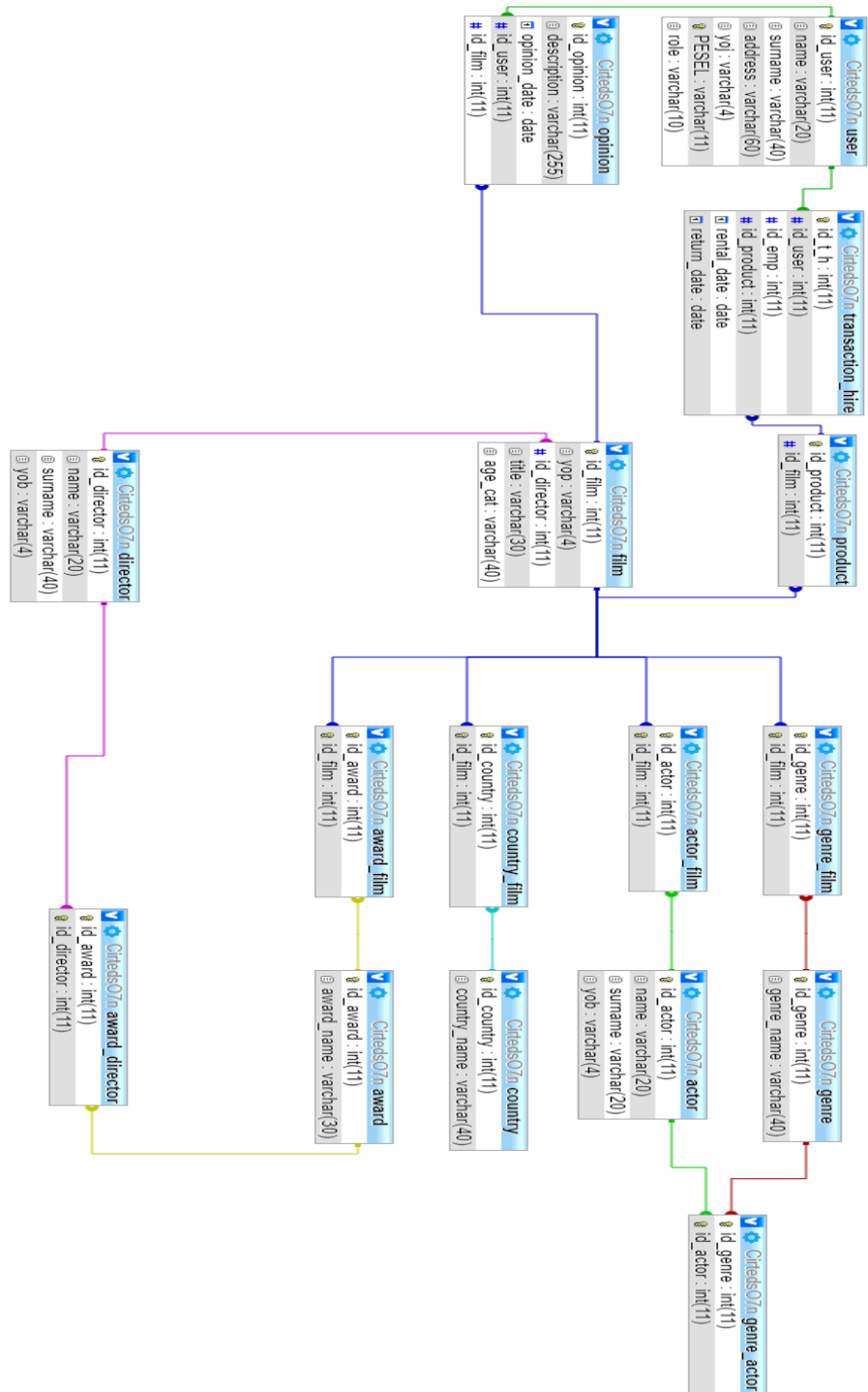


Temat nr 1 - Wypożyczalnia filmów			
Lp.	Imię i nazwisko	Numer albumu	Grupa
1	Aleksandra Piechnik	299733	II
2	Dawid Kruczek		
Rok akademicki		Semestr	Kierunek
2018/2019		IV	Informatyka Stosowana

1. Schemat bazy danych



Lp.	Nazwa tabeli	Znaczenie
1	User	Wszyscy użytkownicy systemu
2	Film	Główna tabela, filmy istniejące w bazie
3	Product	Tabela egzemplarzy filmu
4	Transaction_hire	Tabela transakcji wypożyczenia i zwrotu
5	Director	Tabela reżyserów
6	Actor	Tabela aktorów
7	Country	Tabela krajów
8	Genre	Tabela aktorów
9	Award	Tabela nagród
10	Tabela1_tabela2	Wszystkie tabele, które służą jako pomocnicze w relacji wiele do wiele
11	Opinion	Tabela opinii

2. Podstawowy CRUD

Lp.	Zapytanie
1	"SELECT * " + "FROM user u " + "ORDER BY u.id_user ASC";
	Znaczenie
	Pobranie rekordów z danymi dotyczącymi użytkownika.

Lp.	Zapytanie
2	"SELECT * " + "FROM user u " + "WHERE u.id_user= id";
	Znaczenie
	Pobranie rekordu użytkownika z podanym id.

Lp.	Zapytanie
3	"SELECT f.id_film,f.title,f.yop,f.age_cat,d.name,d.surname,g.genre_name, c.country_name " + "FROM film f " + "INNER JOIN director d ON d.id_director=f.id_director " + "INNER JOIN genre_film gf ON gf.id_film=f.id_film " + "INNER JOIN genre g ON g.id_genre=gf.id_genre " + "INNER JOIN country_film cf ON cf.id_film=f.id_film " + "INNER JOIN country c ON c.id_country=cf.id_country " + "WHERE f.id_film=";
	Znaczenie
	Pobranie rekordu filmu z podanym id.

Lp.	Zapytanie
.	
4	"SELECT f.id_film,f.title,d.name,d.surname,f.age_cat,f.yop,g.genre_name,c.coun try_name " + "FROM film f " + "INNER JOIN director d ON d.id_director=f.id_director " + "INNER JOIN genre_film gf ON gf.id_film=f.id_film " + "INNER JOIN genre g ON g.id_genre=gf.id_genre " + "INNER JOIN country_film cf ON cf.id_film=f.id_film " + "INNER JOIN country c ON c.id_country=cf.id_country;";
	Znaczenie
	Pobranie rekordów z danymi dotyczącymi filmu.

Lp.	Zapytanie
5	"SELECT th.id_t_h, u.PESEL,u.name,u.surname,f.title,th.id_emp,th.rental_date, th.return_date " + "FROM transaction_hire th " + "INNER JOIN user u ON u.id_user=th.id_user " + "INNER JOIN product p ON th.id_product=p.id_product " + "INNER JOIN film f ON f.id_film=p.id_film";
	Znaczenie
	Pobranie rekordów z danymi dotyczącymi zamówienia.

Lp.	Zapytanie
6	"SELECT th.id_t_h, u.PESEL,u.name,u.surname,f.title,th.id_emp,th.rental_date, th.return_date " + "FROM transaction_hire th " + "INNER JOIN user u ON u.id_user=th.id_user " + "INNER JOIN product p ON th.id_product=p.id_product " + "INNER JOIN film f ON f.id_film=p.id_film " + "WHERE th.id_t_h=";
	Znaczenie
	Pobranie rekordu zamówienia z podanym id.

Lp.	Zapytanie
7	"INSERT INTO user(name,surname,address,yoj,PESEL,role) " + "VALUES('" + name + "','" + surname + "','" + address + "','" + yoj + "','" + pesel + "','" + role + "');" ;
	Znaczenie
	Dodanie użytkownika do bazy.

Lp.	Zapytanie
8	"SELECT id_director FROM director WHERE name='" + NameDirector + "'AND surname='" + SurnameDirector + "';";
	Znaczenie
	Sprawdzenie czy istnieje reżyser o podanym imieniu i nazwisku.

Lp.	Zapytanie
9	"Insert Into director(name,surname) VALUES('" + NameDirector + "','" + SurnameDirector + "');" ;
	Znaczenie
	Dodanie reżysera o podanym imieniu i nazwisku.

Lp.	Zapytanie
10	"SELECT id_genre FROM genre WHERE genre_name='" + genre + "';";
	Znaczenie
	Sprawdzenie czy istnieje w bazie podany gatunek filmu.

Lp.	Zapytanie
11	"Insert Into genre(genre_name) VALUES('" + genre + "');";
	Znaczenie
	Dodanie gatunku filmu.

Lp.	Zapytanie
12	"SELECT id_country FROM country WHERE country_name='" + country + "';";
	Znaczenie
	Sprawdzenie czy istnieje w bazie podany kraj.

Lp.	Zapytanie
13	"Insert Into country(country_name) VALUES('" + country + "');";
	Znaczenie
	Dodanie kraju.

Lp.	Zapytanie
14	"Insert into film(yop,id_director,title,age_cat) VALUES('" + yop + "',' + last_director_id + "',' + title + "',' + age_category + "');";
	Znaczenie
	Dodanie filmu.

Lp.	Zapytanie
15	"Insert Into genre_film(id_genre,id_film) Values(" + last_genre_id + "," + last_film_id + ");";
	Znaczenie
	Dodanie danych do tabeli genre_film.

Lp.	Zapytanie
16	"Insert Into country_film(id_country,id_film) Values(" + last_country_id + "," + last_film_id + ");";
	Znaczenie
	Dodanie danych do tabeli country_film.

Lp.	Zapytanie
17	"Insert Into product(id_film) Values(" + last_film_id + ");";
	Znaczenie
	Dodanie produktu.

Lp.	Zapytanie
18	"UPDATE user SET name='" + userRecord.getName() + "',surname='" + userRecord.getSurname() + "',PESEL='" + userRecord.getPesel() + "',address='" + userRecord.getAddress() + "',yoy='" + userRecord.getYoy() + "',role='" + userRecord.getRole() + "'" WHERE id_user=" + userRecord.getUser_id() + ";";
	Znaczenie
	Aktualizacja danych użytkownika.

Lp.	Zapytanie
19	"DELETE FROM user WHERE id_user=" + id + ";";
	Znaczenie
	Usunięcie użytkownika z bazy.

Lp.	Zapytanie
20	UPDATE WYŚWIETLANYCH DANYCH FILMU (NIE WSZYSTKIE POCHODZĄ Z TABELII FILM)
	<p>Sprawdzamy czy nowy reżyser istnieje w bazie. String sql_check_director = "SELECT id_director FROM director WHERE name='" + filmRecord.getDirector_name() + "'AND surname='" + filmRecord.getDirector_surname() + "';";</p> <p>Jeśli reżyser nie istnieje w bazie dodajemy go. String sql_insert_director = "Insert Into director(name,surname) VALUES('" + filmRecord.getDirector_name() + "'," + filmRecord.getDirector_surname() + "');";</p> <p>Aktualizujemy dane w tabelii film. "UPDATE film SET title='" + filmRecord.getTitle() + "', yoy='" + filmRecord.getYoy() + "', age_cat='" + filmRecord.getAge_category() + "'," + " id_director=" + last_director_id + " WHERE id film=" +</p>

	<pre>filmRecord.getId_film() + ";;";</pre> <p>Sprawdzamy czy podany gatunek filmu istnieje już w bazie. "SELECT id_genre FROM genre WHERE genre_name=" + filmRecord.getGenre() + "';";</p> <p>Jeśli nie istnieje dodajemy go do bazy. "Insert Into genre(genre_name) VALUES(' + filmRecord.getGenre() + "');";</p> <p>Aktualizujemy tabelę genre_film "UPDATE genre_film SET id_genre=" + last_genre_id + " WHERE id_film=" + filmRecord.getId_film() + ";;";</p> <p>Sprawdzamy czy podany kraj istnieje w bazie. "SELECT id_country FROM country WHERE country_name=" + filmRecord.getCountry() + "';";</p> <p>Jeśli nie istnieje dodajemy go. "Insert Into country(country_name) VALUES(' + filmRecord.getCountry() + "');";</p> <p>Aktualizujemy tabelę country_film "UPDATE country_film SET id_country=" + last_country_id + " WHERE id_film=" + filmRecord.getId_film() + ";;";</p>
	Znaczenie
	Zapytanie jest rozbudowane w celu zachowania spójności danych. Komentarze bezpośrednio nad każdym pod zapytaniem.

Lp.	Zapytanie
21	<pre>"DELETE FROM film WHERE id_film=" + id + ";;";</pre>
	Znaczenie
	Usunięcie filmu z bazy.

Lp.	Zapytanie
22	<p>Sprawdzenie czy istnieje użytkownik o podanym id. "SELECT id_user FROM user WHERE PESEL=" + user_pesel + "';";</p> <p>Sprawdzenie ile filmów obecnie jest wypożyczonych przez użytkownika. "SELECT COUNT(id_user) as amount FROM transaction_hire WHERE return_date is null and id_user=" + get_id_user + ";;";</p> <p>Sprawdzenie czy istnieje film o podanym id. "SELECT id_film FROM film WHERE title=" + title + "';";</p> <p>Sprawdzenie ile sztuk filmu jest w bazie "SELECT COUNT(p.id_product) AS counter_product FROM product p WHERE p.id_film=" + get_id_film + ";;";</p> <p>Sprawdzenie czy istnieją wolne egzemplarze do wypożyczenia</p>

	<pre>"SELECT COUNT(th.id_t_h) AS counter_trans FROM transaction_hire th WHERE th.return_date is null AND th.id_film=" + get_id_film + ";;"; "SELECT th.id_product FROM transaction_hire th WHERE th.return_date is not null AND th.id_film=" + get_id_film + " LIMIT 1"; "SELECT p.id_product from product p WHERE not EXISTS (Select th.id_product FROM transaction_hire th WHERE th.id_product=p.id_product) AND p.id_film=" + get_id_film + " LIMIT 1;"; "INSERT INTO transaction_hire(id_user,id_emp,id_product,id_film,rental_date) " + VALUES(" + get_id_user + "," + id_emp + "," + get_id_product + "," + get_id_film + "," + rental_date + "');";</pre>
	Znaczenie
	Dodanie filmu do bazy

Lp.	Zapytanie
23	<pre>"select id_user, description, opinion_date from opinion where id_film=" + id + ";;";</pre>
	Znaczenie
	Pobranie opinii o wybranym filmie.

Lp.	Zapytanie
24	<pre>"select a.id_actor, a.name, a.surname from actor a " + "Inner join actor_film af on af.id_actor=a.id_actor " + "Where af.id_film=" + id + ";;";</pre>
	Znaczenie
	Pobranie listy aktorów wybranego filmu.

Lp.	Zapytanie
.	
24	<pre>"SELECT f.id_film,f.title,d.name,d.surname,f.age_cat,f.yop,g.genre_name,c.count ry_name " + "FROM film f " + "INNER JOIN director d ON d.id_director=f.id_director " + "INNER JOIN genre_film gf ON gf.id_film=f.id_film " + "INNER JOIN genre g ON g.id_genre=gf.id_genre " + "INNER JOIN country_film cf ON cf.id_film=f.id_film " + "INNER JOIN country c ON c.id_country=cf.id_country " + "WHERE g.genre_name='" + genre + "';";</pre>
	Znaczenie
	Pobranie filmów o podanym gatunku filmu. Wykorzystywane podczas filtrowania wyników.

Lp	Zapytanie
.	
26	<pre>"SELECT f.id_film,f.title,d.name,d.surname,f.age_cat,f.yop,g.genre_name,c.count ry_name " + "FROM film f " + "INNER JOIN director d ON d.id_director=f.id_director " + "INNER JOIN genre_film gf ON gf.id_film=f.id_film " + "INNER JOIN genre g ON g.id_genre=gf.id_genre " + "INNER JOIN country_film cf ON cf.id_film=f.id_film " + "INNER JOIN country c ON c.id_country=cf.id_country " + "WHERE c.country_name='" + country + "';";</pre>
	Znaczenie
	Pobranie filmów wyprodukowanych w podanym kraju. Wykorzystywane podczas filtrowania wyników.

Lp	Zapytanie
.	
27	<pre>"SELECT f.id_film,f.title,d.name,d.surname,f.age_cat,f.yop,g.genre_name,c.count ry_name " + "FROM film f " + "INNER JOIN director d ON d.id_director=f.id_director " + "INNER JOIN genre_film gf ON gf.id_film=f.id_film " + "INNER JOIN genre g ON g.id_genre=gf.id_genre " + "INNER JOIN country_film cf ON cf.id_film=f.id_film " + "INNER JOIN country c ON c.id_country=cf.id_country " + "WHERE g.genre_name='" + genre + "' AND c.country_name='" + country + "';";</pre>
	Znaczenie
	Pobranie filmów wyprodukowanych w podanym kraju i będący podanym gatunkiem. Wykorzystywane podczas filtrowania wyników. 2 filtry naraz.

Lp.	Zapytanie
28	<pre>"SELECT d.surname, COUNT(f.id_film) AS Amount " + "FROM director d " + "INNER JOIN film f ON d.id_director=f.id_director " + "GROUP BY d.surname " + "ORDER BY Amount DESC";</pre>
	Znaczenie
	Wykorzystywane przy wyświetlaniu statystyk: liczba filmów na każdego reżysera.

Lp.	Zapytanie
28	"SELECT COUNT(cf.id_film) as amount, c.country_name " + "FROM country c " + "INNER JOIN country_film cf ON cf.id_country=c.id_country " + "GROUP BY c.country_name";
	Znaczenie
	Wykorzystywane przy wyświetlaniu statystyk: liczba filmów na każdy kraj.

3. Funkcjonalności oparte o złączenia

Lp.	Zapytanie
3	<pre>"SELECT f.id_film,f.title,f.yop,f.age_cat,d.name,d.surname,g.genre_name, c.country_name " + "FROM film f " + "INNER JOIN director d ON d.id_director=f.id_director " + "INNER JOIN genre_film gf ON gf.id_film=f.id_film " + "INNER JOIN genre g ON g.id_genre=gf.id_genre " + "INNER JOIN country_film cf ON cf.id_film=f.id_film " + "INNER JOIN country c ON c.id_country=cf.id_country " + "WHERE f.id_film=";</pre>
	Znaczenie
	Pobranie rekordu filmu z podanym id.

Lp.	Zapytanie
4	<pre>"SELECT f.id_film,f.title,d.name,d.surname,f.age_cat,f.yop,g.genre_name,c.country_name " + "FROM film f " + "INNER JOIN director d ON d.id_director=f.id_director " + "INNER JOIN genre_film gf ON gf.id_film=f.id_film " + "INNER JOIN genre g ON g.id_genre=gf.id_genre " + "INNER JOIN country_film cf ON cf.id_film=f.id_film " + "INNER JOIN country c ON c.id_country=cf.id_country;"</pre>
	Znaczenie
	Pobranie rekordów z danymi dotyczącymi filmu.

Lp.	Zapytanie
5	<pre>"SELECT th.id_t_h, u.PESEL,u.name,u.surname,f.title,th.id_emp,th.rental_date, th.return_date " + "FROM transaction_hire th " + "INNER JOIN user u ON u.id_user=th.id_user " + "INNER JOIN product p ON th.id_product=p.id_product " + "INNER JOIN film f ON f.id_film=p.id_film";</pre>
	Znaczenie
	Pobranie rekordów z danymi dotyczącymi zamówienia.

Lp.	Zapytanie
6	<pre>"SELECT th.id_t_h, u.PESEL,u.name,u.surname,f.title,th.id_emp,th.rental_date, th.return_date " + "FROM transaction hire th " +</pre>

	"INNER JOIN user u ON u.id_user=th.id_user " + "INNER JOIN product p ON th.id_product=p.id_product " + "INNER JOIN film f ON f.id_film=p.id_film " + "WHERE th.id_t_h=";
	Znaczenie
	Pobranie rekordu zamówienia z podanym id.

4. Funkcjonalności oparte o operacje na zbiorach

Lp.	Zapytanie
1	<pre>SELECT d.name,d.surname,d.yob FROM director d UNION SELECT a.name, a.surname, a.yob FROM actor a</pre>
	Znaczenie
	Wyświetla wszystkie osoby kina

Lp.	Zapytanie
2	<pre>(SELECT d.name,d.surname,f.title as title FROM director d INNER JOIN film f ON f.id_director=d.id_director WHERE f.yop='2019' UNION SELECT a.name, a.surname,f.title as title FROM actor a INNER JOIN actor_film fa ON fa.id_actor=a.id_actor INNER JOIN film f ON f.id_film=fa.id_film WHERE f.yop='2019') ORDER BY title;</pre>
	Znaczenie
	Wyświetla osoby związane z filmem wyprodukowanym w 2019

Lp.	Zapytanie
3	<pre>SELECT f.title FROM film f LEFT JOIN genre_film gf ON gf.id_film=f.id_film INNER JOIN genre g ON g.id_genre=gf.id_genre WHERE g.genre_name<>"Sci-Fi"</pre>
	Znaczenie
	Wszystkie filmy oprócz gatunku Sci-Fi

Lp.	Zapytanie
4	<pre>SELECT * FROM actor a1 WHERE a1.id_actor NOT IN (SELECT DISTINCT a.id_actor FROM actor a LEFT JOIN actor_film fa ON fa.id_actor=a.id_actor WHERE a.id_actor=fa.id_actor) ORDER BY a1.id_actor;</pre>
	Znaczenie
	Aktorzy, którzy są w bazie ale nie zagraли w żadnym filmie

5. Funkcjonalności oparte o grupowanie

Lp.	Zapytanie
1	<pre>SELECT d.surname, COUNT(f.id_film) AS Amount FROM director d INNER JOIN film f ON d.id_director=f.id_director GROUP BY d.surname ORDER BY Amount DESC</pre>
	Znaczenie
	Ilość filmów dla danego reżysera

Lp.	Zapytanie
2	<pre>SELECT a.name,a.surname, COUNT(f.id_film) AS Amount FROM actor a INNER JOIN film_actor fa ON a.id_actor=fa.id_actor INNER JOIN film f ON f.id_film=fa.id_film GROUP BY a.surname ORDER BY Amount DESC</pre>
	Znaczenie
	Ilość filmów dla aktora

Lp.	Zapytanie
3	<pre>SELECT COUNT(cf.id_film) as amount, c.country_name FROM country c INNER JOIN country_film cf ON cf.id_country=c.id_country GROUP BY c.country_name;</pre>
	Znaczenie
	Ilość filmów dla danego kraju