

Где коннект?



КАКОЙ



КОННЕКТ?

rixx

Управление подключением

```
psql -d database -h host -p port -U username
```

```
psql -h "localhost" -d "demo" -p 5432 -U "postgres"
```

```
psql -h "localhost" -d "demo" -U "postgres"
```

```
export PGHOST=localhost
```

```
export PGDATABASE=demo
```

```
export PGUSER=postgres
```

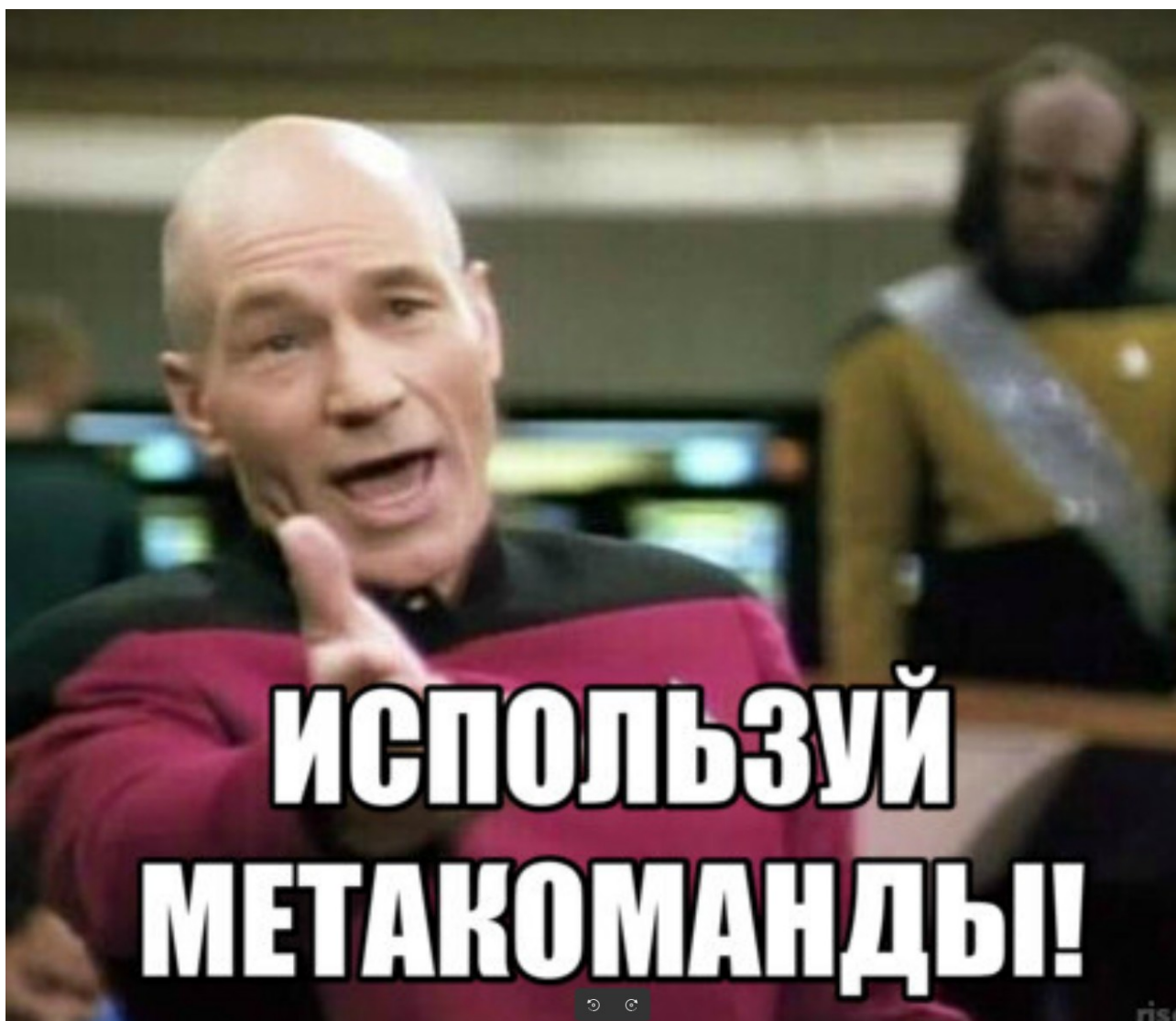
```
export PGPORT=5432
```

```
\c[onnect] database username host port
```

```
\c template1
```

```
\c postgres - - -
```

```
\conninfo
```



Meta команды

Для вывода списка баз данных используем \l

Для вывода списка схем базы данных используем \dn

Для вывода списка таблиц используем \dt

Для вывода описания таблицы используем \d participants



Выполнение команд SQL

```
select * from participants;
```

```
select *  
from participants;
```

```
\h CREATE TABLE
```

Управление выводом результатов запроса

`\pset`

`select * from participants;`

`\x`

`\pset expanded on`

`select * from participants;`

`\pset expanded off`

`select * from participants \gx`

`tuples_only`

`\t`

`select * from participants;`

`\pset tuples_only off`

`format`

`\a`

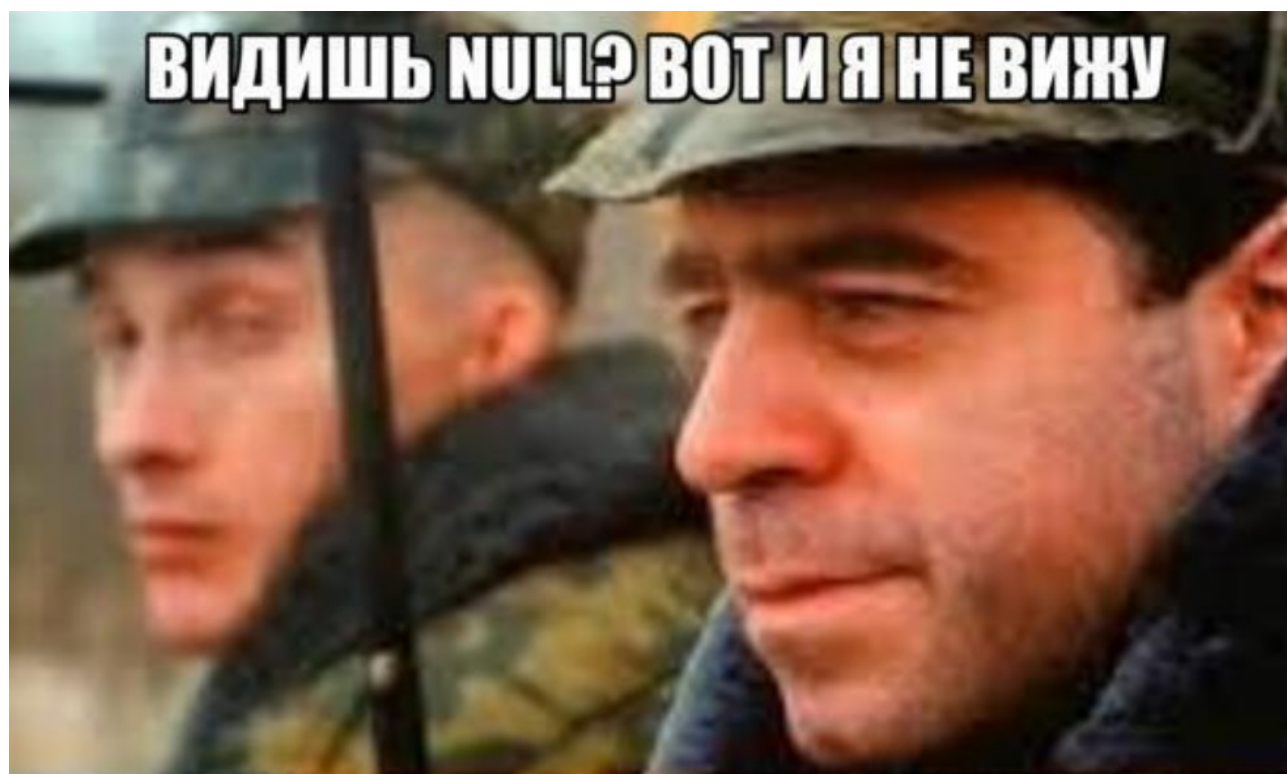
`select * from participants;`

`\H`

`select * from participants;`

`\pset format aligned`

ВИДИШЬ НУЛЛ? ВОТ И Я НЕ ВИЖУ



А ОН ЕСТЬ



NULL

```
select null;
```

```
\pset null '[null]'
```

```
select null;
```

```
\pset null ''
```

Выполнение команд shell

```
\! pwd
```

```
\! ls -la
```

```
\! printenv
```

Управление выводом

```
\o[ut] [ filename ]
```

```
\out psql.out
```

```
select * from participants;
```

```
select 1;
```

```
\! cat psql.ou
```

```
\o
```

```
select 1;
```



Управление вводом

```
\i
\! cat psql.input
\i psql.input
psql < psql.input
psql -f psql.input
```

Буфер запросов

```
select now();
\p
\g
\watch
\watch 1
```




Простая
таблица



Сводная
таблица

Команда \crosstabview

```
select * from lectures_ratings;
```

```
\crosstabview
```

```
select p.first_name || ' ' || p.last_name, l.topic, lr.rating from lectures_ratings lr, participants p,  
lectures l where lr.participant_id = p.id and lr.lecture_id = l.id order by p.id;
```

```
\crosstabview
```

Команда \dconfig

\dconfig
\dconfig *

Команда \timing

\timing

select pg_sleep(10);

Переменные psql

\set

\set VAR VALUE
\echo :VAR

\! echo \$USER

\unset VAR

\gset

select now() as curr_time \gset

\echo :curr_time

\set version 'SELECT version();'

:version

AUTOCOMMIT

psql сеанс 1	psql сеанс 2
	select * from event_staff; получаем ошибку - ERROR: relation "students" does not exist таблица не найдена
CREATE TABLE event_staff (id SERIAL, first_name TEXT, last_name TEXT, email TEXT); создадим таблицу event_staff	
select * from event_staff; получаем результат из 0 строк	
	select * from event_staff; получаем результат из 0 строк
\set AUTOCOMMIT off уберем опцию автоматического коммита	
INSERT INTO event_staff (first_name, last_name, email) VALUES ('Volunteer 1 FN', 'Volunteer 1 LN', '1@mail.ru');	
select * from event_staff; проверим результат видим запись в таблице	

	select * from event_staff; проверим результат получаем результат из 0 строк
demo=#> select * from event_staff; давайте обратим внимание на значок звездочки в командой строке. это означает, что мы находимся в открытой транзакции. поэтому в другом сеансе psql мы не видим результата.	
выполним команду для завершения транзакции: COMMIT;	
	select * from event_staff; проверим результат видим запись в таблице
\set AUTOCOMMIT off вернем опцию автоматического коммита	
drop table event_staff; удалим таблицу	
	select * from event_staff; получаем ошибку - ERROR: relation "students" does

	not exist таблица не найдена
--	------------------------------------

COMP_KEYWORD_CASE

\set COMP_KEYWORD_CASE upper

ON_ERROR_ROLLBACK

```
CREATE TABLE event_staff (id SERIAL, first_name TEXT, last_name TEXT, email TEXT);
```

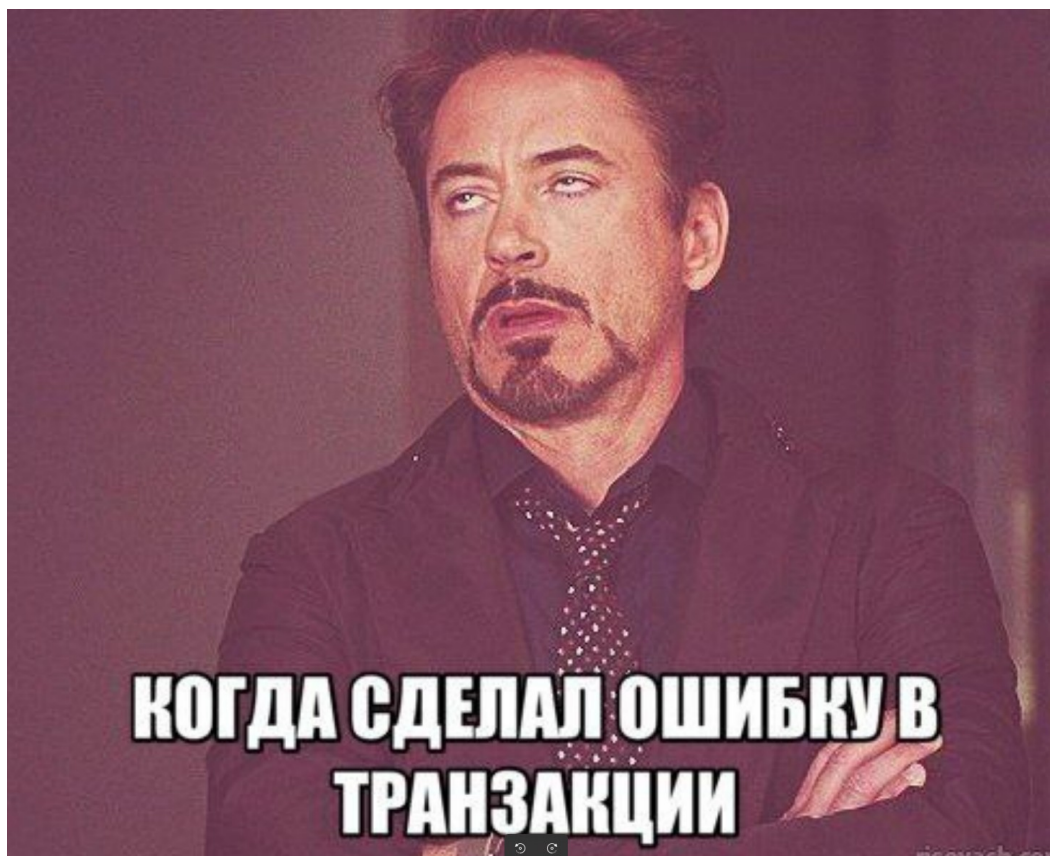
```
select * from event_staff;
```

```
begin;
```

```
INSERT INTO event_staff (first_name, last_name, email) VALUES ('Volunteer 1 FN', 'Volunteer 1  
LN', '1@mail.ru');
```

```
INSERT INTO event_staff (first_name, last_name, email) VALUES ('Volunteer 2 FN', 'Volunteer 2  
LN', '2@mail.ru');
```

```
INSERT INTO event_ staf (first_name, last_name, email) VALUES ('Volunteer 3 FN', 'Volunteer 3  
LN', '3@mail.ru');
```

demo=**|**>

ROLLBACK;

select * from event_staff;

\set ON_ERROR_ROLLBACK on

begin;

INSERT INTO event_staff (first_name, last_name, email) VALUES ('Volunteer 1 FN', 'Volunteer 1 LN', '1@mail.ru');

INSERT INTO event_staff (first_name, last_name, email) VALUES ('Volunteer 2 FN', 'Volunteer 2 LN', '2@mail.ru');

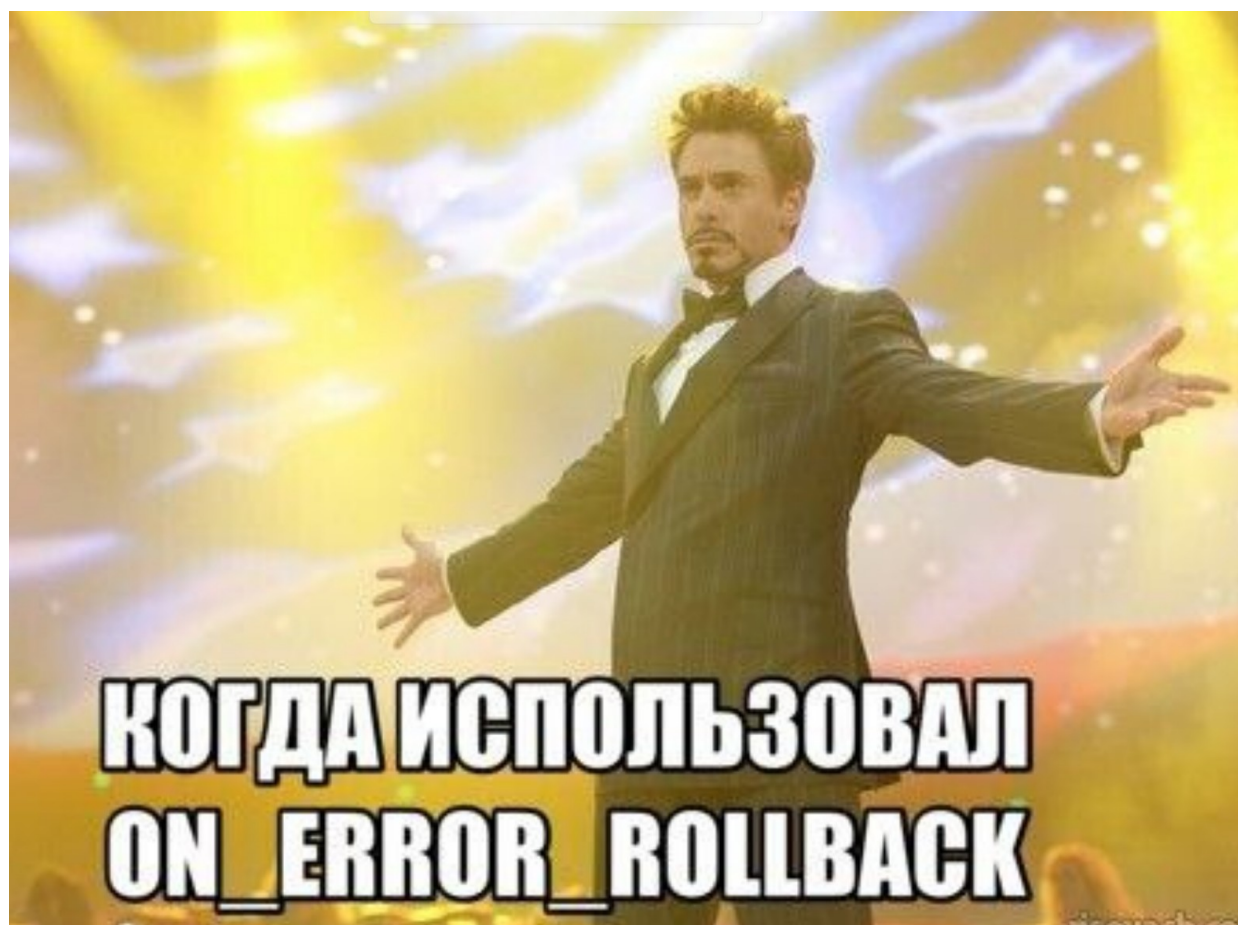
```
INSERT INTO event_staff (first_name, last_name, email) VALUES ('Volunteer 3 FN', 'Volunteer 3 LN', '3@mail.ru');
```

demo=*****>

```
INSERT INTO event_staff (first_name, last_name, email) VALUES ('Volunteer 3 FN', 'Volunteer 3 LN', '3@mail.ru');
```

```
COMMIT;
```

```
select * from event_staff;
```





Конфигурационные файлы psql

```
mv .psqlrc.bkp .psqlrc
```

```
\set QUIET 1
```

```
\set PROMPT1 '%M:%[%033[1;31m%]%>[%033[0m%] %n@%/%R%#%x '
```

```
\set PROMPT2 '%M %n@%/%R %# '
```

```
\pset null '[null]'
```

```
\set COMP_KEYWORD_CASE upper
```

```
\timing
```

```
\set HISTSIZE 2000
```

```
\x auto
```

```
\set VERBOSITY verbose
```

```
\echo 'Welcome to psql demonstration! \n'
```

```
\echo 'Type :version to see the PostgreSQL version. \n'
```

```
\echo 'Type \q to exit. \n'
```

```
\set version 'SELECT version();'
```

```
\set QUIET 0
```

```
:version
```

```
select * from pg_user;
```

```
select * from lectures_ratings;
```