

# **Bazy danych**

## **Dokumentacja Projektu**

pt.: „Implementacja bazy danych dla systemu  
wspomagającego bibliotekę”

**Data wykonania:** 05.12.2020

**Grupa:** L02  
Dawid Byś  
161863  
3EF-ZI

## Spis treści

1. Cel pracy	3
2. Przebieg Pracy	3
2.1. Etap 1	3
- Określenie funkcji bazy danych i ich priorytetu:	3
- Wybór technologii, narzędzi i typu bazy danych do zrealizowania projektu:	3
- Prezentacja przygotowanego repozytorium z opisem:	3
2.2. Etap 2	4
- Diagram związków encji ERD:	4
- Opis tabel:	5
- Problemy powstałe w trakcie tworzenia diagramów:	5
2.3. Etap 3	6
- Skrypt tworzący bazę danych:	6
- Skrypt wypełniający bazę danymi:	6
- Funkcje bazodanowe:	7
- Przykładowe selecty:	11
3. Wnioski	12

# 1. Cel pracy

Tematem projektu będzie implementacja bazy danych dla systemu wykorzystywanego w bibliotece. Będzie on zapewniał szybsze i wygodniejsze zarządzanie wszystkimi książkami dostępnymi w danej bibliotece oraz ich czytelnikami.

## 2. Przebieg Pracy

### 2.1. Etap 1

#### **Określenie funkcji bazy danych i ich priorytetu:**

Baza danych aktualnie jest jednym z najważniejszych elementów, mający duży wpływ na szybkość działania aplikacji. Będą w niej przechowywane szczegółowe dane na temat książek, bibliotekarzy, czytelników oraz o wszystkich wypożyczeniach.

#### **Wybór technologii, narzędzi i typu bazy danych do zrealizowania projektu:**

W projekcie zostanie wykorzystana relacyjna baza danych MySQL firmy Oracle. Podyktowane jest to przede wszystkim popularnością, niezawodnością i wydajnością oferowanych narzędzi. Do komunikacji z bazą danych zostanie wykorzystany MySQL Workbench, który zapewnia łatwy sposób na wykonywanie zapytań sql i tworzenie diagramów.

#### **Prezentacja przygotowanego repozytorium z opisem:**

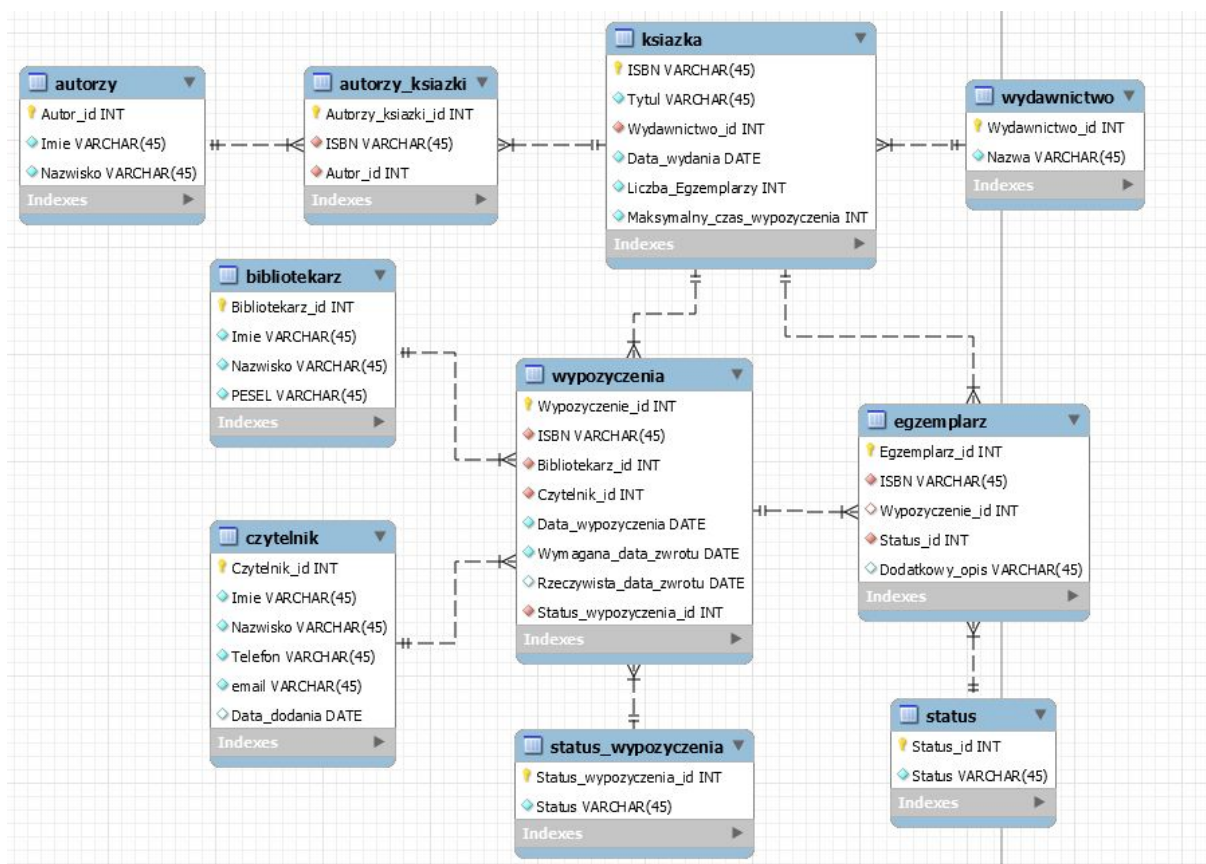
Repozytorium z projektem jest dostępne na platformie GitHub:  
<https://github.com/devqos/projektDB>

## 2.2. Etap 2

### Diagram związków encji ERD:

Służy do opisu związków pomiędzy poszczególnymi encjami czyli klasami danych występującymi w danej aplikacji.

Rysunek 1 Diagram ERD



## **Opis tabel:**

Książka - tabela z wszystkimi książkami dostępnymi w bibliotece, identyfikowanymi po numerze ISBN

Wydawnictwo - tabela przechowująca wydawnictwa poszczególnych książek

Egzemplarz - tabela przechowująca dane pojedynczego egzemplarza danej książki

Status - tabela słownikowa przechowująca możliwe statusy danego egzemplarza

Wypożyczenia - tabela przechowująca podstawowe dane na temat wypożyczenia danej książki przez czytelnika

Status\_wypożyczenia - tabela słownikowa przechowująca możliwe statusy danego wypożyczenia

Bibliotekarz - tabela przechowująca podstawowe dane potrzebne do identyfikacji bibliotekarza wydającego książkę czytelnikowi

Czytelnik - tabela przechowująca podstawowe dane potrzebne do identyfikacji czytelnika wypożyczającego książkę

Autorzy - tabela przechowująca dane na temat autorów książek

Autorzy\_książki - tabela pomocnicza potrzebna do przedstawienia związku "wiele do wielu" pomiędzy autorem a książką.

## **Problemy powstałe w trakcie tworzenia diagramów:**

Jedynym kłopotem była kwestia przechowywania statusów egzemplarzy oraz wypożyczenia w osobnych tabelach lub jako zwykłe pole tekstowe w danych encjach. Po namyśle zdecydowałem się że wykorzystam do tego dedykowane tabele żeby ujednolicić możliwe statusy i wykorzystać je później do łatwiejszego wyszukiwania danych.

## 2.2. Etap 3

### Skrypt tworzący bazę danych:

#### Wycinek 1. Fragment skryptu tworzącego bazę

```
CREATE TABLE IF NOT EXISTS biblioteka.Bibliotekarz (  
    Bibliotekarz_id INT AUTO_INCREMENT NOT NULL,  
    Imie VARCHAR(45) NOT NULL,  
    Nazwisko VARCHAR(45) NOT NULL,  
    PESEL VARCHAR(45) UNIQUE NOT NULL,  
    PRIMARY KEY (Bibliotekarz_id));  
  
CREATE TABLE IF NOT EXISTS biblioteka.Czytelnik (  
    Czytelnik_id INT AUTO_INCREMENT NOT NULL,  
    Imie VARCHAR(45) NOT NULL,  
    Nazwisko VARCHAR(45) NOT NULL,  
    Telefon VARCHAR(45) UNIQUE NOT NULL,  
    email VARCHAR(45) UNIQUE NOT NULL,  
    Data_dodania DATE,  
    PRIMARY KEY (Czytelnik_id));
```

### Skrypt wypełniający bazę danymi:

#### Wycinek 2. Fragment skryptu wypełniającego bazę danymi

```
INSERT INTO biblioteka.bibliotekarz (Imie, Nazwisko, PESEL)  
VALUES ('Jan', 'Kowalski', '12345678901'),  
       ('Jan', 'Nowak', '23456789012'),  
       ('Łukasz', 'Kowalski', '34567890123'),  
       ('Łukasz', 'Nowak', '45678901234'),  
       ('Łukasz', 'Kowalski', '56789012345'),  
       ('Tomek', 'Kowalski', '67890123456'),  
       ('Tomek', 'Nowak', '78901234567');  
  
INSERT INTO biblioteka.autorzy (Imie, Nazwisko)  
VALUES ('Juliusz', 'Słowacki'),  
       ('Adam', 'Mickiewicz'),  
       ('Bolesław', 'Prus'),  
       ('Eliza', 'Orzeszkowa');
```

## Funkcje bazodanowe:

### Wycinek 3. Dodanie czytelnika

Procedura dodająca nowego czytelnika do bazy danych. Jako argumenty przyjmuje imię, nazwisko, telefon i email czytelnika.

```
DELIMITER $$
CREATE PROCEDURE biblioteka.NowyCzytelnik(
    Imie VARCHAR(45),
    Nazwisko VARCHAR(45),
    Telefon VARCHAR(45),
    email VARCHAR(45)
)
BEGIN
    START TRANSACTION;
    INSERT INTO biblioteka.czytelnik(Imie, Nazwisko, Telefon, email, Data_dodania)
    VALUE (Imie, Nazwisko, Telefon, email, CURRENT_DATE());
    COMMIT;
END $$
DELIMITER ;
```

### Wycinek 4. Dodanie bibliotekarza

Procedura odpowiadająca za dodanie nowego bibliotekarza do bazy danych. Jako argumenty przyjmuje imię, nazwisko i PESEL.

```
DELIMITER $$
CREATE PROCEDURE biblioteka.NowyBibliotekarz(
    Imie VARCHAR(45),
    Nazwisko VARCHAR(45),
    PESEL VARCHAR(45)
)
BEGIN
    START TRANSACTION;
    INSERT INTO biblioteka.bibliotekarz(Imie, Nazwisko, PESEL)
    VALUE (Imie, Nazwisko, PESEL);
    COMMIT;
END $$
DELIMITER ;
```



### Wycinek 5. Dodanie nowej książki

Procedura dodająca nową książkę do bazy danych. Jako argumenty przyjmuje kod ISBN, tytuł, autora, wydawnictwo, datę wydania oraz dopuszczalny maksymalny czas wypożyczenia.

```
DELIMITER $$
CREATE PROCEDURE biblioteka.DodajKsiazke(
    ISBN VARCHAR(45),
    Tytul VARCHAR(45),
    Autor VARCHAR(45),
    Wydawnictwo VARCHAR(45),
    Data_wydania DATE,
    Maksymalny_czas_wypozyczenia INT
)
BEGIN
    DECLARE wyd_id INT;
    DECLARE space_pos INT;
    DECLARE aut_id INT;
    START TRANSACTION;
    IF NOT EXISTS (SELECT * FROM biblioteka.wydawnictwo WHERE wydawnictwo.Nazwa =
        Wydawnictwo) THEN
        INSERT INTO biblioteka.wydawnictwo(Nazwa) VALUE (Wydawnictwo);
    END IF;

    SET space_pos = INSTR(Autor, ' ');

    IF NOT EXISTS (SELECT * FROM biblioteka.autorzy WHERE CONCAT(autorzy.Imie, '',
        autorzy.Nazwisko) = Autor) THEN
        INSERT INTO biblioteka.autorzy(Imie, Nazwisko) VALUE (SUBSTRING(Autor, 1, space_pos - 1),
            SUBSTRING(Autor, space_pos + 1));
    END IF;

    SELECT Wydawnictwo_id INTO wyd_id FROM wydawnictwo WHERE wydawnictwo.Nazwa =
        Wydawnictwo;

    INSERT INTO biblioteka.ksiazka(ISBN, Tytul, Wydawnictwo_id, Data_wydania,
        Liczba_egzemplarzy, Maksymalny_czas_wypozyczenia)
        VALUE (ISBN, Tytul, wyd_id, Data_wydania, 0, Maksymalny_czas_wypozyczenia);

    SELECT Autor_id INTO aut_id FROM biblioteka.autorzy WHERE CONCAT(autorzy.Imie, '',
        autorzy.Nazwisko) = Autor;

    INSERT INTO biblioteka.autorzy_ksiazki (ISBN, Autor_id) VALUE (ISBN, aut_id);

    COMMIT;
END $$
DELIMITER ;
```



### Wycinek 6. Dodanie nowego egzemplarza danej książki

Procedura dodająca nowy egzemplarz podanej książki. Argumenty to kod ISBN, oraz możliwy dodatkowy opis.

```
DELIMITER $$
CREATE PROCEDURE biblioteka.DodajEgzemplarz(
    ISBN VARCHAR(45),
    Dodatkowy_opis VARCHAR(45)
)
BEGIN
    DECLARE Nowa_liczba_egzemplarzy INT;
    START TRANSACTION;
    INSERT INTO biblioteka.egzemplarz(ISBN, Status_id, Dodatkowy_opis)
    VALUE (ISBN, 2, Dodatkowy_opis);

    SELECT Liczba_egzemplarzy + 1 INTO Nowa_liczba_egzemplarzy FROM ksiazka WHERE
    ksiazka.ISBN = ISBN;

    UPDATE ksiazka SET Liczba_egzemplarzy = Nowa_liczba_egzemplarzy WHERE ksiazka.ISBN =
    ISBN;
    COMMIT;
END $$
DELIMITER ;
```

### Wycinek 7. Wypożyczenie danej książki

Procedura umożliwiająca wypożyczenie danej książki. Argumenty to kod ISBN książki, PESEL bibliotekarza który wydaje książkę oraz email czytelnika wypożyczającego książkę.

```
DELIMITER $$
CREATE PROCEDURE biblioteka.WypozyczKsiazke(
    ISBN VARCHAR(45),
    PESEL VARCHAR(45),
    email VARCHAR(45)
)
BEGIN
    DECLARE Wypozyczany_egzemplarz_id INT;
    DECLARE bib_id INT;
    DECLARE czyt_id INT;
    DECLARE Data_zwrotu INT;
    DECLARE wyp_id INT;
    START TRANSACTION;
    SELECT Egzemplarz_id INTO Wypozyczany_egzemplarz_id FROM egzemplarz WHERE
    egzemplarz.ISBN = ISBN AND egzemplarz.Status_id = 2 LIMIT 1;
    SELECT Bibliotekarz_id INTO bib_id FROM bibliotekarz WHERE bibliotekarz.PESEL = PESEL;
    SELECT Czytelnik_id INTO czyt_id FROM czytelnik WHERE czytelnik.email = email;
    SELECT Maksymalny_czas_wypozyczenia INTO Data_zwrotu FROM ksiazka WHERE
    ksiazka.ISBN = ISBN;

    INSERT INTO wypozyczenia(ISBN, Bibliotekarz_id, Czytelnik_id, Data_wypozyczenia,
    Wymagana_data_zwrotu, Status_wypozyczenia_id)
    VALUE (ISBN, bib_id, czyt_id, CURRENT_DATE(), DATE_ADD(CURRENT_DATE(), INTERVAL
    Data_zwrotu DAY), 1);

    SELECT MAX(Wypozyczenie_id) INTO wyp_id FROM wypozyczenia;

    UPDATE egzemplarz SET Wypozyczenie_id = wyp_id, Status_id = 1 WHERE
    egzemplarz.Egzemplarz_id = Wypozyczany_egzemplarz_id;

    COMMIT;
END $$
DELIMITER ;
```

### Wycinek 8. Oddanie książki

Procedura odpowiadająca za oddanie książki. Argumenty to kod ISBN oddawanej książki oraz email czytelnika.

```
DELIMITER $$
CREATE PROCEDURE biblioteka.OddajKsiazke(
    ISBN VARCHAR(45),
    Email VARCHAR(45)
)
BEGIN
    DECLARE wyp_id INT;
    START TRANSACTION;
    SELECT MIN(Wypozyczenie_id) INTO wyp_id FROM wypozyczenia WHERE wypozyczenia.ISBN
    = ISBN AND wypozyczenia.Czytelnik_id = (SELECT Czytelnik_id FROM biblioteka.czytelnik
    WHERE czytelnik.email = Email) AND wypozyczenia.Status_wypozyczenia_id = 1;

    UPDATE biblioteka.wypozyczenia SET Rzeczywista_data_zwrotu = CURRENT_DATE(),
    Status_wypozyczenia_id = 2 WHERE wypozyczenia.ISBN = ISBN AND
    wypozyczenia.Czytelnik_id = (SELECT Czytelnik_id FROM biblioteka.czytelnik WHERE
    czytelnik.email = Email) AND wypozyczenia.Status_wypozyczenia_id = 1;

    UPDATE biblioteka.egzemplarz SET Status_id = 2, Wypozyczenie_id = NULL WHERE
    egzemplarz.Wypozyczenie_id = wyp_id;
    COMMIT;
END $$
DELIMITER ;
```

### Przykładowe selecty:

#### Wycinek 9. Spis wszystkich książek

Za pomocą tego selecta baza zwraca tabelę z spisem wszystkich książek, ich wydawnictw i autorów.

```
SELECT k.ISBN, k.Tytul AS Tytuł, w.Nazwa AS Wydawnictwo, CONCAT(a.Imie, ' ', a.Nazwisko) AS Autor
FROM ksiazka k
INNER JOIN wydawnictwo w
    ON k.Wydawnictwo_id = w.Wydawnictwo_id
INNER JOIN autorzy a
    ON a.Autor_id = (SELECT Autor_id FROM autorzy_ksiazki WHERE autorzy_ksiazki.ISBN = k.ISBN);
```

#### **Wycinek 10. Spis czytelników którzy mają wypożyczone książki**

Select zwracający tabelę z wszystkimi czytelnikami którzy mają wypożyczoną chociaż 1 książkę. Dodatkowe informacje to tytuł wypożyczonej książki oraz wymagana data zwrotu.

```
SELECT cz.Imie, cz.Nazwisko, cz.email, k.Tytuł AS Tytuł, w.Wymagana_data_zwrotu AS Wymagana data zwrotu
FROM czytelnik cz
INNER JOIN wypozyczenia w
    ON w.Czytelnik_id = cz.Czytelnik_id
INNER JOIN ksiazka k
    ON k.ISBN = w.ISBN;
```

#### **Wycinek 11. Lista książek wypożyczonych przez danego czytelnika**

Select działający podobnie do poprzedniego ale zwracający tylko listę wypożyczonych książek przez danego czytelnika.

```
SELECT cz.Imie, cz.Nazwisko, cz.email, k.Tytuł AS Tytuł, w.Wymagana_data_zwrotu AS 'Wymagana data zwrotu'
FROM czytelnik cz
INNER JOIN wypozyczenia w
    ON w.Czytelnik_id = cz.Czytelnik_id
INNER JOIN ksiazka k
    ON k.ISBN = w.ISBN
WHERE cz.Czytelnik_id = 15;
```

#### **Wycinek 12. Ilość dostępnych egzemplarzy danej książki**

Select zwracający liczbę dostępnych czyli nie wypożyczonych egzemplarzy danej książki.

```
SELECT k.ISBN, k.Tytuł AS Tytuł, w.Nazwa AS Wydawnictwo, COUNT(*) AS 'Liczba dostępnych egzemplarzy'
FROM ksiazka k
INNER JOIN wydawnictwo w
    ON k.Wydawnictwo_id = w.Wydawnictwo_id
INNER JOIN egzemplarz e
    ON k.ISBN = e.ISBN
WHERE k.ISBN = '9788328352341' AND e.Status_id = 2
```

### **3. Wnioski**

Bazy danych są aktualnie jednym z najważniejszych części aplikacji. Źle zaprojektowana baza może wpłynąć na wydajność i skalowalność aplikacji.

## 4. Spis rysunków i tabel

1. Rysunek 1 Diagram ERD	4
2. Wycinek 1. Fragment skryptu tworzącego bazę	6
3. Wycinek 2. Fragment skryptu wypełniającego bazę danymi	6
4. Wycinek 3. Dodanie czytelnika	7
5. Wycinek 4. Dodanie bibliotekarza	7
6. Wycinek 5. Dodanie nowej książki	8
7. Wycinek 6. Dodanie nowego egzemplarza danej książki	9
8. Wycinek 7. Wypożyczenie danej książki	10
9. Wycinek 8. Oddanie książki	11
10. Wycinek 9. Spis wszystkich książek	11
11. Wycinek 9. Spis czytelników którzy mają wypożyczone książki	12
12. Wycinek 9. Lista książek wypożyczonych przez danego czytelnika	12
13. Wycinek 9. Lista książek wypożyczonych przez danego czytelnika	12