Оглавление

[Вступ……………….. 6](#_Toc481141988)

[1. Аналіз предметної області і постановка задачі 7](#_Toc481141989)

[1.1 Аналіз предметної області 7](#_Toc481141990)

[1.2 Постанова задачі 8](#_Toc481141991)

[2. Концептуальна модель бази даних 10](#_Toc481141992)

[2.1 Сутності та їх атрибути 10](#_Toc481141993)

[4. Фізична модель бази даних 14](#_Toc481141994)

[5. Опис реалізації проекту бази даних 15](#_Toc481141995)

[5.1 Опис інтерфейсу розробленої системи функціональне призначення основних опцій меню 15](#_Toc481141996)

[5.2 Опис метаданих, що зберігаються на сервері 21](#_Toc481141997)

[5.2.1 Таблиці 21](#_Toc481141998)

[5.2.2 Тригери 23](#_Toc481141999)

[5.2.3 Збережені процедури 24](#_Toc481142000)

[6. Висновки……………… 25](#_Toc481142001)

[Список використаних джерел 26](#_Toc481142002)

# [Вступ](file:///C:\Users\111\Desktop\Курсовой%20по%20БД\Пояснювальна%20записка%20(шаблон).docx#_Toc417391100)

Похід до кінотеатру – один з найпопулярніших способів проведення дозвілля в наш час. Це величезна область ринку розваг, в якій задіяно безліч працівників. Одночасно в кінотеатрі може проводитися безліч сеансів з різним розкладом, фільмами, різними залами. Тому ця індустрія дуже потребує автоматизації, що дозволить спростити роботу в ній.

Предметною областю мого курсового проекту є робота саме касиру кінотеатру. Необхідно було розробити систему для автоматизації його робочого місяця.

Для вирішення даної проблеми потрібно було розробити базу даних та програмне забезпечення для неї.

# [Аналіз предметної області і постановка задачі](file:///C:\Users\111\Desktop\Курсовой%20по%20БД\Пояснювальна%20записка%20(шаблон).docx#_Toc417391101)

## 1.1 [Аналіз предметної області](file:///C:\Users\111\Desktop\Курсовой%20по%20БД\Пояснювальна%20записка%20(шаблон).docx#_Toc417391102)

Кінотеатр - громадська будівля або її частина, обладнані для публічної де-монстрації кінофільмів. Головне приміщення кінотеатру - зал для глядачів з екраном великого розміру і системою відтворення звуку, що складається з декількох гучномовців, які забезпечують об'ємний звук.

Основні функції кінотеатру:

* Прокат фільмів;
* Продаж білетів;
* Ведення розпорядку сеансів.

База даних створюється саме для спрощення функції продажу білетів на сеанси. Користувачем системи буде касир. Основний обов’язок касира – продаж білетів.

Організація ведення продажу білетів на сеанси така: працює продавець; він веде власний журнал обліку продажу білетів; при покупці білету клієнтом продавець повинен записати інформацію про продаж таким чином (назва кінотеатру, назва фільму, назва кінозалу, дата сеансу, ряд та місце, ціна білету, дата продажу, номер білету). Також, при зверненні покупця продавець може проконсультувати його при виборі фільму для перегляду. Така можливість забезпечує зручну організацію продажу білетів, як для клієнта, так і для продавця.

Для забезпечення продажу білетів треба забезпечити побічні можливості редагування – редагування списку сеансів, фільмів, кінозалів. Цим може займатися касир або адміністратор, тому потрібно зробити систему, розрахована на багато користувачів з різним рівнем доступу.

Для надання користувачу способу роботи з БД також необхідно розробити графічний інтерфейс, який забезпечує зручну роботу з БД.

Актуальність розробки бази даних полягає у тому, що кінотеатри дуже поширені в наш час, при цьому, організація роботи в ньому дуже складна, а дана розробка дозволяє спрощувати їх роботу.

## 1.2 Постанова задачі

База даних «Автоматизація робочого місця продавця білетів кінотеатру» створюється для економії часу при обліку продажу білетів. Завдяки системі можливо фіксувати сеанси, які проходять в кінотеатрі, перевіряти які білети продані, додавати в базу даних нові кінозали, фільми, сеанси.

Вхідні дані:

* Інформація про продані білети (назва кінотеатру, назва фільму, назва кінозалу, дата сеансу, ряд та місце, ціна білету, дата продажу, номер білету);
* Інформація про фільми в прокаті (назва фільму, віковий рейтинг, опис фільму);
* Розпорядок сеансів (час, фільм, зал);
* Структура кінозалу (ім’я та об’єкт, який описує структуру місць кінозалу(скоріше за все – це стрічка з символами)).

Вихідні дані:

* Звітність про продані білети;
* Список сеансів;
* Список кінозалів;
* Список фільмів;
* Статистика.

Функції:

* Отримання списків сеансів, кінотеатрів, фільмів, білетів;
* Отримання звітів про продаж;
* Формування списку загальної суми продажу білетів на кожний сеанс;
* Виведення списку білетів на схемі кінотеатру;
* Отримання списку білетів проданих на даний сеанс;
* Пошук за різними критеріями.

Системні вимоги:

* Універсальність (не прив’язана до конкретного кінотеатру);
* Розмежування рівнів доступу між користувачами.

# Концептуальна модель бази даних

## Сутності та їх атрибути

Опис таблиці «Таблиця сутності та атрибути» показаний у таблиці 2.1

Таблиця 2.1 – Таблиця сутності та атрибути

|  |  |
| --- | --- |
| Сутність | Атрибути |
| 1 | 2 |
| Фільм | Назва  Опис |
| Проданий білет | Ряд  Місце  Дата продажу  Ціна  Продавець  Сеанс |
| Сеанс | Кінозал  Фільм  Дата сеансу |
| Кінозал | Назва  Схема кінозалу |
| Продавець | Логін  Пароль  Ім’я  Рівень доступу |

1. Діаграма сутність-зв’язок представлена на рисунку 2.1

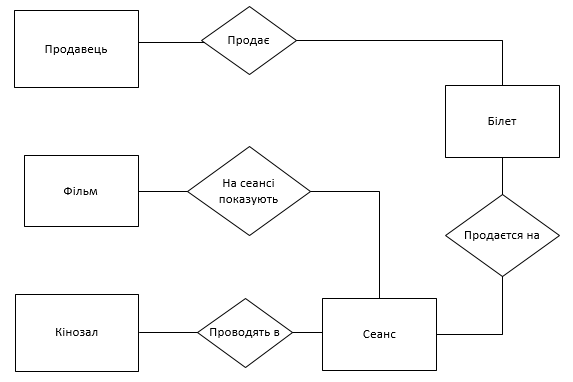


Рисунок 2.1 – ER-діаграма

1. Логічна модель даних представлена реляційною моделлю даних (Рис. 3.1)

Фільм – films:

* Назва – film\_name;
* Опис – description.

Проданий білет – sold\_tickets:

* Ряд – seat\_row;
* Місце – seat\_column;
* Дата продажу – sale\_date;
* Ціна – price;
* Продавець – seller\_id;
* Сеанс – séance\_id.

Сеанс – séance\_schedule:

* Кінозал – cinema\_room\_id;
* Фільм – film\_id;
* Дата сеансу – séance\_date.

Кінозал – cinema\_rooms:

* Назва – room\_name;
* Схема кінозалу – room\_scheme\_object.

Продавець – sellers:

* Логін – login;
* Пароль – pass;
* Ім’я – name;
* Рівень доступу – rules\_level.

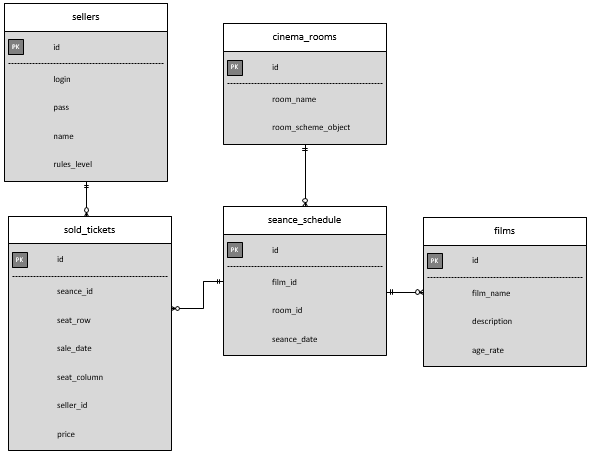


Рисунок 3.1 –Логічна модель

# Фізична модель бази даних

База даних реалізована засобами СУБД MySQL за допомогою графічного редактору dbForge Studio.

Опис структури бази даних показаний на Рис. 4.1

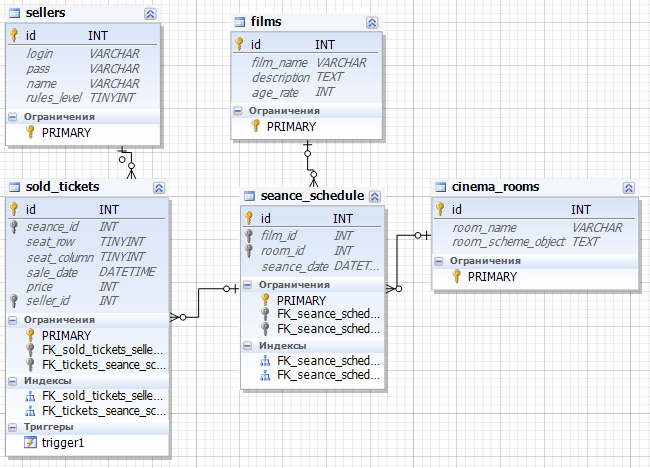


Рисунок 4.1 – Фізична модель БД

# Опис реалізації проекту бази даних

Моє програмне забезпечення побудовано за принципами клієнт-серверної моделі. Для реалізації такої моделі був використаний сервер MySql, на якому зберігається структура даних, збережені процедури, тригери та самі дані, та розроблений графічний інтерфейс користувача на стороні клієнта.

## 5.1 Опис інтерфейсу розробленої системи функціональне призначення основних опцій меню

Графічний інтерфейс системи розроблений засобами мови програмування C# в ide Visual Studio.

Розроблене програмне забезпечення дозволяє переглядати, вносити, видаляти та редагувати дані таблиць films, cinema\_rooms, seance\_schedule, sold\_tickets, а також отримувати їх списки.

Після запуску застосування відкривається головна форма, на якій розташовані кнопки переходу (Рисунок 5.1).

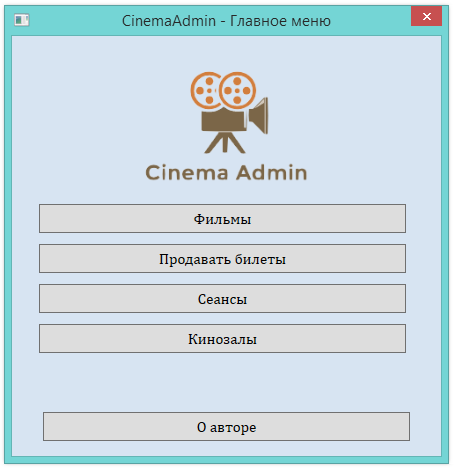


Рисунок 5.1 – Головна форма.

При натисканні на кнопки «Фильми», «Сеансы», «Кинозалы» відриється вікно (Рис. 5.2) зі списком елементів та кнопками для редагування цього списку. Використовується одне вікно, далі «Список элементов», для редагування всіх списків, але данні можуть відображатися незалежно від типу.

При натисканні на кнопку «Изменить» або «Добавить» відкриється редактор. Якщо натиснути кнопку «Просмостреть» - відкриється такий же редактор, але без можливості зміни даних. При натисканні кнопки «Удалить» виділений елемент видаляється с БД.

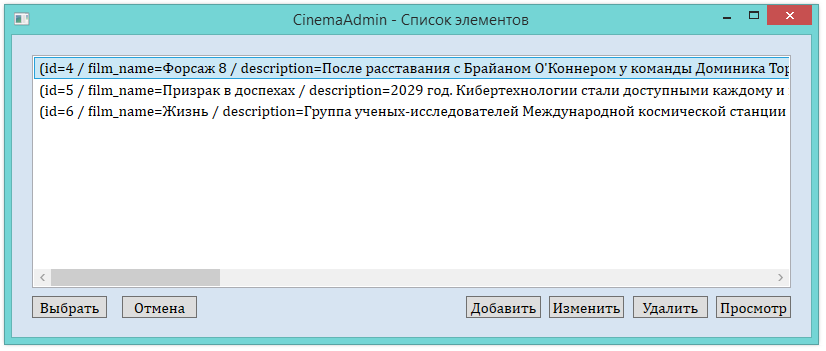


Рисунок 5.2 – Список елементів - фільмів

Редактори відповідних таблиць представлені на Рис. 5.3 – 5.5 .

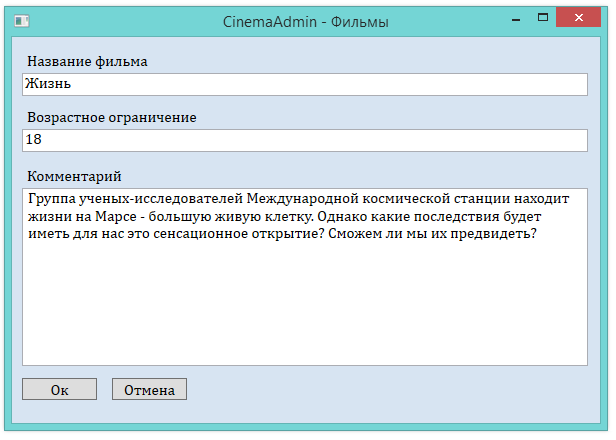


Рисунок 5.3 – Редактор фільму

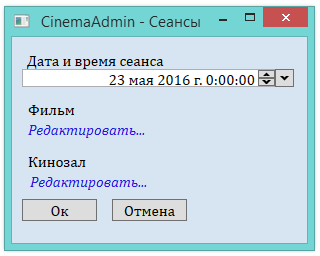


Рисунок 5.4 – Редактор сеансу

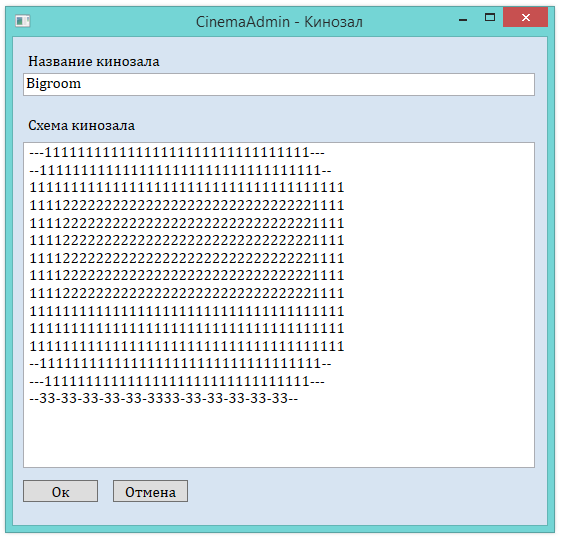


Рисунок 5.5 – Редактор кінозалу

При натисканні на кнопку «Продавать билеты» відкриється список сеансів, але він буде в режимі перегляду. При виборі елементу зі списку відкриється схема залу (Рис. 5.6), де можна продавати білети на даний сеанс. Схема залу редагується в редакторі кінозалів.

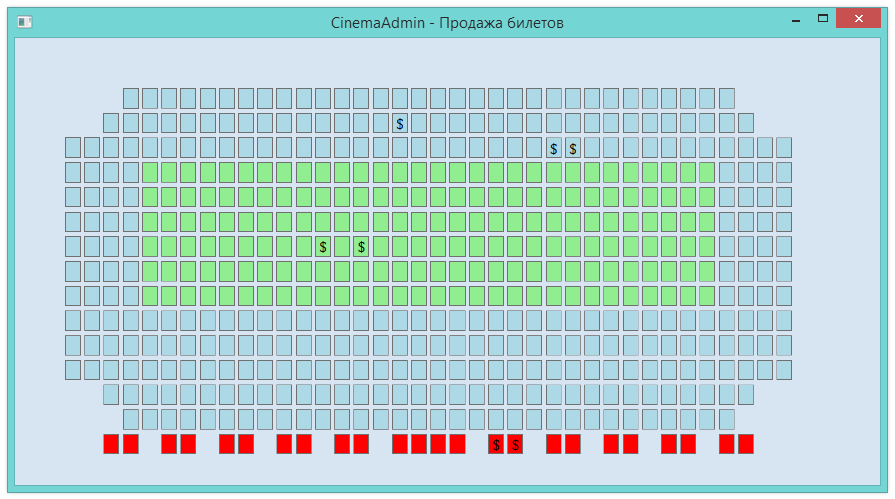


Рисунок 5.6 – Форма для продажу білетів

Кнопка «О авторе» виводить додаткову інформацію (Рис 5.7).

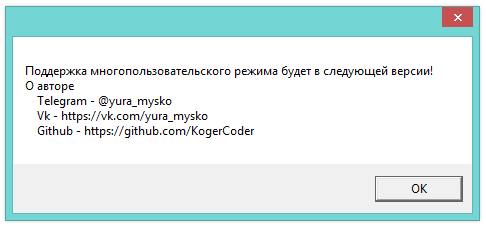


Рисунок 5.7 – Форма з інформація про автора

## Опис метаданих, що зберігаються на сервері

### Таблиці

Лістинг коду створення таблиці cinema\_rooms

CREATE TABLE cinema\_admin.cinema\_rooms (

id int(11) NOT NULL AUTO\_INCREMENT,

room\_name varchar(255) DEFAULT NULL,

room\_scheme\_object text DEFAULT NULL,

PRIMARY KEY (id)

)

ENGINE = INNODB

AUTO\_INCREMENT = 3

AVG\_ROW\_LENGTH = 16384

CHARACTER SET utf8

COLLATE utf8\_general\_ci

ROW\_FORMAT = DYNAMIC;

Лістинг коду створення таблиці films

CREATE TABLE cinema\_admin.films (

id int(11) NOT NULL AUTO\_INCREMENT,

film\_name varchar(50) DEFAULT NULL,

description text DEFAULT NULL,

age\_rate int(11) DEFAULT NULL,

PRIMARY KEY (id)

)

ENGINE = INNODB

AUTO\_INCREMENT = 7

AVG\_ROW\_LENGTH = 16384

CHARACTER SET utf8

COLLATE utf8\_general\_ci

ROW\_FORMAT = DYNAMIC;

Лістинг коду створення таблиці seance\_schedule

CREATE TABLE cinema\_admin.seance\_schedule (

id int(11) NOT NULL AUTO\_INCREMENT,

film\_id int(11) DEFAULT NULL,

room\_id int(11) DEFAULT NULL,

seance\_date datetime DEFAULT NULL,

PRIMARY KEY (id),

CONSTRAINT FK\_seance\_schedule\_cinema\_rooms\_id FOREIGN KEY (room\_id)

REFERENCES cinema\_admin.cinema\_rooms (id) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT FK\_seance\_schedule\_films\_id FOREIGN KEY (film\_id)

REFERENCES cinema\_admin.films (id) ON DELETE NO ACTION ON UPDATE NO ACTION

)

ENGINE = INNODB

AUTO\_INCREMENT = 4

AVG\_ROW\_LENGTH = 16384

CHARACTER SET utf8

COLLATE utf8\_general\_ci

ROW\_FORMAT = DYNAMIC;

Лістинг коду створення таблиці sellers

CREATE TABLE cinema\_admin.sellers (

id int(11) NOT NULL AUTO\_INCREMENT,

login varchar(255) DEFAULT NULL,

pass varchar(255) DEFAULT NULL,

name varchar(50) DEFAULT NULL,

rules\_level tinyint(4) DEFAULT NULL,

PRIMARY KEY (id)

)

ENGINE = INNODB

AUTO\_INCREMENT = 2

AVG\_ROW\_LENGTH = 16384

CHARACTER SET utf8

COLLATE utf8\_general\_ci

ROW\_FORMAT = DYNAMIC;

Лістинг коду створення таблиці sold\_tickets

CREATE TABLE cinema\_admin.sold\_tickets (

id int(11) NOT NULL AUTO\_INCREMENT,

seance\_id int(11) DEFAULT NULL,

seat\_row tinyint(4) DEFAULT NULL,

seat\_column tinyint(4) DEFAULT NULL,

sale\_date datetime DEFAULT NULL,

price int(11) DEFAULT NULL,

seller\_id int(11) DEFAULT NULL,

PRIMARY KEY (id),

CONSTRAINT FK\_sold\_tickets\_sellers\_id FOREIGN KEY (seller\_id)

REFERENCES cinema\_admin.sellers (id) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT FK\_tickets\_seance\_schedule\_id FOREIGN KEY (seance\_id)

REFERENCES cinema\_admin.seance\_schedule (id) ON DELETE NO ACTION ON UPDATE NO ACTION

)

ENGINE = INNODB

AUTO\_INCREMENT = 41

AVG\_ROW\_LENGTH = 16384

CHARACTER SET utf8

COLLATE utf8\_general\_ci

ROW\_FORMAT = DYNAMIC;

### Тригери

Тригер для збереження білетів в окремій таблиці перед видаленням. Лістинг коду тригера

BEGIN

INSERT INTO sold\_tickets\_deleted (old\_id,seance\_id,seat\_row,seat\_column,sale\_date,price,seller\_id)

VALUES (OLD.id,OLD.seance\_id,OLD.seat\_row,OLD.seat\_column,OLD.sale\_date,OLD.price,OLD.seller\_id);

END

### Збережені процедури

Збережена процедура для створення резервних копій таблиць за їх ім’ям.

Лістинг коду процедури

CREATE DEFINER = 'root'@'localhost'

PROCEDURE cinema\_admin.backupTable(IN table\_name varchar(255))

BEGIN

SET @query = CONCAT('DROP TABLE IF EXISTS `', @table\_name, '\_backup`;');

PREPARE preparedQuery FROM @query;

EXECUTE preparedQuery;

SET @query = CONCAT('CREATE TABLE `', @table\_name, '\_backup` AS (SELECT \* FROM `', @table\_name, '`);');

PREPARE preparedQuery FROM @query;

EXECUTE preparedQuery;

END

# Висновки

У даному курсовому проекті була розроблена база даних «Автоматизація робочого місця продавця білетів кінотеатру» та застосування для неї.

Етапи розробки:

1) Проведення аналізу предметної області, визначення вимог до ПЗ;

2) Проектування системи;

3) Конструювання системи.

Система була розроблена з використанням засобів СУБД MySQL та графічного редактору dbForge Studio, а також - середовища розробки клієнтських застосувань Visual Studio та засобами мови програмування C#.

Був розроблений графічний інтерфейс користувача та сервер з даними.

В результаті роботи були реалізовані наступні функції:

* Редактор сеансів;
* Редактор фільмів;
* Редактор кінозалів;
* Можливість продажу білетів використовуючи схему кінотеатру.

Не реалізовано:

* Редактор списку продавців. Але можливе використання багатьма користувачами одночасно;
* Можливість пошуку за різними критеріями.

Але система спроектована так, що додавання цих функцій не потребуватиме редагування готових частин системи. В цілому у виконаній роботі реалізовано більшу частину поставлених задачі.

# Список використаних джерел

1. Опис СУБД MySQL. [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/MySQL>
2. Статья про ER-модель. [Електронний ресурс] – Режим доступу: <https://goo.gl/twN5jH>
3. Приклад реалізації АРМ продавця білетів кінотеатру на 1С. [Електронний ресурс] – Режим доступу: <http://infostart.ru/public/201081>
4. Робота з MySQL в C#. [Електронний ресурс] – Режим доступу: <https://goo.gl/twN5jH>
5. Статья про ER-модель. [Електронний ресурс] – Режим доступу: <https://habrahabr.ru/post/256823>
6. Проектування реляційних БД. [Електронний ресурс] – Режим доступу: <https://habrahabr.ru/post/194738/>