

Министерство образования и науки Российской Федерации  
Российский государственный университет нефти и газа  
(национальный исследовательский университет)  
имени И.М. Губкина

Факультет Автоматики и вычислительной техники  
Кафедра Автоматизированных систем управления

Отчёт по лабораторной работе №1  
«ОСНОВЫ РАБОТЫ С SQL»  
по дисциплине *Базы данных*

Выполнил:  
студент группы АС-23-04  
Ханевский Ярослав

Проверили:  
доцент кафедры АСУ, к.т.н. Волков Д. А.  
ст. преп. кафедры АСУ Мухина А. Г.

Москва, 2025 г.

Ход работы:

1. Установлена реляционная СУБД PostgreSQL, выбран способ установки с официального сайта (<https://postgrespro.ru/windows>):

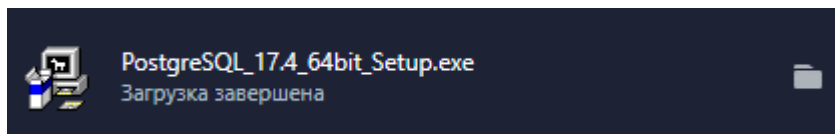


Рисунок 1. Загрузка PostgreSQL

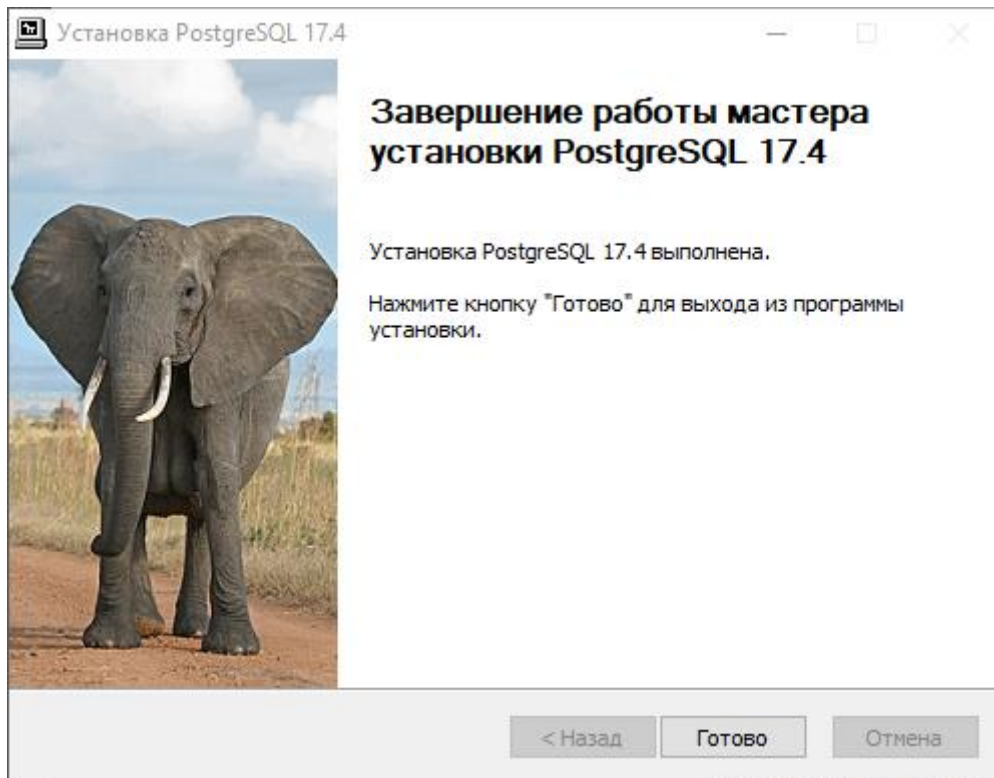


Рисунок 2. Установка PostgreSQL

2. Выполнение команд лабораторной работы:

- 1) Подключаемся к серверу СУБД:

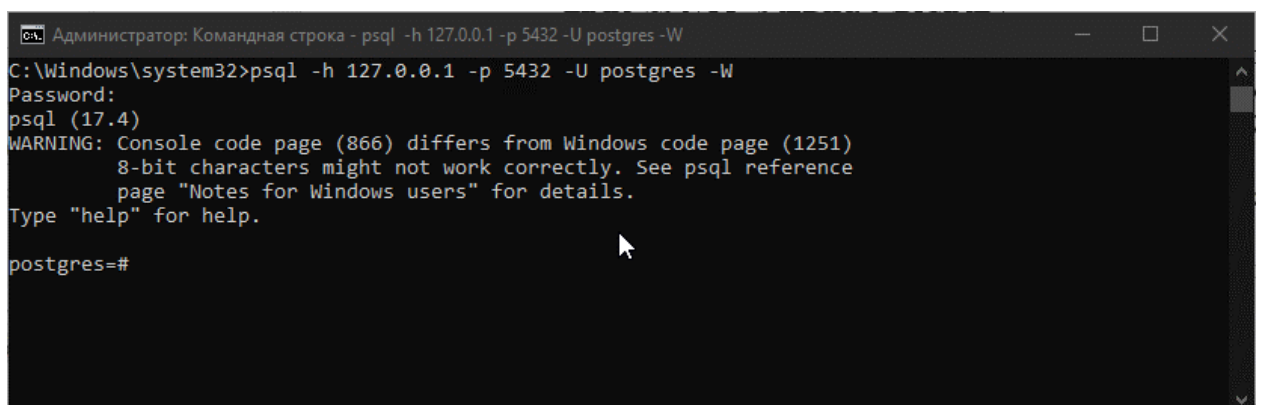


Рисунок 3. Подключение к серверу через командную строку

- 2) К символьным и строковым типам данных в PostgreSQL относятся:

CHAR(n) - строка фиксированной длины. Если строка короче, она дополняется пробелами.

VARCHAR(n) - строка переменной длины с ограничением по количеству символов (n).

TEXT - строка неограниченной длины.

NAME - внутренний тип для хранения имен объектов (например, таблиц, столбцов).

К числовым типам данных относятся:

SMALLINT (от -32,768 до 32,767) - 2-байтовое целое число.

INT или INTEGER (от -2,147,483,648 до 2,147,483,647) - 4-байтовое целое число.

BIGINT (от -9,223,372,036,854,775,808 до 9,223,372,036,854,775,807) - 8-байтовое целое число.

BOOLEAN - логическое значение (TRUE, FALSE или NULL).

К типам данных с плавающей точкой относятся:

REAL - 4-байтовое число с плавающей точкой (6 десятичных знаков).

DOUBLE PRECISION - 8-байтовое число с плавающей точкой (15 десятичных знаков).

DECIMAL(p, s) - точное число с фиксированной точностью (p — общее количество цифр, s — количество цифр после запятой).

К типам данных дата/время относятся:

DATE - дата (год, месяц, день).

TIME - время (часы, минуты, секунды).

TIMESTAMP - дата и время.

TIMESTAMPZ - дата и время с учетом временной зоны.

INTERVAL - интервал времени (например, "2 часа 30 минут").

3) Выводим список БД:

```
Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# \l

```

Name	Owner	Encoding	Locale	Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
postgres	postgres	UTF8	libc		Russian_Russia.1251	Russian_Russia.1251			
template0	postgres	UTF8	libc		Russian_Russia.1251	Russian_Russia.1251			=c/postgres + postgres=CtC/postgres
template1	postgres	UTF8	libc		Russian_Russia.1251	Russian_Russia.1251			=c/postgres + postgres=CtC/postgres

```
(3 rows)

postgres=#
postgres=# ^Z
```

Рисунок 4. Список баз данных сервера

#### 4) Выбираем БД «postgres»:

```
postgres=# \c postgres
Password:
You are now connected to database "postgres" as user "postgres".
```

Рисунок 5. Подключение к определенной базе данных

5) Далее создадим таблицу «chord», которая относится к сущности «аккорд». Таблица будет включать следующие столбцы: название, мажорный/минорный аккорд, лад, число зажатых струн, с барре/без барре. Выведем список таблиц для этой БД:

```
Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# \dt
Did not find any relations.
postgres=# CREATE TABLE chord(
postgres(# name VARCHAR(50) PRIMARY KEY NOT NULL,
postgres(# is_major BOOLEAN NOT NULL DEFAULT FALSE,
postgres(# lad SMALLINT NOT NULL DEFAULT 0,
postgres(# number_clamped SMALLINT,
postgres(# is_include_bare BOOLEAN NOT NULL DEFAULT FALSE);
CREATE TABLE
postgres=# \dt

```

Schema	Name	Type	Owner
public	chord	table	postgres

```
(1 row)

postgres=#
```

Рисунок 6. Создание таблицы и отображение таблиц в выбранной базе данных

6) Создадим ещё одну таблицу, после чего удалим её:

```
Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# CREATE TABLE song(
postgres(# name VARCHAR(50));
CREATE TABLE
postgres=# \dt

```

Schema	Name	Type	Owner
public	chord	table	postgres
public	song	table	postgres

```
(2 rows)

postgres=# DROP TABLE song;
DROP TABLE
postgres=# \dt

```

Schema	Name	Type	Owner
public	chord	table	postgres

```
(1 row)
```

Рисунок 7. Создание новой таблицы и её удаление

## 7) Выведем описание таблицы:

```
Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# \d chord;
               Table "public.chord"
  Column      |      Type      | Collation | Nullable | Default |
-----+-----+-----+-----+-----+
 name         | character varying(50) |           | not null |          |
 is_major     | boolean         |           | not null | false   |
 lad          | smallint        |           | not null | 0        |
 number_clamped | smallint        |           | not null |          |
 is_include_bare | boolean         |           | not null | false   |
Indexes:
    "chord_pkey" PRIMARY KEY, btree (name)

postgres=#
```

Рисунок 8. Применение команды описания таблицы

## 8) Добавим в таблицу данные (INSERT) и посмотрим содержимое таблицы(SELECT):

```
Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# INSERT INTO chord(name, is_major)
postgres=# VALUES ('C', TRUE);
INSERT 0 1
postgres=# \d chord
               Table "public.chord"
  Column      |      Type      | Collation | Nullable | Default |
-----+-----+-----+-----+-----+
 name         | character varying(50) |           | not null |          |
 is_major     | boolean         |           | not null | false   |
 lad          | smallint        |           | not null | 0        |
 number_clamped | smallint        |           | not null |          |
 is_include_bare | boolean         |           | not null | false   |
Indexes:
    "chord_pkey" PRIMARY KEY, btree (name)

postgres=# SELECT * FROM chord;
 name | is_major | lad | number_clamped | is_include_bare |
-----+-----+-----+-----+-----+
  C   | t       | 0   |               | f               |
(1 row)
```

Рисунок 9. Использование команды INSERT

## 9) Добавим ещё две строки, в одной из строк изменим данные(UPDATE), а другую удалим(DELETE):

```

Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# INSERT INTO chord
postgres=# VALUES ('Dm', DEFAULT, 1, 3, FALSE),
postgres=# ('Am', DEFAULT, 1, 3, DEFAULT);
INSERT 0 2
postgres=# SELECT * FROM chord;
 name | is_major | lad | number_clamped | is_include_bare 
-----+-----+-----+-----+-----
 C    | t        | 0   |                | f
 Dm   | f        | 1   |                | f
 Am   | f        | 1   |                | f
(3 rows)

postgres=# UPDATE chord
postgres=# SET lad=5, number_clamped=6, is_include_bare=TRUE
postgres=# WHERE name='Am';
UPDATE 1
postgres=# SELECT * FROM chord
postgres=# WHERE name='Am';
 name | is_major | lad | number_clamped | is_include_bare 
-----+-----+-----+-----+-----
 Am   | f        | 5   |                | t
(1 row)

postgres=# DELETE FROM chord
postgres=# WHERE name='Dm';
DELETE 1
postgres=# SELECT * FROM chord;
 name | is_major | lad | number_clamped | is_include_bare 
-----+-----+-----+-----+-----
 C    | t        | 0   |                | f
 Am   | f        | 5   |                | t
(2 rows)

```

Рисунок 10. Добавление 2-х строк, редактирование одной из них и удаление другой

## 10) Переименуем таблицу chord в khanewskiy:

```

Выбрать Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# ALTER TABLE chord
postgres=# RENAME khanewskiy ;
ERROR:  syntax error at or near "ALTER"
PŲPŲP P?P?P? 3: ALTER TABLE chord
      ^
postgres=# ALTER TABLE chord
postgres=# RENAME TO khanewskiy;
ALTER TABLE
postgres=# \dt
List of relations
Schema | Name       | Type  | Owner
-----+-----+-----+-----
 public | khanewskiy | table | postgres
(1 row)

postgres=#

```

Рисунок 11. Переименование таблицы

## 11) Изменим тип столбца «is\_major» без изменения имени:

```

Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# ALTER TABLE khanewskiy
postgres=# ALTER is_major
postgres=# TYPE VARCHAR(50);
ALTER TABLE
postgres=# \d khanewskiy
Table "public.khanewskiy"
  Column      | Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name         | character varying(50) |           | not null | 
 is_major     | character varying(50) |           | not null | false
 lad          | smallint      |           | not null | 0
 number_clamped | smallint      |           | not null | 
 is_include_bare | boolean       |           | not null | false
Indexes:
    "chord_pkey" PRIMARY KEY, btree (name)

postgres=#

```

Рисунок 12. Изменение типа столбца без изменения имени

## 12) Изменим тип столбца «number\_clamped» с изменением имени:

```
Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# ALTER TABLE khanewskiy RENAME COLUMN number_clamped TO age;
ALTER TABLE
postgres=# ALTER TABLE khanewskiy ALTER COLUMN age TYPE SMALLINT;
ALTER TABLE
postgres=# \d khanewskiy
Table "public.khanewskiy"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name         | character varying(50) |           | not null |
 is_major     | boolean         |           | not null | false
 lad          | smallint        |           | not null | 0
 age          | smallint        |           |          |
 is_include_bare | boolean         |           | not null | false
Indexes:
    "chord_pkey" PRIMARY KEY, btree (name)

postgres=#
```

Рисунок 3. Изменение типа столбца с изменением имени

### 13) Добавим новый столбец «is\_male»:

```
Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# ALTER TABLE khanewskiy
postgres=# ADD COLUMN is_male BOOLEAN DEFAULT TRUE;
ALTER TABLE
postgres=# \d khanewskiy
Table "public.khanewskiy"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name         | character varying(50) |           | not null |
 is_major     | boolean         |           | not null | false
 lad          | smallint        |           | not null | 0
 age          | smallint        |           |          |
 is_include_bare | boolean         |           | not null | false
 is_male      | boolean         |           |          | true
Indexes:
    "chord_pkey" PRIMARY KEY, btree (name)

postgres=#
```

Рисунок 4. Добавление нового столбца

### 14) Удалим столбцы, относящиеся к сущности «аккорд»:

```
Администратор: Командная строка - psql -h 127.0.0.1 -p 5432 -U postgres -W
postgres=# ALTER TABLE khanewskiy
postgres=# DROP COLUMN is_major,
postgres=# DROP COLUMN lad;
ALTER TABLE
postgres=# \d khanewskiy
Table "public.khanewskiy"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name         | character varying(50) |           | not null |
 age          | smallint        |           |          |
 is_include_bare | boolean         |           | not null | false
 is_male      | boolean         |           |          | true
Indexes:
    "chord_pkey" PRIMARY KEY, btree (name)

postgres=#
```

Создадим резервную копию БД:

```
C:\Windows\system32>pg_dump -U postgres -d postgres -f postgres_backup.sql
C:\Windows\system32>
```

Рисунок 5. Создание резервной копии через командную строку

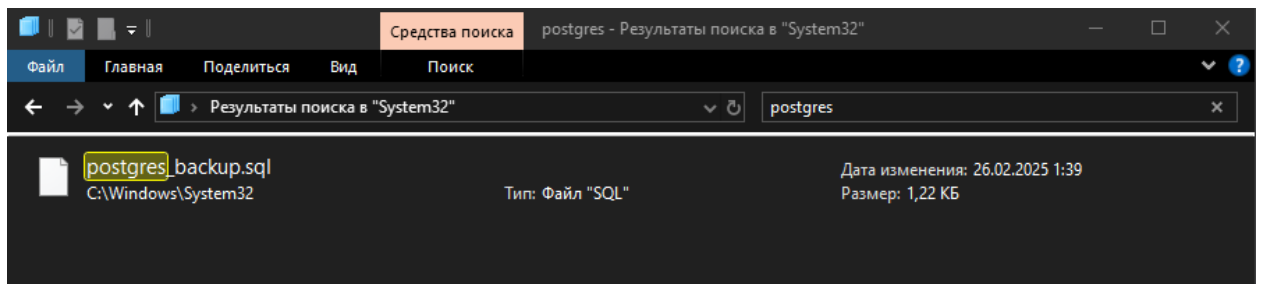


Рисунок 6. Проверка копии в файлах ОС

Удаляем таблицу для дальнейшего восстановления всей таблицы с помощью резервной копии:

```
postgres=# \dt
          List of relations
Schema |      Name      | Type | Owner
-----+-----+-----+-----
public | khanewskiy     | table | postgres
(1 row)

postgres=# DROP TABLE khanewskiy;
DROP TABLE
postgres=# \dt
Did not find any relations.
postgres=#
```

Рисунок 17. Удаление таблицы

Восстанавливаем БД из резервной копии:

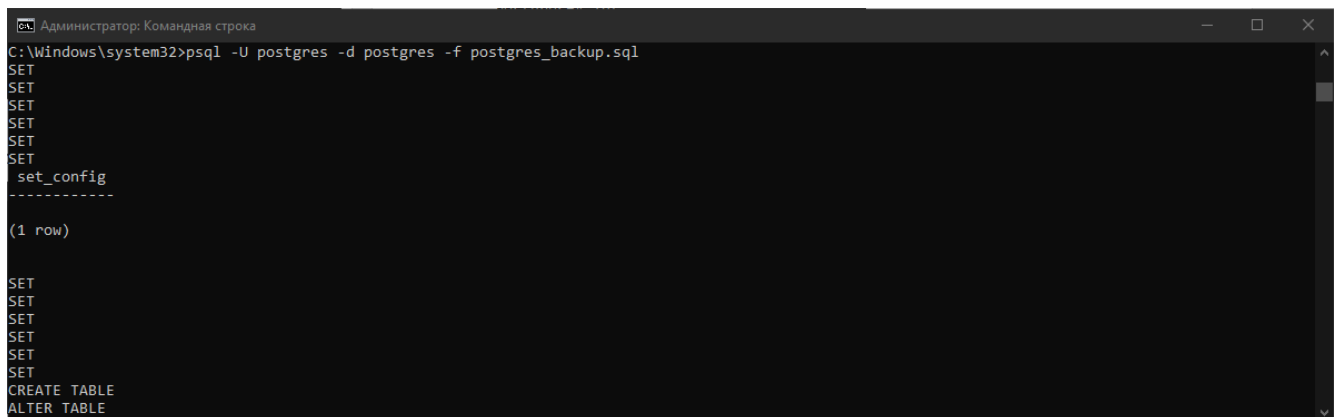


Рисунок 7. Загрузка резервной копии в конкретную базу данных



```
Администратор: Командная строка - psql -U postgres -h 127.0.0.1 -p 5432 -d postgres
C:\Windows\system32>psql -U postgres -h 127.0.0.1 -p 5432 -d postgres
psql (17.4)
WARNING: Console code page (866) differs from Windows code page (1251)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \dt
          List of relations
Schema | Name      | Type | Owner
-----+-----+-----+-----
public | khanewskiy | table | postgres
(1 row)

postgres=#
```

Рисунок 19.. Проверка восстановления

Создадим нового пользователя:

```
postgres=# CREATE USER teacher WITH PASSWORD 'test';
CREATE ROLE
postgres=#
```

Рисунок 20. Создание нового пользователя со своим паролем

```
postgres=> INSERT INTO khanewskiy(name)
postgres-> VALUES ('IVAN')
postgres-> ;
ERROR: permission denied for table khanewskiy
postgres=>
```

Рисунок 8. Проверка возможности редактирования таблицы

Наделим пользователя привилегиями для доступа к БД:

```
postgres=# GRANT SELECT ON TABLE khanewskiy TO teacher;
GRANT
postgres=#
```

Рисунок 22. Наделение пользователя правами доступа

Проверим, добавлен ли пользователь в БД:

```
postgres=> SELECT * FROM khanewskiy;
 name | age | is_include_bare | is_male
-----+-----+-----+-----
 C    |    | f                | t
 Am   |  6 | t                | t
(2 rows)

postgres=>
```

Рисунок 23. Проверка добавления пользователя к базе данных с помощью SELECT

Удалим права доступа для ранее созданного пользователя:

```

postgres=# REVOKE SELECT ON TABLE khanewskiy FROM teacher;
REVOKE
postgres=# quit

C:\Windows\system32>psql -U teacher -h 127.0.0.1 -p 5432 -d postgres
psql (17.4)
WARNING: Console code page (866) differs from Windows code page (1251)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=> SELECT * FROM khanewskiy;
ERROR:  permission denied for table khanewskiy
postgres=>

```

Рисунок 24. Удаление прав доступа для пользователя

Удалим пользователя:

```

postgres=# DROP USER teacher;
DROP ROLE
postgres=# quit

C:\Windows\system32>psql -U teacher -h 127.0.0.1 -p 5432 -d postgres
psql: error: PIP?P?PeP>C?C+PëC'C?C?C? Pe C?P4C?P?P4C?C? "127.0.0.1", PIP?C?C'C? 5432 P?P4 C?P?P?P>P?C?C?: FATAL:  role "teacher" does not exist
C:\Windows\system32>

```

Рисунок 25. Удаление пользователя