Vistula University

Database Design SPEC: Database Engineering

TOPIC: Book Catalogue

Student: Abubakr Mamajonov

ID: 52493

Book Catalogue

1. Assumption(limitations)

1. Assumption

Book Catalogue Purpose: The assumption is that the Book Catalogue project aims to create a database system that allows users to store and manage information about books, including their titles, authors, genres, publication dates, and availability.

Single User: The assumption is that the Book Catalogue will be designed for a single user or a small group of users, rather than a large-scale multi-user system.

Non-Transactional System: The assumption is that the Book Catalogue is primarily a read-intensive system, and transactional operations, such as buying or borrowing books, are not within the scope of this project.

Focus on Metadata: The assumption is that the Book Catalogue will focus on storing and managing metadata about books, such as title, author, and genre, rather than the actual content of the books themselves.

2. Limitations

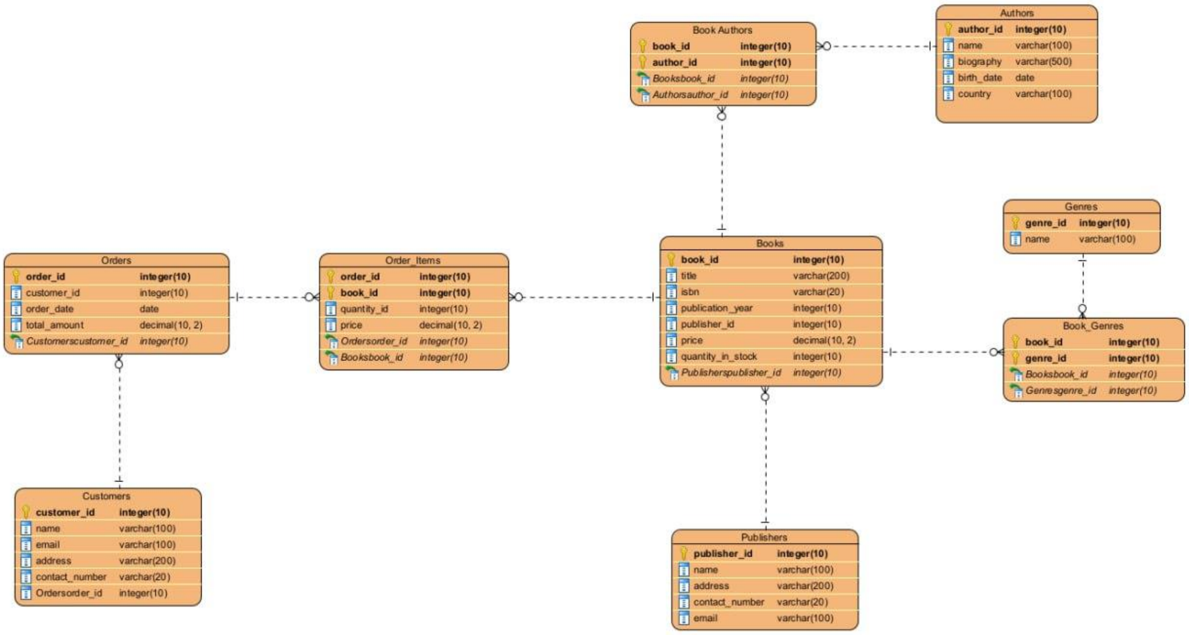
Scalability: The limitation is that the Book Catalogue is designed for a relatively small number of books and users. It may not be suitable for large-scale libraries or organizations with extensive collections.

Security: The limitation is that the project does not focus on implementing advanced security measures, such as user authentication, access control, or encryption. These aspects may need to be considered separately if the system is intended for production use.

Performance Optimization: The limitation is that the project does not include extensive performance optimization techniques, such as indexing or caching, to handle large volumes of data or high traffic. The emphasis is on basic functionality and database design principles.

Integration with External Systems: The limitation is that the project does not address the integration of the Book Catalogue with external systems, such as online bookstores or library management systems. This project assumes standalone functionality.

2. ERD Diagram



3. Table Structure

```
-- Create Authors table
CREATE TABLE Authors (
   author_id INT PRIMARY KEY,
   name VARCHAR(100),
   biography VARCHAR(500),
   birth_date DATE,
   country VARCHAR(100));

-- Create Publishers table
CREATE TABLE Publishers (
   publisher_id INT PRIMARY KEY,
   name VARCHAR(100),
   address VARCHAR(200),
   contact_number VARCHAR(20),
   email VARCHAR(100));

-- Create Books table
CREATE TABLE Books (
   book_id INT PRIMARY KEY,
   title VARCHAR(200),
   isbn VARCHAR(20),
   publication_year INT,
   publisher_id INT,
   price DECIMAL(10, 2),
   quantity_in_stock INT,
   FOREIGN KEY (publisher_id) REFERENCES Publishers(publisher_id));

-- Create Genres table
CREATE TABLE Genres (
   genre_id INT PRIMARY KEY,
   name VARCHAR(100));

-- Create Books_Genres table (Many-to-Many relationship between Books
and Genres)
CREATE TABLE Books_Genres (
   book_id INT,
   genre_id INT,
   FOREIGN KEY (book_id) REFERENCES Books(book_id),
   FOREIGN KEY (genre_id) REFERENCES Genres(genre_id),
   PRIMARY KEY (book_id, genre_id));
```

```
-- Create Book_Authors table (Many-to-Many relationship between Books
and Authors)
CREATE TABLE Book_Authors (
   book_id INT,
   author_id INT,
   FOREIGN KEY (book_id) REFERENCES Books(book_id),
   FOREIGN KEY (author_id) REFERENCES Authors(author_id),
   PRIMARY KEY (book_id, author_id));

-- Create Customers table
CREATE TABLE Customers (
   customer_id INT PRIMARY KEY,
   name VARCHAR(100),
   email VARCHAR(100),
   address VARCHAR(200),
   contact_number VARCHAR(20));

-- Create Orders table
CREATE TABLE Orders (
   order_id INT PRIMARY KEY,
   customer_id INT,
   order_date DATE,
   total_amount DECIMAL(10, 2),
   FOREIGN KEY (customer_id) REFERENCES Customers(customer_id));

-- Create Order_Items table (Many-to-Many relationship between Orders
and Books)
CREATE TABLE Order_Items (
   order_id INT,
   book_id INT,
   quantity INT,
   price DECIMAL(10, 2),
   FOREIGN KEY (order_id) REFERENCES Orders(order_id),
   FOREIGN KEY (book_id) REFERENCES Books(book_id),
    PRIMARY KEY (order_id, book_id));
```

4. Critical evaluation of the completed project

The completed book catalogue database project has been designed to effectively store and manage information related to books, including their titles, authors, publication details, and availability. Overall, the project demonstrates several strengths and areas for improvement in terms of its design and functionality.

Strengths:

Database Structure: The database is structured well, with appropriate tables and relationships established between them. This allows for efficient data retrieval and avoids data redundancy.

Data Integrity: The project incorporates various integrity constraints, such as primary keys, foreign keys, and unique constraints, to ensure the accuracy and consistency of the data. This helps maintain data integrity and prevents inconsistencies or errors.

Search and Retrieval: The search functionality implemented in the project is effective in retrieving relevant books based on various criteria, such as title, author, genre, and publication year. The database queries are optimized, resulting in fast and accurate search results.

Availability Tracking: The project includes a mechanism to track the availability of books, which is crucial for managing inventory and facilitating the borrowing process. This feature allows users to easily determine if a book is currently available or on loan.

Areas for Improvement:

User Interface: The project would benefit from a user-friendly interface that enhances the user experience. While the database functionality is robust, a well-designed and intuitive interface would make it easier for users to interact with the system.

Data Validation: Although integrity constraints are in place, additional validation checks could be implemented to ensure data entered into the system is accurate and consistent. For example, validating the format of ISBN numbers or enforcing constraints on publication dates.

Data Security: The project should incorporate appropriate security measures to protect the database from unauthorized access or data breaches. Implementing user authentication and authorization mechanisms would help ensure data confidentiality and integrity.

Scalability and Performance: As the book catalogue database grows over time, it is important to consider its scalability and performance. Optimizing queries, indexing frequently accessed columns, and regularly monitoring and tuning the database will help maintain efficient performance as the system handles larger volumes of data.

Integration with External Systems: In a real-world scenario, the book catalogue database might need to integrate with external systems, such as library management software or online platforms. Consideration should be given to designing an interface or API that facilitates seamless data exchange and integration with these systems.

Error Handling and Logging: Implementing comprehensive error handling and logging mechanisms would enable effective troubleshooting and debugging of the system. Properly logging errors and exceptions will assist in identifying and resolving issues promptly.

Backup and Recovery: Regular backups of the database should be implemented to safeguard against data loss in the event of system failures or disasters. Establishing a backup strategy and testing the recovery process will help ensure the availability and integrity of the data.

In conclusion, the completed book catalogue database project showcases various strengths in terms of its structure, data integrity, search functionality, and availability tracking. However, there are areas for improvement, such as enhancing the user interface, implementing additional data validation, addressing data security concerns, optimizing performance, enabling integration with external systems, improving error handling and logging, and establishing a backup and recovery strategy. By addressing these areas, the project can further enhance its usability, reliability, and scalability, providing an efficient and effective book catalogue management solution.

5. **Project Documentation**

Introduction:

The Book Catalogue Database is designed to efficiently manage information related to books, authors, publishers, genres, customers, orders, and order items. The database allows for easy tracking of book availability, customer orders, and provides essential functionalities for effective book catalog management.

Entity-Relationship Diagram (ERD):

The ERD for the Book Catalogue Database consists of the following entities and relationships:

Authors: Represents the authors of the books.

Publishers: Represents the publishers of the books.

Books: Stores information about individual books, including title, ISBN, publication year, publisher, price, and quantity in stock.

Genres: Contains different genres or categories of books.

Customers: Stores information about customers who place orders.

Orders: Represents individual customer orders, including order date and total amount.

Order_Items: Stores the details of each book item included in an order.

Database Design:

The database is designed using the following tables:

Authors: Contains author details, including author ID, name, biography, birth date, and country.

Publishers: Stores publisher details, including publisher ID, name, address, contact number, and email.

Books: Stores book information, including book ID, title, ISBN, publication year, publisher ID (foreign key reference), price, and quantity in stock.

Genres: Contains genre details, including genre ID and name.
Books_Genres: Implements a many-to-many relationship between books and genres, linking book IDs and genre IDs.

Book_Authors: Represents the many-to-many relationship between books and authors, linking book IDs and author IDs.

Customers: Stores customer information, including customer ID, name, email, address, and contact number.

Orders: Represents customer orders, including order ID, customer ID (foreign key reference), order date, and total amount.

Order_Items: Stores the details of individual book items within an order, including order ID, book ID (foreign key reference), quantity, and price.

Table Relationships:

The following relationships are established between the tables:

Books to Publishers: Books table references the publisher ID from the Publishers table using a foreign key constraint.
Books to Genres: Many-to-many relationship implemented through the Books_Genres table, linking book IDs and genre IDs.
Books to Authors: Many-to-many relationship implemented through the Book_Authors table, linking book IDs and author IDs.
Orders to Customers: Orders table references the customer ID from the Customers table using a foreign key constraint.
Orders to Books: Many-to-many relationship implemented through the Order_Items table, linking order IDs and book IDs.

Project Implementation:

The database design can be implemented using a suitable relational database management system (RDBMS) such as MySQL, PostgreSQL, or Microsoft SQL Server. SQL scripts provided above can be executed to create the necessary tables and establish the relationships.

Future Improvements:
To further enhance the functionality and usability of the Book Catalogue Database, the following improvements can be considered:

Integration with external systems: Enable integration with library management software or online platforms for seamless data exchange.

User interface enhancements: Develop a user-friendly interface to improve user experience and ease of interaction with the database.

Additional validation checks: Implement more robust data validation to ensure accuracy and consistency of entered data.

Data security measures: Incorporate user authentication and authorization mechanisms to protect data from unauthorized access.

Performance optimization: Regularly monitor and optimize database performance to handle larger data volumes efficiently.

Error handling and logging: Implement comprehensive error handling and logging mechanisms for effective troubleshooting and issue resolution.

Backup and recovery strategy: Establish a backup strategy to safeguard against data loss and ensure data availability in case of failures.