Laboratory 8 - JPA Entity Relationships Summary

Author: Piotr Copek, Zuzanna Micorek

Date: 11.06.2025

Key Findings

- Successfully implemented **bidirectional one-to-many relationship** between `Client` and `Order` using JPA annotations.
- Used `@OneToMany` (Client → Orders) with `mappedBy` and `@ManyToOne` (Order → Client) for proper foreign key mapping.
- Configured `cascade = CascadeType.ALL` to automatically persist related entities.

Fetch Strategy Analysis

- Defaulted to **EAGER loading** for simplicity in initial implementation.
- Identified performance risks: Unnecessary data loading when accessing parent entities.
- Recommendation: Switch to **LAZY loading** for better efficiency, with transaction management to prevent `LazyInitializationException`.

Best Practices Applied

1. **Bidirectional synchronization**: Maintained relationship integrity by updating both sides when modifying associations.
2. **Cascading**: Ensured proper persistence lifecycle for related entities.
3. **Column mappings**: Explicitly defined database column names where needed.

Optimization Opportunities

- Replace EAGER with LAZY fetching for better performance
- Implement `JOIN FETCH` or DTO projections for query optimization
- Add batch fetching to prevent N+1 query problems

Conclusion

The current implementation correctly models the database relationships while following JPA standards. The main improvement needed is optimizing fetch strategies for production use. The bidirectional mapping and cascading configuration provide a solid foundation for entity management.