

Gabriel Q. Escobido

BSCpE – 2B

Laboratory Activity 6:

Laboratory Title: Normalization - Second Normal Form (2NF)

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

Discussions:

This activity will cover the process of converting a table to the Second Normal Form (2NF).

Activity Description:

Given a 1NF table, remove partial dependencies to achieve 2NF.

Objectives:

- Remove partial dependencies and achieve 2NF.

Materials:

- SQL client

Procedure:

1. Create a 1NF table:

sql

Copy code

```
CREATE TABLE Books_1NF (  
    BookID INT,  
    Title VARCHAR(100),  
    Author VARCHAR(100),  
    Genre VARCHAR(50),  
    Publisher VARCHAR(100),  
    PublisherAddress VARCHAR(100)  
);
```

1. Insert sample data:

sql

Copy code

```
INSERT INTO Books_1NF (BookID, Title, Author, Genre, Publisher, PublisherAddress)
```

VALUES

```
(1, 'Book A', 'Author1', 'Fiction', 'Publisher1', 'Address1'),  
(2, 'Book B', 'Author2', 'Non-Fiction', 'Publisher1', 'Address1');
```

1. Create two separate tables to remove partial dependency:

sql

Copy code

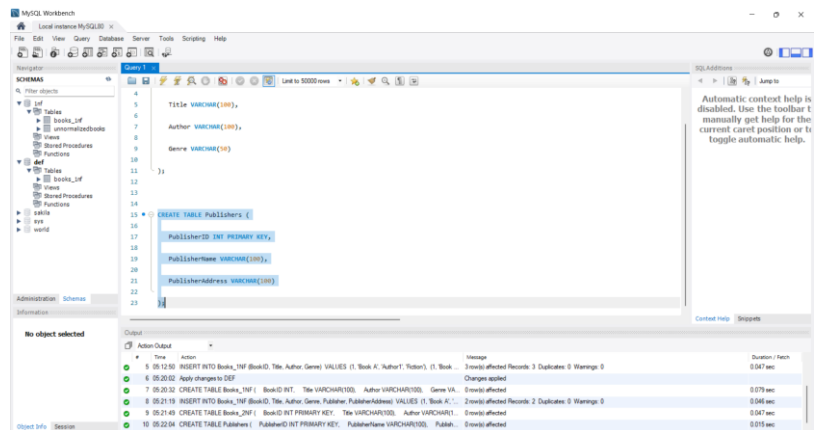
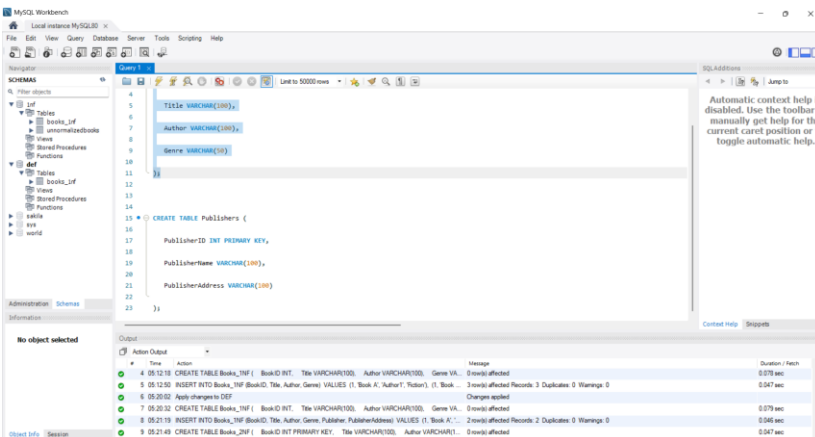
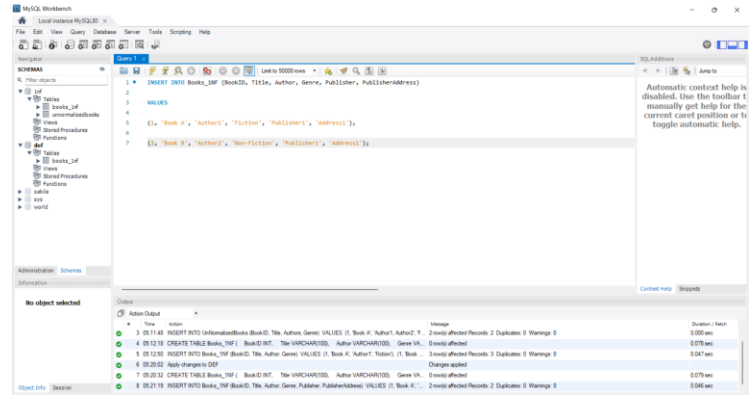
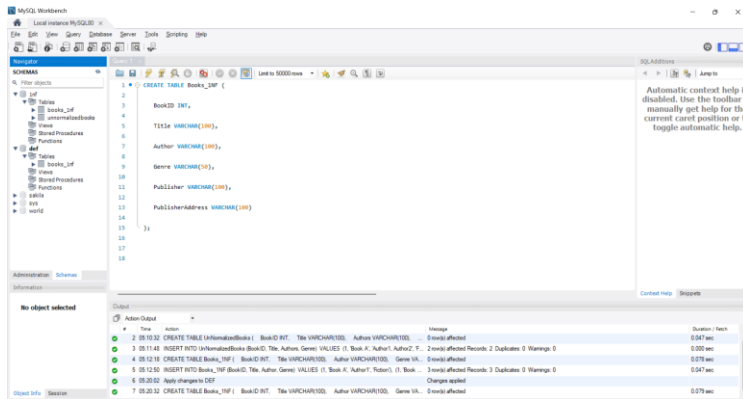
```
CREATE TABLE Books_2NF (  
    BookID INT PRIMARY KEY,  
    Title VARCHAR(100),  
    Author VARCHAR(100),  
    Genre VARCHAR(50)  
);
```

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

1. Move Publisher data into the Publishers table and adjust Books_2NF to include PublisherID as a foreign key.

Result:

The table is now in 2NF with no partial dependencies.



Additional Questions/Discussions:

- What is a partial dependency, and how does 2NF eliminate it?

ANSWER:

A **partial dependency** occurs when a non-prime attribute (an attribute that is not part of a candidate key) depends on only a part of a composite primary key, rather than on the entire key. In our example, the Publisher and PublisherAddress depended only on the Publisher part of the composite key (BookID + Publisher).

To achieve **2NF**, we remove partial dependencies by separating the data into different tables. The Publisher and PublisherAddress are moved to a new table (Publishers), and the original table (Books_2NF) now only contains attributes related to the book itself. We use a foreign key (PublisherID) to link the two tables.

- How do foreign keys help maintain data integrity?

ANSWER:

Foreign keys help maintain data integrity by ensuring that the data in one table corresponds to data in another. In our case, the PublisherID in the Books_2NF table links each book to an entry in the Publishers table. This relationship prevents issues like assigning a book to a publisher that doesn't exist. Foreign keys ensure that data remains consistent across related tables and prevent errors like inserting invalid data into the database.

Conclusions:

ANSWER:

In this activity, we learned how to convert a table from First Normal Form (1NF) to Second Normal Form (2NF) by removing partial dependencies.

- **Partial Dependency** occurs when some attributes depend only on part of a composite primary key, not the entire key. In our example, the Publisher and PublisherAddress depended only on Publisher, not on the whole key (BookID + Publisher).
- **Achieving 2NF:** To remove partial dependencies, we split the table into two: one for books (Books_2NF) and another for publishers (Publishers). We added a PublisherID to the Books_2NF table as a foreign key to connect the tables.
- **Foreign Keys** help maintain data integrity by linking tables and ensuring that related data stays consistent. In our case, the PublisherID ensures that each book is correctly linked to an existing publisher.

In short, normalization helps organize data efficiently and keeps it consistent by removing redundancy and using foreign keys to link related tables.