

## Практика №5.1

### Администрирование в СУБД PostgreSQL. Табличные пространства

<b>Дисциплина</b>	Базы данных для промышленных задач
<b>Институт</b>	Перспективных технологий и промышленного программирования
<b>Кафедра</b>	Промышленного программирования
<b>Вид учебного материала</b>	Практика
<b>Преподаватель</b>	Евдошенко Олег Игоревич
<b>Семестр</b>	1 семестр, 2025-2026

### Часть 1. Содержимое табличного пространства

1. Создайте директорию test
2. Создайте табличное пространство: *CREATE TABLESPACE u01tbs LOCATION 'd:\test';*

3. Посмотрите содержимое директории табличного пространства

Была создана поддиректория с названием PG\_16\_202307071. В имени поддиректории присутствует номер основной версии СУБД. Такие директории создаются и удаляются автоматически, чтобы упростить обновление программного обеспечения на новую основную версию.






4. Создайте в табличном пространстве таблицу:

*CREATE TABLE tb (id bigserial, t text) TABLESPACE u01tbs;*

5. Заполните таблицу данными:

*INSERT INTO tb(t) SELECT encode((floor(random()\*1000)::numeric ^ 100)::numeric)::text::bytea, 'base64') from generate\_series(1,5000000);*

6. Посмотрим, какие файлы появились

 83701	01.07.2025 18:07	Файл	1 048 576 КБ
 83701.1	01.07.2025 18:08	Файл "1"	781 600 КБ
 83701_fsm	01.07.2025 18:08	Файл	528 КБ
 83705	01.07.2025 18:07	Файл	0 КБ
 83707	01.07.2025 18:07	Файл	8 КБ

Файл с суффиксом ".1". Это второй файл основного слоя (main fork).

7. Вставьте ещё миллион строк

Имя	Дата изменения	Тип	Размер
83701	01.07.2025 18:09	Файл	1 048 576 КБ
83701.1	01.07.2025 18:09	Файл "1"	1 048 576 КБ
83701.2	01.07.2025 18:09	Файл "2"	333 368 КБ
83701_fsm	01.07.2025 18:09	Файл	632 КБ
83705	01.07.2025 18:07	Файл	0 КБ
83707	01.07.2025 18:07	Файл	8 КБ

Добавился файл с суффиксом ".2". Это третий файл основного слоя.

8. Выполните удаление табличного пространства:

`drop tablespace u01tbs;` Сделайте выводы.

9. Посмотрите утилитой `oid2name` информацию о таблице:

`oid2name -t tb`

Это полезно, если вы хотите найти названия файлов основного слоя таблицы.

10. Посмотрите, что выдаёт утилита про оставшиеся файлы:

`oid2name -f 83705`

```
From database "postgres":
  Filenode      Table Name
-----
      83705    pg_toast_83701
```

Это файлы TOAST таблицы и TOAST-индекса. Для таблицы (обычного типа heap) может быть создана одна TOAST таблица и один индекс на эту TOAST таблицу.

## Часть 2. Файл объекта «последовательность»

1. Посмотрите определение таблицы: `\d tb`

2. Посмотрите определение последовательности: `\ds+`

3. Посмотрите характеристики последовательности как «объекта»:

`select * from pg_class where relname='tb_id_seq' \gx`

```
oid          | 83699
relname       | tb_id_seq
relnamespace  | 2200
reltype       | 83700
reloftype     | 0
relowner      | 10
relam         | 0
relfilenode   | 83699
```

Мы получили `oid`, номер файла, `oid` табличного пространства (ноль означает табличное пространство по умолчанию для базы данных). Также видно, что последовательность физически представляет собой одну запись (`reltuples`) в одном блоке (`relpages`).

4. Посмотрите путь к файлу последовательности:

```
SELECT pg_relation_filepath(relfilenode);
```

Файл последовательности был создан и располагается в табличном пространстве pg\_default, которое является табличным пространством по умолчанию для базы данных postgres:

```
select dattablespace, datname from pg_database;
```

```
select oid, spcname from pg_tablespace;
```

### Часть 3. Перемещение таблицы в другое табличное пространство

1. Переместим таблицу t в табличное пространство pg\_default.

2. В окне psql в целях оценки сколько журнальных данных сгенерируется посмотрим текущий LSN:

```
SELECT pg_current_wal_lsn();
```

```
pg_current_wal_lsn
-----
1/805AE0C8
(1 строка)
```

3. В окне psql дайте команду перемещения. Воспользуйтесь, например, синтаксисом перемещения всех таблиц:

```
alter table ALL IN TABLESPACE u01tbs SET TABLESPACE pg_default;
```

```
alter table tb SET TABLESPACE pg_default;
```

4. Посмотрите текущий LSN

```
pg_current_wal_lsn
-----
2/19317E68
(1 строка)
```

5. Посчитайте какой объем данных прошел через журналы:

```
select pg_size_pretty('2/19317E68'::pg_lsn - '1/805AE0C8'::pg_lsn);
```

6. Посмотрите размер таблицы:

```
select pg_size_pretty(pg_total_relation_size('tb'));
```

Через журнал кластера прошел весь объем перемещаемых данных.