

Spis użytych w projekcie zapytań MySQL

Spis treści

1.	OKNO LOGOWANIA	2
1.1.	Logowanie	2
1.2.	Rejestracja	2
2.	OKNO PROGRAMU	2
2.1.	Zakładka Books.....	2
2.1.1.	Wyszukiwanie książki	2
2.1.2.	Rezerwacja książki	2
2.1.3.	Wypożyczenie książki.....	3
2.1.4.	Pobranie zarezerwowanych książek.....	3
2.1.5.	Pobranie wypożyczonych książek.....	3
2.1.6.	Usunięcie rezerwacji książki	3
2.2.	Zakładka „Settings”	4
2.2.1.	Pobranie danych zalogowanego użytkownika	4
2.2.2.	Zmiana danych personalnych zalogowanego użytkownika	4
2.2.3.	Zmiana hasła zalogowanego użytkownika	4
2.3.	Zakładka „Statistics”	5
2.3.1.	Najbardziej aktywny użytkownik	5
2.3.2.	Najczęściej wypożyczana książka	5
2.3.3.	Najbardziej popularny autor	6
2.3.4.	Popularność danego autora	6
2.3.5.	Popularność danej książki	6
2.3.6.	Ilość wypożyczeń danego użytkownika	6
2.4.	Zakładka „Librarian”	7
2.4.1.	Zwrócenie książki	7
2.4.2.	Pobranie wydawnictw	7
2.4.3.	Pobranie autorów	7
2.4.4.	Pobranie gatunków	7
2.4.5.	Dodanie/usunięcie książki.....	7
2.4.6.	Dodanie/usunięcie autora	7
2.4.7.	Dodanie/usunięcie wydawnictwa	8
2.4.8.	Dodanie/usunięcie gatunku.....	8
2.5.	Zakładka „Admin Panel”	8
2.5.1.	Wyszukiwanie użytkowników	8
2.5.2.	Zablokowanie/odblokowanie wybranego użytkownika	9
2.5.3.	Zmiana typu konta wybranego użytkownika(admin/normal user)	9
2.5.4.	Zmiana danych personalnych wybranego użytkownika.....	9
2.5.5.	Zmiana hasła wybranego użytkownika	9

1. OKNO LOGOWANIA

1.1. Logowanie

Zapytanie SQL:

```
SELECT idUzytkownika, czy_pracownik, czy_zablokowane FROM uzytkownik
WHERE login = '"' + tbLUser->Text + '"'
AND haslo = PASSWORD('"' + tbLPassword->Text + '"');
```

1.2. Rejestracja

Zapytanie SQL:

```
BEGIN;
INSERT INTO uzytkownik (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 + '"', '"' + tbSSStreet->Text + '"',
'"' + tbSBNumber->Text + '"', '"' + tbSANumber->Text + '"', LAST_INSERT_ID());
COMMIT;
```

2. OKNO PROGRAMU

2.1. Zakładka Books

2.1.1. 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;
```

2.1.2. 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;
```

2.1.3. 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_wypozycczen = l_wypozycczen+1
WHERE idKsiazka = " + id_ksiazki + ";

UPDATE uzytkownik SET calkowita_l_wypozycczen = calkowita_l_wypozycczen+1
WHERE idUzytkownika = " + id_uzytkownika + ";

COMMIT;
```

2.1.4. 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;
```

2.1.5. 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;
```

2.1.6. Usunięcie rezerwacji książki

Zapytanie MySQL:

```
BEGIN;
DELETE FROM rezerwacja
WHERE idKsiazka = " + id_ksiazki + ";

UPDATE ksiazka SET dostepnosc = 1 WHERE idKsiazka = " + id_ksiazki + ";
COMMIT;
```

2.2. Zakładka „Settings”

2.2.1. 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 + ";
```

2.2.2. 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 +''; "
```

2.2.3. Zmiana hasła zalogowanego użytkownika

Zapytanie MySQL:

```
UPDATE uzytkownik SET haslo = PASSWORD("'" + tbSNew1->Text + "')  
WHERE idUzytkownika = " + id_uzytkownika + " AND haslo = PASSWORD("'" + tbSOld->Text + "');
```

2.3. Zakładka „Statistics”

2.3.1. 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;
```

2.3.2. 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;
```

2.3.3. 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;
```

2.3.4. 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;
```

2.3.5. Popularność danej książki

Zapytanie MySQL:

```
CREATE PROCEDURE statystyki_ksiazka (in tytuł VARCHAR(255))  
    SELECT ksiazka.tytuł, count(distinct idWypozyczenie) as Ilość_wypożyczeń  
    FROM wypozyczenie inner join ksiazka on ksiazka.idKsiazka =  
    wypozyczenie.idKsiazka  
    WHERE ksiazka.tytuł like concat('%', tytuł, '%');  
CALL statystyki_ksiazka('" + tbStBook->Text + "');  
DROP PROCEDURE statystyki_ksiazka;
```

2.3.6. Ilość wypożyczeń 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 login = uzytkownik.login;  
CALL statystyki_uzytkownik('" + tbSSetLogin->Text + "');  
DROP PROCEDURE statystyki_uzytkownik;
```

2.4. Zakładka „Librarian”

2.4.1. 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;
```

2.4.2. Zwroćenie 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 + ";
```

2.4.3. Pobranie wydawnictw

Zapytanie MySQL:

```
SELECT idWydawnictwo, nazwa FROM wydawnictwo ORDER BY idWydawnictwo;
```

2.4.4. Pobranie autorów

Zapytanie MySQL:

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

2.4.5. Pobranie gatunków

Zapytanie MySQL:

```
SELECT idGatunek, nazwa FROM gatunek ORDER BY idGatunek;
```

2.4.6. 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 + ";
```

2.4.7. 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 +";
```

2.4.8. Dodanie/usunięcie wydawnictwa

Zapytanie MySQL (dodawanie):

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

Zapytanie MySQL (usuwanie):

```
DELETE FROM wydawnictwo WHERE idWydawnictwo = " + id_wydawnictwo + ";
```

2.4.9. Dodanie/usunięcie gatunku

Zapytanie MySQL (dodawanie):

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

Zapytanie MySQL (usuwanie):

```
DELETE FROM gatunek WHERE idGatunek = " + id_gatunek + ";
```

2.5. Zakładka „Admin Panel”

2.5.1. 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_wypozycczen  
FROM   uzytkownik join dane on uzytkownik.idUzytkownika = dane.idUzytkownika;
```

```
CREATE PROCEDURE wyszukaj_uzytkownika(in tekst VARCHAR(255))  
SELECT * from dane_konta  
WHERE login like concat('%', tekst, '%') or  
       imie like concat('%', tekst, '%') or  
       nazwisko like concat('%', tekst, '%') or  
       email like concat('%', tekst, '%') or  
       miasto like concat('%', tekst, '%');
```

```
CALL wyszukaj_uzytkownika('" + tbAPSearch->Text + "');
```

```
DROP VIEW dane_konta;
```

```
DROP PROCEDURE wyszukaj_uzytkownika;
```


2.5.2. 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 + ";
```

2.5.3. 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 pracę zwykłego użytkownika:

```
UPDATE uzytkownik SET czy_pracownik = 0 WHERE idUzytkownika = "+ id_rekordu + ";
```

2.5.4. 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;
```

2.5.5. Zmiana hasła wybranego użytkownika

Zapytanie SQL:

```
UPDATE uzytkownik SET haslo = PASSWORD('" + tbAPNewP->Text + "')
WHERE idUzytkownika = " + id_rekordu + ";
```