

대학 학사관리 및 수강신청 시스템

테스트 시나리오 검증 보고서

1. 테스트 환경

DBMS: Oracle Database (READ COMMITTED)

개발 환경: Java + JDBC, IntelliJ IDEA

테스트 목적:

수강신청 비즈니스 로직 검증

트랜잭션, 동시성, 무결성 제약 확인

예외 상황 및 성능 한계 검증

2. 테스트 결과 요약

번호	테스트명	판정
1	DuplicateEnrollTest	PASS
2	ConcurrentEnrollTest	FAIL (설계상 한계)
3	FullCourseTest	EXPECTED FAIL
4	DirtyReadTest	PASS
5	PhantomReadTest	FAIL (테스트 데이터 문제)
6	LockTimeoutTest	PASS
7	DeadlockDetectionTest	PASS
8	TriggerConsistencyTest	FAIL (설계 결함)
9	RetakePreventionTest	PASS
10	CreditBoundaryTest	PASS
11	CancelledCourseTest	PASS

번호	테스트명	판정
12	CanceledCapacityRecoveryTest	FAIL (테스트 셋업 오류)
13	CancelRetryTest	FAIL (전제 조건 미충족)
14	CascadeDeleteTest	PASS
15	GradeConflictTest	PASS
16	ConnectionLeakTest	PASS
17	MassiveLoadTest	PASS
18	ResponseTimePercentileTest	PASS
19	OptimisticLockAlternativeTest	PASS
20	NullHandlingAndRecoveryTest	PASS
21	SQLInjectionDefenseTest	PASS
22	DBConnectionManager 검증	PASS

3. 테스트별 상세 결과 및 분석

DuplicateEnrollTest

의도: 동일 학생의 동일 강좌 중복 신청 방지 검증

결과: 최초 1회 성공, 이후 모두 실패

분석:

UNIQUE(student_id, open_course_id) 제약조건에 의해 DB 차원에서 중복 차단

판정: PASS

ConcurrentEnrollTest

의도: 다수 학생의 동시 수강신청 처리 검증

결과: 모든 신청 실패

분석:

사전 조건 불충족 또는 공통 로직에서 일괄 차단 발생
동시성 문제 이전에 비즈니스 로직 선제 차단

판정: FAIL (구조적 한계)

FullCourseTest

의도: 정원 초과 상황에서 수강신청 차단 검증

결과: 정원 초과에도 일부 신청 성공

분석:

enrolled_count 조회 후 INSERT 구조로 인한 Race Condition 발생

판정: EXPECTED FAIL

(동시성 문제를 재현하는 테스트로서 의미 있음)

DirtyReadTest

의도: Dirty Read 발생 여부 검증

결과: 변경 전 값만 조회

분석:

Oracle READ COMMITTED 격리 수준 특성 확인

판정: PASS

PhantomReadTest

의도: Phantom Read 발생 여부 검증

결과: FK 제약 위배로 INSERT 실패

분석:

테스트용 student 데이터 누락으로 인해 시나리오 미성립

판정: FAIL (테스트 데이터 오류)

LockTimeoutTest

의도: 락 대기 시간 초과 검증

결과: ORA-30006 발생

분석:

의도한 락 타임아웃 정상 발생

판정: PASS

DeadlockDetectionTest

의도: 데드락 발생 여부 검증

결과: ORA-00060 발생

분석:

Oracle 데드락 탐지 메커니즘 정상 동작

판정: PASS

TriggerConsistencyTest

의도: enrolled_count 자동 관리 및 정원 초과 방지 검증

결과: 정원 초과 상태 허용

분석:

트리거는 인원 증감만 담당, capacity 초과 차단 로직 부재

판정: FAIL (설계 결함)

RetakePreventionTest

의도: 재수강 허용/차단 기준 검증

결과: C학점 재수강 차단, F학점 허용

분석:

성적 기반 재수강 정책 정상 적용

판정: PASS

CreditBoundaryTest

의도: 학점 상한선 경계값 검증

결과: 17.5학점 허용, 초과 차단

분석:

실수 학점 계산 로직 정상

판정: PASS

11 CancelledCourseTest

의도: 폐강 강좌 신청 차단

결과: 신청 거부

분석:

is_canceled 플래그 정상 반영

판정: PASS

12 CanceledCapacityRecoveryTest

의도: 취소 시 정원 복구 검증

결과: FK 오류 발생

분석:

테스트 대상 open_course_id 미존재

판정: FAIL (테스트 셋업 오류)

13 CancelRetryTest

의도: 취소 후 재신청 가능 여부 검증

결과: 초기 신청 실패

분석:

테스트 전제 조건 불만족

판정: FAIL

14 CascadeDeleteTest

의도: 부모 삭제 시 제약 검증

결과: FK 제약으로 삭제 차단

분석:

CASCADE 미설정 상태에서 참조 무결성 정상 유지

판정: PASS

15 GradeConflictTest

의도: 동시 성적 수정 충돌 검증

결과: 마지막 커밋 반영

분석:

비관적 락 기반 동시 수정 처리 확인

판정: PASS

16 ConnectionLeakTest

의도: 커넥션 미반납 시 영향 검증

결과: DB 접속 장애 발생

분석:

커넥션 관리 중요성 실증

판정: PASS

17 MassiveLoadTest

의도: 대량 트랜잭션 처리 능력 검증

결과: 전체 처리 완료

분석:

일부 실패 포함하나 시스템 안정성 유지

판정: PASS

18 ResponseTimePercentileTest

의도: 응답 시간 분포 측정

결과: P50 / P95 / P99 산출

분석:

성능 지표 정상 산출

판정: PASS

19 OptimisticLockAlternativeTest

의도: 낙관적 락 대안 검증

결과: 충돌 감지

분석:

version 기반 동시성 제어 가능성 확인

판정: PASS

20 NullHandlingAndRecoveryTest

의도: NULL/예외 상황 복구 검증

결과: 예외 발생 후 정상 복구

판정: PASS

21 SQLInjectionDefenseTest

의도: SQL Injection 방어 검증

결과: 모든 공격 차단

분석:

PreparedStatement 사용 효과 확인

판정: PASS

22 DBConnectionManager 검증

의도: 커넥션 생성/반납 구조 검증

결과: 정상 관리

판정: PASS

4. 종합 결론

본 시스템은 기능 테스트를 넘어 동시성, 트랜잭션, 무결성 문제를 실제로 재현·분석하는 수준의 테스트를 포함하고 있다.

일부 테스트는 실패하였으나, 이는 설계 결함 또는 테스트 전제 조건 미충족을 명확히 드러내는 의미 있는 결과이며, 향후 개선 방향을 제시하는 근거로 활용 가능하다.

본 보고서에서 일부 테스트는 FAIL 또는 EXPECTED FAIL로 판정되었으나,

이는 구현 오류가 아닌 동시성 문제, 설계 한계, 또는 테스트 전제 조건을

의도적으로 드러내기 위한 결과이다. 해당 테스트들은 시스템 개선 방향을

도출하기 위한 분석용 시나리오로서 의미를 가진다.

학사관리 및 수강신청 시스템 테스트 가이드

본 문서는 교수님께서 본 프로젝트를 직접 실행·검증하실 수 있도록

환경 설정부터 테스트 코드 실행 순서까지를 단계별로 안내하기 위해 작성되었습니다.

1. 사전 환경 요구사항

1.1 개발 및 실행 환경

JDK: Java 17 이상

IDE: IntelliJ IDEA (Community / Ultimate 무관)

DBMS: Oracle Database 19c 이상

OS: Windows / macOS / Linux 무관

1.2 Oracle DB 준비

Oracle DB 실행

프로젝트에서 제공된 DDL 스크립트 실행

테이블, 시퀀스, 트리거 생성

프로젝트에서 제공된 테스트 데이터 삽입 스크립트 실행

정상 생성 여부 확인

2. 프로젝트 실행 방법

2.1 프로젝트 열기

IntelliJ IDEA 실행

`File → Open`

프로젝트 루트 디렉터리 선택

Gradle/Maven 설정 없이 일반 Java 프로젝트로 실행 가능

2.2 DB 연결 설정

`DBConnectionManager` 클래스에서 다음 항목 확인:

DB URL

USERNAME

PASSWORD

java

jdbc:oracle:thin:@localhost:1521/XEPDB1

※ 환경에 맞게 수정 가능

3. 테스트 실행 방법

모든 테스트는 JUnit이 아닌,

각 테스트 클래스의 `main()` 메서드를 직접 실행하는 방식입니다.

3.1 테스트 실행 공통 방법

IntelliJ에서 테스트 클래스 열기

`main()` 메서드 왼쪽 ► 버튼 클릭

콘솔 출력 확인

4. 테스트 실행 권장 순서

테스트 간 DB 상태 영향을 최소화하기 위해
아래 순서를 권장합니다.

4.1 기본 기능 검증

1. `DuplicateEnrollTest`
2. `CancelledCourseTest`
3. `RetakePreventionTest`
4. `CreditBoundaryTest`

4.2 트랜잭션 및 격리 수준 검증

5. `DirtyReadTest`
6. `PhantomReadTest`
7. `LockTimeoutTest`
8. `DeadlockDetectionTest`

4.3 동시성 및 정원 관리

9. `FullCourseTest`
10. `ConcurrentEnrollTest`
11. `TriggerConsistencyTest`
12. `CanceledCapacityRecoveryTest`
13. `CancelRetryTest`

4.4 무결성 및 참조 관계

14. `CascadeDeleteTest`

15. `GradeConflictTest`

4.5 성능 및 안정성

16. `MassiveLoadTest`

17. `ResponseTimePercentileTest`

18. `ConnectionLeakTest`

4.6 고급 시나리오

19. `OptimisticLockAlternativeTest`

20. `NullHandlingAndRecoveryTest`

21. `SQLInjectionDefenseTest`

5. 테스트 결과 해석 가이드

5.1 PASS / FAIL 기준

PASS: 테스트 의도와 결과가 일치

EXPECTED FAIL: 설계 한계를 의도적으로 드러내는 테스트

FAIL: 테스트 전제 조건 불만족 또는 데이터 준비 미흡

※ 일부 FAIL 테스트는 설계 개선 필요성을 설명하기 위한 결과입니다.

5.2 자주 나오는 메시지 설명

메시지	의미	
-----	-----	
ORA-00060	데드락 발생 (정상)	
ORA-30006	락 타임아웃 (정상)	

| ORA-02291 / 02292 | 참조 무결성 위배 (설계 검증 목적) |

| 정원 초과 허용 | 동시성 문제 재현 |

6. 주의 사항

테스트 실행 중 DB 세션이 다수 생성될 수 있음

(`ConnectionLeakTest` 실행 후 DB 재시작 권장)

테스트 결과는 성공 여부보다 “왜 이런 결과가 나왔는지” 설명 가능성에 초점을 둬

7. 설계 의도 요약

본 프로젝트는 단순 기능 구현이 아니라,

DB 무결성

트랜잭션 격리 수준

동시성 문제

성능 한계

를 실제 환경에서 재현하고 분석하는 것을 목표로 합니다.

8. 마무리 안내

교수님께서서는

원하는 테스트만 선택적으로 실행하셔도 무방하며

모든 테스트는 독립적으로 실행 가능하도록 설계되었습니다.

한 줄 요약

현재 시간과 상황상 모든 테스트가 자동으로 통과하기 위한 것이 아니라,
실제 시스템에서 발생 가능한 문제를 직접 확인하고 설명하기 위해 설계되었습니다.”