

Dokumentacja projektu wykonywanego w ramach zajęć BAZY DANYCH I

Wiktoria Szewczyk

1. Projekt koncepcji, założenia

Projekt polega na stworzeniu aplikacji opartej na bazie danych PostgreSQL dla biblioteki. Do aplikacji można zalogować się jako pracownik biblioteki lub czytelnik.

Pracownik ma możliwość:

- dodawania książek lub usuwania i modyfikacji już istniejących,
- dodawania i usuwania egzemplarzy książek,
- dodawanie autorów lub usuwanie i modyfikacja już istniejących,
- dodawanie wydawnictw lub usuwanie i modyfikacja już istniejących
- dodawanie dziedzin (kategorii książki) lub usuwanie i modyfikacja już istniejących
- dodawanie nowego pracownika

Czytelnik może:

- przeglądać i przeszukiwać katalog książek
- jeśli to możliwe, wypożyczyć egzemplarz książki
- oddać egzemplarz książki
- zarezerwować książkę, lub zrezygnować z rezerwacji
- przeglądać historię swoich wypożyczeń
- opłacić karę naliczoną za przetrzymanie książki

2. Projekt diagramów (konceptualny)

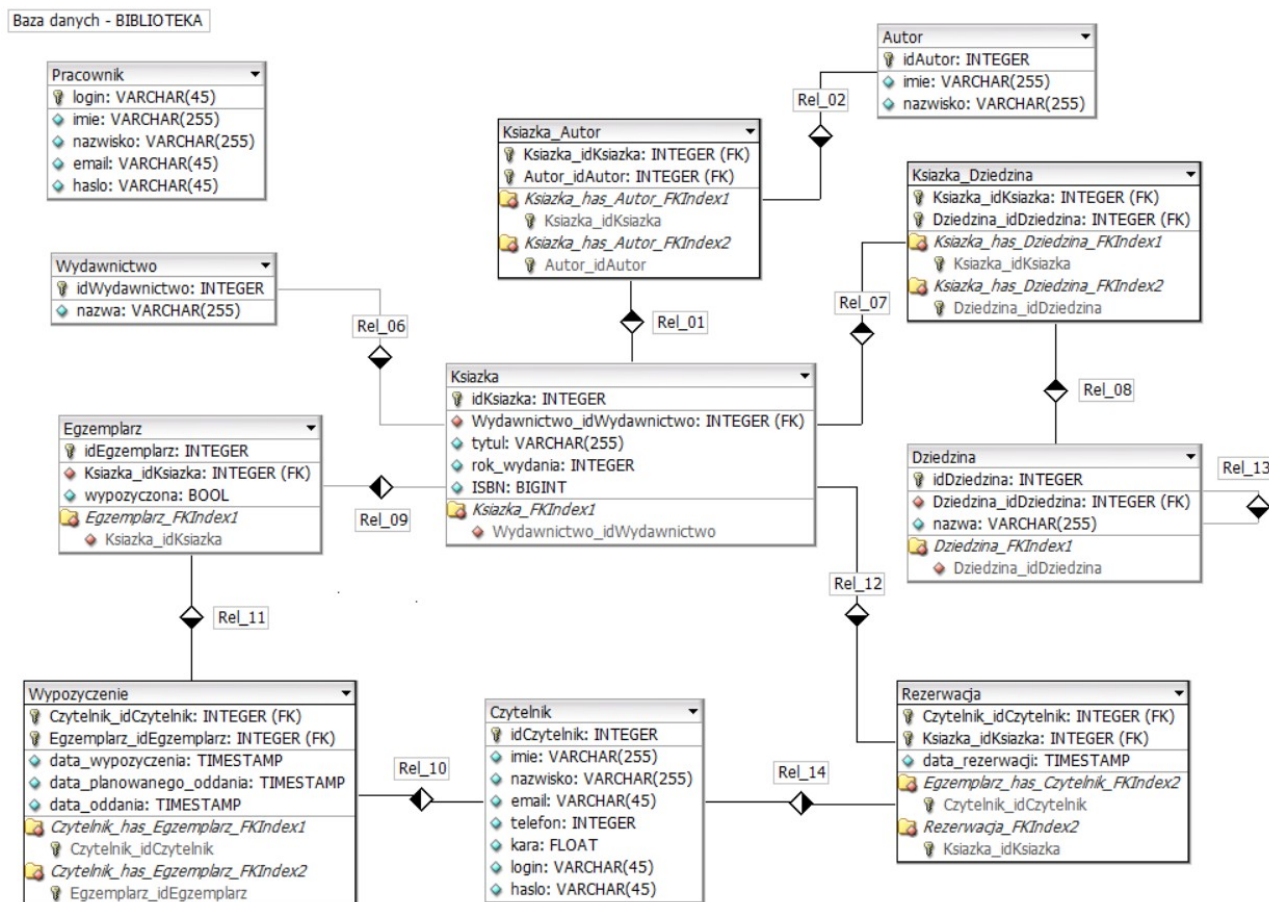


Figura 1: Projekt ERD bazy.

3. Projekt logiczny

Encja Pracownik

Przechowuje informacje na temat pracownika biblioteki.

- **login** – login pracownika, klucz główny encji,
- **imie** – imię pracownika,
- **nazwisko** – nazwisko pracownika,
- **email** – adres e-mail pracownika (zaimplementowana walidacja poprawności – tylko format aa123@bb.cc)
- **haslo** – hasło pracownika.

Encja Wydawnictwo

Przechowuje informacje o wydawnictwie. Jedno wydawnictwo może wydać wiele książek.

- **idWydawnictwo** – automatycznie inkrementowany klucz główny encji,
- **nazwa** – nazwa wydawnictwa.

Encja Dziedzina

Przechowuje informacje na temat dziedzin (kategorii). Zakładam że książka może przynależeć do więcej niż jednej dziedziny dlatego ta encja łączy się z encją Książka_Dziedzina. Dodatkowo dziedzina może mieć dziedziny pochodne (np. dziedziną pochodną do beletrystyki jest kryminał lub fantasy), stąd związek rekurencyjny w encji.

- **idDziedzina** – automatycznie inkrementowany klucz główny encji,
- **Dziedzina_idDziedzina** – klucz obcy dziedziny nadrzędnej (związek rekurencyjny),
- **nazwa** – nazwa dziedziny.

Encja Autor

Przechowuje informacje na temat autorów. Jedna książka może mieć wielu autorów dlatego ta encja łączy się z encją Książka_Autor.

- **idAutor** – automatycznie inkrementowany klucz główny encji,
- **imie** – imię autora,
- **nazwisko** – nazwisko autora.

Encja Czytelnik

Przechowuje informację na temat czytelnika – czyli osoby korzystającej z biblioteki.

- **idCzytelnik** – automatycznie inkrementowany klucz główny encji,
- **imie** – imię czytelnika,
- **nazwisko** – nazwisko czytelnika,
- **email** – adres e-mail czytelnika (zaimplementowana walidacja poprawności – tylko format aa123@bb.cc),
- **telefon** – numer telefonu czytelnika (zaimplementowana walidacja poprawności – 9 cyfr),
- **kara** – nieopłacona kara za przetrzymanie książek, domyślnie przyjmuje wartość 0,
- **login** – login użytkownika,
- **haslo** – hasło użytkownika.

Encja Książka

Przechowuje część informacji o książce (reszta informacji (autorzy i dziedziny książki) w encjach asocjacyjnych).

- **idKsiążka** - automatycznie inkrementowany klucz główny encji,
- **Wydawnictwo_idWydawnictwo** – klucz obcy określający wydawnictwo książki,
- **tytuł** – tytuł książki,
- **rok_wydania** – rok wydania książki (zaimplementowana walidacja poprawności – od roku 1800 do bieżącego),
- **ISBN** - *International Standard Book Number*, 13 cyfrowy numer unikalny dla każdej książki (zaimplementowana walidacja poprawności – 13 cyfr),

Encja Książka_Autor

Jest to encja asocjacyjna podporządkowująca autora książce (jedna książka może mieć wielu autorów). Kluczem głównym jest para Książka_idKsiążka, Autor_idAutor, dzięki czemu niemożliwe jest przypisanie dwa razy tego samego autora jednej książce.

- **Książka_idKsiążka** – klucz obcy określający książkę (część klucza głównego),
- **Autor_idAutor** – klucz obcy określający autora (część klucza głównego).

Encja Książka_Dziedzina

Jest to encja asocjacyjna podporządkowująca dziedzinę książce (jedna książka może mieć wiele dziedzin). Kluczem głównym jest para Książka_idKsiążka, Dziedzina_idDziedzina, dzięki czemu niemożliwe jest przypisanie dwa razy tej samej dziedziny jednej książce.

- **Książka_idKsiążka** – klucz obcy określający książkę (część klucza głównego),
- **Dziedzina_idDziedzina** – klucz obcy określający dziedzinę (część klucza głównego).

Encja Egzemplarz

Przechowuje informacje o egzemplarzu książki. Encja powstała ponieważ biblioteka może posiadać więcej niż jeden egzemplarz danej książki.

- **idEgzemplarz** - automatycznie inkrementowany klucz główny encji,
- **Ksiazka_idKsiazka** - klucz obcy określający książkę,
- **wypożyczona** – pole informujące o dostępności książki. Wartość *false* oznacza, że książka jest dostępna (nie jest wypożyczona), wartość *true* – książka niedostępna (wypożyczona). Domyślna wartość to *false*.

Encja Rezerwacja

Przechowuje informacje na temat rezerwacji książek przez czytelników. Kluczem głównym jest para Czytelnik_idCzytelnik, Ksiazka_idKsiazka, dzięki czemu nie możliwe jest zarezerwowanie dwa razy tej samej książki przez jednego czytelnika. Czytelnicy rezerwując książkę czekają na dostępność jakiegokolwiek egzemplarza – dlatego w tej encji klucz obcy Ksiazka_idKsiazka odnosi się do książki a nie do egzemplarza.

- **Czytelnik_idCzytelnik** – klucz obcy odnoszący się do czytelnika (część klucza głównego),
- **Ksiazka_idKsiazka** – klucz obcy odnoszący się do książki (część klucza głównego),
- **data_rezerwacji** – data dokonania rezerwacji przez użytkownika, domyślnie aktualna data (zaimplementowana walidacja poprawności – data nie może być datą z przyszłości).

Encja Wypożyczenie

Przechowuje informację o istotnych datach dotyczących wypożyczenia egzemplarza przez czytelnika. Kluczem głównym jest grupa Czytelnik_idCzytelnik, Ksiazka_idKsiazka, data_wypożyczenia, dzięki czemu czytelnik może wypożyczyć ten sam egzemplarz wielokrotnie (tylko w innym czasie).

- **Czytelnik_idCzytelnik** – klucz obcy odnoszący się do czytelnika (część klucza głównego),
- **Egzemplarz_idEgzemplarz** – klucz obcy odnoszący się do egzemplarza (część klucza głównego),
- **data_wypożyczenia** – data wypożyczenia egzemplarza przez czytelnika (część klucza głównego), domyślnie aktualna data (zaimplementowana walidacja poprawności – data nie może być datą z przyszłości, musi być wcześniejsza niż data_planowanego_oddania oraz data_oddania),
- **data_planowanego_oddania** – termin do którego czytelnik powinien oddać egzemplarz, domyślnie miesiąc po wypożyczeniu (zaimplementowana walidacja poprawności – data musi być późniejsza niż data_wypożyczenia),
- **data_oddania** – faktyczna data oddania egzemplarza przez czytelnika, domyślnie przyjmuje wartość NULL, która oznacza, że egzemplarz nie został oddany, (zaimplementowana walidacja poprawności – data nie może być z datą przyszłości oraz musi być późniejsza niż data_wypożyczenia).

Aby dane w bazie były spójne zostały zdefiniowane następujące wyzwalacze:

- **wypożyczenie_insert** – wykonuje funkcję **nowe_wypożyczenie()**, przy próbie dodania rekordu do encji **Wypożyczenie**. Funkcja sprawdza czy czytelnik może wypożyczyć dany egzemplarz (czy jest dostępny i czy czytelnik nie ma za wysokiej (> 10) kary). Jeśli nie może, rekord nie zostaje dodany do encji. W przeciwnym wypadku rekord zostaje dodany oraz wartość pola *wypożyczona*, w odpowiednim rekordzie encji *Egzemplarz*, ustawiana jest na *true*. Dodatkowo jeśli użytkownik rezerwował wcześniej książkę, której egzemplarz wypożycza, informacja o tej rezerwacji zostaje usunięta z encji *Rezerwacja*.

- **wypozyczenie_update_oddanie** – wykonuje funkcję **oddanie_książki()**, przy próbie zmiany rekordu w encji **Wypozyczenie**. Jeżeli zmianie podlega pole *data_oddania* to wartość pola *wypozyczona*, w odpowiednim rekordzie encji *Egzemplarz*, ustawiana jest na false. Dodatkowo wykonywana jest funkcja *dodaj_kara* opisana poniżej.
- **rezerwacja_insert** – wykonuje funkcję **nowe_rezerwacja()**, przy próbie dodania rekordu do encji **Rezerwacja**. Uniemożliwia rezerwację już wypożyczonej książki.
- **ksiazka_delete** – wykonuje funkcję **usuwanie_książki()**, przy próbie usunięcia rekordu z encji **Książka**. Usunięcie książki jest możliwe tylko jeśli nie ma w bazie jej egzemplarzy. Wtedy usuwane są też informacje o książce z encji *Książka_Dziedzina*, *Książka_Autor* i *Rezerwacja*.
- **egzemplarz_delete** – wykonuje funkcję **usuwanie_egzemplarza()**, przy próbie usunięcia rekordu z encji **Egzemplarz**. Usunięcie egzemplarza jest możliwe tylko jeśli nie jest on wypożyczony. Wtedy usuwane są też informacje o wypożyczeniach (z encji *Wypozyczenie*) o danym egzemplarzu.

Dodatkowo zostały stworzone następujące funkcje:

- **logowanie_pracownik(l text, p text)** – funkcja przyjmuje kolejno dwa argumenty – login i hasło. Sprawdza czy w encji *Pracownik* istnieje rekord o podanym loginie i hasle. Jeśli tak zwraca wartość 1, jeśli nie - wartość 0. Funkcja jest wykorzystywana w programie podczas prób logowania jako pracownik.
- **logowanie_czytelnik(l text, p text)** – funkcja przyjmuje kolejno dwa argumenty – login i hasło. Sprawdza czy w encji *Czytelnik* istnieje rekord o podanym loginie i hasle. Jeśli tak zwraca wartość *idCzytelnik*, jeśli nie - wartość 0. Funkcja jest wykorzystywana w programie podczas prób logowania jako czytelnik, a zwracana wartość pozwala jednoznacznie zidentyfikować zalogowanego czytelnika.
- **dostepnosc_ksiazki(idK int)** – funkcja przyjmuje jako argument wartość *idKsiążka* i sprawdza czy jest ona dostępna. Jeśli istnieje chociaż jeden niewypożyczony egzemplarz książki, funkcja zwraca wartość pola *idEgzemplarz*. W przeciwnym wypadku zwraca 0.
- **rezerwacja_dostepnosc(idK int, idC int)** – funkcja przyjmuje dwa argumenty – *idKsiążka* i *idCzytelnik*. Funkcja sprawdza czy zarezerwowana przez czytelnika książka jest już dostępna oraz czy czytelnik ma pierwszeństwo wypożyczenia (czy żaden czytelnik nie zarezerwował jej wcześniej). Jeśli warunki zostają spełnione funkcja zwraca *idEgzemplarz* dostępnego egzemplarza. W przeciwnym wypadku zwraca 0.
- **dodaj_kara(idC int, idE int)** – funkcja przyjmuje dwa argumenty – *idCzytelnik* i *idEgzemplarz*. Jeśli książka została przetrzymana nalicza czytelnikowi karę (0.1 za każdy dzień przetrzymania).

4. Projekt funkcjonalny

Na potrzeby programu zostały stworzone następujące widoki:

- **dziedziny_naddziedziny** – informacje o dziedzinach i ich dziedzinach nadrzędnych (nazwy i klucze główne),
- **dziedziny_książki** – informacje o książce i jej wszystkich dziedzinach,
- **książka_info** – wszystkie informacje o książce (autor, tytuł, rok wydania, ISBN, wydawnictwo, autorzy, dziedziny),
- **egzemplarz_info** – wszystkie informacje o egzemplarzu (takie same jak o książce plus informacja czy jest wypożyczony),
- **wypożyczone_egzemplarze_nieoddane** – wszystkie informacje o nieoddanych egzemplarzach,
- **wypożyczone_egzemplarze_oddane** – wszystkie informacje o oddanych egzemplarzach,
- **zarezerwowane_książki** – wszystkie informacje o zarezerwowanej książce (takie same jak o książce plus klucz obcy definiujący czytelnika rezerwującego)

Widoki te są wykorzystywane w programie np. podczas wyświetlania wypożyczonych, oddanych lub zarezerwowanych przez czytelnika książek lub podczas dodawania, usuwania lub modyfikacji informacji w różnych encjach jako listy z których można wybierać odpowiednie pola.

Instrukcja obsługi aplikacji

Jako niezalogowany użytkownik aplikacji mamy możliwość przeglądania oraz wyszukiwania (po autorze, tytule lub dziedzinie) książek zapisanych w bazie. Możemy także zarejestrować się jako czytelnik.

Po zalogowaniu się jako czytelnik (login: test, hasło: test) mamy dostęp do menu czytelnika. W zakładce Moje dane możemy przeglądać informacje zalogowanego czytelnika oraz opłacić karę. Sekcja Wypożyczone wyświetla listę wypożyczonych książek. Po kliknięciu w wybraną pozycję wyświetlają się dodatkowe informacje o książce oraz pojawia się możliwość oddania książki. Sekcja Zarezerwowane działa podobnie. Mamy tam możliwość zrezygnowania z rezerwacji, oraz wypożyczenia książki jeśli jest dostępna. Historia wypożyczeń wyświetla książki już oddane wraz z datą wypożyczenia. W zakładce Katalog książek po wybraniu interesującej nas pozycji możemy ją wypożyczyć lub zarezerwować. Dostępny jest też przycisk „Wyloguj” przenoszący nas automatycznie do widoku startowego aplikacji.

Po zalogowaniu się jako pracownik (login: admin, hasło: admin) przechodzimy do menu pracownika. Zakładka Pracownicy pozwala na dodanie nowego pracownika. Sekcja Autor pozwala na dodawanie, usuwanie i modyfikację danych w encji *Autor*. Aby usunąć autora należy wybrać go z listy i kliknąć przycisk „Usuń”. Aby zaktualizować autora, także należy wybrać go z listy (wtedy automatycznie formularz lewym dolnym rogu wypełnia się informacjami na jego temat), zmienić dane w formularzu i kliknąć przycisk „Aktualizuj”. Zakładki Wydawnictwo i Dziedzina wyglądają podobnie a proces dodawania, usuwania i aktualizacji jest analogiczny. Zakładka Książka działa podobnie jednak po kliknięciu przycisku „Dodaj” lub „Aktualizuj” zostajemy przeniesieni do panelów w których możemy dodać lub usunąć dowolną liczbę autorów do książki. Po kliknięciu przycisku „Dalej” przenosimy się do podobnego panelu w którym dodajemy lub usuwamy dziedziny książki. Klikamy „Zakończ” aby wrócić do zakładki Książki. Sekcja *Egzemplarz* wyświetla listę wszystkich książek. Po kliknięciu w interesujący nas tytuł możemy dodawać lub usuwać egzemplarze. Tak samo jak w przypadku użytkownika dostępny jest przycisk „Wyloguj” przenoszący nas do widoku startowego aplikacji.

5. Dokumentacja

W katalogu SQL projektu znajdują się pliki:

- **create.sql** – zawiera polecenia pozwalające na utworzenie encji,
- **drop.sql** – zawiera polecenia pozwalające na usunięcie tabel oraz funkcji, wyzwalaczy i widoków odnoszących się do nich,
- **function.sql** – zawiera wszystkie funkcję opisane wyżej,
- **insert.sql** – zawiera przykładowe polecenia dodające dane do wszystkich encji,
- **trigger.sql** – zawiera wszystkie wyzwalacze opisane wyżej,
- **view.sql** – zawiera wszystkie widoki opisane wyżej.

W pliku GUI znajdują się pliki źródłowe aplikacji.

Aplikacja została napisana za pomocą języka Java (wersja 17.0.1) z wykorzystaniem biblioteki Swing (warstwa wizualna) oraz interfejsu JDBC (Java DataBase Connectivity). Do projektu został dołączony zarówno plik jar oraz plik wykonywalny aplikacji.