

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

## **Лабораторна робота №2**

з дисципліни

«Організація баз даних та знань»

**Виконав:**

студент групи КН-208

Матвіїв Микола

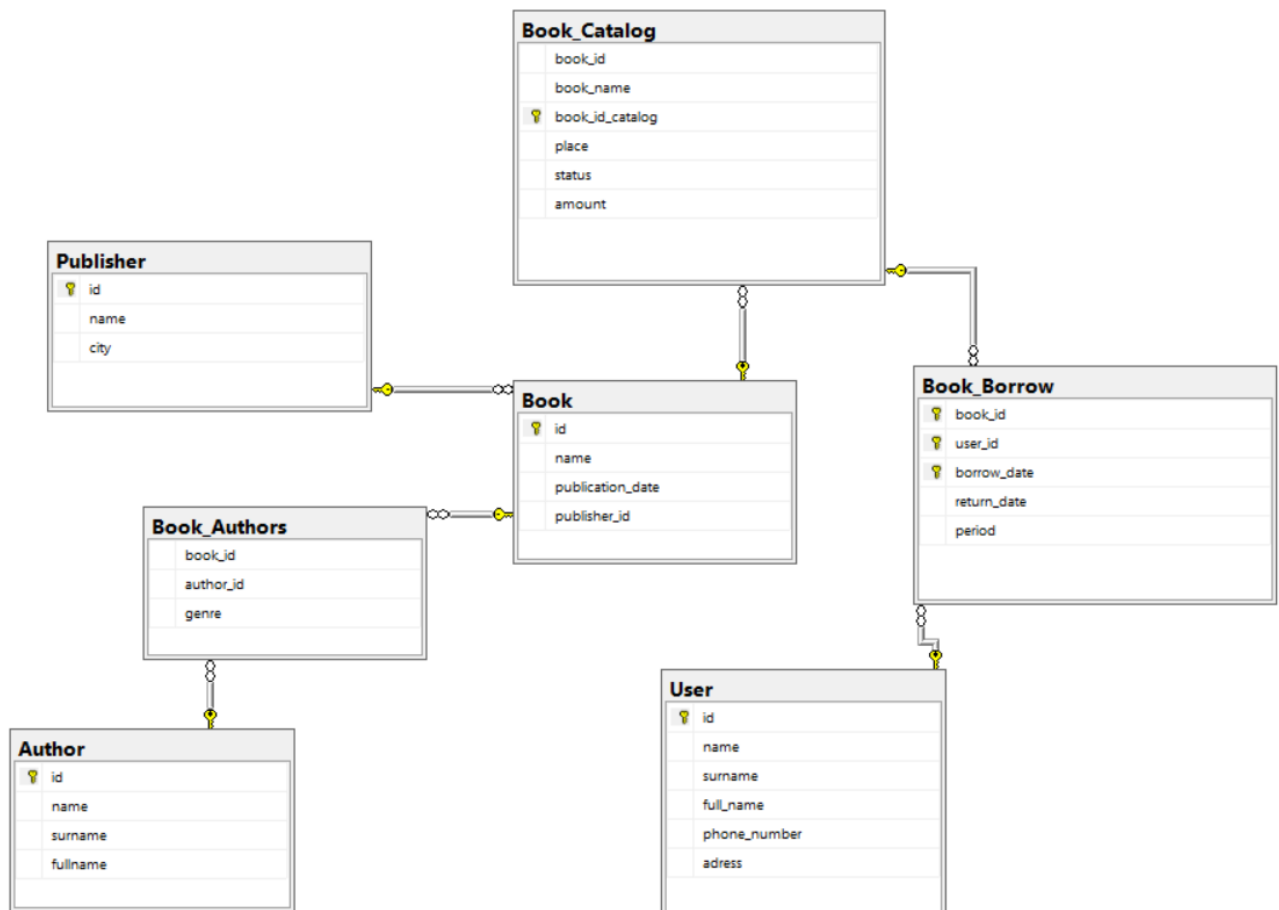
**Викладач:**

Мельникова Н. І.

Львів – 2019 р.

**Мета роботи:** Побудувати даталогічну модель бази даних; визначити типи, розмірності та обмеження полів; визначити обмеження таблиць; розробити SQL запити для створення спроектованих таблиць.

На основі діаграми побудувати Базу Даних.



Отже почнемо

Створюємо базу даних

```
CREATE DATABASE Library_test;
```

Далі створюємо таблицю Publisher,

присвоюємо атрибуту 'id' Первинний ключ, автоінкрементацію та not null

```
create table Publisher(  
    id int not null identity(1,1),  
    name nvarchar(50),  
    city nvarchar(50),  
    PRIMARY KEY (id)  
);
```

Після того створюємо таблицю Book,

присвоюємо атрибуту 'id' Первинний ключ, автоінкрементацію та not null

Також робимо зв'язок між Book та Publisher

В таблиці Book publisher\_id присвоюємо Foreign Key та з'єднуємо з id в таблиці Publisher

```
create table Book(  
    id int not null IDENTITY(1,1),  
    name nvarchar(50) not null,  
    publication_date date,  
    publisher_id int,  
    primary key(id),  
  
    CONSTRAINT FK_BOOK_PUBLISHER FOREIGN KEY (publisher_id)  
    REFERENCES Publisher (id) ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

Наступною таблицею створюємо Author,

присвоюємо атрибуту 'id' Первинний ключ, автоінкрементацію та not null

```
create table Author(  
    id int not null identity(1,1),  
    name nvarchar(50),  
    surname nvarchar(50),  
    fullname nvarchar(100)  
  
    primary key(id)  
);
```

Далі створюємо таблицю Book\_Authors для зв'язку ManyToMany між таблицями Books та Authors

Присвоюємо book\_id Foreign Key та з'єднуємо з 'id' в таблиці Book,

Також присвоюємо Foreign Key author\_id та з'єднуємо з 'id' в таблиці Author

```
create table Book_Authors(  
    book_id int,  
    author_id int,  
    genre nvarchar(50),  
  
    constraint FK_Book_Authors_Book foreign key(book_id)  
    references Book(id) ON DELETE NO ACTION ON UPDATE NO ACTION,  
    constraint FK_Book_Authors_Author foreign key(author_id)  
    references Author(id) ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

Далі створюємо таблицю Book\_Catalog

присвоюємо атрибуту 'book\_id\_catalog' Первинний ключ та not null

Атрибуту 'book\_id' присвоюємо Foreign Key та з'єднуємо з 'id' з таблиці Book

Також йому присвоюємо UNIQUE

```
create table Book_Catalog(  
    book_id int,  
    book_name nvarchar(50),  
    book_id_catalog int not null,  
    place nvarchar(50),  
    status bit,  
    amount int,  
    primary key(book_id_catalog),  
    constraint FK_Book_Catalog_Book foreign key (book_id)  
    references Book (id) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  
);
```

Наступною таблицею буде User,

присвоюємо атрибуту 'id' Первинний ключ, автоінкрементацію та not null

```
create table [User](  
    id int not null identity(1,1),  
    name nvarchar(50),  
    surname nvarchar(50),  
    full_name nvarchar(50),  
    phone_number int,  
    adress nvarchar(50),  
    primary key (id)  
);
```

І останньою таблицею буде Book\_Borrow

Робимо комплексний ключ (book\_id,user\_id,borrow\_date)

Тут а нас також два Foreign Key:

'book\_id' з таблиці Borrow\_Book з'єднуємо з 'book\_id' з Book\_Catalog

А 'user\_id' з'єднуємо з 'id' з таблиці User

```
create table Book_Borrow(  
    book_id int,  
    user_id int,  
    borrow_date date,  
    return_date date,  
    period int,  
    primary key(book_id,user_id,borrow_date),  
  
    constraint FK_Book_Borrow_User foreign key (user_id)  
    references [User] (id) ON DELETE NO ACTION ON UPDATE NO ACTION,  
    constraint FK_Borrow_Book_Book_Catalog foreign key(book_id)  
    references Book_Catalog (book_id_catalog) ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

Код запису:

```
IF EXISTS(SELECT * FROM sys.databases where name = 'Library_test')
DROP DATABASE Library_test;

CREATE DATABASE Library_test;

use Library_test;

create table Publisher(
    id int not null identity(1,1),
    name nvarchar(50),
    city nvarchar(50),
    PRIMARY KEY (id)
);

create table Book(
    id int not null IDENTITY(1,1),
    name nvarchar(50) not null,
    publication_date date,
    publisher_id int,
    primary key(id),

    CONSTRAINT FK_BOOK_PUBLISHER FOREIGN KEY (publisher_id)
    REFERENCES Publisher (id) ON DELETE NO ACTION ON UPDATE NO ACTION
);

create table Author(
    id int not null identity(1,1),
    name nvarchar(50),
    surname nvarchar(50),
    fullname nvarchar(100)

    primary key(id)
);

create table Book_Authors(
    book_id int,
    author_id int,
    genre nvarchar(50),

    constraint FK_Book_Authors_Book foreign key(book_id)
    references Book(id) ON DELETE NO ACTION ON UPDATE NO ACTION,
    constraint FK_Book_Authors_Author foreign key(author_id)
    references Author(id) ON DELETE NO ACTION ON UPDATE NO ACTION
);

create table Book_Catalog(
    book_id int,
    book_name nvarchar(50),
    book_id_catalog int not null,
    place nvarchar(50),
    status bit,
    amount int,
    primary key(book_id_catalog),
    constraint FK_Book_Catalog_Book foreign key (book_id)
    references Book (id) ON DELETE NO ACTION ON UPDATE NO ACTION,
);

create table [User](
    id int not null identity(1,1),
    name nvarchar(50),
    surname nvarchar(50),
```

```

        full_name nvarchar(50),
        phone_number int,
        adress nvarchar(50),
        primary key (id)
    );

create table Book_Borrow(
    book_id int,
    user_id int,
    borrow_date date,
    return_date date,
    period int,
    primary key(book_id,user_id,borrow_date),

    constraint FK_Book_Borrow_User foreign key (user_id)
    references [User] (id) ON DELETE NO ACTION ON UPDATE NO ACTION,
    constraint FK_Borrow_Book_Book_Catalog foreign key(book_id)
    references Book_Catalog (book_id_catalog) ON DELETE NO ACTION ON UPDATE NO ACTION
)

```

## Висновок

На цій лабораторній роботі я завершив моделювання і засобами SQL створив базу даних, що складається з семи таблиць.