

JOHN CARLO S. GUMAD

Laboratory Activity 6:

The screenshot shows the SQL Enterprise Manager interface with the 'librarymanagement' schema selected. The 'books' table is highlighted in the Navigator. The main window displays the following SQL script:

```
1 CREATE TABLE Books_1NF (  
2     BookID INT,  
3     Title VARCHAR(100),  
4     Author VARCHAR(100),  
5     Genre VARCHAR(50),  
6     Publisher VARCHAR(100),  
7     PublisherAddress VARCHAR(100)  
8 );  
9  
10 INSERT INTO Books_1NF (BookID, Title, Author, Genre, Publisher, PublisherAddress)  
11 VALUES  
12 (1, 'Book A', 'Author1', 'Fiction', 'Publisher1', 'Address1'),  
13 (2, 'Book B', 'Author2', 'Non-Fiction', 'Publisher1', 'Address1');  
14  
15 CREATE TABLE Publishers (  
16     PublisherID INT PRIMARY KEY,  
17     PublisherName VARCHAR(100),  
18     PublisherAddress VARCHAR(100)  
19 );
```

The Output window shows the following messages:

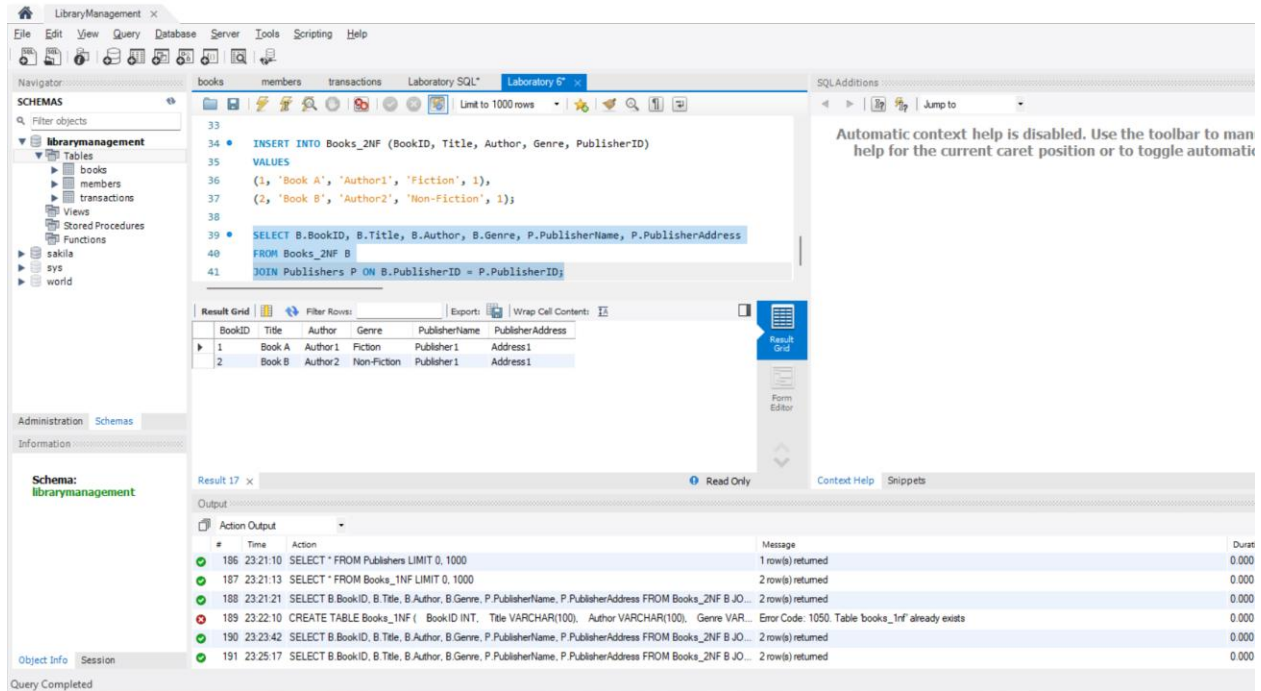
#	Time	Action	Message
185	23:20:48	SELECT * FROM Books_1NF LIMIT 0, 1000	2 row(s) returned
186	23:21:10	SELECT * FROM Publishers LIMIT 0, 1000	1 row(s) returned
187	23:21:13	SELECT * FROM Books_1NF LIMIT 0, 1000	2 row(s) returned
188	23:21:21	SELECT B.BookID, B.Title, B.Author, B.Genre, P.PublisherName, P.PublisherAddress FROM Books_1NF B JOIN Publishers P ON B.Publisher = P.PublisherID	2 row(s) returned
189	23:22:10	CREATE TABLE Books_1NF (BookID INT, Title VARCHAR(100), Author VARCHAR(100), Genre VARCHAR(50), Publisher VARCHAR(100), PublisherAddress VARCHAR(100))	Error Code: 1050. Table 'books_1nf' already exists
190	23:23:42	SELECT B.BookID, B.Title, B.Author, B.Genre, P.PublisherName, P.PublisherAddress FROM Books_1NF B JOIN Publishers P ON B.Publisher = P.PublisherID	2 row(s) returned

The screenshot shows the SQL Enterprise Manager interface with the 'librarymanagement' schema selected. The 'books' table is highlighted in the Navigator. The main window displays the following SQL script:

```
21 CREATE TABLE Books_2NF (  
22     BookID INT PRIMARY KEY,  
23     Title VARCHAR(100),  
24     Author VARCHAR(100),  
25     Genre VARCHAR(50),  
26     PublisherID INT,  
27     FOREIGN KEY (PublisherID) REFERENCES Publishers(PublisherID)  
28 );  
29  
30 INSERT INTO Publishers (PublisherID, PublisherName, PublisherAddress)  
31 VALUES  
32 (1, 'Publisher1', 'Address1');  
33  
34 INSERT INTO Books_2NF (BookID, Title, Author, Genre, PublisherID)  
35 VALUES  
36 (1, 'Book A', 'Author1', 'Fiction', 1),  
37 (2, 'Book B', 'Author2', 'Non-Fiction', 1);  
38  
39 SELECT B.BookID, B.Title, B.Author, B.Genre, P.PublisherName, P.PublisherAddress  
40 FROM Books_2NF B JOIN Publishers P ON B.PublisherID = P.PublisherID
```

The Output window shows the following messages:

#	Time	Action	Message
185	23:20:48	SELECT * FROM Books_1NF LIMIT 0, 1000	2 row(s) returned
186	23:21:10	SELECT * FROM Publishers LIMIT 0, 1000	1 row(s) returned
187	23:21:13	SELECT * FROM Books_1NF LIMIT 0, 1000	2 row(s) returned
188	23:21:21	SELECT B.BookID, B.Title, B.Author, B.Genre, P.PublisherName, P.PublisherAddress FROM Books_1NF B JOIN Publishers P ON B.Publisher = P.PublisherID	2 row(s) returned
189	23:22:10	CREATE TABLE Books_1NF (BookID INT, Title VARCHAR(100), Author VARCHAR(100), Genre VARCHAR(50), Publisher VARCHAR(100), PublisherAddress VARCHAR(100))	Error Code: 1050. Table 'books_1nf' already exists
190	23:23:42	SELECT B.BookID, B.Title, B.Author, B.Genre, P.PublisherName, P.PublisherAddress FROM Books_1NF B JOIN Publishers P ON B.Publisher = P.PublisherID	2 row(s) returned



Additional Questions/Discussions:

- What is a partial dependency, and how does 2NF eliminate it?
 - When a non-key characteristic only depends on a portion of a composite primary key rather than the entire key, this is known as a partial dependency. All non-key attributes must be completely reliant on the entire main key in order to remove partial dependencies using Second Normal Form (2NF). In order to reduce redundancy and improve data integrity, the dependent data is moved into several tables and linked together using foreign keys.
- How do foreign keys help maintain data integrity?
 - Through the enforcement of referential integrity between related tables, foreign keys aid in the maintenance of data integrity in Second Normal Form (2NF). They avoid orphaned records by guaranteeing that a value in a foreign key column must exist in the referenced primary key column of another table. This maintains table relationships, guarantees consistency, and prevents data duplication. A structured and dependable data model is maintained by databases utilizing foreign keys to stop unintentional deletions or updates that can destroy dependencies.

Conclusions:

- In order to remove partial dependencies, Second Normal Form (2NF) makes sure that every non-key attribute depends entirely on the complete primary key. This increases database efficiency, improves data integrity, and decreases redundancy. A better organized and manageable database design is guaranteed by 2NF, which divides related data into distinct tables and makes use of foreign keys.