

SQLAlchemy

10 класс



Flask SQLAlchemy

ORM (Object-Relational Mapping) — это технология, которая позволяет сопоставлять модели, типы которых несовместимы. Например: таблица базы данных и объект языка программирования.



Flask SQLAlchemy

Flask-SQLAlchemy is an extension for [Flask](#) that adds support for [SQLAlchemy](#) to your application. It simplifies using SQLAlchemy with Flask by setting up common objects and patterns for using those objects, such as a session tied to each web request, models, and engines.

Flask-SQLAlchemy does not change how SQLAlchemy works or is used. See the [SQLAlchemy documentation](#) to learn how to work with the ORM in depth. The documentation here will only cover setting up the extension, not how to use SQLAlchemy.

User Guide

- [Quick Start](#)
 - [Check the SQLAlchemy Documentation](#)
 - [Installation](#)
 - [Initialize the Extension](#)
 - [Configure the Extension](#)
 - [Define Models](#)
 - [Create the Tables](#)
 - [Query the Data](#)
 - [What to Remember](#)
- [Configuration](#)
 - [Configuration Keys](#)
 - [Connection URL Format](#)
 - [Default Driver Options](#)
 - [Engine Configuration Precedence](#)
 - [Timeouts](#)



Project Links

[Donate](#)
[PyPI Releases](#)
[Source Code](#)
[Issue Tracker](#)
[Website](#)
[Twitter](#)
[Chat](#)

Contents


[Flask-SQLAlchemy](#)
[User Guide](#)
[API Reference](#)
[Additional Information](#)

Quick search



Реляционные базы данных

Реляционные базы данных состоят из **таблиц**, которые в свою очередь состоят из **записей**. У каждой записи есть **поля**.

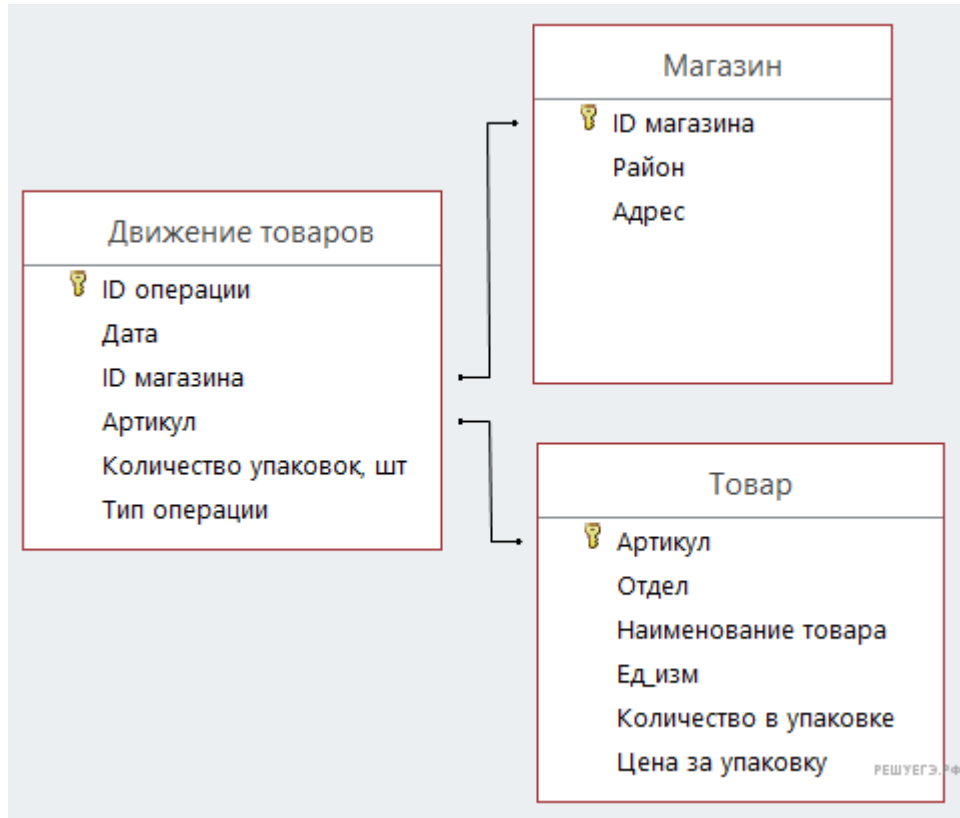
| Движение товаров | |
|---|-------------|
|  | ID операции |
| Дата | |
| ID магазина | |
| Артикул | |
| Количество упаковок, шт | |
| Тип операции | |

Первичный ключ (primary key) – уникальный идентификатор записи в таблице.

Некоторые свойства полей в sqlalchemy: **nullable** (может ли быть пустым), **unique** (должно ли быть уникальным), **primary_key** (является ли поле первичным ключом)



Реляционные базы данных



Таблицы могут быть связаны между собой различными отношениями:

- один к одному
- один ко многим
- многие ко многим



Подключение базы данных к проекту

Шаг 1. Импортируем модуль flask_sqlalchemy

```
app.py x
1  from flask import Flask
2  from flask_sqlalchemy import SQLAlchemy
3
4
5  app = Flask(__name__)
6
7
8  @app.route('/')
9  def hello_world(): # put application's code here
10     return 'Hello World!'
11
12
13  if __name__ == '__main__':
14     app.run()
15
```



Подключение базы данных к проекту

Шаг 2. Устанавливаем секретный ключ и подключаем базу данных

```
app.py x
1 from flask import Flask
2 from flask_sqlalchemy import SQLAlchemy
3
4
5 app = Flask(__name__)
6
7 app.secret_key = '1F7VKTpXpSB09P60skv9Kq$23QwD9FG44U'
8 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///hotel.db'
9 db = SQLAlchemy(app)
10
11
12 @app.route('/')
13 def hello_world(): # put application's code here
14     return 'Hello World!'
15
16
17 if __name__ == '__main__':
18     app.run()
19
```



Подключение базы данных к проекту

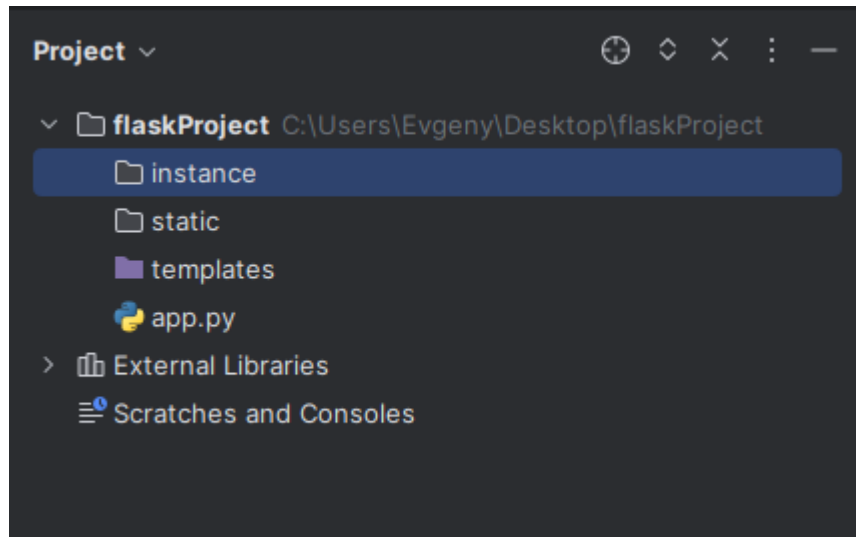
Шаг 3. Создаем класс таблицы в базе данных

```
12 class Pupil(db.Model):
13     __tableName__ = 'pupil'
14     id = db.Column(db.Integer, primary_key=True)
15     name = db.Column(db.String(20), nullable=False)
16     surname = db.Column(db.String(20), nullable=False, unique=True)
17     birth_date = db.Column(db.Date, nullable=False)
18     age = db.Column(db.Integer, nullable=False)
19
20     def __init__(self, name, surname, birth_date, age):
21         self.name = name
22         self.surname = surname
23         self.birth_date = birth_date
24         self.age = age
```



Подключение базы данных к проекту

- Если просто запустить проект, то база данных автоматически не будет создана!



Появится лишь папка **instance**, в которой в последствии и будет храниться файл базы данных.



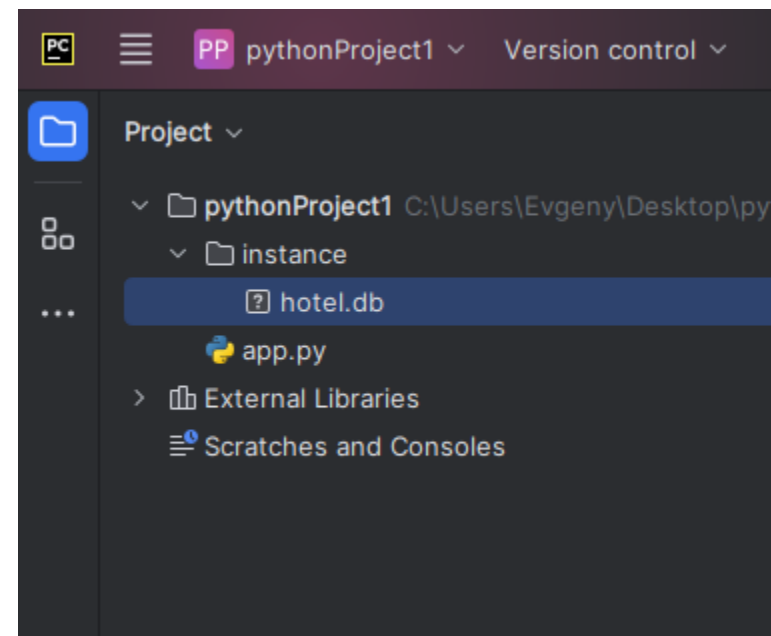
Подключение базы данных к проекту

Шаг 4. flask shell + db.create_all()

```
Terminal Local x + v
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены
Установите последнюю версию PowerShell для новых функций и улучшени


(venv) PS C:\Users\Evgeny\Desktop\pythonProject1> flask shell
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.
App: app
Instance: C:\Users\Evgeny\Desktop\pythonProject1\instance
>>> db.create_all()
>>>
```



pythonProject1 > app.py





DBeaver и ручное редактирование базы данных



**DBeaver Community**
Free Universal Database Tool

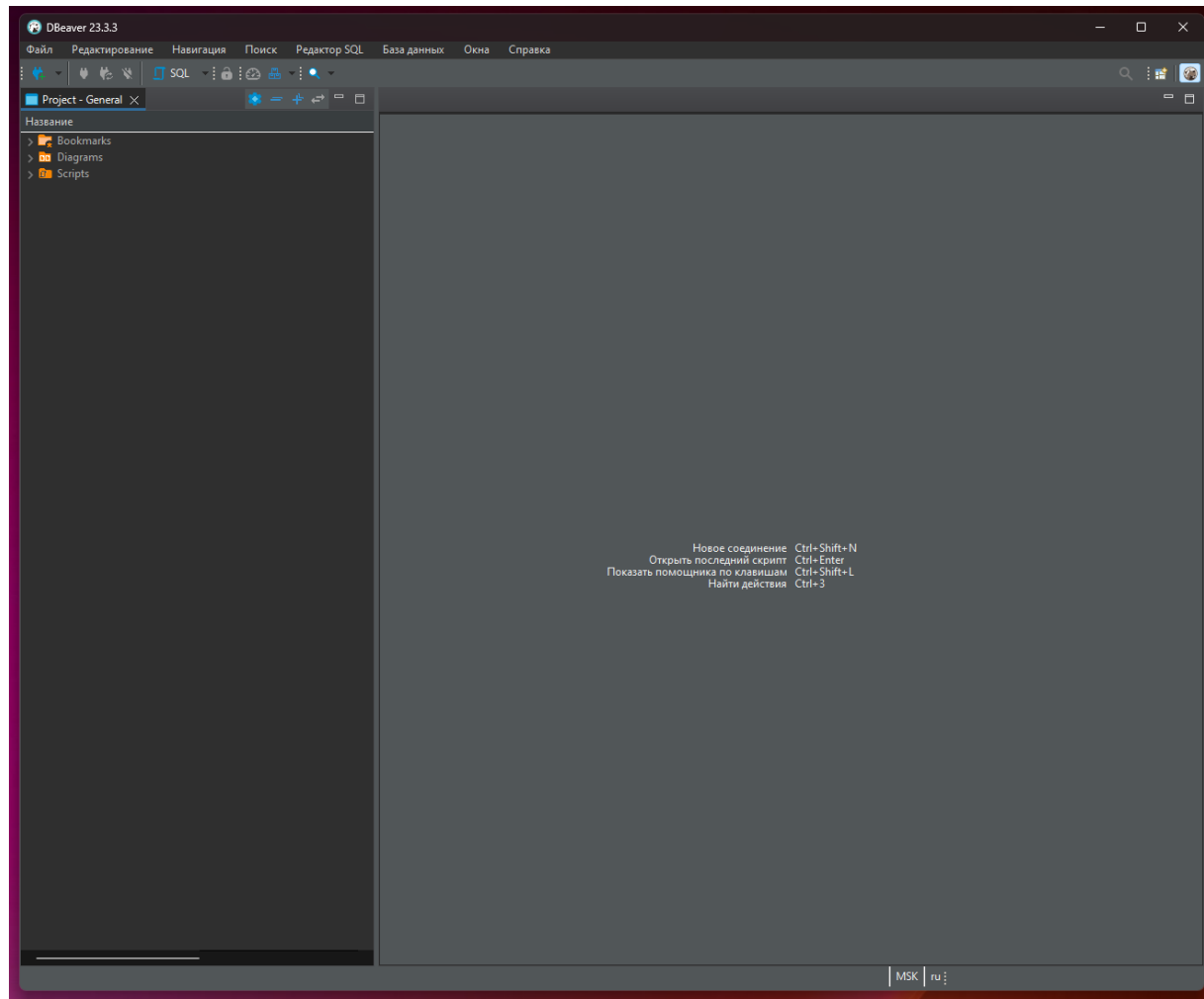
Star  Follow  @dbeaver_news

search here ...

Home About Download Documentation News Support DBeaver PRO CloudBeaver DBeaver Merch

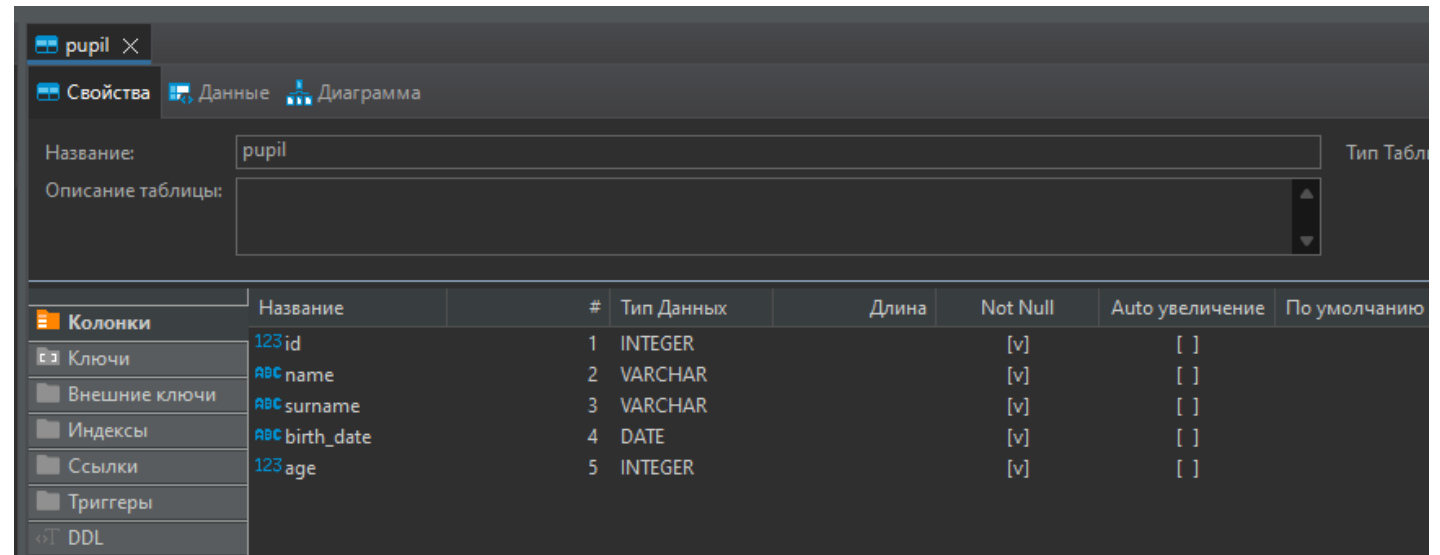
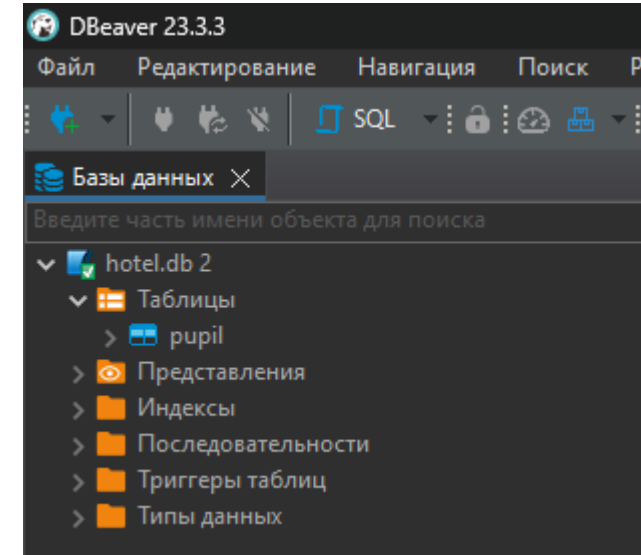
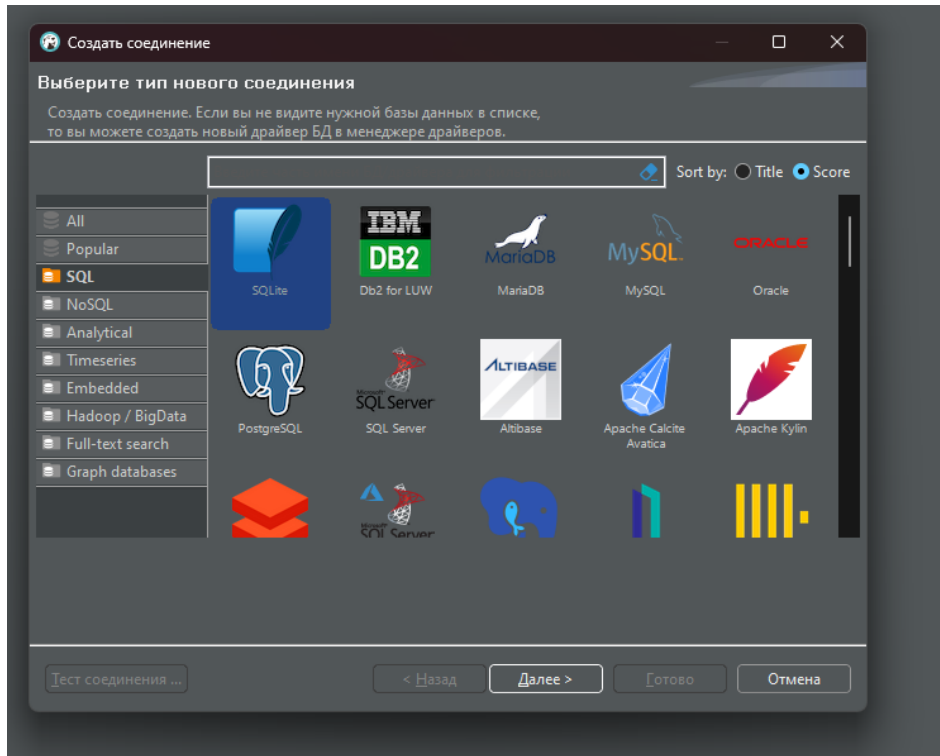
Download

| | |
|--|---|
| DBeaver Community 23.3.3 Released on January 22nd 2024 (Milestones). It is free and open source (license). Also you can get it from the GitHub mirror . | DBeaver PRO 23.3 Released on December 11th, 2023 PRO version website: dbeaver.com Trial version is available. |
|  Windows <ul style="list-style-type: none">Windows (installer)Windows (zip)Chocolatey (choco install dbeaver)Install from Microsoft Store  Mac OS X <ul style="list-style-type: none">MacOS for Intel (dmg)MacOS for Apple Silicon (dmg)Brew Cask (brew install --cask dbeaver-community) | DBeaver PRO features: <ul style="list-style-type: none">Advanced security: master password and strong credentials encryption to secure and easy database connection.Enterprise-level authentication methods: SAML, SSO, OKTA, and Kerberos. SSO authentication cloud services such as GCP, AWS, and Azure.Advanced SQL databases support: MySQL, PostgreSQL, SQL Server, Oracle and othersNoSQL databases support: MongoDB, Cassandra, Redis, CouchDB and othersCloud databases support: Redshift, Google BigQuery, Oracle Cloud and othersNative cloud support for Google Cloud, AWS, and Azure.Access all files in any region with the S3 browser to upload, store, share, and save files like in a regular file system |



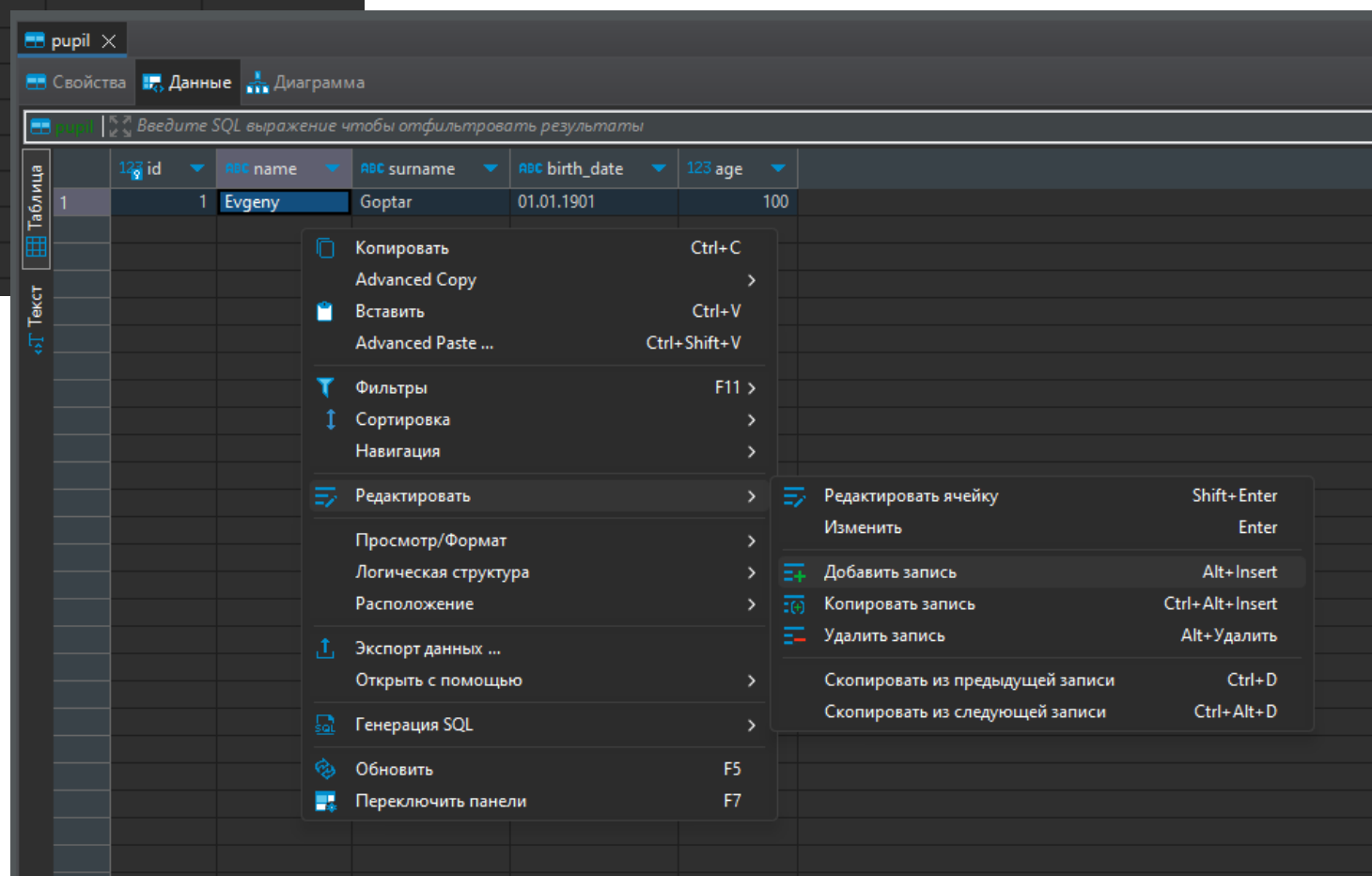
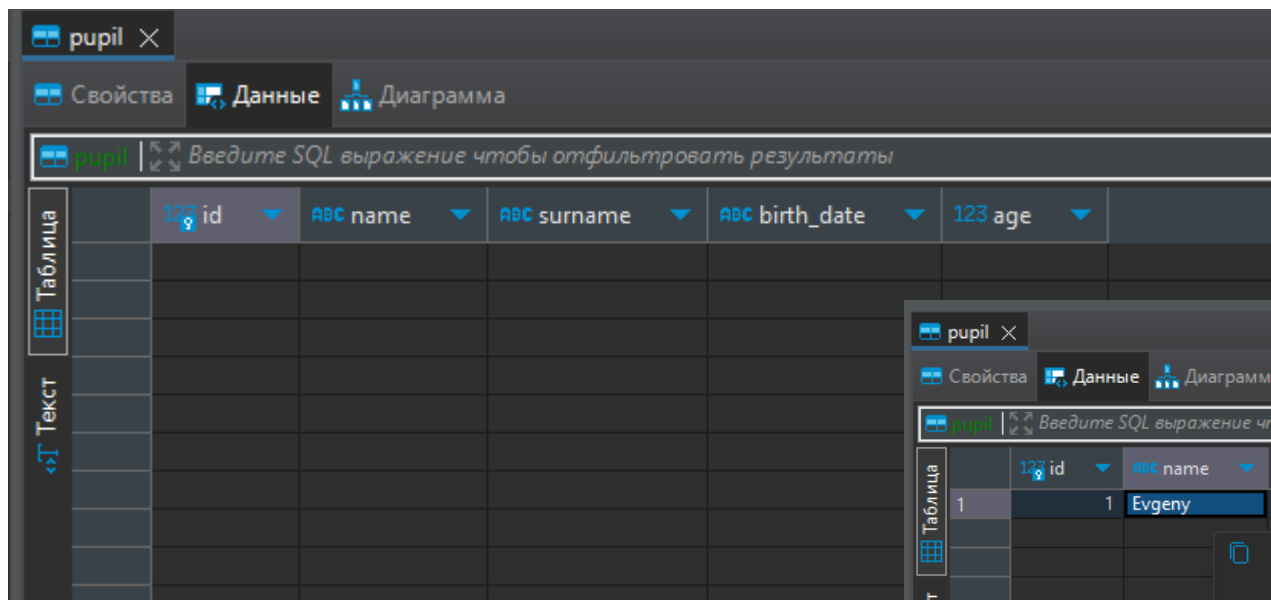


DBeaver и ручное редактирование базы данных





DBeaver и ручное редактирование базы данных



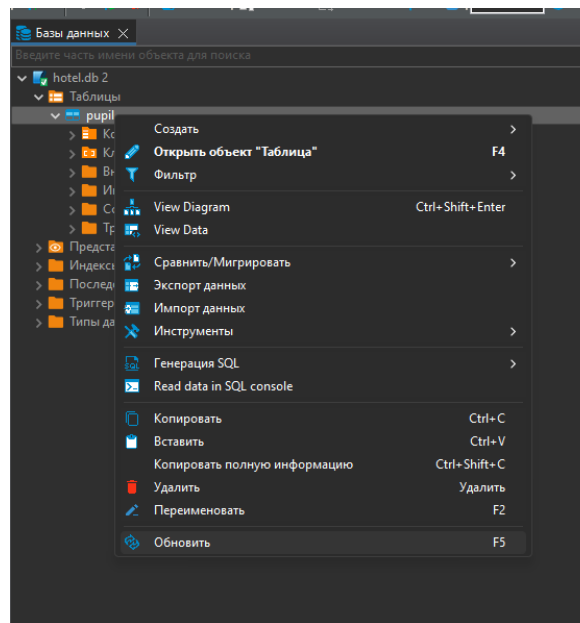


Получение записей и создание записей в базе данных из Flask

```
27 @app.route('/')
28 def hello_world():
29     pupil = Pupil(name="Ivan", surname="Ivanov", datetime.date(year=1999, month=5, day=5), age=40)
30     db.session.add(pupil)
31     db.session.flush()
32     db.session.commit()
33     return 'Запись добавлена в базу данных'
34
```



Запись добавлена в базу данных



| *pupil | | pupil | | | | |
|----------|--|--|--------|---------|------------|-----|
| Свойства | | Данные | | | | |
| Таблица | | Введите SQL выражение чтобы отфильтровать результаты | | | | |
| | | id | name | surname | birth_date | age |
| 1 | | 1 | Evgeny | Goptar | 01.01.1901 | 100 |
| 2 | | 2 | Ivan | Ivanov | 1999-05-05 | 40 |
| Текст | | | | | | |



Получение записей и создание записей в базе данных из Flask

```
@app.route("/users_list")
def users_list():
    s = []
    for pupil in Pupil.query.all():
        print(pupil.name)
        s.append(pupil.name)
    return str(s)
```

A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:5000/users_list`. Below the address bar, the response is displayed as `['Evgeny', 'Ivan']`.

```
C:\users\evgeny\venv\scripts\python.exe C:\users\evgeny\desktop\python
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [22/Jan/2024 23:09:13] "GET /users_list HTTP/1.1" 200 -
Evgeny
Ivan
```

Домашнее задание. Создайте три таблицы базы данных про магазины (связи между таблицами пока не реализовывайте). Добавьте через DBeaver по 3-4 записи в каждую таблицу. Создайте три url-адреса по каждому из которых можно получить строку со списком магазинов, со списком торговых операций и со списком товаров.

