Name: Richard Raymond J. Canda **Date:** February 21, 2025

C.Y.S.: BSCpE – 3A

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:

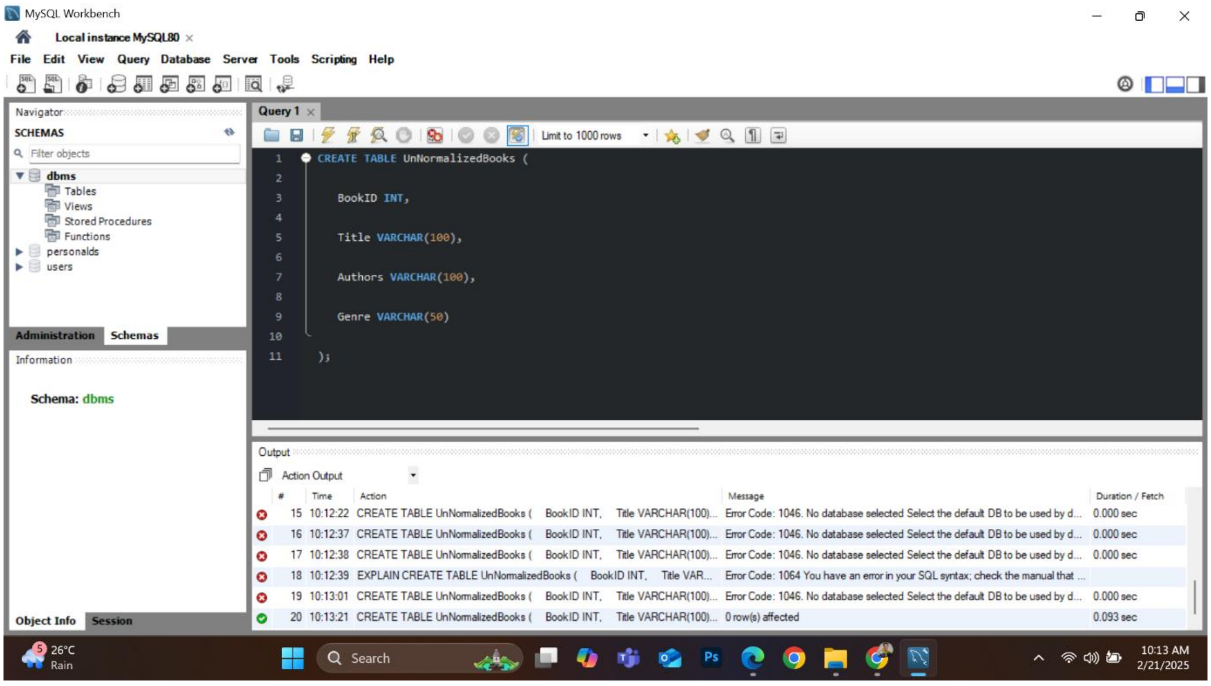
```
sql

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.

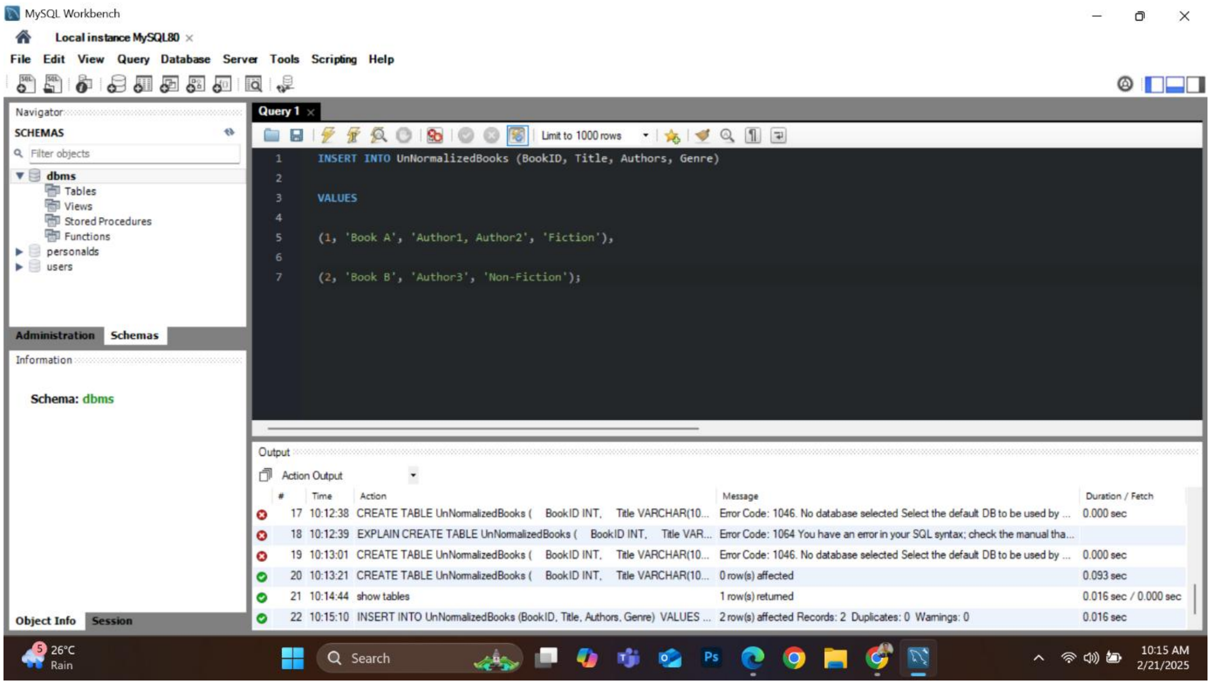
1. Start by creating a sample non-normalized table:



```
CREATE TABLE UnNormalizedBooks (

    BookID INT,

    Title VARCHAR(100),

    Authors VARCHAR(100),

    Genre VARCHAR(50)

 );
```

```
#   Time      Action                                                          Message                                                              Duration / Fetch
15  10:12:22  CREATE TABLE UnNormalizedBooks (   BookID INT,   Title VARCHAR(100)...  Error Code: 1046. No database selected Select the default DB to be used by d...  0.000 sec
16  10:12:37  CREATE TABLE UnNormalizedBooks (   BookID INT,   Title VARCHAR(100)...  Error Code: 1046. No database selected Select the default DB to be used by d...  0.000 sec
17  10:12:38  CREATE TABLE UnNormalizedBooks (   BookID INT,   Title VARCHAR(100)...  Error Code: 1046. No database selected Select the default DB to be used by d...  0.000 sec
18  10:12:39  EXPLAIN CREATE TABLE UnNormalizedBooks (   BookID INT,   Title VAR...   Error Code: 1064 You have an error in your SQL syntax; check the manual that ...
19  10:13:01  CREATE TABLE UnNormalizedBooks (   BookID INT,   Title VARCHAR(100)...  Error Code: 1046. No database selected Select the default DB to be used by d...  0.000 sec
20  10:13:21  CREATE TABLE UnNormalizedBooks (   BookID INT,   Title VARCHAR(100)...  0 row(s) affected                                                              0.093 sec
```

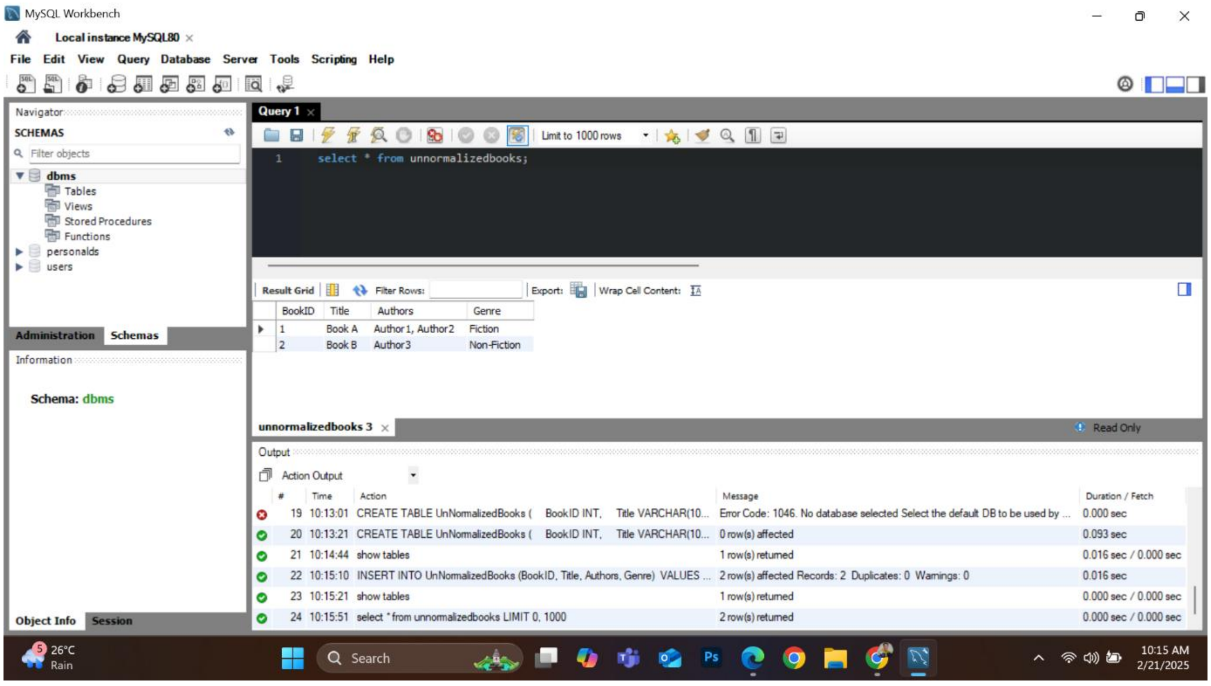
2. Insert data into the table:



```sql
INSERT INTO UnNormalizedBooks (BookID, Title, Authors, Genre)

VALUES

(1, 'Book A', 'Author1, Author2', 'Fiction'),

(2, 'Book B', 'Author3', 'Non-Fiction');
```

```
#   Time      Action                                                          Message                                                               Duration / Fetch
17  10:12:38  CREATE TABLE UnNormalizedBooks (    BookID INT,    Title VARCHAR(10...   Error Code: 1046. No database selected Select the default DB to be used by ...   0.000 sec
18  10:12:39  EXPLAIN CREATE TABLE UnNormalizedBooks (    BookID INT,    Title VAR...   Error Code: 1064 You have an error in your SQL syntax; check the manual tha...
19  10:13:01  CREATE TABLE UnNormalizedBooks (    BookID INT,    Title VARCHAR(10...   Error Code: 1046. No database selected Select the default DB to be used by ...   0.000 sec
20  10:13:21  CREATE TABLE UnNormalizedBooks (    BookID INT,    Title VARCHAR(10...   0 row(s) affected                                                     0.093 sec
21  10:14:44  show tables                                                     1 row(s) returned                                                     0.016 sec / 0.000 sec
22  10:15:10  INSERT INTO UnNormalizedBooks (BookID, Title, Authors, Genre)  VALUES ...   2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0   0.016 sec
```

UnNormalizedbooks table with the inputted data.

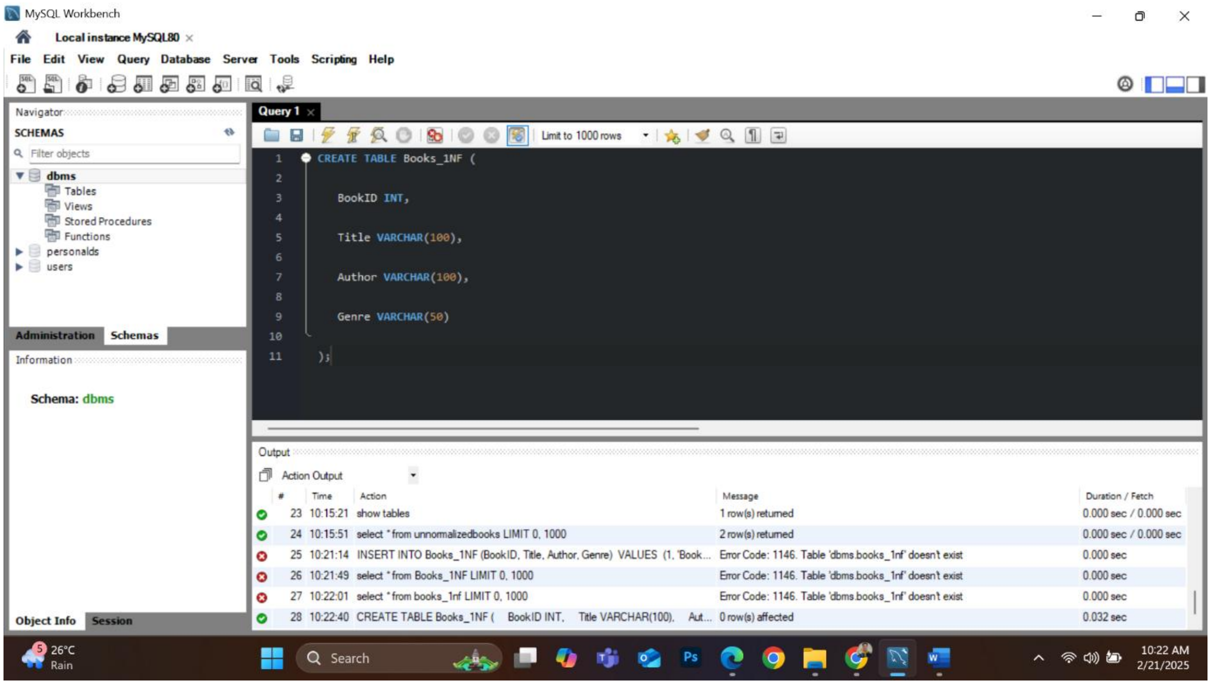


```sql
select * from unnormalizedbooks;
```

BookID	Title	Authors	Genre
1	Book A	Author1, Author2	Fiction
2	Book B	Author3	Non-Fiction

```
#   Time      Action                                                          Message                                                               Duration / Fetch
19  10:13:01  CREATE TABLE UnNormalizedBooks (    BookID INT,    Title VARCHAR(10...   Error Code: 1046. No database selected Select the default DB to be used by ...   0.000 sec
20  10:13:21  CREATE TABLE UnNormalizedBooks (    BookID INT,    Title VARCHAR(10...   0 row(s) affected                                                     0.093 sec
21  10:14:44  show tables                                                     1 row(s) returned                                                     0.016 sec / 0.000 sec
22  10:15:10  INSERT INTO UnNormalizedBooks (BookID, Title, Authors, Genre)  VALUES ...   2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0   0.016 sec
23  10:15:21  show tables                                                     1 row(s) returned                                                     0.000 sec / 0.000 sec
24  10:15:51  select * from unnormalizedbooks LIMIT 0, 1000                   2 row(s) returned                                                     0.000 sec / 0.000 sec
```

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



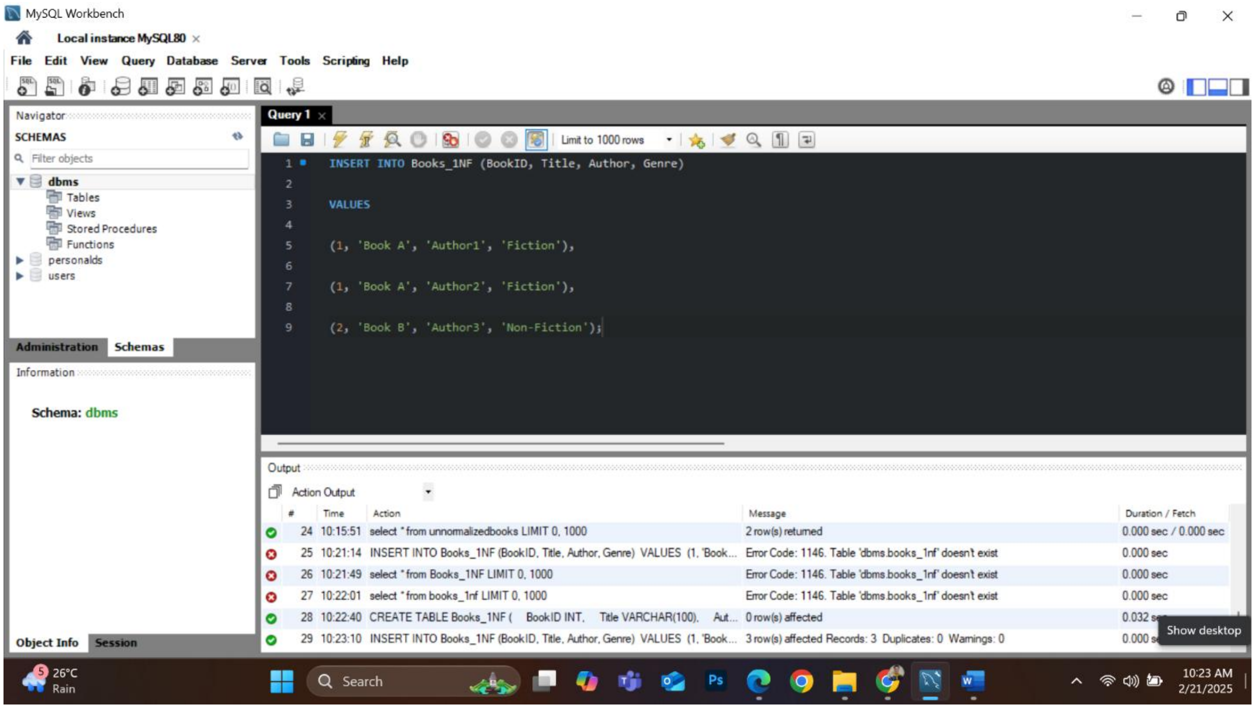
```sql
CREATE TABLE Books_1NF (

    BookID INT,

    Title VARCHAR(100),

    Author VARCHAR(100),

    Genre VARCHAR(50)

  );
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
23	10:15:21	show tables	1 row(s) returned	0.000 sec / 0.000 sec
24	10:15:51	select * from unnormalizedbooks LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
25	10:21:14	INSERT INTO Books_1NF (BookID, Title, Author, Genre) VALUES (1, 'Book...	Error Code: 1146. Table 'dbms.books_1nf' doesn't exist	0.000 sec
26	10:21:49	select * from Books_1NF LIMIT 0, 1000	Error Code: 1146. Table 'dbms.books_1nf' doesn't exist	0.000 sec
27	10:22:01	select * from books_1nf LIMIT 0, 1000	Error Code: 1146. Table 'dbms.books_1nf' doesn't exist	0.000 sec
28	10:22:40	CREATE TABLE Books_1NF (BookID INT, Title VARCHAR(100), Aut...	0 row(s) affected	0.032 sec

4. Insert normalized data:



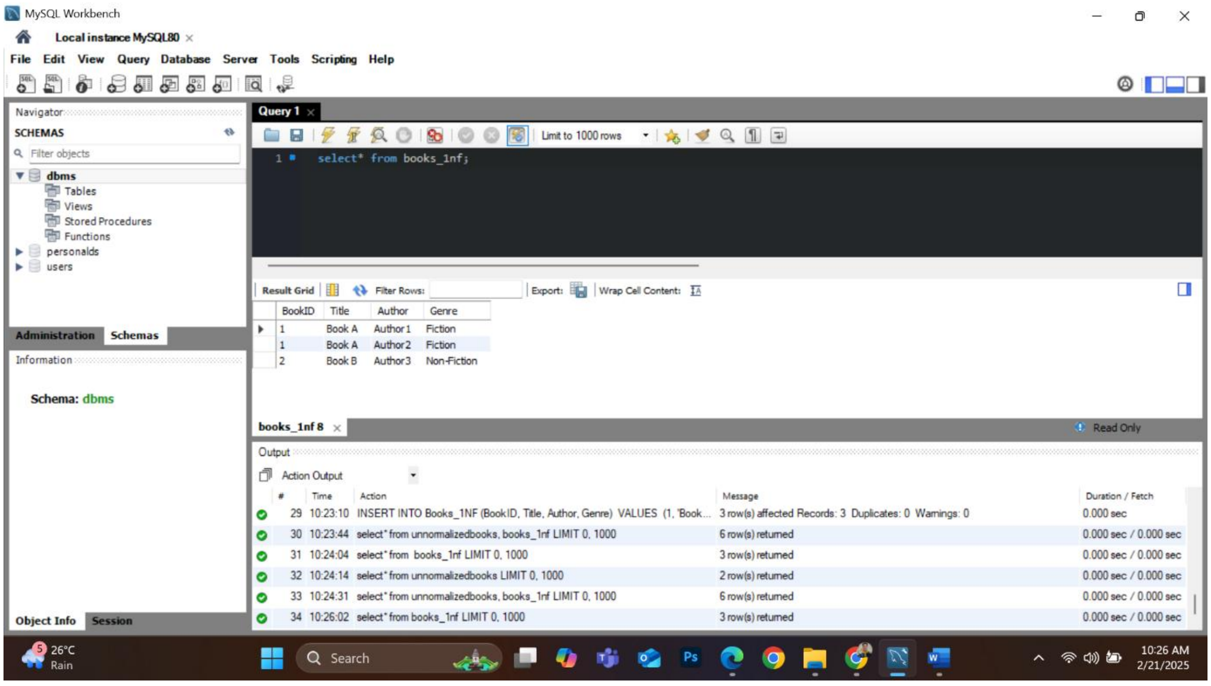
```sql
INSERT INTO Books_1NF (BookID, Title, Author, Genre)

VALUES

(1, 'Book A', 'Author1', 'Fiction'),

(1, 'Book A', 'Author2', 'Fiction'),

(2, 'Book B', 'Author3', 'Non-Fiction');
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
24	10:15:51	select * from unnormalizedbooks LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
25	10:21:14	INSERT INTO Books_1NF (BookID, Title, Author, Genre) VALUES (1, 'Book...	Error Code: 1146. Table 'dbms.books_1nf' doesn't exist	0.000 sec
26	10:21:49	select * from Books_1NF LIMIT 0, 1000	Error Code: 1146. Table 'dbms.books_1nf' doesn't exist	0.000 sec
27	10:22:01	select * from books_1nf LIMIT 0, 1000	Error Code: 1146. Table 'dbms.books_1nf' doesn't exist	0.000 sec
28	10:22:40	CREATE TABLE Books_1NF (BookID INT, Title VARCHAR(100), Aut...	0 row(s) affected	0.032 se
29	10:23:10	INSERT INTO Books_1NF (BookID, Title, Author, Genre) VALUES (1, 'Book...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 s

Bools_1NF table with the inputted data.



```
select* from books_1nf;
```

BookID	Title	Author	Genre
1	Book A	Author1	Fiction
1	Book A	Author2	Fiction
2	Book B	Author3	Non-Fiction

```
#   Time      Action                                                                Message                                                   Duration / Fetch
29  10:23:10  INSERT INTO Books_1NF (BookID, Title, Author, Genre) VALUES (1, 'Book...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0  0.000 sec
30  10:23:44  select* from unnomalizedbooks, books_1nf LIMIT 0, 1000                   6 row(s) returned                                          0.000 sec / 0.000 sec
31  10:24:04  select* from  books_1nf LIMIT 0, 1000                                    3 row(s) returned                                          0.000 sec / 0.000 sec
32  10:24:14  select* from unnomalizedbooks LIMIT 0, 1000                              2 row(s) returned                                          0.000 sec / 0.000 sec
33  10:24:31  select* from unnomalizedbooks, books_1nf LIMIT 0, 1000                   6 row(s) returned                                          0.000 sec / 0.000 sec
34  10:26:02  select* from books_1nf LIMIT 0, 1000                                     3 row(s) returned                                          0.000 sec / 0.000 sec
```

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 p**revents 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.