Name: V. Charan Kumar

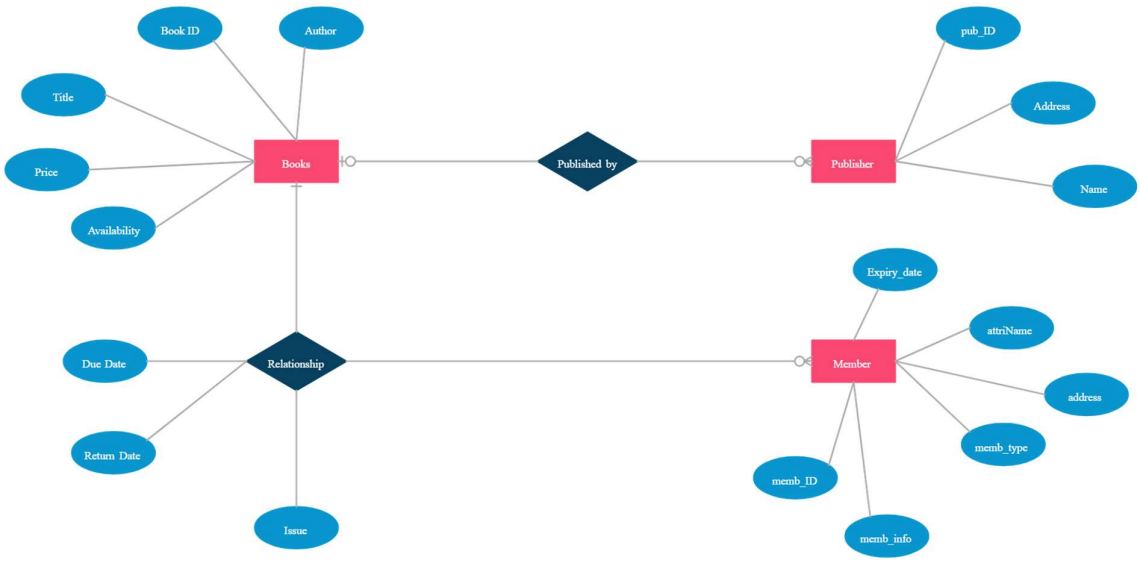
Assignment 1: Analyze a given business scenario and create an ER diagram that includes entities, relationships, attributes, and cardinality. Ensure that the diagram reflects proper normalization up to the third normal form.

1. Table: Authors
    - Fields:
        - author_id (Primary Key)
        - author_name
2. Table: Books
    - Fields:
        - book_id (Primary Key)
        - title
        - author_id (Foreign Key referencing Authors)
        - ISBN (UNIQUE)
        - publication_year
        - genre
3. Table: Membership
    - Fields:
        - member_id (Primary Key)
        - member_name
        - email (UNIQUE)
        - phone_number
4. Table: Book lends
    - Fields:
        - l_id (Primary Key)
        - book_id (Foreign Key referencing Books)
        - member_id (Foreign Key referencing Members)
        - taken_date
        - return_date
        - returned (CHECK: Boolean, default = false)

This schema establishes relationships between the following tables:

- Authors and Books (One-to-Many relationship)
- Books and Book Lends (One-to-Many relationship)
- Members and Book Lends (One-to-Many relationship)

E-R Diagram of Library Management System



Assignment 2: Write SQL statements to CREATE a new database and tables that reflect the library schema you designed earlier. Use ALTER statements to modify the table structures and DROP statements to remove a redundant table.

Queries:

```sql
CREATE DATABASE library;

USE library;

CREATE TABLE authors (
    author_id INT AUTO_INCREMENT PRIMARY KEY,
    author_name VARCHAR(100) NOT NULL
);

CREATE TABLE books (
    book_id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(200) NOT NULL,
    author_id INT,
    publication_year INT,
    FOREIGN KEY (author_id) REFERENCES authors(author_id)
);
```

```sql
CREATE TABLE membership (
    member_id INT PRIMARY KEY,
    member_name VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    phone_number VARCHAR(20) NOT NULL
);

CREATE TABLE book_lends (
    l_id INT PRIMARY KEY,
    book_id INT,
    member_id INT,
    taken_date DATE,
    return_date DATE,
    returned BOOLEAN DEFAULT false,
    FOREIGN KEY (book_id) REFERENCES Books(book_id),
    FOREIGN KEY (member_id) REFERENCES Members(member_id)
);

ALTER TABLE authors
ADD COLUMN author_bio TEXT;

/* replace example with redundant table name*/
DROP TABLE example;
```