# Case Study: Online Bookstore Database

**Scenario:**

This is an online bookstore's database. The database contains several tables that store information about books, authors, customers, orders, and order details.

Table Creation Commands:

1. **Authors Table**:

```sql
CREATE TABLE Authors (
    author_id INT AUTO_INCREMENT PRIMARY KEY,
    author_name VARCHAR(100) NOT NULL,
    country VARCHAR(50)
);
```

2. **Books Table**:

```sql
CREATE TABLE Books (
    book_id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(200) NOT NULL,
    author_id INT,
    price DECIMAL(10, 2) NOT NULL,
    publication_year INT,
    FOREIGN KEY (author_id) REFERENCES Authors(author_id)
);
```

3. **Customers Table**:

```sql
CREATE TABLE Customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_name VARCHAR(100) NOT NULL,
    email VARCHAR(100),
    join_date DATE
);
```

4. **Orders Table**:

```sql
CREATE TABLE Orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    order_date DATE,
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);
```

5. **Order_Details Table**:

```
CREATE TABLE Order_Details (
    order_detail_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT,
    book_id INT,
    quantity INT NOT NULL,
    subtotal DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (book_id) REFERENCES Books(book_id)
);
```

---

## Explanation of Columns and Constraints:

1. **Authors Table**:
    - `author_id`: Unique identifier for each author (Primary Key).
    - `author_name`: Name of the author.
    - `country`: Country of the author.
2. **Books Table**:
    - `book_id`: Unique identifier for each book (Primary Key).
    - `title`: Title of the book.
    - `author_id`: Foreign key referencing the `author_id` in the `Authors` table.
    - `price`: Price of the book (stored as a decimal).
    - `publication_year`: Year the book was published.
3. **Customers Table**:
    - `customer_id`: Unique identifier for each customer (Primary Key).
    - `customer_name`: Name of the customer.
    - `email`: Email address of the customer.
    - `join_date`: Date the customer joined the bookstore.
4. **Orders Table**:
    - `order_id`: Unique identifier for each order (Primary Key).
    - `customer_id`: Foreign key referencing the `customer_id` in the `Customers` table.
    - `order_date`: Date the order was placed.
5. **Order_Details Table**:
    - `order_detail_id`: Unique identifier for each order detail (Primary Key).
    - `order_id`: Foreign key referencing the `order_id` in the `Orders` table.

- book_id: Foreign key referencing the book_id in the Books table.
- quantity: Number of copies of the book ordered.
- subtotal: Total cost for this line item (quantity * price).

---

## Sample Data Insertion Commands:

If you want to insert the sample data provided earlier, you can use the following commands:

1. **Insert into Authors**:

```
INSERT INTO Authors (author_id, author_name, country) VALUES
(1, 'J.K. Rowling', 'UK'),
(2, 'George R.R. Martin', 'USA'),
(3, 'Haruki Murakami', 'Japan');
```

2. **Insert into Books**:

```
INSERT INTO Books (book_id, title, author_id, price, publication_year)
VALUES
(1, 'Harry Potter and the Philosopher\'s Stone', 1, 20.99, 1997),
(2, 'A Game of Thrones', 2, 25.99, 1996),
(3, 'Norwegian Wood', 3, 15.99, 1987);
```

3. **Insert into Customers**:

```
INSERT INTO Customers (customer_id, customer_name, email, join_date) VALUES
(1, 'Alice Johnson', 'alice@example.com', '2020-01-15'),
(2, 'Bob Smith', 'bob@example.com', '2019-05-20');
```

4. **Insert into Orders**:

```
INSERT INTO Orders (order_id, customer_id, order_date) VALUES
(1, 1, '2023-01-10'),
(2, 2, '2023-02-15');
```

5. **Insert into Order_Details**:

```
INSERT INTO Order_Details (order_detail_id, order_id, book_id, quantity,
subtotal) VALUES
(1, 1, 1, 2, 41.98),
(2, 1, 3, 1, 15.99),
(3, 2, 2, 1, 25.99);
```

Notes:

- The `AUTO_INCREMENT` keyword ensures that the primary keys (`author_id`, `book_id`, `customer_id`, `order_id`, `order_detail_id`) are automatically generated.
- Foreign keys enforce referential integrity between tables.
- The `DECIMAL(10, 2)` data type is used for monetary values to ensure precision.

Let me know if you need further assistance! 😊