Rationale: Data Structures and Implementation

This library management system uses three main Python data structures: dictionary, list, and tuple. Each was chosen for efficiency, readability, and simplicity.

1. Dictionary (dict) for Books

The dictionary stores books using their ISBN as a unique key.

Each value contains another dictionary with title, author, genre, and total copies.- This structure allows fast lookup, update, and deleting using the ISBN

Example:

books = {

"111": {"title": "Book A", "author": "Author 1", "genre": "Fiction", "total_copies": 3}

}

2. List of Dictionaries for Members

Each member record is a dictionary, and all members are stored in a list.

This allows iteration, searching, and updates while keeping records flexible

Example:

members = [

{"member_id": 1, "name": "Alice", "email": "alice@example.com", "borrowed_books": []}

]

3. Tuple for Genres

Genres rarely change, so they are stored in a tuple an immutable structure

Example:

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

Advantages

- ❖ Fast retrieval using dictionaries.
- ❖ Simple and dynamic storage for members using lists.
- ❖ Data integrity with immutable tuples.

Conclusion

This combination ensures efficient management of library data with minimal complexity. The system supports key CRUD operations (Create, Read, Update, Delete) and borrowing/returning mechanisms efficiently