**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №2**

з дисципліни

«Організація баз даних та знань»

**Виконав:**

студент групи КН-208

Матвіїв Микола

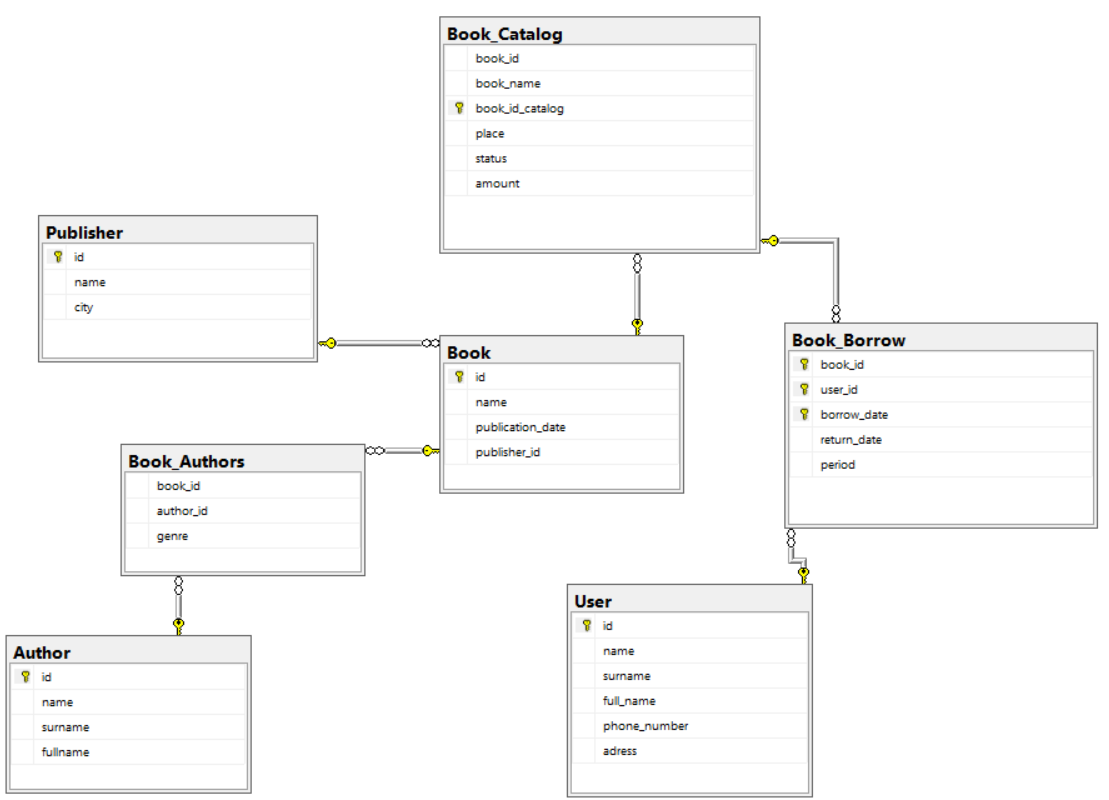
**Викладач:**

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.



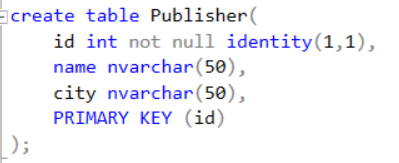
Отже почнемо

Створюємо базу даних



Далі створюємо таблицю Publisher,

присвоюємо атрибуту ‘id’ Первинний ключ, автоінкрементацію та not null

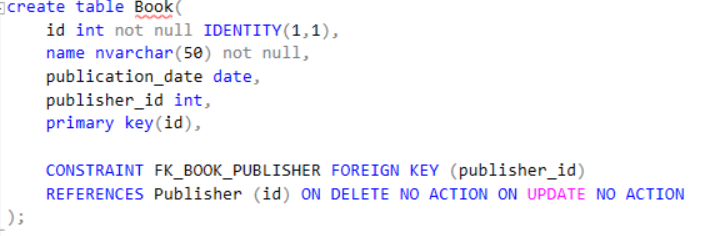


Після того створюємо таблицю Book,

присвоюємо атрибуту ‘id’ Первинний ключ, автоінкрементацію та not null

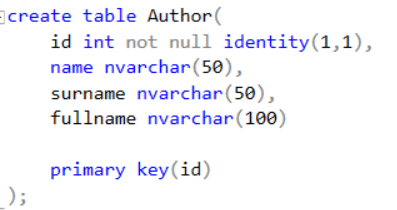
Також робимо зв’язок між Book та Publisher

В таблиці Book publisher\_id присвоюємо Foreing Key та з’єднуємо з id в таблиці Publisher



Наступною таблицею створюємо Author,

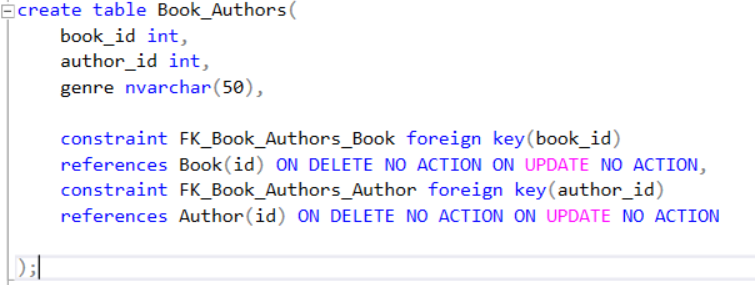
присвоюємо атрибуту ‘id’ Первинний ключ, автоінкрементацію та not null



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

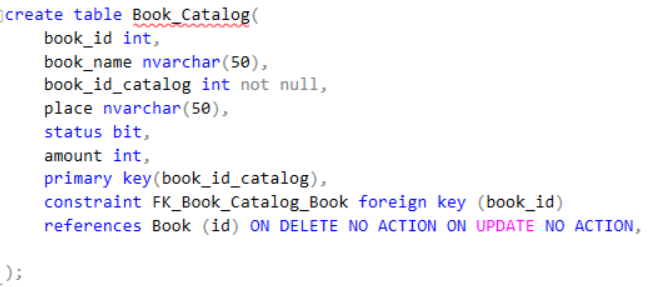


Далі створюємо таблицю Book\_Catalog

присвоюємо атрибуту ‘book\_id\_catalog’ Первинний ключ та not null

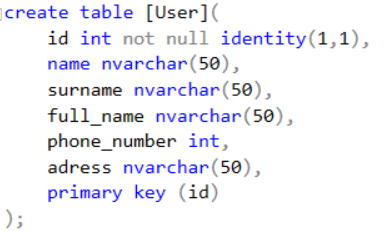
Атрибуту ‘book\_id’ присвоюємо Foreign Key та з’єднуємо з ‘id’ з таблиці Book

Також йому присвоюємо UNIQUE



Наступною таблицею буде User,

присвоюємо атрибуту ‘id’ Первинний ключ, автоінкрементацію та not null



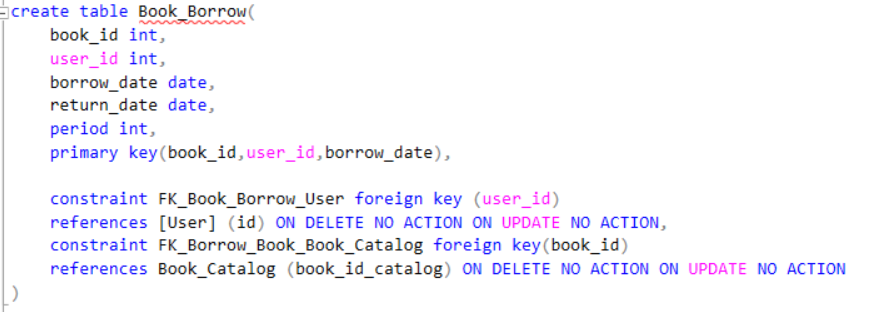
І останньою таблицею буде Book\_Borrow

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’ зєднуємо з ‘id’ з таблиці User



Код запиту:

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.