

Университет ИТМО  
Факультет ФПИ и КТ Р33131

**Отчет по лабораторной работе №3**  
**«Распределенные системы хранения**  
**данных»**

Вариант 796

Студент:

Чжоу Хунсян

Гр.Р33131

Преподаватель:

Афанасьев Дмитрий Борисович

ЦЕЛЬ РАБОТЫ .....	3
ЭТАП 1. ИНИЦИАЛИЗАЦИЯ КЛАСТЕРА БД.....	4
ЭТАП 2. КОНФИГУРАЦИЯ И ЗАПУСК СЕРВЕРА БД .....	5
ЭТАП 3. ДОПОЛНИТЕЛЬНЫЕ ТАБЛИЧНЫЕ ПРОСТРАНСТВА И НАПОЛНЕНИЕ БАЗЫ.....	9
ВЫВОД .....	18

## Цель работы

Настроить процедуру периодического резервного копирования базы данных, сконфигурированной в ходе выполнения лабораторной работы №2, а также разработать и отладить сценарии восстановления в случае сбоев.

Узел из предыдущей лабораторной работы используется в качестве основного. Новый узел используется в качестве резервного. Учётные данные для подключения к новому узлу выдаёт преподаватель. В сценариях восстановления необходимо использовать копию данных, полученную на первом этапе данной лабораторной работы.

pg117:postgres2:aXXt3haQ

pg120:postgres2:t2JiKjVc

## Этап 1. Резервное копирование

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

Первоначальная полная копия + непрерывное архивирование.

Включить для СУБД режим архивирования WAL; настроить копирование WAL (scp) на резервный узел; создать первоначальную резервную копию (pg\_basebackup), скопировать на резервный узел (rsync).

- Подсчитать, каков будет объем резервных копий спустя месяц работы системы, исходя из следующих условий:
  - Средний объем новых данных в БД за сутки: **100МБ**.
  - Средний объем измененных данных за сутки: **250МБ**.
- Проанализировать результаты.

### postgres2@pg117

```
ssh -J s336184@helios.cs.ifmo.ru:2222 postgres2@pg117
```

generate and send key to pg120 ssh key to ensure that archive command can be executed without password of pg120

```
[postgres2@pg117 ~]$ ssh-keygen -t rsa -b 4096 -C "postgres2@pg117"
Generating public/private rsa key pair.
Enter file in which to save the key (/var/db/postgres2/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/db/postgres2/.ssh/id_rsa.
Your public key has been saved in /var/db/postgres2/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:E/tnh4BM6nHeJrdS4cJzMirJEKZuAGz03aV5T9NLbWY postgres2@pg117
The key's randomart image is:
+---[RSA 4096]-----+
|
| .         .
| o . . . * . .
|.oo. . B =.o o E
|oo .  o.S.+o =
|o . . +*++o o
|o o ...oB= + .
| o + . .+ + .
|. . . .
+---[SHA256]-----+
[postgres2@pg117 ~]$
[postgres2@pg117 ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub postgres2@pg120
(postgres2@pg120) Password for postgres2@pg120.cs.ifmo.ru:
```

## step1\_config.sh (pg117)

```
# stop db
pg_ctl -D /var/db/postgres2/ewe49 stop

# remove old postgresql.conf with new postgresql.conf in script directory
rm /var/db/postgres2/ewe49/postgresql.conf
cp $(pwd)/postgresql.conf /var/db/postgres2/ewe49/

# start db
pg_ctl -D /var/db/postgres2/ewe49 start

# create tables
psql -U postgres2 -d postgres -p 9806 -f $(pwd)/set_test_table.sql

# check current wal file and archive status
psql -U postgres2 -d postgres -p 9806 -f $(pwd)/show_current_wal_file.sql
ls ~/svq55/archive_status

# use pg_basebackup to make a backup copy
# -T: rebind the tablespaces into new directories in backup files
# -Xs: send wal file in stream while backup is going
# -R: automatically set standby.signal to make that the backup runs as a standby node which is not
allow write operation and will get wal file from main node
# -P show backup progress
pg_basebackup -h localhost -p 9806 -D /var/db/postgres2/pg_backup -U postgres2 -Xs -R -P -T
/var/db/postgres2/gje71=/var/db/postgres2/pg_backup/gje71 -T
/var/db/postgres2/xca33=/var/db/postgres2/pg_backup/xca33

# check current wal file and archive status again
psql -U postgres2 -d postgres -p 9806 -f $(pwd)/show_current_wal_file.sql
ls ~/svq55/archive_status

# use rsync to send backup to pg120
rsync -avz --quiet /var/db/postgres2/pg_backup/ postgres2@pg120:/var/db/postgres2/pg_backup/

# check current wal file and archive status again
psql -U postgres2 -d postgres -p 9806 -f $(pwd)/show_current_wal_file.sql
ls ~/svq55/archive_status

# insert new data after pg_basebackup to check whether new change will be sync to pg120 by wal
archive
psql -U postgres2 -d postgres -p 9806 -c "INSERT INTO test_table (data) SELECT md5(random()::text)
FROM generate_series(1, 100);"
psql -U postgres2 -d postgres -p 9806 -c "SELECT * FROM test_table LIMIT 10;

# check current wal file and archive status again
psql -U postgres2 -d postgres -p 9806 -f $(pwd)/show_current_wal_file.sql
ls ~/svq55/archive_status
```

## **\$(pwd)/postgresql.conf** (pg117)

enable archive mode to archive wal file to pg120

use scp command as archive command to send wal files to pg120

```
archive_mode = on
archive_command = 'scp %p postgres2@pg120:/var/db/postgres2/wal_files/%f'
```

## **set\_test\_table.sql** (pg117)

```
--table for test
CREATE TABLE test_table (id serial PRIMARY KEY, data text);

SELECT * FROM test_table;

--table with fkey for step4
```

```
CREATE TABLE test_fkey_table (
    id serial PRIMARY KEY,
    data text
);

ALTER TABLE test_fkey_table ADD COLUMN test_table_id INT REFERENCES test_table(id);

SELECT * FROM test_fkey_table;
```

## show\_current\_wal\_file.sql (pg117)

```
DO $$
DECLARE
    current_lsn pg_lsn;
    wal_file_name text;
BEGIN
    -- ????? WAL LSN
    SELECT pg_current_wal_lsn() INTO current_lsn;

    -- ?? LSN ????? WAL ???
    SELECT pg_walfile_name(current_lsn) INTO wal_file_name;

    -- ???
    RAISE NOTICE 'Current WAL LSN: %', current_lsn;
    RAISE NOTICE 'Corresponding WAL file: %', wal_file_name;
END $$;
```

## postgres2@pg120

### step1.sh(pg120)

```
# remove old postgresql.conf with new postgresql.conf in script directory
rm ~/pg_backup/postgresql.conf
cp $(pwd)/postgresql.conf ~/pg_backup/

# create recovery.signal to get new change from wal files which is from pg117
touch ~/pg_backup/recovery.signal

# restart db
pg_ctl -D ~/pg_backup stop
pg_ctl -D ~/pg_backup start
```

\$(pwd)/postgresql.conf(pg120)

set the postgresql.conf on pg120 backup to make sure that archive mode is off, so that database will not create new wal files and only get them from pg117

```
archive_mode = off
#archive_command = 'scp %p postgres2@pg120:/var/db/postgres2/wal_files/%f'
restore_command = 'cp /var/db/postgres2/wal_file/%f %p'
```

## Result

```
[postgres2@pg117 ~/script_lab3]$ ./step1_config.sh
ожидание завершения работы сервера..... готово
сервер остановлен
ожидание запуска сервера....2024-09-23 13:47:14.502 MSK [90643] СООБЩЕНИЕ: передача вывода в
протокол процессу сбора протоколов
2024-09-23 13:47:14.502 MSK [90643] ПОДСКАЗКА: В дальнейшем протоколы будут выводиться в каталог
"log".
готово
сервер запущен
CREATE TABLE
  id | data
-----+-----
(0 строк)

CREATE TABLE
ALTER TABLE
  id | data | test_table_id
-----+-----+-----
(0 строк)

psql:/var/db/postgres2/script_lab3/show_current_wal_file.sql:15: ЗАМЕЧАНИЕ: Current WAL LSN:
0/1996D38
psql:/var/db/postgres2/script_lab3/show_current_wal_file.sql:15: ЗАМЕЧАНИЕ: Corresponding WAL file:
00000001000000000000000000000001
D0
31394/31394 КБ (100%), табличное пространство 3/3
psql:/var/db/postgres2/script_lab3/show_current_wal_file.sql:15: ЗАМЕЧАНИЕ: Current WAL LSN:
0/3000000
psql:/var/db/postgres2/script_lab3/show_current_wal_file.sql:15: ЗАМЕЧАНИЕ: Corresponding WAL file:
00000001000000000000000000000002
D0
00000001000000000000000000000002.00000028.backup.done 00000001000000000000000000000002.done
psql:/var/db/postgres2/script_lab3/show_current_wal_file.sql:15: ЗАМЕЧАНИЕ: Current WAL LSN:
0/3000060
psql:/var/db/postgres2/script_lab3/show_current_wal_file.sql:15: ЗАМЕЧАНИЕ: Corresponding WAL file:
00000001000000000000000000000003
D0
00000001000000000000000000000002.00000028.backup.done 00000001000000000000000000000002.done
INSERT 0 100
./step1_config.sh: строка 38: неожиданный конец файла во время поиска <<">>
[postgres2@pg117 ~/script_lab3]$
```

## Расчёты

In order to calculate the size of db backup after one month, we may divide it into two parts:

1. Initial full backup: this is the backup that created by pg\_basebackup, at the beginning of the first month
2. WAL archive: while db is running, WAL files will be created everyday, it records all the changes of data. so we need to calculate this part

As the condition given above, we may calculate:

- New data everyday: 100MB
- Changed data everyday: 250
- Duration: 30 days

Since we don't know the size of initial backup created by pg\_backup, we set the value as 500MB. And then we calculate the WAL size as follow

- WAL size per day:  $100\text{MB}(\text{New Data}) + 250(\text{Changed Data}) = 350\text{MB} / \text{per day}$
- WAL size per month:  $350\text{MB} * 30 = 10500\text{MB}$
- Total backup size: initial backup size + WAL size per month = 11000MB



## Этап 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.

### postgres2@pg120

```
ssh -J s336184@helios.cs.ifmo.ru:2222 postgres2@pg120
```

### Result

```
[postgres2@pg120 ~/script_lab3]$ ./step1_config.sh
pg_ctl: файл PID "/var/db/postgres2/pg_backup/postmaster.pid" не существует
Запущен ли сервер?
ожидание запуска сервера....2024-09-23 13:54:01.792 MSK [93235] СООБЩЕНИЕ:  передача вывода в
протокол процессу сбора протоколов
2024-09-23 13:54:01.792 MSK [93235] ПОДСКАЗКА:  В дальнейшем протоколы будут выводиться в каталог
"log".
    готово
сервер запущен
[postgres2@pg120 ~/script_lab3]$
```

We can see that in new data inserted in pg117 after pg\_basebackup now shows in pg120, it shows that wal files work correctly

```
[postgres2@pg120 ~/script_lab3]$ pg_ctl -D ~/pg_backup restart
ожидание завершения работы сервера.... готово
сервер остановлен
ожидание запуска сервера....2024-09-23 13:56:21.931 MSK [93872] СООБЩЕНИЕ:  передача вывода в
протокол процессу сбора протоколов
2024-09-23 13:56:21.931 MSK [93872] ПОДСКАЗКА:  В дальнейшем протоколы будут выводиться в каталог
"log".
    готово
сервер запущен
```

```
postgres=# select * from test_table limit 10;
 id | data
-----+-----
  1 | 74c35c3dd378bd92a6eac025060b7dbf
  2 | fab8bc88a855df22ea500f8746fb73a2
  3 | 06e88807f2847df70d99a8f584d73cb3
  4 | 05cad7f12fc77d697713908d93425dda
  5 | f2e057a636d1d51fc9b9a1fcf2789b6b
  6 | a76e2a6953b52c67d224af51303a594c
  7 | bbe3b97e7b497ce5537154057ff3eacc
  8 | c0974d7122c1d099a1afd08da62dbbb6
  9 | a4601464d556a677850bde87d8eb516e
 10 | 804422addc8ac6b5a348ed449c0dcc15
(10 строк)
```

## Этап 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:

- Симулировать сбой:
  - удалить с диска директорию любой таблицы со всем содержимым.
- Проверить работу СУБД, доступность данных, перезапустить СУБД, проанализировать результаты.
- Выполнить восстановление данных из резервной копии, учитывая следующее условие:
  - исходное расположение директории PGDATA недоступно - разместить данные в другой директории и скорректировать конфигурацию.
- Запустить СУБД, проверить работу и доступность данных, проанализировать результаты.

**postgres2@pg117**

### **step3\_delete\_table\_file.sh(pg117)**

```
# input DB_NAME and TALE_NAME through parameters
DB_NAME=$1
TABLE_NAME=$2
PG_USER="postgres2"

# restart db
pg_ctl -D ~/ewe49 restart

# get TABLE_OID and DB_OID from sql requests
# -tA: remove empty space from results
TABLE_OID=$(psql -U $PG_USER -d $DB_NAME -p 9806 -tA -c "SELECT oid FROM pg_class WHERE relname = '$TABLE_NAME';")
DB_OID=$(psql -U $PG_USER -d $DB_NAME -p 9806 -tA -c "SELECT oid FROM pg_database WHERE datname = '$DB_NAME';")

# get and print TABLE_FILE_PATH
TABLE_FILE_PATH="~/ewe49/base/$DB_OID/$TABLE_OID"
echo "TABLE FILE PATH: $TABLE_FILE_PATH"

# show table which will be deleted
psql -U postgres2 -d $DB_NAME -p 9806 -c "SELECT * FROM $TABLE_NAME LIMIT 5;"

# locate and remove table file
cd ~/ewe49/base/$DB_OID
rm $TABLE_OID

# show table which will is already deleted, it will show error here because table file is deleted
psql -U postgres2 -d $DB_NAME -p 9806 -c "SELECT * FROM $TABLE_NAME LIMIT 5;"

echo "File deleted"
```

## Result

As we can see it shows error when access to table after table file was deleted

```
[postgres2@pg117 ~/script_lab3]$ ./step3_delete_table_file.sh postgres test_table
ожидание завершения работы сервера.... готово
сервер остановлен
ожидание запуска сервера....2024-09-23 14:06:27.486 MSK [96717] СООБЩЕНИЕ:  передача вывода в
протокол процессу сбора протоколов
2024-09-23 14:06:27.486 MSK [96717] ПОДСКАЗКА:  В дальнейшем протоколы будут выводиться в каталог
"log".
готово
сервер запущен
TABLE FILE PATH: ~/ewe49/base/5/16412
id | data
-----+-----
1 | 74c35c3dd378bd92a6eac025060b7dbf
2 | fab8bc88a855df22ea500f8746fb73a2
3 | 06e88807f2847df70d99a8f584d73cb3
4 | 05cad7f12fc77d697713908d93425dda
5 | f2e057a636d1d51fc9b9a1fcf2789b6b
(5 строк)

ОШИБКА:  не удалось открыть файл "base/5/16412": Нет такого файла или каталога
File deleted
[postgres2@pg117 ~/script_lab3]$
```

## Restore

### step3\_restore.sh(pg117)

```
# stop db on pg117
pg_ctl -D ~/ewe49 stop

# make new directory to store pgdata from second node
mkdir ~/pgdata_new

# remote connect to pg120 and stop db
ssh postgres2@pg120 "pg_ctl status -D ~/pg_backup"
ssh postgres2@pg120 "pg_ctl stop -D ~/pg_backup"

# use rsync to get pg_backup from pg120
rsync -avz --quiet postgres2@pg120:/var/db/postgres2/pg_backup/ ~/pgdata_new

# remove recovery.signal and standby.signal and reset postgresql.conf
rm ~/pgdata_new/postgresql.conf
rm ~/pgdata_new/recovery.signal
rm ~/pgdata_new/standby.signal

cp $(pwd)/postgresql.conf ~/pgdata_new/
```

### reset postgresql.conf

```
archive_mode = on
archive_command = 'scp %p postgres2@pg120:/var/db/postgres2/wal_files/%f'
data_directory = '/var/db/postgres2/pgdata_new' # use data in another directory
```

## Result

As we can see, the data of test\_table now is restored from pg120

```
[postgres2@pg117 ~]$ pg_ctl restart -D ~/pgdata_new
ожидание завершения работы сервера.... готово
сервер остановлен
ожидание запуска сервера....2024-09-23 14:21:44.370 MSK [875] СООБЩЕНИЕ:  передача вывода в протокол
процессу сбора протоколов
2024-09-23 14:21:44.370 MSK [875] ПОДСКАЗКА:  В дальнейшем протоколы будут выводиться в каталог
"log".
готово
сервер запущен[postgres2@pg117 ~]$ psql -U postgres2 -p 9806 -d postgres
psql (16.4)
Введите "help", чтобы получить справку.

postgres=# select * from test_table limit 10;
 id | data
-----+-----
  1 | 74c35c3dd378bd92a6eac025060b7dbf
  2 | fab8bc88a855df22ea500f8746fb73a2
  3 | 06e88807f2847df70d99a8f584d73cb3
  4 | 05cad7f12fc77d697713908d93425dda
  5 | f2e057a636d1d51fc9b9a1fcf2789b6b
  6 | a76e2a6953b52c67d224af51303a594c
  7 | bbe3b97e7b497ce5537154057ff3eacc
  8 | c0974d7122c1d099a1afd08da62dbbb6
  9 | a4601464d556a677850bde87d8eb516e
 10 | 804422addc8ac6b5a348ed449c0dcc15
(10 строк)

postgres=# \q
```

## Этап 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:

- Генерация файла на резервном узле с помощью pg\_dump и последующее применение файла на основном узле.

Ход работы:

- В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.
- Зафиксировать время и симулировать ошибку:
  - в любой таблице с внешними ключами подменить значения ключей на случайные (INSERT, UPDATE)
- Продемонстрировать результат.
- Выполнить восстановление данных указанным способом.
- Продемонстрировать и проанализировать результат.

### Mess up

**postgres2@pg117**

**step4\_messup.sh(pg117)**

Insert new data into tables and then messed them up

```
pg_ctl -D ~/pgdata_new restart

# insert new data into tables
psql -U postgres2 -d postgres -p 9806 -c "INSERT INTO test_table(id, data) VALUES (11, 'test_data1'), (12, 'test_data2'), (13, 'test_data3');"

psql -U postgres2 -d postgres -p 9806 -c "SELECT * FROM test_table;"

psql -U postgres2 -d postgres -p 9806 -c "INSERT INTO test_fkey_table(data, test_table_id) VALUES ('test_data1', 11), ('test_data2', 12), ('test_data3', 13);"

psql -U postgres2 -d postgres -p 9806 -c "SELECT * FROM test_fkey_table;"

pg_ctl -D ~/pgdata_new restart

# mess up fkey with random value
psql -U postgres2 -d postgres -p 9806 -f $(pwd)/set_random_fkey.sql
```

**set\_random\_fkey.sql (pg117)**

```
UPDATE test_fkey_table
SET test_table_id = FLOOR (RANDOM() * 100) + 1
WHERE test_table_id IS NOT NULL;

SELECT * FROM test_fkey_table;
```

```
SELECT * FROM test_table
ORDER BY id DESC
LIMIT 5;
```

## Restore

### postgres2@pg120

#### postgresql.conf(pg120)

```
restore_command = 'scp postgres2@pg120:/path/to/archive/%f %p'
recovery_target_time = '2024-09-23 15:19:35.825293+03' # time stamp when data was not messed up
recovery_target_inclusive = off
```

#### step4\_dump.sh

```
# use pg_dump to dump db
pg_dump -U postgres2 -p 9806 -d postgres > ~/backup.sql

# use scp to send backup.sql to pg117:~/
scp ~/backup.sql pg117:~/
```

### postgres2@pg117

#### step4\_restore.sh(pg117)

```
psql -U postgres2 -d postgres -p 9806 -f ~/backup.sql
```

#### postgresql.conf(pg120)

```
restore_command = 'scp postgres2@pg120:/path/to/archive/%f %p'
recovery_target_time = '2024-09-23 15:19:35.825293+03' # time stamp when data was not messed up
```

## Result

### postgres2@pg117

messup fkey with random value

```
[postgres2@pg117 ~/script_lab3]$ ./step4_messup.sh
ожидание завершения работы сервера..... готово
сервер остановлен
ожидание запуска сервера....2024-09-24 15:37:04.597 MSK [78745] СООБЩЕНИЕ:  передача вывода в
протокол процессу сбора протоколов
2024-09-24 15:37:04.597 MSK [78745] ПОДСКАЗКА:  В дальнейшем протоколы будут выводиться в каталог
"log".
готово
сервер запущен
INSERT 0 3
id | data
-----+-----
 1 | 8506e096052496c7a34395453a7b5e31
 2 | 196b0a8c9c9ac8fc16c05c8cf29c8994
 3 | 5f4abaff49af0352ed7568ea293a5d3f
 4 | bb179105354f4955cac489a1fa2ccc5a
 5 | fa18433dbd56376d5ef538202425d70c
 6 | 0ddc03730cda62da5e90d2a95e27baea
 7 | 1ffaca4224a46679bdf1a9faf12b82d6
 8 | 994e7ca1710f9e53b6559488c4b8d3ff
 9 | 9e89f28911e0b983af8d717e6b026d01
10 | 0561f6c0d1835ae49248b2ff8d7ef750
11 | test_data1
12 | test_data2
13 | test_data3
(13 строк)

INSERT 0 3
id | data | test_table_id
-----+-----+-----
 1 | test_data1 | 11
 2 | test_data2 | 12
 3 | test_data3 | 13
(3 строки)

now | pg_current_wal_lsn
-----+-----
2024-09-24 15:37:04.777004+03 | 0/5000988
(1 строка)

UPDATE 3
id | data | test_table_id
-----+-----+-----
 1 | test_data1 | 2
 2 | test_data2 | 7
 3 | test_data3 | 2
(3 строки)
```

restore from pg\_dump and wal

We first remove old table and then use backup.sql to restore

```
[postgres2@pg117 ~/script_lab3]$ psql -U postgres2 -d postgres -p 9806
psql (16.4)
Введите "help", чтобы получить справку.

postgres=# drop table test_table cascade;
ЗАМЕЧАНИЕ:  удаление распространяется на объект ограничение test_fkey_table_test_table_id_fkey в
отношении таблица test_fkey_table
DROP TABLE
postgres=# drop table test_fkey_table;
```

```

DROP TABLE
postgres=# \dt
Отношения не найдены.
postgres=# \q

[postgres2@pg117 ~/script_lab3]$ psql -U postgres2 -d postgres -p 9806 -f ~/backup.sql
SET
SET
SET
SET
SET
set_config
-----

(1 строка)

SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER SEQUENCE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER SEQUENCE
ALTER SEQUENCE
ALTER TABLE
ALTER TABLE
COPY 3
COPY 13
setval
-----
33
(1 строка)

setval
-----
33
(1 строка)

ALTER TABLE
ALTER TABLE
ALTER TABLE

```

As we can see the fkey value is back as normal

```

[postgres2@pg117 ~/script_lab3]$ psql -U postgres2 -d postgres -p 9806
psql (16.4)
Введите "help", чтобы получить справку.

postgres=# select * from test_table;
 id | data
-----+-----
  1 | 8506e096052496c7a34395453a7b5e31
  2 | 196b0a8c9c9ac8fc16c05c8cf29c8994
  3 | 5f4abaff49af0352ed7568ea293a5d3f
  4 | bb179105354f4955cac489a1fa2ccc5a
  5 | fa18433dbd56376d5ef538202425d70c
  6 | 0ddc03730cda62da5e90d2a95e27baea
  7 | 1ffaca4224a46679bdf1a9faf12b82d6
  8 | 994e7ca1710f9e53b6559488c4b8d3ff
  9 | 9e89f28911e0b983af8d717e6b026d01
 10 | 0561f6c0d1835ae49248b2ff8d7ef750
 11 | test_data1
 12 | test_data2
 13 | test_data3
(13 строк)

```



```
postgres=# select * from test_fkey_table;
```

id	data	test_table_id
1	test_data1	11
2	test_data2	12
3	test_data3	13

(3 строки)

## **Вывод**

В ходе выполнения работы научился создавать, инициализировать, настраивать и использовать базы данных с помощью команд.