УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Распределенные системы хранения данных»

**Лабораторная работа №3**

*Вариант 47*

Студент

*Кузнецов М. А.*

*P33131*

Преподаватель

*Шешуков Д. М.*

Санкт-Петербург, 2023 г.

Описание 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.  
  
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 обособленные (standalone) полные копии.

Полное резервное копирование (pg\_basebackup) по расписанию (cron) два раза в

сутки. Необходимые файлы WAL должны быть в составе полной копии, отдельно их не

архивировать. Срок хранения копий на основной системе - 1 неделя, на резервной - 1

месяц. 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 новых данных в БД за сутки: ~650 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 конфигурационных файлов СУБД 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:

Удалить любые две таблицы (DROP TABLE)

4.3 Продемонстрировать результат.

4.4 Выполнить восстановление данных указанным способом.

4.5 Продемонстрировать и проанализировать результат.

Этап 1. Резервное копирование

*Создаем пользователя с replicant*

create role replicant with replication login password ‘123’;

*В файле postgresql.conf*:

wal\_level = replica

*Подход -- создадим bash скрипт, в котором пропишем следующий код:*

#!/bin/bash

backup\_dir="/var/db/postgres2/u06/dir6/backups"

current\_datetime=$(date +'%Y%m%d\_%H%M')

backup\_name="backup\_$current\_datetime"

# Проверяем, существует ли директория backups

if [ ! -d "$backup\_dir" ]; then

mkdir -p "$backup\_dir"

fi

# Создаем новую папку с именем вида "backup\_20230522\_1200"

new\_backup\_dir="$backup\_dir/$backup\_name"

mkdir "$new\_backup\_dir"

echo "Created backup directory: $new\_backup\_dir"

# Выполняем команду pg\_basebackup и записываем резервную копию в новую папку

pg\_basebackup --progress -p 9054 -U replicant --format=t --wal-method=fetch -D "$new\_backup\_dir"

# Копируем

scp -r "$new\_backup\_dir" postgres2@pg132:backups/

Добавим в файл pg\_hba.conf также REPLICATION:

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Для автоматического удаления:

0 12 \* \* \* find /var/db/postgres2/u06/dir6/backups -type d -name "backup\_\*" -mtime +7 -exec rm -rf {} \;/

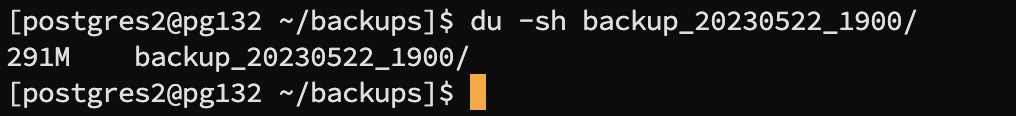
*Также сделаем ssh ключ, чтобы не требовался пароль для отправки файлов на резервный узел.*

*ssh-keygen -t rsa*

*ssh-copy-id -I ~/.ssh/id\_rsa.pub postgres2@132*

*Подсчет размера копий*

Бекап весит:



*a1 = 291 Mb*

*d = 650 Mb*

*n = 30 days*

*S\_30 = (2\*291+650\*29)/2 \* 30 = 582966 Mb = 569 GB*

Инкрементальный подход будет эффективнее.

Этап 2. Потеря основного узла

*Воссоздадим файловую структуру кластера для восстановления*

mkdir ~/u06/dir6

chmod 700 ~/u06/dir6

cd u06/dir6

tar xvf ~/backups/backup\_\*\*\*/base.tar

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Файлы табличного пространства*

mkdir ~/u06/dir7

cd ~/u06/dir7

tar xvf ~/backups/backup\_\*\*\*/16387.tar

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

*Отчищаем директорию для wal-файлов*

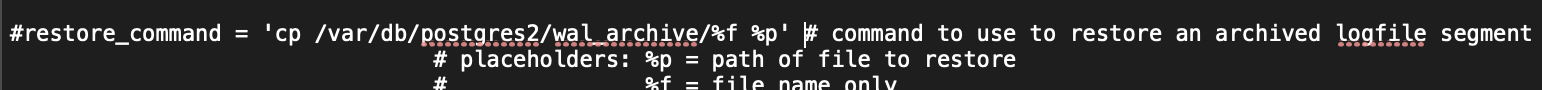
rm -rf ~/u06/dir6/pg\_wal/\*

ln -s /var/db/postgres2/u06/dir7 ~/u06/dir6/pg\_tblspc/16387

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

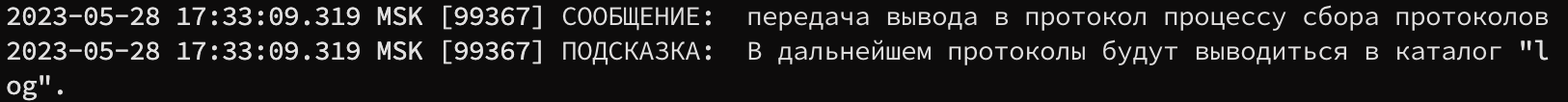
*Указываем команду для загрузки wal-файлов*



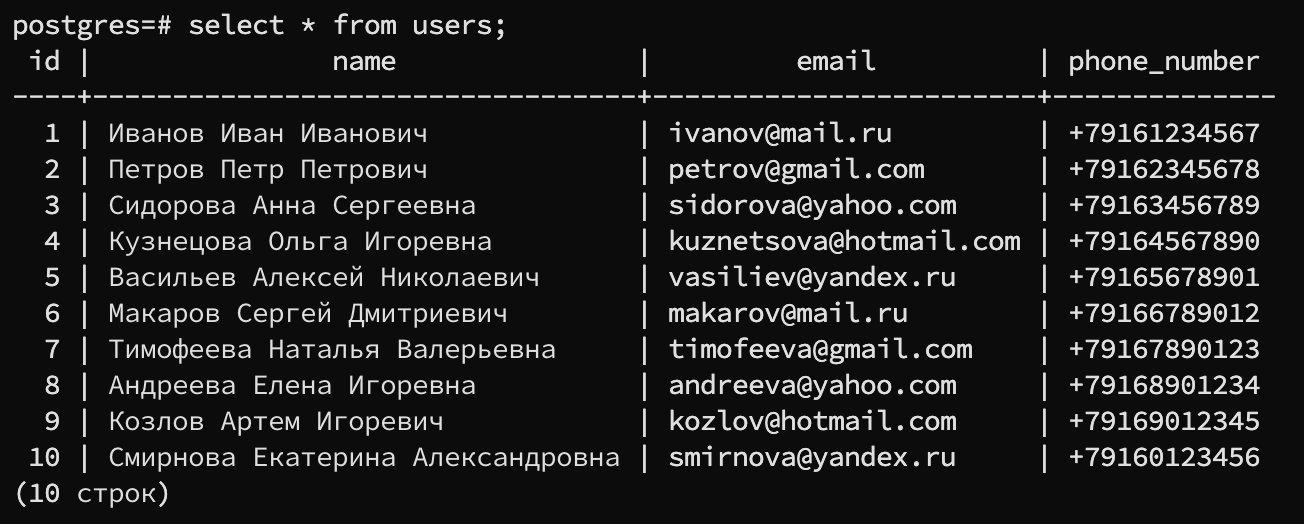
*Файл – флаг о восстановлении*

touch recovery.signal

*Запускаем кластер*



*Проверяем работоспособность*

**

Анализ выполнения

*Восстановление завершилось успешно. Но для корректного запуска необходимо подкорректировать ссылку*

Повреждение файлов БД

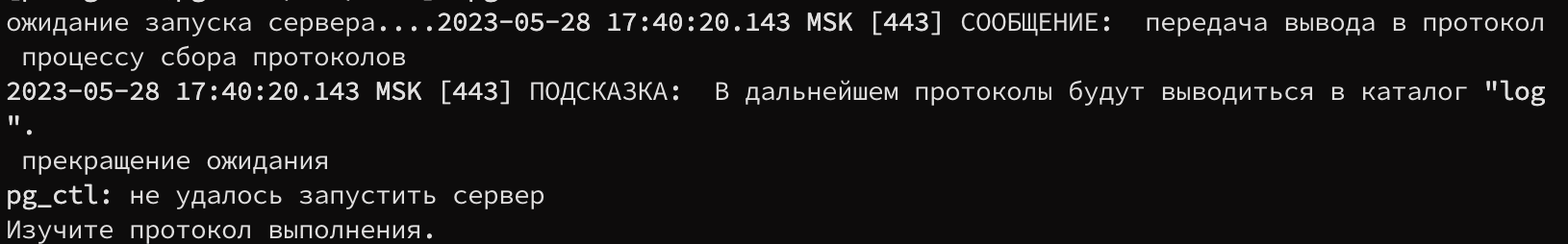
*Последняя актуальная копия находится на узле, скачивать еще раз нет смысла.*

*Отчищаем директорию для wal-файлов*

rm -rf ~/u06/dir6/pg\_wal/\*

*Подключаемся к базе, которая была в пространстве*





*Из-за того, что расположение директории PGDATA недоступно, распакуем копию в новое местоположение.*

mkdir dir6\_new

cd dir6\_new/

scp -r postgres2@pg132:u06/dir6/ ~u06/dir6\_new/

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

*Запускаем поврежденный кластер в режиме восстановления wal-файлов и указываем команду для восстановления*

restore\_command = ‘cp /var/db/postgres2/u06/dir6\_new/archives/%f %p’

*Запускаем в режиме восстановления*

touch recovery.signal

postgres -D ~/u06/dir6\_new/

*Проверяем работоспособность*

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, черный, Шрифт

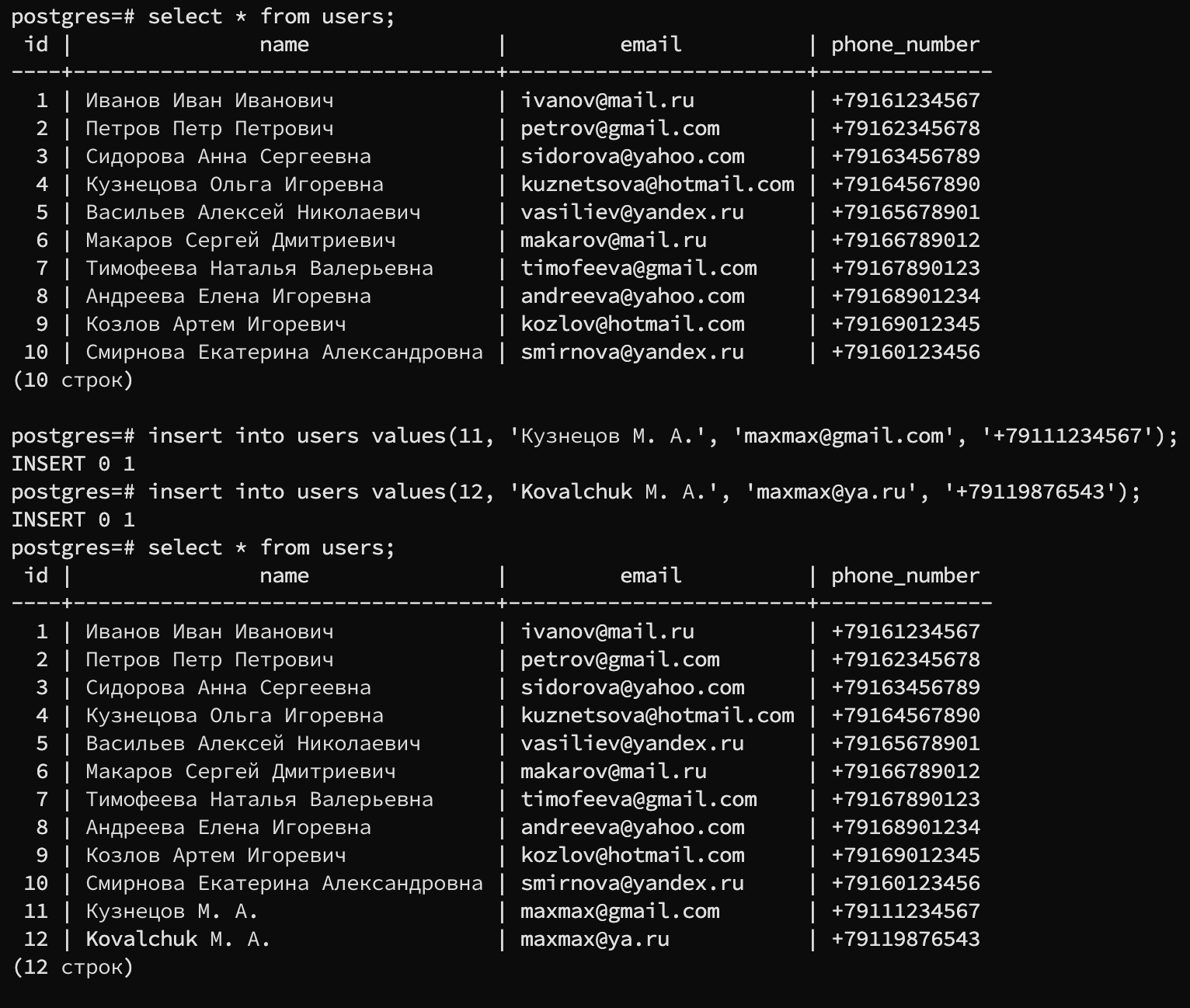
Автоматически созданное описание

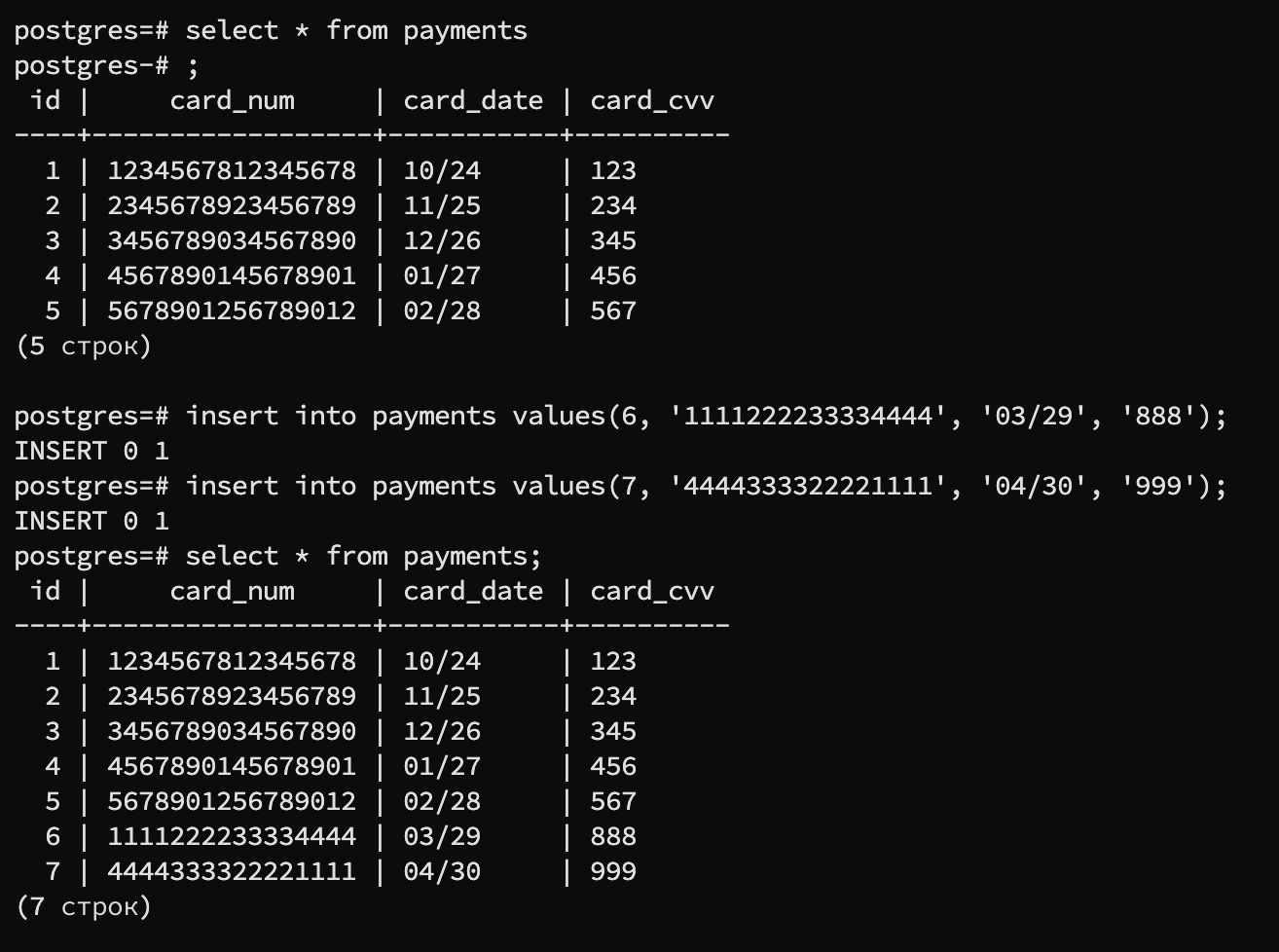
Анализ

*Восстановление завершилось успешно.*

Логическое повреждение данных

*Попробуем создать некорректные изменения*

**

*Фиксируем время*

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

*Удаляем таблички выше*

**

*Останавливаем сервер*

pg\_ctl stop

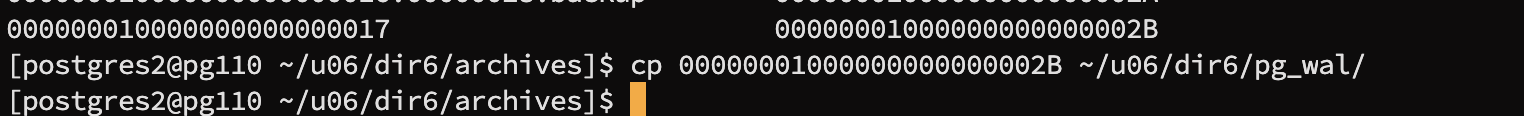
*Произведем синхронизацию между основным узлом и резервной копией*

scp postgres2@pg132:/u06/dir6/ /var/db/postgres2/u06/dir6/

*Очищаем директорию для wal-файлов*

rm -rf ~/u06/dir6/pg\_wal/\*

*Скопируем последний файл*

**

*Восстановление (прописываем в postgresql.conf)*

recovery\_target\_time = ‘2023-05-28 21:55:12.3683+03’

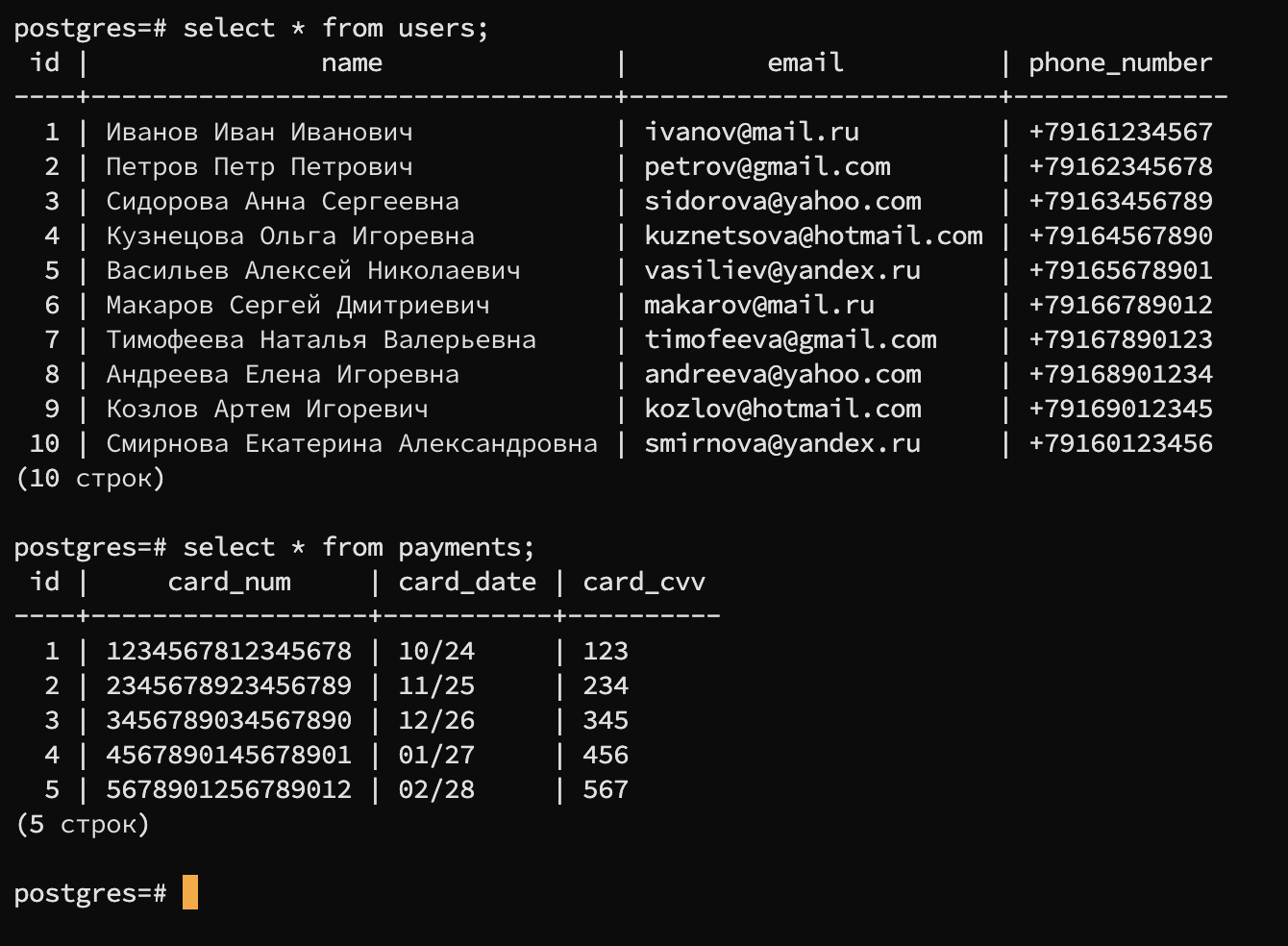
*Перезапустим базу с восстановлением из wal-файла*

touch recovery.signal

pg\_ctl start

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

*Проверяем*

**

*Восстановление прошло успешно*

Анализ

*Wal-архивация полезна, так как можно не только вернуться к последнему состоянию, но и также выбрать определенный момент времени*

Вывод

*Во время выполнения лабораторной работы я:*

1. *изучил способы непрерывного создания резервной копии кластера PostgreSQL.*
2. *настроил и применил его при различных сбоях.*