СОДЕРЖАНИЕ

[Веб страница 2](#_Toc121309738)

[Код программы 4](#_Toc121309739)

# Веб страница

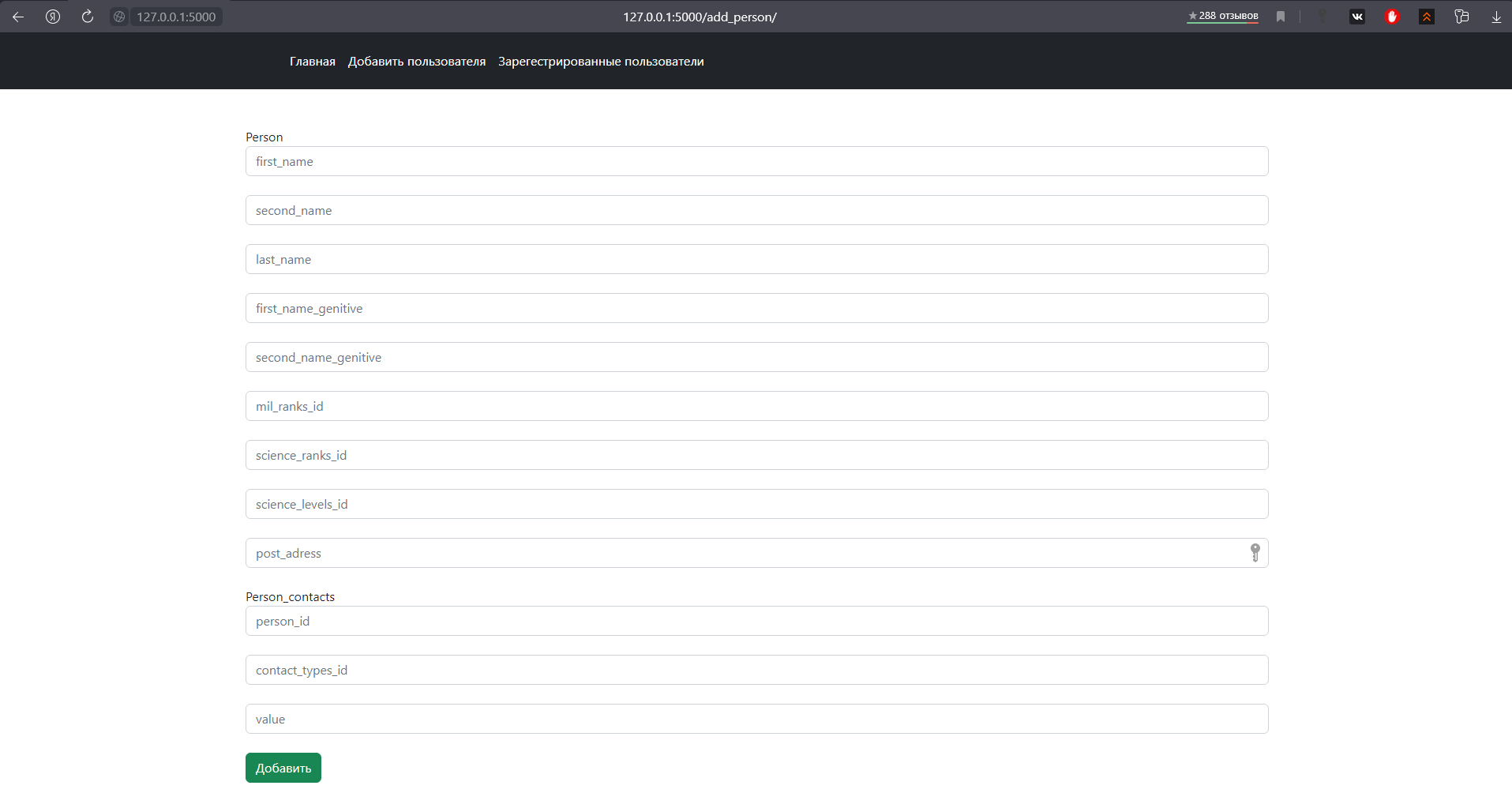


Рисунок 1 — добавить пользователя

На этой странице реализован функционал добавления пользователя c помощью СУБД postgresql в базу данных c названием example\_db. База данных состоит из двух таблиц person и person\_contacts.

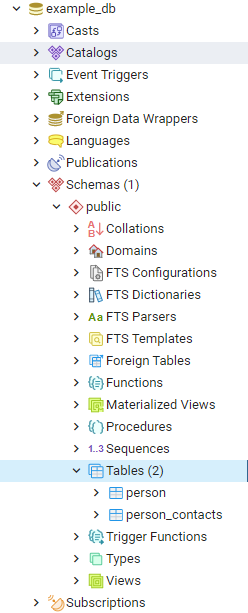


Рисунок 2 — база данных

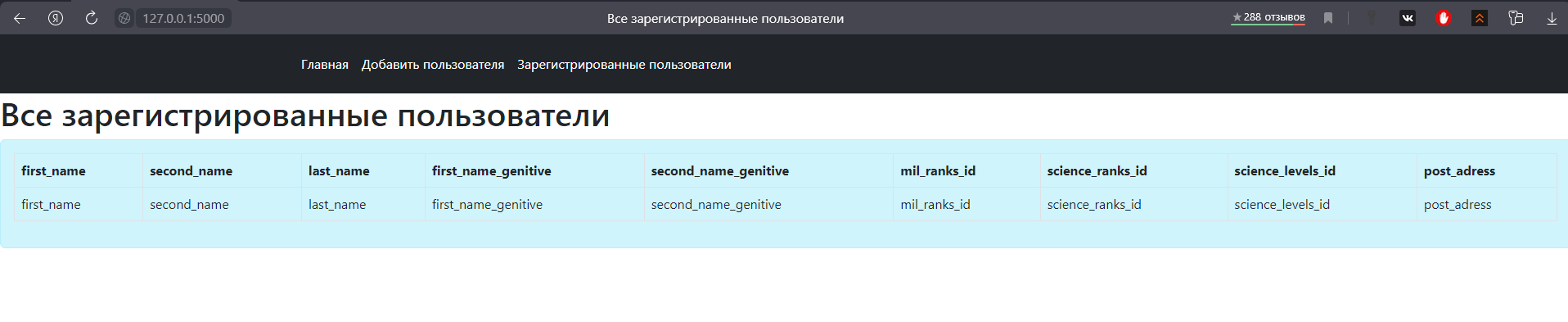


Рисунок 3 — зарегистрированные пользователи

На странице «Зарегистрированные пользователи» выводятся данные таблицы person каждого пользователя

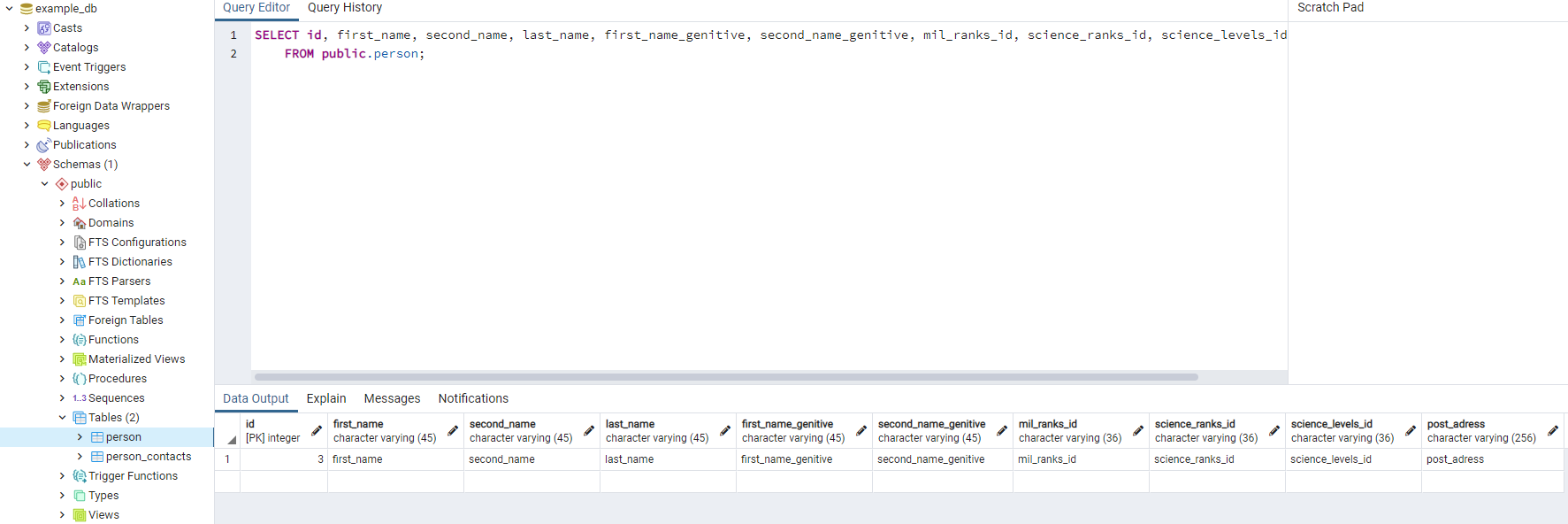


Рисунок 4 — данные в таблице person

# Код программы

Страница «Добавить пользователя» реализована следующим блоком кода

1. @addPerson.route('/', methods**=**['POST', 'GET'])
2. **def** create\_user():
3. **from** main **import** Person, db, app, Person\_contacts
4. **if** request.method **==** "POST":
5. with app.app\_context():
6. first\_name **=** request.form['first\_name']
7. second\_name **=** request.form['second\_name']
8. last\_name **=** request.form['last\_name']
9. first\_name\_genitive **=** request.form['first\_name\_genitive']
10. second\_name\_genitive **=** request.form['second\_name\_genitive']
11. mil\_ranks\_id **=** request.form['mil\_ranks\_id']
12. science\_ranks\_id **=** request.form['science\_ranks\_id']
13. science\_levels\_id **=** request.form['science\_levels\_id']
14. post\_adress **=** request.form['post\_adress']
16. person **=** Person(first\_name**=**first\_name, second\_name**=**second\_name, last\_name**=**last\_name,
17. first\_name\_genitive**=**first\_name\_genitive, second\_name\_genitive**=**second\_name\_genitive,
18. mil\_ranks\_id**=**mil\_ranks\_id, science\_ranks\_id**=**science\_ranks\_id,
19. science\_levels\_id**=**science\_levels\_id, post\_adress**=**post\_adress)
21. db.session.add(person)
22. db.session.flush()
24. person\_id **=** request.form['person\_id']
25. contact\_types\_id **=** request.form['contact\_types\_id']
26. value **=** request.form['value']
28. person\_contacts **=** Person\_contacts(person\_id**=**person\_id, contact\_types\_id**=**contact\_types\_id,
29. value**=**value, Person\_id**=**person.id)
31. db.session.add(person\_contacts)
32. db.session.commit()
33. **return** redirect('/all-users')
34. db.session.rollback()
35. **return** "Ошибка добавления в базу данных"
36. **else**:
37. **return** render\_template("add\_person/myProfile.html")

А также html страница к этому коду

1. {% extends 'pattern.html' %}
3. {% block body %}
4. <**div** class="container mt-5">
5. <**form** action="/add\_person" method="post">
6. Person
7. <**input** type="text" name="first\_name" id="first\_name" class="form-control" placeholder="first\_name"><**br**>
8. <**input** type="text" name="second\_name" id="second\_name" class="form-control" placeholder="second\_name"><**br**>
9. <**input** type="text" name="last\_name" id="last\_name" class="form-control" placeholder="last\_name"><**br**>
10. <**input** type="text" name="first\_name\_genitive" id="first\_name\_genitive" class="form-control" placeholder="first\_name\_genitive"><**br**>
11. <**input** type="text" name="second\_name\_genitive" id="second\_name\_genitive" class="form-control" placeholder="second\_name\_genitive"><**br**>
12. <**input** type="text" name="mil\_ranks\_id" id="mil\_ranks\_id" class="form-control" placeholder="mil\_ranks\_id"><**br**>
13. <**input** type="text" name="science\_ranks\_id" id="science\_ranks\_id" class="form-control" placeholder="science\_ranks\_id"><**br**>
14. <**input** type="text" name="science\_levels\_id" id="science\_levels\_id" class="form-control" placeholder="science\_levels\_id"><**br**>
15. <**input** type="text" name="post\_adress" id="post\_adress" class="form-control" placeholder="post\_adress"><**br**>
16. Person\_contacts
17. <**input** type="text" name="person\_id" id="person\_id" class="form-control" placeholder="person\_id"><**br**>
18. <**input** type="text" name="contact\_types\_id" id="contact\_types\_id" class="form-control" placeholder="contact\_types\_id"><**br**>
19. <**input** type="text" name="value" id="value" class="form-control" placeholder="value"><**br**>
20. <**input** type="submit" class="btn btn-success" value="Добавить">
21. </**form**>
22. </**div**>
23. {% endblock %}

Страница «Зарегистрированные пользователи» реализована следующим блоком кода

1. @app.route('/all-users')
2. **def** all\_users():
3. info **=** Person.query.order\_by(Person.id).all()
4. **return** render\_template("all\_users.html", articles**=**info)

А также html страница к этому коду

1. {% extends 'pattern.html' %}
3. {% block title %}
4. Все зарегистрированные пользователи
5. {% endblock %}

8. {% block body %}
9. <**h1**>Все зарегистрированные пользователи</**h1**>
10. <**div** class="alert alert-info">
11. <**table** class="table table-bordered">
12. <**thead**>
13. <**tr**>
14. <**th**>first\_name</**th**>
15. <**th**>second\_name</**th**>
16. <**th**>last\_name</**th**>
17. <**th**>first\_name\_genitive</**th**>
18. <**th**>second\_name\_genitive</**th**>
19. <**th**>mil\_ranks\_id</**th**>
20. <**th**>science\_ranks\_id</**th**>
21. <**th**>science\_levels\_id</**th**>
22. <**th**>post\_adress</**th**>
23. </**tr**>
24. </**thead**>
25. <**tbody**>
26. {% for el in articles %}
27. <**tr**>
28. <**td**>{{ el.first\_name }}</**td**>
29. <**td**>{{ el.second\_name }}</**td**>
30. <**td**>{{ el.last\_name }}</**td**>
31. <**td**>{{ el.first\_name\_genitive }}</**td**>
32. <**td**>{{ el.second\_name\_genitive }}</**td**>
33. <**td**>{{ el.mil\_ranks\_id }}</**td**>
34. <**td**>{{ el.science\_ranks\_id }}</**td**>
35. <**td**>{{ el.science\_levels\_id }}</**td**>
36. <**td**>{{ el.post\_adress }}</**td**>
37. </**tr**>
38. {% endfor %}
39. </**tbody**>
40. </**table**>
41. </**div**>
42. {% endblock %}

За создание базы данных отвечает следующий блок кода

1. app.config['SQLALCHEMY\_DATABASE\_URI'] **=** '<postgresql://postgres:AlEx1902345@localhost/example_db>'
2. db **=** SQLAlchemy(app)
3. **class** Person(db.Model):
4. id **=** db.Column(db.Integer, primary\_key**=**True)
5. first\_name **=** db.Column(db.String(45), nullable**=**False)
6. second\_name **=** db.Column(db.String(45), nullable**=**False)
7. last\_name **=** db.Column(db.String(45), nullable**=**False)
8. first\_name\_genitive **=** db.Column(db.String(45), nullable**=**False)
9. second\_name\_genitive **=** db.Column(db.String(45), nullable**=**False)
10. mil\_ranks\_id **=** db.Column(db.String(36), nullable**=**False)
11. science\_ranks\_id **=** db.Column(db.String(36), nullable**=**False)
12. science\_levels\_id **=** db.Column(db.String(36), nullable**=**False)
13. post\_adress **=** db.Column(db.String(256), nullable**=**False)
15. **def** \_\_repr\_\_(self):
16. **return** f"<Person {self.id}>"

19. **class** Person\_contacts(db.Model):
20. id **=** db.Column(db.Integer, primary\_key**=**True)
21. person\_id **=** db.Column(db.String(36), nullable**=**False)
22. contact\_types\_id **=** db.Column(db.String(36), nullable**=**False)
23. value **=** db.Column(db.String(256), nullable**=**False)
25. Person\_id **=** db.Column(db.Integer, db.ForeignKey('person.id'))
27. **def** \_\_repr\_\_(self):
28. **return** f"<Person\_contacts {self.id}>"

31. with app.app\_context():
32. db.create\_all()