

# **Zaawansowane techniki bazodanowe**

## **Zadanie 7**

**Tomasz Jakubowski**

**nr albumu 7043**

# Dodawanie rodzaju projektu

Rodzaj	rodzaj proje	Zapisz
--------	--------------	--------

Figure 1: Dodawanie rodzaju projektu

```
sqlite> select * from rodzaj;  
24|  
25|1  
26|2  
27|3  
sqlite> select * from rodzaj;  
24|  
25|1  
26|2  
27|3  
28|  
29|  
sqlite> █
```

Figure 2: Wciąż można dodawać puste rekordy

```

@app.route('/dodaj_rodzaj', methods=['GET', 'POST'])
def dodaj_rodzaj():
    if request.method == 'POST':
        nowy_rodzaj = request.form['rodzaj']
        print(f'nowy rodzaj: "{nowy_rodzaj}"')
        if nowy_rodzaj != '':
            nowy_rekord = Rodzaj(nazwa_rodzaj=nowy_rodzaj)
            db.session.add(nowy_rekord)
            db.session.commit()
        return render_template('dodaj_rodzaj.html')

```

Figure 3: Kod Python - sprawdzenie czy podano wartość w polu formularza

```

sqlite> select * from rodzaj;
3611
sqlite>

```

Figure 4: Jak widzieć pusty rekord nie zostaje dodany

```

{% with messages = get_flashed_messages() %}
{% if messages %}
    <ul class="text-error">
        {% for message in messages %}
            <li>{{ message }}</li>
        {% endfor %}
    </ul>
{% endif %}
{% endwith %}

```

Figure 5: Stworzenie obiektu wiadomości w głównej szablonie

```

if nowy_rodzaj != '':
    nowy_rekord = Rodzaj(nazwa_rodzaj=nowy_rodzaj)
    db.session.add(nowy_rekord)
    db.session.commit()
else:
    flash('Pole nie może być puste!')

```

Figure 6: Wyświetlenie wiadomości

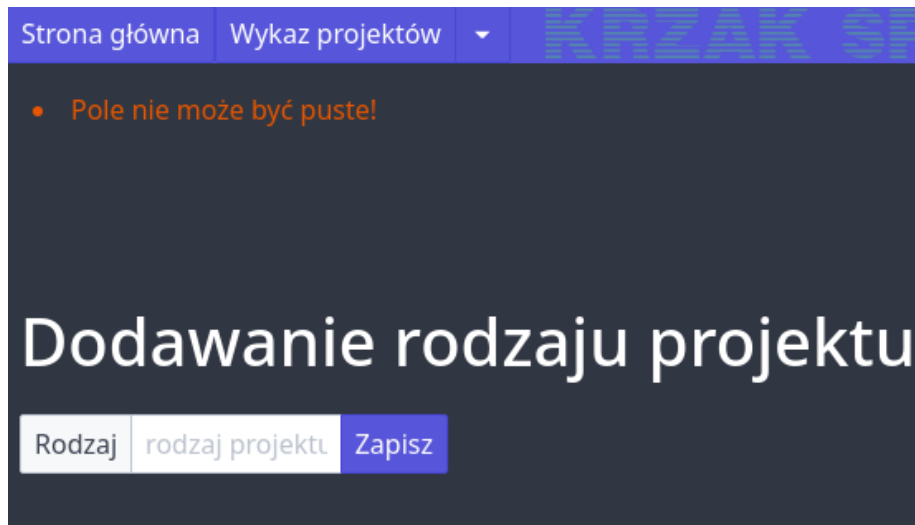


Figure 7: Wyświetlona wiadomość

```
def dodaj_rodzaj():  
    if request.method == 'POST':  
        nowy_rodzaj = request.form['rodzaj']  
  
        if nowy_rodzaj == '':  
            flash('Pole nie może być puste!')  
        elif db.session.query(Rodzaj.nazwa_rodzaj).  
            filter(Rodzaj.nazwa_rodzaj == nowy_rodzaj).  
            first() is not None:  
            flash('Podany rodzaj już istnieje!')  
        else:  
            nowy_rekord = Rodzaj(nazwa_rodzaj=nowy_rodzaj)  
            db.session.add(nowy_rekord)  
            db.session.commit()  
  
    return render_template('dodaj_rodzaj.html')
```

Figure 8: Sprawdzenie już istniejącego rodzaju

```
sqlite> select * from rodzaj;  
43|1  
44|213  
sqlite>
```

Figure 9: Dodanie duplikatu jest teraz niemożliwe

# Dodawanie rodzaju projektu

Rodzaj	333	Zapisz
--------	-----	--------

Figure 10: Dodanie rodzaju projektu

```
sqlite> select * from rodzaj;  
43|1  
44|213  
45|333
```

Figure 11: Jak widać na obrazku, dodawanie i tu działa

# Dodawanie rodzaju projektu

Rodzaj	213	Zapisz
--------	-----	--------

Figure 12: Próba dodania już istniejącego rekordu

- Podany rodzaj już istnieje!

# Dodawanie rodzaju projektu

Rodzaj	rodzaj projektu	Zapisz
--------	-----------------	--------

Figure 13: Komunikat o już istniejącym rekordzie

```
sqlite> select * from rodzaj;  
43|1  
44|213  
45|333  
sqlite> 
```

Figure 14: Jak widać, nie udało się dodać duplikatu

- Pole nie może być puste!

## Dodawanie rodzaju projektu

Rodzaj	rodzaj projektu	Zapisz
--------	-----------------	--------

Figure 15: Komunikat o pustym polu dla rodzaju

```
sqlite> select * from rodzaj;  
43|1  
44|213  
45|333  
sqlite> 
```

Figure 16: Pustego rekordu również nie da się dodać

```

@app.route('/dodaj_status')
def dodaj_status():
    if request.method == 'POST':
        nowy_status = request.form['status']

        if nowy_status == '':
            flash('Pole nie może być puste!')
        elif db.session.query(Status.nazwa_statusj).\
            filter(Status.nazwa_status == nowy_status).\
            first() is not None:
            flash('Podany status już istnieje!')
        else:
            nowy_rekord = Status(nazwa_status=nowy_status)
            db.session.add(nowy_rekord)
            db.session.commit()

    return render_template('dodaj_status.html')

```

Figure 17: Kod Python dodawania statusu

```

{% block content %}
<h1>Dodawanie statusu projektu</h1>
<form action="/dodaj_status" method="POST">
  <div class="form-group">
    <div class="col-3 col-sm-12">
      <div class="input-group">
        <span class="input-group-addon text-dark">Rodzaj</span>
        <input type="text" class="form-input" name="status" placeholder="status projektu">
        <button type="submit" class="btn btn-primary input-group-btn">Zapisz</button>
      </div>
    </div>
  </div>
</form>
{% endblock %}

```

Figure 18: Kod HTML dodawania statusu, analogiczny jest dla dodawania rodzaju

```

@app.route('/dodaj_status', methods=['GET', 'POST'])
def dodaj_status():
    if request.method == 'POST':
        nowy_status = request.form['status']

    if nowy_status == '':

```

Figure 19: Dodanie metody POST w kodzie Python

• Pole nie może być puste!

## Dodawanie statusu projektu

Rodzaj	status projektu	Zapisz
--------	-----------------	--------

Figure 20: Komunikat o pustym polu dla statusu

## Dodawanie statusu projektu

Rodzaj	123	Zapisz
--------	-----	--------

Figure 21: Dodanie statusu

```
sqlite> select * from status;  
1|1  
2|123  
sqlite>
```

Figure 22: Zawartość tabeli status po dodaniu



- Podany status już istnieje!

## Dodawanie statusu projektu

Rodzaj status projektu Zapisz

Figure 23: Jak widać, nie da się dodać duplikatu

```
@app.route('/edytuj_rodzaj', methods=['GET', 'POST'])
def edytuj_rodzaj():
    db_class = Rodzaj
    db_pk = Rodzaj.id_rodzaj
    db_pk_str = 'id_rodzaj'
    editable_columns = ['nazwa_rodzaj']

    if request.method == 'POST':
        if request.form['action'] == 'send':
            query = db.session.query(db_class).filter(db_pk == request.form[db_pk_str])
            print(666)
            print(type(query))
            if query is not None:
                mydict = {column.name : request.form[column.name] for column in db_class.__table__.columns if column.name in editable_columns}
                query.update(mydict)
                db.session.commit()

            elif request.form['action'] == 'delete':
                query = db.session.query(Rodzaj).filter(db_pk == request.form[db_pk_str])
                if query is not None:
                    query.delete()
                    db.session.commit()

    return render_template('edytuj_rodzaj.html',
                            db_class=db_class,
                            db_query=db.session.query(db_class),
                            editable_columns=editable_columns)
```

Figure 24: Kod Python edytowania rodzaju projektu

[illegible]

Figure 25: Kod HTML edytowania rodzaju projektu

```
{% block content %}
<h1>Edytowanie rodzaju projektu</h1>

<table class="table table-scroll bg-dark">
  <thead>
    <tr>
      <th></th>
      <th></th>
      <th></th>
      <th></th>
    </tr>
    {% for column in db_class._table_.columns %}
      <th>{{ column.name }}</th>
    {% endfor %}
  </thead>
  <tbody>
    {% for row in db_query.all() %}
      {% set idx = loop.index %}
      <tr id="{{row.id}}">
        <td id="edit-btn-{{idx}}"><button class="btn btn-primary" onclick="start_edit('{{idx}}')">Edytuj</button></td>
        <td id="filler-1-{{idx}}"></td>
        <td id="filler-2-{{idx}}"></td>
        <td id="cancel-btn-{{idx}}" class="d-hide"><button class="btn btn-primary" onclick="stop_edit('{{idx}}')">Anuluj</button></td>
      </tr>
      <form action="/admin/rodzaj/" method="POST">
        <td id="send-btn-{{idx}}" class="d-hide"><button type="submit" name="action" value="send" class="btn btn-success">Wyślij</td>
        <td id="delete-btn-{{idx}}" class="d-hide"><button type="submit" name="action" value="delete" class="btn btn-error">Usuń</td>
        {% for column in db_class._table_.columns %}
          <td>
            {% if column.name in editable_columns %}
              <div class="editable-label">{{ row._dict_[column.name] }}</div>
            <input class="form-input editable-input d-hide" name="{{ column.name }}" type="text" value="{{ row._dict_[column.name] }}" />
            {% else %}
              <div>{{ row._dict_[column.name] }}</div>
            <input class="form-input d-hide" name="{{ column.name }}" type="text" value="{{ row._dict_[column.name] }}" />
            </td>
          {% endif %}
        {% endfor %}
      </form>
    </tr>
  </tbody>
</table>
{% endblock %}
```

Figure 26: Kod HTML edytowania rodzaju projektu c.d.

## Edytowanie rodzaju projektu

	id_rodzaj	nazwa_rodzaj
Edytuj	62	22222222222222
Edytuj	63	dsfdss
Edytuj	64	r3erw
Edytuj	65	234c
Edytuj	66	fggdgdf
Edytuj	67	dfgdf
Edytuj	68	dfg

Figure 27: Edytowanie rodzaju projektu widok grafiki

			id_rodzaj	nazwa_rodzaj
Anuluj	Wyślij	Usuń	62	22222222222222
Edytuj			63	dsfdss
Anuluj	Wyślij	Usuń	64	r3erw
Edytuj			65	234

Figure 28: Rozwijanie menu dla każdego rekordu i pola do edycji

		id_rodzaj	nazwa_rodzaj
Edytuj		63	dsfdss
Edytuj		64	r3erw
Edytuj		65	234c
Edytuj		66	fggdfgdf
Edytuj		67	dfgdf
Edytuj		68	dfg

Figure 29: Rekord 62 został usunięty

			id_rodzaj	nazwa_rodzaj
Anuluj	Wyślij	Usuń	63	testowanie1234
Edytuj			64	r3erw

Figure 30: Dodawanie nowego rekordu

			id_rodzaj	nazwa_rodzaj
Edytuj			63	testowanie1234
Edytuj			64	r3erw

Figure 31: Nowy rekord widoczny

## Edytowanie statusu projektu

	id_status	nazwa_status
Edytuj	1	1
Anuluj Wyślij Usuń	2	123qqq
Edytuj	3	2
Edytuj	4	4
Edytuj	5	eeee gfd
Edytuj	6	sdf

Figure 32: Edytowanie statusu, kod analogiczny do rodzaju, dodawanie rekordu

```
elif request.form['action'] == 'delete':
    query = db.session.query(db_class).fil
    if query is not None:
        query.delete()
        db.session.commit()
```

Figure 33: Kod Python usuwania - fragment

Edytuj		
Anuluj Wyślij Usuń	5	eeee gfd
Edytuj	6	sdf

Figure 34: Usuwanie rekordu



```

{% block content %}
<form action="/dodaj_projekt" method="POST">
  <div class="form-group">
    <div class="col-3 col-sm-12">
      <label class="form-label" for="rodzaj-input">Rodzaj projektu</label>
      <select id="rodzaj-input" type="text" class="form-select text-dark" name="rodzaj">
        {% for row in db.session.query(Rodzaj).all() %}
          <option>{{ row.__dict__['nazwa_rodzaj'] }}</option>
        {% endfor %}
      </select>
    </div>
    <div class="col-3 col-sm-12">
      <label class="form-label" for="status-input">Status projektu</label>
      <select id="status-input" type="text" class="form-select text-dark" name="status">
        {% for row in db.session.query(Status).all() %}
          <option>{{ row.__dict__['nazwa_status'] }}</option>
        {% endfor %}
      </select>
    </div>
    <div class="col-3 col-sm-12">
      <label class="form-label" for="nr-input">Numer projektu</label>
      <input id="numer-input" type="text" class="form-input" name="nr_projekt" placeholder="Wpisz numer"/>
    </div>
    <div class="col-3 col-sm-12">
      <label class="form-label" for="temat-input">Temat projektu</label>
      <textarea id="temat-input" class="form-input" name="temat_projekt" placeholder="Wpisz temat"></textarea>
    </div>
    <div class="col-3 col-sm-12">
      <label class="form-label" for="data-rozp-input">Data rozpoczęcia</label>
      <input id="data-rozp-input" type="date" class="form-input" name="data_roz poczenia"/>
    </div>
    <div class="col-3 col-sm-12">
      <label class="form-label" for="data-zak-input">Data zakończenia</label>
      <input id="data-zak-input" type="date" class="form-input" name="data_zakonczenia"/>
    </div>
    <div class="col-3 col-sm-12">
      <label class="form-label" for="kwota-input">Kwota projektu</label>
      <input id="kwota-input" type="text" class="form-input" name="kwota" placeholder="Wpisz kwotę"/>
    </div>
    <div class="col-3 col-sm-12">
      <label class="form-label" for="uwagi-input">Uwagi</label>
      <textarea id="uwagi-input" class="form-input" name="uwagi" placeholder="Opisowe uwagi"></textarea>
    </div>
    <br/>
    <button type="submit" class="btn btn-primary">Wyślij</button>
  </div>
</form>
{% endblock %}

```

Figure 37: Kod HTML formularza dodawania projektu



Rodzaj projektu

budowlany

Status projektu

nowy

Numer projektu

Wpisz numer

Temat projektu

Wpisz temat

Data rozpoczęcia

dd.mm.rrrr

Data zakończenia

dd.mm.rrrr

Kwota projektu

Wpisz kwotę

Uwagi

Opcjonalne uwagi

Wyślij

Figure 38: Widok formularza

# Dodawanie projektu

Rodzaj projektu

budowlany

Status projektu

nowy

Numer projektu

1/asd/456

Temat projektu

to jest temat

Data rozpoczęcia

13.01.2022

Data zakończenia

11.01.2022

Kwota projektu

123

Uwagi

przykładowa uwaga

Wyślij

Figure 39: Wypełnione pola formularza

# Dodawanie projektu

Rodzaj projektu

budowlany

Status projektu

nowy

Numer projektu

Wpisz numer

Temat projektu

Wpisz temat

Data rozpoczęcia

dd.mm.rrrr

Data zakończenia

dd.mm.rrrr

Kwota projektu

Wpisz kwotę

Uwagi

Opcjonalne uwagi

Wyślij

Figure 40: Po kliknięciu Wyślij pola są czyszczone

- Wystąpił błąd time data " does not match format '%Y-%m-%d'

Figure 41: Błąd w przyparku niewybrania daty

- Wypełnij poprawnie wszystkie pola!

Figure 42: Błąd w przypadku pozostawienia pustych pól

```
sqlite> select * from projekt;
1|64|1|12312sadasdasd|123|2022-01-06|2022-01-05|123123|33333 cddddd
2|64|1|3333|witam|2022-01-05|2022-01-27|11111111111122|dasdasd
3|71|7|1/asd/456|to jest temat|2022-01-13|2022-01-11|123|przykładowa uwaga
sqlite>
```

Figure 43: Dodane rekordy w bazie

```
@app.route('/wykaz_projektow')
def wykaz_projektow():
    columns = [ x.name for x in Projekt.__table__.columns ]
    columns.remove('id_rodzaj')
    columns.remove('id_status')
    columns.append('nazwa_status')
    columns.append('nazwa_rodzaj')
    rows = db.session.query(Projekt,Rodzaj,Status).all()[0]

    return render_template('wykaz_projektow.html', columns=columns, rows=rows)
```

Figure 44: Wykaz projektów kod Python

Wykaz projektów							
id_projekt	nr_projekt	temat_projekt	data_roz poczenia	data_zakonczenia	kwota	uwagi	na
6	1/2020	dom	2020-01-23	2022-01-05	500000.0000000000	brak cegieł	
6	1/2020	dom	2020-01-23	2022-01-05	500000.0000000000	brak cegieł	
6	1/2020	dom	2020-01-23	2022-01-05	500000.0000000000	brak cegieł	

Figure 45: Wykaz projektów widok grafiki

```

{% extends "bazowa_templatka.html" %}

{% block title %}
Wykaz projektów
{% endblock %}

{% block content %}
<h1>Wykaz projektów</h1>

<table class="table table-scroll bg-dark">
  <thead>
    <tr>
      {% for column in columns %}
        <th>{{ column }}</th>
      {% endfor %}
    </tr>
  </thead>
  <tbody>
    {% for row in rows %}
      <tr>
        {% for column in columns %}
          <td>
            <div>{{ row.__dict__[column] }}</div>
            <div>{{ row.__dict__[column] }}</div>
            <div>{{ row.__dict__[column] }}</div>
          </td>
        {% endfor %}
      </tr>
    {% endfor %}
  </tbody>
</table>
{% endblock %}

```

Figure 46: Wykaz projektów kod HTML