Державний вищий навчальний заклад

«Прикарпатський національний університет імені Василя Стефаника»

Кафедра комп’ютерних наук та інформаційних систем

**Обчислювальна практика**

Виконав:

студент групи КН-2

Яворський І.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2020р

ППрийняла: доц. Семаньків М.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2020р

Індивідуальні задачі

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Варіант 8  Умова задачі | Викладач | Оцінка | Підпис |
| 1 | Створення інформаційної системи: Бібліотека | Семаньків М.В |  |  |

Підсумкова оцінка за практику \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Івано-Франківськ 2020

**Зміст**

* **Вступ**
* **розділ 1. теоретична частина**
  + **Поняття бази даних і СУБД, їх характеристика. Типи СУБД**
  + **Модель даних «сутність-зв’язок»**
  + **Проектування БД**
  + **Класифікація зв’язків за множинністю та повнотою**
  + **MySql**
* **розділ 2. Практична частина**
* **Висновки**

**Вступ**

Завданням на дану роботу є створення інформаційної системи для бібліотеки.

Для виконання цього завдання я вирішив скористатись СУБД MySQL для забезпечення бази даних та технологією .Net Windows Form (на мові програмування C#) для створення графічного інтерфейсу користувача.

**РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА**

**1. Поняття бази даних і СУБД, їх характеристика. Типи СУБД**

***База даних (БД)*** – це засіб накопичення і організації великих масивів інформації про об’єкти деякої предметної області (ПО).

БД має відображати поточні дані про предметну область, накопичувати, зберігати інформацію і надавати різним категоріям користувачів швидкий доступ до даних.

Для цього дані в базі мають бути структуровані відповідно до деякої моделі, що відображає основні об’єкти ПО, їх властивості і зв’язки між ними.

БД є частиною складної системи, яка називається банком даних або системою баз даних (СБД).

***Система управління базами даних*** – це комплекс програмних засобів, призначений для інтегрованого зберігання та обробки даних.

 Система управління базами даних — це прикладна програма, реалізована на електронній обчислювальній машині чи обчислювальному комплексі. За допомогою її можна:

* створювати структуру бази даних, вводити інформацію та зберігати її на зовнішніх носіях;
* виконувати певне коло операцій з даними;
* одержувати результати та зберігати їх на зовнішніх носіях або передавати на віддалені термінали;
* виводити інформацію на термінал у зручній для користувача формі або на друкувальні пристрої;
* давати можливість працювати з базами даних багатьом користувачам.

Прикладом бази даних є бібліотечний каталог, записна книжка, класні журнали, журнали обліку товарів на підприємствах та ін.

**Робочі характеристики БД:**

*1)*       *повнота* ***–*** чим повніша база даних, тим ймовірніше, що вона містить потрібну інформацію (проте, не має бути надмірної інформації);

*2)*       *правильна організація* ***–*** чим краще структурована база даних, тим легко в ній знайти необхідні відомості;

*3)*       *актуальність* ***–*** будь-яка база даних може бути точною і повною, якщо вона постійно оновлюється, тобто необхідно, щоб база даних у кожен момент часу повністю відповідала стану об’єкту, що відображався нею;

4)       *зручність для використання* ***–*** база даних має бути проста і зручна у використанні і мати розвинені методи доступу до будь-якої частини інформації.

***Основні функції СУБД:***

1)       *Визначення даних* (яка саме інформація зберігатиметься в базі даних, властивості даних, їх тип, як ці дані зв’язані між собою).

*2)*       *Обробка даних* (це вибір полів, фільтрація і сортування даних).

3)       *Управління даними* (кому дозволено працювати з даними, корегувати їх або додавати нову інформацію; визначати правила колективного доступу).

**Типи СУБД:**

* ієрархічна;
* мережна;
* реляційна.

***Ієрархічна модель даних***

**Ієрархічна модель**  –  це  модель  даних,  у  якій  зв’язки  між даними мають вигляд ієрархій.

В ієрархічній базі файли пов’язані між собою фізичними покажчиками або полями даних, доданих до окремих записів.

Ієрархічна БД складається з упорядкованого набору дерев; тобто, з впорядкованого набору декількох екземплярів одного типу дерева.

Тип дерева складається з одного «кореневого» типу запису і впорядкованого набору піддерев.

 Дані представляються у вигляді дерева з одним кореневим вузлом і з умовами, що кожен вузол нижче кореневого може бути пов’язаний з одним вищестоячим вузлом і з декількома нижчестоячими вузлами.

***Мережна модель даних***

**Мережна  модель**  –  це  модель,  коли  кожний  запис  може бути підпорядкований записам більше, ніж з одного файлу.

Для зв’язування даних використовують фізичні покажчики.

Мережний підхід до організації даних є розширенням ієрархічного підходу.

В ієрархічних структурах запис-нащадок повинен мати в точності одного батька; у мережній структурі даних у нащадка може бути будь-яке число батьків.

***Реляційна модель даних***

***Реляційна модель*** – це модель, в основі якої лежить математичне поняття відношення.

Назва «реляційна» (англ. relation – відношення) - кожен запис у таблиці містить інформацію, що відноситься тільки до одного конкретного об’єкту.

Відношення подається у вигляді двовимірних таблиць по колонках і рядках.

Дані пов’язані відповідно до їхніх внутрішніх логічних взаємовідносин.

Логічний підхід до даних зробив можливим створення мов запитів, наприклад, мова запитів SQL.

У реляційних системах баз даних цілі файли даних може обробляти одна команда.

**2 Модель даних «сутність-зв’язок»**

**2.1. Поняття відношення, атрибута, ключа, зв’язку**

*Модель даних* (Data Model) є логічною структурою даних, що становить притаманні цим даним властивості, незалежні від апаратного й програмного забезпечення й не пов'язані з функціонуванням комп'ютера.

***Сутність –*** будь-який об’єкт, інформацію про якого необхідно зберігати в базі даних.

**Розрізняють:**

1)       ***тип сутності*** –відноситься до набору однорідних осіб, предметів, подій або ідей, як ціле.

2)       ***екземпляр сутності*** –відноситься до конкретної речі в наборі.

Сутністю можуть бути люди, місця, літаки,…

Екземпляри сутності: рейси, смак, колір і т.д.

Наприклад, типом сутності може бути МІСТО, а екземпляром – Москва, Київ і т. д.

***Атрибут –*** поіменована характеристика сутності.

Атрибути використовуються для визначення того, яка інформація має бути зібрана про сутність.

Приклади атрибутів для  сутності АВТОМОБІЛЬ є:  ТИП, МАРКА, НОМЕРНИЙ ЗНАК, КОЛІР і т.д.

Існує також відмінність між типом і екземпляром.

Тип атрибуту КОЛІР має багато екземплярів (або значень): Червоний, Синій, Банановий, Білий і т.д.

***Ключ –*** мінімальний набір атрибутів, за значеннями яких можна однозначно знайти необхідний екземпляр  сутності.

Наприклад, для сутності Розклад літаків ключем є атрибут: «Номер рейсу», він містить: «Пункт відправлення», «Час вильоту» і «Пункт призначення».

***Зв’язок –*** асоціювання двох або більше за сутність.

Одна з основних вимог до організації бази даних – це забезпечення можливості відшукання однієї  сутності по значеннях інших, для чого необхідно встановити між ними певні зв’язки.

**3. Проектування БД**

*Проектування БД* - це пошук засобів задоволення функціональних вимог засобами наявної комп'ютерної технології з урахуванням заданих обмежень.

Як правило, ІТ-проекти із створення БД містять у собі такі етапи:

1. Визначення стратегії побудови системи.

2. Аналіз вимог до БД.

3. Проектування БД.

4. Реалізація БД.

5. Тестування.

6. Впровадження БД.

Етап проектування БД вважається одним із

найскладніших етапів створення БД, який не має явно вираженого початку й закінчення. У порівнянні з аналізом вимог до БД або розробленням додатків, проектування БД, на думку багатьох провідних фахівців, є невдало структурованим завданням. Якщо всі етапи створення БД перекриваються один з одним у своїй послідовності, то етап проектування перекривається з усіма іншими етапами. Проектування починається з моменту прийняття стратегічних рішень і триває на етапах реалізації й тестування.

Процес проектування БД охоплює кілька основних сфер:

* проектування об'єктів БД (таблиці, подання, індекси, тригери, збережені процедури, функції, пакети) для подання даних ПО в БД;
* проектування інтерфейсу взаємодії з БД (форми, звіти й т.д.), тобто проектування додатків, які будуть супроводжувати дані в БД і реалізовувати питально- відповідні відношення на цих даних;
* проектування БД під конкретне обчислювальне середовище або інформаційну технологію (архітектура "клієнт-сервер", паралельні архітектури, розподілене обчислювальне середовище);
* проектування БД під призначення системи (інтелектуальний аналіз даних, OLAP, OLTP і т.д.).

**4. Класифікація зв’язків за множинністю та повнотою**

***один-до-одного* (1:1)**: у кожен момент часу кожному представникові (екземпляру)  сутності А відповідає 1 або 0 представників  сутності В:

Наприклад: студент може не мати стипендії, отримати звичайну або одну з підвищених стипендій.

***один-до-багатьох* (1:Б)**: одному представникові  сутності А відповідають 0, 1 або декілька представників  сутності В:

Наприклад: квартира може бути порожньою, в ній може жити один або декілька мешканців.

***багато-до-одного* (Б:1)**;

***багато-до-багатьох* (Б:Б)**.

Наприклад: Якщо зв’язок між сутністю ЧОЛОВІКА і ЖІНКИ називається ШЛЮБ, то існує чотири можливі варіанти такого зв’язку:

***безліч зв’язків між однією і тією ж сутністю:***

Наприклад: пацієнт, маючи одного лікаря, може мати також декілька лікарів-консультантів; лікар може бути лікарем декількох пацієнтів, лікар може одночасно консультувати інших пацієнтів.

***тренарні зв’язки:***

Наприклад: лікар може призначити декілька пацієнтів на декілька аналізів, аналіз може бути призначений декількома лікарями декільком пацієнтам і пацієнт може бути призначений на декілька аналізів декількома лікарями.

**5. MySql**

**MySQL** — вільна система керування реляційними базами даних.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система управління базами даних (СУБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних.

**Ліцензування**

MySQL має подвійне ліцензування. MySQL може розповсюджуватися відповідно до умов ліцензії GPL. Але за умовами GPL, якщо якась програма використовує бібліотеки MySQL, то вона теж повинна розповсюджуватися за ліцензією GPL. Проте це може розходитися з планами розробників, які не бажають відкривати сирцеві тексти своїх програм. Для таких випадків передбачена комерційна ліцензія компанії Oracle, яка також забезпечує якісну сервісну підтримку. В разі використання та розповсюдження програмного забезпечення з іншими вільними ліцензіями, такими як BSD, Apache, MIT та інші, MySQL дозволяє використання бібліотек MySQL за ліцензією GPL.

**Історія**

MySQL виникла як спроба застосувати mSQL до власних розробок компанії: таблиць, для яких використовувалися ISAM — підпрограми низького рівня для індексного доступу до даних. У результаті був вироблений новий SQL-інтерфейс, але API-інтерфейс залишився в спадок від mSQL. Звідки походить назва «MySQL» — достеменно не відомо. Розробники дають два варіанти: або тому, що практично всі напрацювання компанії починалися з префікса My, або на честь дівчинки на ім'я My, дочки Майкла Монті Віденіуса, одного з розробників системи.

Логотип MySQL у вигляді дельфіна носить ім'я «Sakila». Він був обраний з великого списку запропонованих користувачами «імен дельфіна». Ім'я «Sakila» було відправлено Open Source-розробником Ambrose Twebaze. В січні-лютому 2008 Sun Microsystems придбала розробника системи керування базами даних MySQL за $1 млрд. Після поглинання у 2009 році Sun Microsystems компанією Oracle Corporation MySQL стала власністю Oracle.

За час розвитку під орудою Oracle дедалі більше відокремлює MySQL від спільноти і робить процес розробки все менш прозорим. Наприклад, повернута практика поставки власницьких розширених функцій в Enterprise-версії MySQL, спостерігається приховування інформації про вразливості, зі складу виключений тестовий набір, закритий доступ до більшої частини системи відстеження помилок та припинено публікація згрупованого логу змін, що дозволяє судити про прив'язку патчів до конкретних змін.

**РОЗДІЛ 2. ПРАКТИЧНА ЧАСТИНА**

**Виконання роботи(СУБД - MySql) :**

1. Кожна книга має назву, автора, видавнитво, тип, жанр.

2. Під кожен тип може підходити декілька книг.

3. Під кожен жанр може підходити декілька книг.

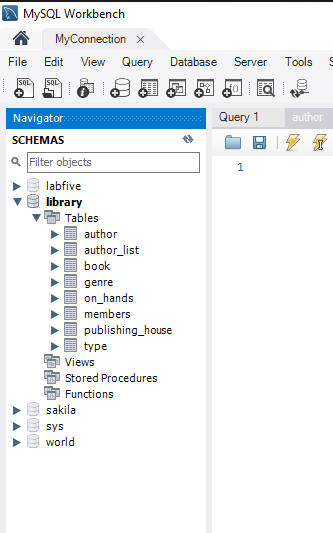
4. Кожне видавництво має назву і адресу,воно може видати безліч книг.

5. Кожен автор має ім’я, прізвище, батьківщину, рік народження і може написати безліч книг.

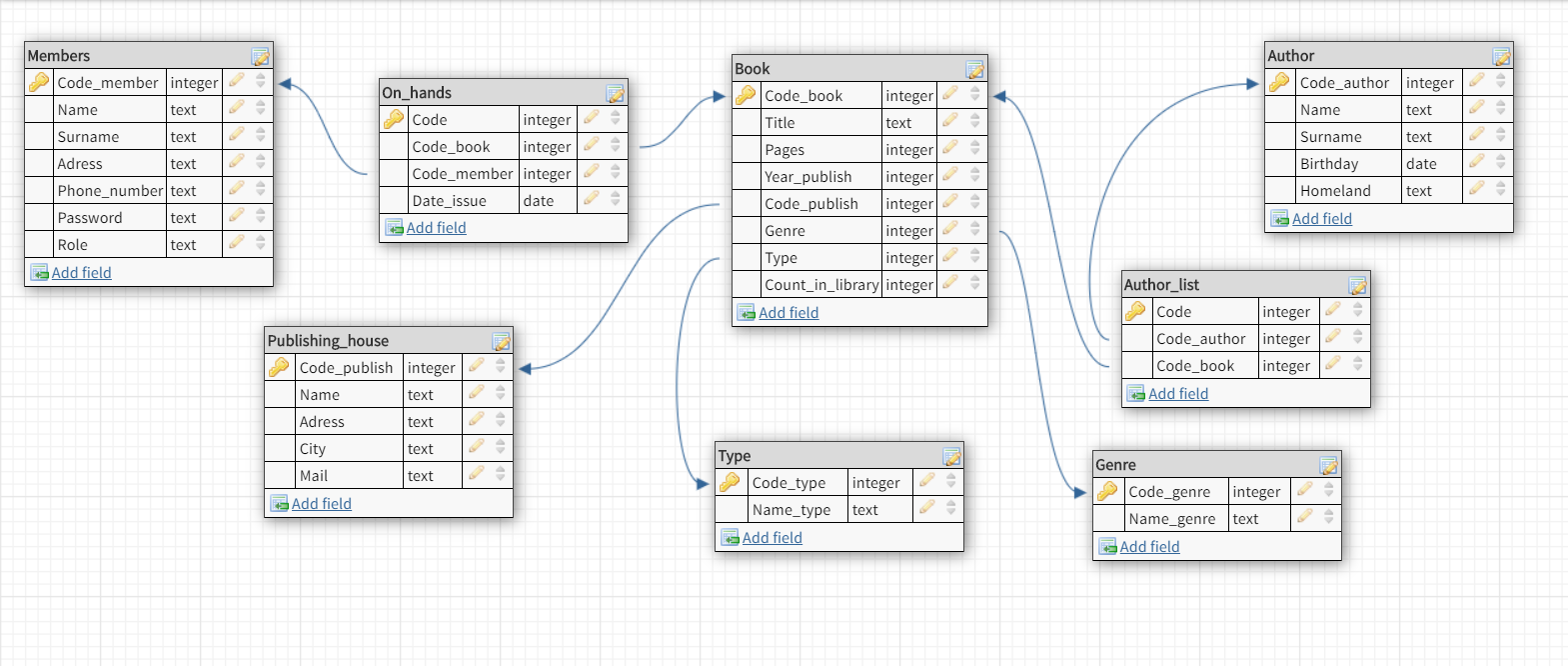
6. Кожен користувач бібліотеки має ім’я, прізвище, адресу і може взяти наявну книгу.

7. Дату взяття книги потрібно запам’ятати.

Базу даних було створено в MySQL Workbench.

****

ER діаграма бази даних.



1. Таблиця **Type –** використовуєтся для зберігання типів книг.

Поля:

* **Code\_type –** первинний ключ.
* **Name\_type –** назва типукниги.

2. Таблиця **Genre -** використовуєтся для зберігання жанрів книг.

Поля:

* **Code\_genre –** первинний ключ.
* **Name\_genre –** назва жанру.

3. Таблиця **Publising\_house -** використовуєтся для зберігання інформації про видавництва книг.

Поля:

* **Code\_publish –** первинний ключ.
* **Name –** ім’я видавнитва.
* **Adress –** адреса видавництва.
* **City –** місто в якому знаходиться видавнитво.
* **Mail –** пошта видавництва.

4. Таблиця **Author -** використовуєтся для зберігання інформації про авторів книг.

Поля:

* **Code\_author –** первинний ключ.
* **Name** – ім’я автора.
* **Surname** – прізвище автора.
* **Birthday –** день народження автора.
* **Homeland** – батьківщина автора.

5. Таблиця **Author\_list -** використовуєтся для зберігання пар значень(код автора, код книги).

Поля:

* **Code –** первинний ключ.
* **Code\_author –** зовнішній ключ (унікальний номер автора).
* **Code\_book –** зовнішній ключ (унікальний номер книги).

6. Таблиця **On\_hands -** використовуєтся для зберігання інформації про видачу книг на руки користувачам.

Поля:

* **Code –** первинний ключ.
* **Code\_book –** зовнішній ключ (унікальний номер книги).
* **Code\_member -** зовнішній ключ (унікальний номер користувача бібліотеки).
* **Date\_issue –** дата видачі книги користувачеві.

7. Таблиця **Members -** використовуєтся для зберігання інформаії про користувачів бібліотеки.

Поля:

* **Code\_member–** первинний ключ.
* **Name –** ім’я користувача.
* **Surname –** прізвище користувача.
* **Adress** – адреса користувача.
* **Phone\_number –** телефонний номер користувача.
* **Password –** пароль користувача.
* **Role –** права користувача (“Admin”,”User”).

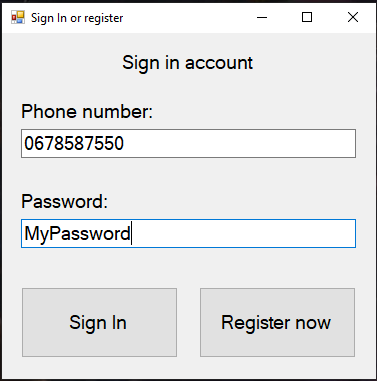
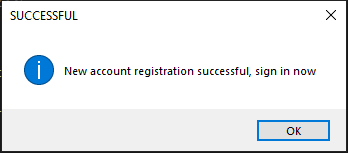
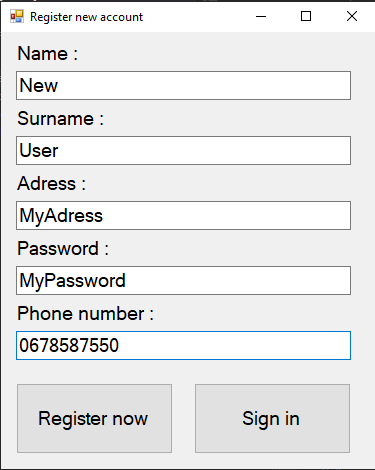
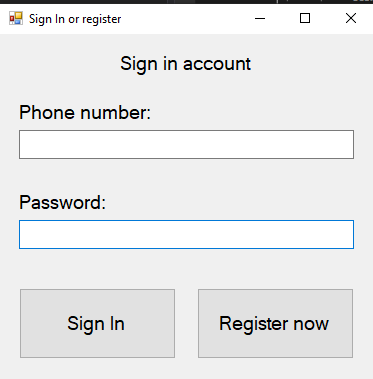
8. Таблиця **Book -** використовуєтся для зберігання інформації про книги.

Поля:

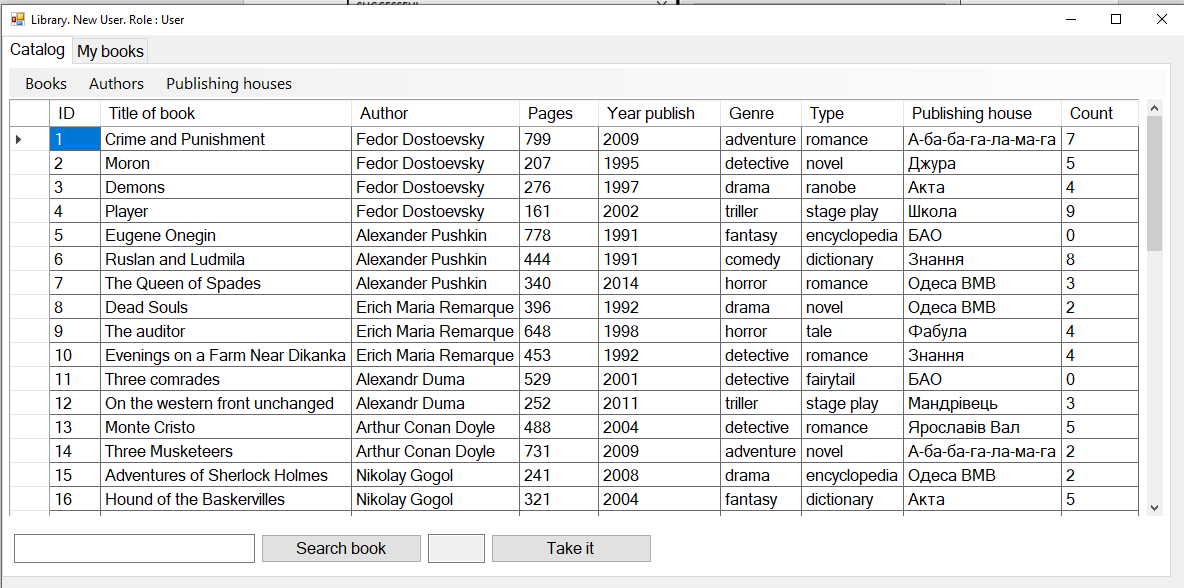
* **Code\_book –** первинний ключ.
* **Title –** назва книги.
* **Pages –** кількість сторінок книги.
* **Year\_publish –** рік печаті.
* **Code\_publish –** зовнішній ключ (унікальний номер видавнитва).
* **Genre –** зовнішній ключ (унікальний номер жанру книги).
* **Type –** зовнішній ключ (унікальний номер типу книги).
* **Count in library –** клькість книг в бібліотеці.

**Хід виконання і робота з програмою:**

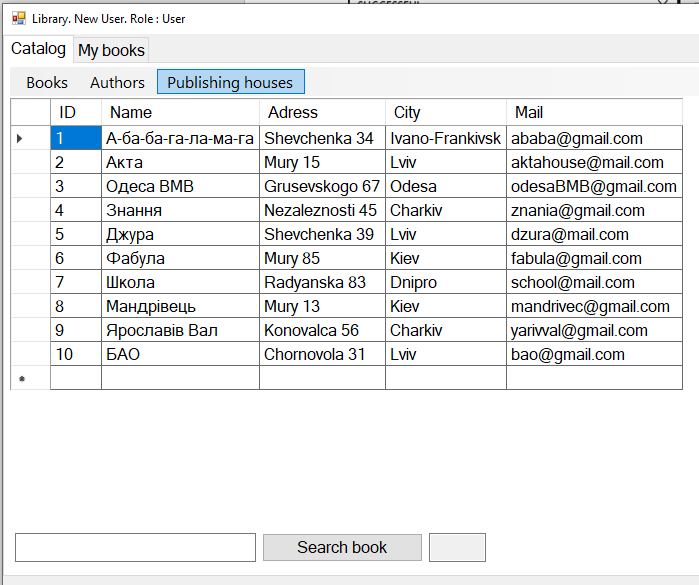
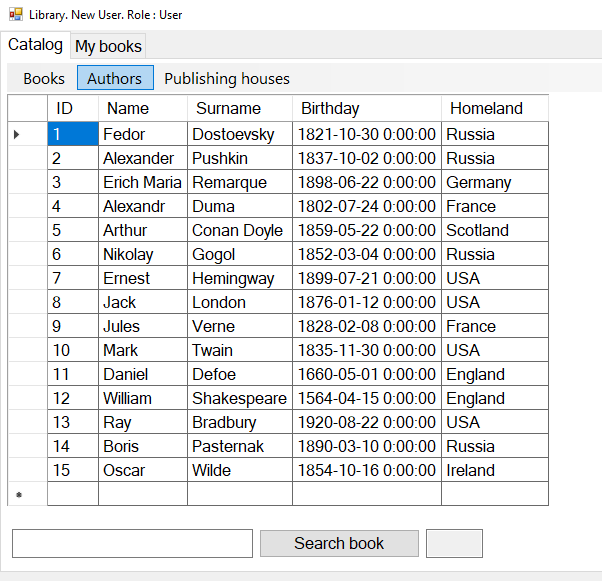
При запуску програми випливає вікно в якому ви можете авторизуватися або прейти до вікна реєстрації. Зареєструємо новий аккаунт і увійдемо в систему з його допомогою.



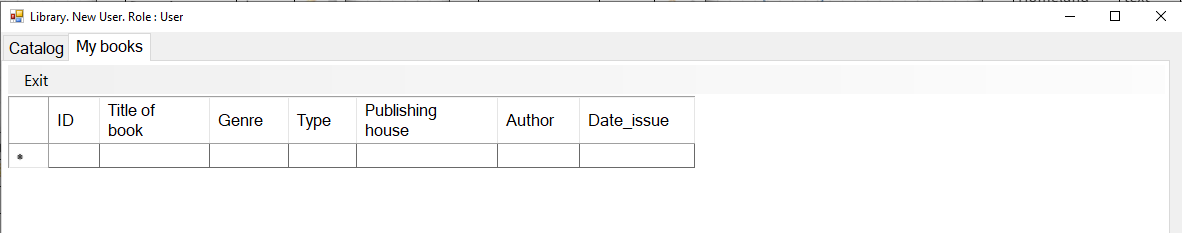
Після авторизації ми бачимо каталог книг і різні кноки для взаємодії з програмою. В назві вікна бачимо наше ім’я та роль в бібліотеці.



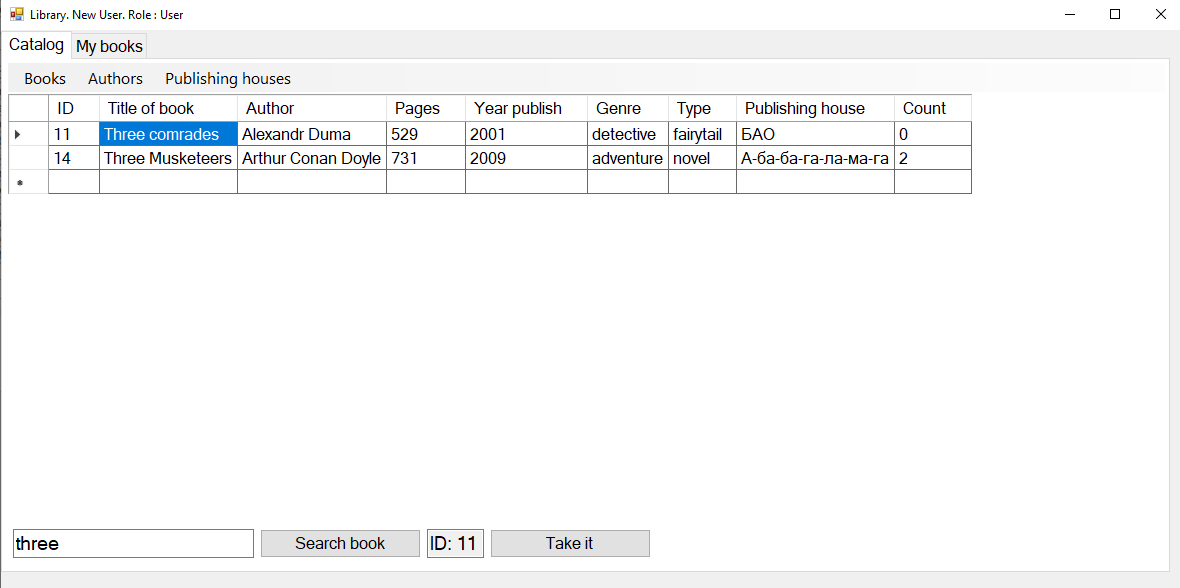
Якщо натиснути на кнопку “Authors” можна побачити інформаію про всіх авторів книги яких є в доступі. А кнопка “Publishing house” відображає інформацію про видавництва.



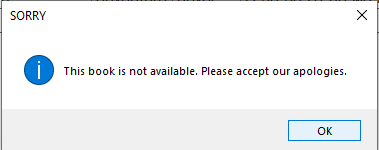
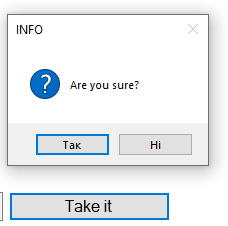
Перейшовши на вкладку “My books” ми можемо побачити перелік книг які були позичені в бібліотеці користувачем цього аккаунту.



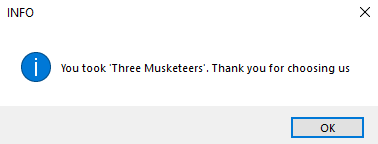
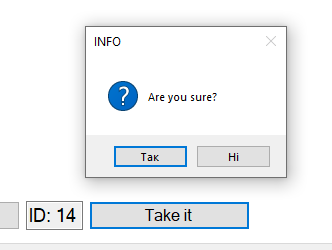
Скориставшись пошуком і кпопкою “Search book” ми можемо шукати книги в нашому каталозі. Потім клікнувши по будь-якому полю цієї книги вибираємо її і кнопкою “Take it” маємо можливість позичити її собі.



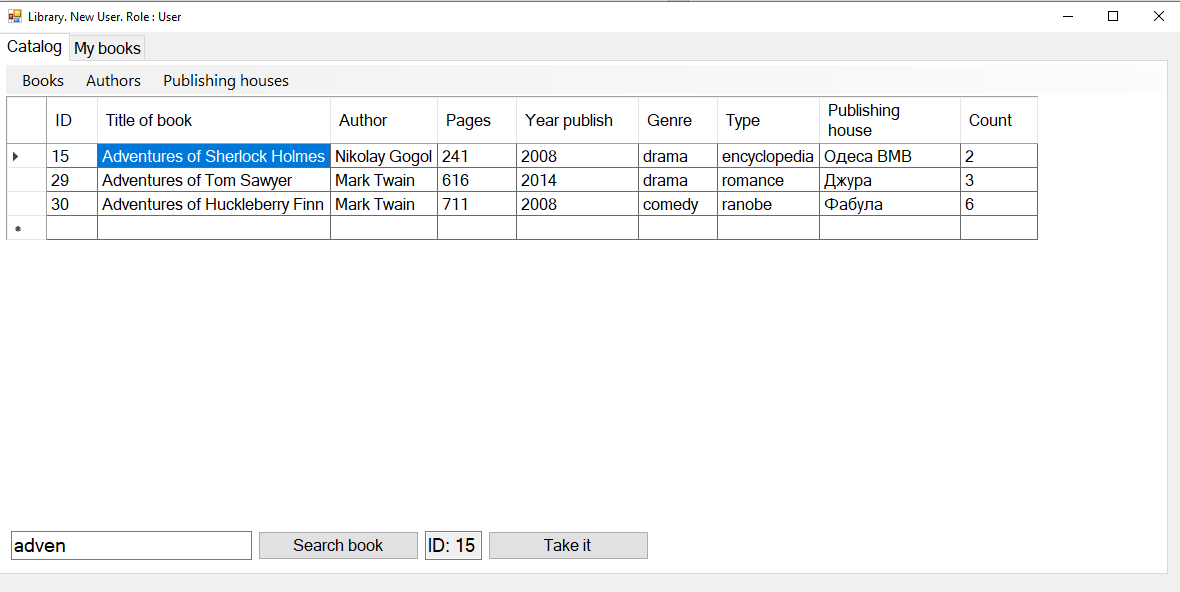
Якщо значення поля “Count” цієї книги = 0 тоді з’являється повідомлення, що книг немає в наявності. В нашому випадку книги “Three comrades” не було тому:

****

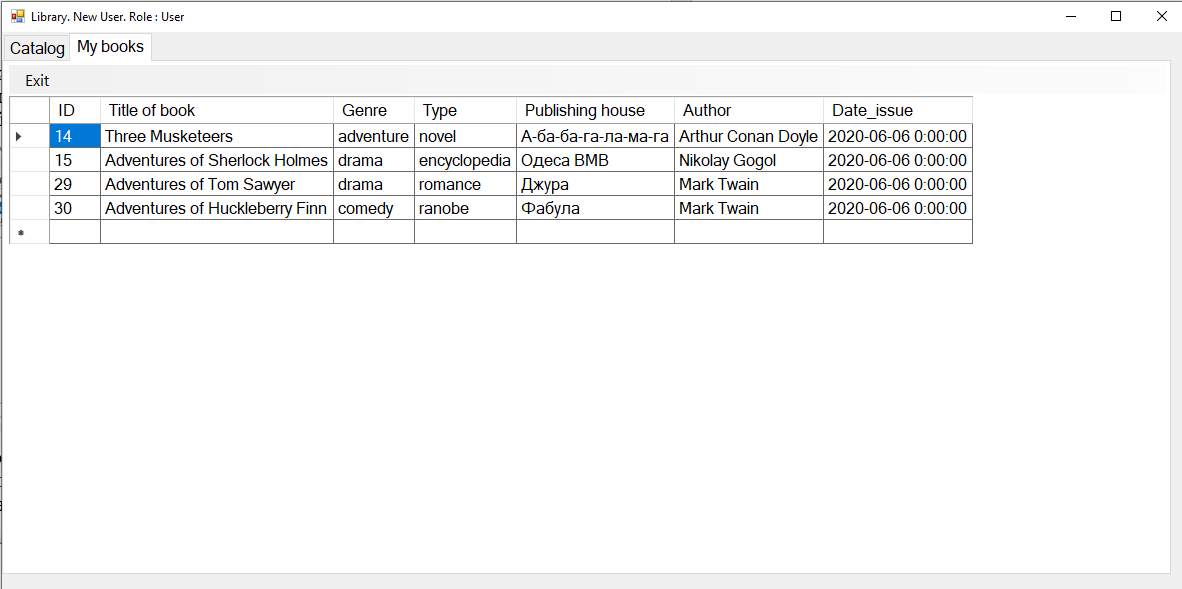
Якщо ж книга є в наявності (як наприклад “Three Musketeers”) то:

****

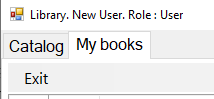
Візьмемо ще пару книг.



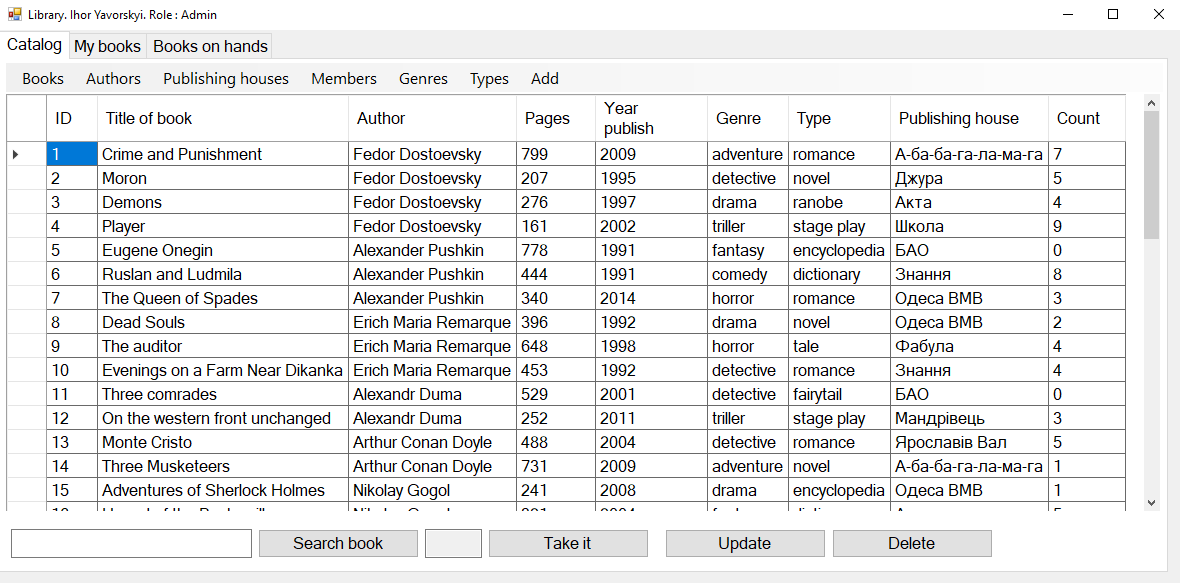
Тоді переглянемо вкладку “My books”. Бачимо що позичені книги успішно відображаюются в таблиці.



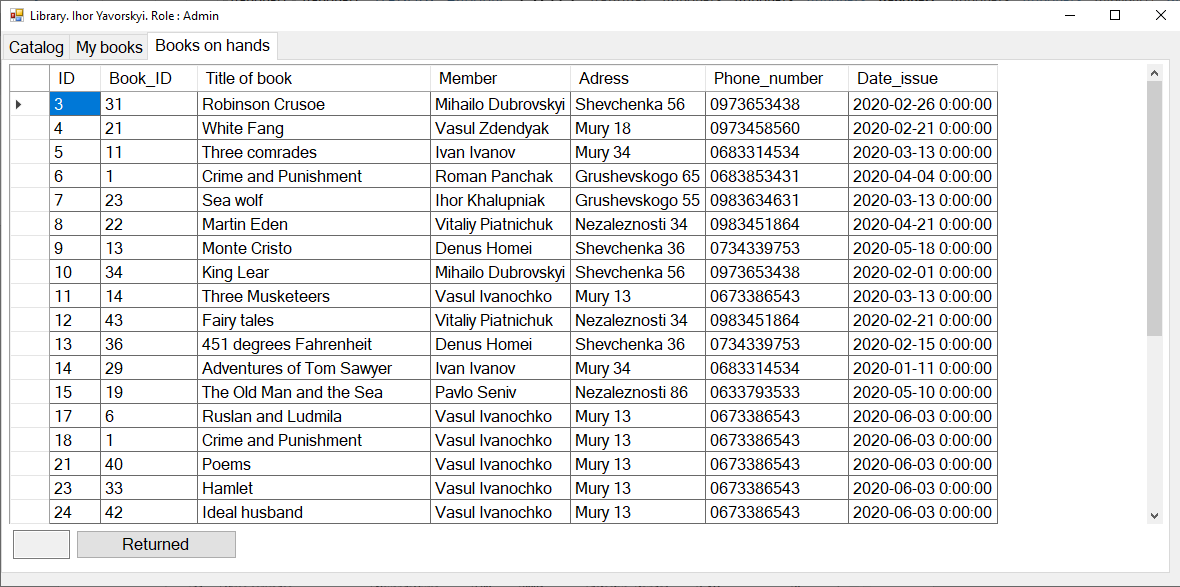
Натиснувши на кнопку “Exit” ми можемо вийти з нашого аккаунту і нас перенаправить зному на форму авторизації.

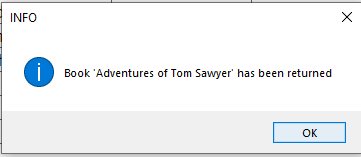
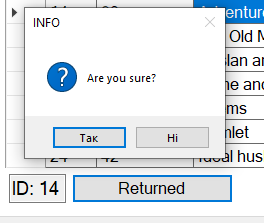
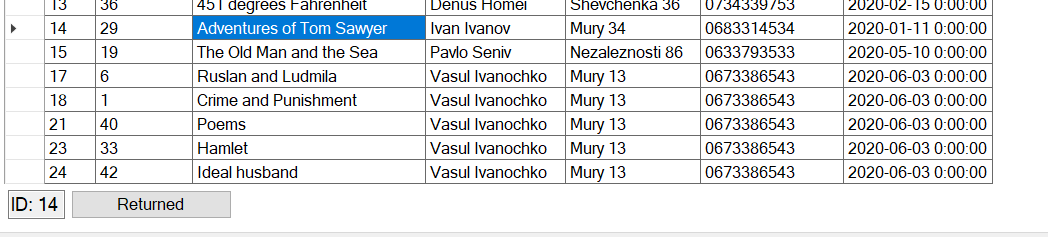


Авторизуємось з аккаунту адмінстратора. Бачимо нові пунтки в меню, кнопки “update” та “delete” на формі, а також нову вкладку “Books on hands”. В назві форми бачимо ім’я та роль “Admin”.

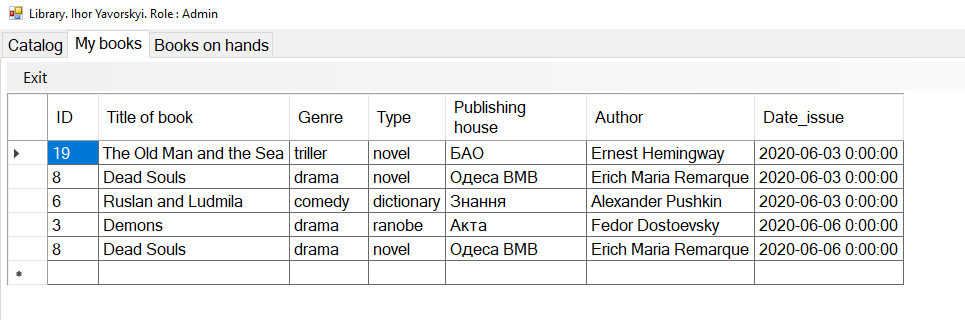


Переглянемо вміст вкладки “Books on hands”. Тут адмін має можливість побачити книги які були позичені користувачами, інформацію про них і дату позики.

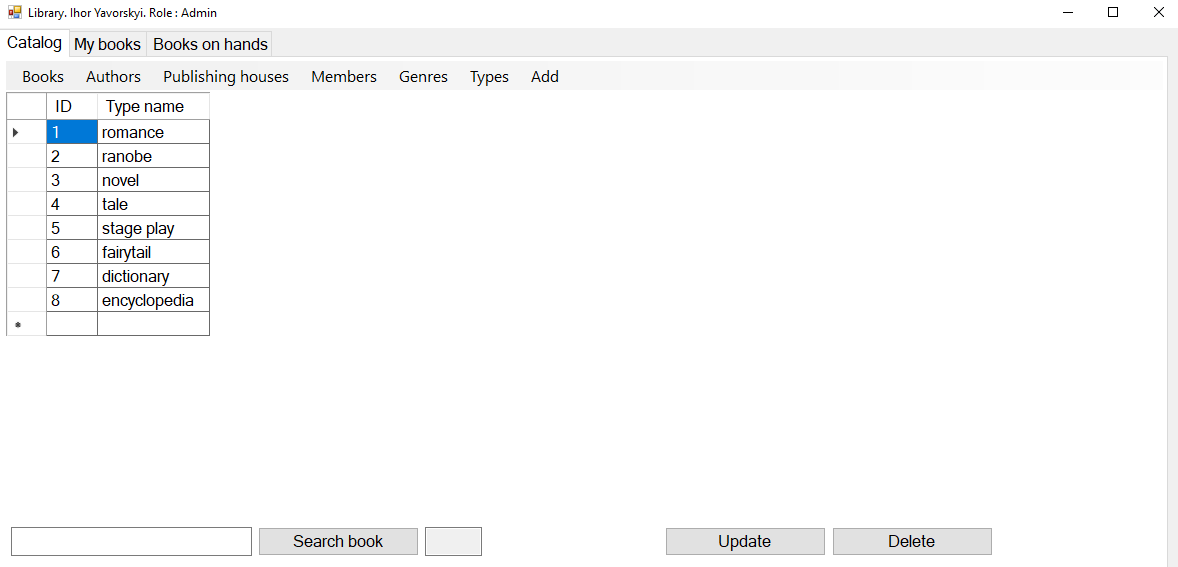
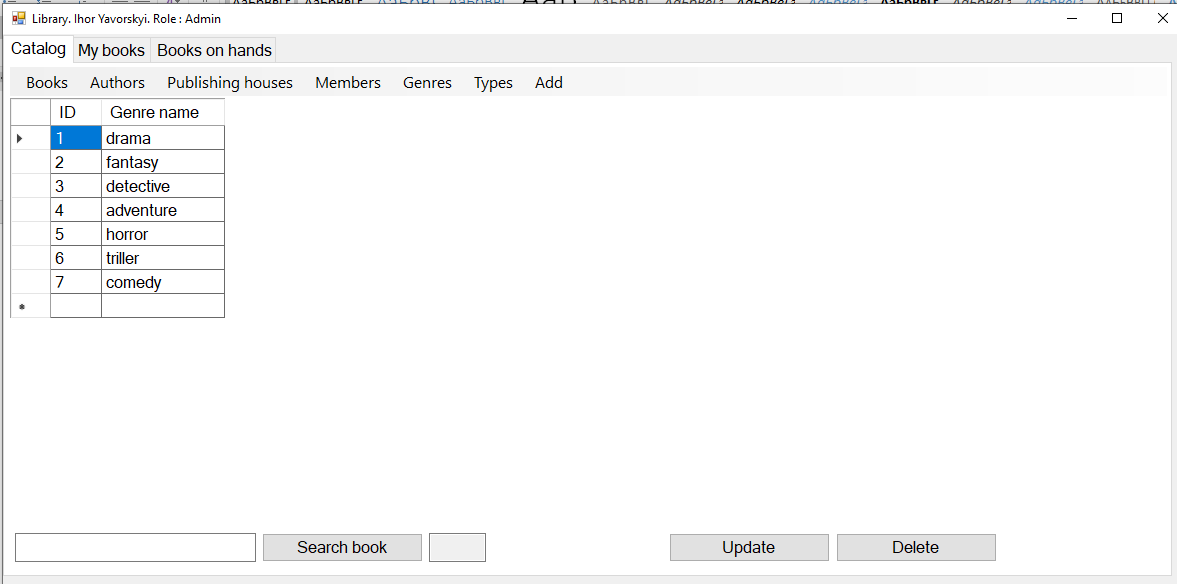
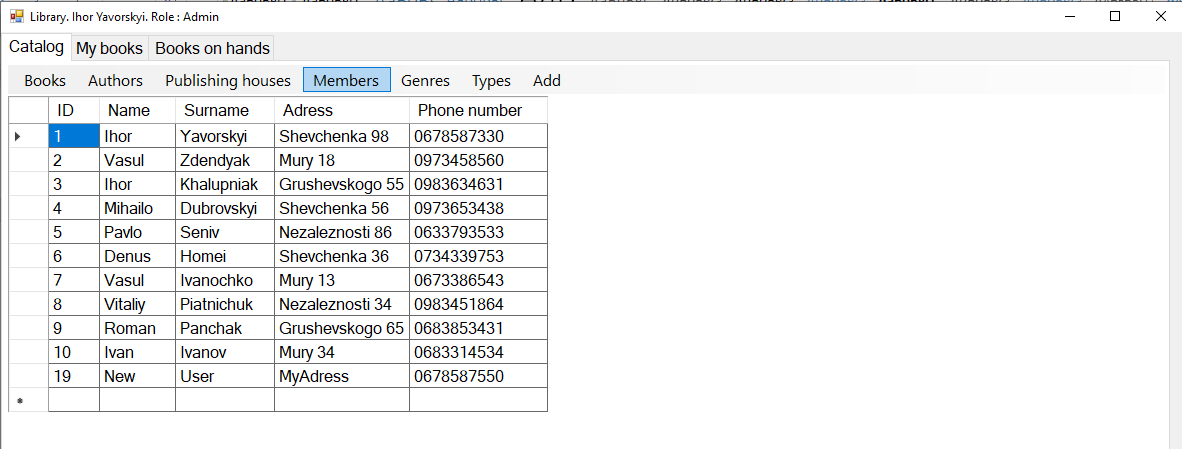


Також вибравши конкретний запис в таблиці адмін може позначити, що дана книга була повернена до бібліотеки даним корстувачем.

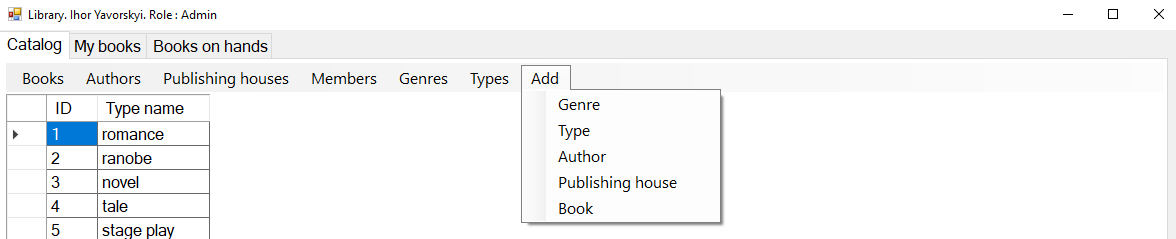
Книги які наш адмін брав в бібліотеці:



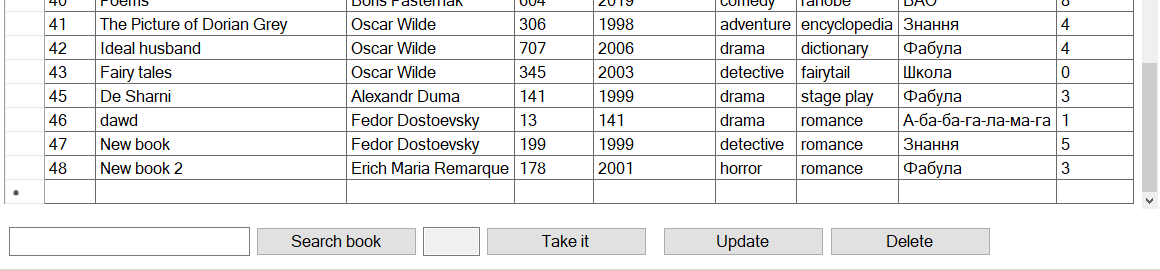
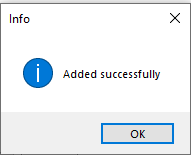
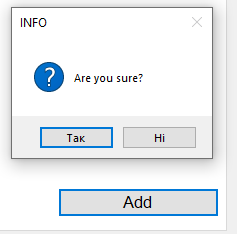
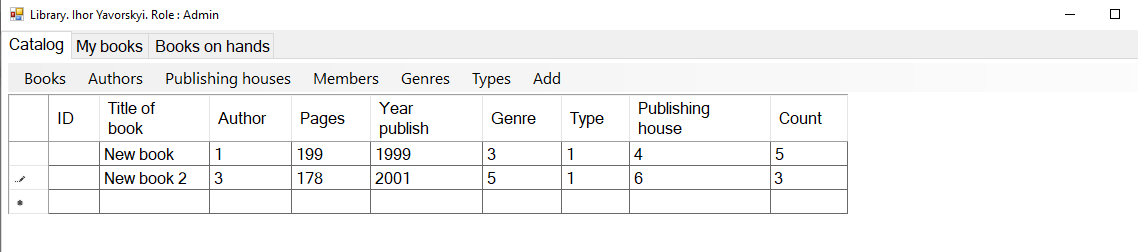
Додаткові таблиі які бачить адмін:

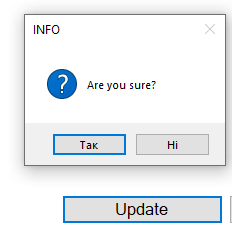


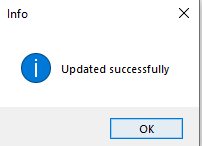
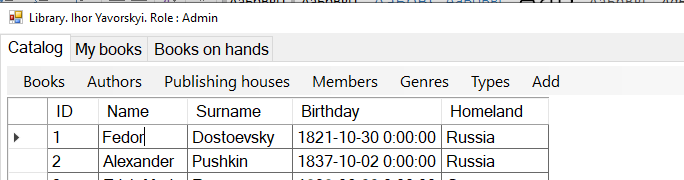
Якщо в меню навести на “Add” можемо вибрати, що нового ми хочемо додати в бібліотеку.

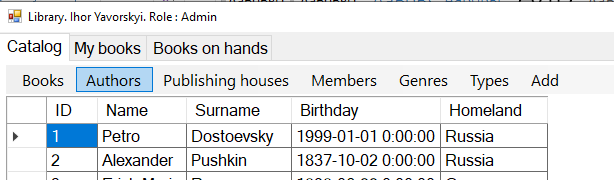


Додамо дві нові книги

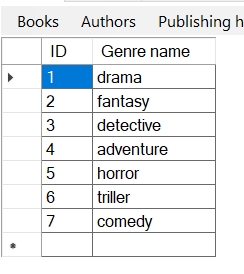
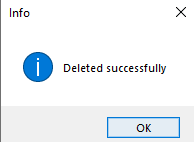
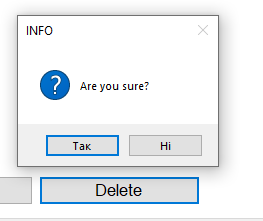
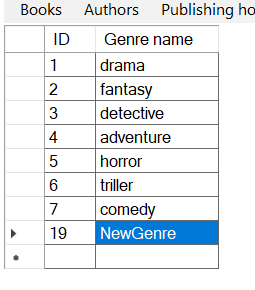


Перейдемо в авторів і нехай змінимо комусь ім’я і дату народження. 





Нехай видалимо якийсь жанр.



**ВИСНОВКИ**

Завданням на дану роботу було створення інформаційної системи для бібліотеки.

В ході виконання завдання було розроблено проект бази даних бібліотеки, в який ввійшло вісім таблиць. Для виконання цього завдання я вирішив скористатись СУБД MySQL для забезпечення бази даних та технологією .Net Windows Form для створення графічного інтерфейсу користувача.

Було передбачено ролі для користувачів. Користувач з роллю “Admin” має можливості обновляти, видаляти та додавати дані до всіх таблиць. Він керує бібіліотекою.