Санкт-Петербургский национально исследовательский университет

информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники



**Лабораторная работа № 3 по дисциплине**

**«РСХД»**

Вариант 23.

Выполнил:

Студент группы: P33121

Нуруллаев Даниил

Преподаватель: Шешуков Д. М.

Санкт-Петерубрг 2023г.

**Текст задания**

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.

**1. Резервное копирование**

1.1 Настроить резервное копирование с основного узла на резервный следующим образом:  
Периодические холодные полные копии.  
Полная копия (rsync) по расписанию (cron) раз в сутки. СУБД на время копирования должна отключаться. На резервном узле хранить 14 копий, после успешного создания пятнадцатой копии, самую старую автоматически уничтожать.  
1.2 Подсчитать, каков будет объем резервных копий спустя месяц работы системы,исходя из следующих условий:  
Средний объем новых данных в БД за сутки: ~200 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 WAL 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 директории PGDATA 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.

**Выполнение**

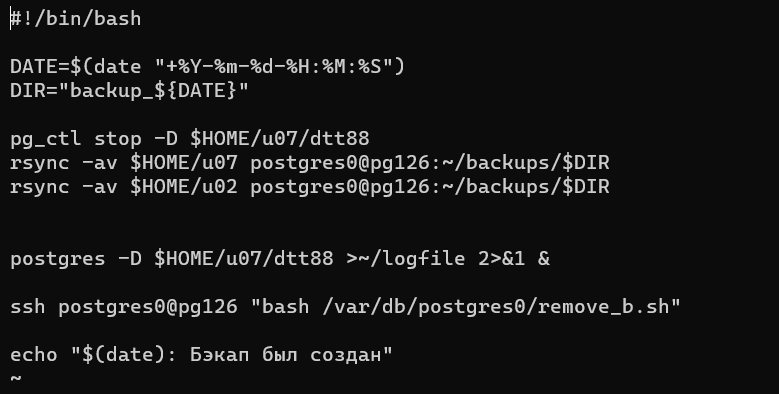
create role admin superuser createdb createrole login password '123'; создаем админа для тестов

**Задание 1. Резервное копирование**

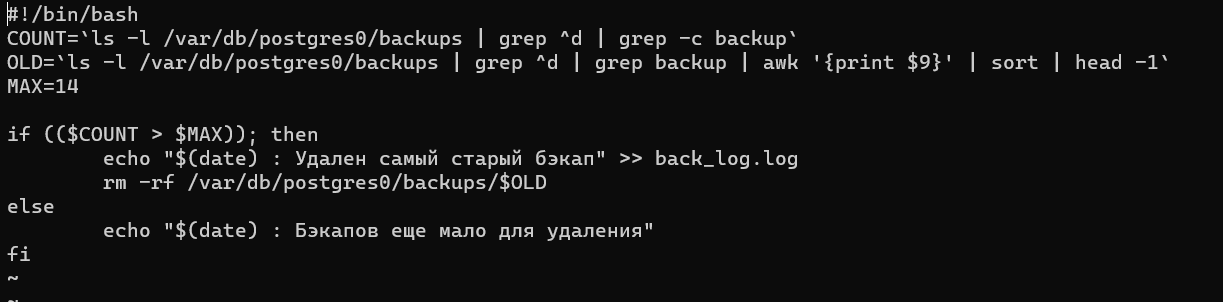
* 1. **Настройка резервного копирования**

Напишем bash-скрипт, который будет останавливать кластер, отправлять полную копию на резервный узел, после удаленно запускать скрипт (для удаления)

Скрипт для создания бэкапов на основном узле



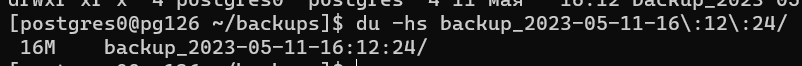
Скрипт для удаления бэкапов на побочном узле



Назначем cron выполнять резервное копирование каждый день в полночь



* 1. **Расчет объема**

****

Размер изначального бэкапа равен 16 МБ

Каждый день в бд будет добавляться 200 МБ новых данных

Предположим что мы начали делать бэкапы с 1 мая вплоть до 31 мая, значит нам надо высчитать сколько места займут бэкапы с пустя месяц

Посчитать это можно с помощью суммы арефметической прогрессии:

(2\*16+200\*30)\*31/2=93496 МБ = 91.3 ГБ

Так же создаются wal-файлы (16 МБ по дефолту), у нас происходит изменения на 200 мб следовательно получается 13 wal-файлов по 16 мб что в сумме дает 208Мб

Итого за месяц накапает еще 103168МБ = 100.75 ГБ

Итого мы получаем 192,05 ГБ

Из условия у нас хранятся только последние 14 бэкапов и того полуаем что у нас отнимается первые 17 бэкапов

Данные за первые 17 дней = 27472 МБ

Wal-файлы за первые 17 дней = 31824 МБ

27472 МБ + 31824 МБ = 57,9 ГБ

Итого за месяц мы потратим 192,05 ГБ – 57,9 ГБ = 134,15 ГБ

* 1. **Анализ результатов**

Полные холодные копии имеют большой достаточно размер и требуют остановки сервера (в отличии, например, от горячих инкрементальных)

**Задание 2. Потеря основного узла**

Копируем все данные из бэкапа и видим что табличное пространство не хочет переносится, так как мы не имеем право на нее

****

Создаем ссылку на табличное пространство 

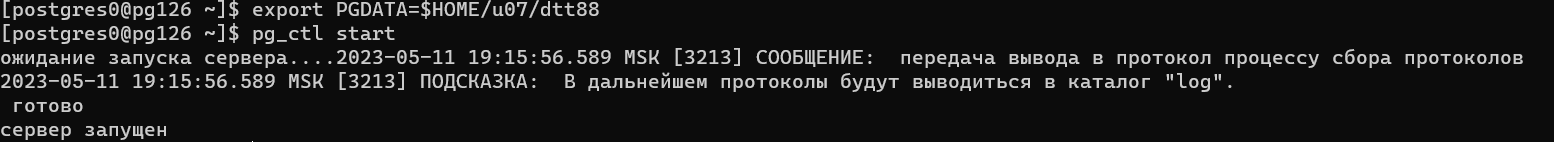
В postgresql.conf ставим путь для подгрузки wal-файлов

**restore\_command = 'cp /var/postgres/postgres0/u03/dcj13/%f %p'**

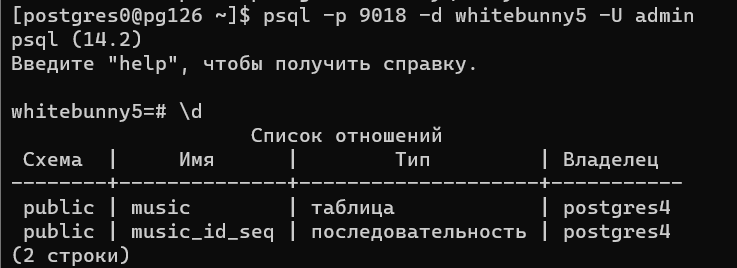
Создаем recovery.signal для сигнализации о том, что мы восстанваливаем систему



Запускаем сервер



Видим что все успешно работает



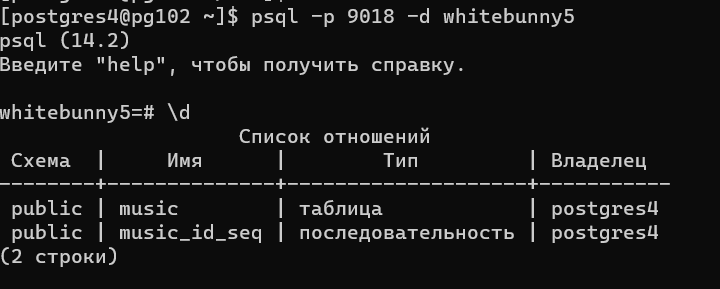
И даже таблицы на месте

**Задание 3. Повреждение файлов БД**

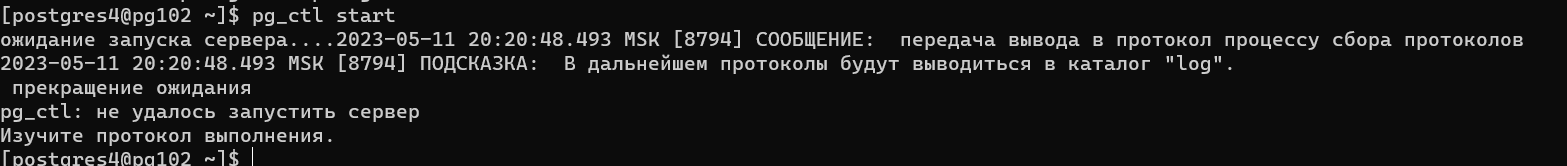
**3.1 Симуляция сбоя и проверка работы**

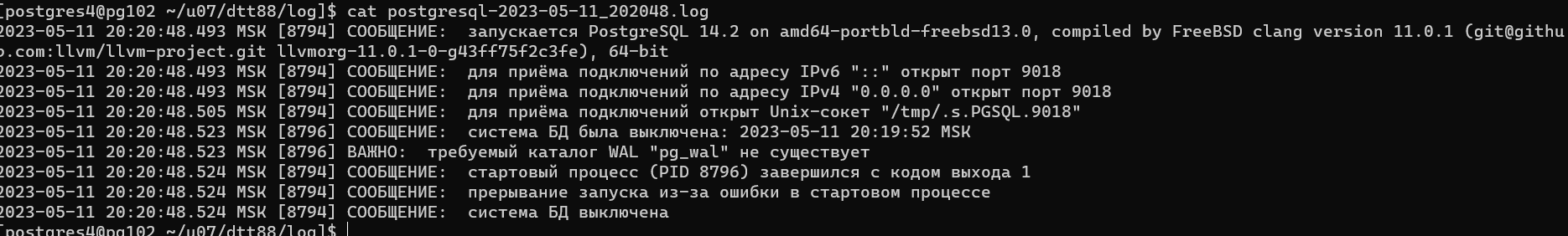
Удаляем директорию с WAL-файлами, проверим файлы БД:

rm -rf ~/u07/dtt88/pg\_wal

****

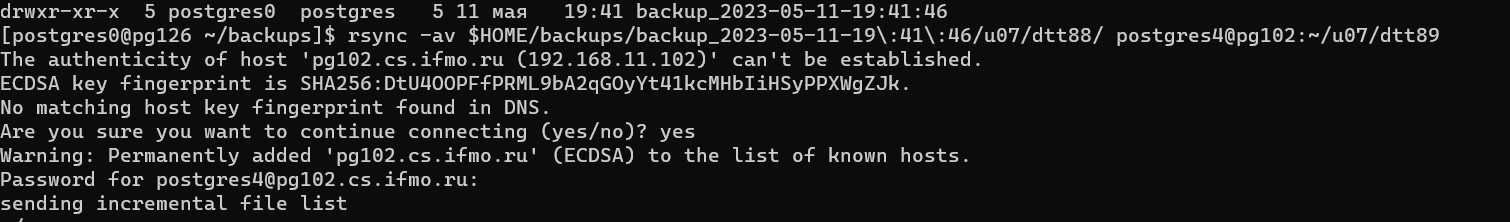
При перезапуске сервера видим данную картину





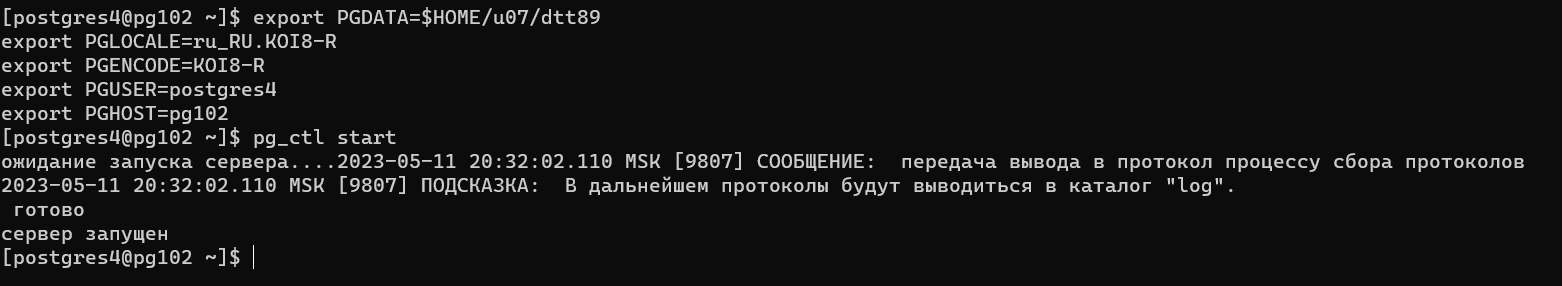
**3.2 Восстановление данных**

Так как кластер не может запуститься без директории pg\_wal, давайте выполним восстановление с последней резервной копии

****

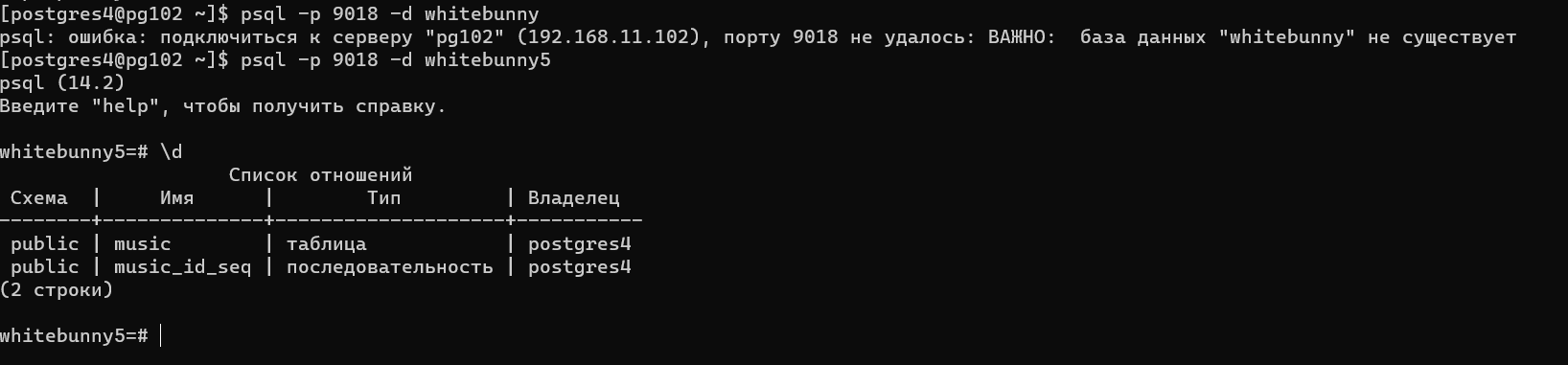
Передаем данные из резервной копии и меняем директорию, так как PGDATA не доступна

**3.3 Запуск СУБД, проверка результатов**

****

Экспортим все данные, не забыва поменять PGDATA и видим что все работает

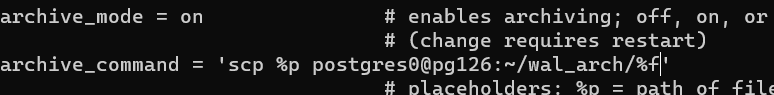
Теперь проверяем данные

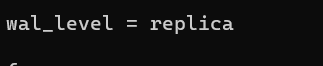


Видим что все в порядке, подъем бэкапа прошел успешно

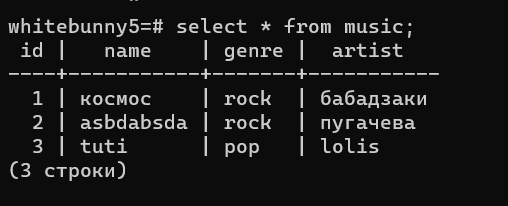
**Задание 4. Логическое повреждение данных**

Так как требуется восстановление с использованием архивных WAL-файлов, в начале нужно включить архивирование. Изменим postgresql.conf

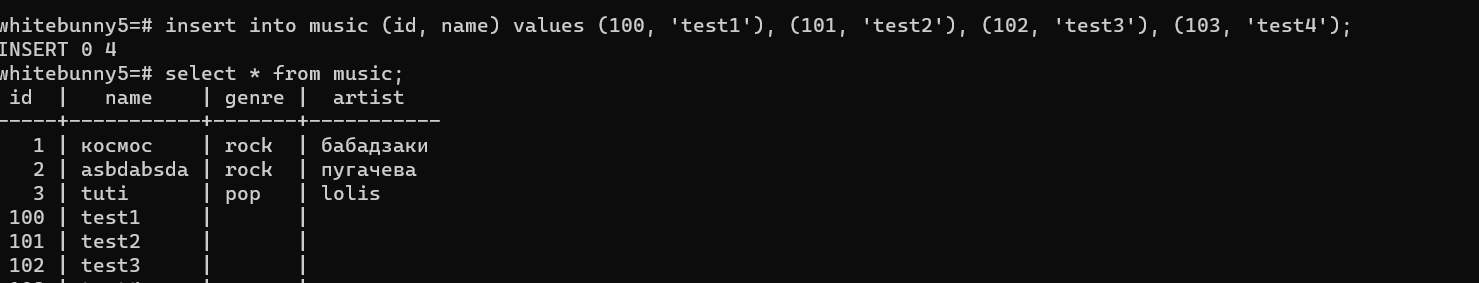
****

****

**4.1 В каждую таблицу базы добавить 2-3 новые строки**

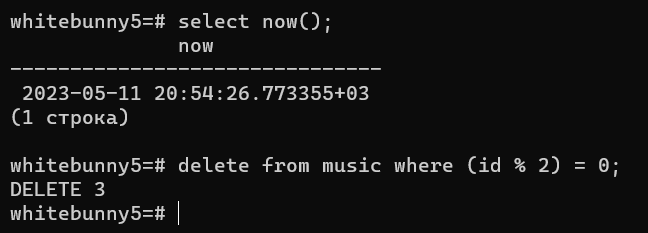
****

До изменений

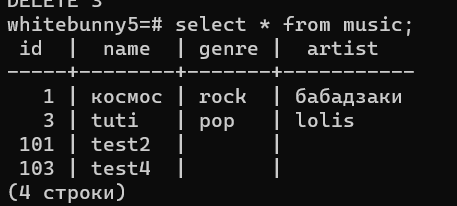


После изменений

**4.2 Зафиксировать время и симулировать ошибку**

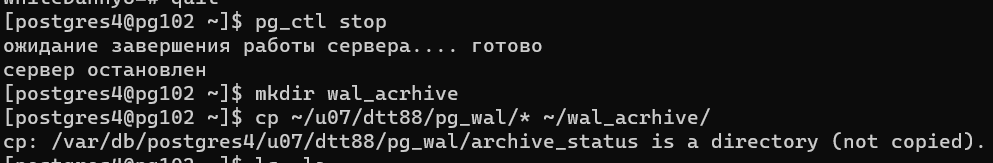
****

**4.3 Демонстрация результата**

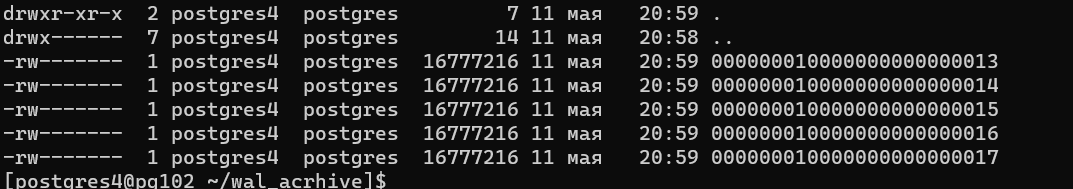
****

**4.4 Восстановление данных**

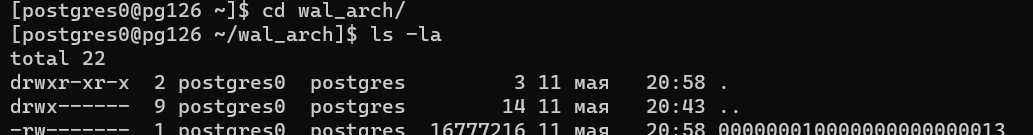
Для начала остановим сервер и навсякий скопируем pg\_wal файлы



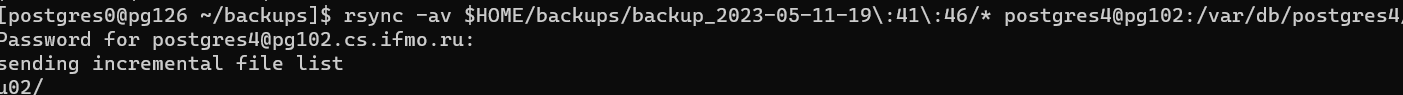
Теперь проверим что содержится в наших wal-фалайх



И что содержится в резервных wal-фалайх



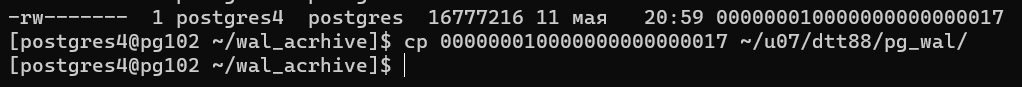
Как мы видим не все файлы у нас заархивированы.



Переносим бэкап на основной хост



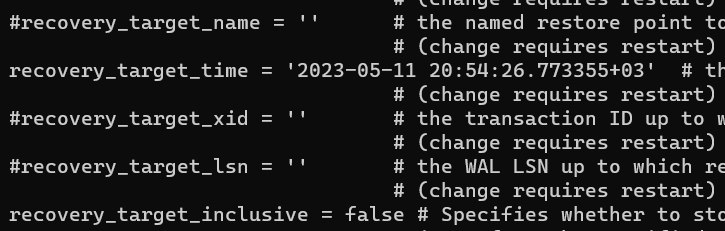
Удаляем все wal-файлы



Переносим последний wal-файл в pg\_wal

Установим время и команду восстановления в postgresql.conf

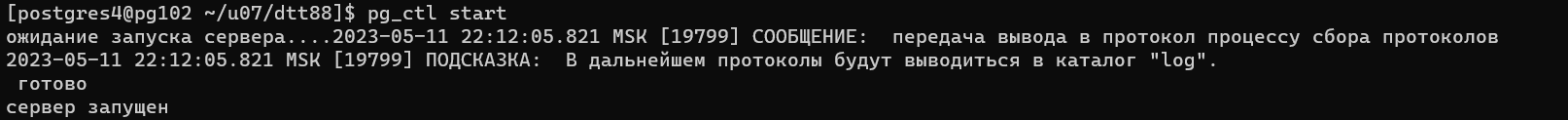




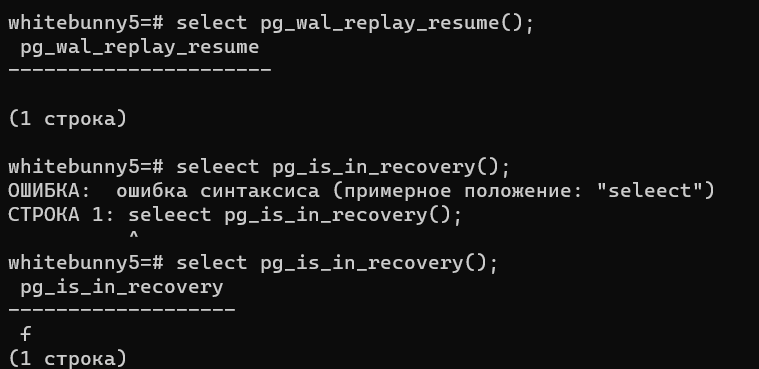
Создаем recovery.signal

Попробуем запустить сервер в режиме восстановления:

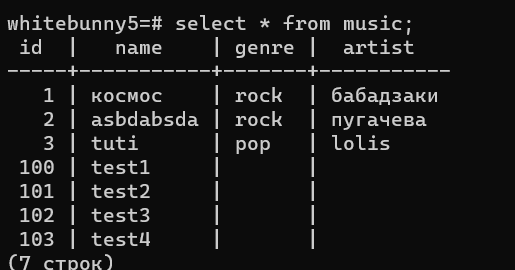
pg\_ctl start



Далее остается прописать команду pg\_wal\_replay\_resume() - это функция, которая используется для возобновления воспроизведения журнала транзакций



**4.5 Демонстрация результата**

****

При восстановлении логических повреждений данных wal - архивация полезна, так как помимо восстановления к самому последнему состоянию позволяет возвратиться в какой-то определенный момент времени.

**Вывод**

В ходе выполнения данной лабораторной работы я познакомился с прерывным бэкапом postgresql кластера, посмотрел на практике как его настроить и применять при различных сбоях: полной потери основного узла, повреждении файлов БД или логического повреждения данных. Во всех случаях WAL бэкапы решили задачу