MINI LIBRARY MANAGEMENT SYSTEM
DESIGN RATIONALE

CHOICE OF DATA STRUCTURES

Dictionary for Books: Books are stored in a dictionary named books, where each ISBN acts as a unique key and the value holds details such as title, author, genre, and copies. This structure was chosen because it allows fast lookups, updates, and deletions using unique identifiers, similar to how real libraries track books through barcodes. It also enables efficient management of available and borrowed copies, making the system scalable and responsive.

List for Members: Members are stored as a list of dictionaries, each containing the member's ID, name, email, and borrowed books. This choice keeps the system simple and easy to manage for small datasets. New members can be easily added, and sequential searches are sufficient for the current scale, ensuring clarity and ease of implementation.

Tuple for Genres: The genres tuple defines a fixed set of book categories such as Fiction, Non-Fiction, Mystrey, Sci-Fi and Biography. A tuple was chosen for its immutability, which prevents accidental modification and ensures data consistency. It also provides a reliable reference for validating book categories during addition or updates.

FUNCTION DESIGN AND REAL-WORLD MODELING

1. **add_book() / add_member()**: These functions register new books and members while enforcing uniqueness checks for ISBNs and member IDs. This mirrors the real-world registration of library assets and patrons.
2. **update_book() / update_member()**: These allow librarians to modify details such as a book title or member name. Validation ensures that updates do not violate data integrity, similar to how real systems track record changes.

3. **search_books()**: This function supports flexible retrieval by title or author, allowing users to quickly find information, similar to a digital catalog search.
4. **borrow_book() and return_book()**: These simulate the process of checking out and returning books. The borrow function decreases the available copies while updating the member's borrowed list. The return function reverses these changes, ensuring accurate record tracking.
5. **delete_book() / delete_member()**: These functions handle removals while ensuring that deletions cannot occur if a book is currently borrowed or if a member still has outstanding loans. This safety mechanism preserves data integrity.

DATA INTEGRITY AND VALIDATION

1. **Unique Identifiers:**
   - Each book's ISBN is unique and must not already exist in the system.
   - Each member's ID is unique, preventing duplicate registrations.
2. **Validation Rules:**
   - Genre entries are checked against the predefined genres tuple.
   - Members cannot borrow more than the set limit (three books).
   - A book cannot be deleted if it has borrowed copies still in circulation.
3. **Automatic Updates:**
   - Borrowing and returning books automatically adjust total_copies, ensuring real-time stock accuracy.
   - Returned books are immediately reflected in the available inventory.
4. **Error Handling:**
   - The system prevents invalid operations such as borrowing unavailable books or deleting active members.