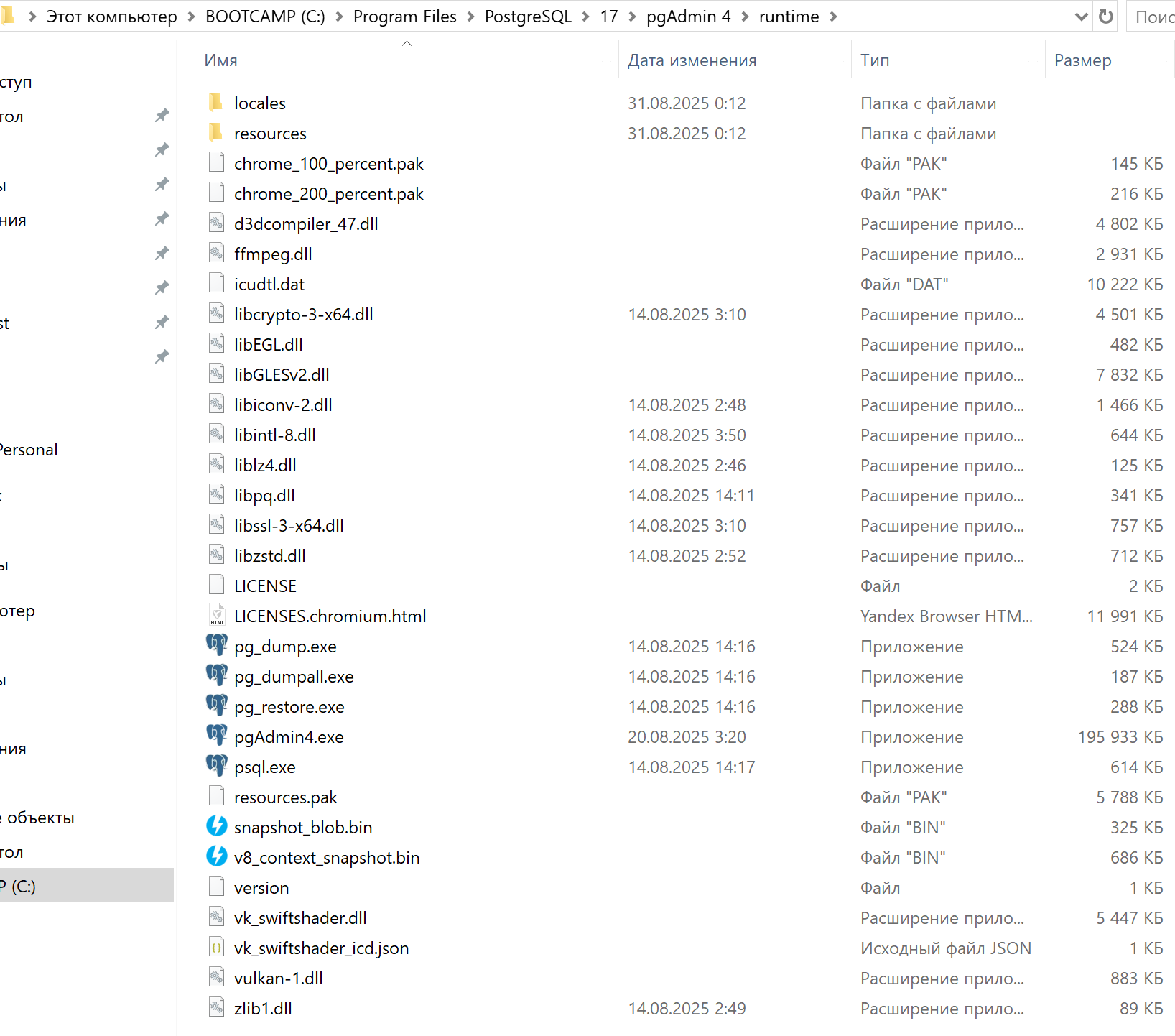
1. Установите интегрированную среду разработки:
   1. PgAdmin - https://www.pgadmin.org/download/pgadmin-4-windows/ .



1. Используя интегрированную среду разработки **PgAdmin** установите подключение к БД **dbSQL:**
   1. Имя сервера: **vpngw.avalon.ru**
   2. Порт сервера: **5432**
   3. Имя БД**: dbSQL**
   4. Имя пользователя: **pguser**
   5. Пароль**: Pa$$w0rd**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, снимок экрана, дисплей, число

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, Шрифт, линия, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.**

1. В интегрированной среде разработке **PgAdmin** правой кнопкой мыши нажмите на базу данных dbSQL и выберите **PSQL Tool**.

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

1. Изучите материал представленный по ссылке <https://tomcam.github.io/postgres/>

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Содержимое, созданное искусственным интеллектом, может быть неверным.

1. В окне консоли введите команду **\conninfo** и нажмите **ENTER**. Какой результат вы получили и что он означает?

Изображение выглядит как текст, Шрифт, число, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

[vpngw.avalon.ru]:5432pguser@dbSQL=> \conninfo

You are connected to database "dbSQL" as user "pguser" on host "vpngw.avalon.ru" (address "195.209.230.149") at port "5432".

1. **"You are connected to database 'dbSQL'"** – я в данный момент подключен именно к базе данных dbSQL.
2. **"as user 'pguser'"** – подключение выполнено от имени пользователя pguser. Это имя пользователя, заданное при настройке.
3. **"on host 'vpngw.avalon.ru' (address '195.209.230.149')"** – подключение установлено к серверу, который доступен по доменному имени vpngw.avalon.ru, а фактически его IP-адрес — 195.209.230.149.
4. **"at port '5432'"** – используется стандартный порт PostgreSQL — 5432.

Таким образом, команда \conninfo подтверждает успешное соединение с БД, а также показывает параметры текущего подключения.

1. Закройте подключение к БД:
   1. в строке приглашения введите команду **\q**
   2. нажмите **ENTER**



Изображение выглядит как текст, Шрифт, снимок экрана, белый

Содержимое, созданное искусственным интеллектом, может быть неверным.

1. Для получения информации о ключах утилиты **psql** введите в консоле **psql -?**

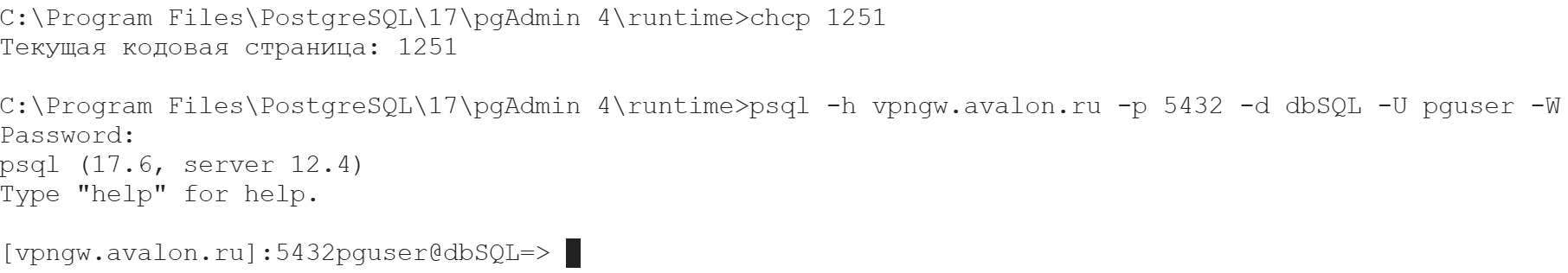
Изображение выглядит как текст, Шрифт, письмо, документ

Содержимое, созданное искусственным интеллектом, может быть неверным.

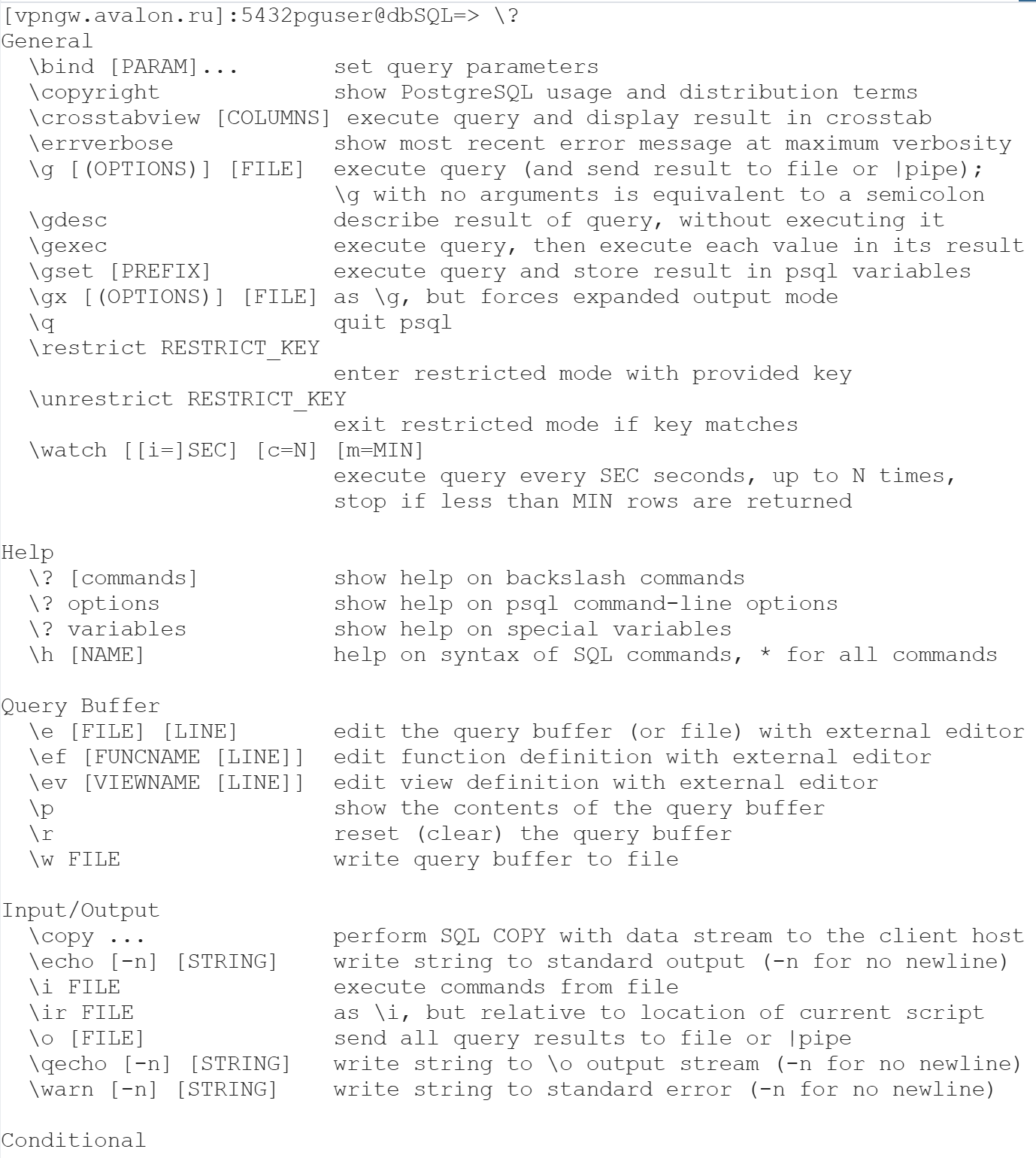
1. Какой ключ используется для указания:
   1. Имени сервера БД \_\_\_\_\_\_\_\_\_\_ -h
   2. Имени БД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -d
   3. Имени пользователя \_\_\_\_\_\_\_\_\_-U
   4. Пароля пользователя \_\_\_\_\_\_\_\_\_ -W
2. Используя соответствующие ключи выполните подключение к БД **dbSQL**.

chcp 1251

psql -h vpngw.avalon.ru -p 5432 -d dbSQL -U pguser -W



1. Изучите список команд **psql**. Для этого введите слэш-команду **\?**



1. С помощью каких команд можно получить:
   1. список баз данных \_\_\_\_\_\_\_\_\_\_\_\_\_\_\l

Изображение выглядит как текст, чек, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

* 1. список таблиц \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\dt

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

* 1. список схем\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\dn

Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

* 1. список индексов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\di

Изображение выглядит как текст, чек, Шрифт, белый

Содержимое, созданное искусственным интеллектом, может быть неверным.

* 1. список представлений \_\_\_\_\_\_\_\_\_\_\_\_\dv

Изображение выглядит как текст, Шрифт, чек, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

1. Выполните данные команды и посмотрите, какой результат получился.

выше

1. Запишите и выполните скрипт:

**SELECT schemaname, tablename, tableowner**

**FROM pg\_tables**

**LIMIT 7;**

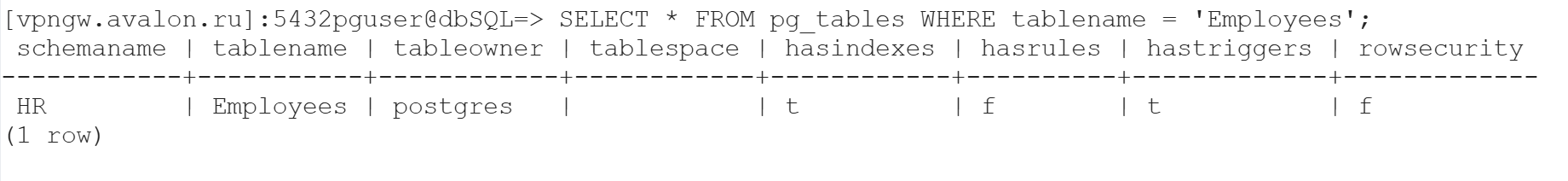
1. Какой получился результат? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Изображение выглядит как текст, чек, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

1. Запишите и выполните следующую команду:

**SELECT \* FROM pg\_tables WHERE tablename = 'Employees';**

****

1. Запишите и выполните следующую команду
2. (N.B.: чтобы показать предыдущий запрос нажмите на клавиатуре стрелку вверх и скорректируйте текст):

**SELECT \* FROM pg\_tables WHERE tablename = 'Employees' \gx**

**Изображение выглядит как текст, чек, снимок экрана, алгебра

Содержимое, созданное искусственным интеллектом, может быть неверным.**

1. В чем разница полученных в п.15 и п.16 результатов?\_\_\_\_\_\_\_\_\_\_\_

Есть два основных отличия:

1. Формат вывода (визуальное представление):

В п.15 вывод показан в табличном (горизонтальном) формате — строки и столбцы. Это стандартный вывод psql.

В п.16 вывод показан в расширенном (вертикальном) формате: каждая колонка отображается как ключ | значение для каждой строки. Это даёт удобное чтение, когда колонок много или значения длинные.

Команда/режим, отвечающий за это — expanded display. В psql можно переключать его командой \x (toggle) или применять к конкретному запросу добавлением \gx в конце запроса (что мы и сделали). \gx — это сочетание <запрос> \g (выполнить) с опцией x (expanded).

1. Семантики совпадения имени таблицы / чувствительность к регистру — практическая причина, почему я увидел «HR.Employees», хотя в общем списке была и public.employees:

PostgreSQL нормализует не в кавычках идентификаторы в нижний регистр. То есть имя, созданное как CREATE TABLE employees или CREATE TABLE EMPLOYEES (без кавычек), в системных представлениях хранится как employees (в нижнем регистре).

Если таблица создавалась с кавычками и смешанным/заглавным регистром, например CREATE TABLE "Employees" (...), то в системных представлениях имя будет храниться точно как Employees (с большой буквой E). Тогда сравнение WHERE tablename = 'Employees' найдёт именно такую таблицу.

В моем выводе:

* + 1. В общем LIMIT 7 я видел public | employees (маленькими буквами) — это одна таблица employees в схеме public.
    2. Запросы в п.15/16 вернули HR | Employees — это отдельная таблица, созданная/именно именованная с заглавной буквы (вероятно была создана с кавычками "Employees"), поэтому её tablename — 'Employees' и сравнение с 'Employees' сработало именно для неё.

Итого: pg\_tables может содержать обе записи (employees и Employees), они считаются разными строками, потому что строковые значения в tablename — точные.

1. По очереди запишите и запустите на выполнение следующие команды:

**SELECT version();**

**SELECT current\_database();**

**SELECT current\_user;**

**SELECT session\_user;**

1. Запишите результаты команд и объясните, что они возвращают.

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Содержимое, созданное искусственным интеллектом, может быть неверным.

SELECT version();

PostgreSQL 12.4, compiled by Visual C++ build 1914, 64-bit

1. Эта команда возвращает полную информацию о версии сервера PostgreSQL, к которому я подключен.
2. В выводе указана:

версия PostgreSQL (12.4),

компилятор, который использовался для сборки (Visual C++ build 1914),

архитектура сервера (64-bit).

1. Полезно для проверки совместимости и отладки, чтобы знать, какие функции и типы данных поддерживаются в данной версии.

SELECT current\_database();

dbSQL

1. Показывает имя базы данных, к которой я в данный момент подключен.
2. В моем случае это база dbSQL.
3. Эта информация полезна, когда работаем с несколькими базами на одном сервере и хотим убедиться, что выполняем запросы в нужной базе.

SELECT current\_user;

pguser

1. Возвращает имя пользователя, под которым выполняются SQL-запросы.
2. В PostgreSQL current\_user отражает права доступа текущей сессии.
3. В моем случае я подключился как pguser, поэтому current\_user = pguser.
4. Это важно для понимания, какие операции разрешены в текущей сессии, поскольку права пользователя влияют на доступ к таблицам и схемам.

4. SELECT session\_user;

pguser

1. Возвращает имя пользователя, который инициировал сессию подключения.
2. В моем случае это также pguser.
3. Отличие от current\_user заключается в том, что current\_user может изменяться внутри сессии через команду SET ROLE, тогда как session\_user остаётся постоянным и отражает изначального пользователя подключения.

Пример различия:

SET ROLE postgres;

SELECT current\_user; -- вернёт 'postgres'

SELECT session\_user; -- всё ещё 'pguser'

Это позволяет различать фактического пользователя, под которым выполняются запросы, и изначального владельца сессии.

1. Изучите команду **\pset**. Что она показывает? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Команда \pset (print settings) показывает текущие настройки формата вывода данных в psql.

Здесь можно увидеть:

* border — стиль рамки таблицы;
* columns — количество колонок в ширине терминала;
* expanded — режим расширенного (вертикального) вывода (\x);
* fieldsep — разделитель столбцов;
* format — формат таблицы (aligned, unaligned, csv и др.);
* pager — использование постраничного вывода;
* и другие настройки отображения.

С помощью этих параметров можно настроить, как psql показывает результаты запросов.



1. Выполните команду **\set**. Что она показывает?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Команда \set без аргументов показывает все переменные и их значения в текущей сессии psql.

Сюда входят:

* системные переменные psql (AUTOCOMMIT, ECHO, PROMPT1 и др.);
* информация о подключении (DBNAME, HOST, PORT, USER);
* версия сервера;
* пользовательские переменные (если вы создавали ранее).

Эти переменные можно использовать для настройки поведения psql или для передачи значений в SQL-запросы.

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Содержимое, созданное искусственным интеллектом, может быть неверным.

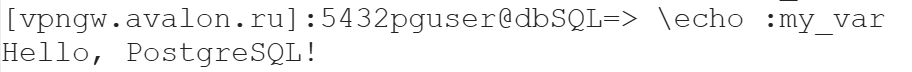
1. Создайте свою переменную по аналогии:
2. **\set *название\_переменной* *содержание\_переменной***

\set my\_var 'Hello, PostgreSQL!'



1. Выполните команду **\echo :*название\_перменной***
2. Что получилось?

\echo :my\_var



Команда \echo выводит текст или значение переменной в консоль.

Синтаксис :my\_var говорит psql подставить значение переменной my\_var.

1. Выполните команду **select :*название\_переменной*;**
2. Каков результат?

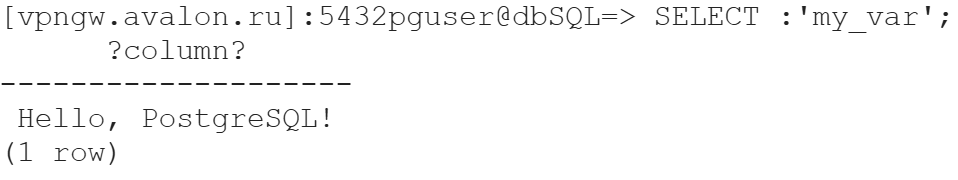
SELECT :my\_var;

Изображение выглядит как текст, Шрифт, чек, белый

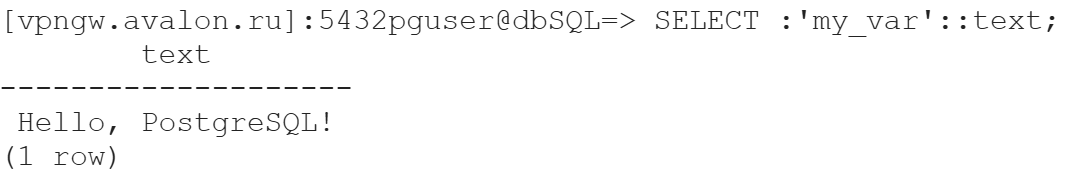
Содержимое, созданное искусственным интеллектом, может быть неверным.

Подставляет текст как Hello, PostgreSQL! без кавычек, и SQL пытается интерпретировать Hello и PostgreSQL! как имена колонок — отсюда ошибка column "hello" does not exist.

SELECT :'my\_var';



SELECT :'my\_var'::text;

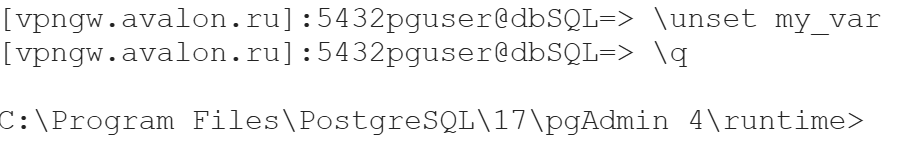


Чтобы точно подставить текст как строку

1. Сбросьте значение переменной с помощью **\unset *название\_переменной****.*

\unset my\_var

1. Закройте соединение с сервером



1. Закройте окно консоли