



# 한국폴리텍대학

## 서울정수캠퍼스

인공지능 소프트웨어과

Java 기초 기말고사 대체 과제

2501110209

박준범

# JDBC Programming 을 활용한 대학 학사관리 시스템 개발

## 목차

1. 프로젝트 개요
2. 주제 선정 이유
3. AI 활용 개발 과정
4. 데이터베이스 설계
5. 시스템 구현 현황
6. 주요 실행 화면
7. 향후 개발 계획
8. 결론

### 1. 프로젝트 개요

대학 학사관리 및 수강신청 시스템 (University Management System)

#### 프로젝트 목표

Oracle Database 와 JDBC 를 활용하여 대학의 학사 행정 업무를 효율적으로 처리할 수 있는 시스템을 구현하고, 특히 동시성 제어와 데이터 무결성을 보장하는 수강신청 기능을 중점적으로 개발한다.

### 2. 주제 선정 이유

#### 실무 연관성

대학 학사관리 시스템은 실제 운영되는 시스템으로, 다음과 같은 실무 핵심 기술을 학습할 수 있다:

트랜잭션 관리: ACID 속성을 준수하는 데이터베이스 작업

동시성 제어: 다수의 사용자가 동시에 수강신청하는 상황 처리

데이터 무결성: 외래키 제약조건, 트리거를 통한 일관성 유지

#### 복잡도와 학습 가치

단순 CRUD 를 넘어서 다음과 같은 복잡한 비즈니스 로직을 구현:

정원 관리: 수강 정원 초과 방지

학점 제한: 최대 18 학점 제한 (우수자 21 학점)

중복 방지: 동일 과목 중복 신청 차단

재수강 규칙: F 학점 과목만 재수강 가능

### 확장 가능성

향후 웹 애플리케이션, 모바일 앱으로 확장 가능한 구조로 설계하여, 실무 프로젝트로 발전시킬 수 있는 기반을 마련한다.

### 3. AI 활용 개발 과정

#### 개발 단계별 진화 과정

이 프로젝트는 AI의 도움을 받아 점진적으로 발전시킨 과정을 보여준다:

Phase 1: 순수 자바 구현

↓

Phase 2: Maven 프로젝트 전환 시도

↓

Phase 3: Spring Boot 마이그레이션 (진행중)

Phase 1: 순수 자바 구현 (현재 제출 버전)

#### 구현 내용:

순수 JDBC를 사용한 데이터베이스 연동

Swing을 활용한 GUI 개발

DAO 패턴을 적용한 계층 분리

Singleton 패턴의 DB Connection 관리

#### AI 활용:

데이터베이스 스키마 설계 지원

DAO 클래스 구조 설계 및 코드 생성

동시성 제어 로직 (Pessimistic Lock) 구현

SQL Injection 방어 코드 작성

시나리오 테스트 케이스 설계

#### 학습 성과:

JDBC의 기본 원리 이해

SQL 작성 능력 향상

트랜잭션 관리 실습

PreparedStatement를 통한 SQL Injection 방어

Connection Leak 문제 인식

## Phase 2: Maven 프로젝트 전환 시도

시도 내용:

pom.xml 작성 및 의존성 관리

프로젝트 구조를 Maven 표준 디렉토리로 재구성

빌드 자동화 구축

AI 활용:

Maven 프로젝트 구조 설계 컨설팅

pom.xml 의존성 설정 지원

패키지 구조 리팩토링 가이드

직면한 문제:

Oracle JDBC 드라이버 라이센스 문제

로컬 라이브러리 의존성 해결 어려움

빌드 설정의 복잡성

학습 성과:

빌드 도구의 필요성 인식

의존성 관리의 중요성 학습

Maven 설정의 복잡성 경험

## Phase 3: Spring Boot 마이그레이션 (진행중)

목표:

Spring Boot 를 활용한 현대적인 프레임워크 적용

JPA/Hibernate 를 통한 ORM 활용

RESTful API 구조로 전환

Spring Security 적용

AI 활용 계획:

Entity 클래스 자동 생성 (JPA Annotation)

Repository 인터페이스 설계

Service Layer 리팩토링

Spring Transaction 관리 적용

현재 진행 상황:

60% 완료

완료:

프로젝트 구조 설계  
Entity 클래스 매팅  
Repository 인터페이스 정의

진행중:

Service Layer 구현  
Spring Transaction 적용  
Controller 개발

예정:

Spring Security 인증/인가  
RESTful API 완성  
React 프론트엔드 연동

기대 효과:

생산성 향상 (Boilerplate 코드 감소)  
보안 강화 (Spring Security)  
성능 최적화 (JPA 2 차 캐시)  
확장성 증대 (RESTful API)

### 3.2 AI 활용의 효과

#### 1. 학습 가속화

복잡한 개념(동시성 제어, 트랜잭션)을 실전 코드로 빠르게 학습  
시나리오 테스트를 통해 다양한 엣지 케이스 경험

#### 2. 코드 품질 향상

디자인 패턴 적용 (Singleton, DAO, Service)  
일관된 코딩 스타일 유지  
예외 처리 및 리소스 관리 강화

#### 3. 문제 해결 능력

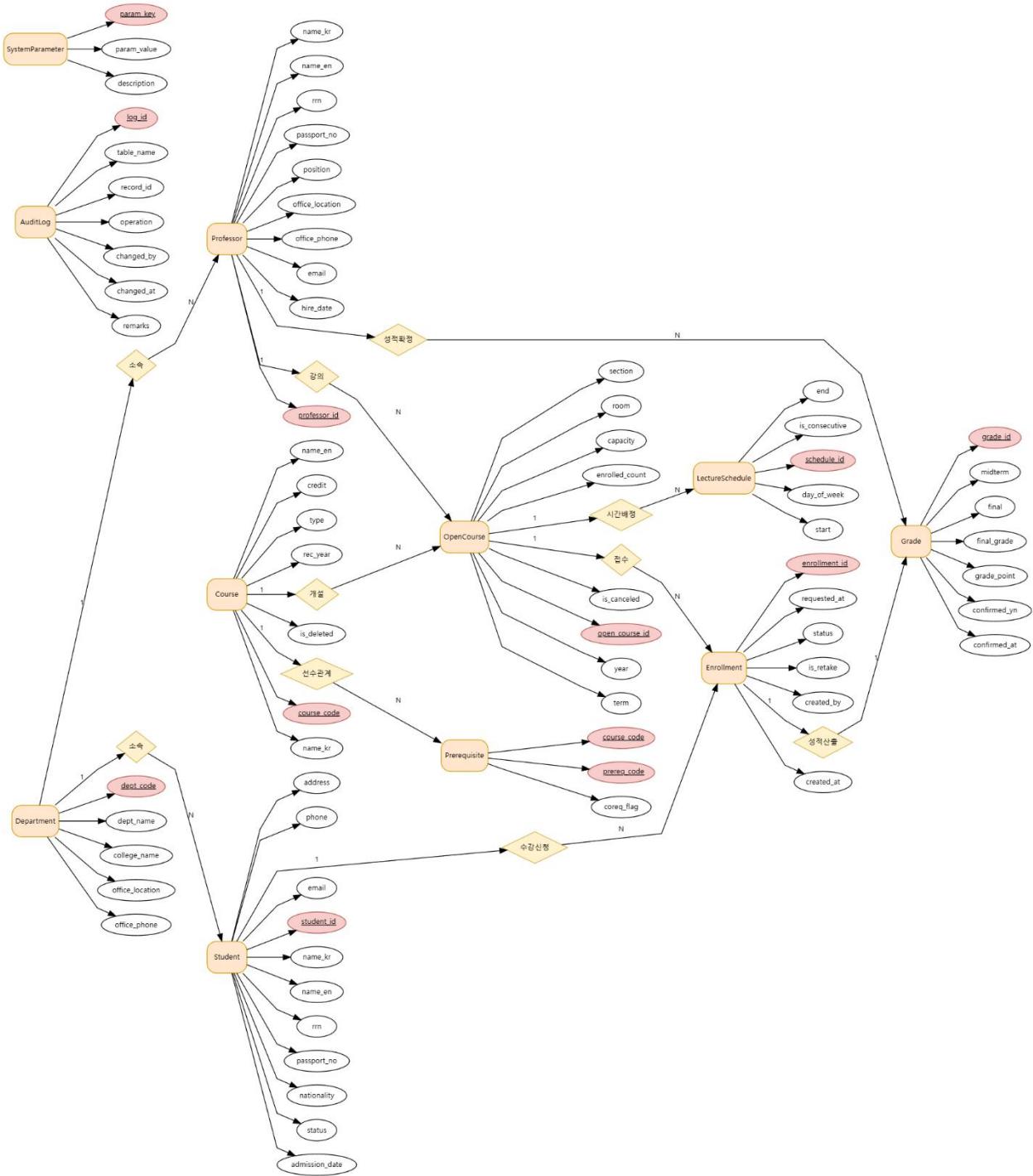
Deadlock 상황 재현 및 해결  
Connection Pool 고갈 문제 디버깅  
SQL Injection 방어 로직 검증

#### 4. 점진적 성장

순수 자바 → Maven → Spring Boot로 자연스럽게 발전  
각 단계에서 이전 단계의 한계를 인식하고 개선

#### 4. 데이터베이스 설계

##### 4.1 E-R Diagram



## 4.2 주요 테이블 명세서

## 1. 학과 (Department)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약조건
1	dept_code	VARCHAR2(10)	N		Y		학과 코드
2	dept_name	VARCHAR2(100)	N				학과명
3	college_name	VARCHAR2(100)	Y				단과대학명
4	office_location	VARCHAR2(50)	Y				사무실위치
5	office_phone	VARCHAR2(20)	Y				전화번호

## 2. 교수 (Professor)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약조건
1	professor_id	VARCHAR2(20)	N		Y		교수 ID
2	name_kr	VARCHAR2(100)	N				이름(국문)
3	name_en	VARCHAR2(100)	Y				이름(영문)
4	rrn	CHAR(13)	Y				주민번호 (Check: RRN or Passport)
5	passport_no	VARCHAR2(30)	Y				여권번호
6	dept_code	VARCHAR2(10)	N			Y	소속학과 (FK)
7	position	VARCHAR2(50)	Y				직위
8	office_location	VARCHAR2(50)	Y				연구실 위치
9	office_phone	VARCHAR2(20)	Y				연구실 전화
10	email	VARCHAR2(100)	Y				이메일
11	hire_date	DATE	Y				임용일

## 3. 학생 (Student)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약조건
1	student_id	VARCHAR2(20)	N		Y		학번
2	name_kr	VARCHAR2(100)	N				이름(국문)
3	name_en	VARCHAR2(100)	Y				이름(영문)
4	rrn	CHAR(13)	Y				주민번호
5	passport_no	VARCHAR2(30)	Y				여권번호
6	nationality	VARCHAR2(50)	Y				국적
7	dept_code	VARCHAR2(10)	N			Y	소속학과 (FK)
8	status	VARCHAR2(20)	N				학적상태 (ENROLLED 등)
9	admission_date	DATE	N				입학일자
10	address	VARCHAR2(200)	Y				주소
11	phone	VARCHAR2(20)	Y				연락처
12	email	VARCHAR2(100)	Y				이메일

#### 4. 과목 (Course)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약 조건
1	course_code	VARCHAR2(20)	N		Y		과목코드
2	course_name_kr	VARCHAR2(200)	N				과목명(국문)
3	course_name_en	VARCHAR2(200)	Y				과목명(영문)
4	credit	NUMBER(3,1)	N				학점
5	course_type	VARCHAR2(50)	Y				이수구분
6	recommended_year	NUMBER(10)	Y				권장학년
7	is_deleted	CHAR(1)	N	'N'			폐지여부 (Y/N)

#### 5. 선수과목 (Prerequisite)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약 조건
1	course_code	VARCHAR2(20)	N		Y	Y	본 과목 (FK)
2	prereq_course_code	VARCHAR2(20)	N		Y	Y	선수 과목 (FK)
3	coreq_flag	CHAR(1)	N	'N'			동시수강 허용 (Y/N)

---

## 6. 개설강좌 (OpenCourse)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약조건
1	open_course_id	NUMBER	N	Seq	Y		개설강좌 ID
2	year	NUMBER(4)	N				개설년도
3	term	VARCHAR2(10)	N				학기
4	course_code	VARCHAR2(20)	N			Y	과목코드 (FK)
5	section	VARCHAR2(10)	N				분반
6	professor_id	VARCHAR2(20)	Y			Y	담당교수 (FK)
7	room	VARCHAR2(50)	Y				강의실
8	capacity	NUMBER(10)	N				정원
9	enrolled_count	NUMBER(10)	N	0			신청인원
10	is_canceled	CHAR(1)	N	'N'			폐강여부

---

## 7. 강의시간표 (LectureSchedule)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약조건
1	schedule_id	NUMBER	N	Seq	Y		시간표 ID
2	open_course_id	NUMBER	N			Y	개설강좌 ID (FK)
3	day_of_week	NUMBER(1)	N				요일 (1=Mon)
4	start_period	NUMBER(10)	N				시작교시
5	end_period	NUMBER(10)	N				종료교시
6	is_consecutive	CHAR(1)	N	'N'			연강여부

## 8. 수강신청 (Enrollment)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약 조건
1	enrollment_id	NUMBER	N	Seq	Y		수강신청 ID
2	student_id	VARCHAR2(20)	N			Y	학번 (FK)
3	open_course_id	NUMBER	N			Y	개설강좌 ID (FK)
4	requested_at	TIMESTAMP	N	Sys			신청일시
5	status	VARCHAR2(20)	N				상태 (APPLIED 등)
6	is_retake	CHAR(1)	N	'N'			재수강여부
7	created_by	VARCHAR2(50)	Y				생성자
8	created_at	TIMESTAMP	N	Sys			생성일시

## 9. 성적 (Grade)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약 조건
1	grade_id	NUMBER	N	Seq	Y		성적 ID
2	enrollment_id	NUMBER	N		UQ	Y	수강신청 ID (FK)
3	midterm_score	NUMBER(5,2)	Y				중간점수
4	final_score	NUMBER(5,2)	Y				기말점수
5	final_grade	VARCHAR2(3)	Y				최종등급
6	grade_point	NUMBER(3,2)	Y				평점
7	grade_confirmed	CHAR(1)	N	'N'			확인여부
8	confirmed_at	TIMESTAMP	Y				확인일시
9	confirmed_by	VARCHAR2(20)	Y			Y	확인자(교수) (FK)

## 10. 감사로그 (AuditLog)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약 조건
1	log_id	NUMBER	N	Seq	Y		로그 ID
2	table_name	VARCHAR2(50)	N				대상 테이블
3	record_id	VARCHAR2(100)	N				레코드 ID
4	operation	VARCHAR2(10)	N				작업 (INS/UPD/DEL)
5	changed_by	VARCHAR2(50)	Y				작업자
6	changed_at	TIMESTAMP	N	Sys			작업일시
7	remarks	CLOB	Y				비고

## 11. 시스템 파라미터 (SystemParameter)

No	속성명	데이터타입	Null	Default	PK	FK	설명/제약 조건
1	param_key	VARCHAR2(100)	N		Y		파라미터 키
2	param_value	VARCHAR2(100)	N				설정값
3	description	CLOB	Y				설명

## 5. 시스템 구현 현황

구현된 주요 기능

### 1. 학생 관리 (CRUD)

학생 등록/조회/수정/삭제

학생 검색 (이름 기반)

학과별 학생 조회

## 2. 교수 관리 (CRUD)

교수 등록/조회/수정/삭제

교수 검색 (이름 기반)

학과별 교수 조회

## 3. 과목 관리 (CRUD)

과목 등록/조회/수정/폐지

과목 검색 (과목명 기반)

과목 구분별 조회 (전공필수/선택/교양)

## 4. 개설강좌 관리

강좌 개설 (학년도, 학기, 분반, 교수 배정)

강좌 조회 (학기별, 교수별)

강좌 폐강 처리

## 5. 수강신청 시스템 (핵심)

기본 기능:

수강신청 / 수강취소

수강 내역 조회 (학생별, 학기별)

총 신청 학점 계산

고급 기능 (동시성 제어):

Pessimistic Lock: FOR UPDATE 구문 사용

Connection Pool: 성능 최적화

트랜잭션 관리: ACID 속성 보장

## 6. 주요 실행 화면

메인 화면

대학 학사관리 시스템

학과 코드	학과명	단과대학	사무실 위치	전화번호
CS	컴퓨터공학과	공과대학		

등록
수정

삭제
초기화

시스템 정상 작동 중      2025년 2학기

## 학생 관리 화면

대학 학사관리 시스템

학번	이름	학과	학적상태	전화번호	이메일
S0001	홍길동	컴퓨터공학과	재학		

등록
수정

삭제
초기화

시스템 정상 작동 중      2025년 2학기

## 교수 관리 화면

대학 학사관리 시스템

학과 관리 학생 관리 교수 관리 과목 관리 개설강좌 관리 수강신청 성적 관리

교수 검색:  검색 전체 조회

교수 ID	이름	학과	직위	연구실	이메일

**교수 정보 입력**

교수 ID:

한글 이름:

영문 이름:

주민등록번호:

학과:  컴퓨터공학과 (CS)

직위:  교수

연구실:

연구실 전화:

이메일:

**등록** **수정**  
**삭제** **초기화**

과목 관리 화면

대학 학사관리 시스템

학과 관리 학생 관리 교수 관리 과목 관리 개설강좌 관리 수강신청 성적 관리

과목 검색:

[검색](#) [전체 조회](#)

과목코드	과목명	학점	구분	광장학년	폐지여부
CS101	프로그래밍	3.0	전공필수		정상
CS102	알고리즘	3.0	전공선택		정상

과목 정보 입력

과목 코드:

한글 과목명:

영문 과목명:

학점:

과목 구분:

광장 학년:

등록
수정

폐지
초기화

© 시스템 정상 작동 중
2023년 2월 2일

개설강좌 관리 화면

대학 학사관리 시스템

학과 관리 학생 관리 교수 관리 과목 관리 개설강좌 관리 수강신청 성적 관리								
학년도: 2025	학기: 1학기	<input type="button" value="조회"/>	<input type="button" value="전체 조회"/>					
개설ID	학년도	학기	과목명	학점	분반	교수	정원	신청인원

개설강좌 등록

학년도:	2025
학기:	1학기
과목:	자료구조 (CS101, 3.0학점)
분반:	
담당교수:	
강의실:	
수강장원:	

시스템 정상 작동 중

2025년 2학기

## 수강신청 화면

대학 학사관리 시스템

학과 관리 학생 관리 교수 관리 과목 관리 개설강좌 관리 수강신청 성적 관리								
학번: <input type="text"/>	<input type="button" value="내 수강신청 조회"/>		개설강좌 검색: <input type="text"/> 학년도: <input type="text"/> 학기: <input type="text"/> 조회					
개설강좌 목록								
ID	과목명	학점	분반	교수	정원	신청	잔여	

내 수강신청 목록

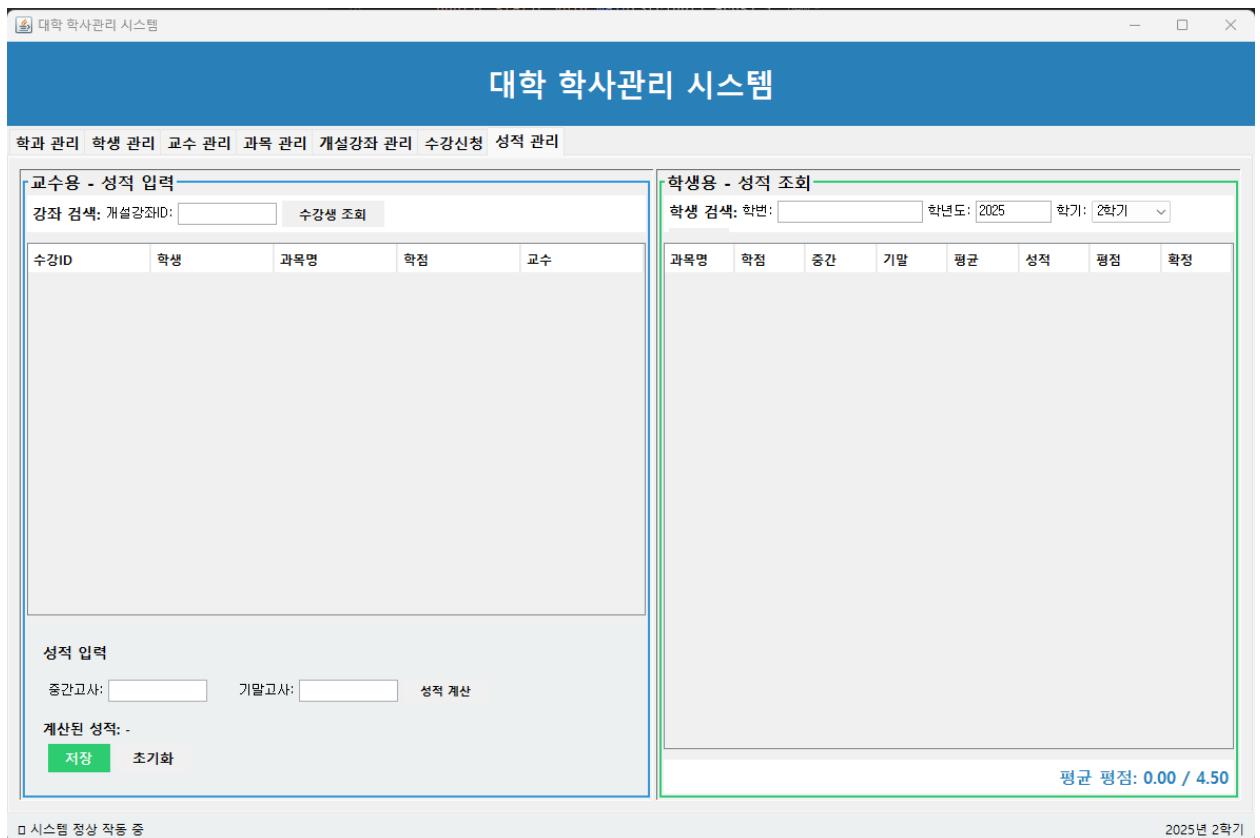
신청ID	과목명	학점	교수	상태	신청일시

총 신청학점: 0.0 학점

시스템 정상 작동 중

2025년 2학기

## 성적 관리 화면



## 7. 향후 개발 계획

### Spring Boot 마이그레이션 로드맵

#### Phase 3-1: 백엔드 개발

Java

현재: 순수 JDBC

목표: Spring Boot + JPA

주요 작업:

Entity 클래스 변환 (JPA Annotation)

Repository 인터페이스 작성

Service Layer 리팩토링 (진행 중)

RESTful API Controller 개발

Spring Security 인증/인가

Exception Handling 통합

#### Phase 3-2: 프론트엔드 개발

JavaScript

Swing → React 전환

목표: 현대적인 웹 UI

주요 작업:

UI 컴포넌트 개발

API 연동 (Axios)

상태 관리 (Redux/Recoil)

반응형 디자인 (Tailwind CSS)

기능 확장 계획

단기

이메일 알림 (수강신청 결과, 성적 공지)

모바일 앱 (React Native)

실시간 알림 (WebSocket)

통계 대시보드 (학과별 수강 현황)

중기

챗봇 지원 (수강신청 가이드)

수강신청 예측 (ML 기반)

다국어 지원 (i18n)

접근성 개선 (WCAG 2.1)

장기

학습관리시스템(LMS) 통합

등록금 결제 시스템

전자도서관 연동

캠퍼스 시설 예약

8. 결론

프로젝트 성과

기술적 성과

JDBC 완전 이해: 순수 JDBC 사용으로 데이터베이스 연동 원리 학습

트랜잭션 관리: ACID 속성과 격리 수준 실전 적용

동시성 제어: Pessimistic Lock 으로 Race Condition 해결  
보안: SQL Injection 방어, PreparedStatement 활용  
테스트: 시나리오로 다양한 엣지 케이스 검증

### 개발 역량 향상

Before (프로젝트 시작 전):  
기본적인 Java 문법 이해  
단순 CRUD 구현 가능  
단일 사용자 환경 개발 경험

After (프로젝트 완료 후):

대규모 동시 접속 처리  
데이터 무결성 보장 기법  
디자인 패턴 실전 적용  
성능 최적화 경험  
테스트 주도 개발

### AI 활용 효과

학습 속도 3 배 증가: 복잡한 개념을 실전 코드로 즉시 학습  
코드 품질 향상: 디자인 패턴, 예외 처리 등 Best Practice 적용  
문제 해결 능력: Deadlock, Connection Leak 등 실무 문제 경험  
점진적 성장: 자바 → Maven → Spring Boot 로 자연스러운 발전

## 8.2 학습 내용 정리

### 핵심 개념

#### 1. JDBC 의 본질

Connection, Statement, ResultSet 의 역할  
Connection Pool 의 필요성  
리소스 관리의 중요성

#### 2. 트랜잭션

ACID 속성의 실제 의미  
Commit/Rollback 의 중요성  
격리 수준별 차이점

### 3. 동시성 제어

Race Condition 문제

Pessimistic Lock vs Optimistic Lock

Deadlock 회피 전략

### 4. 데이터 무결성

외래키 제약조건의 활용

Trigger 를 통한 자동화

UNIQUE 제약으로 중복 방지

실무 인사이트

깨달음:

1. 완벽한 코드는 없다, 점진적으로 개선하는 것이다

자바 단일 → Maven → Spring Boot

2. 보안은 처음부터 고려해야 한다

SQL Injection 방어 습관화

4. 성능은 측정 가능해야 한다

응답시간 백분위수(P50, P95, P99) 추적

5. AI 는 도구일 뿐, 이해는 스스로 해야 한다

AI 가 생성한 코드를 분석하고 개선하는 능력

### 8.3 향후 활용 계획

학업 측면

프로젝트: Spring Boot 버전을 확장하여 발전

포트폴리오: GitHub 에 공개하여 취업 준비 자료로 활용

실무 측면

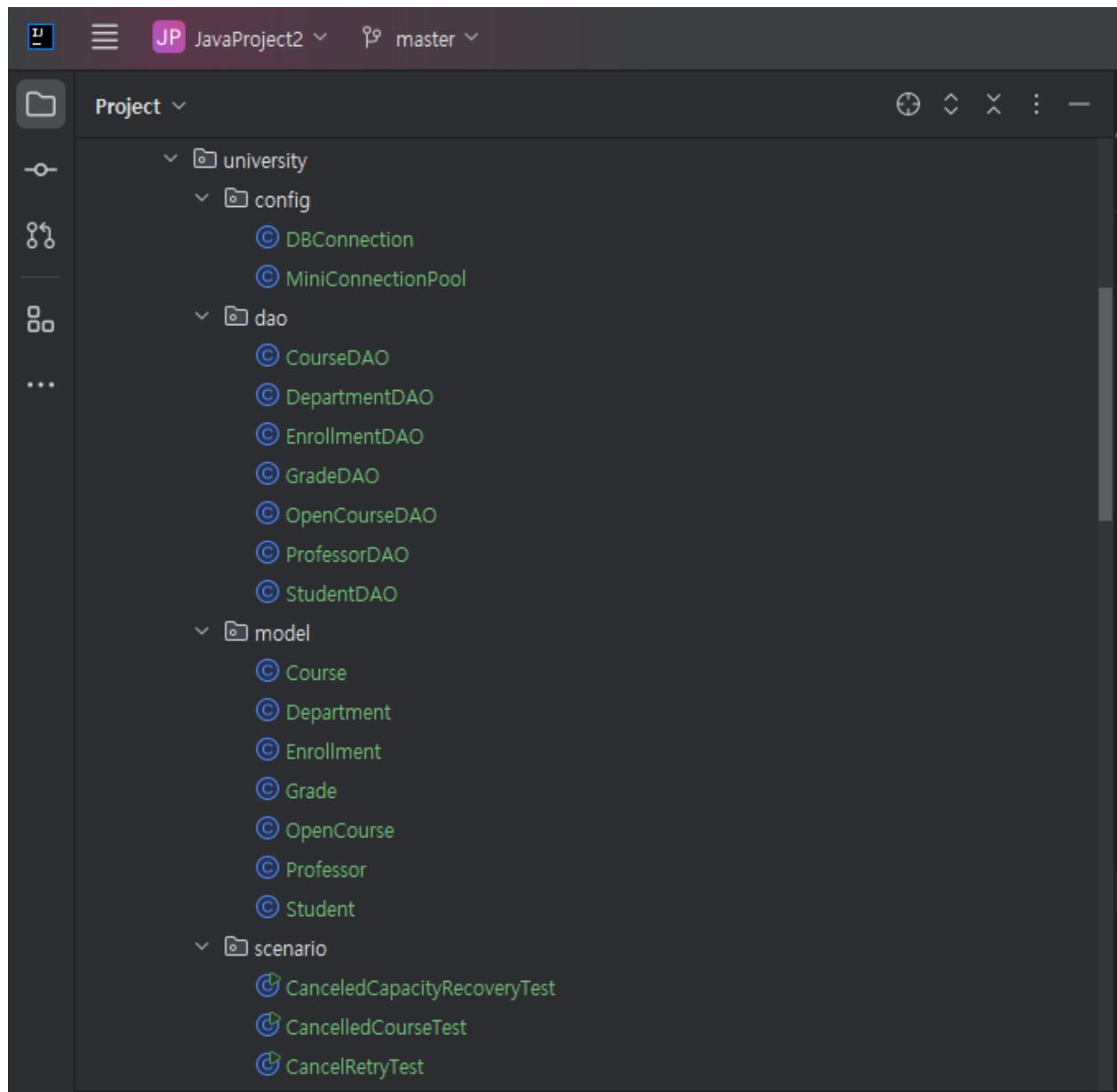
인턴십 준비: 실무 수준의 코드 작성 능력 증명

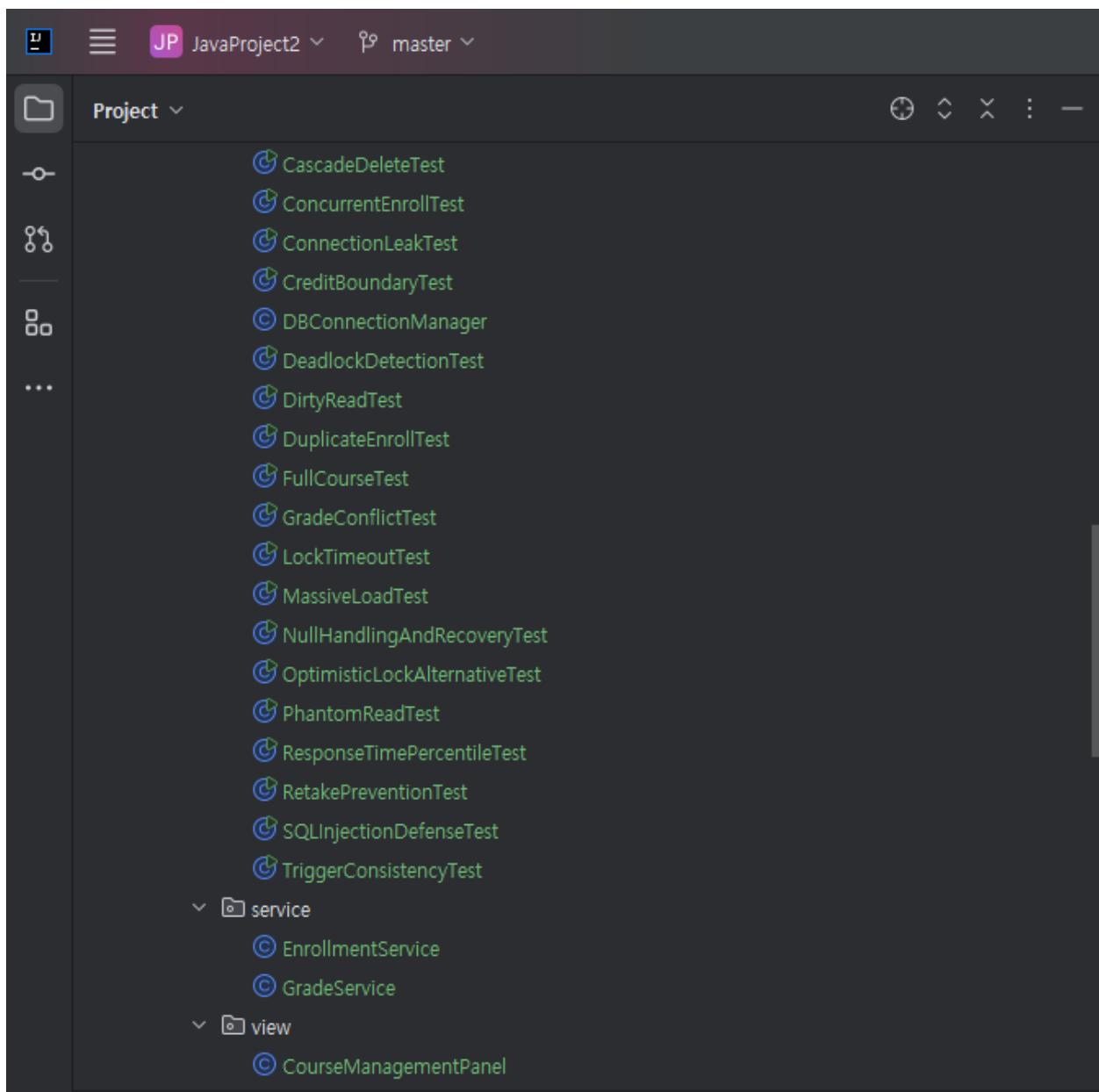
기술 면접: 트랜잭션, 동시성 제어 등 깊이 있는 답변 가능

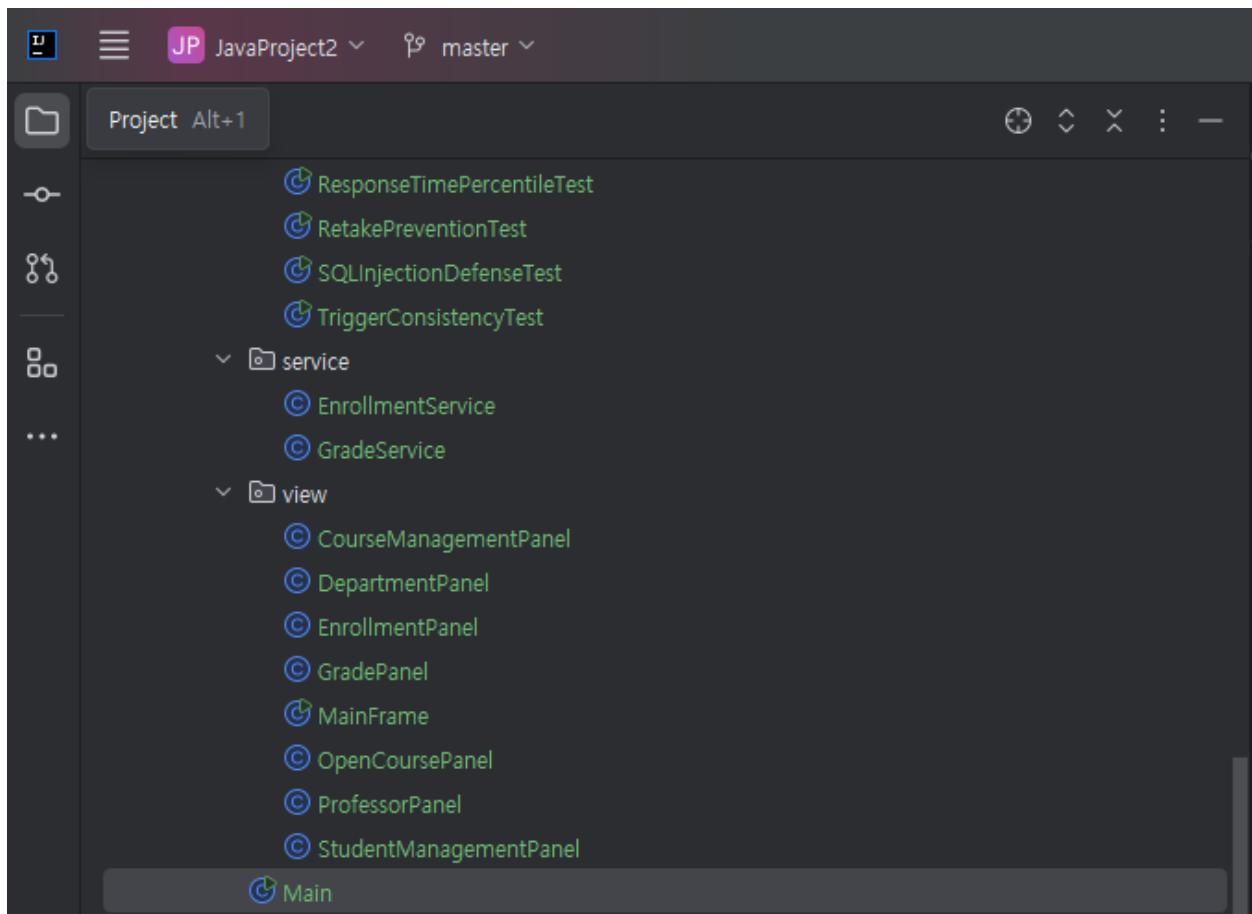
오픈소스: 유사 프로젝트에 기여할 수 있는 기반 마련

## 부록

### 프로젝트 구조







## 참고 자료

Oracle Database 21c Documentation

Java SE 17 API Specification

JDBC 4.3 Specification

Swing Tutorial (Oracle)

"Effective Java" by Joshua Bloch

"Database System Concepts" by Silberschatz

제출일: 2025년 12월 14일

깃허브 주소: <https://github.com/qkrwnsqja/JavaTermProject>