Database Schema Design

Below is the complete database schema design for the Online Bookstore Database, including SQL scripts for creating tables, relationships, constraints, and indexes. This design ensures data integrity, performance optimization, and alignment with industry standards.

SQL Scripts for Table Creation

1. authors Table

```
CREATE TABLE authors (
    author_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);
```

2. books Table

```
CREATE TABLE books (
    book_id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    stock_quantity INT DEFAULT 0,
    author_id INT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (author_id) REFERENCES authors(author_id)
        ON UPDATE CASCADE
        ON DELETE SET NULL
);
```

3. customers Table

```
CREATE TABLE customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    phone VARCHAR(15) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);
```

4. orders Table

```
CREATE TABLE orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    order_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    total_amount DECIMAL(10, 2) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
        ON UPDATE CASCADE
        ON DELETE RESTRICT
);
```

5. order_details Table

```
CREATE TABLE order_details (
    order_details_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT,
    book_id INT,
    quantity INT NOT NULL CHECK (quantity > 0),
    line_total DECIMAL(10, 2),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (order_id) REFERENCES orders(order_id) ON DELETE CASCADE,
    FOREIGN KEY (book_id) REFERENCES books(book_id) ON DELETE RESTRICT
);
```

Indexes

- Index on `email` columns for faster searches.

- Index on `customer_id` and `author_id` for frequent joins.

```
CREATE INDEX idx_author_email ON authors(email);
CREATE INDEX idx_customer_email ON customers(email);
CREATE INDEX idx_order_customer ON orders(customer_id);
CREATE INDEX idx_book_author ON books(author_id);
```

Cascading and Restricting Rules

- **Authors to Books**:
  - On Update: Cascade changes in `author_id`.
  - On Delete: Set `author_id` to NULL in `books`.
- **Customers to Orders**:
  - On Update: Cascade changes in `customer_id`.
  - On Delete: Restrict deletion if the customer has orders.
- **Orders to OrderDetails**:
  - On Update: Cascade changes in `order_id`.
  - On Delete: Cascade to delete all associated order details.
- **Books to OrderDetails**:
  - On Update: Cascade changes in `book_id`.
  - On Delete: Restrict deletion if the book is part of an order.

Data Validation Constraints

- **Books Table**:
  - `price` must be greater than 0.
  - `stock_quantity` cannot be negative.
- **OrderDetails Table**:
  - `quantity` must be greater than 0.

Sample Data Insertion

Insert Sample Authors

```
INSERT INTO authors (name, email) VALUES
('J.K. Rowling', 'jkrowling@example.com'),
('George R.R. Martin', 'grrm@example.com');
```

Insert Sample Books

```
INSERT INTO books (title, price, stock_quantity, author_id) VALUES
('Harry Potter and the Philosopher\'s Stone', 29.99, 50, 1),
('A Game of Thrones', 39.99, 30, 2);
```

Insert Sample Customers

```
INSERT INTO customers (name, email, phone) VALUES
('Alice Smith', 'alice@example.com', '1234567890'),
('Bob Johnson', 'bob@example.com', '9876543210');
```

Insert Sample Orders

```
INSERT INTO orders (customer_id, total_amount) VALUES
(1, 59.98),
(2, 39.99);
```

Insert Sample Order Details

```
INSERT INTO order_details (order_id, book_id, quantity, line_total) VALUES
(1, 1, 2, 59.98),
(2, 2, 1, 39.99);
```

Next Steps

1. **Testing**: Perform validation and performance tests.
   - Test cascading and restricting actions.
   - Validate data integrity by attempting invalid operations.
2. **Optimization**: Analyze the performance of queries and add more indexes if necessary.
3. **Document Testing Scenarios**: Define tests for relationships, constraints, and data retrieval.