

# Wielodostępowe Struktury Danych

## Projekt: Biblioteka

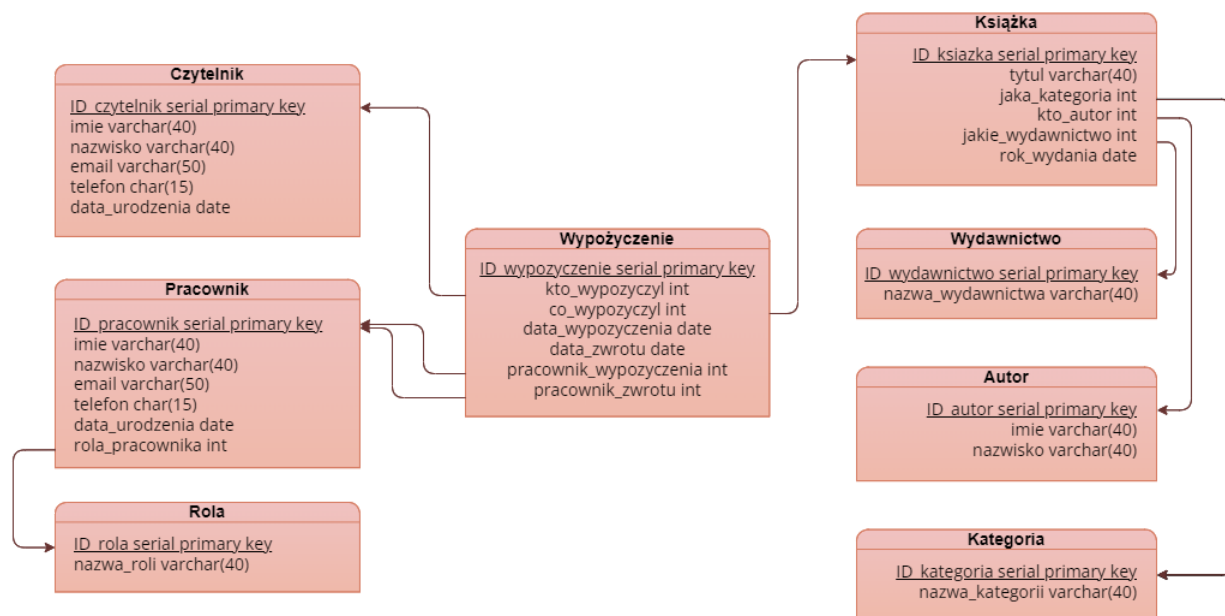
---

### Etap I

Dany etap zawiera:

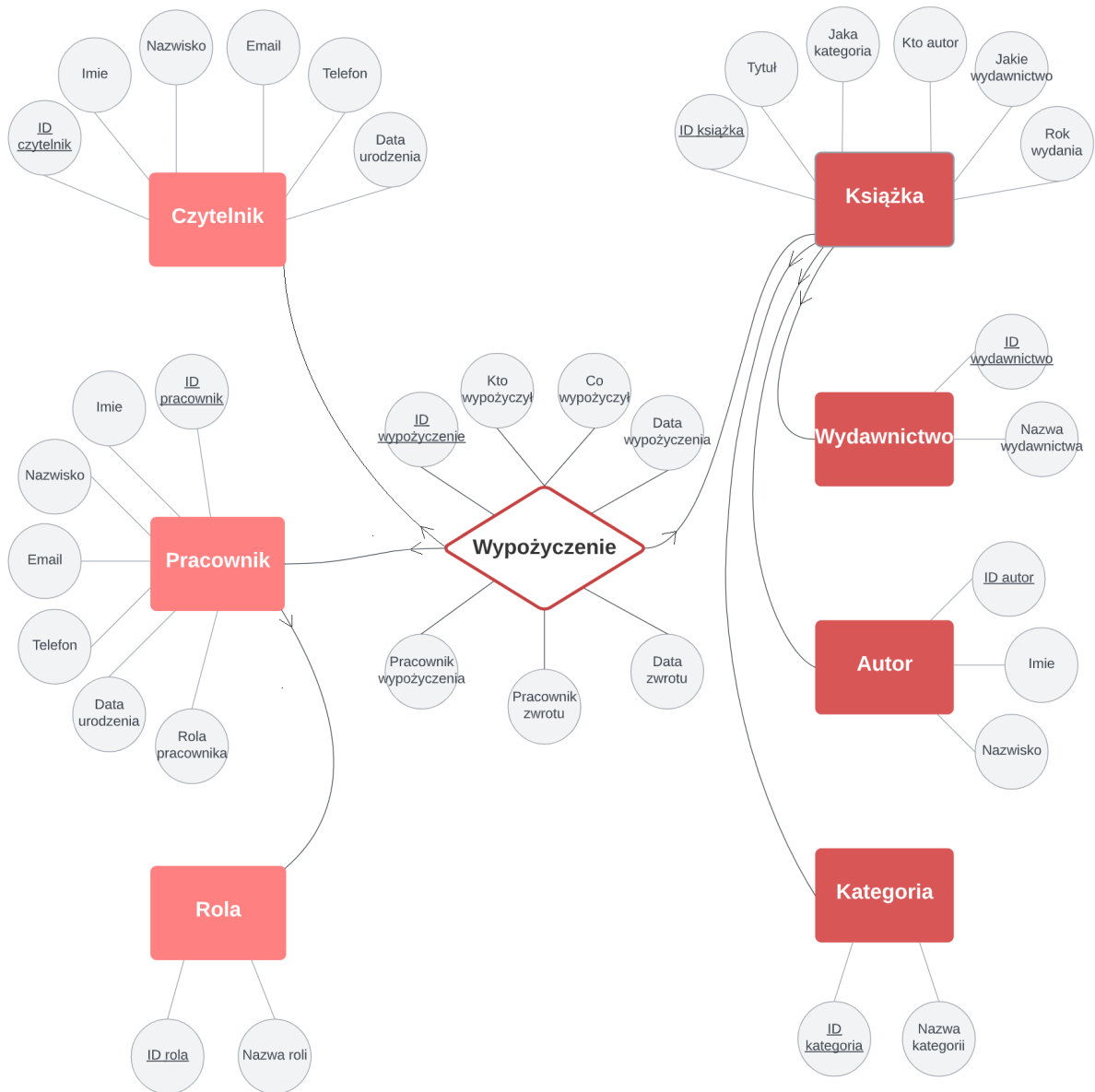
- diagram relacyjny oraz
- diagram związków encji stworzonej bazy danych,
- kod SQL zakładający tabele.

#### 1. Diagram relacyjny stworzonej bazy danych "Biblioteka".



#### 2. Diagram związków encji stworzonej bazy danych "Biblioteka".

---



### 3. Kod SQL zakładający tabele wraz z opisem tabel.

---

Stworzona baza danych składa się z 8 tabel, odpowiednio ze sobą powiązanych. Są to tabeli:

- Czytelnik
- Pracownik
- Rola
- Książka
- Wydawnictwo
- Autor
- Kategoria
- Wypożyczenie.

Tabela "Czytelnik" zawiera wszystkie dane o czytelniku - jego imie, nazwisko, email, telefon, datę urodzenia oraz ID czytelnika - unikalny numer identyfikacyjny czytelnika w bibliotece.

Kod tworzący tabelę:

```
CREATE TABLE czytelnik(  
ID_czytelnik serial primary key,  
imie varchar(40) not null,  
nazwisko varchar(40) not null,  
email varchar(50) not null,  
telefon char(15) not null,  
data_urodzenia date);
```

id_czytelnik	imie	nazwisko	email	telefon	data_urodzenia
1	Jan	Adamski	jan.adamski@gmail.com	765123456	2000-01-04
2	Natalia	Lewandowska	natalia.lewandowska@gmail.com	765125776	2000-11-14
3	Marek	Górecki	marek.gorecki@gmail.com	745165076	1999-08-22
4	Małgorzata	Kalinowska	malgorzata.kalinowska@gmail.com	741155096	2001-09-01
5	Joanna	Gostomska	joanna.gostomska@gmail.com	712386932	2001-03-30
6	Paweł	Rewig	pawel.rewig@gmail.com	707847212	2000-07-15
7	Weronika	Adamiak	weronika.adamiak@gmail.com	722956372	2000-01-29
8	Adam	Jankowski	adam.jankowski@gmail.com	722098764	1998-12-25
9	Łukasz	Szumiński	lukasz.szuminski@gmail.com	701236548	2000-04-11
10	Julia	Wilemska	julia.wilemska@gmail.com	765496032	2002-05-20

(10 wierszy)

Rys.1 - Tabela "Czytelnik"

Tabela "Pracownik", podobnie do poprzedniej tabeli, zawiera dane o pracownikach biblioteki - jest to imie, nazwisko, email, telefon, datę urodzenia, rola w bibliotece oraz ID

pracownika - unikalny numer identyfikacyjny pracownika w bibliotece. Zdefiniowano powiązania tabeli "Pracownik" z tabelą "Rola". Powiązane to polega na zdefiniowaniu klucza obcego w tabeli "Pracownik" (kolumna "rola pracownika"), który odwołuje się do wartości klucza głównego w tabeli "Rola" (tak jak to opisują strzałki na DR).

Kod tworzący tabelę:

```
CREATE TABLE pracownik(
ID_pracownik serial primary key,
imie varchar(40) not null,
nazwisko varchar(40) not null,
email varchar(50),
telefon char(15),
data_urodzenia date,
rola_pracownika int not null
references rola on delete restrict
on update restrict);
```

id_pracownik	imie	nazwisko	email	telefon	data_urodzenia	rola_pracownika
1	Agata	Wilemska	agata.wilemska@gmail.com	776782931	2000-01-04	1
2	Justyna	Nowak	justyna.nowak@gmail.com	712378567	1995-06-19	1
3	Katarzyna	Wasilewska	katarzyna.wasilewska@gmail.com	734562980	1999-05-01	2
5	Jakub	Tomaszewski	jakub.tomaszewski@gmail.com	643242980	1997-11-24	4
4	Natalia	Ziółkowska	natalia.ziolkowska@gmail.com	782516382	1990-08-17	3

(5 wierszy)

Rys.2 - Tabela "Pracownik"

W tabeli "Rola" są zamieszczone możliwe role (stanowiska), które pracownik biblioteki może pełnić. Tabela zawiera jedynie nazwę roli oraz ID roli - unikalny numer identyfikacyjny konkretnej roli.

Kod tworzący tabelę:

```
CREATE TABLE rola(
ID_rola serial primary key,
nazwa_rola varchar(40) not null);
```

id_rola	nazwa_rola
1	bibliotekarz
2	księgowy
3	dyrektor
4	pomocnik bibliotekarza
5	czasowe zastępstwo

(5 wierszy)

Rys.3 - Tabela "Rola"

W tabeli "Książka" są zamieszczone dane o posiadanych przez bibliotekę książkach. Tabela zawiera dane tytułów, kategorii, autorów, wydawnictw, roku wydania i ID każdej z

posiadanych książki, w tym kategoria, autor i wydawnictwo są kluczami obcymi i odwołują się bezpośrednio do tabel "Autor", "Kategoria" i "Wydawnictwo".

Kod tworzący tabelę:

```
CREATE TABLE ksiazka(  
ID_ksiazka serial primary key,  
tytul varchar(40) not null,  
jaka_kategoria int not null  
references kategoria on delete  
restrict on update restrict,  
kto_autor int not null  
references autor on delete  
restrict on update restrict,  
jakie_wydawnictwo int not null  
references wydawnictwo on delete  
restrict on update restrict,  
rok_wydania date);
```

id_ksiazka	tytul	jaka_kategoria	kto_autor	jakie_wydawnictwo	rok_wydania
1	Zaginiona Siostra - Siedem Sióstr	1	4	3	2018-01-01
2	Niewidzialne życie Addie LaRue	3	1	7	2020-06-01
3	Święto Ognia	1	10	4	2022-01-01
4	Kobieta w białym kimonie	2	7	1	2021-06-01
5	Almond	9	2	3	2020-01-01
6	Powrót	1	8	5	2080-06-01
7	Zaraza	6	5	8	2018-01-01
8	Później	5	3	10	2022-01-01
9	Biblioteka o północy	3	6	9	2021-01-01
10	Rozmowy z przyjaciółmi	9	9	2	2020-01-01
11	Zielona Milla	2	3	6	2021-01-01
12	Miasteczko Salem	1	3	8	2012-01-01
13	Ślady	6	10	10	2016-01-01
14	Nasza nerwowa planeta	4	6	3	2018-01-01
15	Chłopiec zwany Gwiazdką	9	6	3	2017-01-01

(15 wierszy)

Rys.4 - Tabela "Książka"

W tabeli "Wydawnictwo" są wypisane wydawnictwa produkujące książki. Tabela zawiera nazwę wydawnictwa oraz ID wydawnictwa - unikalny numer identyfikacyjny wydawnictwa w bazie danych biblioteki.

Kod tworzący tabelę:

```
CREATE TABLE wydawnictwo(  
ID_wydawnictwo serial primary key,  
nazwa_wydawnictwa varchar(40) not null);
```

id_wydawnictwo	nazwa_wydawnictwa
1	Nasza Księgarnia
2	Znak
3	CreateSpace
4	Wiedza i Praktyka
5	Czarne
6	Elipsa
7	Universitas
8	Wydawnictwo RM
9	Mag
10	E-bookowo

(10 wierszy)

Rys.5 - Tabela "Wydawnictwo"

Tabela "Autor" swoją strukturą przypomina powyższą tabelę i zawiera informacje imienia i nazwiska autora oraz ID autora - unikalny numer identyfikacyjny każdego autora w bazie danych biblioteki.

Kod tworzący tabelę:

```
CREATE TABLE autor(
ID_autor serial primary key,
imie varchar(40) not null,
nazwisko varchar(40) not null);
```

id_autor	imie	nazwisko
1	V. E.	Schwab
2	Won-Pyung	Sohn
3	Stephan	King
4	Lucinda	Rilley
5	Przemysław	Piotrkowski
6	Matt	Haig
7	Anna	Johns
8	Nicolas	Sparks
9	Sally	Rooney
10	Jakub	Małecki

(10 wierszy)

Rys.6 - Tabela "Autor"

Tabeli "Kategoria" służy do przechowywania listy możliwych kategorii książek. Składa się ona z nazwy kategorii oraz ID kategorii, który jest unikalnym numerem identyfikacyjnym każdej kategorii w bazie danych biblioteki.

Kod tworzący tabelę:

```
CREATE TABLE kategoria(
ID_kategoria serial primary key,
nazwa_kategorii varchar(40) not null);
```

id_kategoria	nazwa_kategorii
1	literatura obyczajowa
2	romans
3	fantasy
4	science fiction
5	horror
6	kryminał, sensacja, thriller
7	reportaż
8	artykuł naukowy
9	literatura młodzieżowa
10	klasyka

(10 wierszy)

Rys.7 - Tabela "Kategoria"

Ostatnią tabelą w bazie danych jest tabela "Wypożyczenie", która zamieszcza informacje wypożyczeń książek przez czytelników w danej bibliotece. Tabela składa się z kolumn z danymi: osoby wypożyczącej książkę, wypożyczonej książki, daty wypożyczenia i zwrotu oraz pracownika wypożyczającego książkę i odbierającego książkę przy jej zwrocie. Kluczem głównym tej tabeli jest ID wypożyczenia, natomiast dane kto\_wypożyczył, co\_wypożyczył, pracownik\_wypożyczenia i pracownik\_zwrotu są kluczami obcymi i odwołują się bezpośrednio do tabel "Czytelnik", "Książka" i "Pracownik".

Kod tworzący tabelę:

```
CREATE TABLE wypozyczenie(
ID_wypozyczenie serial primary key,
kto_wypozyczyl int not null
references czytelnik on
delete restrict on update restrict,
co_wypozyczyl int not null
references ksiazka on
delete restrict on update restrict,
data_wypozyczenia date,
data_zwrotu date,
pracownik_wypozyczenia int not null
references pracownik on
delete restrict on update restrict,
pracownik_zwrotu int not null
references pracownik on
delete restrict on update restrict);
```

id_wypozyczenie	kto_wypozyczył	co_wypozyczył	data_wypozyczenia	data_zwrotu	pracownik_wypozyczenia	pracownik_zwrotu
1	1	1	2022-01-03	2022-01-13	1	5
2	5	7	2022-03-11	2022-03-29	2	2
3	3	3	2022-02-23	2022-02-27	1	2
4	4	2	2022-01-17	2022-01-19	3	2
5	9	14	2022-01-28	2022-02-07	1	1
6	10	6	2022-02-04	2022-02-14	3	3
7	8	3	2022-03-05	2022-03-15	2	3
8	7	11	2022-04-01	2022-04-09	1	3
9	7	12	2022-04-02	2022-04-11	1	2
10	1	2	2022-04-02	2022-04-13	1	2
11	4	5	2022-04-12	2022-04-13	1	3
12	9	13	2022-04-07	2022-04-10	2	3
14	3	5	2022-01-03	2022-01-11	1	2
15	10	15	2022-02-15	2022-02-25	1	1
13	1	4	2022-04-03	2022-04-10	1	5
16	2	8	2022-03-22	2022-03-31	2	1
17	6	10	2022-03-31	2022-04-15	2	2
18	6	7	2022-04-02	2022-04-09	2	3
19	2	9	2022-03-05	2022-04-01	1	3
20	3	1	2022-02-28	2022-03-16	1	1

(20 wierszy)

Rys.8 - Tabela "Wypożyczenie"



---

## Etap II

W danym etapie do bazy danych Biblioteka stworzono:

- widok,
- funkcję,
- wyzwalacz (zdarzenie + zdefiniowane działanie).

Do każdego z wymienionych punktów dodano opis, kod, oraz przykład działania.

### 1. Widok

Widoki pomagają lepiej zrozumieć informację zawartą w bazie danych. Stworzono widok o nazwie *kto\_co\_wypozyczyl*, który służy do przedstawienia pełniejszego widoku na temat dokonanych wypożyczeń. W widoku zawarte są następujące dane: imię i nazwisko czytelnika, wypożyczającego książkę; nazwa i kategoria wypożyczonej książki; oraz data wypożyczenia i zwrotu. Taki widok można edytować na różne sposoby, aby uzyskać požądane informacje. Na przykład, można dodać takie kolumny, jak autor książki lub imię i nazwisko pracownika, który obsługiwał czytelnika podczas wypożyczenia/zwrotu.

Kod tworzący widok:

```
CREATE VIEW kto_co_wypozyczyl
AS SELECT c.imie, c.nazwisko,
k.tytul, g.nazwa_kategorii,
w.data_wypozyczenia, w.data_zwrotu FROM
czytelnik c, ksiazka k, kategoria g,
wypozyczenie w where WHERE
c.id_czytelnik = w.kto_wypozyczyl
and k.id_ksiazka = w.co_wypozyczyl
and g.id_kategoria = w.jaka_kategoria;
```

```
biblioteka=> select * from kto_co_wypozyczyl;
```

imie	nazwisko	tytul	nazwa_kategorii	data_wypozyczenia	data_zwrotu
Jan	Adamski	Zaginiona Siostra - Siedem Sióstr	literatura obyczajowa	2022-01-03	2022-01-13
Joanna	Gostomska	Zaraza	kryminał, sensacja, thriller	2022-03-11	2022-03-29
Marek	Górecki	Święto Ognia	literatura obyczajowa	2022-02-23	2022-02-27
Małgorzata	Kalinowska	Niewidzialne życie Addie LaRue	fantasy	2022-01-17	2022-01-19
Łukasz	Szumiński	Nasza nerwowa planeta	science fiction	2022-01-28	2022-02-07
Julia	Wilemska	Powrót	literatura obyczajowa	2022-02-04	2022-02-14
Adam	Jankowski	Święto Ognia	literatura obyczajowa	2022-03-05	2022-03-15
Weronika	Adamiak	Zielona Mila	romans	2022-04-01	2022-04-09
Weronika	Adamiak	Miasteczko Salem	literatura obyczajowa	2022-04-02	2022-04-11
Jan	Adamski	Niewidzialne życie Addie LaRue	fantasy	2022-04-02	2022-04-13
Małgorzata	Kalinowska	Almond	literatura młodzieżowa	2022-04-12	2022-04-13
Łukasz	Szumiński	Ślady	kryminał, sensacja, thriller	2022-04-07	2022-04-10
Marek	Górecki	Almond	literatura młodzieżowa	2022-01-03	2022-01-11
Julia	Wilemska	Chłopiec zwany Gwiazdką	literatura młodzieżowa	2022-02-15	2022-02-25
Jan	Adamski	Kobieta w białym kimonie	romans	2022-04-03	2022-04-10
Natalia	Lewandowska	Później	horror	2022-03-22	2022-03-31
Paweł	Rewig	Rozmowy z przyjaciółmi	literatura młodzieżowa	2022-03-31	2022-04-15
Paweł	Rewig	Zaraza	kryminał, sensacja, thriller	2022-04-02	2022-04-09
Natalia	Lewandowska	Biblioteka o północy	fantasy	2022-03-05	2022-04-01
Marek	Górecki	Zaginiona Siostra - Siedem Sióstr	literatura obyczajowa	2022-02-28	2022-03-16

(20 wierszy)

Rys.9 - Przykład działania widoku kto\_co\_wypozyczyl

## 2. Funkcja

Chcąc dowiedzieć się, ile każdy z czytelników ma aktualnie wypożyczonych książek, stworzono funkcję *ile\_wypozyczen(int)*, która przetwarza dane i wyświetla ilość wypożyczeń każdego z czytelników.

Kod tworzący funkcję:

```
CREATE FUNCTION ile_wypozyczen(int) RETURNS int
LANGUAGE plpgsql
AS '
declare
ilosc_wypozyczen float;
begin
select count(*) into ilosc_wypozyczen from wypozyczenie where
wypozyczenie.kto_wypozyczyl=$1;
return ilosc_wypozyczen;
end;';
```

```

biblioteka=> select imie, nazwisko, ile_wypozycczen(id_czytelnik) from czytelnik;
  imie      | nazwisko  | ile_wypozycczen
-----+-----+-----
Jan         | Adamski   | 3
Natalia     | Lewandowska | 2
Marek       | Górecki   | 3
Małgorzata  | Kalinowska | 2
Joanna      | Gostomska | 1
Paweł       | Rewig     | 2
Weronika    | Adamiak   | 2
Adam        | Jankowski | 1
Łukasz      | Szumiński | 2
Julia       | Wilemska  | 2
(10 wierszy)

```

Rys.10 - Przykład działania funkcji *ile\_wypozycczen*

Kolejna stworzona funkcja jest funkcja o nazwie *wypozycczenia\_za\_miesiac(date,date)*, dzięki której można dowiedzieć się, ile łącznie zostało dokonano wypożyczeń w konkretnym miesiącu. Funkcja przyjmuje *date* jako dane wejściowe, a na wyjściu podaje ilość dokonanych wypożyczeń w danym okresie.

Kod tworzący funkcję:

```

CREATE or replace FUNCTION
wypozycczenia_za_miesiac(date, date)
RETURNS int
LANGUAGE plpgsql
AS ' declare
ilosc_wypozycczen_mies float;
begin
select count(*) into ilosc_wypozycczen_mies
from wypozycczenie where
wypozycczenie.data_wypozycczenia >=$1
and wypozycczenie.data_wypozycczenia<=$2;
return ilosc_wypozycczen_mies;
end; ';

```

---

```
biblioteka=> select wypozyczenia_za_miesiac('2022-04-01', '2022-04-30');
wypozyczenia_za_miesiac
-----
7
(1 wiersz)
```

*Rys.10 - Przykład działania funkcji wypozyczenia\_za\_miesiac*

Porównując uzyskane wyniki na rys. 9 i rys.10 z tabelą *Wypożyczenia* (rys.8) można zaobserwować zgodność wyników, co skutkuje poprawnością działania stworzonych funkcji.

### 3. Wyzwalacz

Stworzony wyzwalacz ma za zadanie uniknięcie nadmiarowego wypożyczenia książek przez jednego czytelnika. Przy tworzeniu wyzwalacza skorzystano z funkcji *BEFORE INSERT ON*, która nie pozwala na wpisanie kolejnego wypożyczenia do bazy danych, jeśli nie jest spełniony warunek funkcji *czy\_uzytkownik\_moze()*.

Funkcja *czy\_uzytkownik\_moze()* jest zedytowaną wcześniej stworzoną funkcją *ile\_wypozycczen(int)*, przedstawioną w poprzednim podrozdziale. Nowo utworzona funkcja jest uzupełniona o następujący warunek: czytelnik nie może posiadać więcej niż 3 wypożyczone książki w tym samym czasie. W przypadku niespełnienia danego warunku wyzwalacz nie pozwoli na dodanie nowego wypożyczenia.

Kod tworzący funkcję do wyzwalacza:

```
CREATE or replace FUNCTION
czy_uzytkownik_moze() RETURNS TRIGGER
LANGUAGE plpgsql
AS '
DECLARE
ilosc_wypozycczen int;
BEGIN
```

---

```
SELECT count(*) into ilosc_wypozycczen
FROM wypozycczenie WHERE
wypozycczenie.kto_wypozycczyl=NEW.kto_wypozycczyl;
IF ilosc_wypozycczen >2 THEN raise notice
"Masz za duzo wypozycczonych ksiazek: %",
ilosc_wypozycczen;
RETURN NULL;
END IF;
RETURN NEW;
END; ';
```

Kod tworzący wyzwalacz:

```
CREATE TRIGGER
czy_uzytkownik_moze_wypozycczyc_ksiazke
BEFORE INSERT ON wypozycczenie FOR EACH
row EXECUTE PROCEDURE
czy_uzytkownik_moze();
```

```
biblioteka=> insert into wypozycczenie(id_wypozycczenie, kto_wypozycczyl, co_wypozycczyl, data_wypozycczenia
, data_zwrotu, pracownik_wypozycczenia, pracownik_zwrotu) values (21, 3, 10, '2022-04-25', '2022-05-02',
3, 2);
UWAGA: Masz za duzo wypozycczonych ksiazek: 3
INSERT 0 0
```

Rys.11 - Przykład działania wyzwalacza czy\_uzytkownik\_moze\_wypozycczyc\_ksiazke()

---

## Etap III - Finalna wersja projektu

W danym etapie do bazy danych zaprojektowano interfejs użytkownika w PHP, który pozwala na:

- wyświetlenie zawartości poszczególnych tabel (SELECT);
- dodawanie, usuwanie i edycję rekordów z wybranych tabel (INSERT, DELETE, UPDATE);
- wyświetlenie widoku;
- wywołanie funkcji;
- uruchomienie wyzwalacza.

### 1. Wyświetlenie zawartości tabel oraz zamieszczenie stworzonych i wykorzystanych skryptów .php

Aby wyświetlić zawartość poszczególnych tabel, do każdej tabeli napisano kod w języku PHP. Wyświetlają się dane pod linkiem <http://172.20.2.84/~jelkur/glowna.php>.

Trzeba pamiętać, aby ustawić jednorazowe hasło do bazy danych. Posłużono się poleceniem *ALTER USER jelkur WITH ENCRYPTED PASSWORD '123';*.

#### 1.1. Strona główna interfejsu

Stworzono stronę główną interfejsu bazy danych biblioteki, która pełni rolę pośrednika pomiędzy różnymi tabelami. Jest to strona, zamieszczająca podstawowe informacje na temat danej bazy danych oraz posiada przyciski, które przekierowują użytkownika do poszczególnych tabel bazy.

## Baza danych biblioteki

Dana baza danych zostanie zaprojektowana na potrzebę biblioteki.

Zawiera ona:

1. Tabele czytelników pracowników (i ich role) danej biblioteki.

Czytelnicy

Role

Pracownicy

2. Tabele książek znajdujących się w bibliotece.

Książki

3. Tabele z dodatkowymi informacjami: autorzy, wydawnictwa oraz kategorie książek.

Autorzy

Wydawnictwa

Kategorie

4. Tabelę wypożyczeń książek.

Wypożyczenia

5. Widok (połączenie kilku tabel), ułatwiający czytanie informacji na temat dokonanych wypożyczeń.

Widok

Rys.12 - Strona główna interfejsu */glowna.php*

Kod zawarty w skrypcie **glowna.php**:

```
<html>
<head>
<title>PostgreSQL i php</title>
</head>
<body>
<h1> Baza danych biblioteki</h1>
<?php
// tu umieszczamy kod skryptu
// Połączmy się z bazą danych i rozłączmy się z nią
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie moge
polaczyc sie z baza danych !");
// tu powinno byc polaczenie
```

---

```

echo "Dana baza danych zostanie zaprojektowana na potrzebę biblioteki. ";
echo "<td><form action=czytelnik.php
method=POST> <br /> Zawiera ona: <br/> <br/> 1. Tabele czytelników pracowników (i
ich role) danej biblioteki. <br />
<input type=submit name=usun value=Czytelnicy></form></td>";

echo "<td><form action=rola.php
method=POST>
<input type=submit name=usun value=Role pracowników></form></td>";

echo "<td><form action=pracownik.php
method=POST>
<input type=submit name=usun value=Pracownicy></form></td>";

echo "<td><form action=ksiazka.php
method=POST> <br /> 2. Tabelę książek znajdujących się w bibliotece. <br />
<input type=submit name=usun value=Książki></form></td>";

echo "<td><form action=autor.php
method=POST> <br /> 3. Tabele z dodatkowymi informacjami: autorzy, wydawnictwa
oraz kategorie książek. <br />
<input type=submit name=usun value=Autorzy></form></td>";

echo "<td><form action=wydawnictwo.php
method=POST>
<input type=submit name=usun value=Wydawnictwa></form></td>";

echo "<td><form action=kategoria.php
method=POST>
<input type=submit name=usun value=Kategorie></form></td>";

echo "<td><form action=wypozyczenie.php
method=POST> <br /> 4. Tabelę wypożyczeń książek. <br />
<input type=submit name=usun value=Wypożyczenia></form></td>";

echo "<td><form action=widok.php
method=POST> <br /> 5. Widok (połączenie kilku tabel), ułatwiający czytanie informacji
na temat dokonanych wypożyczeń. <br />
<input type=submit name=usun value=Widok></form></td>";

pg_close($dbh);
?>
</body>
</html>

```

---



## 1.2. Interfejs tabeli "Czytelnik"

### Tabela "Czytelnik"

id_czytelnik	imie	nazwisko	email	telefon	data_urodzenia	Usun	Edytuj
1	Jan	Adamski	jan.adamski@gmail.com	765123456	2000-01-04	Usun	Edytuj
2	Natalia	Lewandowska	natalia.lewandowska@gmail.com	765125776	2000-11-14	Usun	Edytuj
3	Marek	Górecki	marek.gorecki@gmail.com	745165076	1999-08-22	Usun	Edytuj
4	Małgorzata	Kalinowska	malgorzata.kalinowska@gmail.com	741155096	2001-09-01	Usun	Edytuj
5	Joanna	Gostomska	joanna.gostomska@gmail.com	712386932	2001-03-30	Usun	Edytuj
6	Paweł	Rewig	pawel.rewig@gmail.com	707847212	2000-07-15	Usun	Edytuj
7	Weronika	Adamiak	weronika.adamiak@gmail.com	722956372	2000-01-29	Usun	Edytuj
9	Łukasz	Szumiński	lukasz.szuminski@gmail.com	701236548	2000-04-11	Usun	Edytuj
10	Julia	Wilemska	julia.wilemska@gmail.com	765496032	2002-05-21	Usun	Edytuj
8	Adam	Jankowski	adam.jankowski@gmail.com	722098764	1998-12-25	Usun	Edytuj

Dodaj nową osobę do bazy

Imię:  Nazwisko:  Email:  Telefon:

Data urodzenia:

Rys.13 - Interfejs z danymi czytelników /czytelnik.php

Kod zawarty w skrypcie **czytelnik.php**:

```
<html>
<head>
<title>PostgreSQL i php</title>
</head>
<body>
<h1> Tabela "Czytelnik" </h1>
<?php
// tu umieszczamy kod skryptu
// Połączmy się z bazą danych i rozłączmy się z nią natychmiast
$dbh = pg_connect("dbname=biblioteka user=jelkur
```

---

```

password=123 host=localhost") or die("Nie moze
polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
// wyswietlmy dane
$query = "Select * from czytelnik";
$wynik = pg_query($dbh,$query);
// odbierzmy rozmiary tabeli:
$liczba_kolumn = pg_num_fields($wynik);
$liczba_wierszy = pg_num_rows($wynik);
/// teraz wyswietlmy dane
// teraz wyswietlmy dane
echo "<TABLE border width=1>";
echo "<TABLE border width=1>";
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_field_name($wynik,$k);
echo "</TD>"; //echo "\t";
}
echo "<td>Usun</td><td>Edytuj</td></TR>";
for($w =0;$w<$liczba_wierszy;$w++)
{
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_fetch_result($wynik,$w,$k);
echo "</TD>"; //echo "\t";
}
$idos = pg_fetch_result($wynik,$w,$liczba_kolumn-6);
$imie = pg_fetch_result($wynik,$w,$liczba_kolumn-5);
$nazwisko = pg_fetch_result($wynik,$w,$liczba_kolumn-4);
$email = pg_fetch_result($wynik,$w,$liczba_kolumn-3);
$telefon = pg_fetch_result($wynik,$w,$liczba_kolumn-2);
$data_urodzenia = pg_fetch_result($wynik,$w,$liczba_kolumn-1);
// tu dodajemy formularz do usuwania osoby
echo "<td><form action=usun.php
method=POST>
<input type=hidden name=idos value=$idos>
<input type=submit name=usun value=Usun></form></td>";
// tu dodajemy formularz do edycji osoby
echo "<td><form action=edytuj.php
method=POST>
<input type=hidden name=idos value=$idos>
<input type=hidden name=imie value=$imie>
<input type=hidden name=nazwisko value=$nazwisko>

```

---

```

<input type=hidden name=email value=$email>
<input type=hidden name=telefon value=$telefon>
<input type=hidden name=data_urodzenia value=$data_urodzenia>
<input type=submit name=edytuj value=Edytuj></form></td>";
echo "</TR>"; //echo "<br />";
}
echo "</TABLE>";
// Tu dopiszmy nową osobę
echo "<FORM action=dodaj.php method=POST> <br />
Dodaj nową osobę do bazy <br /> Imię:
<input type=text name=imie> Nazwisko:
<input type=text name=nazwisko> Email:
<input type=text name=email> Telefon:
<input type=text name=telefon> Data urodzenia:
<input type=text name=data_urodzenia> <br />
<input type=submit name=Dodaj
value=Dodaj> </form>";
//echo "Jest Polaczenie z baza danych ! ";
echo "<td><form action=glozna.php
method=POST> <br/>
<input type=submit name=usun value=Wróć></form></td>";
pg_close($dbh);
?>
</body>
</html>

```

### 1.3. Interfejs tabeli "Pracownik"

#### Tabela "Pracownik"

id_pracownik	imie	nazwisko	email	telefon	data_urodzenia	rola_pracownika
1	Agata	Wilemska	agata.wilemska@gmail.com	776782931	2000-01-04	1
2	Justyna	Nowak	justyna.nowak@gmail.com	712378567	1995-06-19	1
3	Katarzyna	Wasilewska	katarzyna.wasilewska@gmail.com	734562980	1999-05-01	2
5	Jakub	Tomaszewski	jakub.tomaszewski@gmail.com	643242980	1997-11-24	4
4	Natalia	Ziółkowska	natalia.ziolkowska@gmail.com	782516382	1990-08-17	3

Wróć

Rys.14 - Interfejs z danymi pracowników /pracownik.php

Kod zawarty w skrypcie **pracownik.php**:

---

```

<html>
<head>
<title>PostgreSQL i php</title>
</head>
<body>
<h1> Tabela "Pracownik" </h1>
<?php
// tu umieszczamy kod skryptu
// Połączmy się z bazą danych i rozłączmy się z nią natychmiast
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie moge
polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
// wyświetlmy dane
$query = "Select * from pracownik";
$wynik = pg_query($dbh,$query);
// odbierzmy rozmiary tabeli:
$liczba_kolumn = pg_num_fields($wynik);
$liczba_wierszy = pg_num_rows($wynik);
/// teraz wyświetlmy dane
// teraz wyświetlmy dane
echo "<TABLE border width=1>";
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_field_name($wynik,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>";
for($w =0;$w<$liczba_wierszy;$w++)
{
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_fetch_result($wynik,$w,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>"; //echo "<br />";
}
echo "</TABLE>";
//echo "Jest Polaczenie z baza danych ! ";
echo "<td><form action=glowna.php
method=POST> <br/>
<input type=submit name=usun value=Wróć></form></td>";
pg_close($dbh);

```

---

---

```
?>
</body>
</html>
```

#### 1.4. Interfejs tabeli "Rola"

Tabela "Rola"	
id_rola	nazwa_rola
1	bibliotekarz
2	księgowy
3	dyrektor
4	pomocnik bibliotekarza
5	czasowe zastępstwo

Wróć

Rys.15 - Interfejs z danymi ról pracowników /rola.php

Kod zawarty w skrypcie **rola.php**:

```
<html>
<head>
<title>PostgreSQL i php</title>
</head>
<body>
<h1> Tabela "Rola" </h1>
<?php
// tu umieszczamy kod skryptu
// Połączmy się z bazą danych i rozłączmy się z nią natychmiast
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie moge
polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
// wyświetlmy dane
$query = "Select * from rola";
$wynik = pg_query($dbh,$query);
// odbierzmy rozmiary tabeli:
```

---

```
$liczba_kolumn = pg_num_fields($wynik);
$liczba_wierszy = pg_num_rows($wynik);
/// teraz wyświetlmy dane
// teraz wyświetlmy dane
echo "<TABLE border width=1>";
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_field_name($wynik,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>";
for($w =0;$w<$liczba_wierszy;$w++)
{
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_fetch_result($wynik,$w,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>"; //echo "<br />";
}
echo "</TABLE>";
//echo "Jest Polaczenie z baza danych ! ";
echo "<td><form action=glowna.php
method=POST> <br/>
<input type=submit name=usun value=Wróć></form></td>";
pg_close($dbh);
?>
</body>
</html>
```

## 1.5. Interfejs tabeli "Książka"

Tabela "Książka"					
id_książka	tytuł	jaka_kategoria	kto_autor	jakie_wydawnictwo	rok_wydania
1	Zaginiona Siostra - Siedem Sióstr	1	4	3	2018-01-01
2	Niewidzialne życie Addie LaRue	3	1	7	2020-06-01
3	Święto Ognia	1	10	4	2022-01-01
4	Kobieta w białym kimonie	2	7	1	2021-06-01
5	Almond	9	2	3	2020-01-01
6	Powrót	1	8	5	2080-06-01
7	Zaraza	6	5	8	2018-01-01
8	Później	5	3	10	2022-01-01
9	Biblioteka o północy	3	6	9	2021-01-01
10	Rozmowy z przyjaciółmi	9	9	2	2020-01-01
11	Zielona Miła	2	3	6	2021-01-01
12	Miasteczko Salem	1	3	8	2012-01-01
13	Ślady	6	10	10	2016-01-01
14	Nasza nerwowa planeta	4	6	3	2018-01-01
15	Chłopiec zwany Gwiazdką	9	6	3	2017-01-01

[Wróć](#)

Rys.16 - Interfejs z danymi książek */ksiazka.php*

Kod zawarty w skrypcie **ksiazka.php**:

```
<html>
<head>
<title>PostgreSQL i php</title>
</head>
```

---

```

<body>
<h1> Tabela "Książka" </h1>
<?php
// tu umieszczamy kod skryptu
// Połączmy się z bazą danych i rozłączmy się z nią natychmiast
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie moge
polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
// wyświetlmy dane
$query = "Select * from ksiazka";
$wynik = pg_query($dbh,$query);
// odbierzmy rozmiary tabeli:
$liczba_kolumn = pg_num_fields($wynik);
$liczba_wierszy = pg_num_rows($wynik);
/// teraz wyświetlmy dane
// teraz wyświetlmy dane
echo "<TABLE border width=1>";
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_field_name($wynik,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>";
for($w =0;$w<$liczba_wierszy;$w++)
{
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_fetch_result($wynik,$w,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>"; //echo "<br />";
}
echo "</TABLE>";
//echo "Jest Polaczenie z baza danych ! ";
echo "<td><form action=glowna.php
method=POST> <br/>
<input type=submit name=usun value=Wróć></form></td>";
pg_close($dbh);
?>
</body>
</html>

```

---



## 1.6. Interfejs tabeli "Autor"

id_autor	imie	nazwisko
1	V. E.	Schwab
2	Won-Pyung	Sohn
3	Stephan	King
4	Lucinda	Rilley
5	Przemysław	Piotrkowski
6	Matt	Haig
7	Anna	Johns
8	Nicolas	Sparks
9	Sally	Rooney
10	Jakub	Małecki

[Wróć](#)

Rys.17 - Interfejs z danymi autorów /*autor.php*

Kod zawarty w skrypcie **autor.php**:

```
<html>
<head>
<title>PostgreSQL i php</title>
</head>
<body>
<h1>Tabela "Autor" </h1>
<?php
// tu umieszczamy kod skryptu
// Połączmy się z bazą danych i rozłączmy się z nią natychmiast
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie moge
polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
// wyświetlmy dane
$query = "Select * from autor";
$wynik = pg_query($dbh,$query);
// odbierzmy rozmiary tabeli:
$liczba_kolumn = pg_num_fields($wynik);
$liczba_wierszy = pg_num_rows($wynik);
```

---

```
/// teraz wyświetlmy dane
// teraz wyświetlmy dane
echo "<TABLE border width=1>";
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_field_name($wynik,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>";
for($w =0;$w<$liczba_wierszy;$w++)
{
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_fetch_result($wynik,$w,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>"; //echo "<br />";
}
echo "</TABLE>";
//echo "Jest Polaczenie z baza danych ! ";
echo "<td><form action=glowna.php
method=POST> <br/>
<input type=submit name=usun value=Wróć></form></td>";
pg_close($dbh);
?>
</body>
</html>
```

## 1.7. Interfejs tabeli "Wydawnictwo"

Tabela "Wydawnictwo"	
id_wydawnictwo	nazwa_wydawnictwa
1	Nasza Księgarnia
2	Znak
3	CreateSpace
4	Wiedza i Praktyka
5	Czarne
6	Elipsa
7	Universitas
8	Wydawnictwo RM
9	Mag
10	E-bookowo

[Wróć](#)

Rys.18 - Interfejs z danymi wydawnictw /wydawnictwo.php

Kod zawarty w skrypcie **wydawnictwo.php**:

```
<html>
<head>
<title>PostgreSQL i php</title>
</head>
<body>
<h1> Tabela "Wydawnictwo" </h1>
<?php
// tu umieszczamy kod skryptu
// Połączmy się z bazą danych i rozłączmy się z nią natychmiast
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie moge
polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
// wyświetlmy dane
$query = "Select * from wydawnictwo";
$wynik = pg_query($dbh,$query);
// odbierzmy rozmiary tabeli:
$liczba_kolumn = pg_num_fields($wynik);
$liczba_wierszy = pg_num_rows($wynik);
```

---

```
/// teraz wyświetlmy dane
// teraz wyświetlmy dane
echo "<TABLE border width=1>";
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_field_name($wynik,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>";
for($w =0;$w<$liczba_wierszy;$w++)
{
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_fetch_result($wynik,$w,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>"; //echo "<br />";
}
echo "</TABLE>";
//echo "Jest Polaczenie z baza danych ! ";
echo "<td><form action=glowna.php
method=POST> <br/>
<input type=submit name=usun value=Wróć></form></td>";
pg_close($dbh);
?>
</body>
</html>
```

## 1.8. Interfejs tabeli "Kategoria"

id_kategoria	nazwa_kategorii
1	literatura obyczajowa
2	romans
3	fantasy
4	science fiction
5	horror
6	kryminal, sensacja, thriller
7	reportaż
8	artykuł naukowy
9	literatura młodzieżowa
10	klasyka

[Wróć](#)

Rys.19 - Interfejs z danymi kategorii */kategoria.php*

Kod zawarty w skrypcie **kategoria.php**:

```
<html>
<head>
<title>PostgreSQL i php</title>
</head>
<body>
<h1> Tabela "Kategoria" </h1>
<?php
// tu umieszczamy kod skryptu
// Połączmy się z bazą danych i rozłączmy się z nią natychmiast
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie moge
polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
// wyświetlmy dane
$query = "Select * from kategoria";
$wynik = pg_query($dbh,$query);
```

---

```
// odbierzmy rozmiary tabeli:
$liczba_kolumn = pg_num_fields($wynik);
$liczba_wierszy = pg_num_rows($wynik);
/// teraz wyświetlmy dane
// teraz wyświetlmy dane
echo "<TABLE border width=1>";
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_field_name($wynik,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>";
for($w =0;$w<$liczba_wierszy;$w++)
{
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_fetch_result($wynik,$w,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>"; //echo "<br />";
}
echo "</TABLE>";
//echo "Jest Polaczenie z baza danych ! ";
echo "<td><form action=glowna.php
method=POST> <br/>
<input type=submit name=usun value=Wróć></form></td>";
pg_close($dbh);
?>
</body>
</html>
```

## 1.9. Interfejs tabeli "Wypożyczenie"

**Tabela "Wypożyczenie"**

id_wypozyczenie	kto_wypozyczył	co_wypozyczył	data_wypozyczenia	data_zwrotu	pracownik_wypozyczenia	pracownik_zwrotu	Usun	Edytuj
1	1	1	2022-01-03	2022-01-13	1	5	Usun	Edytuj
2	5	7	2022-03-11	2022-03-29	2	2	Usun	Edytuj
3	3	3	2022-02-23	2022-02-27	1	2	Usun	Edytuj
4	4	2	2022-01-17	2022-01-19	3	2	Usun	Edytuj
5	9	14	2022-01-28	2022-02-07	1	1	Usun	Edytuj
6	10	6	2022-02-04	2022-02-14	3	3	Usun	Edytuj
7	8	3	2022-03-05	2022-03-15	2	3	Usun	Edytuj
8	7	11	2022-04-01	2022-04-09	1	3	Usun	Edytuj
10	1	2	2022-04-02	2022-04-13	1	2	Usun	Edytuj
11	4	5	2022-04-12	2022-04-13	1	3	Usun	Edytuj
12	9	13	2022-04-07	2022-04-10	2	3	Usun	Edytuj
14	3	5	2022-01-03	2022-01-11	1	2	Usun	Edytuj
15	10	15	2022-02-15	2022-02-25	1	1	Usun	Edytuj
16	2	8	2022-03-22	2022-03-31	2	1	Usun	Edytuj
17	6	10	2022-03-31	2022-04-15	2	2	Usun	Edytuj
18	6	7	2022-04-02	2022-04-09	2	3	Usun	Edytuj
19	2	9	2022-03-05	2022-04-01	1	3	Usun	Edytuj
20	3	1	2022-02-28	2022-03-16	1	1	Usun	Edytuj
13	1	4	2022-04-08	2022-04-10	1	5	Usun	Edytuj
32	7	12	2022-04-02	2022-04-11	1	2	Usun	Edytuj
35	7	1	2022-06-05		2	1	Usun	Edytuj
27	2	4	2022-06-02	2022-06-03	2	1	Usun	Edytuj

Dodaj nowe wypożyczenie bazy

Kto wypożyczył:  Co wypożyczył:  Data wypożyczenia:  Pracownik przyjmujący wypożyczenie:

Ile dokonano wypożyczeń za podany okres? Wpisz XXXX-XX-XX

Od:  Do:

Chcesz się dowiedzieć, ile wypożyczeń ma konkretny czytelnik? Wpisz jego indeks:

Indeks czytelnika:

*Rys.20 - Interfejs z danymi wypożyczeń /wypozyczenie.php*

---

Kod zawarty w skrypcie **wypozyczenie.php**:

```
<html>
<head>
<title>PostgreSQL i php</title>
</head>
<body>
<h1> Tabela "Wypożyczenie" </h1>
<?php
// tu umieszczamy kod skryptu
// Połączmy się z bazą danych i rozłączmy się z nią natychmiast
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie moge
polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
// wyświetlmy dane
$query = "Select * from wypozyczenie";

$wynik = pg_query($dbh,$query);
// odbierzmy rozmiary tabeli:
$liczba_kolumn = pg_num_fields($wynik);
$liczba_wierszy = pg_num_rows($wynik);
/// teraz wyświetlmy dane
// teraz wyświetlmy dane
echo "<TABLE border width=1>";
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_field_name($wynik,$k);
echo "</TD>"; //echo "\t";
}
echo "<td>Usun</td><td>Edytuj</td></TR>";
for($w =0;$w<$liczba_wierszy;$w++)
{
echo "<TR>";
for($k =0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_fetch_result($wynik,$w,$k);
echo "</TD>"; //echo "\t";
}
$id_wypozyczenie = pg_fetch_result($wynik,$w,$liczba_kolumn-7);
$kto_wypozyczył = pg_fetch_result($wynik,$w,$liczba_kolumn-6);
$co_wypozyczył = pg_fetch_result($wynik,$w,$liczba_kolumn-5);
$data_wypozyczenia = pg_fetch_result($wynik,$w,$liczba_kolumn-4);
$data_zwrotu = pg_fetch_result($wynik,$w,$liczba_kolumn-3);
```



---

```

$pracownik_wypozyczenia = pg_fetch_result($wynik,$w,$liczba_kolumn-2);
$pracownik_zwrotu = pg_fetch_result($wynik,$w,$liczba_kolumn-1);
// tu dodajemy formularz do usuwania osoby
echo "<td><form action=usun_wypozyczenie.php
method=POST>
<input type=hidden name=id_wypozyczenie value=$id_wypozyczenie>
<input type=submit name=usun value=Usun></form></td>";
// tu dodajemy formularz do edycji osoby
echo "<td><form action=edytuj_wypozyczenie.php
method=POST>
<input type=hidden name=id_wypozyczenie value=$id_wypozyczenie>
<input type=hidden name=kto_wypozyczył value=$kto_wypozyczył>
<input type=hidden name=co_wypozyczył value=$co_wypozyczył>
<input type=hidden name=data_wypozyczenia value=$data_wypozyczenia>
<input type=hidden name=data_zwrotu value=$data_zwrotu>
<input type=hidden name=pracownik_wypozyczenia value=$pracownik_wypozyczenia>
<input type=hidden name=pracownik_zwrotu value=$pracownik_zwrotu>
<input type=submit name=edytuj value=Edytuj></form></td>";

echo "</TR>"; //echo "<br />";
}
echo "</TABLE>";

// Tu dopiszmy nową osobę
echo "<FORM action=dodaj_wypozyczenie.php method=POST> <br />
Dodaj nowe wypożyczenie bazy <br /> Kto wypożyczył:
<input type=text name=kto_wypozyczył> Co wypożyczył:
<input type=text name=co_wypozyczył> Data wypożyczenia:
<input type=text name=data_wypozyczenia> Pracownik przyjmujący wypożyczenie:
<input type=text name=pracownik_wypozyczenia> <br />
<input type=submit name=Dodaj
value=Dodaj> </form>";

// Tu dowiemy się ile wypożyczeń za miesiąc
echo "<FORM action=wypozyczenia_za_miesiac.php method=POST> <br />
Ile dokonano wypożyczeń za podany okres? Wpisz XXXX-XX-XX <br /> Od:
<input type=text name=okres_od> Do:
<input type=text name=okres_do> <br />
<input type=submit name=sprawdz
value=Sprawdź> </form>";

// Tu dowiemy się ile wypożyczeń ma każda osoba
echo "<FORM action=ile_wypozyczyen.php method=POST> <br />
Chcesz się dowiedzieć, ile wypożyczeń ma konkretny czytelnik? Wpisz jego indeks: <br />
Indeks czytelnika:
<input type=text name=id_czytelnik>
<input type=submit name=sprawdz
value=Sprawdź> </form>";

```

---

---

```
//echo "Jest Polaczenie z baza danych ! ";  
echo "<td><form action=glowna.php  
method=POST> <br/>  
<input type=submit name=usun value=Wróć></form></td>";  
pg_close($dbh);  
?>  
</body>  
</html>
```

## 1.10. Interfejs tabeli "Widok"

### Widok "Kto co wypożyczył"

imie	nazwisko	tytul	nazwa_kategorii	data_wypozyczenia	data_zwrotu
Jan	Adamski	Zaginiona Siostra - Siedem Sióstr	literatura obyczajowa	2022-01-03	2022-01-13
Joanna	Gostomska	Zaraza	kryminał, sensacja, thriller	2022-03-11	2022-03-29
Marek	Górecki	Święto Ognia	literatura obyczajowa	2022-02-23	2022-02-27
Małgorzata	Kalinowska	Niewidzialne życie Addie LaRue	fantasy	2022-01-17	2022-01-19
Lukasz	Szumiński	Nasza nerwowa planeta	science fiction	2022-01-28	2022-02-07
Julia	Wilemska	Powrót	literatura obyczajowa	2022-02-04	2022-02-14
Adam	Jankowski	Święto Ognia	literatura obyczajowa	2022-03-05	2022-03-15
Weronika	Adamiak	Zielona Mila	romans	2022-04-01	2022-04-09
Jan	Adamski	Niewidzialne życie Addie LaRue	fantasy	2022-04-02	2022-04-13
Małgorzata	Kalinowska	Almond	literatura młodzieżowa	2022-04-12	2022-04-13
Lukasz	Szumiński	Ślady	kryminał, sensacja, thriller	2022-04-07	2022-04-10
Marek	Górecki	Almond	literatura młodzieżowa	2022-01-03	2022-01-11
Julia	Wilemska	Chłopiec zwany Gwiazdką	literatura młodzieżowa	2022-02-15	2022-02-25
Natalia	Lewandowska	Później	horror	2022-03-22	2022-03-31
Paweł	Rewig	Rozmowy z przyjaciółmi	literatura młodzieżowa	2022-03-31	2022-04-15
Paweł	Rewig	Zaraza	kryminał, sensacja, thriller	2022-04-02	2022-04-09
Natalia	Lewandowska	Biblioteka o północy	fantasy	2022-03-05	2022-04-01
Marek	Górecki	Zaginiona Siostra - Siedem Sióstr	literatura obyczajowa	2022-02-28	2022-03-16
Jan	Adamski	Kobieta w białym kimonie	romans	2022-04-08	2022-04-10
Weronika	Adamiak	Miasteczko Salem	literatura obyczajowa	2022-04-02	2022-04-11
Weronika	Adamiak	Zaginiona Siostra - Siedem Sióstr	literatura obyczajowa	2022-06-05	
Natalia	Lewandowska	Kobieta w białym kimonie	romans	2022-06-02	2022-06-03

Wróć

Rys.21 - Interfejs z danymi widoku /widok.php

---

Kod zawarty w skrypcie **widok.php**:

```
<html>
<head>
<title>PostgreSQL i php</title>
</head>
<body>
<h1> Widok "Kto co wypożyczył" </h1>
<?php
// tu umieszczamy kod skryptu
// Połączmy się z bazą danych i rozłączmy się z nią natychmiast
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie moge
polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
// wyświetlmy dane
$query = "Select * from kto_co_wypożyczyl";
$wynik = pg_query($dbh,$query);
// odbierzmy rozmiary tabeli:
$liczba_kolumn = pg_num_fields($wynik);
$liczba_wierszy = pg_num_rows($wynik);
/// teraz wyświetlmy dane
// teraz wyświetlmy dane
echo "<TABLE border width=1>";
echo "<TR>";
for($k=0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_field_name($wynik,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>";
for($w=0;$w<$liczba_wierszy;$w++)
{
echo "<TR>";
for($k=0;$k<$liczba_kolumn;$k++)
{
echo "<TD>";
echo pg_fetch_result($wynik,$w,$k);
echo "</TD>"; //echo "\t";
}
echo "</TR>"; //echo "<br />";
}
echo "</TABLE>";
//echo "Jest Polaczenie z baza danych ! ";
echo "<td><form action=glowna.php
method=POST> <br/>
```

---

```
<input type=submit name=usun value=Wróć></form></td>";  
pg_close($dbh);  
>  
</body>  
</html>
```

### 1.11. Skrypty wykorzystane w skrypcie *czytelnik.php*

Kod zawarty w skrypcie **dodaj.php** (do dodania nowych czytelników):

```
<html>  
<head>  
<title>Dodaj osobę</title>  
</head>  
<body>  
<h1> Dodaję osobę </h1>  
<?php  
// tu umieszczamy kod skryptu  
//po pierwsze - odbierzmy parametry dla skryptu przekazane  
//metoda //POST - użyjemy je w zapytaniu poniżej  
$imie = $_POST['imie'];  
$nazwisko = $_POST['nazwisko'];  
$email = $_POST['email'];  
$telefon = $_POST['telefon'];  
$data_urodzenia = $_POST['data_urodzenia'];  
// Połączmy się z bazą danych  
$dbh = pg_connect("dbname=biblioteka user=jelkur  
password=123 host=localhost") or die("Nie  
moge polaczyc sie z baza danych ! ");  
// tu powinno byc polaczenie  
// wykonajmy zapytanie  
$query = "INSERT INTO czytelnik(imie,nazwisko,email,telefon,data_urodzenia) VALUES  
('$imie','$nazwisko','$email','$telefon','$data_urodzenia')";  
$wynik = pg_query($query);  
// sprawdzmy ile wierszy podmieniono  
$lz = pg_affected_rows($wynik);  
echo " Dodano $lz osob <br /> \n";  
pg_close($dbh);  
// zapewnijmy powrot do strony poprzedniej  
echo "<form action=czytelnik.php method=post>  
<input type=submit name=Ok value=OK>  
</form>";  
>  
</body>  
</html>
```

---

Kod zawarty w skrypcie **edytuj.php** (do edycji czytelników):

```
<html>
<head>
<title>Edytuj osobę</title>
</head>
<body>
<h1> Edytuj osobę </h1>
<?php
// tu umieszczamy kod skryptu
//po pierwsze – odbierzmy parametry dla skryptu przekazane
//metoda //POST – użyjemy je w zapytaniu poniżej
$id = $_POST['idos'];
$imie = $_POST['imie'];
$nazwisko = $_POST['nazwisko'];
$email = $_POST['email'];
$telefon = $_POST['telefon'];
$data_urodzenia = $_POST['data_urodzenia'];
$zm=$_POST['zmien'];
// Połączmy się z bazą danych
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie
moge polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie – najpierw wyświetlmy dane
$query = "SELECT imie,nazwisko,email,telefon,data_urodzenia,id_czytelnik from czytelnik
where id_czytelnik=$id";
$wynik = pg_query($query);
echo "<form action=edytuj.php method=post>";
echo "<table border width=1>";
$lk = pg_num_fields($wynik);
for($k = 0;$k<$lk -1; $k++)
{
echo "<tr>";
echo "<td>";
echo pg_field_name($wynik,$k);
echo "</td>";
echo "<td>";
$f_name = pg_field_name($wynik, $k);
$f_val = pg_fetch_result($wynik,0,$k);
echo "<input type=text name='$f_name' value='$f_val'>";
echo "</td>";
echo "</tr>";
}
echo "</table>";
echo "<input type=hidden name=idos value=$id>";
echo "<input type=submit name=zmien value=zmien>";
```

---

```

echo "</form>";
if ($zm)
{
    $query = "UPDATE czytelnik SET imie='$imie', nazwisko='$nazwisko', email='$email',
telefon='$telefon', data_urodzenia='$data_urodzenia' where id_czytelnik=$id";
    $wynik = pg_query($query);
    // sprawdzmy ile wierszy podmieniono
    $lz = pg_affected_rows($wynik);
    echo " Edytowano $lz osob <br /> \n";
}
pg_close($dbh);
// zapewnijmy powrot do strony poprzedniej
echo "<form action=czytelnik.php method=post>
<input type=submit name=Ok value=OK>
</form>";
?>
</body>
</html>

```

Kod zawarty w skrypcie **usun.php** (do usuwania czytelników):

```

<html>
<head>
<title>Usuń osobę</title>
</head>
<body>
<h1> Usuwam osobę </h1>
<?php
// tu umieszczamy kod skryptu
//po pierwsze – odbierzmy parametry dla skryptu przekazane
//metodą //POST – użyjemy je w zapytaniu poniżej
$id = $_POST['idos'];
// Połączmy się z bazą danych
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie
moge polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
$query = "DELETE FROM czytelnik where id_czytelnik= '$id'";
$wynik = pg_query($query);
// sprawdzmy ile wierszy podmieniono
$lz = pg_affected_rows($wynik);
echo " Usunieto $lz osobe <br />\n";
pg_close($dbh);
// zapewnijmy powrot do strony poprzedniej
echo "<form action=czytelnik.php method=post>
<input type=submit name=Ok value=OK>

```

---

```
</form>";  
?>  
</body>  
</html>
```

## 1.12. Skrypty wykorzystane w skrypcie *wypozyczenie.php*

Kod zawarty w skrypcie ***dodaj\_wypozyczenie.php*** (do dodania nowych wypożyczeń, aktywacja wyzwalacza):

```
<html>  
<head>  
<title>Dodaj wypozyczenie</title>  
</head>  
<body>  
<h1> Dodaję wypozyczenie </h1>  
<?php  
// tu umieszczamy kod skryptu  
//po pierwsze – odbierzmy parametry dla skryptu przekazane  
//metoda //POST – użyjemy je w zapytaniu poniżej  
$kto_wypozyczył = $_POST['kto_wypozyczył'];  
$co_wypozyczył = $_POST['co_wypozyczył'];  
$data_wypozyczenia = $_POST['data_wypozyczenia'];  
$pracownik_wypozyczenia = $_POST['pracownik_wypozyczenia'];  
// Połączmy się z bazą danych  
$dbh = pg_connect("dbname=biblioteka user=jelkur  
password=123 host=localhost") or die("Nie  
moge polaczyc sie z baza danych ! ");  
// tu powinno byc polaczenie  
// wykonajmy zapytanie  
$query = "INSERT INTO  
wypozyczenie(kto_wypozyczył,co_wypozyczył,data_wypozyczenia,pracownik_wypozyczenia,praco  
wnik_zwrotu) VALUES  
('$kto_wypozyczył','$co_wypozyczył','$data_wypozyczenia','$pracownik_wypozyczenia','1')"  
;  
$wynik = pg_query($query);  
// sprawdzmy ile wierszy podmieniono  
$lz = 0;  
$lz = pg_affected_rows($wynik);  
echo " Dodano $lz wypozyczenie. <br /> \n";  
echo " *Jeśli liczba wypożyczeń równa 0, oznacza to że czytelnik już ma 3 wypożyczone  
książki.\n";  
pg_close($dbh);  
// zapewnijmy powrot do strony poprzedniej  
echo "<form action=wypozyczenie.php method=post>  
<input type=submit name=Ok value=OK>
```



---

```
</form>";  
?>  
</body>  
</html>
```

Kod zawarty w skrypcie **edytuj\_wypozyczenie.php** (do edycji wypożyczeń):

```
<html>  
<head>  
<title>Edytuj wypozyczenie</title>  
</head>  
<body>  
<h1> Edytuj wypozyczenie </h1>  
<?php  
// tu umieszczamy kod skryptu  
//po pierwsze - odbierzmy parametry dla skryptu przekazane  
//metoda //POST - użyjemy je w zapytaniu ponizej  
$id_wypozyczenie = $_POST['id_wypozyczenie'];  
$kto_wypozyczył = $_POST['kto_wypozyczył'];  
$co_wypozyczył = $_POST['co_wypozyczył'];  
$data_wypozyczenia = $_POST['data_wypozyczenia'];  
$data_zwrotu = $_POST['data_zwrotu'];  
$pracownik_wypozyczenia = $_POST['pracownik_wypozyczenia'];  
$pracownik_zwrotu = $_POST['pracownik_zwrotu'];  
$zm=$_POST['zmien'];  
// Połączmy się z bazą danych  
$dbh = pg_connect("dbname=biblioteka user=jelkur  
password=123 host=localhost") or die("Nie  
moge polaczyc sie z baza danych ! ");  
// tu powinno byc polaczenie  
// wykonajmy zapytanie - najpierw wyswietlmy dane  
$query = "SELECT  
kto_wypozyczył,co_wypozyczył,data_wypozyczenia,data_zwrotu,pracownik_wypozyczenia,pracow  
nik_zwrotu, id_wypozyczenie from wypozyczenie  
where id_wypozyczenie=$id_wypozyczenie";  
$wynik = pg_query($query);  
echo "<form action=edytuj_wypozyczenie.php method=post>";  
echo "<table border width=1>";  
$lk = pg_num_fields($wynik);  
for($k = 0;$k<$lk -1; $k++)  
{  
echo "<tr>";  
echo "<td>";  
echo pg_field_name($wynik,$k);  
echo "</td>";  
echo "<td>";  
$f_name = pg_field_name($wynik, $k);
```

---

---

```

    $f_val = pg_fetch_result($wynik,0,$k);
    echo "<input type=text name='$f_name' value='$f_val'>";
    echo "</td>";
    echo "</tr>";
}
echo "</table>";
echo "<input type=hidden name=id_wypozyczenie value=$id_wypozyczenie>";
echo "<input type=submit name=zmien value=zmien>";
echo "</form>";
if ($zm)
{
    $query = "UPDATE wypozyczenie SET kto_wypozyczyl='$kto_wypozyczyl',
co_wypozyczyl='$co_wypozyczyl', data_wypozyczenia='$data_wypozyczenia',
data_zwrotu='$data_zwrotu', pracownik_wypozyczen$
    $wynik = pg_query($query);
    // sprawdzmy ile wierszy podmieniono
    $lz = pg_affected_rows($wynik);
    echo " Edytowano $lz wypozyczenie <br /> \n";
}
pg_close($dbh);
// zapewnijmy powrot do strony poprzedniej
echo "<form action=wypozyczenie.php method=post>
<input type=submit name=Ok value=OK>
</form>";
?>
</body>
</html>

```

Kod zawarty w skrypcie **usun\_wypozyczenie.php** (do usuwania wypożyczeń):

```

<html>
<head>
<title>Usuń wypożyczenie</title>
</head>
<body>
<h1> Usuam wypożyczenie </h1>
<?php
// tu umieszczamy kod skryptu
//po pierwsze – odbierzmy parametry dla skryptu przekazane
//metoda //POST – użyjemy je w zapytaniu poniżej
$id_wypozyczenie = $_POST['id_wypozyczenie'];
// Połączmy się z bazą danych
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie
moge polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie

```

---

---

```

$query = "DELETE FROM wypozyczenie where id_wypozyczenie=$id_wypozyczenie";
$wynik = pg_query($query);
// sprawdzmy ile wierszy podmieniono
$lz = pg_affected_rows($wynik);
echo " Usunieto $lz wypozyczenie <br />\n";
pg_close($dbh);
// zapewnijmy powrot do strony poprzedniej
echo "<form action=wypozyczenie.php method=post>
<input type=submit name=Ok value=OK>
</form>";
?>
</body>
</html>

```

Kod zawarty w skrypcie **ile\_wypozycczen.php** (do sprawdzenia ile wypożyczeń ma konkretna osoba, wywoływanie funkcji):

```

<html>
<head>
<title>Ile wypożyczeń</title>
</head>
<body>
<h1> Ile wypożyczeń ma konkretna osoba?</h1>
<?php
// tu umieszczamy kod skryptu
//po pierwsze – odbierzmy parametry dla skryptu przekazane
//metoda //POST – użyjemy je w zapytaniu poniżej
$id_czytelnik = $_POST['id_czytelnik'];
// Połączmy się z bazą danych
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie
moge polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
$query = "SELECT ile_wypozycczen('$id_czytelnik')";
$wynik = pg_query($dbh,$query);
$lz = pg_fetch_result($wynik,0,0);
// sprawdzmy ile wierszy podmieniono
echo " Czytelnik ma $lz wypożyczonych książek <br />\n";
pg_close($dbh);
// zapewnijmy powrot do strony poprzedniej
echo "<form action=wypozycczenie.php method=post>
<input type=submit name=Ok value=OK>
</form>";
?>
</body>
</html>

```

---

---

Kod zawarty w skrypcie **wypozyczenia\_za\_miesiac.php** (do sprawdzenia ile dokonano wypożyczeń za określony czas, wywołanie funkcji):

```
<html>
<head>
<title>Usuń osobę</title>
</head>
<body>
<h1> Ile dokonano wypożyczeń za podany okres?</h1>
<?php
// tu umieszczamy kod skryptu
//po pierwsze – odbierzmy parametry dla skryptu przekazane
//metoda //POST – użyjemy je w zapytaniu poniżej
$okres_od = $_POST['okres_od'];
$okres_do = $_POST['okres_do'];
// Połączmy się z bazą danych
$dbh = pg_connect("dbname=biblioteka user=jelkur
password=123 host=localhost") or die("Nie
mogę polaczyc sie z baza danych ! ");
// tu powinno byc polaczenie
// wykonajmy zapytanie
$query = "SELECT wypozyczenia_za_miesiac('$okres_od','$okres_do')";
$wynik = pg_query($dbh,$query);
$lz = pg_fetch_result($wynik,0,0);
// sprawdzmy ile wierszy podmieniono
//$lz = pg_affected_rows($wynik);
echo " Za ten okres dokonano $lz wypożyczeń <br />\n";
pg_close($dbh);
// zapewnijmy powrot do strony poprzedniej
echo "<form action=wypozyczenie.php method=post>
<input type=submit name=Ok value=OK>
</form>";
?>
</body>
</html>
```

## 2. Dodawanie, usuwanie i edycja rekordów z wybranych tabel

Poniżej zaprezentowano działanie stworzonego interfejsu PHP. Dodawać, usuwać i edytować rekordy można w dwóch różnych tabelach: w tabeli "Czytelnik" i tabeli "Wypożyczenie".

Chcąc dodać nowego czytelnika do bazy danych, na końcu strony **/czytelnik.php** należy podać dane nowego czytelnika, po czym trzeba nacisnąć przycisk "Dodaj" (rys. 22).

Dodaj nową osobę do bazy

Imię:  Nazwisko:  Email:  Telefon:

Data urodzenia:

Rys.22 - Dodawanie nowego czytelnika do bazy danych 1/3

Przycisk "Dodaj" przekierowuje użytkownika na stronę **/dodaj.php**, gdzie wyświetla się komunikat o dodanej osobie (rys. 23).

## Dodaję osobę

Dodano 1 osob

Rys.23 -Dodawanie nowego czytelnika do bazy danych 2/3

Po naciśnięciu przycisku "OK", wracamy do strony **/czytelnik.php**, gdzie na końcu tabeli "Czytelnik" widać dodanego do bazy danych nowego czytelnika.

8	Adam	Jankowski	adam.jankowski@gmail.com	722098764	1998-12-25	<input type="button" value="Usun"/>	<input type="button" value="Edytuj"/>
13	Jelizaveta	Kurilcik	jelizaveta.kurilcik@gmail.com	712345678	2001-03-08	<input type="button" value="Usun"/>	<input type="button" value="Edytuj"/>

Rys.24 -Dodawanie nowego czytelnika do bazy danych 3/3

Chcąc edytować czytelnika w bazie danych, należy kliknąć na przycisk "Edytuj" obok użytkownika, którego dane trzeba edytować. Po kliknięciu przechodzimy na stronę **/edytuj.php**, w której można dokonać edycji różnych informacji czytelnika. (rys. 25).

## Edytuj osobę

imie	<input type="text" value="Jelizaveta"/>
nazwisko	<input type="text" value="Kurilcik"/>
email	<input type="text" value="jelizaveta.kurilcik@gmail.co"/>
telefon	<input type="text" value="712345678"/>
data_urodzenia	<input type="text" value="2001-03-08"/>

Edytowano 1 osob

Rys.25 - Edycja czytelnika w bazie danych 1/2

Po naciśnięciu przycisku "Zmień" dostajemy informację, że zmieniono dane osoby. Przycisk "OK" cofa do strony **/czytelnik.php**, gdzie na końcu tabeli "Czytelnik" widać edytowane dane czytelnika (rys. 26).

13	Jelizaveta	Kurilcik	jk@gmail.com	712345678	2000-01-01	Usun	Edytuj
----	------------	----------	--------------	-----------	------------	------	--------

Rys.25 - Edycja czytelnika w bazie danych 2/2

Chcąc usunąć czytelnika z bazy danych, należy kliknąć na przycisk "Usun" obok użytkownika, którego dane chcemy usunąć.. Po kliknięciu przyciska przechodzimy na stronę **/usun.php**, gdzie wyświetla się komunikat o usunięciu osoby (rys. 26). Po powrocie do strony **/czytelnik.php** można zauważyć, że w bazie danych nie ma już osoby, którą usunęliśmy (rys.27).

UWAGA! Nie można usuwać osoby, która ma wypożyczoną książkę - nie pozwala na to sama struktura danej bazy danych.

## Usuwać osobę

Usunięto 1 osobę

OK

Rys.26 - Usuwanie czytelnika z bazy danych 1/2

8	Adam	Jankowski	adam.jankowski@gmail.com	722098764	1998-12-25	Usun	Edytuj
---	------	-----------	--------------------------	-----------	------------	------	--------

Rys.27 - Usuwanie czytelnika z bazy danych 2/2

Analogiczny proces dodawania, edycji i usuwania wypożyczeń występuje w tabeli "Wypożyczenie" na stronie **/wypozyczenie.php**. Przykład dodawania wypożyczenia można zobaczyć na rys. 28-30. Przykłady z edycją i usuwaniem wypożyczenia nie zostały uwzględnione w dokumentacji ze względu na swoje podobieństwo do bliźniaczych przykładów z tabeli "Czytelnik".

Dodaj nowe wypożyczenie bazy

Kto wypożyczył: 4 Co wypożyczył: 10 Data wypożyczenia:

2022-06-05

Pracownik przyjmujący wypożyczenie: 1

Dodaj

Rys.28 - Dodawanie nowego wypożyczenia do bazy danych 1/3

## Dodaję wypożyczenie

Dodano 1 wypożyczenie.

\*Jeśli liczba wypożyczeń równa 0, oznacza to że czytelnik już ma 3 wypożyczone książki.

OK

Rys.29 - Dodawanie nowego wypożyczenia do bazy danych 2/3

35	7	1	2022-06-05		2	1	Usun	Edytuj
27	2	4	2022-06-02	2022-06-03	2	1	Usun	Edytuj
38	4	10	2022-06-05		1	1	Usun	Edytuj

Rys.30 - Dodawanie nowego wypożyczenia do bazy danych 2/3

### 3. Wyświetlenie widoku

Wyświetlenie widoku można zobaczyć po naciśnięciu przycisku "Widok" na stronie **/glowna.php**. Interfejs z danymi widoku został przedstawiony na rys.21.

### 4. Wywołanie funkcji

Działanie zaimplementowanych funkcji w bazie danych można zobaczyć na końcu strony **/wypozyczenie.php**.

Pierwszą funkcją można sprawdzić, ile zostało zrealizowano wypożyczeń w ciągu podanego przez użytkownika okresu. Daty należy wpisywać w postaci XXXX-XX-XX. Przykładowo sprawdzono, ile wypożyczeń dodano od 2022-04-01 do 2022-04-30 (rys. 31).

Ile dokonano wypożyczeń za podany okres? Wpisz XXXX-XX-XX

Od: 2022-04-01

Do: 2022-04-30

Sprawdź

Rys.31 - Działanie funkcji `ile_dokonano_wypozychen()` 1/2

---

Po naciśnięciu przycisku "Sprawdź" zostajemy przekierowani na stronę **/wypożyczenia\_za\_miesiac.php**, w której uzyskujemy odpowiedź: Za ten okres dokonano 7 wypożyczeń (rys. 32). Po werifikacji danej informacji z zamieszczonymi danymi w tabeli "Wypożyczenie" można dojść do wniosku, że odpowiedź jest poprawna.

### Ile dokonano wypożyczeń za podany okres?

Za ten okres dokonano 7 wypożyczeń

OK

Rys.32 - Działanie funkcji `ile_dokonano_wypozycczen()` 2/2

Korzystanie z drugiej funkcji pozwala na uzyskanie informacji, ile wypożyczeń ma konkretny czytelnik. Wystarczy, że podamy indeks czytelnika i naciśniemy przycisk "Sprawdź" (rys. 33). Zostaniemy wówczas przekierowani na stronę **ile\_wypozycczen.php**, na której podano odpowiedź na zapytanie. Konkretnie w danym przypadku czytelnik o indeksie 4 ma 3 wypożyczone książki (rys. 34).

Chcesz się dowiedzieć, ile wypożyczeń ma konkretny czytelnik? Wpisz jego indeks:

Indeks czytelnika:

4

Sprawdź

Rys.33 - Dodawanie nowego czytelnika do bazy danych 1/2

### Ile wypożyczeń ma konkretna osoba?

Czytelnik ma 3 wypożyczonych książek

OK

Rys.34 - Dodawanie nowego czytelnika do bazy danych 2/2

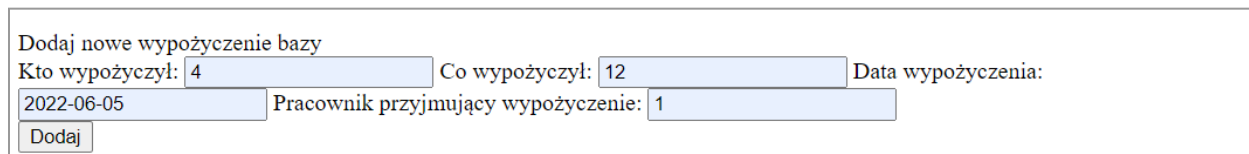


---

## 5. Uruchomienie wyzwalacza

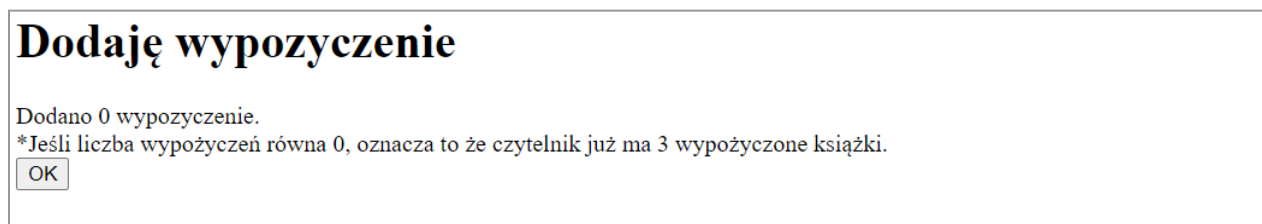
W danej bazie danych uruchomiony wyzwalacz nadzoruje, żeby czytelnik nie przekroczył limitu wypożyczonych książek i blokuje nowe wypożyczenie, jeśli czytelnik ma/miał wypożyczone 3 książki. Działanie wyzwalacza można zobaczyć na poniższym przykładzie.

Nieco wcześniej skorzystaliśmy z funkcji sprawdzającej że czytelnik o indeksie 4 ma wypożyczone 3 książki. Można spróbować dokonać zapisu wypożyczenia przez czytelnika "4" książki "12" (rys. 35).



Rys.35 - Działanie wyzwalacza 1/3

Klikając na przycisk "Dodaj" i przechodząc do strony ***/dodaj\_wypozyczenie.php*** widzimy komunikat, że dodano 0 wypożyczeń (rys.36). Oznacza to, że zadziałał wyzwalacz, który nie pozwolił zapisać nowego wypożyczenia, gdyż dany czytelnik już ma 3 wypożyczone książki. W przypadku, gdy czytelnik nie może więcej wypożyczyć książek, należy usunąć jedno z jego trzech wypożyczeń (wyzwalacz nie uwzględnia, czy książkę zwrócono, a jedynie reaguje na to, czy dokonano wypożyczenia).



Rys.36 - Działanie wyzwalacza 2/3

Po powrocie do strony ***/wypozyczenie.php*** widzimy, że rzeczywiście nowe wypożyczenie nie zostało dodane (rys.37).

---

35	7	1	2022-06-05		2	1	Usun	Edytuj
27	2	4	2022-06-02	2022-06-03	2	1	Usun	Edytuj
38	4	10	2022-06-05		1	1	Usun	Edytuj

*Rys.37 - Działanie wyzwalacza 3/3*