Assignment 1: Database and Collection Setup

1. Using MongoDB Shell, create a database called `SchoolDB`. Within this database, create a collection named `Students`.

Inserting Data and Querying Documents

1. Insert multiple documents into the `Students` collection. Each document should contain the following fields: `name`, `age`, `grade`, and `city`. Add at least five student records with varied data.

Assignment 3: Updating and Deleting Data

1. Update the `Students` collection by modifying specific fields in the documents. Update all records where `city` is `"New York"` to change their `grade` to `"A+"`. Then, delete any document where `age` is less than 10.

Assignment: Modeling Relationships in MongoDB

In this assignment, you will model a database for a Library Management System. You will create collections and documents that demonstrate one-to-one, one-to-many, and many-to-many relationships.

Task 1: One-to-One Relationship

1. Create a one-to-one relationship between the `Members` collection and the `Profiles` collection.

2. Each member in the library has a unique profile containing details like `address`, `phone`, and `email`.

3. Instructions:
   - Create a collection named `Members` with fields `member_id`, `name`, and `profile_id`.
   - Create a collection named `Profiles` with fields `profile_id`, `address`, `phone`, and `email`.
   - Link each member to a profile by using the `profile_id` as a reference.

Task 2: One-to-Many Relationship

1. Create a one-to-many relationship between the `Members` collection and `Loans` collection.

2. Each member can borrow multiple books, but each loan entry should be associated with only one member.

3. **Instructions:**
   - Add a new collection named `Loans` with fields `loan_id`, `member_id`, `book_id`, and `loan_date`.
   - Each document in the `Loans` collection should refer to a `member_id` and a `book_id`.
   - Insert multiple loan records for a single member to demonstrate the one-to-many relationship.

Task 3: Many-to-Many Relationship

1. Create a many-to-many relationship between the `Books` collection and `Authors` collection.

2. A book can be written by multiple authors, and each author may write multiple books.

3. **Instructions:**
   - Create a collection named `Books` with fields `book_id`, `title`, and `author_ids` (an array of IDs referencing authors).
   - Create a collection named `Authors` with fields `author_id` and `name`.
   - Insert multiple books with multiple authors and ensure each author can be associated with multiple books.