КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет комп’ютерних наук та кібернетики

Спеціальність 122 «Комп’ютерні науки»

Звіт

Лабораторна робота

з дисципліни «Бази даних та інформаційні системи»

тема проекту: «Фільми»

Виконав: студент 2 курсу

Групи К-27

**Шевченко Максим Олексійович**

Київ 2019

# Реферат

Робота складається зі вступу, 5 розділів, висновків, списку використаних джерел та додатків. Обсяг роботи: 21 сторінка, 17 ілюстрацій, 6 таблиць, 9 джерел посилання.

ФІЛЬМИ, ОНЛАЙН-КІНОТЕАТР, БАЗА ДАНИХ, ОПИС БАЗИ ДАНИХ, SQL, ADO.NET, ЗАПИТИ, ІНТЕРФЕЙС, ПРОДУКТ, ПРОГРАМА.

Об’єктом роботи є процес розробки програмного забезпечення для зручного збереження фільмів, продюсерів, жанрів та акторів за допомогою бази даних «База даних Фільмів». Предметом роботи є інтерфейс для виведення результатів пошуку та редагування елементів бази даних.

Метою роботи є реалізація пошукового механізму, інструментарію редагування та зручного виведення інформації користувачу.

Методи розроблення: моделювання, розробка інтерфейсу на основі еволюційної моделі. Інструменти розроблення: Microsoft SQL Server Management Studio [[3]](#_Висновки), середовище програмування Microsoft Visual Studio 2017[[1]](#_Список_використаних_джерел), мова програмування C# [[9].](#_Список_використаних_джерел)

Результат роботи: успішно виконано ґрунтовний огляд існуючих систем, пов’язаних із наданням інформації про фільми. Розроблено реалістичну базу даних, яка повністю імітує базові можливості онлайн-кінотеатру та інтерфейс, який дозволяє зручно нею керувати.

Проект «Фільми» в теорії повинен використовуватися шикорим колом користувачів для пошуку, додавання та керування інформацією про фільми, продюсерів, акторів та жанри.

# Зміст

[Вступ 4](#_Toc6478943)

[Розділ 1. Огляд існуючих на ринку систем 6](#_Toc6478944)

[Розділ 2. Огляд використаних технологій та технічні вимоги 9](#_Toc6478945)

[Розділ 3. Опис організації інформаційної бази 10](#_Toc6478946)

[3.1 Опис таблиць 11](#_Toc6478947)

[3.2.Табличні дані 12](#_Toc6478948)

[Розділ 4. Інструкція користувача 14](#_Toc6478949)

[Розділ 5. Запити 18](#_Toc6478950)

[Висновки 21](#_Toc6478951)

[Список використаних джерел 22](#_Toc6478952)

[ДОДАТКИ 23](#_Toc6478953)

# **Вступ**

**Актуальність**: На даний момент існує безліч онлайн-кінотеатрів, які надають різну інформацію про фільми. Також є сайти, які зберігають якісь специфічні дані про фільми та пов’язані з ними речі (біографію виконавців, гонорари, технічне обладнання тощо). Більшість з них дуже схожі між собою, хоча є відмінності в інтерфейсі та будові системи.

Створений проект має на меті надати програмне забезпечення всім бажаючим збирати інформацію про кінематограф.

Отже, **метою** проекту є процес створення простої програми з інтерфейсом онлайн-кінотеатру, яка працює на основі баз даних там має можливості пошуку, додавання та редагування елементів.

Для досягнення мети потрібно виконати такі **завдання**:

* проаналізувати ринок існуючих сервісів та програм;
* відокремити базовий функціонал продукту;
* створити простий та дружній до користувача інтерфейс;
* сформувати власну базу даних;
* ознайомитися із навчальними матеріалами, які допоможуть реалізувати проект;
* визначитися із середовищем розробки на мовою програмування.

Основною відмінністю даної системи є простота та безкоштовність використання, легка робота із інформацією про фільми, жанри, акторів та продюсерів. Фільм не може з’явитися нізвідки. Кожен товар має продюсера, якого необхідно завчасно додати до бази даних. Це дозволить запобігти випадковому повторюванню однакових фільмів. Продюсер може мати декілька фільмів. Фільм може мати декілька жанрів, у ньому можуть зніматися декілька акторів. Наявна можливість пошуку фільму по назві, року зйомки, актору, жанру. У даній лабораторній роботі було реалізовано прототип цієї системи з можливістю легкого розширення функціоналу та даних.

Є фільми, які характеризуються назвою та роком зйомки. У кожного фільму є продюсер, який характеризуються ім’ям, роком народження та смерті.

Також фільм може мати декілька жанрів, які характеризуються назвою. У фільмі можуть зніматися актори, котрі характеризуються ім’ям, роком народження та смерті.

Специфікація комп’ютера, операційної системи та базової системи керування базами даних.

Система була створена з використанням Microsoft Visual Studio 2017 [[1]](#_Список_використаних_джерел). Операційна система Windows 10 Home [[2]](#_Список_використаних_джерел).

Під час виконання проекту, також було вивчено та вдосконалено вміння користування технологіями ADO.Net [[3]](#_Список_використаних_джерел) та Microsoft SQL Server [[4]](#_Список_використаних_джерел).

# Розділ 1. Огляд існуючих на ринку систем

1. Онлайн-кінотеатр «Moviestape.net»[[5]](#_Список_використаних_джерел)

Це український портал про кіно, де можна ознайомитись з біографіями та фільмографіями відомих акторів і продюсерів, почитати останні кіно-новини, переглянути нові трейлери до очікуваних фільмів. Має велику базу даних, проте не володіє зручним пошуком. (рис.1)

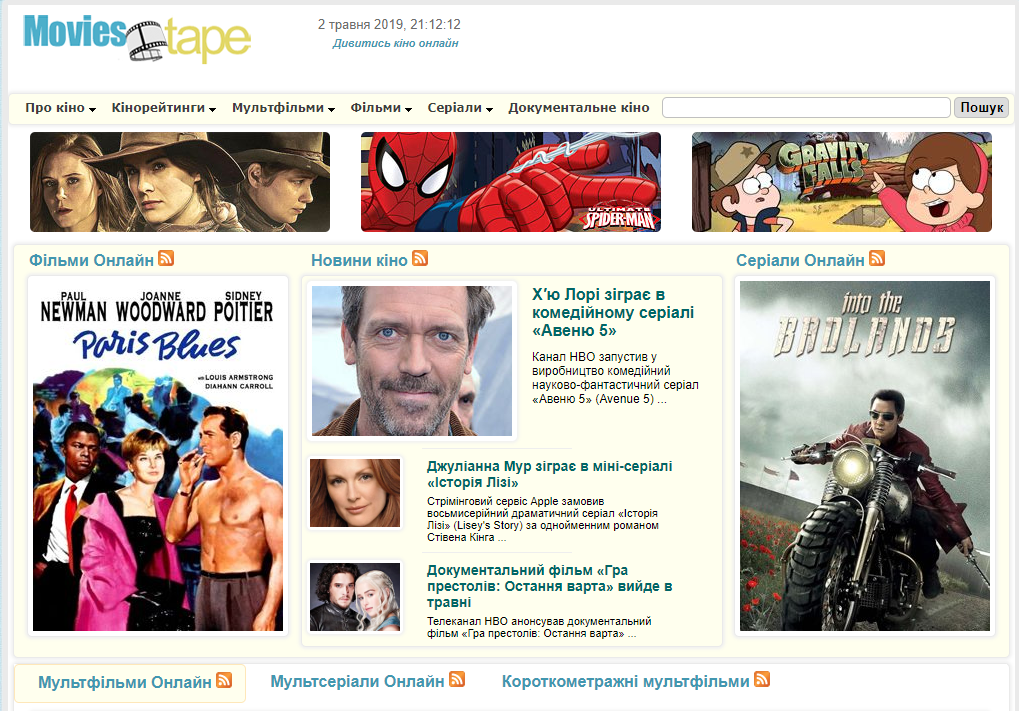


Рис 1. Головна сторінка moviestape.net

1. Онлайн-кінотеатр «uaKINO.club» [[6]](#_Список_використаних_джерел)

Один із найпопулярніших українських онлайн-кінотеатрів. Має велику фільмотеку, яка швидко оновлюється. Проте не збирає інформацію, пов’язану із кінематографом. (рис.2)

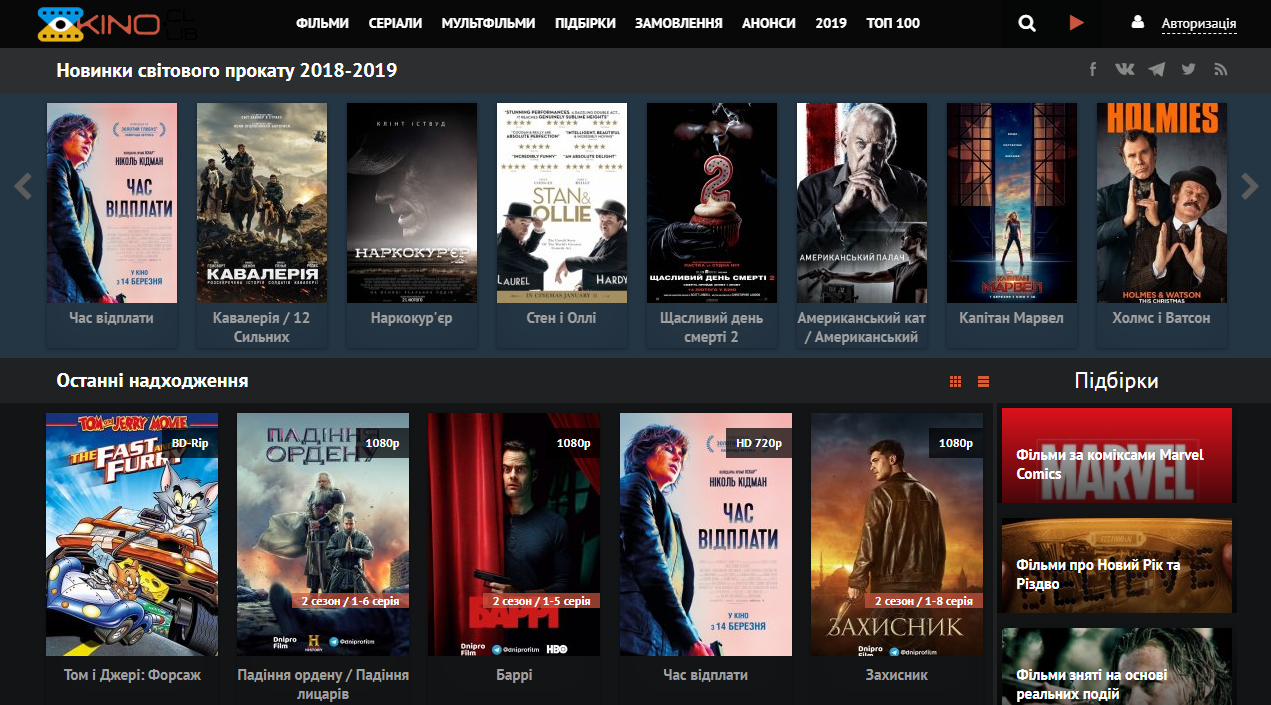


Рис 2. Головна сторінка uakino.club

1. Онлайн-кінотеатр «KINOGO.co.ua» [[7]](#_Список_використаних_джерел)

Відносно новий український онлайн-кінотеатр, який уже має досить велику фільмотеку. Головною перевагою є зручний пошук. До недоліків можна віднести незручний інтерфейс та погану якість перекладу інформації на сайті. (рис.3)

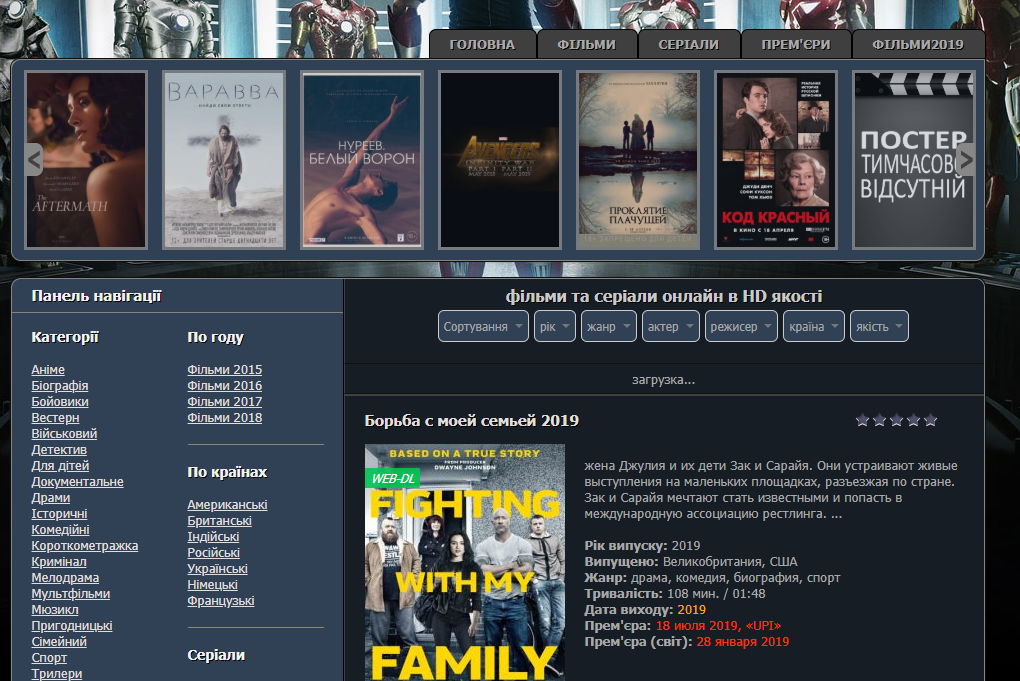


Рис 3. Головна сторінка kinogo.co.ua

Як можна помітити, усі ці сайти мають притаманні онлайн-кінотеатрам властивості, таких як: пошук фільмів, реєстрація користувачів тощо. Вище зазначені властивості були реалізовані в продукті.

# Розділ 2. Огляд використаних технологій та технічні вимоги

Проект було зроблено на OS Windows 10[[2]](#_Список_використаних_джерел). Для побудови бази даних було використано Microsoft SQL Server Management Studio[[4]](#_Список_використаних_джерел). Для створення самого проекту Microsoft Visual Studio 2017[[1]](#_Список_використаних_джерел) та технологія ADO.Net[[3]](#_Список_використаних_джерел).

Знання мови SQL чи мов програмування не вимагається.

Операційна система Windows[[2]](#_Список_використаних_джерел). Програма буде працювати для всіх OS Windows[[2]](#_Список_використаних_джерел) новіших за Windows Vista[[8].](#_Список_використаних_джерел)

Увесь інтерфейс україномовний. Знання іноземних мов для розуміння інтерфейсу та роботи не потрібні.

Джерелом даних для системи повинна бути інформаційна система Microsoft SQL Server[[3]](#_Список_використаних_джерел).

# Розділ 3. Опис організації інформаційної бази

Логічна структура бази даних

Нижче представлена логічна структура бази даних, яка ілюструє зв’язки між таблицями. (рис.4)

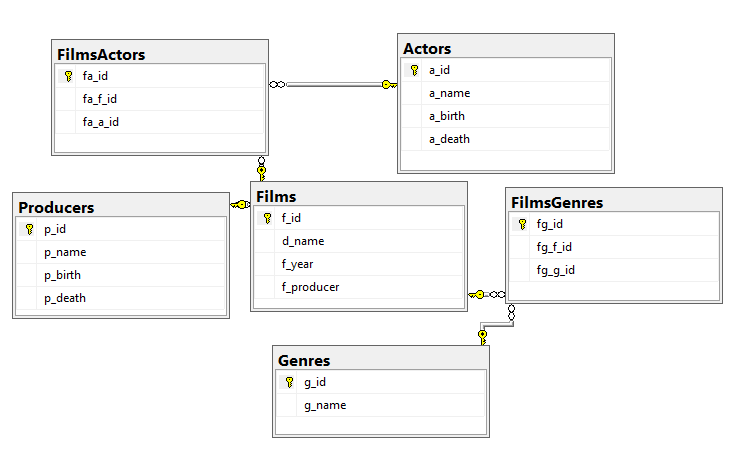


Рис 4. Візуалізація Бази Даних

***Films*** *– список фільмів, наявних у системі*

***Actors*** *– список акторів, які можуть зніматися у фільмах*

***Genres*** *– список жанрів, які можуть мати фільми*

***Producers*** *– список продюсерів, які можуть мати фільми*

***FilmsActors*** *– допоміжна таблиця для зв’язку акторів із фільмами*

***FilmsGenres*** *– допоміжна таблиця для зв’язку жанрів із фільмами*

# 3.1 Опис таблиць

У таблицях 1-8 наведено опис кожного поля таблиці.

Таблиця 1. Films

|  |  |  |
| --- | --- | --- |
| f\_id | int | Id фільму |
| d\_name | nvarchar(MAX) | Назва фільму |
| f\_year | int | Рік зйомки |
| f\_producer | int | Id продюсера |

Таблиця 2. Producers

|  |  |  |
| --- | --- | --- |
| p\_id | int | Id продюсера |
| p\_name | nvarchar(MAX) | Ім’я продюсера |
| p\_birth | int | Рік народження |
| P\_death | int | Рік смерті |

Таблиця 3. Actors

|  |  |  |
| --- | --- | --- |
| a\_id | int | Id актора |
| a\_name | nvarchar(MAX) | Ім’я актора |
| a\_birth | int | Рік народження |
| a\_death | int | Рік смерті |

Таблиця 4. Genres

|  |  |  |
| --- | --- | --- |
| g\_id | int | Id жанру |
| g\_name | nvarchar(MAX) | Назва жанру |

Таблиця 5. FilmsActors

|  |  |  |
| --- | --- | --- |
| fa\_id | int | Id ланки |
| fa\_f\_id | int | Id фільму |
| fa\_a\_id | int | Id актора |

Таблиця 6. FilmsGenres

|  |  |  |
| --- | --- | --- |
| fg\_id | int | Id ланки |
| fg\_f\_id | int | Id фільму |
| fg\_g\_id | int | Id жанру |

# 

# Табличні дані

На рисунках 5-10 наведено приклад заповнення полів таблиць.

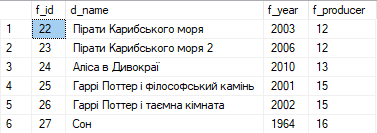


Рис 5. Таблиця Фільмів

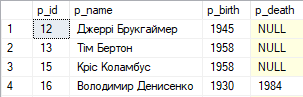


Рис 6. Таблиця Продюсерів



Рис 7. Таблиця Акторів

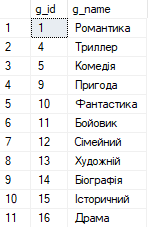


Рис 8. Таблиця Жанрів

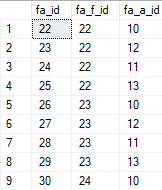


Рис 9. Таблиця зв’язку фільмів із акторами

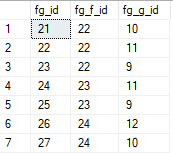


Рис 10. Таблиця зв’язку фільмів із жанрами

Ця модель бази чудово підходить для збереження інформації про фільми, є зручною для реалізації та досить гнучкою у модернізації.

# Розділ 4. Інструкція користувача

Після запуску програми користувач бачить стартову сторінку (меню), на якій може виконати попередньо створені запити й перейти до перегляду всіх таблиць. (рис. 11)

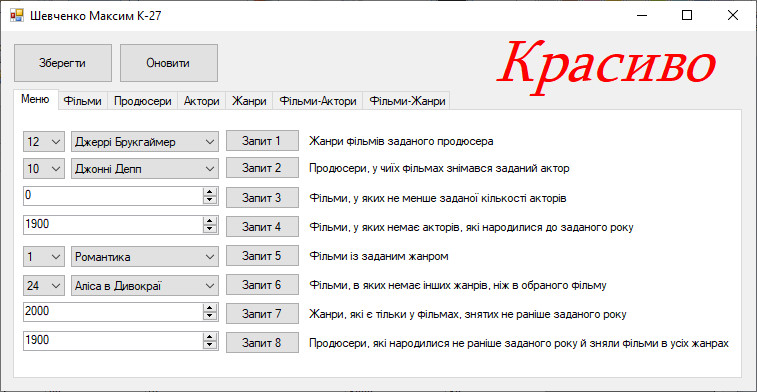


Рис 11. Початкова сторінка

Кожна вкладка має відповідну назві таблицю із бази даних. (рис. 12)

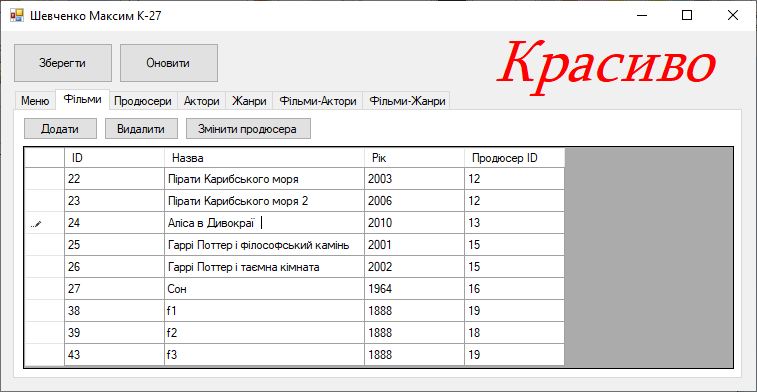


Рис 12. Сторінка фільмів

Кнопка «Зберегти» зберігає всі внесені зміни. Кнопка «Оновити» оновлює інформацію в таблицях. Автоматичне оновлення таблиць було заблоковане, щоб користувач не втратив внесені зміни в разі, якщо забув зберегти дані. При цьому дані в базу даних додаються.

Користувач може додавати необхідні дані в базу даних, натиснувши кнопку «Додати». Після цього з’явиться форма додавання нового елемента. (рис. 13)

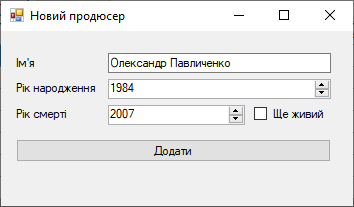


Рис 13. Сторінка додавання продюсера

Щоб видалити дані, необхідно обрати елемент для видалення (перемістити вказівник на будь-яке поле відповідного рядка) й натиснути кнопку «Видалити». Якщо елемент бере участь у зв’язках, з’явиться повідомлення про це й елемент не буде видалено. (рис. 14) Щоб видалити такий елемент, потрібно спочатку видалити зв’язок.

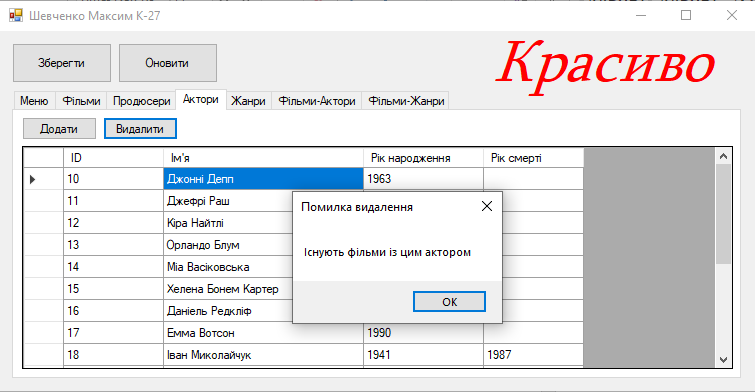


Рис 14. Помилка видалення

Для забезпечення зручності та коректності зміни продюсера фільму була додана спеціальна кнопка «Змінити продюсера». Якщо її натиснути, з’явиться форма, на якій можна обрати нового продюсера для фільму. (рис. 15)

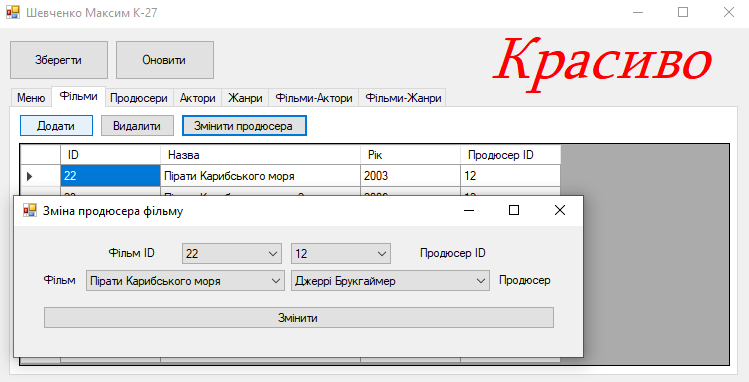


Рис 15. Зміна продюсера

Користувач може змінювати кожне поле, за винятком ID, які відображаються лише для зручності пошуку й навігації. У разі некоректності змінених даних зміни будуть скасовані та з’явиться повідомлення про некоректний ввід. (Додаток А) Варто зазначити, що перевірка коректності даних відбувається й при додаванні нового елемента. (Додаток Б)

Щоб переглянути детальну інформацію про фільм, необхідно двічі натиснути на відповідне даному фільму поле в таблиці. Після цього з’явиться форма з усією інформацією, наявною на цей фільм (рис.16)

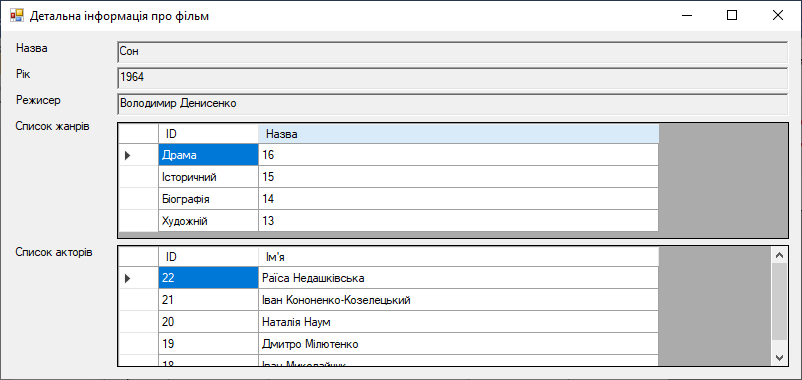


Рис 16. Детальна інформація про фільм

На головній сторінці (вкладка «Меню») можна виконати попередньо створені SQL-запити. Результат буде відображений на новій формі. (рис. 17)

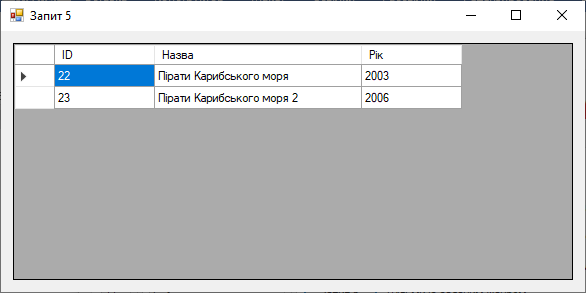


Рис 17. Результат запиту

# Розділ 5. Запити

Користувач може виконати описані нижче запити.

1. Жанри фільмів заданого продюсера

SELECT DISTINCT Genres.g\_id, Genres.g\_name

FROM Genres INNER JOIN

FilmsGenres ON FilmsGenres.fg\_g\_id = Genres.g\_id INNER JOIN

Films ON Films.f\_id = FilmsGenres.fg\_f\_id

WHERE (Films.f\_producer = @pid)

2. Продюсери, у чиїх фільмах знімався заданий актор

SELECT DISTINCT Producers.p\_name, Producers.p\_birth, Producers.p\_death, Producers.p\_id

FROM Actors INNER JOIN

FilmsActors ON FilmsActors.fa\_a\_id = Actors.a\_id INNER JOIN

Films ON Films.f\_id = FilmsActors.fa\_f\_id INNER JOIN

Producers ON Producers.p\_id = Films.f\_producer

WHERE (Actors.a\_id = @aid)

3. Фільми, у яких не менше заданої кількості акторів

SELECT Films.f\_id, Films.d\_name, Films.f\_year

FROM Films INNER JOIN

FilmsActors ON FilmsActors.fa\_f\_id = Films.f\_id INNER JOIN

Actors ON Actors.a\_id = FilmsActors.fa\_a\_id

GROUP BY Films.d\_name, Films.f\_id, Films.f\_year

HAVING (COUNT(DISTINCT Actors.a\_id) >= @n)

1. Фільми, у яких немає акторів, які народилися до заданого року

SELECT DISTINCT f\_id, d\_name, f\_year

FROM Films AS x

WHERE (NOT EXISTS

(SELECT Films.f\_id

FROM Films INNER JOIN

FilmsActors ON FilmsActors.fa\_f\_id = Films.f\_id INNER JOIN

Actors ON Actors.a\_id = FilmsActors.fa\_a\_id

WHERE (Actors.a\_birth <= @year) AND (Films.f\_id = x.f\_id)))

1. Фільми із заданим жанром

SELECT Films.f\_id, Films.d\_name, Films.f\_year

FROM Films INNER JOIN

FilmsGenres ON FilmsGenres.fg\_f\_id = Films.f\_id INNER JOIN

Genres ON FilmsGenres.fg\_g\_id = Genres.g\_id

WHERE (Genres.g\_id = @gid)

1. Фільми, в яких немає інших жанрів, ніж в обраного фільму

SELECT DISTINCT x.f\_id, x.d\_name, x.f\_year

FROM Films AS x

WHERE (x.f\_id NOT IN

(SELECT DISTINCT z.fg\_f\_id

FROM FilmsGenres AS z

WHERE (z.fg\_g\_id NOT IN

(SELECT FilmsGenres.fg\_g\_id

FROM FilmsGenres INNER JOIN

Genres ON Genres.g\_id = FilmsGenres.fg\_g\_id

WHERE (FilmsGenres.fg\_f\_id = @fid) ))))

1. Жанри, які є тільки у фільмах, знятих не раніше заданого року

SELECT g\_id, g\_name

FROM Genres AS x

WHERE (NOT EXISTS

(SELECT Films.f\_id

FROM Films INNER JOIN

FilmsGenres ON FilmsGenres.fg\_f\_id = Films.f\_id INNER JOIN

Genres ON Genres.g\_id = FilmsGenres.fg\_g\_id

WHERE (Films.f\_year < @year) AND (Genres.g\_id = x.g\_id))) AND EXISTS

(SELECT fg\_id

FROM FilmsGenres AS FilmsGenres\_1

WHERE (fg\_g\_id = x.g\_id))

1. Продюсери, які народилися не раніше заданого року й зняли фільми в усіх жанрах

SELECT p\_id, p\_name, p\_birth, p\_death

FROM Producers AS x

WHERE (p\_birth >= @year) AND (NOT EXISTS

(SELECT g\_id

FROM Genres

WHERE (g\_id NOT IN

(SELECT FilmsGenres.fg\_g\_id

FROM Producers INNER JOIN

Films ON Producers.p\_id = Films.f\_producer INNER JOIN

FilmsGenres ON FilmsGenres.fg\_f\_id = Films.f\_id

WHERE (Producers.p\_id = x.p\_id)))))

# Висновки

У даній лабораторній роботі було проаналізовано ринок існуючих сервісів та програм, які займаються збором інформації про фільми. Було досліджено характерні риси, переваги та недоліки кожної з них. Спільні базові риси були виокремлені та реалізовані в готовому продукті. Також у програмі було створено простий, легкий та дружній інтерфейс для додавання, видалення та редагування інформації ресурсу.

Продукт було створено на основі власної бази даних, яку легко можна модернізувати. Мову програмування та середовище розробки було обрано із розрахунку на наявний досвід роботи. Створено простий та дружній інтерфейс, що орієнтується на потреби користувача. Поглиблено знання з використання технологій ADO.Net та Microsoft SQL Server.

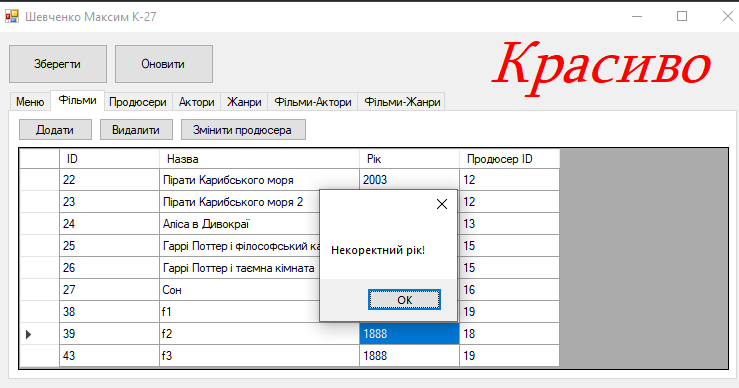
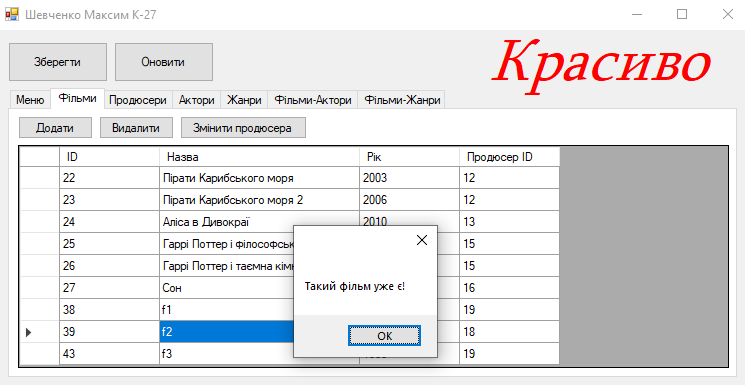
Використовувалися допоміжні навчальні матеріали та освітні ресурси, які дозволили полегшити процес розробки та стабільність системи. Реалізована система виконує всі покладені на неї вимоги та легка в модернізації. Отже, проект є актуальним і потрібним на сьогоднішній день та планується його подальший розвиток у майбутньому.

# Список використаних джерел

1. Visual Studio [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://visualstudio.microsoft.com/ru/>.
2. Windows 10 [Електронний ресурс] / Corporation. Microsoft. – 2018. – Режим доступу до ресурсу:<https://www.microsoft.com/uk-ua/windows>
3. ADO.NET [Електронний ресурс] / Corporation. Microsoft. – 2017. – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/>
4. Microsoft SQL Server [Електронний ресурс]. – 2017. – Режим доступу до ресурсу:<https://www.microsoft.com/ru-ru/sql-server/sql-server-2017-editions>.
5. Moviestape.net [Електронний ресурс]. – Режим доступу до ресурсу: <http://moviestape.net/>
6. uaKINO.club [Електронний ресурс]. – Режим доступу до ресурсу: <https://uakino.club/>
7. KINIGO.co.ua [Електронний ресурс]. – Режим доступу до ресурсу: <https://kinogo.co.ua/>
8. Windows Vista [Електронний ресурс] / Corporation. Microsoft. – 2007. – Режим доступу до ресурсу:<https://www.microsoft.com/uk-ua/windows>
9. C Sharp [Електронний ресурс] / Corporation. Microsoft. – 2000. – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/>

# ДОДАТКИ

Додаток А



Додаток Б

