NORMALIZATION REPORT FOR LIBRARY SYSTEM

- Step 1: First Normal Form (1NF)

Rule: No repeating groups or multi-valued attributes.

We separate repeating groups into rows. Now each row has one book per issue record.

IssueRecords (
  User_ID,
  User_Name,
  Phone_number,
  Book_ID,
  Book_Title,
  Author_Name,
  Loans_Date,
  Due_Date
  Return_Date,
  Staff_Name
);
Still redundant, but no multi-valued attributes.

- Step 2: Second Normal Form (2NF)

Rule: Eliminate partial dependencies (non-key attributes depending on part of a composite key).

User_Name, Phone_number depend only on user_ID (not on Book_ID).

Book_Title, Author_Name depend only on Book_ID.

Staff_Name depends only on staff, not on issue.

So we split:

```
Create table User (
   User_ID INT PRIMARY KEY,
   User_Name VARCHAR(100),
   Phone_number VARCHAR(15),
   Membership_date DATE

);

Create table Books (
   Book_ID INT PRIMARY KEY,
   Book_Title VARCHAR(150),
   Author_Name VARCHAR(100)
);

Create table Staff (
   Staff_ID INT PRIMARY KEY,
   Staff_Name VARCHAR(100)
);

Create table loans (
   Loans_ID INT PRIMARY KEY,
   Staff_ID INT,
   Book_ID INT,
   User_ID INT,
   Loans_Date DATE,
   Due_Date DATE,
   Return_Date DATE,
   FOREIGN KEY (user_ID) REFERENCES user(user_ID),
   FOREIGN KEY (Book_ID) REFERENCES Books(Book_ID),
   FOREIGN KEY (satff_ID) REFERENCES staff(staff_ID)
```

);

- Step 3: Third Normal Form (3NF)

Rule: Remove transitive dependencies (non-key attributes depending on other non-key attributes).

Problem: Author_Name depends on Book_ID, but authors themselves could repeat across books.
So we create an Authors table:

```
Create table Authors (
   Author_ID INT PRIMARY KEY,
   Author_Name VARCHAR(100)
);

-- Update Books to reference Author_ID instead of Author_Name
Create table Books (
   Book_ID INT PRIMARY KEY,
   Book_Title VARCHAR(150),
   Author_ID INT,
   FOREIGN KEY (Author_ID) REFERENCES Authors(Author_ID)
);
```

Now Books → AuthorID → AuthorName (no redundancy).