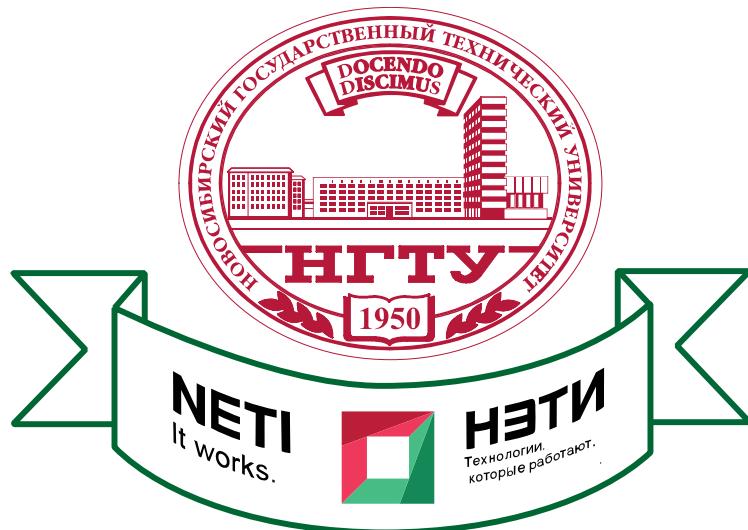


Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования

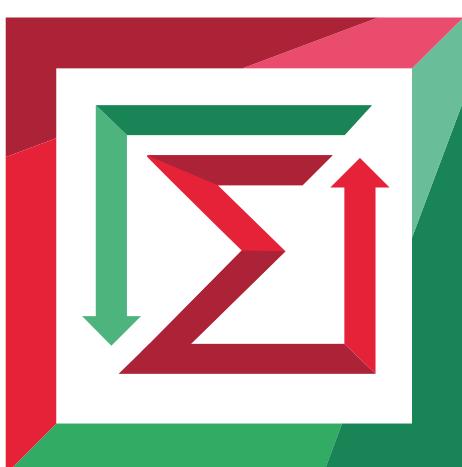
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Кафедра теоретической и прикладной информатики

Лабораторная работа № 2
по дисциплине «Администрирование информационных систем»

Хранение данных



Факультет: ПМИ
Группа: ПМИ-02
Бригада: 8
Студенты: Сидоров Даниил,
Дюков Богдан
Преподаватель: Аврунев О.Е.

Новосибирск

2026

Цель работы

Ознакомиться с принципами организации логической структуры и физического хранения объектов БД в PostgreSQL.

Исходные данные

Номер задания	№ Бригады	Таблица
2	3,8	flights

Номер задания	№ Бригады	Fillfactor, %
6	7,8	45

Ход работы

1. Для базы данных demo получили:

- список схем:

```
[dba@centos-stream-8 ~]$ psql -d demo
psql (15.5)
Type "help" for help.

demo=# \dn
          List of schemas
   Name    |      Owner
-----+-----
 bookings | dba
 public   | pg_database_owner
(2 rows)

demo=#
```

- список таблиц в схеме bookings:

```
demo=# select table_name from information_schema.tables
where table_schema = 'bookings';
   table_name
-----
 bookings
 flights_v
 ticket_flights
 boarding_passes
 aircrafts
 flights
 airports
 seats
 tickets
(9 rows)

demo=#
```

- СПИСОК ИНДЕКСОВ:

```
demo=# select indexname from pg_indexes where schemaname
= 'bookings';
          indexname
-----
aircrafts_pkey
airports_pkey
boarding_passes_flight_id_boarding_no_key
boarding_passes_flight_id_seat_no_key
boarding_passes_pkey
bookings_pkey
flights_flight_no_scheduled_departure_key
flights_pkey
seats_pkey
ticket_flights_pkey
tickets_pkey
(11 rows)

demo=#

```

2. Для таблицы flights из схемы bookings получили:

- расположение файла данных:

```
demo=# select pg_relation_filepath('bookings.flights');
 pg_relation_filepath
-----
base/16389/16409
(1 row)

```

(содержимое таблицы bookings.flights находится в файле с идентификатором 16409)

- размер таблицы в страницах:

```
demo=# select relpages from pg_class where relname = 'flights'
AND relkind = 'r';
 relpages
-----
      392
(1 row)

```

(т.к. в pg_class описываются не только таблицы, но и другие объекты, например индексы, то целесообразно в запросе указать тип объекта 'r', который соответствует обычным таблицам)

- размер соответствующего файла данных средствами операционной системы (получаем размер в килобайтах):

```
[dba@centos-stream-8 ~]$ sudo du -k /var/lib/pgpro/std-15/
data/base/16389/16409
3136    /var/lib/pgpro/std-15/data/base/16389/16409
[dba@centos-stream-8 ~]$
```

3. Сравнили размер файла и объем таблицы в страницах (размер одной страницы равен 8 килобайтам).

Для этого вычислили общий размер страниц в килобайтах: $392 * 8 \text{ КБ} = 3136 \text{ КБ}$.
Это значение совпадает с размером соответствующего файла.

4. Создали табличное пространство в предварительно созданном в домашнем каталоге пользователя dba каталоге table (даем пользователю postgres соответствующие права для возможности работы службы postgres):

```
[dba@centos-stream-8 ~]$ sudo mkdir table  
[dba@centos-stream-8 ~]$ sudo chown postgres:postgres ./table
```

```
[dba@centos-stream-8 ~]$ ls -la  
total 16  
drwxrwxrwx. 4 dba      dba      112 Feb 15 16:36 .  
drwxr-xr-x. 3 root     root     17 Feb 13 03:28 ..  
-rw-----. 1 dba      dba      179 Feb 13 05:49 .bash_history  
-rw-r--r--. 1 dba      dba      18 Oct  1 2021 .bash_logout  
-rw-r--r--. 1 dba      dba      141 Oct  1 2021 .bash_profile  
-rw-r--r--. 1 dba      dba      376 Oct  1 2021 .bashrc  
drwxrwxr-x. 4 dba      dba      76 Feb 15 16:10 .pgadmin  
drwxr-xr-x. 2 postgres postgres 6 Feb 15 16:36 table
```

```
demo=# create tablespace table_dba owner dba location '/home/dba/table';  
CREATE TABLESPACE
```

5. Создали отдельную схему:

```
demo=# create schema schema_dba authorization dba;  
CREATE SCHEMA
```

6. В созданной схеме:

- сделали копию выбранной таблицы, указав в качестве хранения созданное табличное пространство:

```
demo=# create table schema_dba.flights_copies(like bookings.flights)  
  with (fillfactor=45) tablespace table_dba;  
CREATE TABLE
```

(fillfactor = 45 означает, что в таблице flights_copies каждая страница данных заполняется только на 45%, а оставшиеся 55% свободного места используются для будущих обновлений => улучшение производительности обновлений)

- заполнили копию данными из исходной таблицы:

```
demo=# insert into schema_dba.flights_copies select * from bookings.  
flights;  
INSERT 0 33121
```

7. Получили для созданной таблицы данные о занятом пространстве и расположении ее файлов:

```
demo=# select pg_size.pretty(pg_total_relation_size('schema_dba.flights_copies'));
 pg_size.pretty
-----
 7064 kB
(1 row)

demo=# select pg_relation_filepath('schema_dba.flights_copies');
 pg_relation_filepath
-----
 pg_tblspc/16524/PG_15_202209071/16389/16535
(1 row)
```

Получили также количество возвращаемых страниц:

```
demo=# select relpages from pg_class where relname = 'flights_copies';
 relpages
-----
 879
(1 row)
```

(количество страниц больше, чем в исходной таблице, это логично, так как у исходной таблицы заполняемость страниц максимальна, vacuum analyze не требуется)

8. Отключили автоматическую очистку:

```
demo=# alter table schema_dba.flights_copies set (autovacuum_enabled=off);
ALTER TABLE
```

9. Удалили все данные из таблицы (транзакция подтверждается автоматически за счет включенного AUTOCOMMIT):

```
demo=# delete from schema_dba.flights_copies;
DELETE 33121
```

Выполнили повторно пункт 7:

```
demo=# select pg_size.pretty(pg_total_relation_size('schema_dba.flights_copies'));
 pg_size.pretty
-----
 7064 kB
(1 row)
```

10. Выполнили сборку мусора для таблицы:

```
demo=# vacuum schema_dba.flights_copies;
VACUUM
```

Выполнили повторно пункт 7:

```
demo=# select pg_size.pretty(pg_total_relation_size('schema_dba.flights_copies'));
 pg_size.pretty
-----
 16 kB
(1 row)
```

Автоматическая очистка в PostgreSQL работает в фоновом режиме и периодически освобождает пространство, занимаемое удаленными строками, делая его доступным для повторного использования. Однако, если автоматическая очистка отключена, операция DELETE не приводит к уменьшению размера таблицы, так как пространство, занимаемое удаленными строками, не возвращается операционной системе.