

Temat: System rezerwacji – Fitness Klub – język Python + Flask

Część 4: Obsługa wielu tabeli w bazie danych. Sporządzanie raportów. Kontrola integralności bazy danych. Implementacja transakcji i blokowania optymistycznego.

1. Obsługa wielu tabel

Z powodu mało czytelnego sposobu wyświetlania danych z wielu tabel (klienci, trenerzy, zajęcia, sale, sprzęt) zamiast aplikacji okienkowej w TK Inter zdecydowałem się na przeniesienie obsługi bazy poprzez stronę internetową z wykorzystaniem micro frameworka Flask.

a) prezentacja danych

Klienci	Trenerzy	Sale	Zajęcia	Sprzet			
Nazwa	Ilość miejsc	Trener	Godzina rozpoczęcia	Godzina zakończenia	Sala	Modyfikuj	Usun
Crossfit	5	Filip Kulon	10.15	11.15	2	Zmodyfikuj	Usun
Taśmy	2	Wojciech Cichy	15.10	15.50	3	Zmodyfikuj	Usun
<div>Dodaj nowe zajecie</div>							

b) edycja rekordu (na przykładzie tabeli Zajęcia z listy rozwijanej można wybrać wcześniej pobrane rekordy z tabeli Trenerzy oraz Sale)

← → ↺ ⌂ 127.0.0.1:5000/zajecia/8 ☆ ⓘ Ⓞ

Klienci Trenerzy Sale Zajęcia Sprzet

Nazwa

Crossfit

Ilość miejsc

5

Trener

Filip Kulon

Wojciech Cichy

Filip Kulon

10.15

Godzina zakończenia

11.15

Sala

2

[Modyfikuj](#) [Modyfikuj procedura](#)

c) pobieranie i przekazywanie rekordów do odpowiedniej podstrony (na przykładzie tabeli Zajęcia + pobranie tabel Sale i Trenerzy)

```
@app.route('/zajecia/<int:ID_Zajecia>')
def zajecie_poID(ID_Zajecia):
    try:
        zajecie = Zajecia.zwroc_poID(ID_Zajecia)
        sale = Sala.odczyt()
        trenerzy = Trenerzy.odczyt()
        return render_template('zajecia/modyfikuj_zajecie.html', zajecie=zajecie, sale=sale, trenerzy=trenerzy)
    except NameError as error:
        flash(str(error))
        return render_template('main.html')

@app.route('/zajecia/<int:ID_Zajecia>', methods=['POST'])
def zajecia_modyfikuj(ID_Zajecia):
    if request.method == 'POST':
        try:
            Zajecia.modyfikuj(ID_Zajecia, request.form['Nazwa'], request.form['Ilosc_miejsc'],
                              request.form['ID_Trenera'], request.form['Godzina_rozpoczecia'],
                              request.form['Godzina_zakonczenia'], request.form['ID_Sali'])
            return redirect(url_for('wszystkie_zajecia'))
        except NameError as error:
            flash(str(error))
            return redirect(url_for('zajecie_poID', ID_Zajecia=ID_Zajecia))
```

d) wygląd szablonu w języku HTML + zastosowanie silnika Jinja2 (generowanie widoku prezentowanego w podpunkcie b)

```
{% extends "main.html" %}
{% block title %}modyfikuj klienta{% endblock %}
{% block content %}
<form method="post">
  <div class="form-group">
    <label>Nazwa</label>
    <input type="text" class="form-control" name="Nazwa" value="{{zajecie.Nazwa}}">
  </div>
  <div class="form-group">
    <label>Ilość miejsc</label>
    <input type="text" class="form-control" name="Ilosc_miejsc" value="{{zajecie.Ilosc_miejsc}}">
  </div>
  <div class="form-group">
    <label for="Trener">Trener</label>
    <select class="form-control" id="Trener" name="Trener">
      {% for tr in trenerzy %}
        {% if tr.ID_Trenera == zajecie.ID_Trenera %}
          <option selected>{{tr.Imie}} {{tr.Nazwisko}}</option>
        {% else %}
          <option>{{tr.Imie}} {{tr.Nazwisko}}</option>
        {% endif %}
      {% endfor %}
    </select>
  </div>
  <div class="form-group">
    <label>Godzina rozpoczecia</label>
    <input type="text" class="form-control" name="Godzina_rozpoczecia" value="{{zajecie.Godzina_rozpoczecia}} placeholder="GG.MM">
  </div>
</div>
```

```

<div class="form-group">
<label>Godzina zakończenia</label>
<input type="text" class="form-control" name="Godzina_zakonczenia" value="{{zajecie.Godzina_zakonczenia}}" placeholder="GG.MM">
</div>
<div class="form-group">
<label for="Sala">Sala</label>
<select class="form-control" id="Sala" name="Sala">
{% for sa in sale %}
    {% if sa.ID_Sali == zajecie.ID_Sali %}
    <option selected>{{sa.ID_Sali}}</option>
    {% else %}
    <option>{{sa.ID_Sali}}</option>
    {% endif %}
{% endfor %}
</select>

</div>
<button type="submit" class="btn btn-primary">Modyfikuj</button>
<button type="submit" class="btn btn-primary">Modyfikuj procedura</button>
</form>
{% endblock %}

```

2. Złożone triggery i procedury składowe

a) procedura dodająca i sprawdzająca klientów znajdujących się na Liście uczestników. Jeżeli klient znajduje się na jednej z list (uczestników lub oczekujących) pojawi się stosowny komunikat. Jeżeli klient nie znajduje się na liście uczestników oraz na zajęciach znajduje się miejsce klient zostanie dopisany do uczestników. W przeciwnym wypadku zostanie dopisany do listy oczekujących.

```

alter procedure dodaj_klienta_do_zajec
@ID_Klienta int,
@ID_Zajecia int
as begin
declare @Pojemnosc int set @Pojemnosc = (select Ilosc_miejsc from Zajecia where ID_Zajecia=@ID_Zajecia)
declare @Ilosc_klientow_na_zajeciach int
set @Ilosc_klientow_na_zajeciach = (select count(*) from Uczestnicza where ID_Zajecia=@ID_Zajecia)
declare @Klient_jest_juz_na_liscie int
set @Klient_jest_juz_na_liscie = (select count(*) from Uczestnicza where ID_Klienta=@ID_Klienta and ID_Zajecia=@ID_Zajecia)
declare @Klient_na_liscie_oczekujacych int
set @Klient_na_liscie_oczekujacych = (select count(*) from Lista_oczekujacych where ID_Klienta=@ID_Klienta and ID_Zajecia=@ID_Zajecia)
if @Klient_jest_juz_na_liscie = 1
begin
raiserror ('Uczestniczysz w tych zajeciach. Wybierz inne',16,1)
end
else
if @Pojemnosc>@Ilosc_klientow_na_zajeciach
begin
insert into Uczestnicza(ID_Klienta,ID_Zajecia)
values (@ID_Klienta,@ID_Zajecia)
end
else
if @Klient_na_liscie_oczekujacych=1
begin
raiserror ('Jestes juz na liscie oczekujacych', 16,1)
end
else
begin
insert into Lista_oczekujacych(ID_Klienta,ID_Zajecia)
values (@ID_Klienta,@ID_Zajecia)
raiserror ('Niestety brak miejsc. Zostales dodany do listy oczekujacych',16,1)
end
end
end

```

b) trigger „przesuwający” klientów z listy oczekujących (tabela Lista_oczekujacych) do listy uczestników (tabela Uczestnicza)

```
use System_rezerwacji
go
alter trigger Oczekujacy_jako_Uczestnik
ON Uczestnicza
AFTER DELETE
as
declare @id_zajecia int
set @id_zajecia = (select ID_Zajecia from deleted)
if exists (select * from Lista_oczekujacych where ID_Zajecia=@id_zajecia)
begin
declare @id_klienta int
set @id_klienta = (select top 1 ID_Klienta from Lista_oczekujacych where ID_Zajecia=@id_zajecia)
insert into Uczestnicza values (@id_klienta, @id_zajecia)
delete from Lista_oczekujacych where ID_Klienta=@id_klienta and ID_Zajecia=@id_zajecia
end
```

3. Generowanie raportów

Z racji tego, że raporty generowane przez narzędzie Report Services są dość ograniczone, wygenerowałem interaktywny raport na stronie WWW.

a) procedura – po podaniu ID Trenera zwracani są wszyscy „podopieczni danego trenera”

```
create procedure raport
@id_trenera int
as
begin
select k.Imie, k.Nazwisko, z.Nazwa from Zajecia z inner join Uczestnicza u on z.ID_Zajecia=u.ID_Zajecia
inner join Klienci k on k.ID_Klienta=u.ID_Klienta inner join Trenerzy t on z.ID_Trenera=t.ID_Trenera
where t.ID_Trenera=@id_trenera order by z.Nazwa
end
```

b) tworzenie widoku w HTML

```
{% extends "main.html" %}
{% block title %}dodaj klienta{% endblock %}
{% block content %}
<h1>Lista obecności na zajęciach dla trenera {{trener.Imie}} {{trener.Nazwisko}}</h1>
{% for li in lista %}
{% if trener.ID_Trenera == li.ID_Trenera %}
<h4>{{li.Nazwa}}</h4>
<div>
<table class="table">
<thead>
<tr>
<th>Imie</th>
<th>Nazwisko</th>
<th>Był</th>
</tr>
</thead>
<tbody>
{% for uc in uczestnicy %}
{% if uc.Nazwa == li.Nazwa %}
<form method="get">
<tr>
<td>{{ uc.Imie }}</td>
<td>{{ uc.Nazwisko }}</td>
<td><select class="form-control">
<option>Nieobecny</option>
<option>Obecny</option>
</select>
</td>
</tr>
</form>
{% endif %}
{% endfor %}

```

c) prezentacja „raportu”

[Klenci](#) [Trenerzy](#) [Sale](#) [Zajecia](#) [Sprzet](#)

Lista obecności na zajęciach dla trenera Zenon Cichy

Taśmy

Imię	Nazwisko	Był
Adam	Malysz	<div>Nieobecny</div>
Piotr	Witczak	<div>Nieobecny</div>

Rowerki

Imię	Nazwisko	Był
Anton	Braciszko	<div>Nieobecny</div>
Stefan	Was	<div>Nieobecny</div>

Zatwierdź

4. Transakcje

Stworzony trigger w punkcie 2 b mógł spowodować pojawienie się Klienta (dwóch takich samych rekordów) w dwóch tabelach Uczestnicza i Lista_oczekujacych (Klient nie może być jednocześnie uczestnikiem zajęć oraz znajdować się w liście oczekujących na zwolnienie miejsc na zajęciach). Z tego powodu w triggerze dodano mechanizm transakcji.

```
use System_rezerwacji
go
alter trigger Oczekujacy_jako_Uczestnik
ON Uczestnicza
AFTER DELETE
as
declare @id_zajecia int
set @id_zajecia = (select ID_Zajecia from deleted)
if exists (select * from Lista_oczekujacych where ID_Zajecia=@id_zajecia)
begin
declare @id_klienta int
set @id_klienta = (select top 1 ID_Klienta from Lista_oczekujacych where ID_Zajecia=@id_zajecia)
BEGIN TRANSACTION
insert into Uczestnicza values (@id_klienta, @id_zajecia)
IF @@ERROR <> 0
BEGIN
RAISERROR ('Błąd, operacja nie udana!', 16, 1)
ROLLBACK TRANSACTION
END
delete from Lista_oczekujacych where ID_Klienta=@id_klienta and ID_Zajecia=@id_zajecia
IF @@ERROR <> 0
BEGIN
RAISERROR ('Błąd, operacja nie udana!', 16, 1)
ROLLBACK TRANSACTION
END
COMMIT TRANSACTION
end
```

5. Współbieżność i blokowanie optymistyczne

W implementowanej aplikacji może wystąpić problem podczas współbieżnej pracy pracowników. W trakcie pracy kilka osób może próbować edytować ten sam rekord. W celu zapobiegania występowania tego problemu, zastosowałem blokowanie optymistyczne (pozwala na równoczesny odczyt danych przez wielu użytkowników, a dostęp blokowany jest rzadziej niż w przypadku blokowania pesymistycznego). Do implementacji blokowania optymistycznego zastosowałem metodę z biblioteki SQL Alchemy `with_for_update()`

Przykład użycia dla modyfikacji sprzętu

```
@staticmethod
def modyfikuj(ID_Sprzetu, Nazwa, Ilosc, ID_Sali):
    try:
        session = create_session(bind=engine)
        q = session.query(Sprzet).with_for_update().filter(Sprzet.ID_Sprzetu == ID_Sprzetu)
        record = q.one()
        record.Nazwa = Nazwa
        record.Ilosc_miejsc = Ilosc
        record.ID_Sali = ID_Sali
        session.flush()
        session.close()

    except:
        session.rollback()
```