

**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота №2**

з дисципліни  
«Бази даних»

Виконала:

студентка групи ІМ-44  
Козій Олександра Ігорівна  
номер у списку групи: 7

Перевірив:

Русінов Володимир Володимирович

Київ 2025

## 1. SQL-скрипт

```
DROP TABLE IF EXISTS "BookProductOwner" CASCADE;
DROP TABLE IF EXISTS "BookReview" CASCADE;
DROP TABLE IF EXISTS "OrderDetails" CASCADE;
DROP TABLE IF EXISTS "Inventory" CASCADE;
DROP TABLE IF EXISTS "Order" CASCADE;
DROP TABLE IF EXISTS "Book" CASCADE;
DROP TABLE IF EXISTS "Review" CASCADE;
DROP TABLE IF EXISTS "ProductOwner" CASCADE;
DROP TABLE IF EXISTS "Book Store" CASCADE;
DROP TABLE IF EXISTS "Publisher" CASCADE;
DROP TABLE IF EXISTS "User" CASCADE;
```

```
CREATE TABLE "User" (
    "UserID" SERIAL PRIMARY KEY,
    "Username" VARCHAR(100) NOT NULL UNIQUE,
    "Phone" VARCHAR(20) UNIQUE,
    "Email" VARCHAR(255) NOT NULL UNIQUE,
    "PasswordHash" VARCHAR(255) NOT NULL,
    "DeliveryAddress" TEXT
);
```

```
CREATE TABLE "Publisher" (
    "PublisherID" SERIAL PRIMARY KEY,
    "Name" VARCHAR(255) NOT NULL,
    "Address" TEXT
);
```

```
CREATE TABLE "Book Store" (
    "BookStoreID" SERIAL PRIMARY KEY,
    "Name" VARCHAR(255) NOT NULL,
    "Address" TEXT NOT NULL
);
```

```
CREATE TABLE "ProductOwner" (
    "ProductOwnerID" SERIAL PRIMARY KEY,
    "Name" VARCHAR(255) NOT NULL,
    "Email" VARCHAR(255) UNIQUE
);
```

```
CREATE TABLE "Review" (
    "ReviewNumber" SERIAL PRIMARY KEY,
    "Rating" INTEGER NOT NULL CHECK (
        "Rating" >= 1
        AND "Rating" <= 5
    )
);
```

```
),  
"Comment" TEXT,  
"Date" TIMESTAMP WITH TIME ZONE DEFAULT  
CURRENT_TIMESTAMP NOT NULL  
);
```

```
CREATE TABLE "Book" (  
    "BookID" SERIAL PRIMARY KEY,  
    "ISBN" VARCHAR(13) NOT NULL UNIQUE,  
    "Name" VARCHAR(255) NOT NULL,  
    "NumberOfPages" INTEGER CHECK ("NumberOfPages" > 0),  
    "Price" NUMERIC(10, 2) NOT NULL CHECK ("Price" >= 0),  
    "PublisherID" INTEGER NOT NULL,  
    CONSTRAINT "FK_Book_Publisher" FOREIGN KEY ("PublisherID")  
    REFERENCES "Publisher" ("PublisherID") ON DELETE RESTRICT  
);
```

```
CREATE TABLE "Order" (  
    "OrderNumber" SERIAL PRIMARY KEY,  
    "Price" NUMERIC(10, 2) NOT NULL CHECK ("Price" >= 0),  
    "UserID" INTEGER NOT NULL,  
    "OrderDate" TIMESTAMP WITH TIME ZONE DEFAULT  
    CURRENT_TIMESTAMP NOT NULL,  
    CONSTRAINT "FK_Order_User" FOREIGN KEY ("UserID")  
    REFERENCES "User" ("UserID") ON DELETE RESTRICT  
);
```

```
CREATE TABLE "Inventory" (  
    "BookID" INTEGER NOT NULL,  
    "StockQuantity" INTEGER NOT NULL DEFAULT 0 CHECK  
    ("StockQuantity" >= 0),  
    "BookStoreID" INTEGER NOT NULL,  
    CONSTRAINT "PK_Inventory" PRIMARY KEY ("BookID",  
    "BookStoreID"),  
    CONSTRAINT "FK_Inventory_Book" FOREIGN KEY ("BookID")  
    REFERENCES "Book" ("BookID") ON DELETE CASCADE,  
    CONSTRAINT "FK_Inventory_BookStore" FOREIGN KEY  
    ("BookStoreID") REFERENCES "Book Store" ("BookStoreID") ON  
    DELETE RESTRICT  
);
```

```
CREATE TABLE "OrderDetails" (  
    "OrderNumber" INTEGER NOT NULL,  
    "BookID" INTEGER NOT NULL,  
    "Quantity" INTEGER NOT NULL CHECK ("Quantity" > 0),  
    "PriceAtPurchase" NUMERIC(10, 2) NOT NULL CHECK  
    ("PriceAtPurchase" >= 0),
```

```

CONSTRAINT "PK_OrderDetails" PRIMARY KEY ("OrderNumber",
"BookID"),
CONSTRAINT "FK_OrderDetails_Order" FOREIGN KEY
("OrderNumber") REFERENCES "Order" ("OrderNumber") ON DELETE
CASCADE,
CONSTRAINT "FK_OrderDetails_Book" FOREIGN KEY ("BookID")
REFERENCES "Book" ("BookID") ON DELETE RESTRICT
);

CREATE TABLE "BookReview" (
    "BookID" INTEGER NOT NULL,
    "ReviewNumber" INTEGER NOT NULL,
    CONSTRAINT "PK_BookReview" PRIMARY KEY ("BookID",
"ReviewNumber"),
    CONSTRAINT "FK_BookReview_Book" FOREIGN KEY ("BookID")
REFERENCES "Book" ("BookID") ON DELETE CASCADE,
    CONSTRAINT "FK_BookReview_Review" FOREIGN KEY
("ReviewNumber") REFERENCES "Review" ("ReviewNumber") ON
DELETE CASCADE
);

CREATE TABLE "BookProductOwner" (
    "ProductOwnerID" INTEGER NOT NULL,
    "BookID" INTEGER NOT NULL,
    CONSTRAINT "PK_BookProductOwner" PRIMARY KEY
("ProductOwnerID", "BookID"),
    CONSTRAINT "FK_BookProductOwner_Owner" FOREIGN KEY
("ProductOwnerID") REFERENCES "ProductOwner" ("ProductOwnerID")
ON DELETE CASCADE,
    CONSTRAINT "FK_BookProductOwner_Book" FOREIGN KEY
("BookID") REFERENCES "Book" ("BookID") ON DELETE CASCADE
);

INSERT INTO
    "User" (
        "Username",
        "Phone",
        "Email",
        "PasswordHash",
        "DeliveryAddress"
    )
VALUES
    (
        'ivan_franko',
        '380971112233',
        'franko@example.com',
        'hash123_placeholder',

```

```

        'м. Львів, вул. Коперника 1'
    ),
    (
        'lesya_ukrainka',
        '380504445566',
        'lesya@example.com',
        'hash456_placeholder',
        'м. Київ, вул. Саксаганського 97'
    ),
    (
        'taras_shevchenko',
        '380679998877',
        'taras@example.com',
        'hash789_placeholder',
        'м. Канів, Тарасова гора'
    );

```

INSERT INTO

"Publisher" ("Name", "Address")

VALUES

```

    (
        'А-БА-БА-ГА-ЛА-МА-ГА',
        'м. Київ, вул. Басейна 1/2'
    ),
    (
        'Видавництво Старого Лева',
        'м. Львів, вул. Старознесенська 24-26'
    ),
    ('Фоліо', 'м. Харків, вул. Римарська 21');

```

INSERT INTO

"Book Store" ("Name", "Address")

VALUES

```

    ('Книгарня "Є"', 'м. Київ, вул. Лисенка 3'),
    ('Книжковий Лев', 'м. Львів, просп. Шевченка 14'),
    ('Буква', 'м. Київ, ТРЦ "Dream Town");

```

INSERT INTO

"ProductOwner" ("Name", "Email")

VALUES

```

    ('Іван Якович Франко', 'ivan.franko@gmail.com'),
    ('Леся Українка', 'lesya.ukrainka@gmail.com'),
    (
        'Тарас Григорович Шевченко',
        'taras.shevchenko@gmail.com'
    ),
    ('Ліна Костенко', 'lina.kostenko@gmail.com'),

```

('Сергій Жадан', 'serhiy.zhadan@gmail.com');

INSERT INTO

"Review" ("Rating", "Comment", "Date")

VALUES

```
(
    5,
    'Неймовірна книга! Читається на одному диханні.',
    '2024-10-01 10:00:00'
),
(
    4,
    'Гарна історія, але кінцівка трохи розчарувала.',
    '2024-10-02 11:30:00'
),
(
    5,
    'Класика, яку має прочитати кожен.',
    '2024-10-03 14:15:00'
),
(
    2,
    'Дуже нудно. Не зміг дочитати.',
    '2024-10-04 17:00:00'
),
(
    5,
    'Глибокий твір, що змушує задуматись.',
    '2024-10-05 09:20:00'
);
```

INSERT INTO

```
"Book" (
    "ISBN",
    "Name",
    "NumberOfPages",
    "Price",
    "PublisherID"
)
```

VALUES

```
('9786175850987', 'Захар Беркут', 288, 150.00, 1),
('9786176791452', 'Лісова пісня', 256, 220.00, 2),
('9789660350901', 'Кобзар', 704, 350.00, 3),
('9786176798093', 'Ворошиловград', 320, 200.00, 2);
```

INSERT INTO

"Order" ("UserID", "Price", "OrderDate")

VALUES

(1, 150.00, '2024-11-10 10:00:00'),  
(2, 420.00, '2024-11-11 11:30:00'),  
(1, 350.00, '2024-11-12 14:15:00');

INSERT INTO

"Inventory" ("BookID", "StockQuantity", "BookStoreID")

VALUES

(1, 50, 1),  
(2, 30, 1),  
(3, 100, 2),  
(4, 75, 3),  
(1, 20, 2);

INSERT INTO

"OrderDetails" (  
    "OrderNumber",  
    "BookID",  
    "Quantity",  
    "PriceAtPurchase"  
)

VALUES

(1, 1, 1, 150.00),  
(2, 2, 1, 220.00),  
(2, 4, 1, 200.00),  
(3, 3, 1, 350.00);

INSERT INTO

"BookReview" ("BookID", "ReviewNumber")

VALUES

(1, 5),  
(2, 1),  
(3, 3),  
(4, 2),  
(1, 2);

INSERT INTO

"BookProductOwner" ("ProductOwnerID", "BookID")

VALUES

(1, 1),  
(2, 2),  
(3, 3),  
(5, 4);

SELECT

\*

FROM

"User";

SELECT  
\*  
FROM  
"Book";

SELECT  
\*  
FROM  
"Order";

SELECT  
\*  
FROM  
"OrderDetails";

SELECT  
B."Name" AS "Назва Книги",  
P."Name" AS "Видавецъ"  
FROM  
"Book" B  
JOIN "Publisher" P ON B."PublisherID" = P."PublisherID";

SELECT  
O."OrderNumber",  
O."Price",  
U."Username"  
FROM  
"Order" O  
JOIN "User" U ON O."UserID" = U."UserID";

SELECT  
O."OrderNumber",  
B."Name",  
OD."Quantity",  
OD."PriceAtPurchase"  
FROM  
"OrderDetails" OD  
JOIN "Book" B ON OD."BookID" = B."BookID"  
JOIN "Order" O ON OD."OrderNumber" = O."OrderNumber"  
WHERE  
O."OrderNumber" = 2;



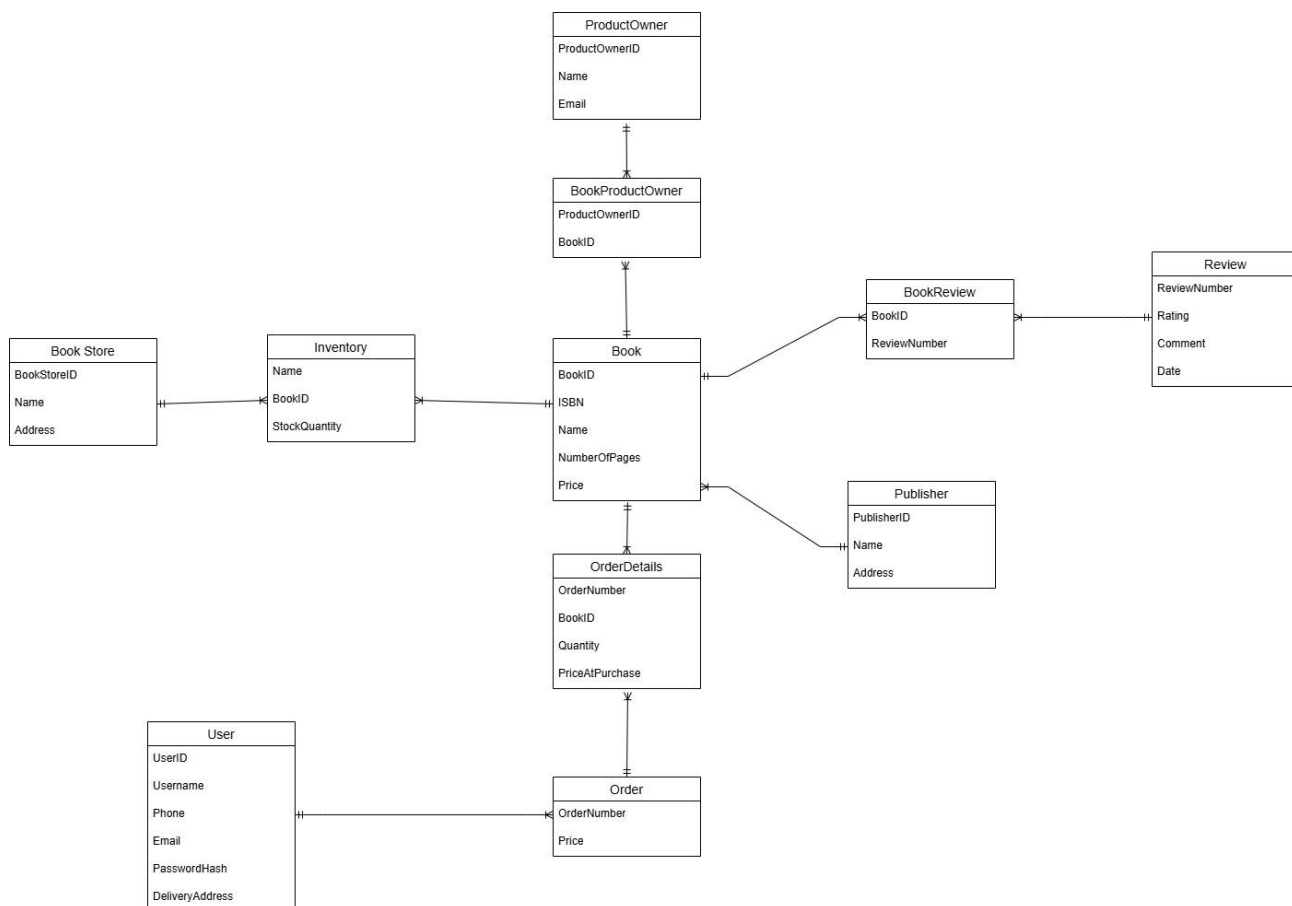
## 2. Короткий письмовий звіт

Схема бази даних розроблена для управління інтернет-магазином книг

Опис таблиць та обмежень

1. **User:** Зберігає інформацію про клієнтів
  - **PK:** UserID
  - **Обмеження:** Username, Phone, Email повинні бути унікальними (UNIQUE)
2. **Publisher:** Довідник видавництв
  - **PK:** PublisherID
3. **Book Store:** Довідник фізичних магазинів або складів
  - **PK:** BookStoreID
4. **ProductOwner:** Автори або правовласники книг
  - **PK:** ProductOwnerID
5. **Book:** Центральна сутність каталогу
  - **PK:** BookID
  - **FK:** PublisherID посилається на таблицю Publisher
  - **Обмеження:** Ціна (Price) та кількість сторінок (NumberOfPages) мають бути додатними
6. **Order:** Заголовки замовлень
  - **PK:** OrderNumber
  - **FK:** UserID посилається на таблицю User
7. **Inventory:** Проміжна таблиця (зв'язок "багато-до-багатьох") для обліку залишків книг у різних магазинах
  - **PK:** Складений ключ (BookID, BookStoreID)
  - **FK:** Посилаються на Book та Book Store
8. **OrderDetails:** Деталізація замовлення (позиції чека)
  - **PK:** Складений ключ (OrderNumber, BookID)
  - **FK:** Посилаються на Order та Book
  - **Припущення:** PriceAtPurchase зберігається окремо, щоб фіксувати ціну на момент продажу, навіть якщо ціна книги в каталозі зміниться
9. **Review та BookReview:** Система відгуків
  - Review зберігає текст і оцінку (1-5)
  - BookReview зв'язує відгук з конкретною книгою
10. **BookProductOwner:** Зв'язує авторів із книгами ("багато-до-багатьох")

### 3. Діаграма ER



## 4. Результати

	OrderNumber integer	Name character varying (255)	Quantity integer	PriceAtPurchase numeric (10,2)
1	2	Лісова пісня	1	220.00
2	2	Ворошиловград	1	200.00