Dokumentacja – Projekt na przedmiot Bazy Danych 1

Aplikacja Baza Danych Teatru w PostgreSQL

Adam Młyńczak 410702, Informatyka Stosowana, WFiIS

1. Projekt koncepcji

1.1. 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.

1.2. 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.

2. Projekt diagramów

2.1. Przepływ danych

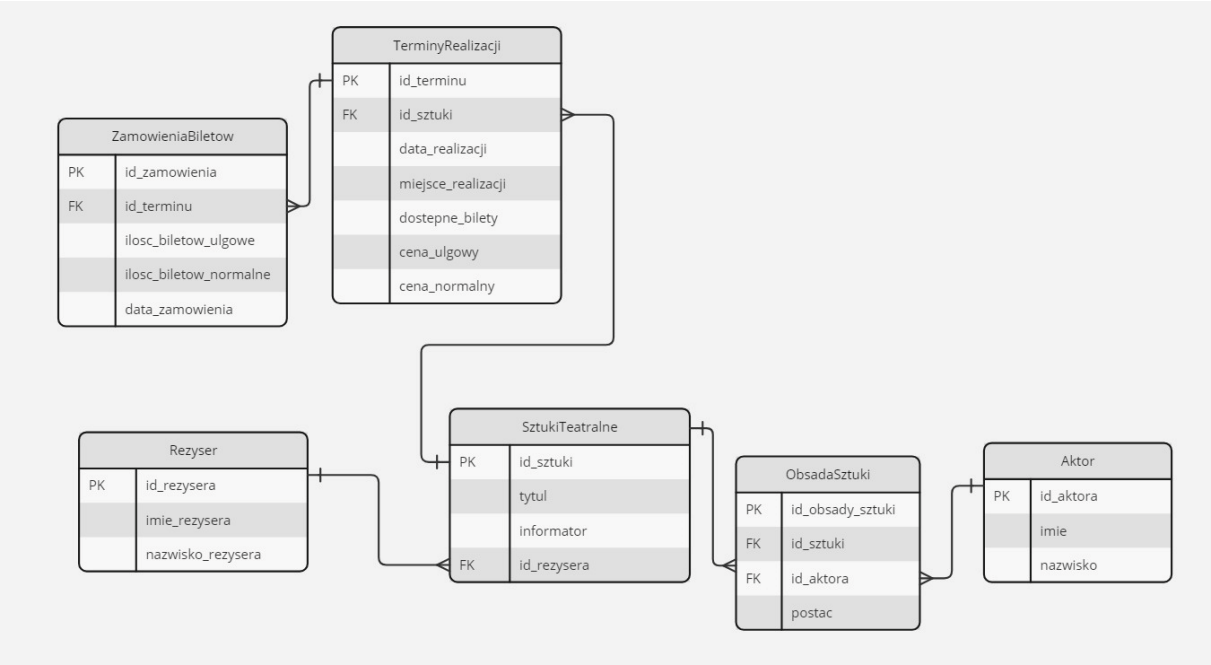
Ze strony bazy danych istnieje tylko jeden użytkownik (przełączanie pomiędzy klient-pracownik odbywa się ze strony aplikacji). Wszystkie dane przechowywane są na serwerze ElephantSQL, a co za tym idzie przepływ informacji odbywa się pomiędzy użytkownikiem oraz serwerem.

2.2. 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.

2.3. 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.



3. Projekt logiczny

3.1. 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
2) Aktorzy
    - id_aktora SERIAL – automatycznie przypisywany klucz główny
    - imie VARCHAR (50) NOT NULL – imię aktora
    - nazwisko VARCHAR(50) NOT NULL – nazwisko aktora
3) 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
4) 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

5) 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
6) 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

3.2. Zaprojektowanie operacji danych

Tak samo jak plik z tworzeniem bazy danych, tak i pliki odpowiadające za utworzenie widoków, funkcji oraz wyzwalaczy znajdują się w folderze /sql.

3.2.1. Widoki:

1) Harmonogram_app – wszystkie dane potrzebne do wyświetlenia harmonogramu teatru wraz z informatorem danej sztuki (tytuł, reżyser, informacje o biletach)
2) Aktorzy_app – informacje o jednym aktorze
3) Rezyser_app – informacje o jednym reżyserze
4) Zamowienia_app – dane do wyświetlenia tabeli z zamówieniami biletów
5) Ilosc_sprzedanych_biletow – informacje o ilości zamówionych już biletów

3.2.2. Funkcje:

1) SztukaPoId – zwracana jest tabela z pełnymi informacjami o sztuce, gdy znamy jej ID
2) SztukiDanegoAktora – otrzymujemy tabelę z sztukami (oraz postaciami w nich granymi) dla aktora, którego id podajemy
3) SztukiDanegoRezysera – taka sama funkcjonalność, jak dla Aktora
4) ObsadaPoId – otrzymujemy tabelę z obsadą dla danej pojedynczej sztuki

3.2.3. Wyzwalacze:

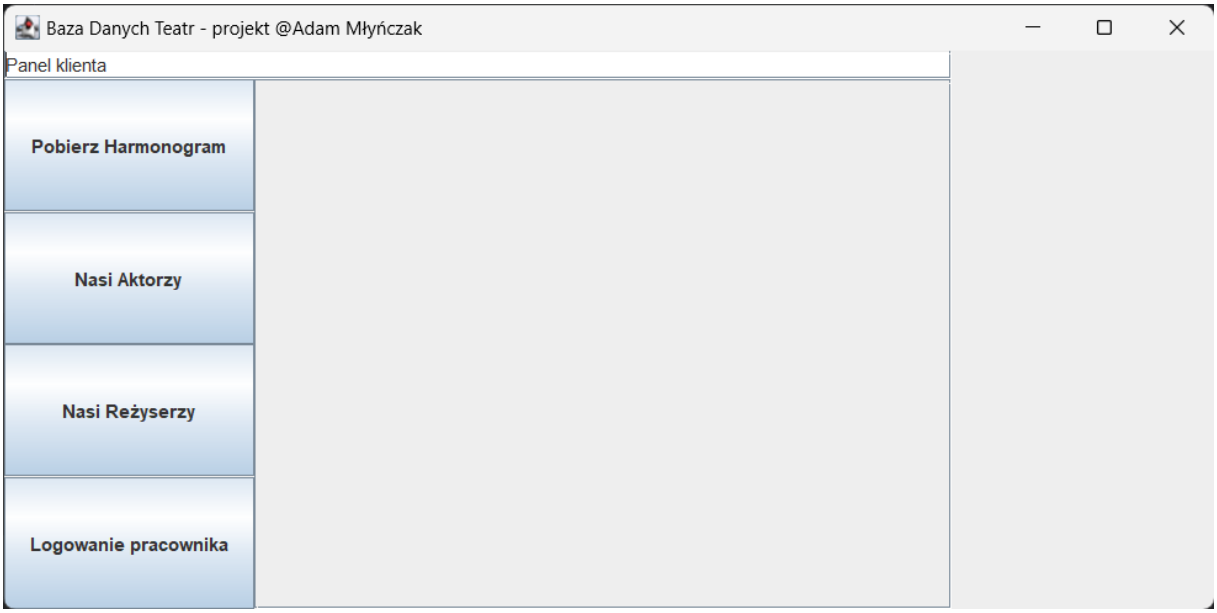
1) TRIGGER po_wstawieniu_do_ZamowieniaBiletow wraz z funkcją aktualizuj_liczbe_biletow – po zamówieniu przez użytkownika danej liczby biletów wyzwalacz aktualizuje ilość dostępnych biletów w tabeli TerminyRealizacji

3.2.4. Przykładowe dane

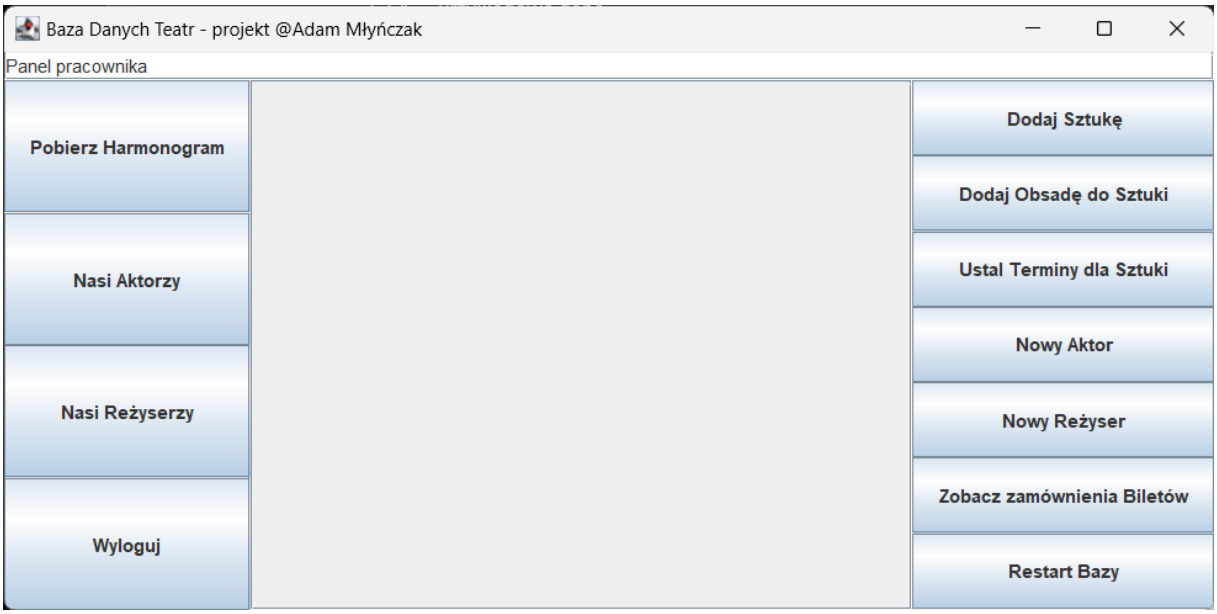
Dodatkowo w folderze /sql znajduje się plik data.sql, który definiuje zapytania typu INSERT, aby baza danych miała jakieś przykładowe dane przy pierwszym uruchomieniu.

4. Projekt funkcjonalny

4.1. Interfejs

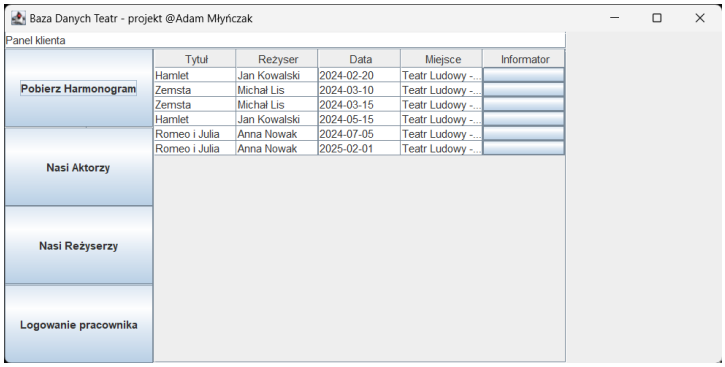


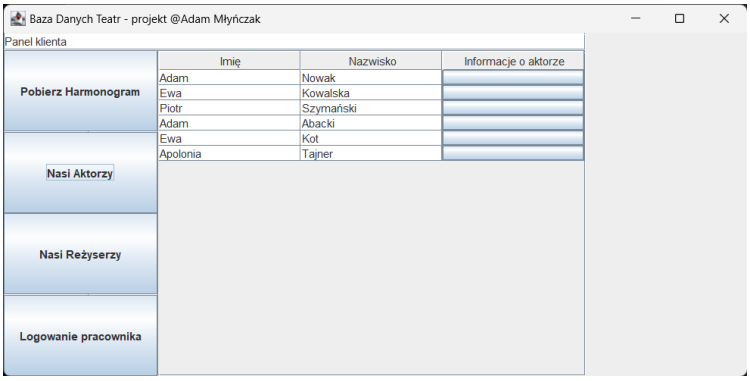
Główny panel aplikacji dla klienta znajduje się po lewej stronie okna, gdy zalogujemy się jako pracownik, wszystkie dodatkowe dla niego funkcje pojawiają się po prawej stronie.



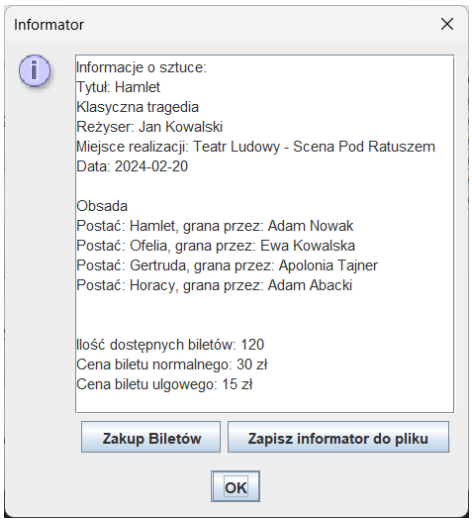
4.1.1. Przyciski klienta

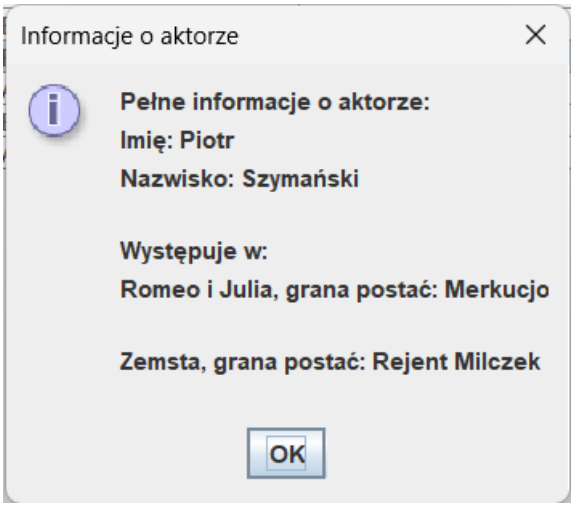
Nie ważne, który z przycisków klienta (panel po lewej stronie) klikniemy, na panelu po środku wyświetli nam się tabela z informacjami, które chcemy pobrać.





Po pobraniu informacji mamy możliwość sprawdzić dodatkowe informacje klikając przycisk w ostatniej kolumnie tabeli.



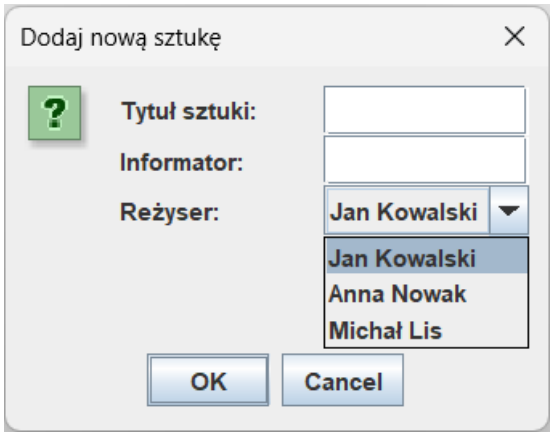


Przy oknie informatora dla sztuki mamy możliwość zapisania tego informatora w formie pliku (program poprosi nas o wskazanie lokalizacji) lub zakupu biletów – w następnym oknie wybieramy ile biletów ulgowych pragniemy zakupić i ile normalnych.

4.1.2. Przyciski pracownika

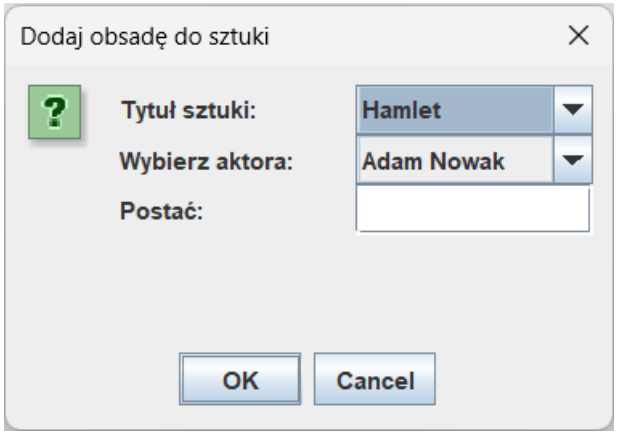
4.1.2.1.Dodaj Sztukę

Po wywołaniu tej funkcji pojawia nam się oknienko, w którym wpisujemy informacje potrzebne do utworzenia rekordu (reżysera wybieramy spośród zapisanych już w bazie danych).



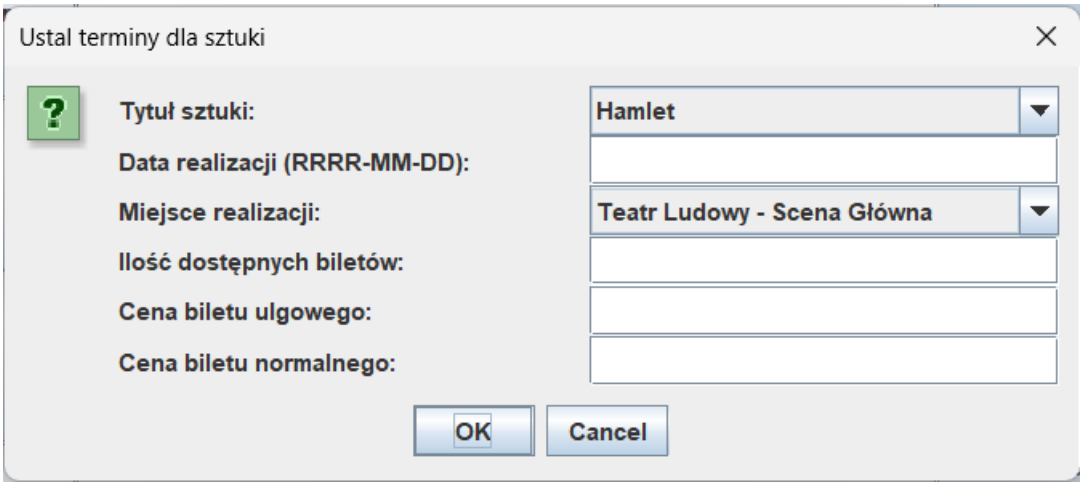
4.1.2.2.Dodaj Obsadę do Sztuki

W tej funkcji aplikacji wybieramy spośród utworzonych już wcześniej spektakli oraz z posiadanych już w bazie aktorów kombinację dla właściwej postaci, którą wpisujemy.



4.1.2.3.Ustal Terminy dla Sztuki

Tak samo jak w pierwszym przycisku, wypełniamy po prostu wszystkie informacje potrzebne do utworzenia rekordu.



4.1.2.4.Przyciski Nowy Aktor oraz Nowy Reżyser

Obie funkcje działają tak samo – pobierają informacje o imieniu oraz nazwisku nowego pracownika teatralnego.

4.1.2.5.Restart bazy

Po kliknięciu tego przycisku nie mamy do czynienia z twardym restartem, ponieważ przykładowe dane ponownie się wgrają (jest to kwestia usunięcia wykonania jednego pliku), jednak po jej uruchomieniu wszystkie dane, które my wpisaliśmy zostaną wyczyszczone.

5. Dokumentacja

Całość kodu tworzącego aplikację w języku Java znajduje się w katalogu /src, kod odpowiedzialny za tworzenie i dodawanie danych do bazy jest w ktalogu /sql. W katalogu out/artifacts/BD1_projekt_jar znajduje się plik wykonywalny jar dla projektu.

5.1. Wprowadzanie danych

Jak już wcześniej zostało wspomniane, przykładowe dane są wprowadzanie przy tworzeniu/restarcie bazy (sql/data.sql) jak również można wprowadzać dane poprzez aplikację. Dodatkowo w miejscach gdzie jest to możliwe, wprowadzane dane są automatycznie (PK, timestamps).

5.2. Literatura

1) Dokumentacja PostgreSQL.
2) Materiały przekazywane przez prowadzącą grupy na zajęcia laboratoryjne z przedmiotu.
3) Dokumentacja Oracle Help Center na temat Java Swing.