

# Программная инженерия. Разработка ПО (Python для продвинутых специалистов. Машинное обучение)

Модуль: Введение в Python для машинного обучения

Лекция 6: Работа с базой данных из Python

Дата: 10.04.2024

Q&A



# Содержание лекции

- Для чего вообще нужно уметь работать с базой данных?
- SQLite – удобная подручная БД
- PostgreSQL
- Практическая часть



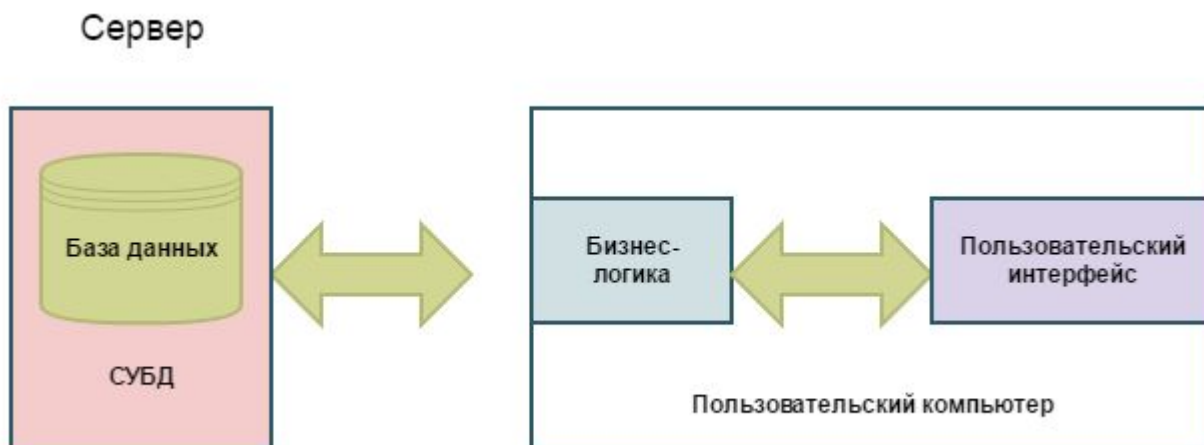
# Зачем нужны базы данных

Количество данных в мире растет огромными темпами.  
Поэтому их нужно: А – хранить, и Б – уметь извлекать из них информацию.

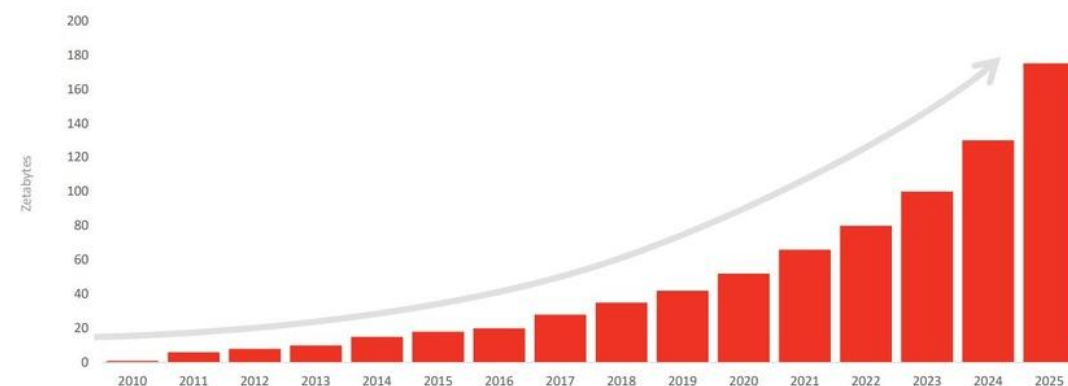
Для разных данных существуют разные виды баз:

- **Реляционные СУБД**
  - Oracle, PostgreSQL, MS SQL, MySQL, Teradata, Greenplum
- **Не реляционные (No SQL)**
  - Документно-ориентированные (Mongo, Elasticsearch)
  - Ключ-значение (Redis)
  - Графовые (Neo4j)
  - Колоночные (Clickhouse, HBase)
  - ...

В архитектуре любой ИС будет слой с данными



Annual Size of Global Digital Data Generated (ZB)



iab.

iAB Proprietary Research

Sources: <https://www.forbes.com/sites/tonmoughlin/2018/11/27/275-zettabytes-by-2025/#1407575450>; <https://www.networkworld.com/article/3323397/ibc-expect-275-zettabytes-of-data-worldwide-by-2025.html>

# Как мы будем работать с БД?

Для практики будем использовать SQLite или PostgreSQL.

SQLite будет работать локально без установки, входит в набор базовых пакетов Python.

Как можно получить базу PostgreSQL в свое распоряжение:

1. **Docker** - <https://habr.com/ru/articles/578744/>
2. **Установка локальной БД** - <https://www.postgresql.org/download/>
3. **Удаленная VM от РТ** - <https://195.19.105.128:7654>  
(подключение к postgres под одноименным логином/паролем)
4. **\* Аренда** базы в облаке (Yandex Cloud, Sber Cloud...)

Если кому то интересно потрогать Oracle - <https://dockerhosting.ru/blog/oracle-database-v-docker/>

*Для работы напрямую с Базой Данных удобнее всего использовать IDE, например DBeaver*



Передовые  
инженерные  
школы



МИНОБРНАУКИ  
РОССИИ



УНИВЕРСИТЕТ  
ИННОПОЛИС



онлайн  
университет

# SQLite – удобная подручная БД

SQLite - это компактная встраиваемая база данных, open-source. Она не использует парадигмы клиент-сервер, вместо этого SQLite представляет собой библиотеку, которая компонуется с программой и становится ее частью. Вся база данных хранится в единственном стандартном файле (или в памяти) на компьютере, где исполняется программа.

```
import sqlite3

conn = sqlite3.connect( "mydatabase.db" ) # файл
# conn = sqlite3.connect ( ":memory:" ) # RAM
cursor = conn.cursor()

# Создание таблицы
cursor.execute( "CREATE TABLE testtable ( id int, val text )" )

# Вставляем данные в таблицу
cursor.execute( "INSERT INTO testtable ( id, val ) VALUES ( 1, 'One' )" )
cursor.execute( "INSERT INTO testtable ( id, val ) VALUES ( 2, 'Two' )" )

# Сохраняем изменения
conn.commit()

# Закрываем курсор и соединение
cursor.close()
conn.close()
```



```
import sqlite3

conn = sqlite3.connect( "mydatabase.db" ) # файл
# conn = sqlite3.connect ( ":memory:" ) # RAM
cursor = conn.cursor()

# Чтение таблицы 1
sql = "SELECT * FROM testtable"
cursor.execute( sql )
print( cursor.fetchall() ) # или fetchone() если нужно построчно

# Чтение таблицы 2
for row in cursor.execute( sql ):
    print( row )

# Закрываем курсор и соединение
cursor.close()
conn.close()
```

## Преимущества:

- Работает без выделенного сервера
- Встроена в приложение
- Удобная консоль sqlite3

## Недостатки:

- ограничения по производительности
- нет полноценной многопользовательской поддержки -> нет ролевой модели
- высокая связанность с приложением





Передовые  
инженерные  
школы



МИНОБРНАУКИ  
РОССИИ



УНИВЕРСИТЕТ  
ИННОПОЛИС



онлайн  
университет

# PostgreSQL

# Запуск Postgres в Docker

Скачайте последнюю версию образа с базой PostgreSQL:

```
docker pull postgres # https://hub.docker.com/\_/postgres
```

Создайте контейнер с базой данных из образа и запустите его:

```
docker run --name postgres_cont -p 5432:5432 -e POSTGRES_PASSWORD=postgres -d postgres
```

Здесь:

- *--name postgres\_cont # имя контейнера*
- *-p 5432:5432 # маппинг внутреннего и внешнего портов*
- *-e POSTGRES\_PASSWORD=postgres # установка пароля в переменные окружения БД*
- *-d postgres # имя образа, из которого создать контейнер*

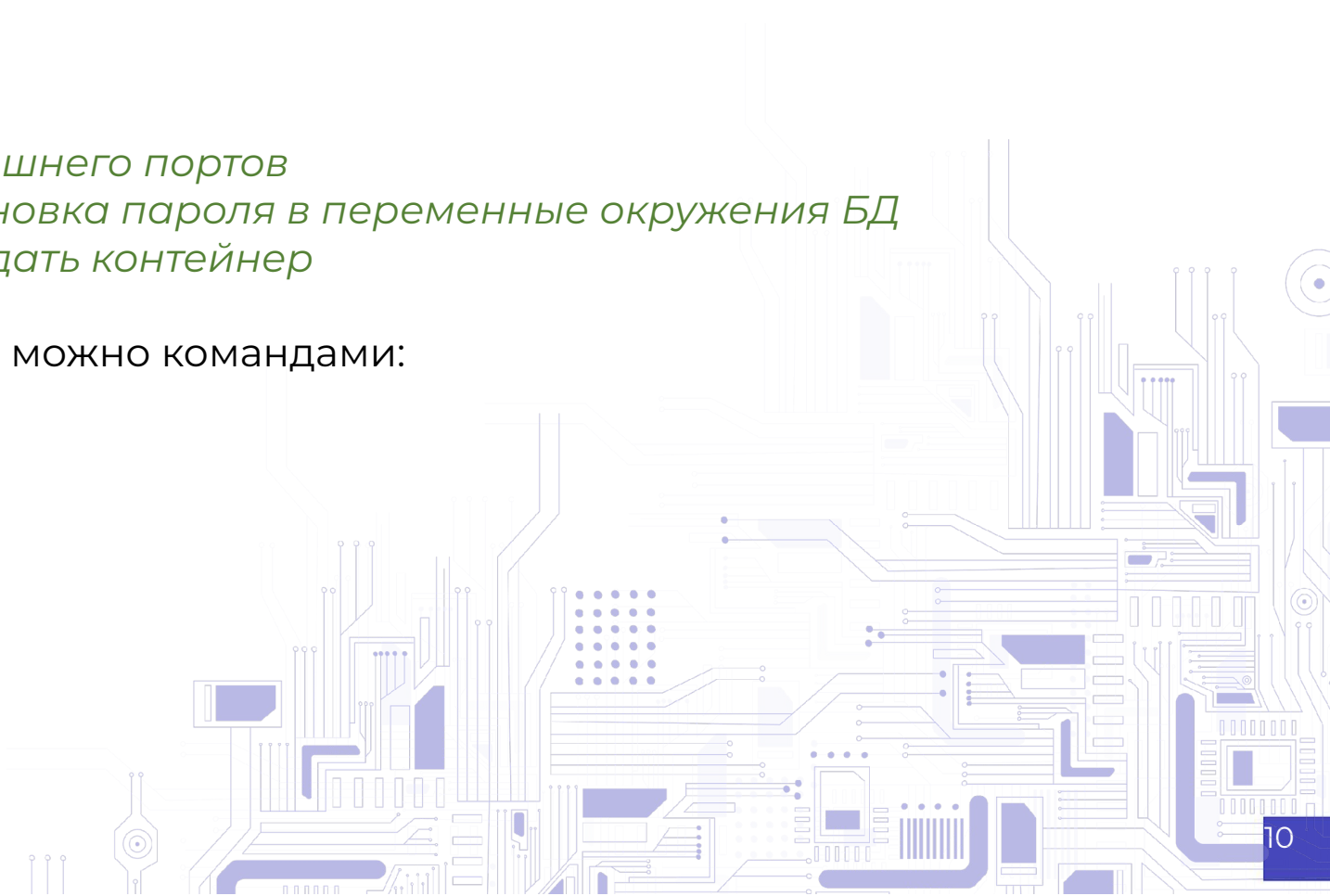
Остановить и повторно запустить контейнер можно командами:

```
docker stop postgres_cont
```

```
docker start postgres_cont
```

Подключиться к базе данных можно через:

- IDE, например Dbeaver
- psql



PostgreSQL - свободная объектно-реляционная система управления базами данных (СУБД).

Реализует клиент-серверную архитектуру.

Используется в основе таких СУБД как: Postgres PRO, Pangolin, Greenplum и тд.

```
import psycopg2

conn = psycopg2.connect(
    database = "postgres",
    host     = "localhost",
    user     = "postgres",
    password = "postgres",
    port     = "5432")

conn.autocommit = False # Выключение автокоммита
cursor = conn.cursor()

# Выполнение SQL кода в базе данных без возврата результата
cursor.execute( "INSERT INTO deaise.testtable( id, val ) VALUES ( 1, 'ABC' )" )
conn.commit()

# Выполнение SQL кода в базе данных с возвратом результата
cursor.execute( "SELECT * FROM deaise.testtable" )
records = cursor.fetchall()
for row in records:
    print( row )

# Закрываем курсор и соединение
cursor.close()
conn.close()
```



Передовые  
инженерные  
школы



МИНОБРНАУКИ  
РОССИИ



УНИВЕРСИТЕТ  
ИННОПОЛИС



онлайн  
университет

# Спасибо за внимание

