

Laboratory Activity 5:

Laboratory Title: Normalization - First Normal Form (1NF)

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

Discussions:

This activity demonstrates how to normalize a table to the First Normal Form (1NF).

Activity Description:

Given a sample non-normalized table, convert it to 1NF by ensuring that all columns contain atomic values.

Objectives:

- Understand how to apply 1NF to a database design.
- Convert a table into 1NF.

Materials:

- SQL client

Procedure:

1. Start by creating a sample non-normalized table:

sql

Copy code

```
CREATE TABLE UnNormalizedBooks (  
  
    BookID INT,  
  
    Title VARCHAR(100),  
  
    Authors VARCHAR(100),  
  
    Genre VARCHAR(50)  
  
);
```

1. Insert data into the table:

sql

Copy code

```
INSERT INTO UnNormalizedBooks (BookID, Title, Authors, Genre)  
  
VALUES  
  
(1, 'Book A', 'Author1, Author2', 'Fiction'),  
  
(2, 'Book B', 'Author3', 'Non-Fiction');
```

- 1. Convert to 1NF by creating separate rows for multiple authors:

sql

Copy code

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

- 1. Insert normalized data:

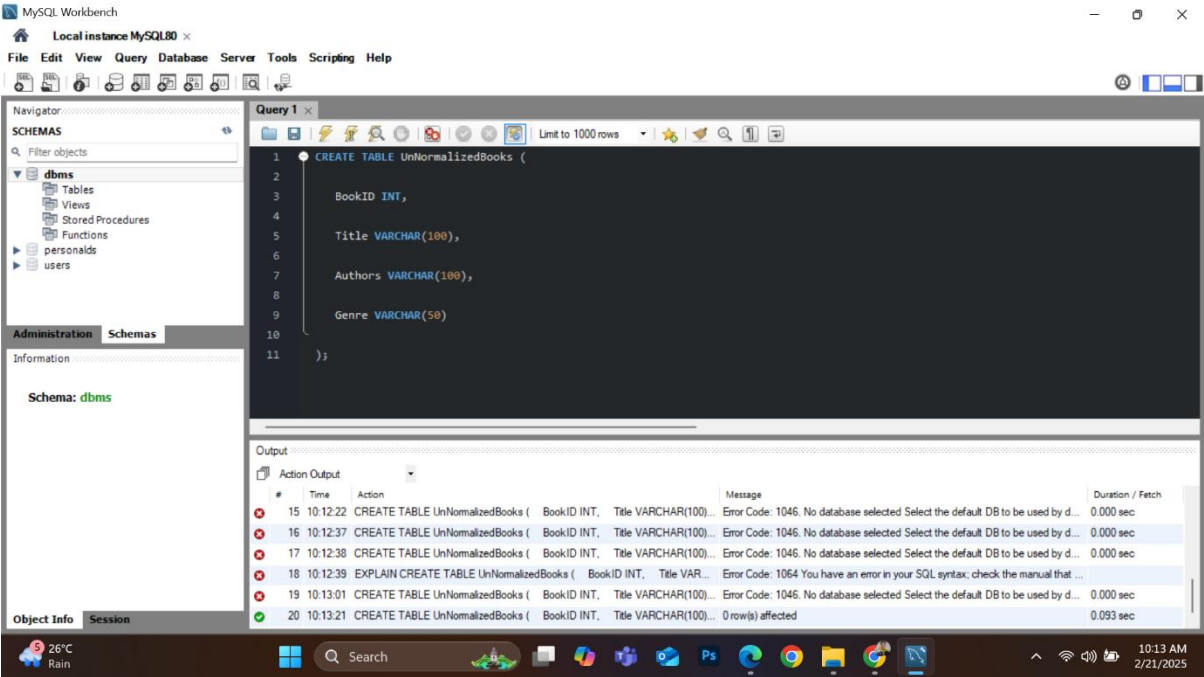
sql

Copy code

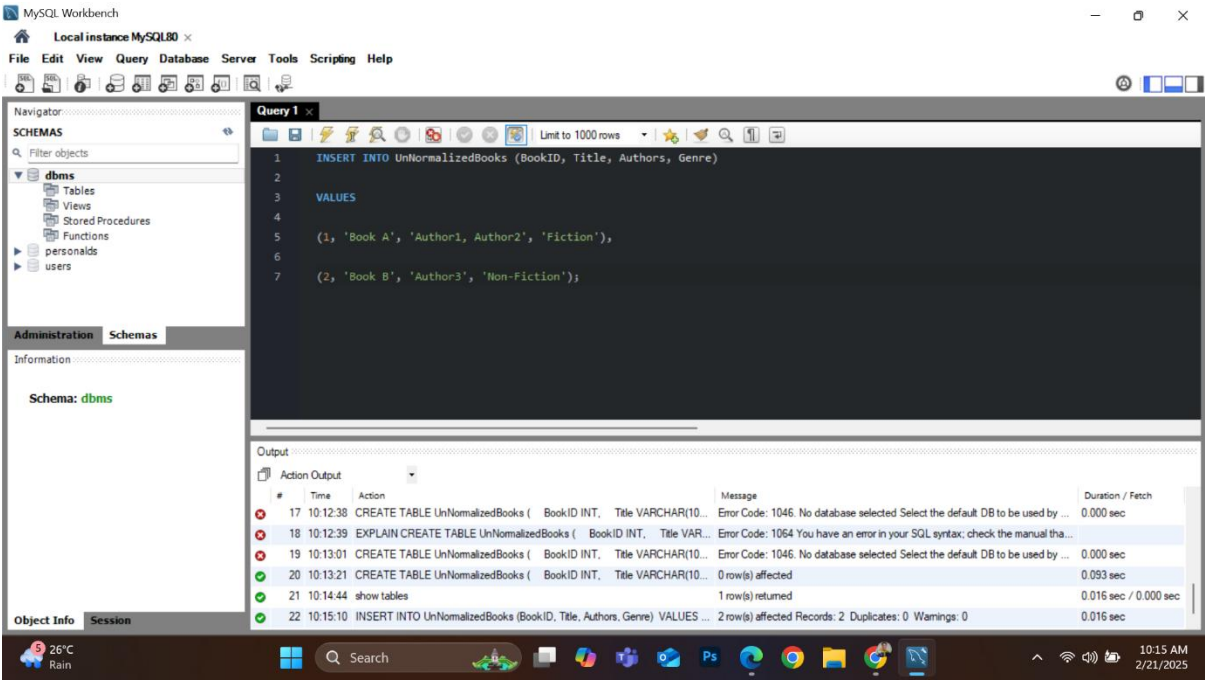
```
INSERT INTO Books_1NF (BookID, Title, Author, Genre)  
  
VALUES  
  
(1, 'Book A', 'Author1', 'Fiction'),  
  
(1, 'Book A', 'Author2', 'Fiction'),  
  
(2, 'Book B', 'Author3', 'Non-Fiction');
```

Result:
The table is now in 1NF with atomic values for each column.

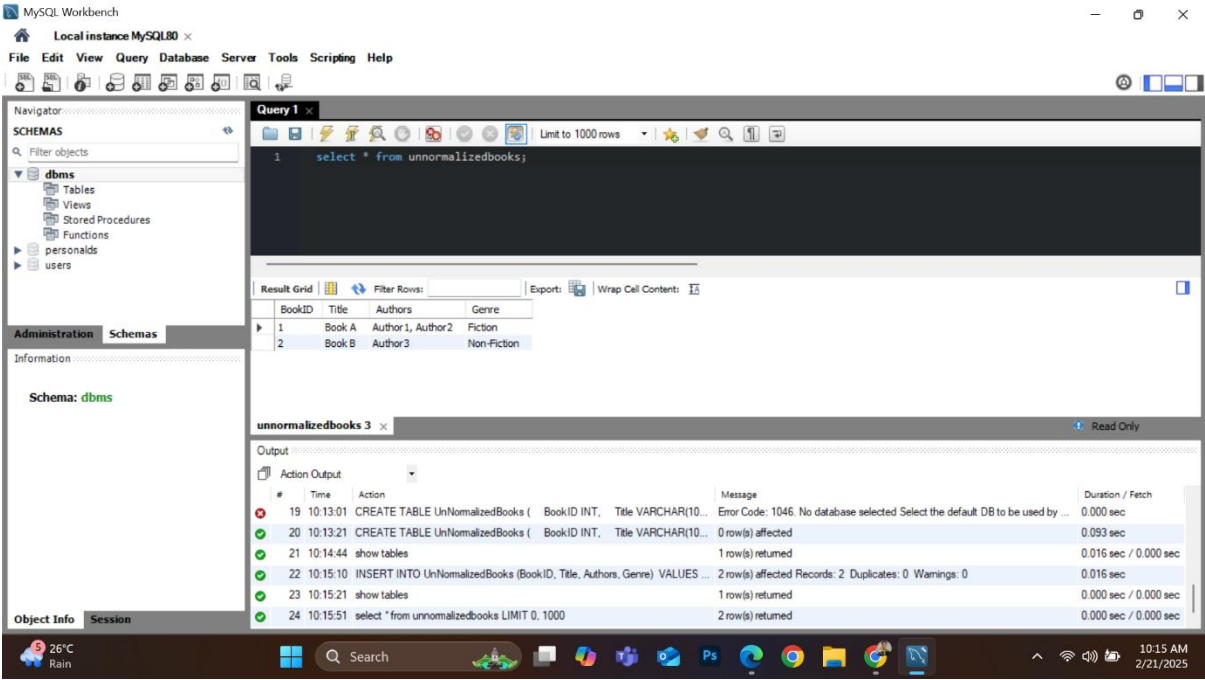
- 1. Start by creating a sample non-normalized table:



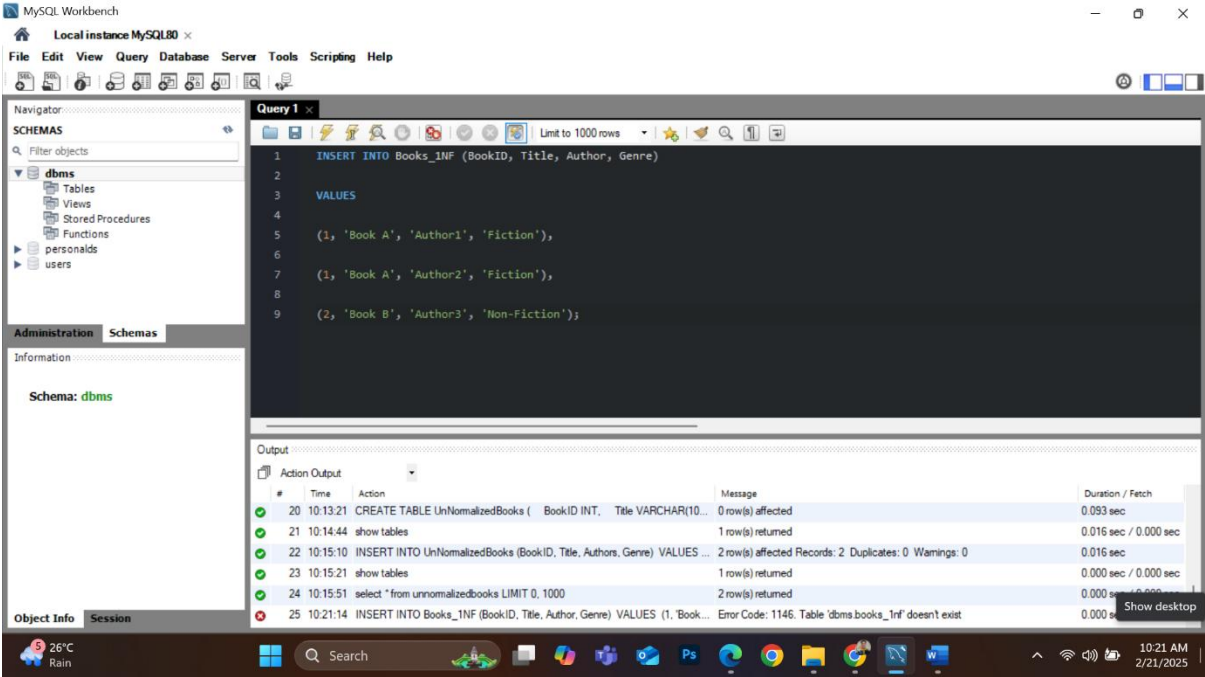
2. Insert data into the table:



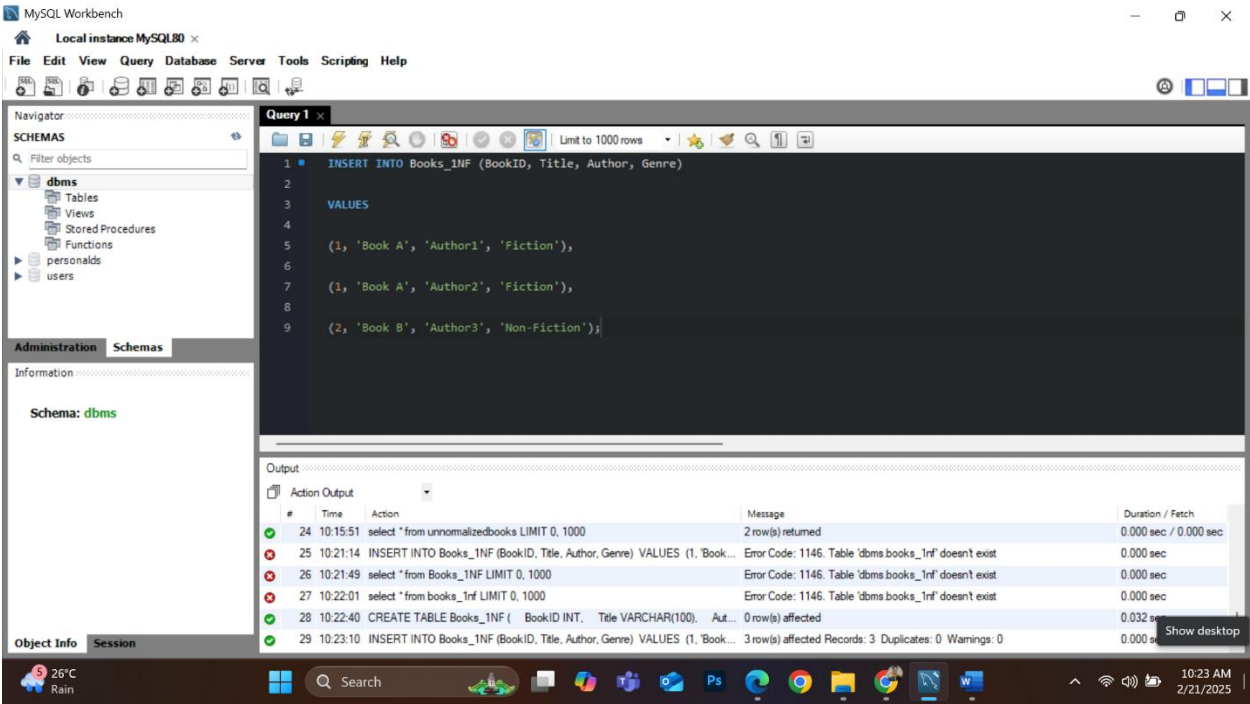
UnNormalizedbooks table with the inputted data.



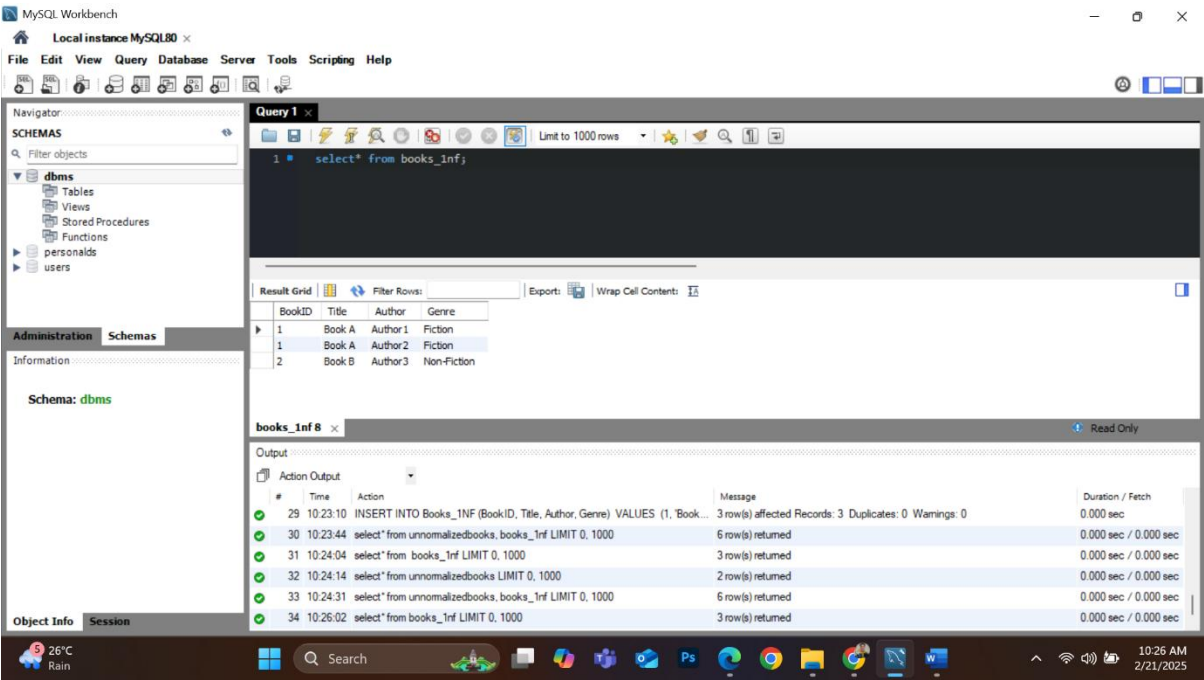
3. Convert to 1NF by creating separate rows for multiple authors:



4. Insert normalized data:



Bools_1NF table with the inputted data.



Additional Questions/Discussions:

- How does 1NF improve data integrity?

Answer:

1NF (First Normal Form) improves data integrity by ensuring that all columns in a table contain atomic (indivisible) values and that each row is unique. This eliminates duplicate data and repeating groups, reducing redundancy and the risk of inconsistent data.

For example, if a table has a column storing multiple values (like a list of phone numbers in one field), it can lead to data anomalies when updating or searching. By enforcing 1NF, we separate these values into different rows or related tables, making queries more efficient and data management more reliable. Since I just learned MySQL, I found that applying 1NF makes writing queries easier because I don’t have to deal with complex string manipulations or inconsistencies.

- What are atomic values, and why are they important?

Answer:

Atomic values are indivisible pieces of data, meaning they cannot be broken down further while still maintaining their meaning. In MySQL (or any relational database), an atomic value ensures that each column holds a single, discrete piece of information. For example, storing multiple phone numbers in a single column as "123-456-7890, 987-654-3210" violates atomicity. Instead, each phone number should be in its own row or a separate table.

Atomic values are important because it makes **querying easier**. Searching, filtering, and sorting data becomes straightforward. It **prevents anomalies**. It makes updates, inserts, and deletes more consistent. And **supports normalization**. It ensures proper database structure and avoids redundancy. I realized that when working with MySQL, using atomic values helps prevent messy queries and makes data retrieval much smoother.

Conclusions:

Through this activity, we demonstrated the process of converting a table into First Normal Form (1NF) by ensuring that all columns contain atomic values and eliminating repeating groups. By applying 1NF, we improved data integrity, reduced redundancy, and made the database structure more efficient for querying and maintenance. This normalization step is essential in designing a well-structured relational database, as it prevents anomalies and ensures consistency. Understanding 1NF lays the foundation for further normalization steps, helping to create a more optimized and scalable database system.