MySQL Joins and Subqueries - Practice Exercises

Database Schema: Online Retail Store

Below is a schema with sample data for an online retail database. Use this to practice various MySQL joins and subqueries.

Create Database Schema

```sql
CREATE DATABASE online_retail;
USE online_retail;

-- Customers table
CREATE TABLE customers (
    customer_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    city VARCHAR(50),
    registration_date DATE
);

-- Categories table
CREATE TABLE categories (
    category_id INT PRIMARY KEY,
    category_name VARCHAR(50),
    description TEXT
);

-- Products table
CREATE TABLE products (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    category_id INT,
    unit_price DECIMAL(10, 2),
    stock_quantity INT,
    FOREIGN KEY (category_id) REFERENCES categories(category_id)
);

-- Orders table
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    order_date DATE,
    total_amount DECIMAL(12, 2),
    status VARCHAR(20),
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);

-- Order Details table
CREATE TABLE order_details (
    order_detail_id INT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT,
    unit_price DECIMAL(10, 2),
    FOREIGN KEY (order_id) REFERENCES orders(order_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);

-- Employees table
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    hire_date DATE,
    supervisor_id INT,
    FOREIGN KEY (supervisor_id) REFERENCES employees(employee_id)
);
```

Insert Sample Data

```sql
-- Insert data into customers
INSERT INTO customers VALUES
(1, 'John', 'Smith', 'john.smith@email.com', 'New York', '2022-01-15'),
(2, 'Jane', 'Doe', 'jane.doe@email.com', 'Los Angeles', '2022-02-20'),
(3, 'Robert', 'Johnson', 'robert.j@email.com', 'Chicago', '2022-03-10'),
(4, 'Maria', 'Garcia', 'maria.g@email.com', 'Miami', '2022-04-05'),
(5, 'David', 'Brown', 'david.b@email.com', 'Boston', '2022-05-12'),
(6, 'Lisa', 'Wilson', 'lisa.w@email.com', 'Seattle', '2022-06-18'),
(7, 'Michael', 'Taylor', 'michael.t@email.com', 'Denver', '2022-07-22');

-- Insert data into categories
INSERT INTO categories VALUES
(1, 'Electronics', 'Electronic devices and accessories'),
(2, 'Clothing', 'Apparel and fashion items'),
(3, 'Books', 'Books and publications'),
(4, 'Home & Kitchen', 'Home appliances and kitchen items'),
(5, 'Sports', 'Sports equipment and accessories');

-- Insert data into products
INSERT INTO products VALUES
(101, 'Smartphone', 1, 699.99, 50),
(102, 'Laptop', 1, 1299.99, 30),
(103, 'T-shirt', 2, 19.99, 200),
(104, 'Jeans', 2, 49.99, 150),
(105, 'Novel', 3, 14.99, 100),
(106, 'Cookbook', 3, 24.99, 75),
(107, 'Blender', 4, 79.99, 40),
(108, 'Basketball', 5, 29.99, 60),
(109, 'Headphones', 1, 149.99, 80),
(110, 'Dress', 2, 89.99, 65);

-- Insert data into orders
INSERT INTO orders VALUES
(1001, 1, '2023-01-10', 699.99, 'Delivered'),
(1002, 2, '2023-01-15', 39.98, 'Delivered'),
(1003, 3, '2023-01-20', 1299.99, 'Shipped'),
(1004, 4, '2023-02-05', 149.99, 'Processing'),
(1005, 1, '2023-02-12', 104.98, 'Delivered'),
(1006, 5, '2023-02-18', 179.98, 'Delivered'),
(1007, 6, '2023-03-05', NULL, 'Cancelled'),
(1008, 2, '2023-03-10', 699.99, 'Shipped');

-- Insert data into order_details
INSERT INTO order_details VALUES
```

```
(10001, 1001, 101, 1, 699.99),
(10002, 1002, 103, 2, 19.99),
(10003, 1003, 102, 1, 1299.99),
(10004, 1004, 109, 1, 149.99),
(10005, 1005, 105, 1, 14.99),
(10006, 1005, 106, 1, 24.99),
(10007, 1005, 103, 2, 19.99),
(10008, 1005, 108, 1, 29.99),
(10009, 1006, 107, 1, 79.99),
(10010, 1006, 108, 1, 29.99),
(10011, 1006, 103, 1, 19.99),
(10012, 1008, 101, 1, 699.99);

-- Insert data into employees
INSERT INTO employees VALUES
(1, 'James', 'Anderson', '2021-01-15', NULL),
(2, 'Sarah', 'Williams', '2021-03-10', 1),
(3, 'Thomas', 'Brown', '2021-05-20', 1),
(4, 'Emily', 'Davis', '2021-07-15', 2),
(5, 'Daniel', 'Miller', '2021-09-05', 2),
(6, 'Jessica', 'Wilson', '2022-01-10', 3);
```

Join Exercises

Basic Joins

1. **Exercise 1:** List all products with their category names.
2. **Exercise 2:** Display all orders with customer information (first and last name).
3. **Exercise 3:** Show all order details with product names and prices.

Intermediate Joins

4. **Exercise 4:** Find all customers who haven't placed any orders.
5. **Exercise 5:** List all products that have never been ordered.
6. **Exercise 6:** Display orders with their total value calculated from order_details.

Advanced Joins

7. **Exercise 7:** Show employees with their supervisor's name.
8. **Exercise 8:** For each category, show the total number of products and the average price.
9. **Exercise 9:** Create a report showing customers, their orders, and the products in each order.

Subquery Exercises

Basic Subqueries

10. **Exercise 10:** Find products that are more expensive than the average product price.
11. **Exercise 11:** List customers who have placed orders worth more than $500.
12. **Exercise 12:** Find categories that have at least 3 products.

Intermediate Subqueries

13. **Exercise 13:** Display customers who have ordered at least one electronic product.
14. **Exercise 14:** Find products that have a higher unit price than all products in the 'Books' category.
15. **Exercise 15:** Show orders that include products from multiple categories.

Advanced Subqueries

16. **Exercise 16:** Find customers who have ordered all products in a specific category.
17. **Exercise 17:** Calculate the difference between each product's price and the average price in its category.
18. **Exercise 18:** For each customer, find the most expensive product they've ordered.