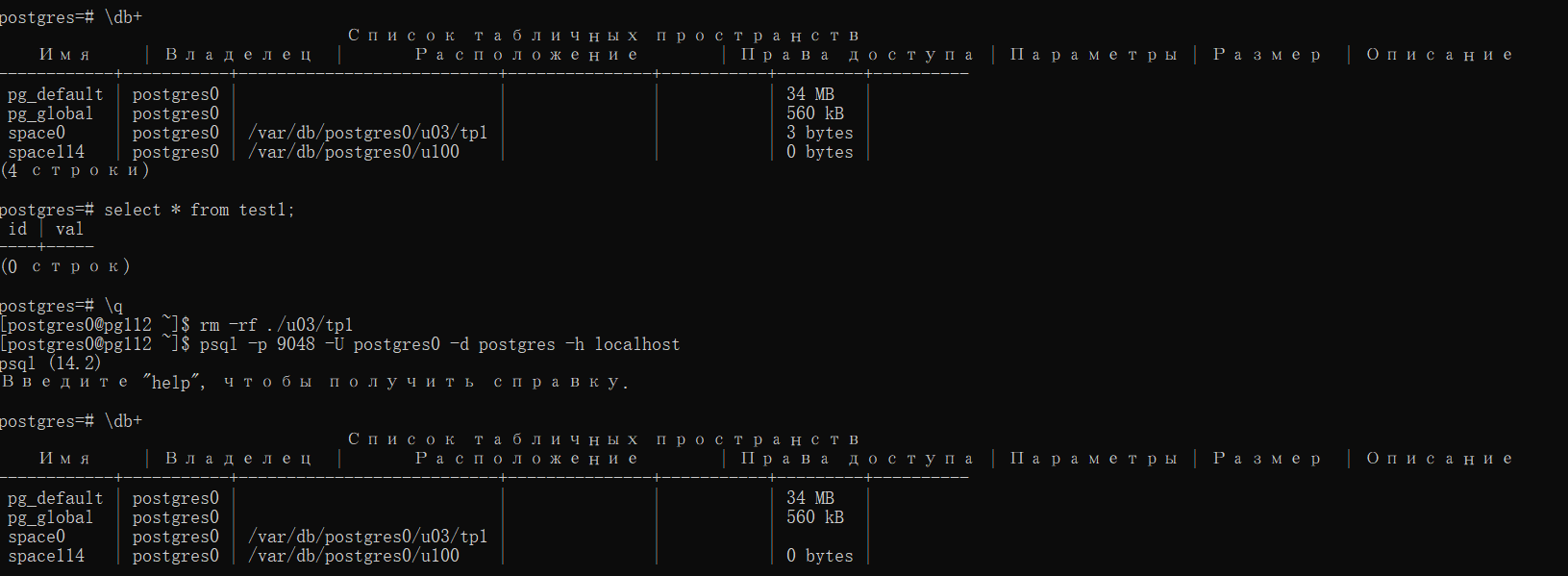
### Dump\_all

1. pg\_dumpall -h localhost -p 9048 -U postgres0 | gzip > $HOME/backupall/all.sql.gz
2. rsync -avz -e “ssh -i ./final-ssh” $HOME/backupAll/mydatabase\_backup.sql.gz postgres3@pg124:backupall/
3. Scp [postgres3@pg124:backupall/all.sql.gz](mailto:postgres3@pg124:backupall/all.sql.gz) $HOME/backall/all.sql.gz

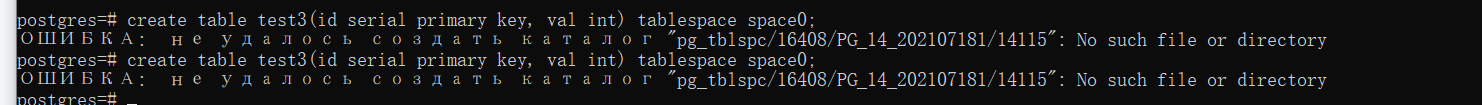
Добавляем те команды в lab.sh и получаем новую копию, которая имеет информацию о tablespace.

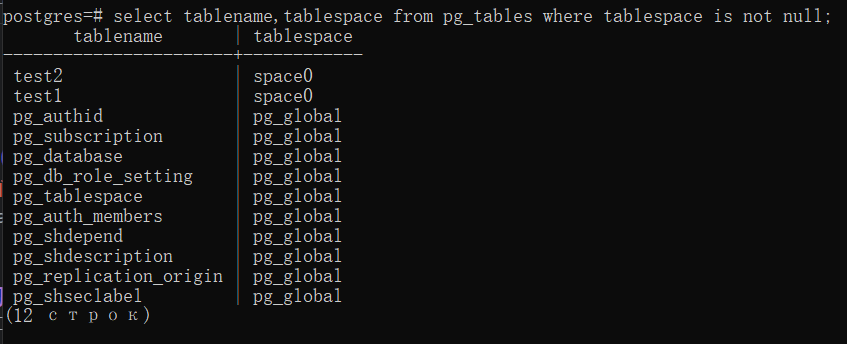


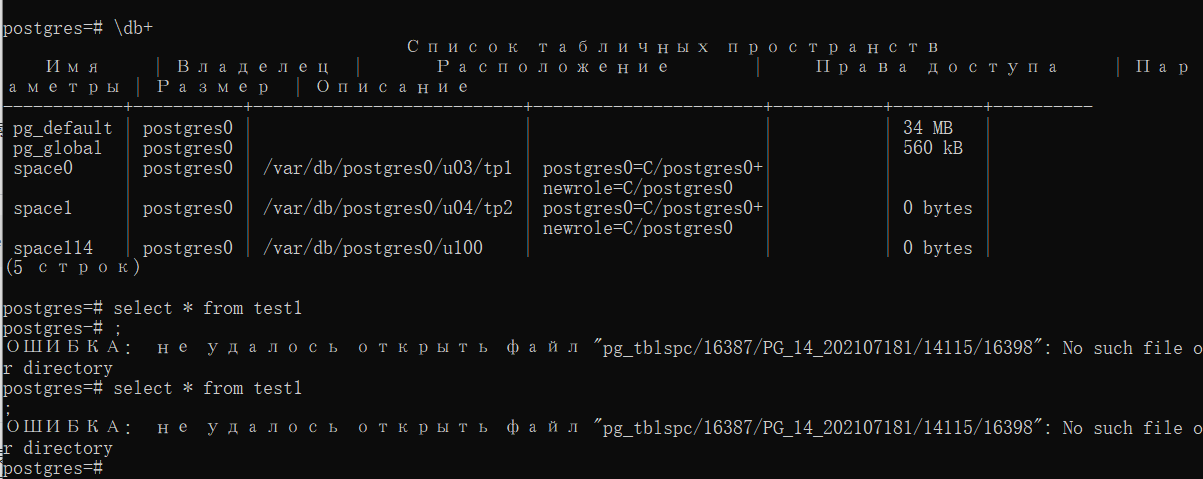
### Перед удалением



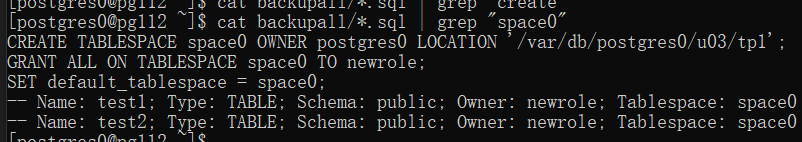
### После удаления

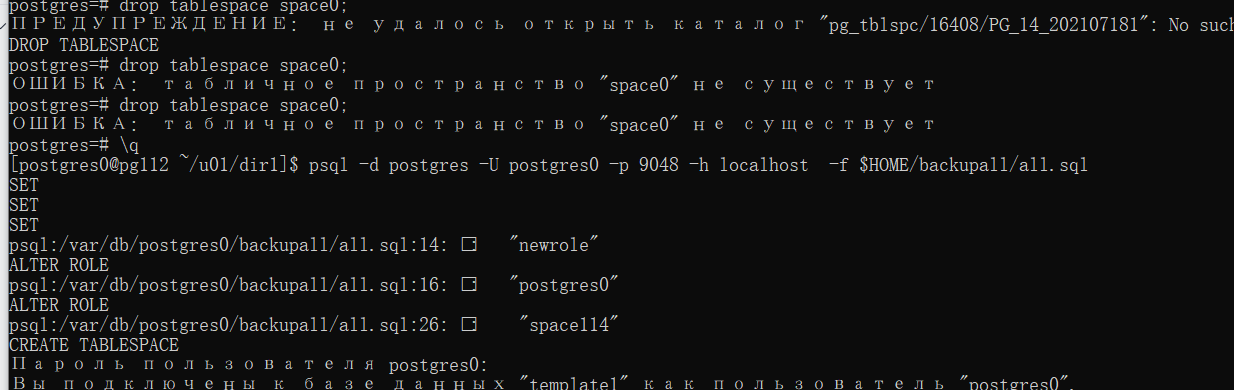


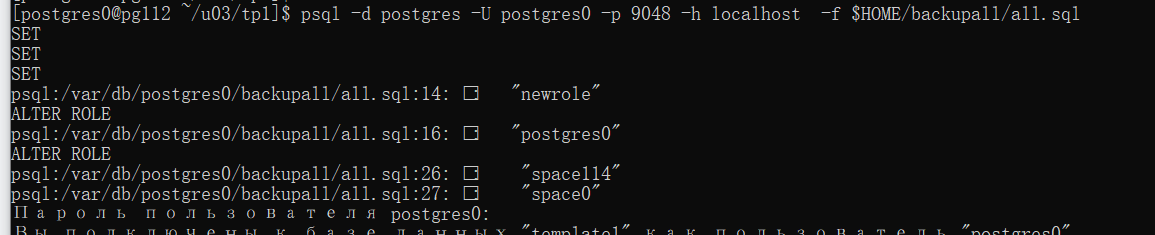


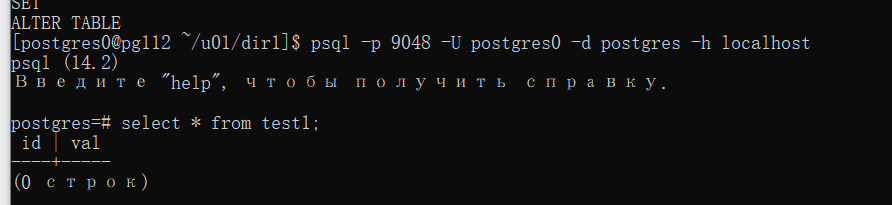


### Востановление





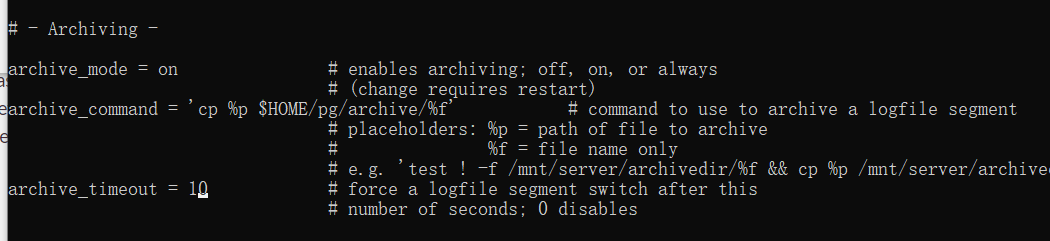


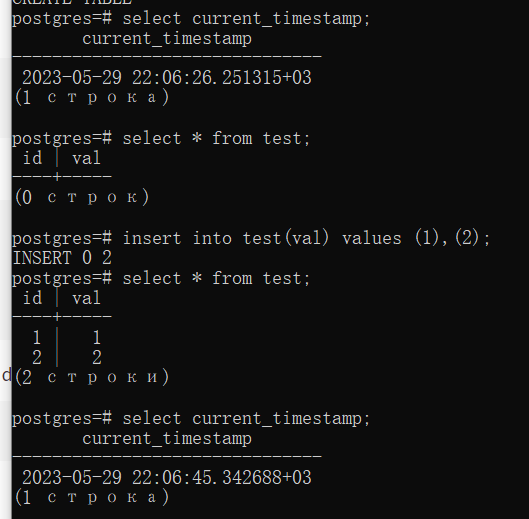


Важно что нужно предварительно создать все катологии, где сохраняются все пространства таблицы, и вручную удалять все таблицы и простраства в БД.

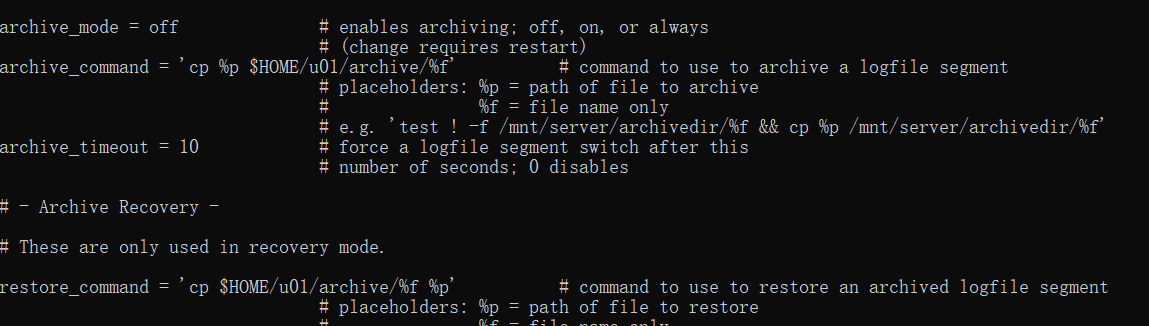
## Логическое повреждение данных

1





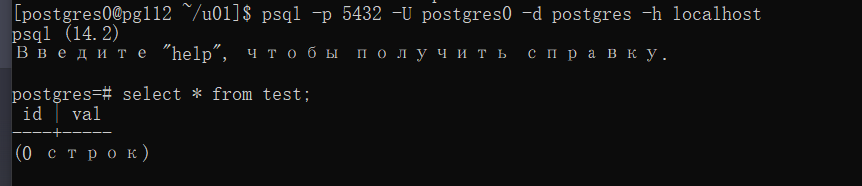
pg\_ctl -D $HOME/u01/dir1 -l $HOME/logfile stop



touch ./recovery.signal

rm -rf pg\_wal/\*

pg\_ctl -D $HOME/u01/dir1 -l $HOME/logfile start



## Вывод

В ходе выполнении работы изучал как создать резервное копирование при использовании sql dump. Успешно создал скрипт,который автомотически периодически выполняется каждый день, при помощи ssh. Упсешно восстановил бд при использовании wal файл.