Field

Customer ID
Customer FirstName
Customer LastName
Customer email ID
Customer Address
Customer City
Customer State
Customer Zip
Order 1 ID
Order 1 Date
Order1 _Cart Value
Order 1_Book_1
Order 1_Book_2
Order 1_Book_Genre_1
Order 1_Book_Genre_2
Order 1_Book_Author_1_First Name
Order 1_Book_Author_1_Last Name
Order 1_Book_Author_2_First Name
Order 1_Book_Author_2_Last Name
Order 1_Quantities
Order 1_Unit Price_1
Order 1_Unit Price_2
Order 2 ID
Order 2 Date
Order2 _Cart Value
Order 2_Book_1
Order 2_Book_2
Order 2_Book_Genre_1
Order 2_Book_Genre_2
Order 2_Book_Author_1_First Name
Order 2_Book_Author_1_Last Name
Order 2_Book_Author_2_First Name
Order 2_Book_Author_2_Last Name
Order 2_Quantities
Order 2_Unit Price_1
Order 2_Unit Price_2
Order 3 ID
Order 3 Date
Order3 _Cart Value
Order 3_Book_1
Order 2_Book_Genre_1
Order 3_Book_Author_1_First Name
Order 3_Book_Author_1_Last Name
Order 2_Quantities
Order 2_Unit Price_1
Book Location ID 1
Book Location Aisle 1
Book Location Row Number 1
Book Location Position 1
Book Location ID 2
Book Location Aisle 2
Book Location Row Number 2
Book Location Position 2
SalesPerson First Name
SalesPerson Last Name
SalesPerson Ph Number

1 NF

Table 1: **Customers**
Customer ID (PK)
Customer FirstName
Customer LastName
Customer email ID
Customer Address
Customer City
Customer State
Customer Zip

Table 2: **Orders**
Order ID (PK)
Customer ID (FK)
Order Date
Order Cart_Value
Order Quantities
SalesPerson First_Name
SalesPerson Last_Name
SalesPerson Ph Number

Table 3: **Books**
Book ID (PK/CPK)
Order ID (CPK/FK)
Book Title
Book Author First Name
Book Author Last Name
Unit Prices
Genre

Table 4: **Location**
Location ID (PK)
Book ID (FK)
Location Aisle
Location Row Number
Location Position

2 NF

Table 1: **Customers**
Customer ID (PK)
Customer FirstName
Customer LastName
Customer email ID
Customer Address
Customer City
Customer State
Customer Zip

Table 2: **Orders**
Order ID (PK)
Customer ID (FK)
Order Date
Order Cart_Value
Order Quantities
SalesPerson First_Name
SalesPerson Last_Name
SalesPerson Ph Number

Table 4: **Book Orders**
Book ID (CPK/ FK)
Order ID (CPK/FK)

Table 3: **Books**
Book ID (PK/CPK)
Book Title
Book Author First Name
Book Author Last Name
Unit Prices
Genre

Table 5: **Location**
Location ID (PK)
Book ID (FK)
Location Aisle
Location Row Number
Location Position

3 NF

Table 1: **Customers**
Customer ID (PK)
Customer FirstName
Customer LastName
Customer email ID
Customer Address
Customer City
Customer State
Customer Zip

Table 2: **Orders**
Order ID (PK)
Customer ID (FK)
Salesperson ID (FK)
Order Date
Order Cart_Value
Order Quantity

Table 6: **Salesperson**
Salesperson ID (PK)
SalesPerson First_Name
SalesPerson Last_Name
SalesPerson Ph Number

Table 4: **Book Orders**
Book ID (CPK/ FK)
Order ID (CPK/FK)

Table 3: **Books**
Book ID (PK/CPK)
Book Title
Book Author First Name
Book Author Last Name
Unit Prices
Genre

Table 5: **Location**
Location ID (PK)
Book ID (FK)
Location Aisle
Location Row Number
Location Position

1 NF (Field): Removing multi-valued attributes
I first looked at the attributes given and selected the ones that didn't have atomic values. For example, the customer address had the street address, the city, the state, and the zip code in it. Therefore, I separated those into different attributes. I did the same for other columns as well such as author name, customer name, salesperson name, book titles, etc.

1 NF: Removing repeating groups
I used the new Field data to remove repeating groups such as orders, books, and book locations. For the orders table, besides including attributes that are part of the orders, I also stored the salesperson information there as well as they have a connection between each other (one to many relationships).
In the orders table, the Customer ID is the foreign key referencing the customers table and maintaining the one-to-many relationship amongst themselves.
In the Books table, Order ID and Book ID are both composite primary keys and a foreign key and primary key, respectively. Books and orders have a many-to-many relationship as one book can be part of many orders and one order can have multiple books.
Lastly, in the location table, because one location has one book and one book can be in multiple locations (one to many relationships), Book ID is a foreign key referencing to the Books table, ensuring the relationship between themselves.

2 NF: Removing partial dependencies
I used the 1NF tables to determine which tables had composite primary keys, to remove partial dependencies. In 2NF, all attributes in a table should solely depend on the primary key of that table. The customers and order tables, so far, had attributes that are functionally dependent on the customers and order IDs respectively.
However, the Books table in 1NF had two composite primary keys (Book ID and Order ID), maintaining a many-to-many relationship. This is where we create a separate linking table that ensures that we are removing the partial dependencies between these two composite primary keys in a single table but maintaining it in the linking table. That way, orders and books table are still connected, and the attributes in the Books table depend only on the Book ID. As for the Location table, there were no composite primary keys and the attributes in that table depended on the location ID. Book ID was referenced as a foreign key to ensure the one-to-many relationship between the Books and the Location table.

3 NF: Removing transitive dependencies
For 3NF, all non-key attributes should solely depend on the primary key of the table and not on any other non-key attribute. To remove the transitive dependency, for each table, I went through the non-key attributes and asked if they were dependent solely on the primary key or not. The customers, books, and location tables' non-key attributes depended on their respective primary keys. However, in the orders table, when checking salesperson information (first name, last name, and phone number), I realized that they were not functionally dependent on the Order ID; they were dependent on each other instead. Therefore, I created a separate table called salesperson. In the salesperson table, I have the salesperson ID as the uniquely identifying primary key. Orders and salesperson tables have a one-to-many relationship (one order can have one salesperson assigned to it, but one salesperson can have multiple orders assigned to them). As a result, we have sales person ID as a foreign key in the orders table ensuring this relationship.