# Dokumentacja – Projekt na przedmiot Bazy Danych 1

## **Aplikacja Baza Danych Teatru w PostgreSQL**

Adam Młyńczak, Informatyka Stosowana, WFiIS

1. Projekt koncepcji
   1. Temat projektu i założenia

Aplikacja została stworzona z myślą o organizacji teatru oraz spektakli które mają tam być wystawiane. Pozwala ona na organizację oraz przyporządkowanie odpowiednich osób do odpowiednich sztuk teatralnych i terminów ich wystawiania.

Aplikacja ma pozwalać klientowi/widzowi zobaczyć aktualny harmonogram i/lub dodatkowe informacje na temat wybranej przez niego sztuki. Dodatkowo powinien móc on zobaczyć kto występuje w danym teatrze lub kto w tym teatrze odpowiada za reżyserię przedstawień.

Ze strony pracowniczej aplikacja powinna umożliwiać wprowadzać nowych aktorów/reżyserów, przypasowywać ich do odpowiednich spektakli, które również ten pracownik powinien móc tworzyć i ustalać, kiedy się odbędą.

* 1. Określenie funkcjonalności aplikacji

W aplikacji użytkownik może robić to co jest w założeniach – w panelu klienta są to:

1. Pobranie Harmonogramu, wyświetlenie informatora dla danej sztuki, ewentualne zapisanie go w formie pliku lub zakup biletów na wybrany spektakl,
2. Zobaczenie listy aktorów, którzy występują na deskach teatru oraz wyświetlenie listy przedstawień, w których dany aktor występuje,
3. Wyświetlenie reżyserów tegoż teatru, wraz z sztukami, które reżyserują.

Jeśli chodzi o panel pracownika, poza funkcjonalnościami „zwykłego” użytkownika dodatkowo można:

1. Dodać nową sztukę do bazy danych,
2. Dodać obsadę do wybranej sztuki,
3. Ustalić termin, kiedy jakie przedstawienie ma odbywać,
4. „Zatrudnienie” nowego aktora – dodanie jego informacji do bazy danych,
5. Dodanie nowego reżysera do bazy,
6. Pobranie informacji o zamówieniach klientów odnośnie biletów na spektakle
7. Reset bazy danych.
8. Projekt diagramów
   1. Przepływ danych

Ze strony bazy danych istnieje tylko jeden użytkownik (przełączanie pomiędzy klient-pracownik odbywa się ze strony aplikacji). Wszystkie dane przechowywane są na serwerze ElephantSQL, a co za tym idzie przepływ informacji odbywa się pomiędzy użytkownikiem oraz serwerem.

* 1. Encje oraz ich atrybuty

1. Encja Rezyser(id\_rezysera, imie\_rezysera, nazwisko\_rezysera) – informacje o reżyserze,
2. Encja Aktorzy(id\_aktora, imie, nazwisko) – informacje o aktorze,
3. Encja SztukiTeatralne(id\_sztuki, tytul\_sztuki, informator, id\_rezysera) – podstawowe informacje o sztuce (wraz z krótkim opisem oraz id\_rezysera),
4. Encja ObsadaSztuki(id\_obsady\_sztuki, id\_sztuki, id\_aktora, postac) – jaki aktor, w jakiej sztuce, gra jaką postać,
5. Encja TerminyRealizacji(id\_terminu, id\_sztuki, data\_realizacji, miejsce\_realizacji, dostępne\_bilety, cena\_ulgowy, cena\_normalny) – jaka sztuka, kiedy jest wystawiana, gdzie jest wystawiana i informacja o bieltach,
6. Encja ZamowieniaBiletow(id\_zamowienia, id\_terminu, ilosc\_biletow\_ulgowe, ilość\_biletow\_normalne, data\_zamowienia) – tabela z zamówieniami biletów na konkretne przedstawienia, wraz z informacją o dacie, kiedy zostało złożone zamówienie.
   1. Relacje pomiędzy encjami

Relacja pomiędzy tabelami „SztukiTeatralne” oraz „Aktorzy” to wiele do wielu, dlatego została pomiędzy nie wstawiona tabela asocjacyjna „ObsadaSztuki”, w której przechowywane są informacje o tym który aktor (id\_aktora) gra w jakiej sztuce (id\_sztuki) – przechowuje klucze główne tych dwóch tabel, które łączy; dodatkowo przechowuje informację o granej postaci.

Pomiędzy tabelą „Rezyser” oraz „SztukiTeatralne” występuje relacja jeden do wielu, dlatego jedno z pól w tabeli „SztukiTeatralne” (id\_rezysera) wskazuje na osobę, która jest odpowiedzialna za reżyserię.

Tabele „TerminyRealizacji” oraz „SztukiTeatralne” to relacja wiele do jednego, dodatkowo pomiędzy „TerminyRealizacji” oraz „ZamowieniaBiletow” rówież występuje realacja 1:N.

Wszystko to zobrazowane jest na poniższym diagramie ERD.

1. Projekt logiczny
   1. Proejktowanie tabel, kluczy oraz indeksów

Kod w SQL, dzięki któremu tworzy się baza danych jest załączona w folderze /sql.

1. Rezyser

* id\_rezysera SERIAL – automatycznie przypisywany klucz główny
* imie\_rezysera VARCHAR(50) NOT NULL – imię reżysera
* naziwsko\_rezysera VARCHAR(50) NOT NULL– nazwisko reżysera

1. Aktorzy
   * id\_aktora SERIAL – automatycznie przypisywany klucz główny
   * imie VARCHAR (50) NOT NULL – imię aktora
   * nazwisko VARCHAR(50) NOT NULL – nazwisko aktora
2. SztukiTeatralne
   * id\_sztuki SERIAL – automatycznie przypisywany klucz główny
   * tytul\_sztuki VARCHAR(255) NOT NULL – tytuł wystawianej sztuki
   * informator TEXT – krótki opis sztuki
   * id\_rezysera INT – kto reżyseruje, klucz obcy (FK) z tabeli Rezyser
3. ObsadaSztuki
   * id\_obsady\_sztuki SERIAL - automatycznie przypisywany klucz główny
   * id\_sztuki INT NOT NULL – klucz obcy (FK) do sztuki
   * id\_aktora INT NOT NULL – klucz obcy (FK) do aktora
   * postac VARCHAR(255) NOT NULL – grana przez aktora postać w danej sztuce
4. TerminyRealizacji
   * id\_terminu SERIAL - automatycznie przypisywany klucz główny
   * id\_sztuki INT NOT NULL – klucz obcy (FK) określający dla jakiej sztuki jest to termin
   * data\_realizacji DATE NOT NULL – data realizacji sztuki
   * miejsce\_realizacji VARCHAR(100) NOT NULL – gdzie spektakl będzie się odbywał
   * dostępne\_bilety INT – ilość dostępnych miejsc na przedstawienie
   * cena\_ulgowy INT – cena za bilet ulgowy na dany spektakl
   * cena\_normalny INT – cena za bilet normalny na dany spektakl
5. ZamowieniaBiletow
   * id\_zamowienia SERIAL - automatycznie przypisywany klucz główny
   * id\_terminu INT– klucz obcy (FK), określający na jaki termin składane jest zamówienie
   * ilosc\_biletow\_ulgowe INT – ilość kupionych biletów ulgowych w zamównieniu
   * ilosc\_biletow\_normalne INT – ilość kupionych biletów normalny w zamówieniu
   * data\_zamowienia DATE – kiedy zamówienie zostało złożone
   1. Analiza zależności funkcyjnych

????

* 1. Zaprojektowanie operacji danych