

Name: Richard Raymond J. Canda

Date: February 28, 2025

C.Y.S.: BSCpE – 3A

Laboratory Activity 7:

Laboratory Title: Normalization - Third Normal Form (3NF)

Chapter No. and Topic: Chapter 3 - Database Design and Modeling

Discussions:

This activity will guide students through converting a table to the Third Normal Form (3NF) by removing transitive dependencies.

Activity Description:

Normalize a table in 2NF to 3NF by eliminating transitive dependencies.

Objectives:

- Achieve 3NF by eliminating transitive dependencies.

Materials:

- SQL client

Procedure:

1. Start with a 2NF table:

sql

Copy code

```
CREATE TABLE Books_2NF (  
    BookID INT,  
    Title VARCHAR(100),  
    Author VARCHAR(100),  
    Genre VARCHAR(50),  
    PublisherID INT,  
    PublisherName VARCHAR(100),  
    PublisherAddress VARCHAR(100)  
);
```

1. Insert data:

sql

Copy code

```
INSERT INTO Books_2NF (BookID, Title, Author, Genre,
PublisherID,
PublisherName, PublisherAddress)
VALUES
(1, 'Book A', 'Author1', 'Fiction', 1, 'Publisher1',
'Address1'),
(2, 'Book B', 'Author2', 'Non-Fiction', 1, 'Publisher1',
'Address1');
```

1. Separate publisher details into a new table and link with PublisherID:

sql

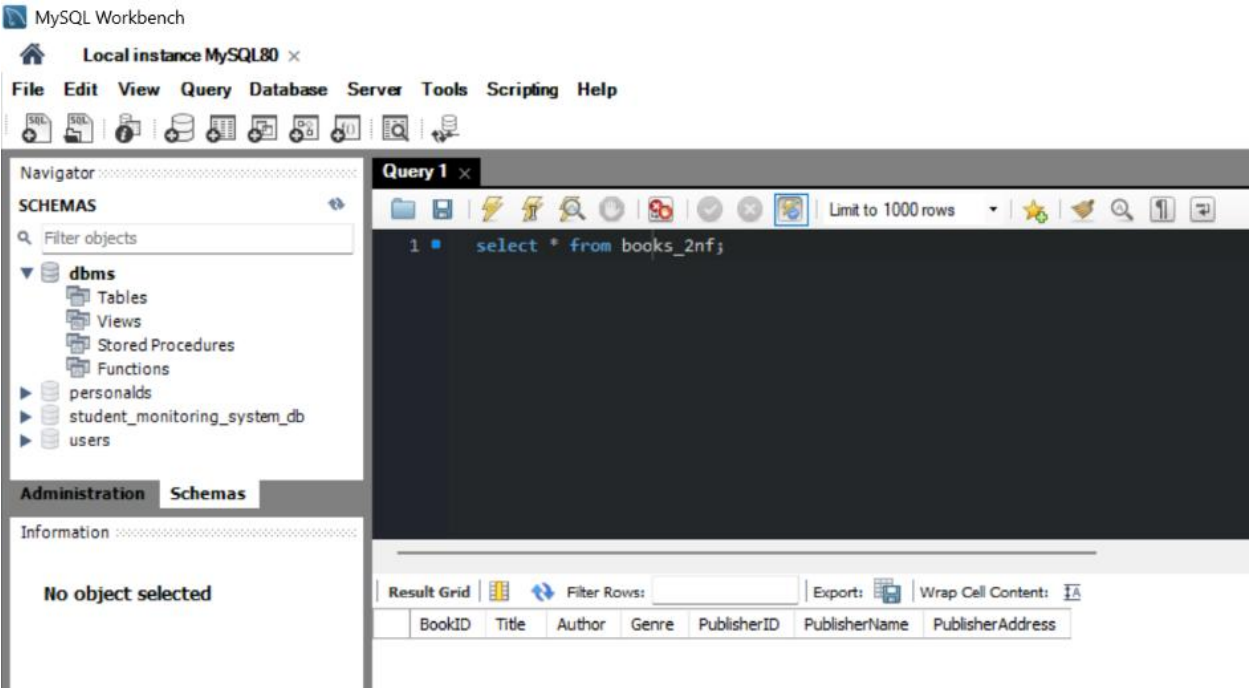
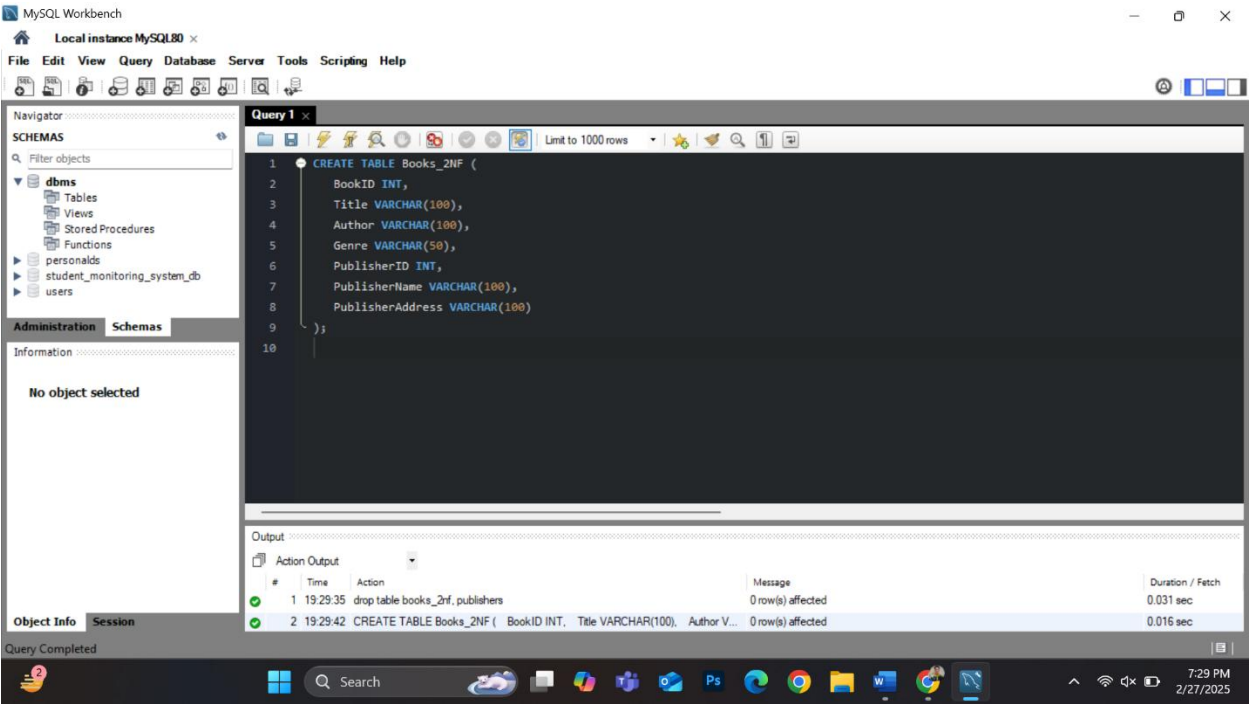
Copy code

```
CREATE TABLE Publishers_3NF (
    PublisherID INT PRIMARY KEY,
    PublisherName VARCHAR(100),
    PublisherAddress VARCHAR(100)
);
```

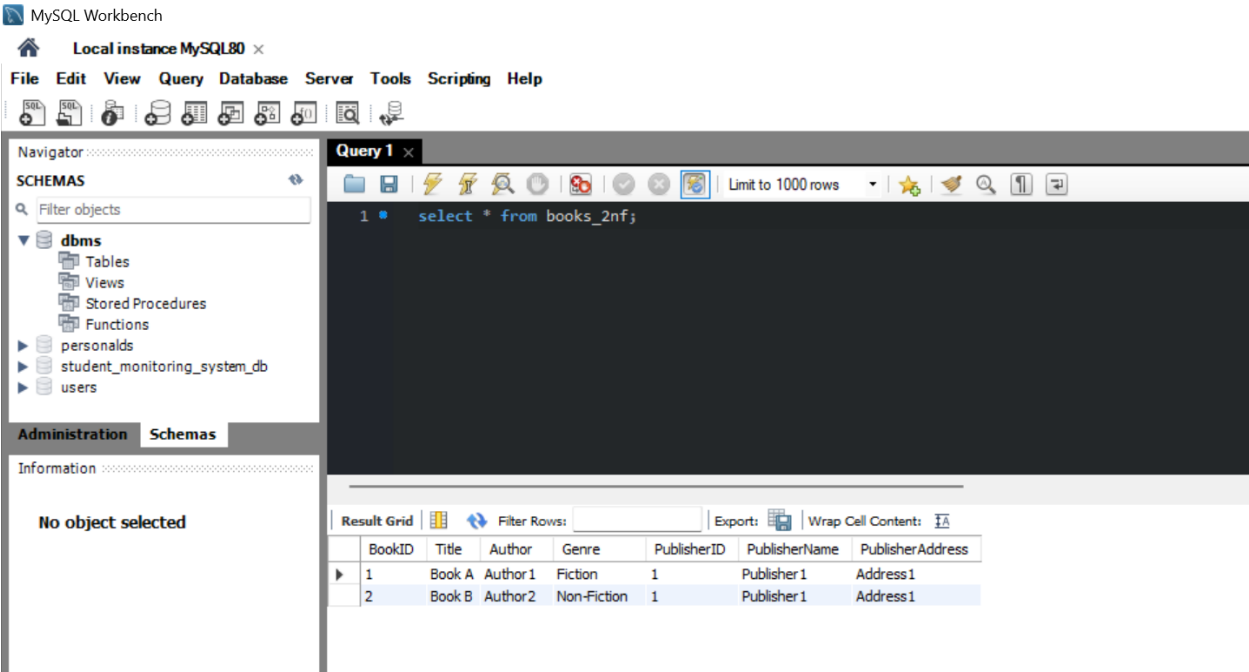
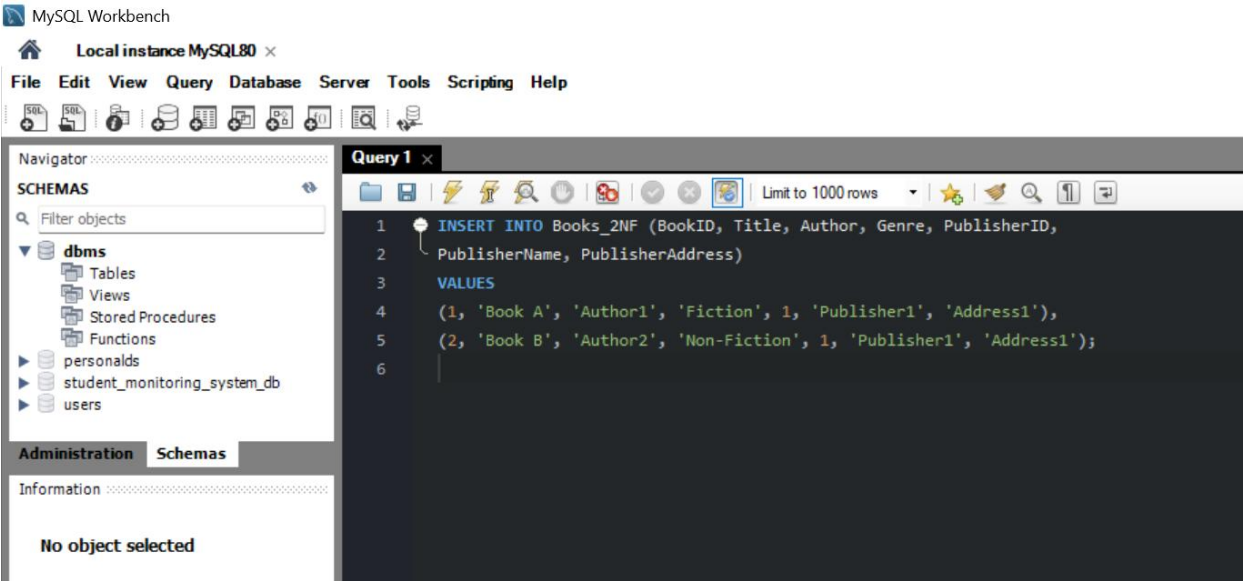
1. Remove PublisherName and PublisherAddress from Books_2NF and adjust the table to use only PublisherID.

Result:

Step 1. Start with a 2NF table:



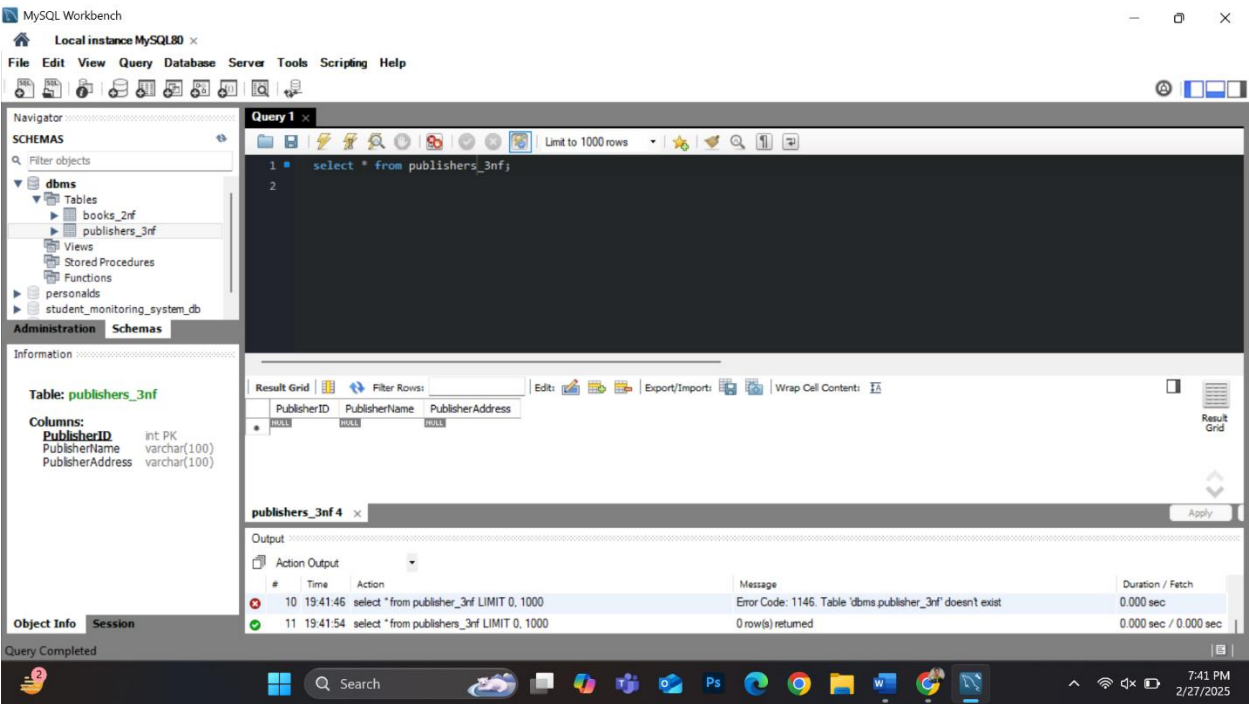
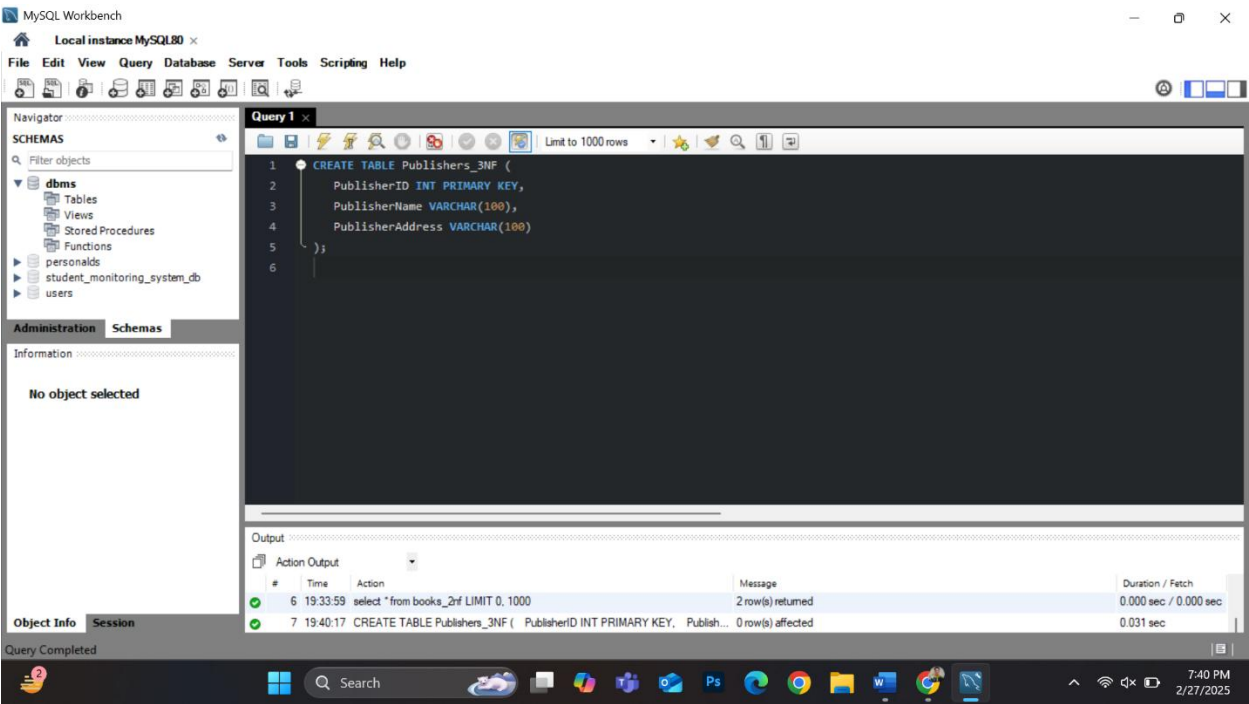
Step 2. Insert data:



This is the data that has been inserted to the books_2nf table.

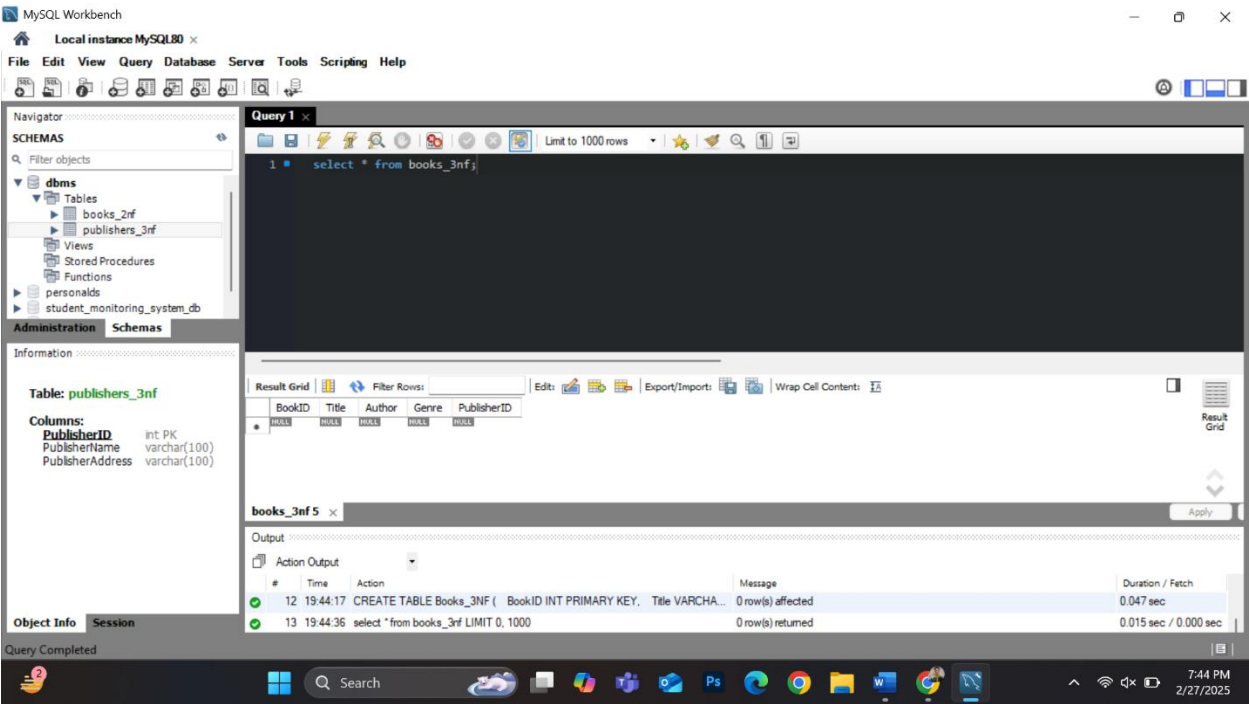
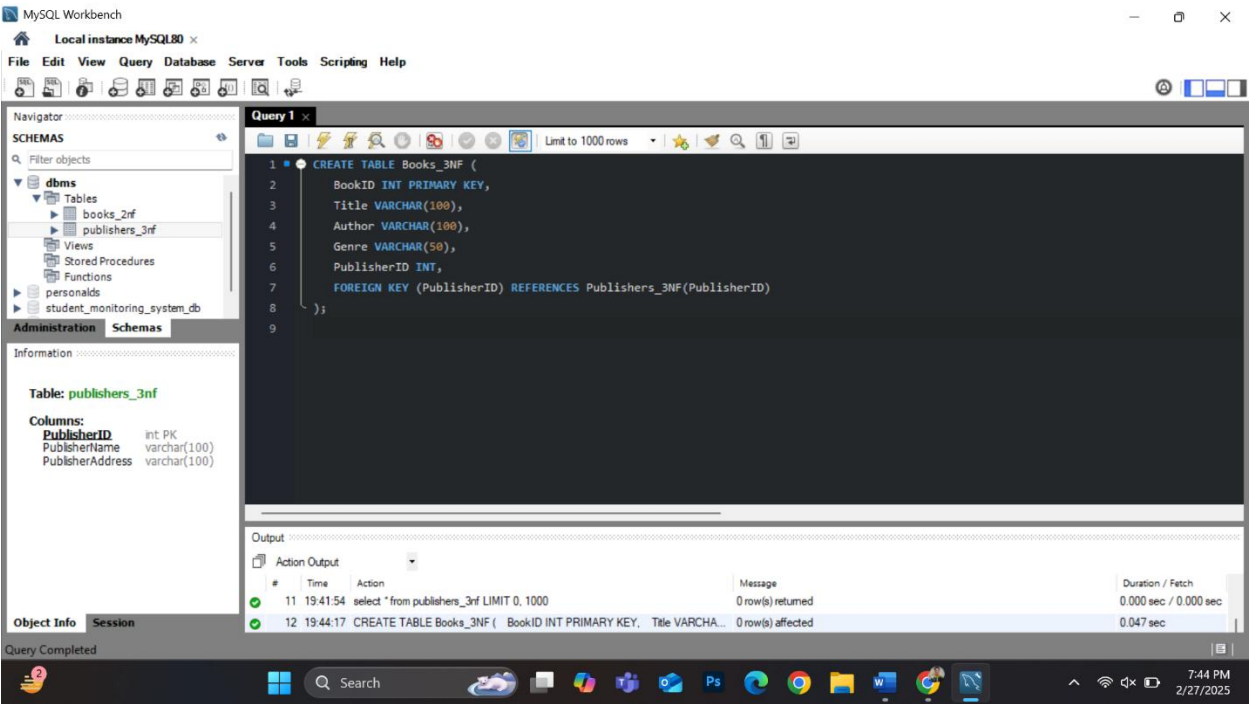
In the Books_2NF table, the PublisherName and PublisherAddress depend on PublisherID, not directly on BookID. This creates a transitive dependency, which violates 3NF.

Step 3: We create the Publishers_3NF table to store publisher details separately.

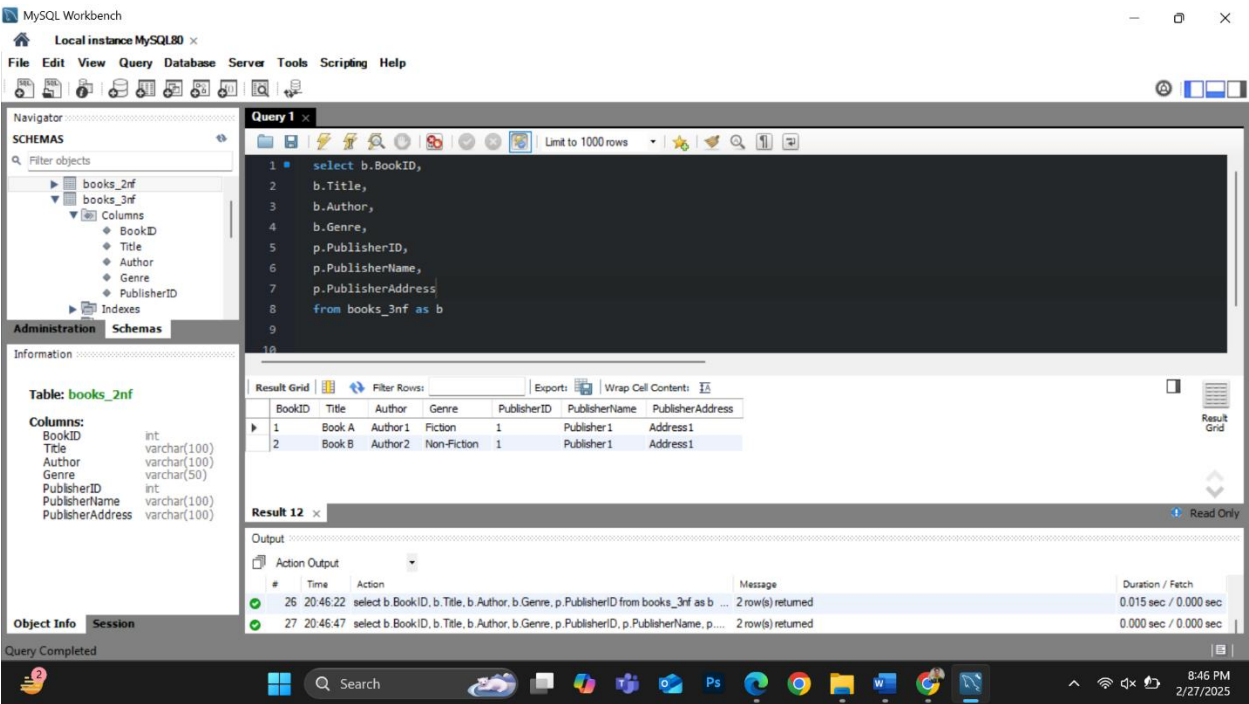
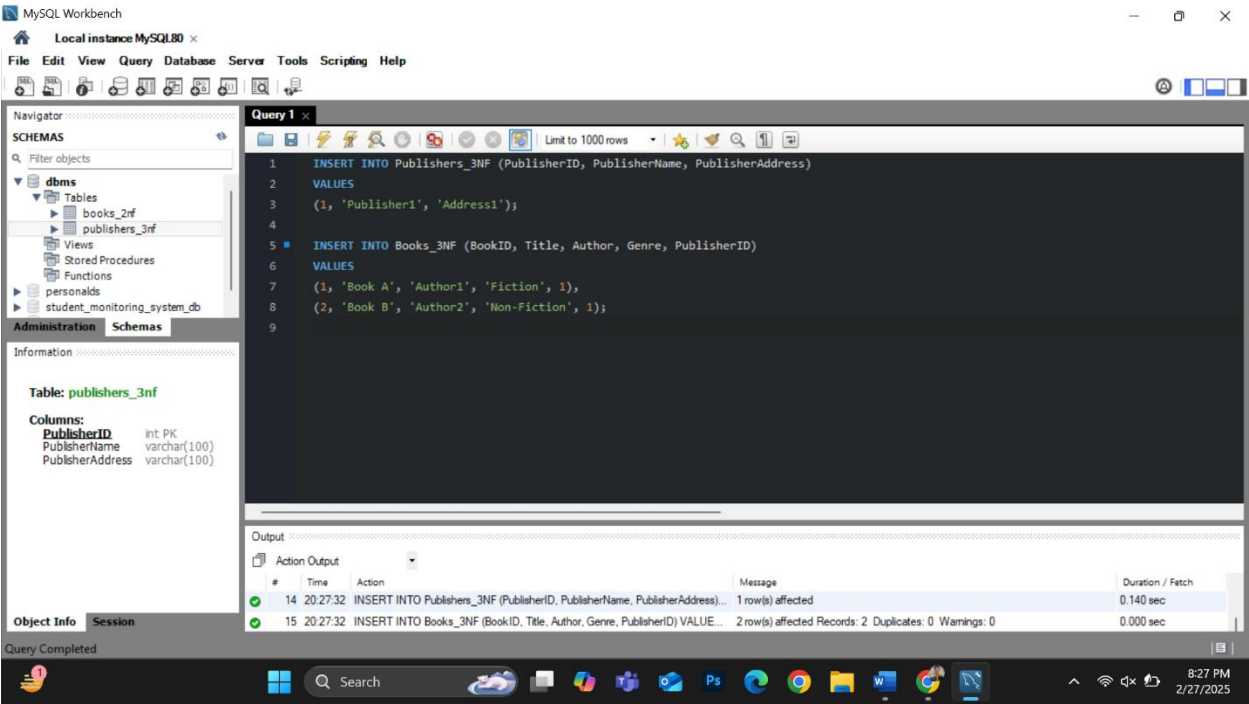


Now, I will modify Books_2NF to eliminate PublisherName and PublisherAddress, leaving only PublisherID:

Step 4: Modify the Books Table to Remove Transitive Dependencies



Step 4: Insert publisher data:



The table is now in 3NF, with no transitive dependencies.

This successfully converts the table from 2NF to 3NF by ensuring **all non-key attributes depend only on the primary key** without transitive dependencies.

Additional Questions/Discussions:

- What are transitive dependencies, and why should they be eliminated?

Answer:

A **transitive dependency** exists when a **non-key attribute** in a table depends on another **non-key attribute**, instead of depending directly on the **primary key**. This indirect relationship creates unnecessary duplication of data and can lead to several issues, including inconsistent updates and inefficient storage.

During the laboratory activity, we analyzed a table that contained book records along with publisher details. We observed that while each book had a unique identifier, the publisher's name and address were stored alongside the book data. Since the publisher's name and address were determined by a publisher ID rather than the book ID, they were not directly dependent on the book's primary key. This created a **transitive dependency**, which needed to be resolved to achieve **Third Normal Form (3NF)**.

Removing transitive dependencies helps keep our database clean and well-organized. It reduces redundancy, prevents inconsistencies, and makes updates and queries more efficient. Through this lab, I realized how important proper database design is—not just for storing data but also for making sure it's easy to manage and retrieve later on.

- How does 3NF improve data integrity?

Answer:

3NF improves data integrity is by **removing redundant data**. When the same information is stored multiple times in a table, there is a higher chance of inconsistencies. For example, if publisher details are stored in a books table and need to be updated, every row containing that publisher must be changed. If even one row is missed, the database will have conflicting information. By moving publisher details to a separate table and linking them with a **foreign key**, we ensure that updates happen in only one place, keeping the data accurate and consistent.

Another way 3NF helps is by **preventing update, insertion, and deletion anomalies**. In an unnormalized database, updating a single piece of information may require modifying multiple records, increasing the risk of human error. Similarly, inserting a new record may be difficult if certain data is required but doesn't yet exist, and deleting records can lead to accidental data loss.

Conclusions:

Through this laboratory activity, we gained practical experience in database normalization, specifically in identifying and eliminating transitive dependencies to achieve Third Normal Form (3NF). By restructuring our tables and properly organizing relationships, we minimized redundancy, improved data integrity, and optimized database performance. This exercise emphasized the importance of a well-designed database, which is crucial for efficient data management in real-world applications.