**Assignment 1**

By Harsh Sharma

Part 3

**Part III Logical Data Modeling – Relational Database Design and Normalization**

**Q6.** Translate the (E)ER-diagram from Q5 (1) into a set of relations suitable for the database of “**Musicmatic**”.

Here's the relational schema for the Musicmatic database, derived from the (E)ER-diagram provided:

**Relations:**

1. Artist(**Name**, DateOfBirth, URL)
   * PK: Name
2. Song(**Title**, **ArtistName**, Year, Length, Genre)
   * PK: Title, ArtistName
   * FK: ArtistName references Artist(Name)
3. User(**ID**, **Name**, Address, UserType)
   * PK: ID
   * Required: ID, Name, UserType
4. BusinessUser(**ID**)
   * PK: ID
   * FK: ID references User(ID)
5. RegularUser(**ID**)
   * PK: ID
   * FK: ID references User(ID)
6. Single(**SongID**, **ArtistName**)
   * PK: SongID, ArtistName
   * FK: (SongID, ArtistName) references Song(Title, ArtistName)
7. Album(**AlbumNumber**, Title, ReleaseYear, UserType)
   * PK: AlbumNumber
   * Required: AlbumNumber, UserType
8. Track(**AlbumNumber**, **TrackNumber**, **SongID**, **ArtistName**)
   * PK: AlbumNumber, TrackNumber
   * FK: (SongID, ArtistName) references Song(Title, ArtistName)
   * FK: AlbumNumber references Album(AlbumNumber)
9. AlbumSuggestion(**SuggestionID**, **AlbumNumber**)
   * PK: SuggestionID
   * FK: AlbumNumber references Album(AlbumNumber)

**Q7.** Review the case study on the “**Local Library Management System**”. Based on the details provided, answer the following:

**(1)** Determine the normal form in which the “library” table currently resides.

Solution: To determine the normal form of the given "library" table:

1. Check for Primary Key: The given primary key is (MemberID, BookID) which uniquely identifies each row.
2. Check for Partial Dependencies: There are partial dependencies - Address depends only on MemberID and Author depends only on BookID
3. Check for Transitive Dependencies: There are no transitive dependencies
4. Check if table is in 1NF: Yes, the table meets 1NF requirements as it has atomic values and there are no repeating groups or arrays.
5. Check if table is in 2NF: No, the table is not in 2NF because it has partial dependencies (Address, Author)
6. Check if table is in 3NF: No, since the table is not in 2NF, it cannot satisfy 3NF requirements.

In summary, the library table is in First Normal Form (1NF) but NOT in Second or Third Normal Forms due to the presence of partial dependencies.

Therefore, the library table is currently residing in the First Normal Form (1NF) only.

**(2)** Enumerate all functional dependencies found in the “library” table. Note: In this scenario, a book has only one author.

Solution: To enumerate all functional dependencies found in the "library" table, we need to analyze the relationships between attributes. Based on the provided information that each book has only one author, we can identify the following functional dependencies:

1. MemberID → MemberName, Address
   * Each MemberID uniquely determines the MemberName and Address.
2. BookID → BookTitle, Author, LibrarySection
   * Each BookID uniquely determines the BookTitle, Author, and LibrarySection.
3. (MemberID, BookID) → BorrowDate, ReturnDate, Librarian
   * Each combination of MemberID and BookID uniquely determines the BorrowDate, ReturnDate, and Librarian.
4. Librarian → LibrarySection
   * Each Librarian is associated with a specific LibrarySection.

These functional dependencies help in understanding the relationships between attributes in the "library" table and are crucial for database normalization and integrity.

**(3)** From the functional dependencies, identify any partial dependencies.

Solution: Based on the library table and requirements provided, here are the partial dependencies:

Address -> MemberID

The Address attribute is partially dependent on MemberID. Each member has only one address, but MemberID does not uniquely identify Address, since multiple members can have the same address.

Author -> BookID

The Author attribute is partially dependent on BookID. Each book has one or more authors, but BookID does not uniquely identify Author, since multiple books can have the same author.

So in summary, the partial dependencies are:

Address -> MemberID

Author -> BookID

**(4)** From the functional dependencies, identify any transitive dependencies.

Solution: Based on the library table and requirements, there are no transitive dependencies.

A transitive dependency occurs when A -> B and B -> C, then A -> C by transitivity.

However, in this case, the functional dependencies are:

* MemberID -> MemberName, Address
* BookID -> BookTitle, Author
* (MemberID, BookID) -> BorrowDate, ReturnDate, Librarian, LibrarySection

There are no dependencies of the form A -> B and B -> C, so there are no transitive dependencies.

The dependencies are direct from the candidate keys (MemberID, BookID) to the other non-key attributes, without any intermediate attributes.

So in summary, there are no transitive dependencies identified in the given library table. Let me know if you need any additional clarification!

**(5)** Transform the current “library” relation into relations in the third normal form (3NF). Clearly indicate the primary key for each new relation. Additionally, specify any foreign keys and their referencing table.

Solution: Here is one way to decompose the library table into 3NF relations:

Member(MemberID, MemberName, Address)

Book(BookID, BookTitle, Author)

Author(AuthorID, AuthorName)

Librarian(LibrarianID, LibrarianName)

LibrarySection(SectionID, SectionName)

BorrowEvent(MemberID, BookID, BorrowDate, ReturnDate, LibrarianID, SectionID)

To explain the decomposition:

- Member and Book contain attributes dependent on MemberID and BookID respectively

- Author is extracted out since Author -> BookID is a partial dependency

- Librarian and LibrarySection extracted as they are independent of keys

- BorrowEvent contains all attributes that depend on the candidate key (MemberID, BookID)

Keys:

- Primary keys are underlined

- MemberID and BookID become foreign keys in BorrowEvent

This decomposition removes all partial and transitive dependencies and ensures each relation is in 3NF.