



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ» (НИЯУ МИФИ)
ОБНИНСКИЙ ИНСТИТУТ АТОМНОЙ ЭНЕРГЕТИКИ (ИАТЭ)



СУБД, SQL и OLAP-технологии

О.А.Мирзеабасов

Отделение ИКС ИАТЭ НИЯУ МИФИ

Осень 2017

Содержание

1 *PostgreSQL*

2 *SQL*

PostgreSQL

Настройка БД

Инициализация БД PostgreSQL в указанном каталоге:

```
initdb -D /var/lib/pgsql/data
```

При инициализации БД создается файловая структура, в том числе служебные файлы `postgresql.conf` и `pg_hba.conf`.
Запуск сервера после настройки (возможны также команды `restart` и `stop`):

```
pg_ctl start -D /var/lib/pgsql/data
```

Каталог можно не указывать, если он определен в системной переменной `PGDATA`:

```
export PGDATA=/var/lib/pgsql/data
```

PostgreSQL

Настройка БД

Изменение конфигурационного файла `postgresql.conf` для возможности удаленного доступа:

```
listen_addresses = '*'
```

Управление доступом пользователей (изменения в файл `pg_hba.conf`):

#	TYPE	DATABASE	USER	CIDR-ADDRESS	METHOD
	host	all	all	0.0.0.0/0	md5

Значение `md5` означает шифрование пароля.

Создание пользователя

Для нормальной работы с БД следует создать непривилегированного пользователя. От имени пользователя postgres (или другого пользователя с правами администратора) следует запустить оболочку psql и в ней выполнить команду:

```
CREATE USER имя CREATEDB LOGIN PASSWORD 'пароль';
```

Создание пользователя с административными правами:

```
CREATE USER имя SUPERUSER CREATEROLE  
CREATEDB LOGIN PASSWORD 'пароль';
```

Точка с запятой в конце команды обязательна!

PostgreSQL

psql

`psql` — командный интерпретатор для работы с сервером Postgres. При запуске можно указать хост (параметр `-h`), порт (параметр `-p`, по умолчанию 5432), имя базы данных, пользователя. Параметр `-W` указывает, что требуется запросить пароль.

```
psql -h host -p 5432 -d dbname -U user -W
```

Неинтерактивное использование:

```
psql -c "SELECT current_time;"
```

Выполнение команд, записанных в файле:

```
psql -f examples.sql
```

Создание БД

Для создания базы данных («табличного пространства») можно либо использовать интерактивное средство администрирования pgAdmin, либо в консоли выполнить команду

```
createdb -U пользователь -W имя_базы_данных
```

Здесь опция -W используется для запроса пароля указанного пользователя.

После этого можно запустить интерпретатор psql для соединения с созданной БД:

```
psql -U пользователь -W имя_базы_данных
```

Сохранение БД: *pg_dump*

Сохранение БД возможно с помощью специализированного клиентского приложения (pgAdmin), однако рекомендуемым способом является использование утилиты *pg_dump*, которая по умолчанию генерирует вывод в виде команд SQL.

В простейшем случае команда может выглядеть так:

```
pg_dump имя_базы_данных > имя_файла
```

Для указания пользователя и других параметров можно использовать опции *-U*, *-W*, *-h*.

Для восстановления БД можно запустить интерпретатор *psql*:

```
psql имя_базы_данных < имя_файла
```

Важно: указанную БД следует предварительно создать!

Полезные команды интерпретатора psql:

- `\?` — вывод справки по командам psql;
- `\h` — вывод справки по доступным командам SQL;
- `\quit` или `\q` — выход из интерпретатора.

По команде `\? psql` выводит список внутренних команд.

- `\i FILENAME` — выполнить команды из указанного файла;
- `\dt` — вывести список таблиц;
- `\du` — список ролей;
- `\password` — изменить пароль пользователя.

SQL

Некоторые типы данных PostgreSQL

```
bigint, int8  
bytea  
character(n), char(n)  
character varying(n), varchar(n)  
date  
double precision, float8  
integer, int, int4  
numeric(p,s), decimal(p,s)  
serial, serial4  
text  
timestamp
```

Таблицы

Жизненный цикл объектов БД управляется операторами CREATE, ALTER, DROP.

```
CREATE TABLE имя (  
    список определений столбцов,  
    список ограничений);
```

Определение столбца может включать, кроме имени и типа данных, значение по умолчанию (директива DEFAULT).

Ограничения целостности:

- CHECK
- NOT NULL
- UNIQUE
- PRIMARY KEY
- внешние ключи

Пример создания таблицы

```
CREATE TABLE faculty (  
    fcid integer not null primary key,  
    fname varchar(80)  
);
```

```
CREATE TABLE cathedra (  
    cathid integer not null primary key,  
    fcid integer  
    cathname varchar(80),  
    preps integer,  
    foreign key(fcid) references faculty  
);
```

Запрос *SELECT*

Простая форма запроса:

```
SELECT список_полей FROM таблица  
WHERE список_ограничений;
```

Боле полный вариант может содержать и другие конструкции:

```
SELECT список_выражений FROM список_таблиц  
WHERE список_ограничений  
GROUP BY список_выражений  
HAVING список_условий  
ORDER BY список_выражений  
LIMIT число;
```

Примеры *SELECT*

```
SELECT fcid, sum(preps)
FROM cathedra
GROUP BY fcid
ORDER BY 2 DESCENDING;
```