**Spis użytych w projekcie zapytań MySQL**

Spis treści

[1. OKNO LOGOWANIA 2](#_Toc469589044)

[1.1. Logowanie 2](#_Toc469589045)

[1.2. Rejestracja 2](#_Toc469589046)

[2. OKNO PROGRAMU 2](#_Toc469589047)

[2.1. Zakładka Books 2](#_Toc469589048)

[2.1.1. Wyszukiwanie książki 2](#_Toc469589049)

[2.1.2. Rezerwacja książki 3](#_Toc469589050)

[2.1.3. Wypożyczenie książki 3](#_Toc469589051)

[2.1.4. Pobranie zarezerwowanych książek 3](#_Toc469589052)

[2.1.5. Pobranie wypożyczonych książek 3](#_Toc469589053)

[2.1.6. Usunięcie rezerwacji książki 3](#_Toc469589054)

[2.2. Zakładka „Settings” 4](#_Toc469589055)

[2.2.1. Pobranie danych zalogowanego użytkownika 4](#_Toc469589056)

[2.2.2. Zmiana danych personalnych zalogowanego użytkownika 4](#_Toc469589057)

[2.2.3. Zmiana hasła zalogowanego użytkownika 4](#_Toc469589058)

[2.3. Zakładka „Statistics” 5](#_Toc469589059)

[2.3.1. Najbardziej aktywny użytkownik 5](#_Toc469589060)

[2.3.2. Najczęściej wypożyczana książka 5](#_Toc469589061)

[2.3.3. Najbardziej popularny autor 6](#_Toc469589062)

[2.3.4. Historia wypożyczeń danego tytułu 6](#_Toc469589063)

[2.3.5. Popularność danego autora 6](#_Toc469589064)

[2.3.6. Popularność danej książki 6](#_Toc469589065)

[2.3.7. Ilość wypożyczeni danego użytkownika 7](#_Toc469589066)

[2.4. Zakładka „Librarian” 7](#_Toc469589067)

[2.4.1. Wyszukiwanie książki 7](#_Toc469589068)

[2.4.2. Zwrócenie książki 7](#_Toc469589069)

[2.4.3. Pobranie wydawnictw 7](#_Toc469589070)

[2.4.4. Pobranie autorów 7](#_Toc469589071)

[2.4.5. Pobranie gatunków 7](#_Toc469589072)

[2.4.6. Dodanie/usunięcie książki 8](#_Toc469589073)

[2.4.7. Dodanie/usunięcie autora 8](#_Toc469589074)

[2.4.8. Dodanie/usunięcie wydawnictwa 8](#_Toc469589075)

[2.4.9. Dodanie/usunięcie gatunku 8](#_Toc469589076)

[2.5. Zakładka „Admin Panel” 8](#_Toc469589077)

[2.5.1. Wyszukiwanie użytkowników 8](#_Toc469589078)

[2.5.2. Zablokowanie/odblokowanie wybranego użytkownika 9](#_Toc469589079)

[2.5.3. Zmiana typu konta wybranego użytkownika(admin/normal user) 9](#_Toc469589080)

[2.5.4. Zmiana danych personalnych wybranego użytkownika 9](#_Toc469589081)

[2.5.5. Zmiana hasła wybranego użytkownika 9](#_Toc469589082)

# OKNO LOGOWANIA

## Logowanie

Zapytanie SQL:

SELECT idUzytkownika, czy\_pracownik, czy\_zablokowane FROM uzytkownik

WHERE login = '" + tbLUser->Text + "'

AND haslo = PASSWORD('" + tbLPassword->Text + "');

## Rejestracja

Zapytanie SQL:

BEGIN;

INSERT INTO użytkownik (login, haslo)

VALUES('"+ tbSLogin->Text +"', PASSWORD('" + tbSPassword1->Text + "'));

INSERT INTO dane (imie, nazwisko, email, nr\_tel, miasto, ulica, nr\_budynku,

nr\_lokalu, idUzytkownika)

VALUES('"+tbSName->Text +"', '"+tbSSurname->Text +"', '"+tbSEmail->Text+ "', '"+ tbSPNumber->Text +"', '"+ tbSCity->Text + "', '" + tbSStreet->Text + "', '" + tbSBNumber->Text + "', '" + tbSANumber->Text + "', LAST\_INSERT\_ID());

COMMIT;

# OKNO PROGRAMU

## Zakładka Books

### Wyszukiwanie książki

Zapytanie MySQL:

CREATE VIEW 'dane\_ksiazki' AS

SELECT idKsiazka,tytul,dostepnosc,ISBN,data\_wydania,

imie AS imie\_autora, nazwisko AS nazwisko\_autora,

kraj\_pochodzenia AS kraj\_pochodzenia\_autora,

wydawnictwo.nazwa AS wydawnictwo,gatunek.nazwa AS gatunek

FROM ksiazka join gatunek on ksiazka.idGatunek = gatunek.idGatunek join

wydawnictwo on ksiazka.idWydawnictwo = wydawnictwo.idWydawnictwo join autor

on autor.idAutor = ksiazka.idAutor;

SELECT \* FROM dane\_ksiazki

WHERE CONCAT\_WS(tytul, ' ',ISBN, ' ',data\_wydania,' ',imie\_autora,' ',

nazwisko\_autora, ' ', kraj\_pochodzenia\_autora, ' ', wydawnictwo, ' ',

gatunek) LIKE '%" + tb->Text + "%' ORDER BY idKsiazka;

DROP VIEW dane\_ksiazki;

### Rezerwacja książki

Zapytanie MySQL:

BEGIN;

INSERT INTO rezerwacja(data\_rezerwacji, idUzytkownika, idKsiazka)

VALUES(STR\_TO\_DATE('" + thisDay + "', '%d.%m.%Y %H:%i:%s'), " + id\_uzytkownika + ", " + id\_ksiazki + ");

UPDATE ksiazka SET dostepnosc = 0 WHERE idKsiazka = " + id\_ksiazki + ";

COMMIT;

### Wypożyczenie książki

Zapytanie MySQL:

BEGIN;

INSERT INTO wypozyczenie(data\_wypozyczenia, przewidywana\_data\_zwrotu, idUzytkownika, idKsiazka)

VALUES(STR\_TO\_DATE('" + thisDay1 + "', '%d.%m.%Y %H:%i:%s'), STR\_TO\_DATE('" + dateOfReturn + "', '%d.%m.%Y %H:%i:%s'), " + id\_uzytkownika + ", " + id\_ksiazki + ");

DELETE FROM rezerwacja WHERE idKsiazka = " + id\_ksiazki + ";

UPDATE ksiazka SET l\_wypozyczen = l\_wypozyczen+1

WHERE idKsiazka = " + id\_ksiazki + ";

UPDATE uzytkownik SET calkowita\_l\_wypozyczen = calkowita\_l\_wypozyczen+1

WHERE idUzytkownika = " + id\_uzytkownika + ";

COMMIT;

### Pobranie zarezerwowanych książek

Zapytanie MySQL:

SELECT rezerwacja.idKsiazka, tytul, data\_rezerwacji

FROM rezerwacja join ksiazka

WHERE rezerwacja.idUzytkownika = "+ id\_uzytkownika +" AND ksiazka.idKsiazka = rezerwacja.idKsiazka;

### Pobranie wypożyczonych książek

Zapytanie MySQL:

SELECT wypozyczenie.idKsiazka, tytul, data\_wypozyczenia, przewidywana\_data\_zwrotu FROM wypozyczenie join ksiazka

WHERE wypozyczenie.idUzytkownika = " + id\_uzytkownika + " AND ksiazka.idKsiazka = wypozyczenie.idKsiazka AND data\_zwrotu IS NULL;

### Usunięcie rezerwacji książki

Zapytanie MySQL:

BEGIN;

DELETE FROM rezerwacja

WHERE idKsiazka = " + id\_ksiazki + ";

UPDATE ksiazka SET dostepnosc = 1 WHERE idKsiazka = " + id\_ksiazki + ";

COMMIT;

## Zakładka „Settings”

### Pobranie danych zalogowanego użytkownika

Zapytanie MySQL:

SELECT uzytkownik.idUzytkownika, uzytkownik.login, uzytkownik.czy\_pracownik,

uzytkownik.czy\_zablokowane,

dane.imie, dane.nazwisko, dane.miasto, dane.ulica, dane.nr\_budynku,

dane.nr\_lokalu, nr\_tel, dane.email

FROM uzytkownik, dane

WHERE uzytkownik.idUzytkownika = dane.idUzytkownika AND uzytkownik.idUzytkownika = " + id\_uzytkownika + ";

### Zmiana danych personalnych zalogowanego użytkownika

Zapytanie MySQL:

UPDATE dane SET imie = '"+ tbSName->Text +"', nazwisko = '"+ tbSSurname->Text+ "',

miasto = '"+ tbSCity->Text +"', ulica = '"+ tbSStreet->Text +"',

nr\_budynku = '"+ tbSBNumber->Text +"', nr\_lokalu = '"+ tbSANumber->Text +"',

nr\_tel = '"+ tbSPNumber->Text +"', email = '"+ tbSEmail->Text +"'

WHERE idUzytkownika = '"+ id\_uzytkownika +"'; "

### Zmiana hasła zalogowanego użytkownika

Zapytanie MySQL:

UPDATE uzytkownik SET haslo = PASSWORD('" + tbSNew1->Text + "')

WHERE idUzytkownika = " + id\_uzytkownika + " AND haslo = PASSWORD('" + tbSOld->Text + "');

## Zakładka „Statistics”

### Najbardziej aktywny użytkownik

Zapytanie MySQL (ASC):

CREATE PROCEDURE uzytkownicy\_stat()

SELECT dane.imie, dane.nazwisko, uzytkownik.login, count(idWypozyczenie) as Ilosc\_wypozyczen

FROM wypozyczenie inner join uzytkownik on uzytkownik.idUzytkownika = wypozyczenie.idUzytkownika left join dane on dane.idUzytkownika = uzytkownik.idUzytkownika

WHERE uzytkownik.idUzytkownika = wypozyczenie.idUzytkownika

GROUP BY login

ORDER BY Ilosc\_wypozyczen ASC;

CALL uzytkownicy\_stat;

DROP PROCEDURE uzytkownicy\_stat;

Zapytanie MySQL (DESC):

CREATE PROCEDURE uzytkownicy\_stat()

SELECT dane.imie, dane.nazwisko, uzytkownik.login, count(idWypozyczenie) as Ilosc\_wypozyczen

FROM wypozyczenie inner join uzytkownik on uzytkownik.idUzytkownika = wypozyczenie.idUzytkownika left join dane on dane.idUzytkownika = uzytkownik.idUzytkownika

WHERE uzytkownik.idUzytkownika = wypozyczenie.idUzytkownika

GROUP BY login

ORDER BY Ilosc\_wypozyczen DESC;

CALL uzytkownicy\_stat;

DROP PROCEDURE uzytkownicy\_stat;

### Najczęściej wypożyczana książka

Zapytanie MySQL (ASC):

CREATE PROCEDURE ksiazki\_stat()

SELECT ksiazka.tytul, autor.imie, autor.nazwisko, count(idWypozyczenie) as Ilosc\_wypozyczen

from wypozyczenie inner join ksiazka on ksiazka.idKsiazka = wypozyczenie.idKsiazka left join autor on autor.idAutor = ksiazka.idAutor

where ksiazka.idKsiazka = wypozyczenie.idKsiazka

GROUP BY ksiazka.tytul

ORDER BY Ilosc\_wypozyczen ASC;

CALL ksiazki\_stat;

DROP PROCEDURE ksiazki\_stat;

Zapytanie MySQL (DESC):

CREATE PROCEDURE ksiazki\_stat()

SELECT ksiazka.tytul, autor.imie, autor.nazwisko, count(idWypozyczenie) as Ilosc\_wypozyczen

from wypozyczenie inner join ksiazka on ksiazka.idKsiazka = wypozyczenie.idKsiazka left join autor on autor.idAutor = ksiazka.idAutor

where ksiazka.idKsiazka = wypozyczenie.idKsiazka

GROUP BY ksiazka.tytul

ORDER BY Ilosc\_wypozyczen DESC;

CALL ksiazki\_stat;

DROP PROCEDURE ksiazki\_stat;

### Najbardziej popularny autor

Zapytanie MySQL (ASC):

CREATE PROCEDURE autor\_stat()

SELECT autor.imie, autor.nazwisko, count(idWypozyczenie) as Ilosc\_wypozyczen FROM wypozyczenie inner join ksiazka on ksiazka.idKsiazka = wypozyczenie.idKsiazka left join autor on autor.idAutor = ksiazka.idAutor

WHERE ksiazka.idKsiazka = wypozyczenie.idKsiazka

GROUP BY autor.nazwisko

ORDER BY Ilosc\_wypozyczen DESC;

CALL autor\_stat;

DROP PROCEDURE autor\_stat;

Zapytanie MySQL (DESC):

CREATE PROCEDURE autor\_stat()

SELECT autor.imie, autor.nazwisko, count(idWypozyczenie) as Ilosc\_wypozyczen FROM wypozyczenie inner join ksiazka on ksiazka.idKsiazka = wypozyczenie.idKsiazka left join autor on autor.idAutor = ksiazka.idAutor

WHERE ksiazka.idKsiazka = wypozyczenie.idKsiazka

GROUP BY autor.nazwisko

ORDER BY Ilosc\_wypozyczen DESC;

CALL autor\_stat;

DROP PROCEDURE autor\_stat;

### Historia wypożyczeń danego tytułu

CREATE VIEW wypozyczenia\_ksiazka as

SELECT ksiazka.tytul, wypozyczenie.data\_wypozyczenia, wypozyczenie.data\_zwrotu, uzytkownik.login, dane.imie, dane.nazwisko

FROM wypozyczenie

INNER JOIN ksiazka on ksiazka.idKsiazka = wypozyczenie.idKsiazka

INNER JOIN uzytkownik on uzytkownik.idUzytkownika = wypozyczenie.idUzytkownika

INNER JOIN dane on dane.idUzytkownika = uzytkownik.idUzytkownika

GROUP BY idWypozyczenie

ORDER BY data\_wypozyczenia;

CREATE PROCEDURE pokaz\_wypozyczenia(in tytul Varchar(255))

SELECT login, imie, nazwisko, data\_wypozyczenia, data\_zwrotu from wypozyczenia\_ksiazka

WHERE wypozyczenia\_ksiazka.tytul like concat('%', tytul, '%' );

CALL pokaz\_wypozyczenia ('" + tbSSetLogin->Text + "');

DROP VIEW wypozyczenia\_ksiazka;

DROP PROCEDURE pokaz\_wypozeczenia;

### Popularność danego autora

Zapytanie MySQL:

CREATE PROCEDURE statystyki\_autor (in imie VARCHAR(40), nazwisko varchar(40))

SELECT imie, nazwisko, count(idWypozyczenie) as Ilość\_wypożyczeń

FROM wypozyczenie inner join ksiazka on ksiazka.idKsiazka = wypozyczenie.idKsiazka inner join autor on autor.idAutor = ksiazka.idAutor

WHERE imie = autor.imie or nazwisko = autor.nazwisko;

CALL statystyki\_autor('"+ tbStAname->Text +"','"+ tbStAsurname->Text +"');

DROP PROCEDURE statystyki\_autor;

### Popularność danej książki

Zapytanie MySQL:

CREATE PROCEDURE statystyki\_ksiazka (in tytul VARCHAR(255))

SELECT ksiazka.tytul, count(distinct idWypozyczenie) as Ilość\_wypożyczeń

FROM wypozyczenie inner join ksiazka on ksiazka.idKsiazka = wypozyczenie.idKsiazka

WHERE ksiazka.tytul like concat('%', tytul, '%');

CALL statystyki\_ksiazka('" + tbStBook->Text + "');

DROP PROCEDURE statystyki\_ksiazka;

### Ilość wypożyczeni danego użytkownika

Zapytanie MySQL:

CREATE PROCEDURE statystyki\_uzytkownik (in login VARCHAR(40))

SELECT uzytkownik.login, count(idWypozyczenie) as Ilość\_wypożyczeń

FROM wypozyczenie inner join uzytkownik on uzytkownik.idUzytkownika = wypozyczenie.idUzytkownika

WHERE BINARY login = BINARY uzytkownik.login

CALL statystyki\_uzytkownik('" + tbSSetLogin->Text + "');

DROP PROCEDURE statystyki\_uzytkownik;

## Zakładka „Librarian”

### Wyszukiwanie książki

Zapytanie MySQL:

CREATE VIEW 'dane\_ksiazki' AS

SELECT idKsiazka,tytul,dostepnosc,ISBN,data\_wydania,

imie AS imie\_autora, nazwisko AS nazwisko\_autora,

kraj\_pochodzenia AS kraj\_pochodzenia\_autora,

wydawnictwo.nazwa AS wydawnictwo,gatunek.nazwa AS gatunek

FROM ksiazka join gatunek on ksiazka.idGatunek = gatunek.idGatunek join

wydawnictwo on ksiazka.idWydawnictwo = wydawnictwo.idWydawnictwo join autor

on autor.idAutor = ksiazka.idAutor;

SELECT \* FROM dane\_ksiazki

WHERE CONCAT\_WS(tytul, ' ',ISBN, ' ',data\_wydania,' ',imie\_autora,' ',

nazwisko\_autora, ' ', kraj\_pochodzenia\_autora, ' ', wydawnictwo, ' ',

gatunek) LIKE '%" + tb->Text + "%' AND dostepnosc = 1 ORDER BY idKsiazka;

DROP VIEW dane\_ksiazki;

### Zwrócenie książki

Zapytanie MySQL:

UPDATE wypozyczenie, ksiazka SET wypozyczenie.data\_zwrotu = STR\_TO\_DATE('" + dateOfReturn + "', '%d.%m.%Y %H:%i:%s'), ksiazka.dostepnosc = 1

WHERE wypozyczenie.idKsiazka = " + tbLReturn->Text + " AND ksiazka.idKsiazka = " + tbLReturn->Text + ";

### Pobranie wydawnictw

Zapytanie MySQL:

SELECT idWydawnictwo, nazwa FROM wydawnictwo ORDER BY idWydawnictwo;

### Pobranie autorów

Zapytanie MySQL:

SELECT idAutor, imie, nazwisko FROM autor ORDER BY idAutor;"

### Pobranie gatunków

Zapytanie MySQL:

SELECT idGatunek, nazwa FROM gatunek ORDER BY idGatunek;

### Dodanie/usunięcie książki

Zapytanie MySQL (dodawanie):

INSERT INTO ksiazka(tytul, ISBN, data\_wydania, l\_wypozyczen, idAutor, idWydawnictwo, idGatunek, dostepnosc) VALUES('"+ tbLTitle->Text +"', '" + tbLISBN->Text + "', STR\_TO\_DATE('"+ tbLDate->Text +"', '%d.%m.%Y'), 0, '" + id\_autor + "', '" + id\_wydawnictwo + "', '" + id\_gatunek + "', 1);

Zapytanie MySQL (usuwanie):

DELETE FROM ksiazka WHERE idKsiazka = " + id\_ksiazki + ";

### Dodanie/usunięcie autora

Zapytanie MySQL (dodawanie):

INSERT INTO autor(imie, nazwisko, kraj\_pochodzenia) VALUES('" + tbLName->Text + "', '" + tbLSurname->Text + "', '" + tbLCountry->Text + "');

Zapytanie MySQL (usuwanie):

DELETE FROM autor WHERE idAutor = "+ id\_autor +";

### Dodanie/usunięcie wydawnictwa

Zapytanie MySQL (dodawanie):

INSERT INTO wydawnictwo(nazwa) VALUES('" + tbLWydawnictwo->Text + "');

Zapytanie MySQL (usuwanie):

DELETE FROM wydawnictwo WHERE idWydawnictwo = " + id\_wydawnictwo + ";

### Dodanie/usunięcie gatunku

Zapytanie MySQL (dodawanie):

INSERT INTO gatunek(nazwa) VALUES('" + tbLType->Text + "');

Zapytanie MySQL (usuwanie):

DELETE FROM gatunek WHERE idGatunek = " + id\_gatunek + ";

## Zakładka „Admin Panel”

### Wyszukiwanie użytkowników

Zapytanie SQL:

SELECT uzytkownik.idUzytkownika, uzytkownik.login, uzytkownik.czy\_pracownik,

uzytkownik.czy\_zablokowane,

dane.imie, dane.nazwisko, dane.miasto, dane.ulica, dane.nr\_budynku, dane.nr\_lokalu, nr\_tel, dane.email

FROM uzytkownik, dane

WHERE uzytkownik.idUzytkownika = dane.idUzytkownika

AND CONCAT\_WS(login, ' ',imie, ' ',nazwisko,' ',miasto,' ',ulica, ' ', nr\_budynku,

' ', nr\_lokalu, ' ', nr\_tel, ' ', email) LIKE '%"+ tbAPSearch->Text +"%'

ORDER BY idUzytkownika;

LUB zapytanie SQL:

CREATE VIEW 'dane\_konta' AS

SELECT uzytkownik.idUzytkownika, login, czy\_pracownik, czy\_zablokowane, imie, nazwisko, email, nr\_tel, miasto, ulica, nr\_budynku, nr\_lokalu, zadluzenie, calkowita\_l\_wypozyczen

FROM uzytkownik join dane on uzytkownik.idUzytkownika = dane.idUzytkownika;

CREATE PROCEDURE wyszukaj\_uzytkownika(in tekst VARCHAR(255))

SELECT \* from dane\_konta

WHERE BINARY login like BINARY concat('%', tekst, '%') or

BINARY imie like BINARY concat('%', tekst, '%') or

BINARY nazwisko like BINARY concat('%', tekst, '%') or

BINARY email like BINARY concat('%', tekst, '%') or

BINARY miasto like BINARY concat('%', tekst, '%');

CALL wyszukaj\_uzytkownika('" + tbAPSearch->Text + "');

DROP VIEW dane\_konta;

DROP PROCEDURE wyszukaj\_uzytkownika;

### Zablokowanie/odblokowanie wybranego użytkownika

Zapytanie SQL blokujące konto:

UPDATE uzytkownik SET czy\_zablokowane = 1 WHERE idUzytkownika = "+ id\_rekordu + ";

Zapytanie SQL odblokowujące konto:

UPDATE uzytkownik SET czy\_zablokowane = 0 WHERE idUzytkownika = "+ id\_rekordu + ";

### Zmiana typu konta wybranego użytkownika(admin/normal user)

Zapytanie SQL nadające prawa administratora:

UPDATE uzytkownik SET czy\_pracownik = 1 WHERE idUzytkownika = "+ id\_rekordu + ";

Zapytanie SQL nadające praca zwykłego użytkownika:

UPDATE uzytkownik SET czy\_pracownik = 0 WHERE idUzytkownika = "+ id\_rekordu + ";

### Zmiana danych personalnych wybranego użytkownika

Zapytanie SQL:

BEGIN;

UPDATE uzytkownik SET login = '" + tbAPLogin->Text + "'

WHERE idUzytkownika = " + id\_rekordu + ";

UPDATE dane SET imie = '" + tbAPName->Text + "', nazwisko ='" + tbAPSurname->Text + "', email = '" + tbAPEmail->Text + "', nr\_tel = '" + tbAPPNumber->Text + "', miasto = '" + tbAPCity->Text + "', ulica = '" + tbAPStreet->Text + "', nr\_budynku = '" + tbAPBNumber->Text + "', nr\_lokalu = '" + tbAPANumber->Text + "'

WHERE idUzytkownika = " + id\_rekordu + ";

COMMIT;

### Zmiana hasła wybranego użytkownika

Zapytanie SQL:

UPDATE uzytkownik SET haslo = PASSWORD('" + tbAPNewP->Text + "')

WHERE idUzytkownika = " + id\_rekordu + ";