### **Coding Test: Order Processing System**

### **Objective:**

Create a simplified model and accompanying method(s) to simulate order processing in a trading system.

### **Setup:**

* Assume you have a Rails model **Order** with the following attributes: **id, user\_id, price, quantity, order\_type** (buy or sell), **status** (pending, completed, canceled), and timestamps.
* The database is PostgreSQL.

### **Task:**

1. **Implement a Scope:**
   * Write a scope in the **Order** model to fetch all completed buy orders.
2. **Business Logic Method:**
   * Implement a method **process\_order** in the **Order** model. This method should take an **Order** instance as an argument and perform the following:
     + If the order is a buy order and the price is below a specified threshold, mark the order as completed.
     + If the order is a sell order and the price is above a specified threshold, mark the order as completed.
     + Otherwise, mark the order as canceled.
   * Ensure the method handles any exceptions and logs appropriate errors.
3. **Database Query:**
   * Write a query (either ActiveRecord or raw SQL) to calculate the total quantity of completed orders for a specific user.

### **Evaluation Criteria:**

* Correctness: The code should correctly implement the requirements.
* Code Quality: Code should be clean, readable, and follow Ruby/Rails best practices.
* Error Handling: The code should gracefully handle potential errors.
* Efficiency: The database query should be efficient and optimized for performance.