

Даний матеріал наводиться в демонстративних цілях. Через це файл не відображає повний обсяг роботи. Для отримання цілісної інформації по цій чи іншій роботі звертайтеся до менеджерів сайту kursach.in.ua

Зміст

Зміст.....	1
Вступ.....	3
1. Загальні відомості про предметну область	4
2. Структура таблиць та загальна логістика	7
2.1 Створення таблиць.	8
2.1 Створення запитів.....	11
2.2 Логістика програми.....	13
3. Техніка безпеки.....	19
Висновки.....	19
Список використаних джерел.....	21

ДОДАТКИ

Додаток А. Блок-схема програмного продукту.

Додаток В. Лістинг програмного продукту.

Додаток С. Скрін-шоти функціонування програмного продукту

Вступ

Використання баз даних є однією з характерних рис більшості сучасних інформаційних систем. По своїй суті бази даних є тим, навколо чого і будується інформаційна система будь-якого підприємства. Тому теорії створення та практиці використання баз даних приділяється достатня увага протягом періоду функціонування інформаційних систем. Досить тривалий час основним типом були реляційні бази даних, які на сьогодні вже вважаються класичними. Проте розвиток інформаційних систем поставив перед сучасними базами даних завдання, вирішення яких неможливе в межах використання тільки реляційних баз даних. Крім класичних завдань, сучасні бази даних повинні забезпечувати багатомашинну обробку та зберігання великих обсягів інформації, оперативний аналіз даних, інтеграцію із мережею Інтернет, розмежування доступу користувачів до зберігаємої інформації, захист інформації під час її передачі по мережі. Хоча на практиці і використовується чимало різноманітних баз даних, але для більшості з них існує велика кількість спільних ознак, як з погляду розробки, так і використання. Це дає можливість вивчати сучасні бази даних і відповідне прикладне та системне програмне забезпечення на прикладах, які, незважаючи на свою новизну, вже стали класичними. Як такі приклади вибрано загальні питання проектування, розробки та використання бази даних Microsoft SQL Server. Це пояснюється тим, що Microsoft SQL є однією із найпоширеніших і вдосконалених баз даних.

В даному курсовому проекті за предметну область взято базу даних яка буде зберігати та виводити з таблиць всю інформацію про топ-50 кінофільмів на екран в програмі Microsoft Visual Studio 2017 (C# & MsSQL). Інформацію про фільми, можна буде редагувати та оновляти за допомогою операторів T-SQL: «UPDATE», «INSERT», «DELETE».

Даний проект являє собою адміністративну панель рейтингу IMDb, з підтримкою можливості підключення до хосту напряму.

1. Загальні відомості про предметну область

База даних (БД) — впорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД.

Головним завданням БД є гарантоване збереження значних обсягів інформації (т.зв. записи даних) та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин : збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Кожна створена база даних має свою предметну область. Предметна область – це необхідний для розробки бази даних об'єкт, який має в собі дані, які будуть зберігатися в базі даних.

Предметна область моєї бази даних – рейтинг фільмів за шкалою IMDb (Internet Movies Database, в перекладі з англ. – «Інтернет-база кіномільфів»).

Найбільша в світі база даних та веб-сайт про кінематограф. В базі зібрана інформація (станом на грудень 2017 року) про більш ніж 4,7 млн кінофільмів, телесеріалів і окремих їх серій, а також про 8,3 млн інтерв'ю, пов'язаних з кіно, - акторах, режисерах, сценаристах тощо.

З 1998 року базою даних IMDb володіє компанія Amazon.com. Майже вся інформація IMDb знаходиться у вільному доступі, сайт IMDb.com

функціонує на основі незалежного програмного забезпечення (Apache, Perl та ін.).

Проектування БД здійснюється на основі трьохрівневої архітектури. Наочно це представлено на рис.1.

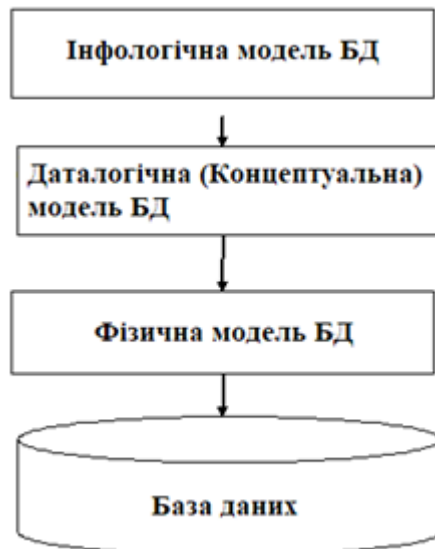


Рис.1

1.1. Інфологічна модель БД – це модель, яка описує дані предметної області з використанням мови людини. Початковими даними можуть бути стандартні довідки, бланки, документи. Інформаційно-логічна модель – це модель, в якій визначена логіка відношень даних між собою, також ця модель людино-орієнтована.

1.2. Даталогічна модель (Концептуальна) БД – це логічна організація даних та їх взаємозв'язок. Структурування даних виконується згідно вибраної СУБД, яка будується на основі вибраної моделі представлення даних: ієрархічна, мережева чи реляційна. На даному етапі проектуючий створює структуру даних і організовує зв'язок між об'єктами.

1.3. Фізична модель БД – це структура БД, реалізована у вибраній СУБД.

Також при створенні БД важливо враховувати наступні параметри:

- Цілісність БД – правильність даних у будь-який момент часу.

- Надмірність БД – дублювання даних в декількох таблицях (надмірність веде до суперечливості даних, її намагаються виключити використанням ключових полів).

Ціль проектування БД – зменшення кількості надмірних даних. Це досягається за допомогою теорії нормалізації баз даних.

Нормалізація – це розбиття таблиць на дві чи більше, які володіють найкращими властивостями для внесення, чи зміни даних. В результаті виходить БД, в якій кожний факт з'являється лиш в одному місці, що виключає надмірність інформації.

Проектування фізичної моделі

Фізична модель БД визначає спосіб розміщення даних на носіях (пристроях зовнішньої пам'яті), а також спосіб і засоби організації ефективного доступу до них. Оскільки СУБД функціонує в складі і під управлінням операційної системи, то організація зберігання даних і доступу до них залежить від принципів і методів управління даними операційної системи.

На відміну від ранніх СУБД, багато сучасних системи не надають розробникові будь-якого вибору на цій стадії. Реально до питань проектування фізичної моделі можна віднести:

- вибір схеми розміщення даних (поділ по файлам або тип RAID-масиву);
- визначення числа і типу індексів (наприклад, кластеризований або некластеризований в разі MS SQL Server).

Спосіб зберігання БД визначається механізмами СУБД автоматично за замовчуванням на основі специфікацій концептуальної схеми БД, і внутрішня схема в явному вигляді в таких системах не використовується. Зовнішні схеми БД зазвичай конструюються на стадії розробки програм.

2. Структура таблиць та загальна логістика

Распишите все таблицы, ключи даже если Вы ими не пользуетесь, кроме словесного описания было бы не плохо указать все взаимосвязи в графическом представлении. Опишите логику Вашей программы. И не забудьте что количество листов не менее 20.

Сам процес розробки бази даних являє собою складний процес проектування відображення опису предметної області у схему внутрішньої моделі даних. Перебіг цього процесу є послідовністю більш простих процесів проектування менш складних відображень. Ця послідовність у процесі проектування весь час уточнюється, вдосконалюється таким чином, щоб були визначені об'єкти, їх властивості та зв'язки, які будуть потрібні майбутнім користувачам системи.

Етапи розробки бази даних у середовищі Microsoft SQL і Microsoft Visual Studio, які ми будемо реалізовувати:

- визначення таблиць, які повинна містити база даних;
- визначення структури таблиць (полів та їх типів);
- призначення ключів таблиць та створення потрібних індексів;
- визначення зв'язків між таблицями;
- завантаження даних;

2.1 Створення таблиць.

Таблиці складають основу бази даних - саме в них зберігаються всі дані. Перед усім, повинна бути спланована структура кожної таблиці. Структура таблиць обумовлюється вмістом тих вихідних форм, запитів та звітів, які повинні бути отримані при роботі з базою даних. При плануванні таблиць При плануванні таблиць необхідно уникати повторення колонок в різних таблицях, тільки якщо вони не слугують для визначення зв'язків між ними. Таблиця складається з записів (рядків), кожний з яких описує одну сутність. Кожна колонка таблиці – це поле.

Перша таблиця – Name. Її сутність полягає у тому, щоб вмістити в собі список найменувань об'єктів (назви фільму), за якими буде організовано

Таблиця Name

Містить в собі ідентифікатор фільму(ID), назву кінокартини, жанр. Загалом структура виглядає так:

Поле	Тип даних	Коментар
ID	int not null	Первинний
Name	Nvarchar(40)	Назва фільму
Genre	Nvarchar(MAX)	Жанр фільм

Таблиця створюється наступним кодом(СУБД MsSQL (Visual Studio 2017)):


```

CREATE TABLE [dbo].[Name] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [Name] NVARCHAR (40) NULL,
    [Genre] NVARCHAR (MAX) NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC)
);

```

Заповнення таблиці даними проводилось через внутрішній інструмент Microsoft Visual Studio.

	Id	Name	Genre
▶	1	Побег из Шоу...	драма
	2	Крестный отец	детектив, драма
	3	Крестный отец 2	детектив, драма
	4	Темный рыцарь	боевик, детект...
	5	12 разгневанн...	детектив, драма
⊙	NULL	NULL	NULL

Рис. 2

Таблиця Films

Містить в собі ідентифікатор фільму (ID), назву кінокартини, жанр, рік випуску, ім'я режиссера та номер місця у ТОП-рейтингу за шкалою IMDb. Зовнішній вигляд структури:

Поле	Тип даних	Коментар
ID	Int not null	Первинний
Name	Nvarchar(40)	Назва фільму
Genre	Nvarchar(MAX)	Жанр фільму
Year	Nvarchar(4)	Рік випуску
Producer	Nvarchar(50)	Ім'я режисера
Top1	Int not null	Місце у топі

Таблиця створюється наступним кодом(СУБД MsSQL):

```
CREATE TABLE [dbo].[Films] (
    [Name] NVARCHAR (40) NULL,
    [Genre] NVARCHAR (MAX) NULL,
    [Year] NVARCHAR (4) NULL,
    [Producer] NVARCHAR (50) NULL,
    [Top1] INT NULL,
    [id] INT IDENTITY (1, 1) NOT NULL,
    CONSTRAINT [PK_Films] PRIMARY KEY CLUSTERED ([id] ASC)
);
```

Заповнення таблиці даними проводилось через внутрішній інструмент Microsoft Visual Studio. Обмеження в 1000 рядків створене, за для нормалізування роботи інструменту, та проти неуважності нових користувачів.

Name	Genre	Year	Producer	Top1	id
Побег из Шоу...	драма	1994	Фрэнк Дарабонт	1	1
Крестный отец	детектив, драма	1972	Фрэнсис Форд ...	2	2
Крестный отец 2	детектив, драма	1974	Фрэнсис Форд ...	3	3
Темный рыцарь	боевик, детект...	2008	Кристофер Но...	4	4
12 разгневанн...	детектив, драма	1957	grrr	5	5
NULL	NULL	NULL	NULL	NULL	NULL

Рис. 3

Логічний зв'язок між таблицями, можливо скласти у однойменних полях ID та Name. Проте, не використовується в проекті, через відсутність рентабельності даного процесу.

2.1 Створення запитів

Запит Select для таблиці Name. Являє собою считування усіх даних з таблиці Name:

```
SELECT * FROM [Name]
```

Запит Select для таблиці Films. Являє собою считування усіх даних з таблиці Films, де поля Name з обох таблиць тотожні.

```
SELECT * FROM Films WHERE Name=@Name
```

Запит Insert для таблиці Name. Спосіб дії: вставляє у таблицю table (змінна типу даних string, що містить у собі назву потрібної таблиці) новий рядок даних, з джерел, в поля Name, Genre. Так як поле ID інкрементоване, його записувати не потрібно.

```
INSERT INTO [" + table + "] (Name, Genre)VALUES(@Name, @Genre)
```

Запит Insert для таблиці Films. Спосіб дії: вставляє нові дані, введені користувачем, у таблицю “table” (де table – змінна, з назвою потрібної нам таблиці) рядок, що містить поля Name, Genre, Year, Producer, Top1.

```
INSERT INTO [" + table + "] (Name, Genre, Year, Producer,  
Top1)VALUES(@Name, @Genre, @Year, @Producer, @Top1
```

Запит Delete для всіх двох таблиць. Видаляє рядок даних, в якому поле ID ідентичне введеному користувачем.

```
DELETE FROM [" + table + "] WHERE [id]=@id
```

ЗапитUpdate для всіх двох таблиць та любого вибраного поля. Оновляє зміст всіх полів в рядку, в якому поле ID еквівалентне введеному користувачем.

```
UPDATE [" + table + "] SET [" + Name + "]=@" + Name + " WHERE [id]=@id
```

2.2 Логістика програми

ASP.NET Core – API технологія, за допомогою якої проект був зв’язаний з системою управління базами даних. Загальноприйняте визначення: технологія створення веб-застосунків і веб-сервісів від компанії Майкрософт. Вона є складовою частиною платформи Microsoft.NET і розвитком старішої технології Microsoft ASP. На цей час останньою версією цієї технології є ASP.NET Core 2.0

Хоча ASP.NET бере свою назву від старої технології Microsoft ASP, вона значно від неї відрізняється. Microsoft повністю перебудувала ASP.NET, ґрунтуючись на Common Language Runtime (CLR), який є основою всіх застосунків Microsoft .NET. Розробники можуть писати код для ASP.NET, використовуючи практично будь-які мови програмування, що входять у комплект .NET Framework (C#, Visual Basic.NET, і JScript.NET). ASP.NET має перевагу у швидкості в порівнянні зі скриптовими технологіями, тому що при першому зверненні код компілюється і поміщається в спеціальний кеш, і згодом тільки виконується, не вимагаючи витрат часу на парсинг, оптимізацію, і т. д.

Переваги:

- ASP.NET має перевагу у швидкості в порівнянні з іншими технологіями, заснованими на скриптах (PHP, тощо)
- Розширюваний набір елементів управління і бібліотек класів дозволяє швидше розробляти застосунки
- ASP.NET спирається на багатомовні можливості .NET, що дозволяє писати код сторінок на C#, VB, C/C++ та ін.
- Поділ візуальної частини та бізнес-логіки
- Розширювана модель обробки запитів

Форма 1

Ця форма є головним меню проекту. Має обмежений функціонал: дві активні кнопки, та стандартна панель інструментів «згорнути, повний екран, вихід». Перша кнопка – «Начать», розпочне роботу програми, та відкриє Форму 2; встановить зв’язок з системою управління базами даних, та автоматично

зачинить саму себе. Друга кнопка – «Выход», повністю вимкне програму, звільнив кеш, потоки, та розірвав (при наявності) з'єднання з базою даних.

Для заднього плану форми було використано .png файл, зроблений власноруч у програмі Adobe Photoshop. Прив'язка до форми була зроблена через файли-ресурси (Project Resources).

Усі елементи керування форми 1 заблоковані, користувач не зможе змінити їх стан, розмір чи положення на формі.

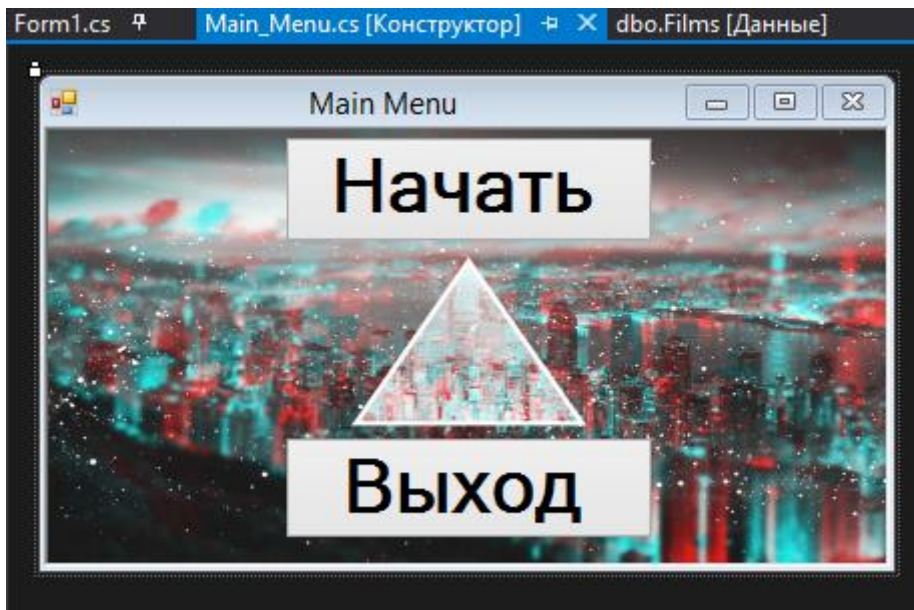


Рис. 4

Форма 2

Основна форма (Рис. 5), на котрій відбувається абсолютно весь зв'язок з системою управління базами даних, базами даних котрі присутні у проекті, API ASP .Net Core 2.0 (ASP .Net 5).

Datagridview1 (Рис. 6) – Перший датагрід, котрий візуально виводить користувачу (GUI англ. – користувацький інтерфейс) дані з таблиці Name, за допомогою оператора Select. Заповнення датагріду відбувається в одночас із запуском другої форми, при натисканні кнопки «Начать» на першій формі. Датагрід має в собі три поля (стовпці) – id, Name, Genre, котрі аналогічні по собі з полями таблиці Name. При натисканні на зміст поля Name буде викликано об'єкт користувацького інтерфейсу MessageBox, в якому буде вказано назву фільму, на котрий натиснув користувач. Разом з цим,

відбудеться здійснення оператора Select, з параметром where, для Datagridview2 (Рис. 7)

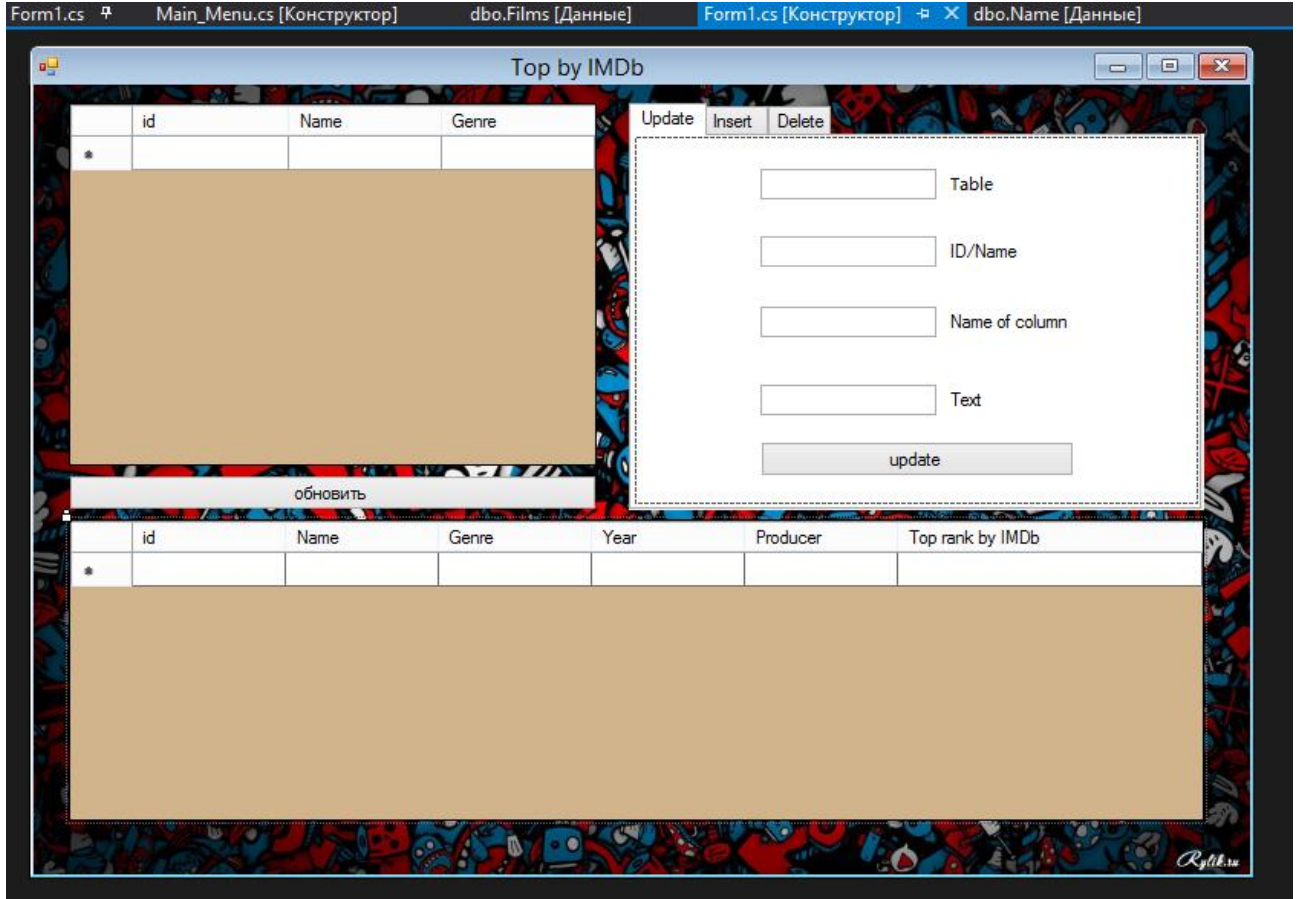


Рис. 5

	id	Name	Genre
*			

Рис. 6

Рис. 7

Datagridview2 (Рис. 8) – другий датагрід, котрий заповнюється при натисканні на зміст комірки поля Name в першому датагріді. Загальний опис аналогічний першому датагріду, з яким має схожу функцію – впливаючий MessageBox, при натисканні на любую комірку з даними.

	id	Name	Genre	Year	Producer	Top rank by IMDb
▶	1	Побег из ...	драма	1994	Фрэнк Д...	1
*						

Рис. 8

TabControl1 (Рис. 9, 10, 11) – елемент управління, котрий створює декілька фреймів, між котрими можна перемикатися в будь-який момент.

Використовується для економії простору. В проекті, використано три tabpages (листи або сторінки, фрейми), кожен з котрих відповідає за, та розміщує на собі елементи керування, такі як textbox, label, button, за для здійснення базових операторів T-SQL – Insert, Update, Delete.

Рис. 9

Кожен з фреймів працює за допомогою новітньої розробки – асинхронування. Прийняте у opensource (вільний доступ для користування) лише в версії .Net 5, асинхронування дозволяє проводити різні взаємозв'язані, чи окремі процеси в одночас, у різних потоках. Це дозволяє уникнути можливих проблем з послідовним виконанням, та вагомо підвищити швидкість роботи програми.

Update Insert Delete

Table

Films

Name

Name

Genre

Year

Producer

Top rank

insert

Рис. 10

Update Insert Delete

Table

ID/Name

delete

Рис. 11

Обновить (Рис. 12) – кнопка оновлення інформації. За типом процесу, вона зупиняє з'єднання з базою даних, закриває всі активні зв'язки з класом SQL, API ASP .Net, а потім, за декілька десятих секунди, відновлює, запускаючи заново. Ця кнопка потребується, за для успішного виведення в перший та другий датагрід інформації, котра була змінена, за допомогою операторів Insert, Update, чи Delete.

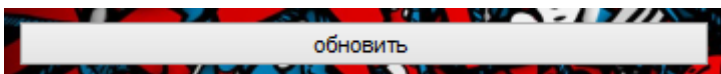


Рис. 12

3. Техніка безпеки

Комп'ютерне обладнання повинне підключатися до електромережі лише за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їх конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Не допускається підключати комп'ютерну техніку до звичайної двопровідної електромережі, зокрема з використанням перехідних пристроїв.

Висновки

Згідно індивідуальним завданням, мета даної роботи полягає у проектуванні та розробці бази даних «Рейтинг IMDb». На першому етапі були визначені переваги баз даних та програмного забезпечення перед іншими схемами впорядкування даних і обрані для програмної реалізації бази даних рейтингу кінокартин мови SQL і C#.

Для реалізації поставленої мети ми скористалися двома методами для проектування БД: метод Сутність - зв'язок і метод нормалізації відношень. А саме: провели детальний аналіз предметної області, сконструювали візуальну модель за створення сутності и зв'язками, провели декомпозицію та нормалізацію відношень методом Сутність-зв'язок. Також було створено універсальне відношення, виявлено первинний ключ, проведено нормалізацію відношень до третьої нормальної форми.

В останньому третьому розділі була реалізована проектована база даних, а так само форми та звіти, котрі в зручному вигляді відображають потрібну нам інформацію.

Список використаних джерел

- 1. Кузнецов С. Д. Основы баз данных. Курс лекций. – М.: Интернет-университет информационных технологий, 2005.
- Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. SQL: полный справочник. – М.: Вильямс, 2011.
- Алекс Кригель, Борис Трухнов: SQL. Библия пользователя, Москва: Вильямс, 2010.
- М. Фленов, Transact-SQL, Санкт-Петербург: БХВ- Петербург, 2006.
- Александр Бондарь, Microsoft SQL Server 2012, Санкт- Петербург: БХВ-Петербург, 2013.
- Кевин Е. Кляйн, Дэниэл Кляйн, Брэнд Хант SQL. Справочник, Москва: Символ-Плюс, 2010.
- Тоу, Дэн Настройка SQL; СПб: Питер, 2009.
- Гринченко, Н.Н. и др. Проектирование баз данных. СУБД Microsoft Access; Горячая Линия Телеком, 2012.
- Душан Петкович, Microsoft SQL Server 2012. Руководство для начинающих, Санкт-Петербург: БХВ-Петербург, 2013.
- Зрюмов Е.А., Зрюмова А.Г. Базы данных для инженеров: Учебное пособие. - Барнаул: Изд-во АлтГТУ, 2010.
- Нейгел К., Ивѐн Б., Глинн Дж., Уотсон К. C# 4.0 и платформа .NET 4 для профессионалов, 2011.
- Руководство пользователя по Fast Reports, Fast Reports Inc, 2008-2013.
- И.И. Семенова Разработка клиент-серверных приложений в MicrosoftSQL Server 2005, Омск: СибАДИ, 2010.
- Дейт К. Дж. Введение в системы баз данных. – М.: Вильямс, 2005.