

ID Zadania: **LAB_2**

Tytuł zadania: **Implementacja i elementarna obsługa bazodanowego systemu informatycznego z wykorzystaniem SQL**

Termin oddania: **24.05.2020**

Temat: **Serwis filmowy**

Patryk KANIEWSKI i Wojciech ZŁOMEK



Podstawy baz danych

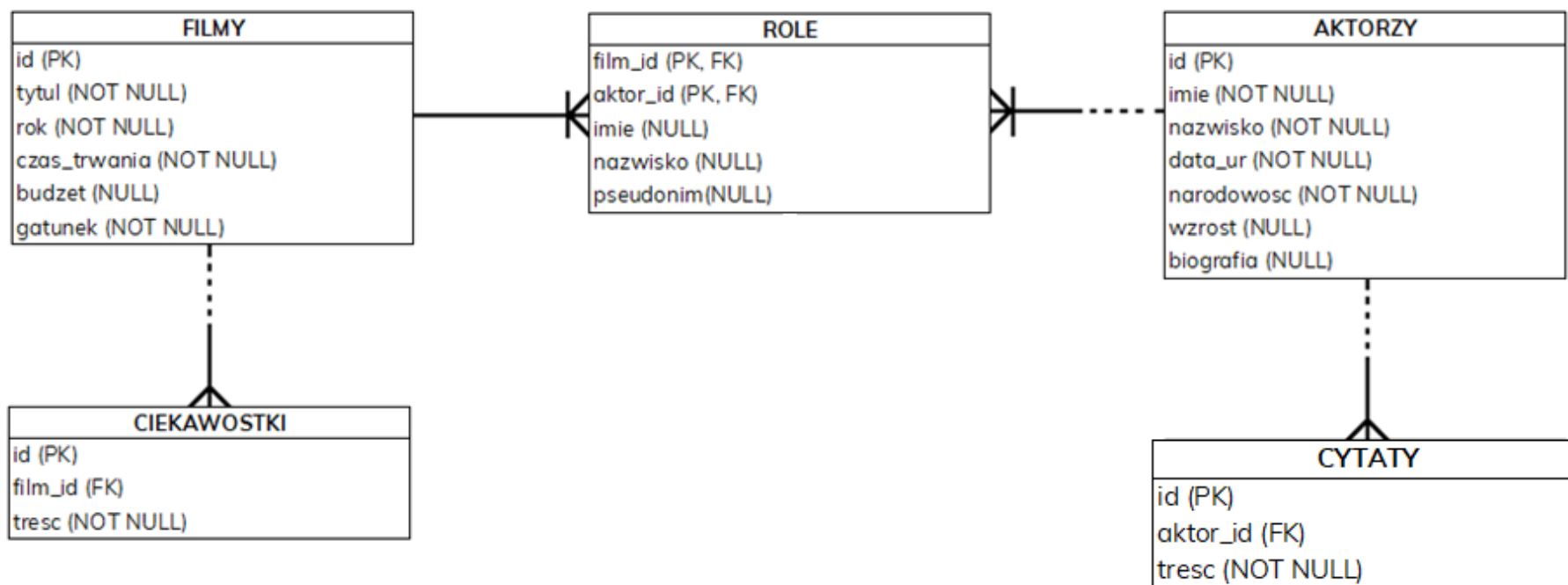
Treść zadania

Zadanie polega na zaimplementowaniu i wykonaniu elementarnych operacji na bazie danych z wykorzystaniem SQL:

- *na ocenę dostateczną (3 - dst):*
 - zaimplementuj model bazy danych przedstawiony poniżej;
 - do każdej tabeli dodaj co najmniej 3 rekordy;
- *na ocenę dobrą (4 - db)*:*
 - wyświetl imiona i nazwiska aktorów, którzy zagraли co najmniej w 1 filmie;
 - wyświetl filmy, w których wystąpił co najmniej 1 polski aktor;
 - wyświetl filmy, do których przypisano ciekawostki;

- wyświetl imiona i nazwiska aktorów, którzy zagraли w filmach wydanych przed rokiem 2000;
- wyświetl aktorów, którzy wypowiedzieli słynne cytaty (wszystkie zapisane w bazie danych), wyniki ogranicz do aktorów o narodowości polskiej;
- na ocenę bardzo dobrą (5 - bdb)*:
 - zaimplementuj perspektywę (VIEW) *Obsada*, która będzie przechowywała imiona i nazwiska aktorów przypisane do imion i nazwisk postaci, w które aktorzy się wcieli.

***wymaga zrealizowania poprzednich podpunktów**



Implementacja relacyjnej bazy danych (ocena: 3 - dst)

W tym miejscu należy krótko napisać jaki RDBMS wykorzystano, a na listingu zamieścić kod tworzący bazę danych zgodną z zadany modelem oraz zapytania uzupełniające każdą z tabel.

```
/*
ten kod verbatim można znaleźć w „createtables.sql” i „filltables.sql”
na systemach z bash można wykonać skrypt „demo.sh”
thisconnect@notarch:~$ mysql --version
mysql Ver 8.0.20-0ubuntu0.20.04.1 for Linux on x86_64 ((Ubuntu))
*/

drop database if exists lab2;
create database lab2 CHARACTER SET utf8 COLLATE utf8_general_ci;
use lab2;

drop table if exists filmy;
create table filmy (
    id int not null auto_increment,
    tytul varchar(50) not null,
    rok int(4) not null,
    czas_trwania int not null,
    budzet int null,
    gatunek varchar(50) not null,
    primary key(id)
);
drop table if exists aktorzy;
create table aktorzy
(
    id int not null auto_increment,
    imie varchar(50) not null,
    nazwisko varchar(50) not null,
    data_ur DATE not null,
    narodowosc varchar(50) not null,
    wzrost int null,
    biografia varchar(255) null,
```

```
        primary key(id)
    );
drop table if exists role;
create table role
(
    film_id int not null,
    aktor_id int not null,
    imie varchar(50) null,
    nazwisko varchar(50) null,
    pseudonim varchar(50) null,
    constraint id primary key(film_id,aktor_id),
    foreign key (film_id) references filmy(id),
    foreign key (aktor_id) references aktorzy(id)
);
drop table if exists cytaty;
create table cytaty (
    id int not null auto_increment,
    aktor_id int not null,
    tresc varchar(255) not null,
    primary key(id),
    foreign key (aktor_id) references aktorzy(id)
);

drop table if exists ciekawostki;
create table ciekawostki (
    id int not null auto_increment,
    film_id int not null,
    tresc varchar(255) not null,
    primary key(id),
    foreign key (film_id) references filmy(id)
);

use lab2;

insert into filmy (tytul,rok,czas_trwania,budzet,gatunek) values ('auta',2006,117,666420123,'animowany');
insert into filmy (tytul,rok,czas_trwania,budzet,gatunek) values ('terminator',1984,107,777420123,'akcja');
insert into filmy (tytul,rok,czas_trwania,budzet,gatunek) values ('seksmisja',1984,117,20123,'komedia');
insert into filmy (tytul,rok,czas_trwania,budzet,gatunek) values ('raiders of the lost
ark',1981,115,73420123,'przygodowy');
```

```
insert into filmy (tytul,rok,czas_trwania,budzet,gatunek) values ('Star wars: Revenge of the
Sith',2005,140,213103,'fantastyczny');
insert into filmy (tytul,rok,czas_trwania,budzet,gatunek) values ('Star wars: Empire strikes
back',1980,124,5123123,'fantastyczny');
insert into filmy (tytul,rok,czas_trwania,budzet,gatunek) values ('Psy',1992,104,23123,'sensacyjny');

insert into aktorzy (imie,nazwisko,data_ur,narodowosc,wzrost,biografia) values ('arnold','schwarzeneger','1947-
07-30','austriacka',181,'test');
insert into aktorzy (imie,nazwisko,data_ur,narodowosc,wzrost,biografia) values ('Jerzy','Stuhr','1947-04-
18','polska',164,'test');
insert into aktorzy (imie,nazwisko,data_ur,narodowosc,wzrost,biografia) values ('Olgierd','Łukaszewicz','1946-09-
7','polska',172,'test');
insert into aktorzy (imie,nazwisko,data_ur,narodowosc,wzrost,biografia) values ('Ian','McDiarmid','1944-08-
11','brytyjska',160,'test');
insert into aktorzy (imie,nazwisko,data_ur,narodowosc,wzrost,biografia) values ('Ewan','McGregor','1971-03-
31','brytyjska',184,'');
insert into aktorzy (imie,nazwisko,data_ur,narodowosc,wzrost,biografia) values ('Cezary','Pazura','1946-09-
07','polska',172,'test');
insert into aktorzy (imie,nazwisko,data_ur,narodowosc,wzrost,biografia) values ('Alec','Guinness','1914-04-
02','brytyjska',184,'umarł 5 sierpnia 2000');

insert into role (film_id,aktor_id,imie,nazwisko,pseudonim) values (2,1,'imie','nazwisko','terminator');
insert into role (film_id,aktor_id,imie,nazwisko,pseudonim) values (3,2,'Maksymilian','Paradys','Maks');
insert into role (film_id,aktor_id,imie,nazwisko,pseudonim) values (3,3,'Albert','Saski','');
insert into role (film_id,aktor_id,imie,nazwisko,pseudonim) values (4,1,'Indiana','Jones','');
insert into role (film_id,aktor_id,imie,nazwisko,pseudonim) values (5,4,'Sheev','Palpatine','Darth Sidious');
insert into role (film_id,aktor_id,imie,nazwisko,pseudonim) values (6,4,'Sheev','Palpatine','Darth Sidious');
insert into role (film_id,aktor_id,imie,nazwisko,pseudonim) values (5,5,'Obi-wan','Kenobi','Master Kenobi');
insert into role (film_id,aktor_id,imie,nazwisko,pseudonim) values (7,6,'Waldemar','Morawiec','Nowy');

insert into cytaty (aktor_id,tresc) values (1,'I ll be back');
insert into cytaty (aktor_id,tresc) values (2,'Kobieta mnie bije');
insert into cytaty (aktor_id,tresc) values (2,'Dzień dobry, zastałem Jolkę?');
insert into cytaty (aktor_id,tresc) values (5,'Its over Anakin, I have the high ground.');
```

```
insert into ciekawostki (film_id,tresc) values (2,'One afternoon during a break in filming, Arnold Schwarzenegger
went into a restaurant in downtown L.A. to get some lunch and realized all too late that he was still in
```

```
Terminator makeup - with a missing eye, exposed jawbone and burned flesh.');
```

```
insert into ciekawostki (film_id,tresc) values (6,'The shots where Luke uses his Jedi powers to retrieve his  
lightsaber from a distance were achieved by having Mark Hamill throw the lightsaber away, and then running the  
film in reverse.');
```

```
insert into ciekawostki (film_id,tresc) values (4,'Freeze-framing during the Well of Souls scene, you can notice  
a golden pillar with a tiny engraving of R2-D2 and C-3PO from the Star Wars saga. They are also on the wall behind  
Indy when they first approach the Ark.');
```

Listing 1 podpis, (źródło: opracowanie własne; createtables.sql, filltables.sql)

Przetwarzanie danych (ocena: 4 - db)

W tym miejscu należy opisać proces wybierania informacji wymaganych w zadaniu, w oparciu o zaimplementowaną bazę danych.

```
/*
ten kod verbatim można znaleźć w „dobra.sql”
*/
use lab2;

select 'wyświetl imiona i nazwiska aktorów, którzy zagraли co najmniej w 1 filmie:' as '';

SELECT DISTINCT aktorzy.imie, aktorzy.nazwisko
FROM (aktorzy
INNER JOIN role ON aktorzy.id = role.aktor_id);

select 'wyświetl filmy, w których wystąpił co najmniej 1 polski aktor:' as '';

SELECT DISTINCT filmy.tytul
FROM (( role
INNER JOIN filmy ON role.film_id = filmy.id)
INNER JOIN aktorzy ON role.aktor_id = aktorzy.id)
WHERE aktorzy.narodowosc="polska";

select 'wyświetl filmy, do których przypisano ciekawostki:' as '';

SELECT DISTINCT filmy.tytul, ciekawostki.tresc
FROM (ciekawostki
INNER JOIN filmy ON ciekawostki.film_id = filmy.id);

select 'wyświetl imiona i nazwiska aktorów, którzy zagraли w filmach wydanych przed rokiem 2000' as '';

SELECT DISTINCT aktorzy.imie, aktorzy.nazwisko
```

```
FROM ((role
INNER JOIN aktorzy ON role.aktor_id = aktorzy.id)
INNER JOIN filmy ON role.film_id = filmy.id)
WHERE filmy.rok <2000;

select 'wyświetl aktorów, którzy wypowiedzieli słynne cytaty (wszystkie zapisane w bazie danych), wyniki ogranicz
do aktorów o narodowości polskiej' as '';

SELECT DISTINCT aktorzy.imie, aktorzy.nazwisko, cytaty.tresc
FROM ((aktorzy
INNER JOIN role ON role.aktor_id = aktorzy.id)
INNER JOIN cytaty ON cytaty.aktor_id = aktorzy.id)
WHERE aktorzy.narodowosc = "polska";
```

Listing 2 podpis, (źródło: opracowanie własne; dobra.sql)

Implementacja perspektywy (ocena: 5 - bdb)

W tym miejscu należy krótko opisać proces tworzenia perspektywy spełniającej wymagania postawione w treści zadania.

```
/*
ten kod verbatim można znaleźć w „bardzodobra.sql”
*/
use lab2
select 'zaimplementuj perspektywę (VIEW) Obsada, która będzie przechowywała imiona i nazwiska aktorów przypisane
do imion i nazwisk postaci, w które aktorzy się wcieli.' as '';

CREATE VIEW obsada as
SELECT distinct aktorzy.imie, aktorzy.nazwisko, role.imie as 'imie postaci', role.nazwisko as 'nazwisko postaci',
role.pseudonim
FROM role inner join aktorzy on aktorzy.id = role.aktor_id;

select 'select * from obsada' as '';

select * from obsada;
```

Listing 3 podpis, (źródło: opracowanie własne; bardzodobra.sql)