

Звіт

по 1 лабораторній роботі
з програмування

Виконав студент групи К-24
Теодор Романус

Електронний архів факультету

Первинна постановка задачі:

Таблиці містять таку інформацію: П.І.П. автора, назва матеріалу, факультет (департамент, відділення), кафедра, вид матеріалу, обсяг, дата створення та ін.

Розробити інформаційну систему за допомогою технології ADO.NET з DataSet, яка б включала підсистему введення нової інформації та підсистему одержання довідок та пошуку інформації (окремо динаміка змін).

Задачі, які реалізовувались:

1. Створення бази даних.
2. Візуалізація даних у вигляді однієї таблиці.
3. Можливість фільтрувати дані за факультетом, кафедрою, категорією. Пошук.
4. Додавання автора.
5. Додавання публікації.

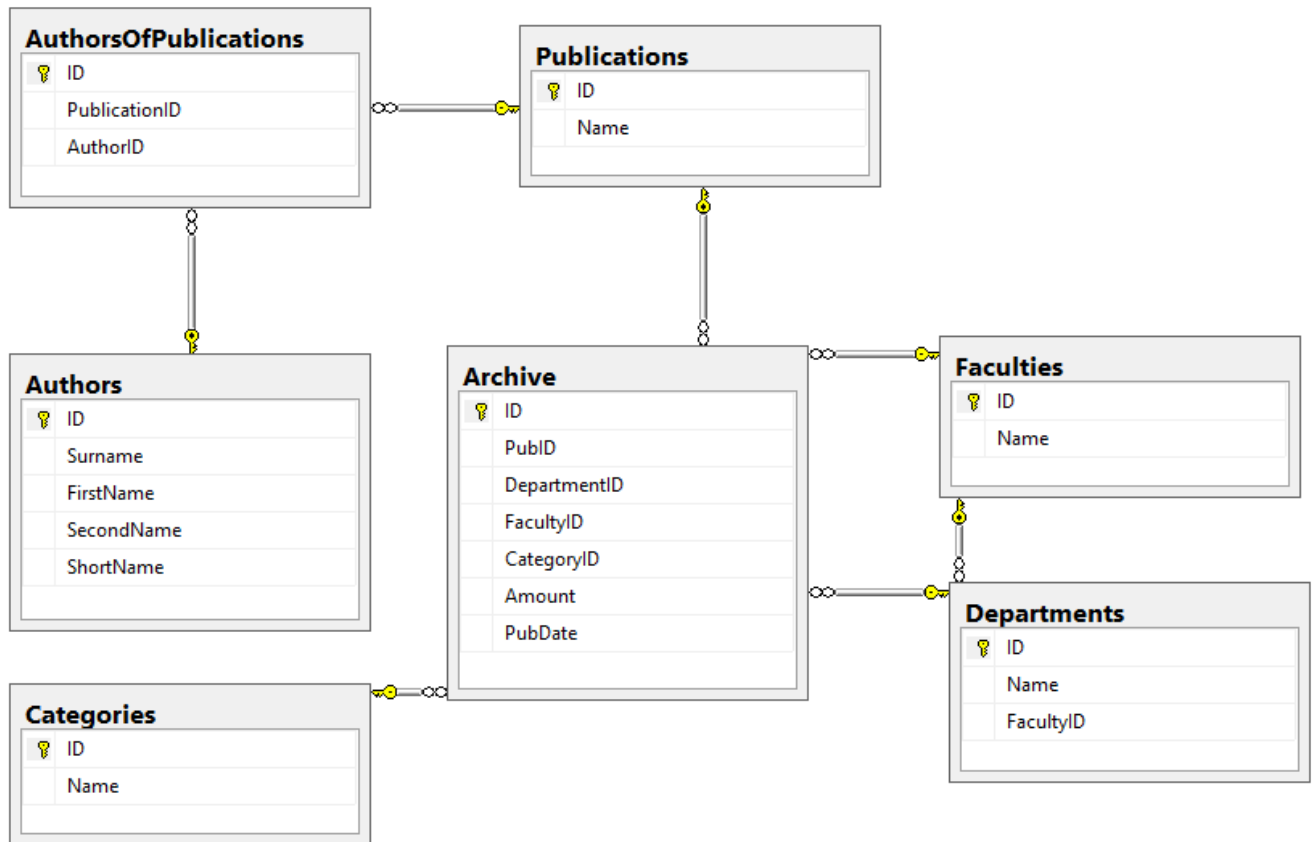
Увесь код лабораторної можна знайти [тут](#).

Тепер детальніше про кожен з етапів.

1. Створення бази даних.

При створення бази даних використовувався шаблон Database First. СКБД для створення бази – Microsoft SQL Server 2014. Створення відбувалось написанням і виконанням [SQL-запиту](#).

Вигляд БД після створення:



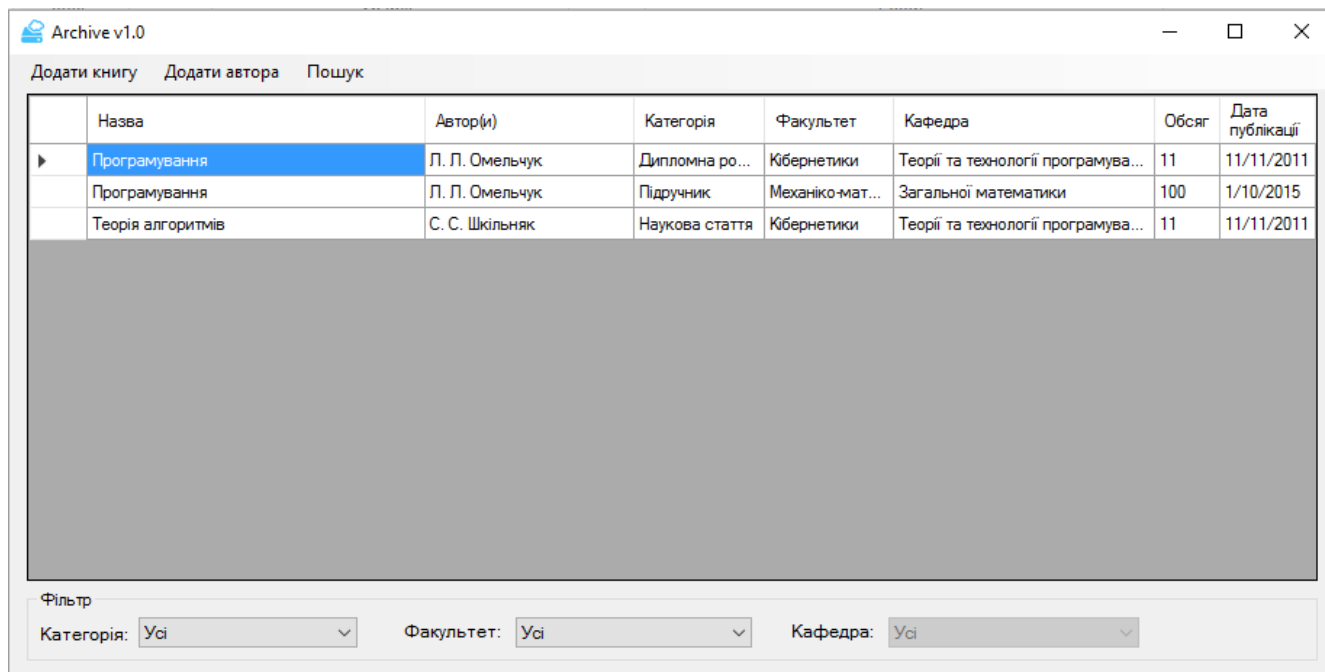
Деякі таблиці (Publications) і зв'язки (Archive -> Faculties) були додані лише з метою навчитися працювати з великою кількістю таблиць. В наступних лабораторних БД була оптимізована.

2. Візуалізація даних у вигляді однієї таблиці.

Для відображення даних у вигляді однієї таблиці у xsd-модель було додано нову DataTable. По суті, це аналог View, тільки на програмному рівні, а не на рівні БД.

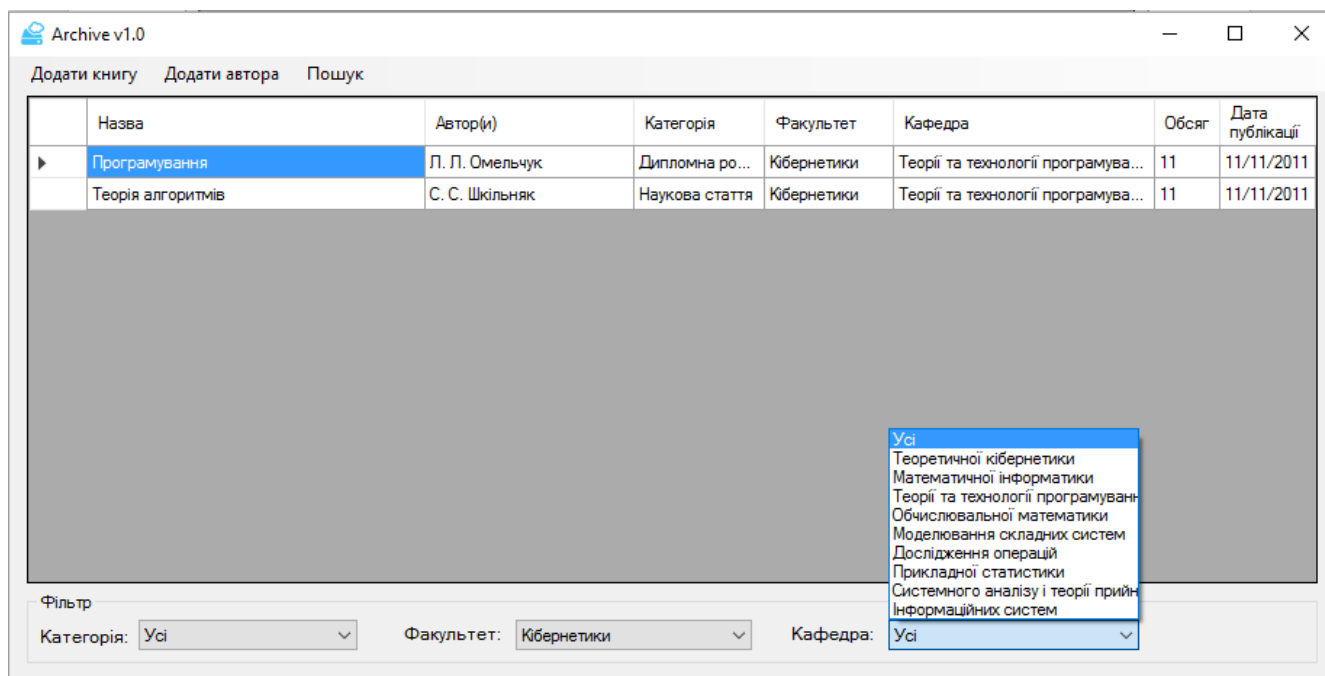
Дані виводились у DataGridView.

Результат виконання:



3. Можливість фільтрувати дані за факультетом, кафедрою, категорією. Пошук.

Для зручності роботи додане фільтрування по категоріях, факультетах і кафедрах в режимі онлайн (після вибору поля з випадаючого списку дані негайно фільтруються):



Пошук, по суті, - це розширена фільтрація. Для пошуку виділена окрема форма, в якій зчитуються усі дані:

Результати пошуку виводяться в окремому вікні.

Для виведення даних, фільтрування і пошуку застосовувався один запит, який був прив'язаний до DataTable, про яку ми вже згадували раніше:

```
SELECT p.Name [Name], (SELECT TOP 1 f.Authors FROM AuthorsOfPub(p.ID) f) AS [Author(s)], c.Name
[Category], f.Name [Faculty], d.Name [Department], a.Amount [Amount], a.PubDate [PublicationDate]
FROM Archive a
JOIN Categories c ON c.ID=a.CategoryID
JOIN Faculties f ON f.ID=a.FacultyID
JOIN Departments d ON d.ID=a.DepartmentID
JOIN Publications p ON p.ID=a.PubID
WHERE p.Name LIKE @nameQuery AND (SELECT TOP 1 f.Authors FROM AuthorsOfPub(p.ID) f) LIKE
@authorQuery AND (@facultyID=0 OR @facultyID=f.ID) AND (@departmentID=0 OR
@departmentID=a.DepartmentID) AND (@categoryID=0 OR @categoryID=c.ID) AND (@publicationYear = " OR
@publicationYear=YEAR(a.PubDate))
ORDER BY p.Name
```

Для виведення усіх даних, наприклад, цей запит виконувався з пустими параметрами. Для фільтрації по факультетах заповнений був тільки параметр facultyID, решта всі – порожні (нульові).

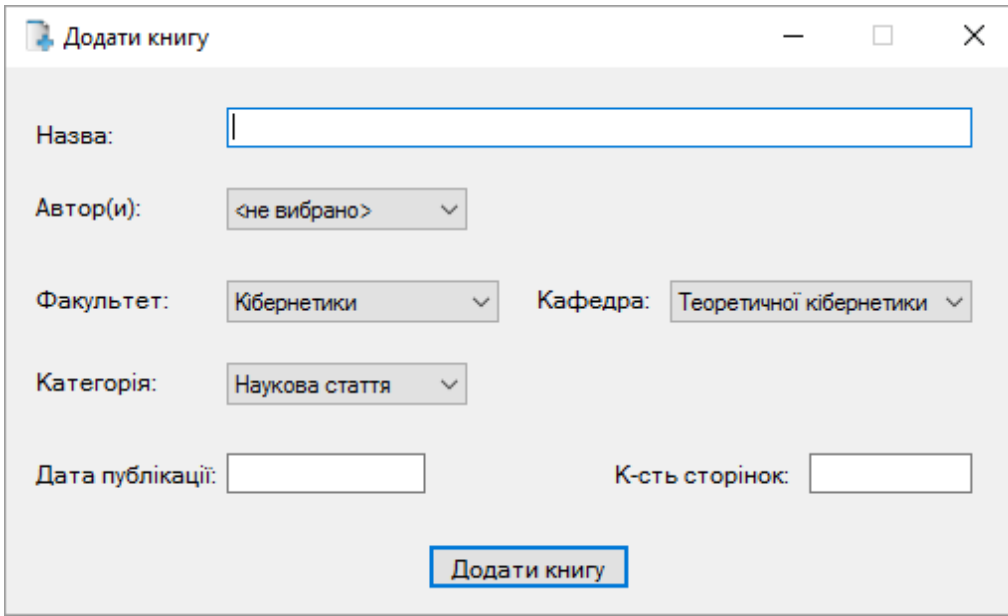
4. Додавання автора.

Для додавання автора виділено окреме вікно:

Після натискання на кнопку перевірялась наявність даного автора в базі, в разі відсутності за допомогою стандартного INSERT-запиту дані про автора додавались в БД.

5. Додавання публікації.

Для додавання публікації виділено окреме вікно:



Список авторів підвантажується з БД у випадаючий список. Функціонал дозволяє вибрати до 3 авторів. Усі поля повинні бути заповнені і коректні. Після натискання на кнопку «Додати книгу» відбувається перевірка наявності книги в БД, у разі її відсутності книга додається:

```
//inserting data into database
this.publicationsTableAdapter.Insert(name);
int name_id = (int)this.queriesTableAdapter.SQIDOfName(name);
foreach (var author_id in authors_id)
    this.authorsOfPublicationsTableAdapter.Insert(name_id, author_id);
this.archiveTableAdapter.Insert(name_id, department_id, faculty_id, category_id, amount, date);

MessageBox.Show("Книга успішно додана!", "Успіх");
```