

??????? ? Postgres SQL

- [Переключение текущей сессии в нужную схему.](#)

???????????? ???? ???? ?
? ?????? ????.

Допустим вам нужно сделать схему **main** текущей.

Это достигается выполнением следующей команды:

```
SET search_path TO main, public;
```

“ **Как это работает:** Любые команды `CREATE TABLE qrtz_...` без явного указания схемы теперь будут создавать таблицы строго внутри `main`.

???????? ???? ?????

Чтобы убедиться, что контекст изменился на `main`, выполните:

```
SHOW search_path;  
SELECT current_schema();
```

В ответе должно быть `main`.

Указание двух схем определяет **порядок поиска объектов и резервный контекст**.

Когда вы пишете `SET search_path TO main, public;`, вы задаете списку схем приоритет слева направо:

1. При создании объектов (CREATE TABLE ...)

PostgreSQL всегда создает новый объект в **первой** схеме списка (в данном случае `main`), если вы не указали схему явно (например, `CREATE TABLE public.my table`).

2. При поиске и чтении данных (SELECT * FROM my_table)

PostgreSQL ищет таблицу по очереди:

- Сначала проверяет, есть ли `my_table` в схеме `main`.
- Если не находит — ищет её в схеме `public`.

- Если её нет и там — выдает ошибку `relation "my_table" does not exist`.

3. Доступ к системным функциям и расширениям

В схеме `public` часто располагаются стандартные расширения (extensions), служебные функции или общие справочники базы данных. Указав `public` вторым номером, вы гарантируете, что PostgreSQL сможет найти эти системные функции, даже если вы работаете в `main`.

??? ?????, ????? ????????? ??????? `main`?

Если написать `SET search_path TO main;`:

- Все таблицы создадутся в `main` (как и нужно).
- Но если вашей программе или функции потребуется объект или расширение из `public`, PostgreSQL его не увидит и выдаст ошибку.

Поэтому запись `SET search_path TO main, public;` является стандартом: **новые объекты создаются в `main`, но старые и системные объекты из `public` остаются доступны для чтения.**