

## Baza danych kina z aplikacją kliencką umożliwia

użytkownikowi:

- dostęp do bazy filmów,
- dostęp do seansów filmowych które się jeszcze nie odbyły,
- zakup biletów na seans filmowy,
- informacje o sali kinowej gdzie ma odbywać się seans na konkretny film

administratorowi:

- wszystko to co użytkownikowi, dodatkowo:
- generowanie raportów dotyczących sprzedaży biletów oraz przychodów generowanych przez każdy film,
- tworzenie filmów i uzupełnianie ich o dodatkowe informacje takich jak np aktorzy w filmie, oceny,
- dostęp do sal kinowych,
- tworzenie seansów filmowych

## skrypty tworzące bazę danych:

### 1. baza danych w kontenerze docker

```
> docker run -d --name mssql -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD='haslo' -p 1433:1433 mcr.microsoft.com/mssql/server:2019-latest
```

### 2. tworzymy wirtualne środowisko poprzez

```
> python venv /sciezka/
```

### 3. i uruchamiamy je:

```
> source venv/bin/activate
```

### 4. > pip install Flask

```
> pip install pyodbc
```

### 5. uruchamiamy:

```
> python createTables.py
```

(kod tworzący tabele)

```
> python createProcedures.py
```

(kod tworzący procedury składowe dodające dane do bazy danych)

```
> python createFunctions.py
```

(kod tworzący funkcje i widoki zwracające tabele z bazy)

```
> python createTriggers.py
```

(kod tworzący wyzwalacze)

```
> python exampleData.py
```

(kod dodaje do bazy danych przykładowe tabele)

### 6. uruchamiamy aplikację

```
> export FLASK_APP=wyswietl_filmy.py
```

```
> export FLASK_DEBUG=1
```

```
> python -m flask run      lub      > python wyswietl_filmy.py
```

Mozemy się połączyć z serverem sql z terminala:

```
> mssql -u sa -p haslo
```

## przywracanie bazy danych z kopii zapasowej:

```
> docker exec -it mssql_docker mkdir /var/opt/mssql/backup  
> docker cp /sciezka/do/folderu/kopia.bak mssql_docker:/var/opt/mssql/backup  
> docker exec -it mssql_docker /opt/mssql-tools/bin/sqlcmd -S localhost -U [nazwa_uzytkown-  
ika] -P '[haslo]' -Q 'RESTORE FILELISTONLY FROM DISK = "/var/opt/mssql/backup/plik.bak"' tr  
-s ' ' | cut -d ' ' -f 1-2  
> docker exec -it mssql_docker /opt/mssql-tools/bin/sqlcmd -S localhost -U [nazwa_uzytkown-  
ika] -P '[haslo_uzytkownika]' -Q 'RESTORE DATABASE [plik] FROM DISK = "/sciezka/'
```

## widoki, funkcje:

- funkcja "l\_sprz\_bilet\_na\_seans" - zwraca liczbę sprzedanych biletów na seans o podanym id
- funkcja "wolne\_miejsca" - zwraca liczbę wolnych miejsc na seans o podanym id
- widok "wyswietl\_filmy" - zwraca tabę z podstawowymi informacjami o filmach z bazy
- funkcja "wyswietl\_szczegoly\_filmu\_po\_id" przyjmuje w argumencie id filmu i zwraca tabelę ze szczegółami filmu o podanym id
- funkcja "wyswietl\_aktorow\_filmu\_po\_id" przyjmuje parametr id filmu i zwraca tabele zawierającą informacje o aktorach występujących w filmie o podanym id
- funkcja "wyswietl\_seans\_po\_id" przyjmuje w parametrze id seansu i zwraca tabelę z informacjami o seansie filmowym
- funkcja "wyswietl\_seanse\_po\_id" przyjmuje w parametrze id filmu i zwraca tabelę z seansami na podany film, jeśli seans się skończył to nie wyświetla się
- funkcja "wyswietl\_sale\_po\_id" przyjmuje w argumencie id sali kinowej i zwraca tabele z informacjami o podanej sali
- widok "raport\_sprzedazy" zwraca tabelę z informacjami o sprzedanych biletach
- widok "wyswietl\_sale\_kinowe" zwraca tabelę z informacjami o wszystkich salach kinowych

## procedury składowe:

- procedura "dodaj\_zamowienie" przyjmuje w argumencie ilość biletów którą chcemy kupić, id seansu i dane o osobie która kupuje bilet
- procedura "dodaj\_seans" przyjmuje w argumencie dane filmu na który tworzymy seans i informacje o seansie takie jak jego data rozpoczęcia, zakończenia i cena za bilet. Sprawdza czy nie ma już seansu o podanych parametrach w bazie danych
- procedura "dodaj\_film" przyjmuje w argumencie podstawowe informacje o filmie takie jak tytuł, premiera i długość i dodaje film do bazy danych. sprawdza czy nie ma już filmu o podanych parametrach w bazie danych
- procedura "dodaj\_film\_szczegoly" służy dodaniu do już istniejącego filmu w bazie danych szczegółów o nim. sprawdza czy wprowadzane dane już nie zostały dodane do bazy
- procedura "dodaj\_film\_aktorzy" przyjmuje w argumencie dane aktora i postaci którą grał, tworzy osobę o podanych danych i dodaje ją do tabeli filmy\_aktorzy
- procedura "dodaj\_film\_rezyserzy" dodaje reżysera który wyreżyserował dany film

## wyzwalacze:

- wyzwalacz "Tr\_INSERT\_filmy" na tabeli filmy służy zapobiegnięciu dodania do bazy danych filmu który już istnieje
- wyzwalacz "Tr\_INSERT\_f\_szczegoly" na tabeli filmy\_szczegoly zapobiega przed dodatniem do bazy danych danych które już zostały wcześniej wprowadzone
- wyzwalacz "Tr\_INSERT\_seans" na tabeli seanse\_filmowe zapobiega powstawaniu duplikatów seansów, przypadku gdzie długość seansu jest mniejsza niż długość filmu, przypadku gdy sala już jest zarezerwowana, przypadku gdzie data jest podana "na odwrot"
- wyzwalacz "Tr\_INSERT\_nr\_tel" na tabeli wlasciciel\_nr\_tel sprawdza czy podany nr tel jest prawidłowy tj czy nie zawiera liter i zawiera dokładnie 9 cyfr
- wyzwalacz "Tr\_INSERT\_email" na tabeli wlasciciel\_adresu\_email sprawdza czy email jest poprawny tj posiadać \_@\_.\_\_
- wyzwalacz "Tr\_INSERT\_osoba" na tabeli osoba zgłasza błąd jeśli imię lub nazwisko zawierają litery
- wyzwalacz "Tr\_INSERT\_plec" na tabeli plec zgłasza błąd jeśli podana plec zawiera litery

test tworzenia seansu:

```
> SELECT COUNT(*) FROM seanse_filmowe      :      10
```

```
> EXEC dodaj_seans 'TITANI', '1997-12-19', '2000-01-01 13:00:00', '2000-01-01 09:00:00',
```

```
'1','20' :      BŁAD - FILM O PODANYM TYTULE I PREMIERZE NIE ISTNIEJE W BAZIE  
DANYCH
```

```
> EXEC dodaj_seans 'TITANIC', '1997-12-19', '2000-01-01 13:00:00', '2000-01-01 09:00:00',
```

```
'1','20' :      BŁAD - DATA ZAKONCZENIA JEST PRZED DATA ROZPOCZECIA
```

```
> EXEC dodaj_seans 'TITANIC', '1997-12-19', '2000-01-01 18:00:00', '2000-01-01 20:00:00',
```

```
'1','20' :      BŁAD - CZAS TRWANIA SEANSU JEST MNIEJSZY NIZ CZAS TRWANIA FILMU
```

```
> EXEC dodaj_seans 'TITANIC', '1997-12-19', '2023-02-14 19:10:00', '2023-02-14 23:05:00',
```

```
'1','28' :      JUZ ISTNIEJE
```

```
> SELECT COUNT(*) FROM seanse_filmowe      :      10
```

```
> EXEC dodaj_seans 'TITANIC', '1997-12-19', '2000-01-01 09:00:00', '2000-01-01 13:00:00',
```

```
'1','20' :      SEANS ZOSTAL DODANY PRAWIDLOWO
```

```
> EXEC dodaj_seans 'TITANIC', '1997-12-19', '2000-01-01 12:00:00', '2000-01-01 16:00:00',
```

```
'1','20' :      BŁAD - NA PODANYM PRZEDZIALE CZASU SEANSU SALA ZOSTALA JUZ  
ZAREZERWOWANA
```

```
> SELECT COUNT(*) FROM seanse_filmowe      :      11
```

transakcje działają pomyślnie

test dodawania aktorów:

> SELECT COUNT(\*) FROM osoba : 48

> EXEC dodaj\_film\_aktorzy 'TITANIC', '1997-12-19', 'LEONARDO', 'DICAPRIO', 'mezczyzna',  
'JACK', 'DAWSON', 'mezczyzna' : Error: JUZ ISTNIEJE

> EXEC dodaj\_film\_aktorzy 'ITANIC', '1997-12-19', 'LEONARDO', 'DICAPRIO', 'mezczyzna',  
'JACK', 'DAWSON', 'mezczyzna' : Error: PODANY FILM NIE ISTNIEJE W BAZIE  
DANYCH

> EXEC dodaj\_film\_aktorzy 'TITANIC', '1997-12-12', 'LEONARDO', 'DICAPRIO', 'mezczyzna',  
'JACK', 'DAWSON', 'mezczyzna' : Error: PODANY FILM NIE ISTNIEJE W BAZIE  
DANYCH

> EXEC dodaj\_film\_aktorzy 'TITANIC', '1997-12-19', '1LEONARDO', 'DICAPRIO', 'mezczyzna',  
'JACK', 'DAWSON', 'mezczyzna' : Error: NIEPOPRAWNE IMIE

> EXEC dodaj\_film\_aktorzy 'TITANIC', '1997-12-19', 'LEONARDO', 'DIC5APRIO', 'mezczyzna',  
'JACK', 'DAWSON', 'mezczyzna' : Error: NIEPOPRAWNE NAZWISKO

> EXEC dodaj\_film\_aktorzy 'TITANIC', '1997-12-19', 'LEONARDO', 'DICAPRIO', 'mezczyzna9',  
'JACK', 'DAWSON', 'mezczyzna' : Error: NIEPOPRAWNA PLEC

> EXEC dodaj\_film\_aktorzy 'TITANIC', '1997-12-19', 'LEONARDO', 'DICAPRIO', 'mezczyzna',  
'JACK', 'DAW33333333SON', 'mezczyzna' : Error: NIEPOPRAWNE NAZWISKO

> SELECT COUNT(\*) FROM osoba : 48

> EXEC dodaj\_film\_aktorzy 'FORREST GUMP', '1994-11-4', 'LEONARDO', 'DICAPRIO',  
'mezczyzna', 'imie', 'nazwisko', 'mezczyzna' : AKTOR FILMU ZOSTAL DODANY  
PRAWIDLOWO

> SELECT COUNT(\*) FROM osoba : 49

>EXEC dodaj\_film\_aktorzy 'FORREST GUMP', '1994-11-4', 'imieeee', 'nazwiskoo', 'mezczyzna',  
'imieeeee', 'nazwiskkkko', 'mezczyzna' : AKTOR FILMU ZOSTAL DODANY PRAWID-  
LOWO

> SELECT COUNT(\*) FROM osoba : 51

transakcje działają pomyślnie



test dodawania zamówienia:

> SELECT COUNT(\*) FROM osoba : 48

> SELECT COUNT(\*) FROM zamowienia : 18

> EXEC dodaj\_zamowienie '0', '1000', 'michal', 'bed', 'michal@qq.pl', '222111333', 'mezczyzna' : ERROR SEANS O PODANYM ID NIE ISTNIEJE

> EXEC dodaj\_zamowienie '1000', '0', 'michal', 'bed', 'michal@qq.pl', '222111333', 'mezczyzna' : ERROR NIE MA TYLU WOLNYCH MIEJSC

> EXEC dodaj\_zamowienie '1', '0', 'imle', 'bed', 'michal@qq.pl', '222111333',  
'mezczyzna' : ERROR NIEPOPRAWNE IMIE

> EXEC dodaj\_zamowienie '1', '0', 'imie', 'lbed', 'michal@qq.pl', '222111333',  
'mezczyzna' : ERROR NIEPOPRAWNE NAZWISKO

> EXEC dodaj\_zamowienie '1', '0', 'imie', 'bed', 'abcd@q.', '222111333', 'mezczyzna' :  
ERROR NIEPOPRAWNY ADRES EMAIL

> EXEC dodaj\_zamowienie '1', '0', 'imie', 'bed', 'abcd@q.pl', '22a111333', 'mezczyzna' :  
ERROR NIEPOPRAWNY NR TEL

> EXEC dodaj\_zamowienie '1', '0', 'imie', 'bed', 'abcd@q.pl', '222111333',  
'lmezczyzna1' : ERROR NIEPOPRAWNA PLEC

> EXEC dodaj\_zamowienie '0', '0', 'imie', 'bed', 'abcd@q.pl', '222111333', 'mezczyzna' :  
ERROR LICZBA BILETOW MUSI BYC DODATNIA

> EXEC dodaj\_zamowienie '1', '5', 'imie', 'bed', 'abcd@q.pl', '222111333', 'mezczyzna' :  
OK

> SELECT COUNT(\*) FROM osoba : 49

> SELECT COUNT(\*) FROM zamowienia : 19

transakcje działają pomyślnie