

If **basic CRUD is already done**, the next step is to **make it production-ready**. Below is a **structured Spring Boot task list** that gradually adds **real-world depth**: JPA power, DB handling, testing, and backend-driven schema management.

Sample Domain (assumption)

You can adapt this to any entity (User, Employee, Student, Order, Invoice).

Example Entity: **Employee**

`id, name, email, department, salary, status, createdAt, updatedAt`

TASK 1: Advanced JPA Fetching (Beyond findAll)

1. Fetch using Multiple Parameters

Task

- Fetch employees by:
 - department + status
 - salary range
 - name contains + department
 - createdAt between two dates

Implement

- Derived query methods
- `@Query` (JPQL)
- Optional `CriteriaBuilder` / `Specification`

Example

```
List<Employee> findByDepartmentAndStatus(String dept, String status);
```

```
@Query("""
SELECT e FROM Employee e
WHERE e.salary BETWEEN :min AND :max
""")
List<Employee> findEmployeesBySalaryRange(
    @Param("min") BigDecimal min,
    @Param("max") BigDecimal max
);
```

TASK 2: Pagination & Sorting (Real APIs Need This)

Task

- Implement pagination using:
 - page
 - size
 - sortBy
 - direction (ASC/DESC)

Controller Example

```
@GetMapping
public Page<Employee> getEmployees(
    @RequestParam int page,
    @RequestParam int size,
    @RequestParam String sortBy) {

    Pageable pageable = PageRequest.of(page, size, Sort.by(sortBy));
    return employeeRepository.findAll(pageable);
}
```

TASK 4: Database & Tables from Backend (No Manual SQL)

(Recommended): Liquibase

- Create tables
- Add indexes
- Add constraints
- Track DB versions

Task

- Create Liquibase changelog
- Create employee table
- Add unique constraint on email
- Add index on department

TASK 5: Unit & Integration Test Cases (Very Important)

Repository Tests

- Use `@DataJpaTest`
- Test custom queries

Service Layer Tests

- Mock repository using Mockito

Controller Tests

- Use `@WebMvcTest`

- Mock service layer

Task

- Write tests for:
 - Create employee
 - Fetch with filters
 - Pagination
 - Error scenarios

✓ This is **mandatory in real companies**