EMMAN ACE MENION **DBMS-2B**

BSCpE 3A **SCORE:_________**

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)

);
```

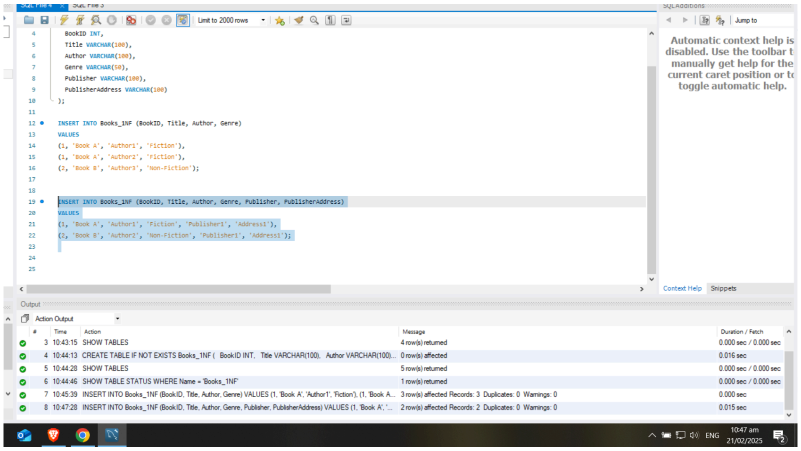
1. Move Publisher data into the `Publishers` table and adjust `Books_2NF` to include `PublisherID` as a foreign key.

EMMAN ACE MENION DBMS-2B

BSCpE 3A SCORE:_________

Result:

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



SQL File 4
SQL File 3
Limit to 2000 rows
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or tc toggle automatic help.
Context Help
Snippets

```sql
    BookID INT,
    Title VARCHAR(100),
    Author VARCHAR(100),
    Genre VARCHAR(50),
    Publisher VARCHAR(100),
    PublisherAddress VARCHAR(100)
);

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');


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');
```

Output

Action Output

```
#  Time      Action                                                                                        Message                                                       Duration / Fetch
3  10:43:15  SHOW TABLES                                                                                   4 row(s) returned                                             0.000 sec / 0.000 sec
4  10:44:13  CREATE TABLE IF NOT EXISTS Books_1NF (  BookID INT,  Title VARCHAR(100),  Author VARCHAR(100)...  0 row(s) affected                                             0.016 sec
5  10:44:28  SHOW TABLES                                                                                   5 row(s) returned                                             0.000 sec / 0.000 sec
6  10:44:46  SHOW TABLE STATUS WHERE Name = 'Books_1NF'                                                    1 row(s) returned                                             0.000 sec / 0.000 sec
7  10:45:39  INSERT INTO Books_1NF (BookID, Title, Author, Genre) VALUES (1, 'Book A', 'Author1', 'Fiction'), (1, 'Book A...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0  0.000 sec
8  10:47:28  INSERT INTO Books_1NF (BookID, Title, Author, Genre, Publisher, PublisherAddress) VALUES (1, 'Book A', '...  2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0  0.015 sec
```

10:47 am 21/02/2025

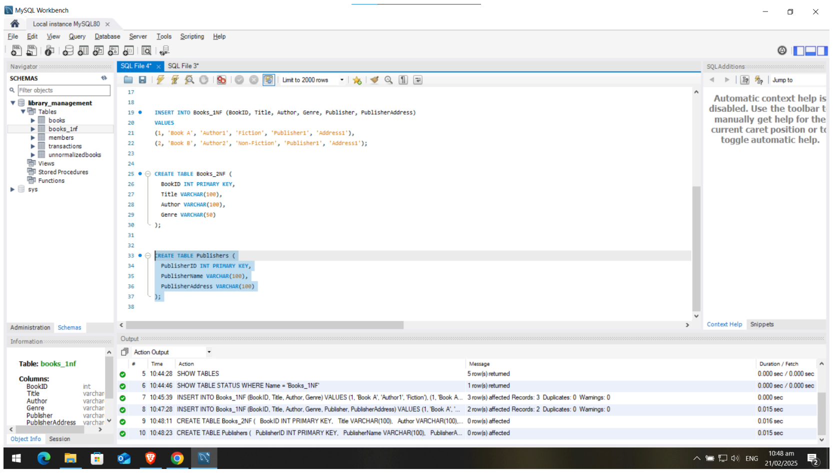


Table: books_1nf

Columns:

BookID	int
Title	varchar
Author	varchar
Genre	varchar
Publisher	varchar
PublisherAddress	varchar

```sql
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');


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)
);
```

Output

Action Output

```
#   Time      Action                                                                                        Message                                                       Duration / Fetch
5   10:44:28  SHOW TABLES                                                                                   5 row(s) returned                                             0.000 sec / 0.000 sec
6   10:44:46  SHOW TABLE STATUS WHERE Name = 'Books_1NF'                                                    1 row(s) returned                                             0.000 sec / 0.000 sec
7   10:45:39  INSERT INTO Books_1NF (BookID, Title, Author, Genre) VALUES (1, 'Book A', 'Author1', 'Fiction'), (1, 'Book A...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0  0.000 sec
8   10:47:28  INSERT INTO Books_1NF (BookID, Title, Author, Genre, Publisher, PublisherAddress) VALUES (1, 'Book A', '...  2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0  0.015 sec
9   10:48:11  CREATE TABLE Books_2NF (  BookID INT PRIMARY KEY,  Title VARCHAR(100),  Author VARCHAR(100),...  0 row(s) affected                                             0.016 sec
10  10:48:23  CREATE TABLE Publishers (  PublisherID INT PRIMARY KEY,  PublisherName VARCHAR(100),  PublisherA...  0 row(s) affected                                             0.015 sec
```

10:48 am 21/02/2025

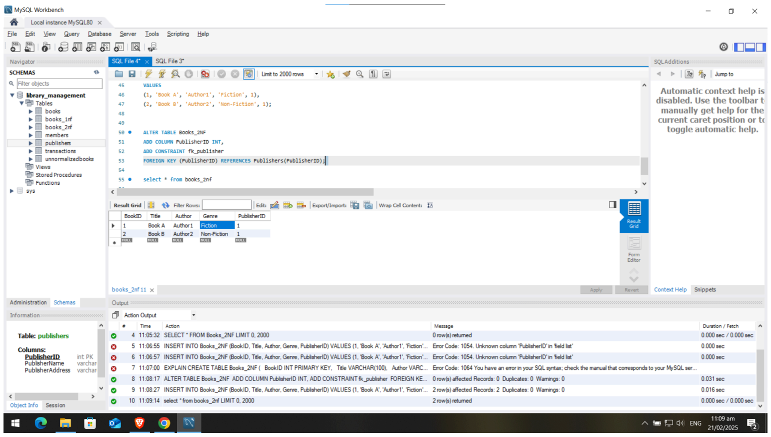


Table: publishers

Columns:

PublisherID	int PK
PublisherName	varchar
PublisherAddress	varchar

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


ALTER TABLE Books_2NF
ADD COLUMN PublisherID INT,
ADD CONSTRAINT fk_publisher
FOREIGN KEY (PublisherID) REFERENCES Publishers(PublisherID);

select * from books_2nf
```

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

Output

Action Output

```
#   Time      Action                                                                                        Message                                                                                  Duration / Fetch
4   11:05:32  SELECT * FROM Books_2NF LIMIT 0, 2000                                                         0 row(s) returned                                                                        0.000 sec / 0.000 sec
5   11:06:55  INSERT INTO Books_2NF (BookID, Title, Author, Genre, PublisherID) VALUES (1, 'Book A', 'Author1', 'Fiction'...  Error Code: 1054. Unknown column 'PublisherID' in 'field list'  0.000 sec
6   11:06:57  INSERT INTO Books_2NF (BookID, Title, Author, Genre, PublisherID) VALUES (1, 'Book A', 'Author1', 'Fiction'...  Error Code: 1054. Unknown column 'PublisherID' in 'field list'  0.000 sec
7   11:07:00  EXPLAIN CREATE TABLE Books_2NF (  BookID INT PRIMARY KEY,  Title VARCHAR(100),  Author VARC...  Error Code: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MySQL ser...
8   11:08:17  ALTER TABLE Books_2NF  ADD COLUMN PublisherID INT, ADD CONSTRAINT fk_publisher  FOREIGN KE...  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0                                  0.031 sec
9   11:08:27  INSERT INTO Books_2NF (BookID, Title, Author, Genre, PublisherID) VALUES (1, 'Book A', 'Author1', 'Fiction'...  2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0  0.016 sec
10  11:09:14  select * from books_2nf LIMIT 0, 2000                                                         2 row(s) returned                                                                        0.000 sec / 0.000 sec
```

11:09 am 21/02/2025

Additional Questions/Discussions:

- What is a partial dependency, and how does 2NF eliminate it?
  - A **partial dependency** occurs when a **non-key attribute** depends on only part of a **composite primary key**, rather than the whole key. This leads to data redundancy and inefficiencies.In **1NF**, the `Publisher` and `PublisherAddress` fields depend only on `PublisherName`, not on `BookID`. To achieve **2NF**, we separate this data into a `Publishers` table and use `PublisherID` as a **foreign key** in `Books_2NF`. This eliminates partial dependencies, ensuring that every non-key attribute depends only on the entire primary key, reducing redundancy and improving data consistency.

- How do foreign keys help maintain data integrity?

- A **foreign key** is a column that establishes a relationship between two tables, ensuring consistency in the database.
- In **2NF**, `PublisherID` in `Books_2NF` references `PublisherID` in the `Publishers` table. This prevents **orphan records** (books with non-existing publishers) and maintains valid relationships. Foreign keys also ensure that updates or deletions in the `Publishers` table follow **referential integrity**, preventing accidental data loss or inconsistencies.

Conclusions: By converting the table from 1NF to 2NF, we eliminate partial dependencies, ensuring that all non-key attributes fully depend on the primary key. This reduces data redundancy, improves data integrity, and simplifies updates.

Using foreign keys further strengthens referential integrity, preventing orphaned data and enforcing logical relationships between tables. Achieving 2NF makes the database more structured, efficient, and easier to manage.