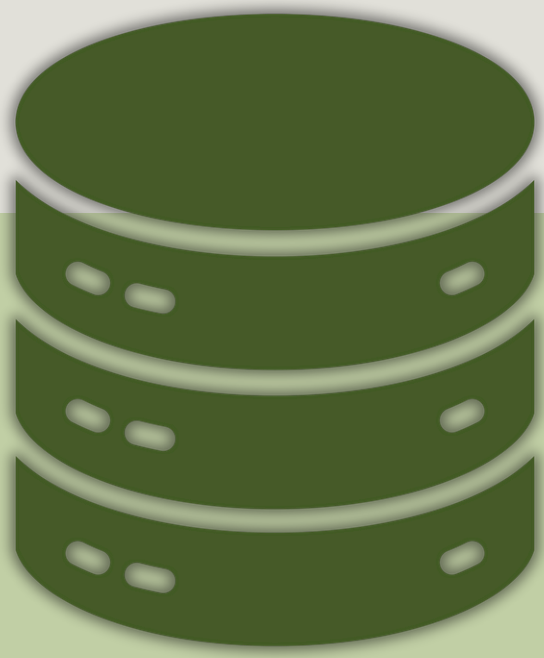
BinaryBytes
BSCS

DATABASE PROJECT

Online Book Store



Written By
Ahmad Hassan

Team Member

Ahmad Hassan (2023-uam-1926)
Muhammad Moeen (2023-uam-1907)
Muaz Islam Babar (2023-uam-1920)

Project Description

This project involves creating a backend database system in **MS SQL Server** to support the core functionalities of an online bookstore. It includes managing users (admins and customers), handling book inventories, processing orders and payments, managing customer carts, categorizing books, and collecting customer feedback via reviews.

Project Objective

This project involves creating a backend database system in **MS SQL Server** to support the core functionalities of an online bookstore. It includes managing users (admins and customers), handling book inventories, processing orders and payments, managing customer carts, categorizing books, and collecting customer feedback via review

Tables - Schema

1. User Table

Stores information about users including their name, email, address, role (admin/customer), and registration timestamp. It serves as the base for all user-related activities such as placing orders or writing reviews.

Column	Data Type	Constraints
UserID	INT	PRIMARY KEY, IDENTITY(1,1)
Name	VARCHAR(100)	NOT NULL
Email	VARCHAR(100)	UNIQUE, NOT NULL
Password	VARCHAR(255)	NOT NULL
Phone	VARCHAR(15)	
Address	TEXT	
Role	VARCHAR(10)	CHECK (Role IN ('Customer', 'Admin'))
CreatedAt	DATETIME	DEFAULT GETDATE()

2. Authors Table

Contains detailed information about each book such as title, author, description, price, stock, and category. Each book is linked to a category and can appear in orders, reviews, and carts.

Column	Data Type	Constraints
AuthorID	INT	PRIMARY KEY, IDENTITY(1,1)
Name	VARCHAR(100)	NOT NULL
Bio	TEXT	

3. Publishers Table

Defines categories like Fiction, Science, History to group books for better browsing and filtering. Each book must belong to one category.

Column	Data Type	Constraints
PublisherID	INT	PRIMARY KEY, IDENTITY(1,1)
Name	VARCHAR(100)	NOT NULL
Website	VARCHAR(255)	
ContactInfo	TEXT	

4. Categories Table

Tracks which users have added which books to their shopping cart, including quantity. Helps simulate a real-time cart system before the user checks out and places an order.

Column	Data Type	Constraints
CategoryID	INT	PRIMARY KEY, IDENTITY(1,1)
CategoryName	VARCHAR(100)	NOT NULL

5. Books Table

Stores data about all completed purchases, including user info, date, total amount, and status. Each order can include multiple books, and links to payments and order items.

Column	Data Type	Constraints
BookID	INT	PRIMARY KEY, IDENTITY(1,1)
Title	VARCHAR(200)	NOT NULL
AuthorID	INT	FOREIGN KEY REFERENCES Authors(AuthorID)
PublisherID	INT	FOREIGN KEY REFERENCES Publishers(PublisherID)
Description	TEXT	
ISBN	VARCHAR(20)	UNIQUE

Column	Data Type	Constraints
Price	DECIMAL(10,2)	NOT NULL
Stock	INT	NOT NULL
CategoryID	INT	FOREIGN KEY REFERENCES Categories(CategoryID)
ImageURL	VARCHAR(255)	

6. Cart Table

Breaks down each order into individual items — books, quantity, and price at time of purchase. Acts as a junction table between Orders and Books to normalize many-to-many relationships.

Column	Data Type	Constraints
CartID	INT	PRIMARY KEY, IDENTITY(1,1)
UserID	INT	FOREIGN KEY REFERENCES Users(UserID)
BookID	INT	FOREIGN KEY REFERENCES Books(BookID)
Quantity	INT	NOT NULL

7. Orders Table

Stores user-submitted ratings and comments for books. Helps future users decide whether to buy a particular book based on previous experiences.

Column	Data Type	Constraints
OrderID	INT	PRIMARY KEY, IDENTITY(1,1)
UserID	INT	FOREIGN KEY REFERENCES Users(UserID)
OrderDate	DATETIME	DEFAULT GETDATE()
TotalAmount	DECIMAL(10,2)	NOT NULL
Status	VARCHAR(50)	CHECK (Status IN ('Pending', 'Shipped', 'Delivered'))

8. OrderItems Table

Keeps record of all payments made, including method (Cash/Card), amount, status, and linked order. Ensures financial tracking of all completed and pending transactions.

Column	Data Type	Constraints
OrderItemID	INT	PRIMARY KEY, IDENTITY(1,1)
OrderID	INT	FOREIGN KEY REFERENCES Orders(OrderID)

Column	Data Type	Constraints
BookID	INT	FOREIGN KEY REFERENCES Books(BookID)
Quantity	INT	NOT NULL
PriceAtPurchase	DECIMAL(10,2)	NOT NULL

9. Payments Table

Stores information about book publishers like name, address, and contact details. Each book can be linked to a publisher for credibility and reference.

Column	Data Type	Constraints
PaymentID	INT	PRIMARY KEY, IDENTITY(1,1)
OrderID	INT	FOREIGN KEY REFERENCES Orders(OrderID)
PaymentDate	DATETIME	
Amount	DECIMAL(10,2)	
PaymentMethod	VARCHAR(50)	
PaymentStatus	VARCHAR(50)	

10. Review Table

Stores detailed information about authors, including their biography and contact info. Each book is associated with an author via foreign keys for proper attribution.

Column	Data Type	Constraints
ReviewID	INT	PRIMARY KEY, IDENTITY(1,1)
UserID	INT	FOREIGN KEY REFERENCES Users(UserID)
BookID	INT	FOREIGN KEY REFERENCES Books(BookID)
Rating	INT	CHECK (Rating BETWEEN 1 AND 5)
Comment	TEXT	
ReviewDate	DATETIME	DEFAULT GETDATE()

Relationships ER Diagram

---

1. Users Table

- One-to-Many with:
  - `Orders` → One User can place multiple Orders.

- `Cart` → One User can have multiple Cart entries.
- `Reviews` → One User can give multiple Reviews.

---

2. Authors Table

- One-to-Many with:
  - `Books` → One Author can write multiple Books.

---

3. Publishers Table

- One-to-Many with:
  - `Books` → One Publisher can publish multiple Books.

---

4. Categories Table

- One-to-Many with:
  - `Books` → One Category can have multiple Books.

---

5. Books Table

- Many-to-One with:
  - `Authors` (via `AuthorID`)
  - `Publishers` (via `PublisherID`)
  - `Categories` (via `CategoryID`)
- One-to-Many with:
  - `Cart` → A Book can be in multiple users' carts.
  - `OrderItems` → A Book can be part of many order items.
  - `Reviews` → A Book can have many reviews.

---

6. Cart Table

- Many-to-One with:
  - `Users` (via `UserID`)
  - `Books` (via `BookID`)

---

7. Orders Table

- Many-to-One with:
  - `Users` (via `UserID`)
- One-to-Many with:
  - `OrderItems` → One Order can have many OrderItems.
  - `Payments` → One Order can have one Payment record.

---

8. OrderItems Table

- Many-to-One with:
  - `Orders` (via `OrderID`)
  - `Books` (via `BookID`)

---

9. Payments Table

- One-to-One or One-to-Many with:
  - `Orders` (depending on design; here, assume One-to-One)

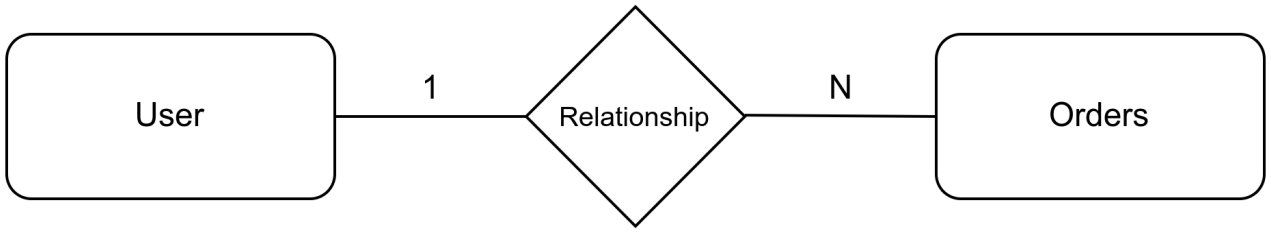
---

10. Reviews Table

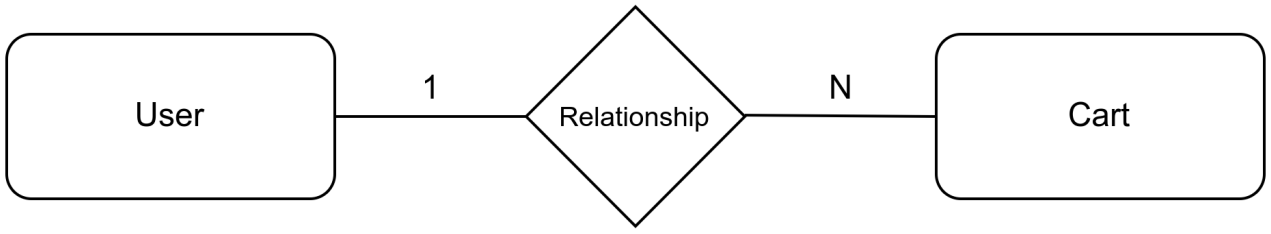
- Many-to-One with:
  - `Users` (via `UserID`)
  - `Books` (via `BookID`)

Cardinalities Diagram

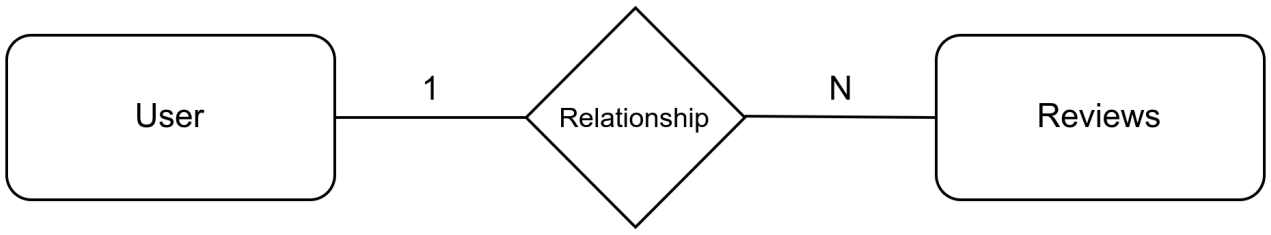
1. User → Orders



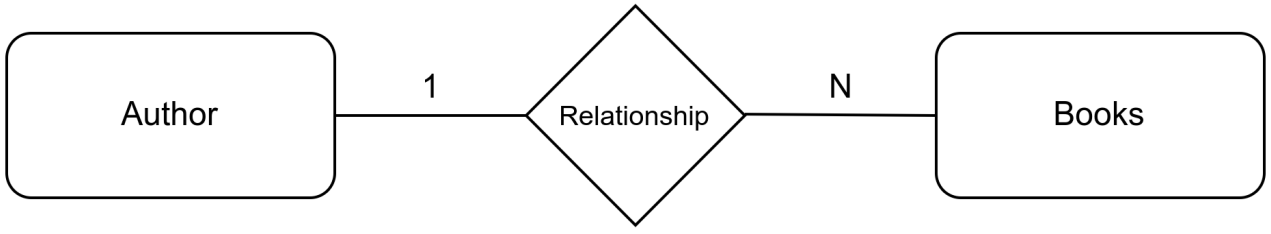
2. User → Cart



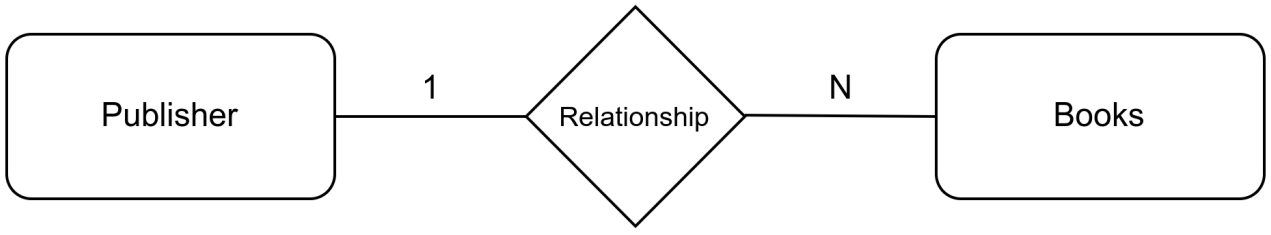
3. User → Reviews



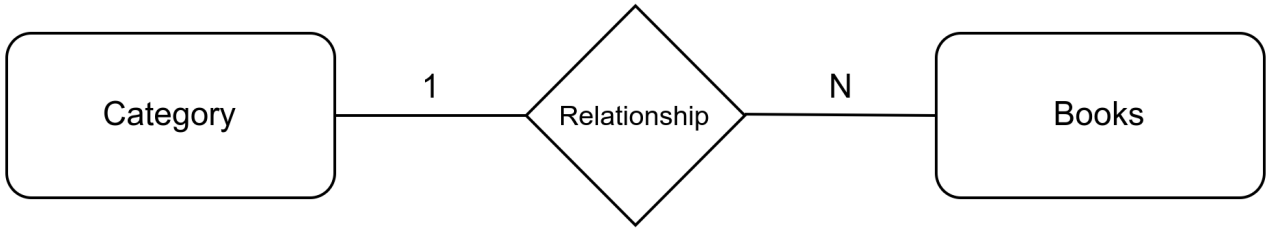
4. Author → Books



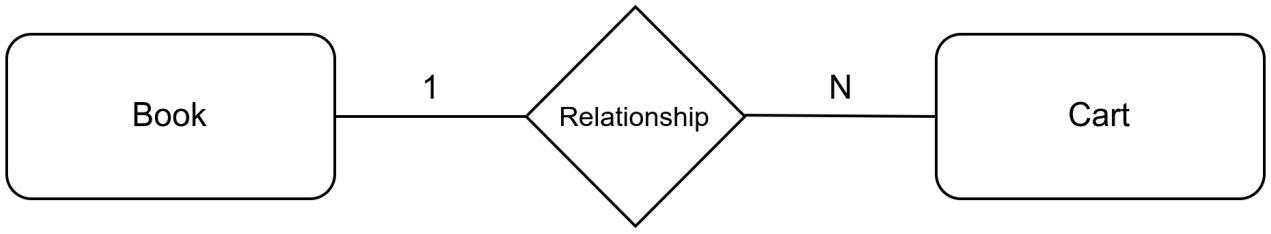
5. Publisher → Books



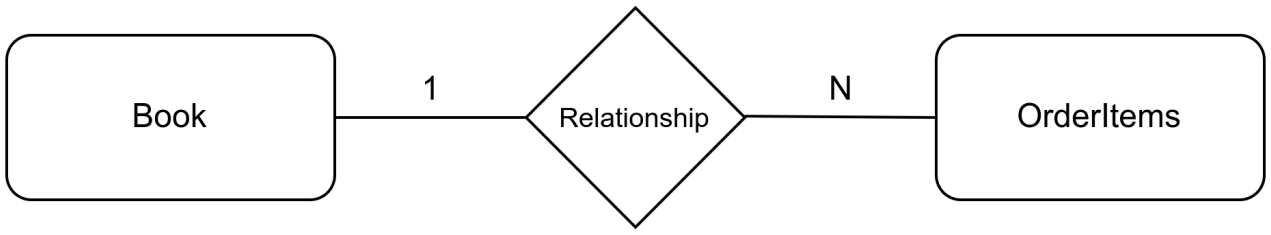
6. Category → Books



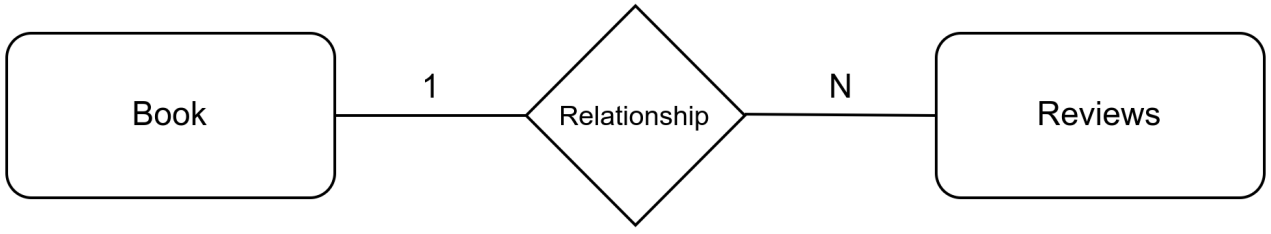
7. Book → Cart



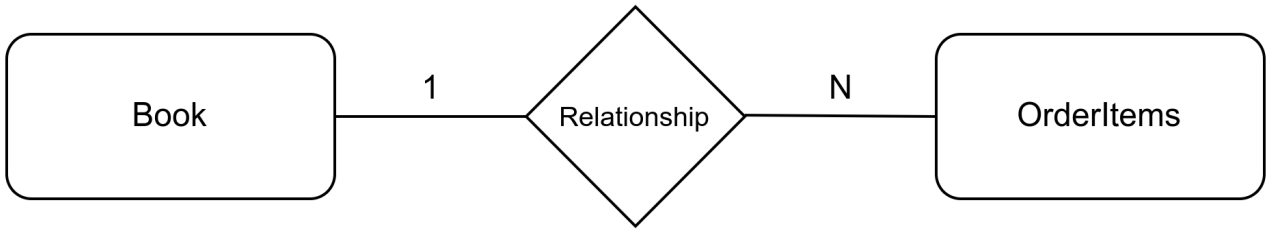
8. Book → OrderItems



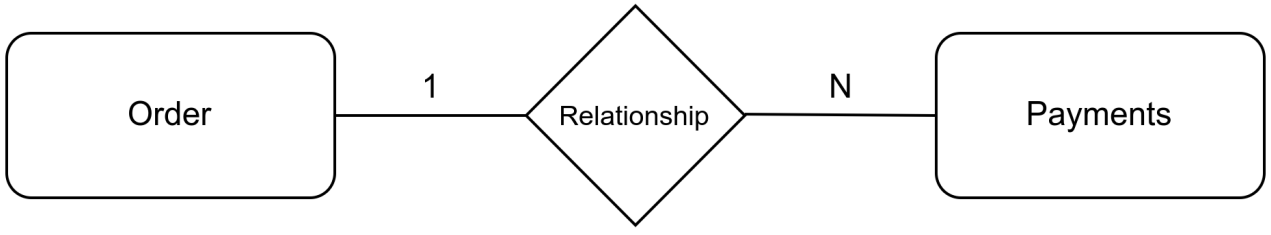
9. Book → Reviews



10. Order → OrderItems



11. Order → Payments



Full Project Workflow