Introduction

This document explains why dictionaries, lists, and tuples were chosen as the primary data structures for the Library Management System and how they work together to create an effective solution.

1. Why Dictionaries?

Book and Member Storage

Each book and member is stored as a dictionary:

```
book = {"isbn": "978-123", "title": "Book Title", "author": "Author Name",
        "genre": "Fiction", "total_copies": 5, "available_copies": 5}


member = {"id": "M001", "name": "John Doe", "email": "john@email.com",
          "contact": "123-456", "borrowed_books": []}
```

Why Dictionaries?

- **Readable Access**: Using book["title"] is clearer than book[0] - the code explains itself
- **Easy Updates**: Can modify specific fields like book["title"] without affecting other properties
- **Flexible Structure**: Adding new fields (like publication year) doesn't require restructuring
- **Real-World Modeling**: Books and members naturally have multiple named properties that dictionaries represent perfectly

2. Why Lists?

Collections and Borrowed Books

```
books = []                # Stores all book dictionaries
members = []                # Stores all member dictionaries
borrowed_books = []         # Inside each member dictionary
```

Why Lists?

- **Dynamic Growth**: Lists automatically expand as books and members are added, essential for a growing library
- **Easy Iteration**: Simple for loops allow searching, displaying, and validating all items

- **Multiple Items**: Members can borrow multiple books - lists handle multiple values naturally
- **Built-in Operations**: append() for adding, remove() for deleting, len() for counting - all operations we need
- **Enforce Limits**: len(borrowed_books) >= 3 makes the 3-book limit easy to check

3. Why Tuples?

Genre Validation

VALID_GENRES = ("Fiction", "Non-Fiction", "Sci-Fi")

Why Tuples?

- **Cannot Be Changed**: Tuples are immutable - valid genres remain fixed throughout the program
- **Data Integrity**: Prevents accidental modification of business rules
- **Clear Intent**: Using a tuple signals this data should never change
- **Fast Validation**: Checking genre in VALID_GENRES is efficient
- **Memory Efficient**: Tuples use less memory than lists for constant data

4. How They Work Together

Example: Borrowing a Book

1. Search the books **list** to find the book
2. Access book details using **dictionary** keys: book["available_copies"]
3. Check genre is valid using the **tuple**: genre in VALID_GENRES