

1. 프로젝트 주제 : 시험 기간에 도서관을 다니면서 느꼈던 고충을 데이터베이스 프로젝트에 녹여내보면 어떨까? 라는 생각으로 도서관 예약 관리 시스템을 구축하기로 결정하였습니다. 좌석을 예약하는 과정에서 생기는 불편함, 또는 불만족을 수용하고 이를 관리한다면 학교 측에서도, 학생 측에서도 좋은 결과를 도출해낼 것이라고 생각하였습니다. 제가 느꼈던 가장 큰 고충은 자리를 미리 선점해놓고 3시간이 지나도 그 자리에 나타나지 않거나, 장시간 자리를 비우는 등 좌석을 맡아두고서 제때 사용하지 않는 학생들로 인해, 자리들을 헛되이 버리는 사례가 많았습니다. 이를 민원 시스템을 도입하여 경고 누적을 쌓고, 그 누적이 많이 쌓이게 되면 3일간 도서관을 이용할 수 없게 하는 패널티를 부여하면 학생들 모두가 경각심을 가지게 되고, 보다 원활한 도서관의 이용이 이루어질 것이라고 생각하였습니다.

2. 요구 수집 목록 및 수집한 별첨

학생 사용자 정보 관리(학번, 이름, 전공, 연락처 등)
도서관 좌석 정보 관리(위치, 좌석 번호, 이용 가능 시간대)
실시간 좌석 예약 및 현황 정보 관리
좌석 이용 이력 관리 (예약 시간, 실제 사용 시간 등)
학생별 예약 제한 관리(노쇼, 부정 예약 등)
학생 건의사항 및 민원 관리
특별실(다인 스터디룸 등) 예약 관리 시스템

3. 요구 분석 내용

데이터 요구 분석 명세서	학생: 학번(pk), 이름, 전공, 연락처
	좌석: 좌석번호(pk), 위치, 유형, 상태
	예약: 예약번호(pk), 학번(fk), 좌석번호(fk), 예약시간, 상태
	실사용이력: 이력번호(pk), 학번(fk), 좌석번호(fk), 이용시작시간, 이용종료시간
트랜잭션 요구 분석 명세서	좌석 예약 트랜잭션
	예약 취소 트랜잭션
	좌석 사용 시작 및 종료 트랜잭션
	예약 상태 조회 트랜잭션
	노쇼 처리 트랜잭션

4. 개념적 설계

- 개체 후보 : 학과(학생이 소속된 학과 정보를 관리), 학생(도서관 시스템을 사용하는 사용자), 좌석(도서관 내의 각 좌석에 대한 정보), 예약(학생이 좌석을 예약한 기록), 실사용 이력(실제 좌석 사용 이력), 즐겨찾기 좌석(자주 사용하는 좌석 설정), 민원/신고(좌석 관련 문제 또는 시스템 민원), 이용제한(규칙 위반에 따라 제재된 사용자 정보)
- 개체 부호 검토 : 모든 개체 후보들은 독립적이며 고유성을 지닌다.

- 개체 및 관계 명세서(pk에 밑줄)

department(학과)	<u>dept_id</u> , dept_name department:student=1:N
student(학생)	<u>student_id</u> , name, phone, dept_id(fk) student:seat=N:M (preferredseat로 해소, 매핑 테이블) student:reservation=1:N student:complaint=1:N student:ban=1:N
seat(좌석)	<u>seat_id</u> , location, seat_type, capacity, is_available seat:reservation=1:N seat:complaint=1:N
reservation(예약)	<u>reservation_id</u> , student_id(fk), seat_id(fk), start_time, end_time, status reservation:seatusagelog=1:1
seatusagelog(실사용 이력)	<u>log_id</u> , reservation_id(fk), checkin_time, checkout_time
preferredseat(즐거찾기 좌석)	student_id(fk), <u>seat_id</u> (fk)
compaint(민원/신고)	<u>complaint_id</u> , submitter_id(fk), complaint_type, category, content, submitted_at, reported_seat_id(fk), reported_date
ban(이용제한)	<u>ban_id</u> , student_id(fk), ban_start, ban_end, ban_reason

- 운영상의 제약사항 : 좌석 이용 중복 방지 (동일 좌석이 같은 시간대에 2건 이상 예약되지 않도록 해야 함), 좌석 이용 시간 제약 (예약 시간이 지나면 체크인 불가하도록 해야 함), 신고 처리 흐름과 제재 누적 규칙 (일정 기간 내 신고 누적 시 자동 제재하도록 해야 함),....등.

[illegible]

The screenshot displays the Oracle SQL Developer interface with a database schema diagram for a library reservation system. The schema includes the following tables and their attributes:

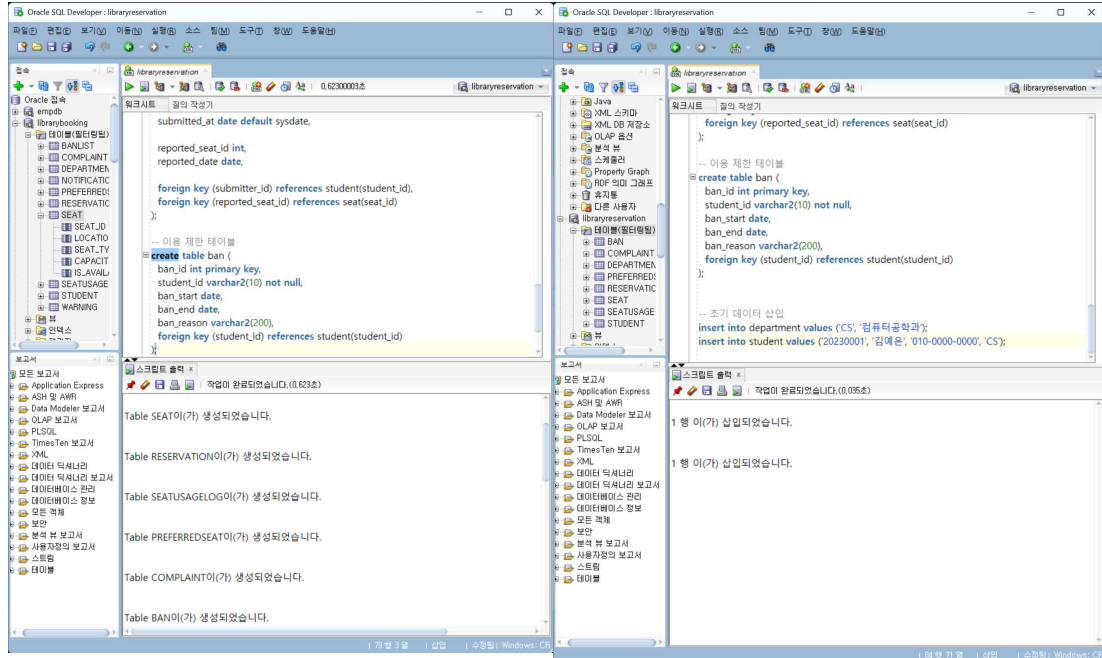
- department**: dept_id (PK), dept_name, dept_fk(dept_id).
- student**: student_id (PK), name, phone, dept_id (FK to department), student_department_fk (FK to department).
- seat**: seat_id (PK), location, seat_type, capacity, is_available, seat_fk(seat_id).
- reservation**: reservation_id (PK), start_time, end_time, status, student_student_id (FK to student), seat_seat_id (FK to seat), reservation_student_fk (FK to student), reservation_seat_fk (FK to seat).
- seatstatuslog**: log_id (PK), reservation_time, check_out_time, reservation_reservation_id (FK to reservation), seatstatuslog_reservation_fk (FK to reservation).
- complaint**: complaint_id (PK), submitter_id, complaint_type, category, content, submitted_at, student_student_id (FK to student), seat_seat_id (FK to seat), complaint_student_fk (FK to student), complaint_seat_fk (FK to seat).
- ban**: ban_id (PK), ban_start, ban_end, ban_reason, student_student_id (FK to student), ban_fk(ban_id), ban_student_fk (FK to student).

Relationships are indicated by lines with crow's foot notation. The 'student' table is the central entity, connected to 'department', 'seat', 'reservation', and 'complaint'. 'seat' is connected to 'reservation' and 'seatstatuslog'. 'reservation' is connected to 'complaint' and 'ban'. 'seatstatuslog' is connected to 'seat' and 'reservation'. 'complaint' is connected to 'ban'.

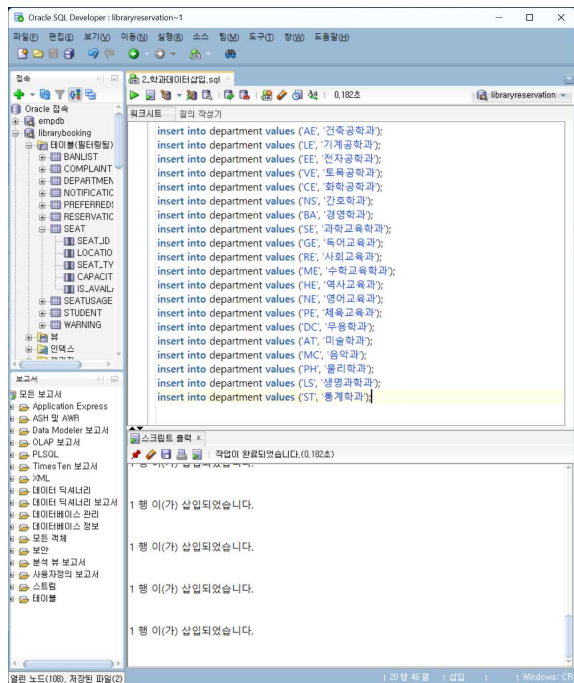
7. 테이블 생성 스크립트

8. 데이터 조작 스크립트 및 실행 화면 캡처

1. 테이블 생성 및 초기 데이터 삽입



2. 학과 데이터 삽입



3. 학생 데이터 삽입 (insert로 3명의 학생 데이터를 삽입한 후, 나머지 학생들은 엑셀 파일을 이용해서 읽어들이는 방식을 사용하였습니다.)

The first screenshot shows the Oracle SQL Developer interface with the 'Insert' menu open. The 'Insert into student values' SQL statement is entered in the main window:

```
insert into student values (20230002, '문상훈', '010-1111-1111', 'EE');
insert into student values (20230003, '박보람', '010-2222-2222', 'NS');
insert into student values (20230004, '유병재', '010-3333-3333', 'BA');
```

The second screenshot shows the 'Import Wizard' dialog box. The 'File name' field is set to 'C:\Users\B2103\Desktop\CSAR\T\database\LIBRARY_MANAGER\STUDENT.csv'. The 'Table' field is set to 'STUDENT'. The 'Table type' is set to 'Table'. The 'Table structure' is set to 'Table structure'. The 'Table data' is set to 'Table data'. The 'Table columns' are listed as 'STUDENT_ID', 'NAME', 'PHONE', and 'DEPT_ID'.

The third screenshot shows the 'STUDENT' table in the 'LIBRARY_MANAGER' schema. The table contains the following data:

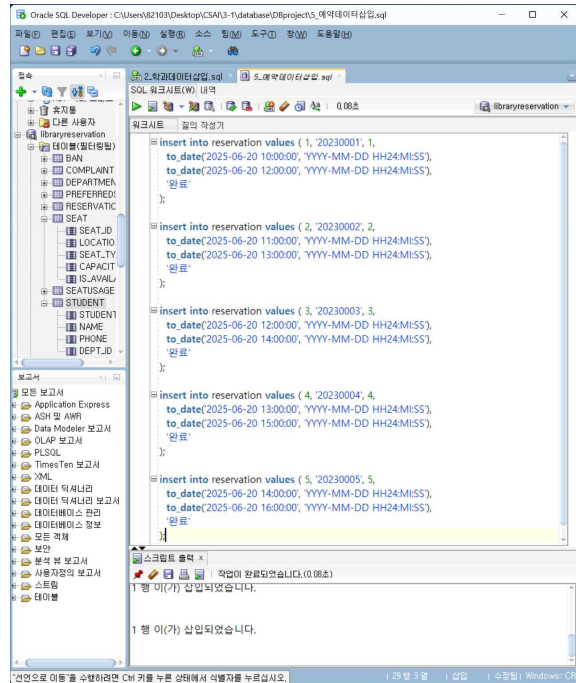
STUDENT_ID	NAME	PHONE	DEPT_ID
1	20230001	김예은	010-0000-0000 CS
2	20230002	문상훈	010-1111-1111 EE
3	20230003	박보람	010-2222-2222 NS
4	20230004	유병재	010-3333-3333 BA
5	20230005	임시환	010-5669-6186 EE
6	20230006	현빈	010-1027-6810 HE
7	20230007	변요한	010-8515-3766 HE
8	20230008	김지환	010-1191-5233 LS
9	20230009	전지현	010-5120-5344 SE
10	20230010	강하늘	010-1099-7636 DC
11	20230011	김예수	010-8556-8883 SE
12	20230012	나문희	010-6673-5313 CS
13	20230013	송강호	010-3675-6038 EE
14	20230014	송혜교	010-2476-5716 PH
15	20230015	박태수	010-7585-2787 AE
16	20230016	정려원	010-3139-7672 CE
17	20230017	김고은	010-5163-2044 HE
18	20230018	최민식	010-5384-4912 SE
19	20230019	김남일	010-6780-6268 NE
20	20230020	이혁진	010-1328-8153 CS
21	20230021	엄정화	010-8956-6035 SE
22	20230022	김우빈	010-3497-9904 CS
23	20230023	공효진	010-6407-5863 MC
24	20230024	박보검	010-2191-6052 HE
25	20230025	심은경	010-8765-1593 NE
26	20230026	공유	010-8149-9523 PE
27	20230027	한효주	010-2712-7306 GE
28	20230028	지성	010-5227-8026 PH
29	20230029	염정아	010-3489-2308 CS
30	20230030	한우희	010-3335-3699 BA
31	20230031	한기인	010-8439-9957 VE
32	20230032	조진웅	010-5156-8559 RE
33	20230033	이세영	010-2222-6602 PH
34	20230034	유재진	010-7352-3461 CE
35	20230035	정유미	010-5407-4982 MC
36	20230036	배두나	010-2450-4202 BA
37	20230037	이영애	010-3381-2539 VE
38	20230038	이준석	010-9171-5536 CS
39	20230039	신혜선	010-5706-8594 PH
40	20230040	김민준	010-8791-1923 AT

4. 전체 좌석 데이터 삽입

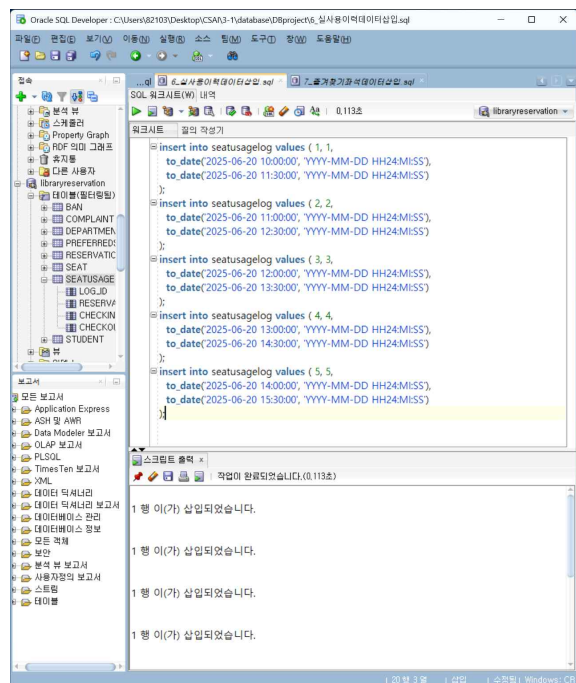
The screenshot shows the Oracle SQL Developer interface with the 'Insert' menu open. The 'Insert into seat values' SQL statement is entered in the main window:

```
INSERT INTO SEAT VALUES (405185, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405184, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405183, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405186, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405187, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405188, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405189, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405190, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405191, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405192, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405193, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405194, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405195, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405196, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405197, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405198, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405199, '4F' 노트북열람실, '노트북', 1, 'Y');
INSERT INTO SEAT VALUES (405200, '4F' 노트북열람실, '노트북', 1, 'Y');
```

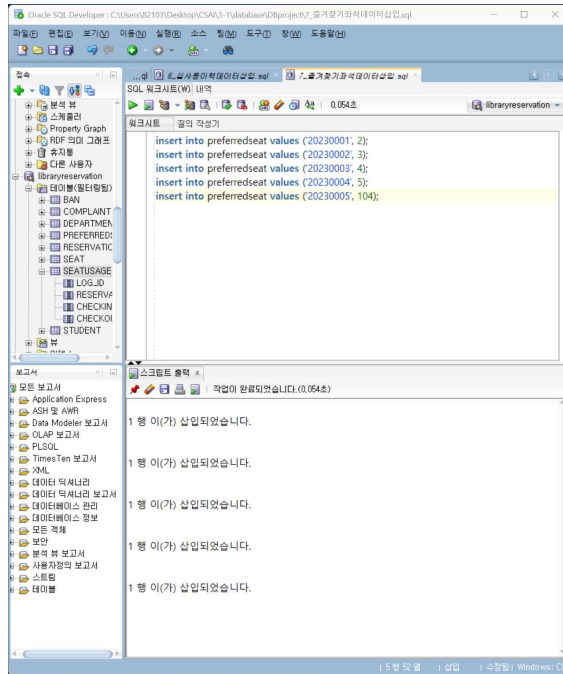
5. 예약 데이터 삽입



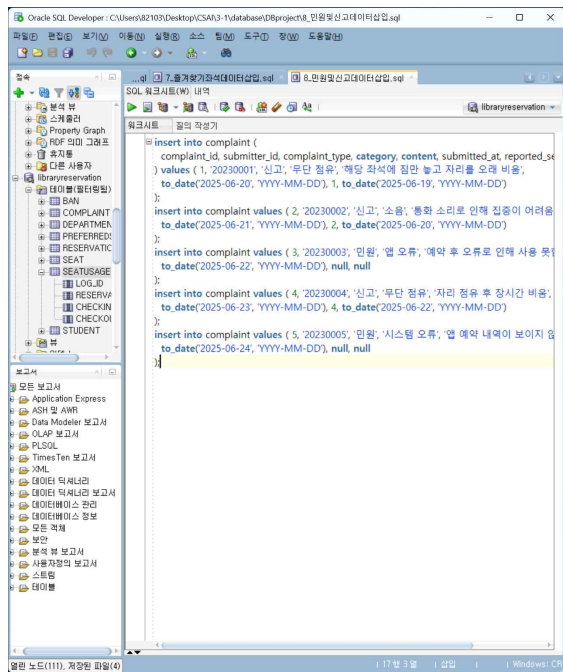
6. 실사용 이력 데이터 삽입



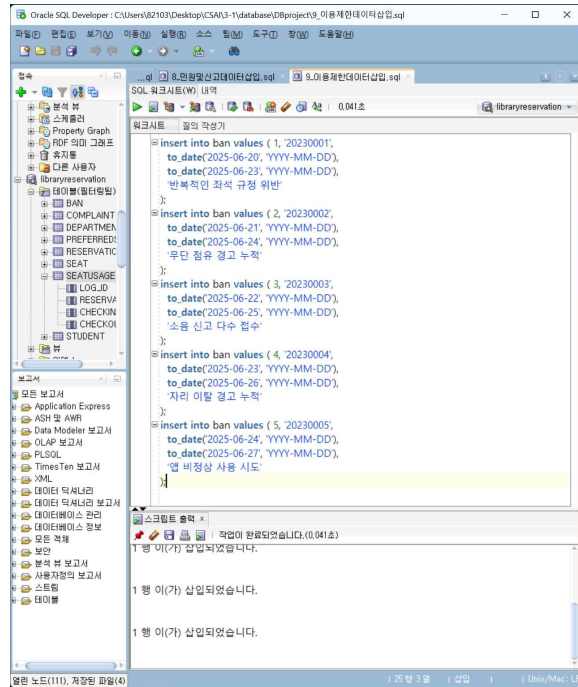
7. 즐겨찾기 좌석 데이터 삽입



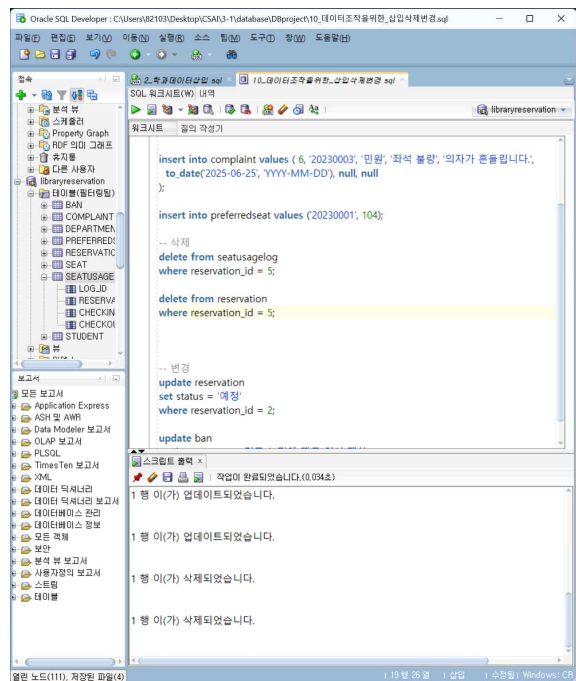
8. 민원 및 신고 데이터 삽입



9. 이용제한 데이터 삽입



10. 데이터 조작을 위한 삽입, 삭제, 변경



11. 검색

The following tables represent the data shown in the screenshots:

Top Left Screenshot: Query Results

RESERVATION_ID	STUDENT_ID	SEAT_ID	START_TIME	END_TIME	STATUS
2	620230001	104	25/06/25	25/06/25	예약

Top Right Screenshot: Query Results

SEAT_ID	RESERVATION_COUNT
1	1
2	2
3	104
4	4
5	3

Bottom Left Screenshot: Query Results

NAME	BAN_START	BAN_END
김예은	25/06/20	25/06/23
문상훈	25/06/21	25/06/24
박보검	25/06/22	25/06/25
유병재	25/06/23	25/06/26
임시환	25/06/24	25/06/27

Bottom Right Screenshot: Query Results

NAME
김예은
문상훈
박보검
유병재

