Case Study: Online Bookstore Database

Scenario: SQL queries to retrieve specific information using joins and sub-queries.

SQL Queries to Practice:

1. **Retrieve all books along with their author's name and country:**

```
SELECT b.title, a.author_name, a.country
FROM Books b
JOIN Authors a ON b.author_id = a.author_id;
```

2. **Find the total revenue generated by each author:**

```
SELECT a.author_name, SUM(od.subtotal) AS total_revenue
FROM Authors a
JOIN Books b ON a.author_id = b.author_id
JOIN Order_Details od ON b.book_id = od.book_id
GROUP BY a.author_name;
```

3. **List all customers who have placed an order, along with the total amount they have spent:**

```
SELECT c.customer_name, SUM(od.subtotal) AS total_spent
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
JOIN Order_Details od ON o.order_id = od.order_id
GROUP BY c.customer_name;
```

4. **Find the book with the highest price:**

```
SELECT title, price
FROM Books
WHERE price = (SELECT MAX(price) FROM Books);
```

5. **Retrieve all books that have never been ordered:**

```
SELECT b.title
FROM Books b
LEFT JOIN Order_Details od ON b.book_id = od.book_id
WHERE od.order_id IS NULL;
```

6. **Find the customer who has spent the most money:**

```
SELECT c.customer_name, SUM(od.subtotal) AS total_spent
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
JOIN Order_Details od ON o.order_id = od.order_id
GROUP BY c.customer_name
ORDER BY total_spent DESC
LIMIT 1;
```

7. **List all authors who have not sold any books:**

```
SELECT a.author_name
FROM Authors a
LEFT JOIN Books b ON a.author_id = b.author_id
LEFT JOIN Order_Details od ON b.book_id = od.book_id
WHERE od.order_id IS NULL;
```

8. **Find the total number of books sold by each author:**

```
SELECT a.author_name, SUM(od.quantity) AS total_books_sold
FROM Authors a
JOIN Books b ON a.author_id = b.author_id
JOIN Order_Details od ON b.book_id = od.book_id
GROUP BY a.author_name;
```

9. **Retrieve the details of the most recent order, including the customer's name and the books ordered:**

```
SELECT c.customer_name, b.title, od.quantity, od.subtotal
FROM Orders o
JOIN Customers c ON o.customer_id = c.customer_id
JOIN Order_Details od ON o.order_id = od.order_id
JOIN Books b ON od.book_id = b.book_id
WHERE o.order_date = (SELECT MAX(order_date) FROM Orders);
```

10. **Find the average price of books published each year:**

```
SELECT publication_year, AVG(price) AS average_price
FROM Books
GROUP BY publication_year;
```