

악보 검색 프로그램

데이터베이스 및 실습 〈최종 보고서〉

학부 컴퓨터공학부

분반 01

제출일 2013. 06. 11.

팀명

TABLE

팀원명	학번
김○○	0000000
문찬호	2008160056
$\circ]\bigcirc\bigcirc$	0000000
홍()()	0000000



목 차

1. 프로젝트 주제 설명		2
2. 프로젝트 수행 범위		2
3. 프로젝트 진행 일정표		3
4. 요구 수집 및 분석 결과		4
5. 개념적 설계 단계		9
6. 논리적 설계 단계		15
7. 물리적 설계 단계		25
8. 테이블 생성 및 초기 데이터 삽입용 스크립트 파일 내용		30
9. 트랜잭션 처리를 위한 스크립트 파일 내용 및 화면 캡쳐		36
10. 팀원들의 프로젝트 소감	•••••	42

1. 프로젝트 주제 설명

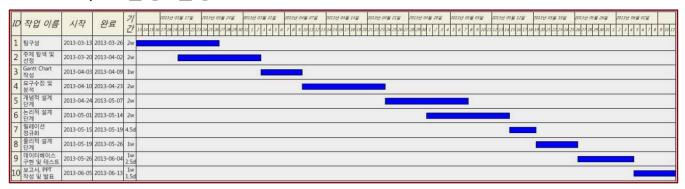
	· 악기 연주를 할 때 원하는 곡의 악보를 찾는 것이 생각보다 어렵습니다.
필요성	· 원하는 곡의 '제목'이나 '장르'와 같은 항목을 선택해서 악보를 검색하면 좀 더 편리하게 악보를 찾아서 연주할 수 있습니다.
목적	· '악보 검색 프로그램'은 판매자가 악보를 등록하고, 회원은 악보를 구매 및 다운로드 할 수 있게 하여 원하는 악보를 쉽게 구할 수 있습니다.

2. 프로젝트 수행 범위

1	팀 구성
2	주제 탐색 및 선정
3	간트 차트 작성
4	요구 수집 및 분석
(5)	개념적 설계 단계
6	논리적 설계 단계
7	릴레이션 정규화
8	물리적 설계 단계
9	데이터베이스 구현 및 테스트
10	PPT 발표 및 보고서 제출



3. 프로젝트 진행 일정표



요구 수집 및 분석은 프로젝트 중 가장 중요한 부분이며, 4월 중순의 시험기간에 프로젝트를 시작하면 일정 상 문제가 발생할 수 있다고 판단하여 가급적 중간고사가 끝나는 시점에서 프로젝트의 시작인 개 념적 설계부분을 시작하도록 하겠습니다.



4. 요구 수집 및 분석 결과

(1) 요구 수집 명세서

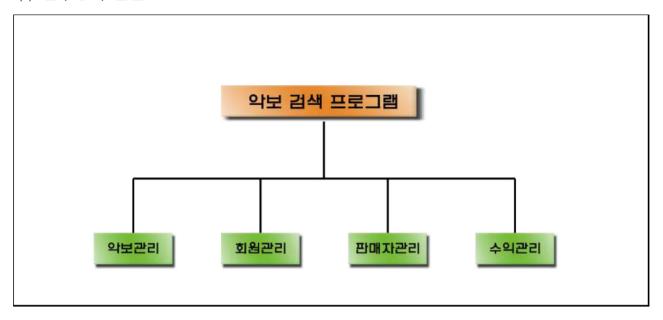
주제	악보 검색 프로그램
요구 수집 방법	인터뷰
요구 수집 대상	Table 조원 및 조원들의 지인
	1. 조원들이 다 같이 질문지를 작성한다.
인터뷰 절차	2. 음악 전공 지인에게 인터뷰를 한다.
인터큐 결사 	3. 조원들에게도 같은 인터뷰를 진행한다.
	4. 인터뷰 결과를 모아 인터뷰 내용을 작성한다.

[첨부 1: 인터뷰 내용]

- 1. 당신이 악보를 검색할 경우 어떤 유형으로 검색하시겠습니까?
- 악보를 검색한다면 먼저 **악보의 이름**이 있으면 좋을 것 같습니다. 그리고 항상 생각했던 것이지만 악보를 검색할 때 **악보의 이름이 생각나지 않는 경우**가 많습니다. 그렇기 때문에 부가적으로 악보의 **장르, 악기, 작곡가, 작사자, 가수이름 등**으로 검색하면 좋을 것 같습니다.
- 2. 악보를 검색한 후 원하는 악보가 있어서 구입 할 경우 어떤 정보가 필요할 것 같습니까?
- 악보 구입 후에 해당 악보가 잘못된 정보를 담고 있는 경우가 있던 것 같습니다. 그렇기 때문에 악보를 판매하는 **판매자의 연락처**나 **이메일** 등을 통해서 악보에 대한 A/S를 받을 수 있으면 좋겠습니다. 또한 신뢰성을 위해서 각 악보에 **평점**이 있으면 좋겠습니다.
- 3. 사이트의 회원 가입 시 필요하지 않다고 생각하는 정보를 모두 골라주세요. (3, 4)
- 1. 아이디 2. 비밀번호 3. 주소지 4. 주민등록번호 5. 이메일 6. 관심 장르
- 4. 사이트 가입 후 악보 검색 후 구입 시 필요하다고 생각하는 것은 무엇인가요?
- G마켓이나 옥션 등에서 물품 구입 시 사용하는 **Wishlist**나 **장바구니** 기능이 있으면 좋을 것 같습니다.
- 5. 판매자 등록을 하는 정책에 대해서는 어떻게 생각하십니까?
- 좋다고 생각합니다. 하지만 다른 판매자가 등록한 악보를 다운 받아 그 악보를 재등록하여, 이득을 챙기는 일은 최소화 할 수 있도록 했으면 좋겠습니다.
- 6. 악보 구매를 한다면 어떤 점을 고려하여 구매할 것 같습니까?
- 일단 악보가 있어야 구매할 것입니다. 그리고 5번 질문에 따르면 판매자를 통해서 악보를 구매한다고 했을 때 악보에 대한 **평가 점수**, 악보 판매자의 **신용 점수** 등이 부가적으로 필요할 것 같습니다.
- 7. 악보에 대한 악기 종류는 어떤 것이 있었으면 좋겠습니까?
- 가장 대중적인 악기인 **피아노**와 **기타**가 있으면 좋겠다고 생각합니다.
- 8. 참신한 기능이 있다면 어떤 기능이 있었으면 좋겠습니까?
- 그냥 악보를 검색해서 찾아주는 것보다 응용 프로그램들을 더 추가해서, 허밍을 통해 악보를 검색 하거나 악보의 **파트별로 검색**할 수 있으면 좋겠습니다. 총 합주악보에서 **원하는 악기만 뽑아서** 검색해 준다던지, 아니면 오페라 악보를 찾는데 레치타티보와 같은 일반적으로 구하기 힘든 악보 들을 찾을 수 있었으면 좋겠습니다.



(2) 업무영역 분할도



(3) 데이터 요구 분석 명세서

ⓐ '악보 관리' 관련 데이터 요구 분석 명세서

번호	'악보 관리' 관련 데이터 요구 분석 명세서
1	악보 는 유일한 일련번호를 가지며, 이름과 가격, 등록일, 발매일자, 장르, 판매수량, 판매자 등록번호, 제작자, 별점, 악기 정보가 필요하다.
2	악보의 장르 는 클래식, Pop, 가요, Jazz, 뮤지컬, 기타로 구분된다.
3	악보의 제작자 는 가수, 작곡가, 작사자 정보를 가질 수 있다. 그리고 각각은 값이 존재하지 않을 수 도 있으며, 여러 개가 될 수도 있다.
4	악보는 악기 로 구별이 가능하다.
5	모든 악보는 판매자 등록번호 를 항상 포함한다.
6	악보의 별점 은 0~5의 값을 가진다.
7	악보의 등록일 은 프로그램 제작일 이후의 값만 가질 수 있다.
8	악보의 일련번호 는 판매자 등록번호와 악보 등록 순서로 이루어진다.
9	악보의 악기는 피아노, 기타 등 대중적인 악기로 구분되며, 그 외의 악기는 기타로 구분된다.



ⓑ '회원 관리' 관련 데이터 요구 분석 명세서

번호	'회원 관리' 관련 데이터 요구 분석 명세서
1	회원은 유일한 ID를 가지며 Password, 관심장르, E-mail, 구매 내역, Wish List, 장바구니, 최근 검색 악보 정보가 필요하다.
2	회원은 최근 검색한 악보를 '최근 검색 악보' 에 최대 3개까지 보관할 수 있다.
3	관심 장르는 클래식, Pop, 가요, Jazz, 뮤지컬, 기타 중 복수로 택할 수 있다.
4	관심 장르는 0개가 될 수도 있다.
5	Wish List는 최대 20개의 악보 정보가 저장될 수 있다.
6	회원은 구입을 원하는 악보를 장바구니 에 저장할 수 있다.
7	회원은 악보를 구매한다.
8	회원은 악보의 평점을 매길 수 있다.

ⓒ '판매자 관리' 관련 데이터 요구 분석 명세서

번호	'악보 관리' 관련 데이터 요구 분석 명세서
1	판매자 는 유일한 ID, 사업자 등록번호를 가지며, 총 매출액, 연락처, Password, 계좌번호 정보가 필요하다.
2	총 매출액은 악보 판매 수량과 악보 가격을 합산하여 계산된다.
3	판매자는 일시적으로 판매한 악보가 없을 수도 있다.
4	판매자는 악보를 등록한다.
5	판매자는 악보를 판매한다.

◎ '수익 관리' 관련 데이터 요구 분석 명세서

번호	'악보 관리' 관련 데이터 요구 분석 명세서
1	수수료는 악보 판매 가격의 10%이다.
2	년/월별 수익 정보 를 가진다.
3	수익 정보는 판매된 즉, 수수료를 얻은 악보의 정보를 갖는다.
4	수익 정보는 판매자, 회원, 수수료 정보를 갖는다.
5	총 수익 정보는 년/월별 수익 정보의 수수료 합산을 타나낸다.



(4) 트랜잭션 요구 분석 명세서

ⓐ '악보 관리' 관련 트랜잭션 요구 분석 명세서

번호	'악보 관리' 관련 트랜잭션 요구 분석 명세서
1	새로운 악보 정보를 삽입한다.
2	판매자가 판매하는 악보의 이름, 장르, 악기, 가수, 작곡가, 작사자를 통해 악보를 검색한다.
3	기존 악보의 가격을 변경한다.
4	판매자가 없는 악보 정보는 삭제한다.
5	악보의 판매 수량을 변경한다.
6	악보의 평점을 변경한다.
7	악보의 가격을 통해 순위를 검색한다.

ⓑ '회원 관리' 관련 트랜잭션 요구 분석 명세서

번호	'회원 관리' 관련 트랜잭션 요구 분석 명세서
1	새로운 회원 정보를 삽입한다.
2	기존 회원 정보를 삭제한다.
3	기족 회원의 Password를 변경한다.
4	기존 회원의 관심 장르를 변경한다.
5	기존회원의 E-mail을 변경한다.
6	회원의 구매 내역 정보를 검색한다.
7	회원의 Wish List를 조회한다.
8	회원의 장바구니를 조회한다.
9	회원의 Wish List에 악보를 삽입한다.
10	회원의 장바구니에 악보를 삽입한다.
11	회원의 최근 검색 악보에 자동으로 최근 검색한 악보 정보 3개를 삽입한다.



ⓒ '판매자 관리' 관련 트랜잭션 요구 분석 명세서

번호	'판매자 관리' 관련 트랜잭션 요구 분석 명세서
1	악보 판매 수량을 자동으로 변경한다.
2	판매자의 총 매출액을 자동으로 계산한다.
3	판매자의 연락처를 변경한다.
4	판매자의 총 매출액을 검색한다.
5	판매자의 판매 수량을 검색한다.
6	판매자가 판매한 악보를 검색한다.
7	판매자가 등록한 악보를 검색한다.
8	기존 판매자 정보를 삭제한다.

◎ '수익 관리' 관련 트랜잭션 요구 분석 명세서

번호	'수익 관리' 관련 트랜잭션 요구 분석 명세서
1	회원이 악보를 구입하면 해당 악보의 수수료가 자동으로 총 수익에 더해진다.
2	관리자는 수익 정보를 조회한다.
3	관리자는 총 수익 정보를 검색한다.
4	판매자가 등록한 악보 가격의 10%가 자동으로 수수료 가격으로 정해진다.



5. 개념적 설계 단계

(1) 개체 후보 목록

번호	개체 후보 목록	관련 업무
1	악보	악보 관리
2	악보 장르	악보 관리
3	음반 참여자	악보 관리
4	악기	악보 관리
5	판매자 등록번호	악보 관리
6	별점	악보 관리
7	등록일	악보 관리
8	일련번호	악보 관리
9	회원	회원 관리
10	구매 내역	회원 관리
11	최근 검색 악보	회원 관리
12	회원 관심 장르	회원 관리
13	Wish List	회원 관리
14	장바구니	회원 관리
15	판매자	판매자 관리
16	총 매출액	판매자 관리
17	수수료	수익 관리
18	수익 정보	수익 관리
19	총 수익 정보	수익 관리



(2) 개체 후보 검토

번호	개체 후보	데이터 수	대표 속성	최종 개체 선정 여부	관련 업무
1	악보	N	일련번호, 이름, 가격, 등록일, 장르, 판매수량, 판매자 등록번호, 제작자, 별점, 악기	0	악보 관리
2	악보 장르	N	악보 일련 번호, 장르종류	Х	악보 관리
3	음반 참여자	N	악보 일련번호, ID, ^① 구분, 이름	0	악보 관리
4	악기	N	악보 일련 번호, 악기 종류	Х	악보 관리
5	판매자 등록번호	1	판매자 등록번호	Х	악보 관리
6	별점	1	별점	Х	악보 관리
7	등록일	1	등록일	Х	악보 관리
8	일련번호	1	일련번호	Х	악보 관리
9	회원	N	ld, Password, 관심 장르, E-mail	0	회원 관리
10	구매 내역	N	악보 일련번호, 구매한 날짜, 주문 번호	Х	회원 관리
11	최근 검색 악보	N	악보 일련번호, 검색 순번	Х	회원 관리
12	회원 관심 장르	5	회원 id, 장르 종류	Х	회원 관리
13	Wish List	20	악보 일련번호, 회원 ld, WishList 저장 유무	Х	회원 관리
14	장바구니	N	악보 일련번호, 회원 id, 장바구니 저장 유무	Х	회원 관리
15	판매자	N	사업자 등록번호, 총 매출액, 연락 처, ld, Password, 개좌번호	0	판매자 관리
16	총 매출액	1	총 매출액, 판매자 등록번호	Х	판매자 관리
17	수수료	N	수수료, 악보 일련번호	Х	수익 관리
18	수익 정보	N	악보 일련번호, 판매자 등록번호, 회원 id, 수수료	Х	수익 관리
19	총 수익 정보	N	총 수익	Х	수익 관리



(3) 개체 식별

번호	개체 후보	대표 속성	관련 업무
1	악보	일련번호, 이름, 가격, 등록일, 악보장르, 판매수량, 판매자 등 록번호, 별점, 악기	악보 관리
2	음반 참여자	ID, 구분, 이름	악보 관리
3	회원	ld, Password, 관심 장르, E-mail	회원 관리
4	판매자	사업자 등록번호, 연락처, ld, 총 매출액, Password, 계좌번호	판매자 관리

(4) 관계 식별

번호	관계 이름	관계된 개체 목록	관계 유형	관계 설명 속성
1	참여하다	악보, 음반참여자	1:N	-
2	검색하다	회원, 악보	1:N	등록일자
3	구매하다	회원, 악보	M:N	구매일자
4	평점을 매기다	회원, 악보	M:N	날짜, 평점
5	판매하다	판매자, 악보	M:N	악보 등록 번호, 악보 등록 일자

4

(5) 최종 개체 식별

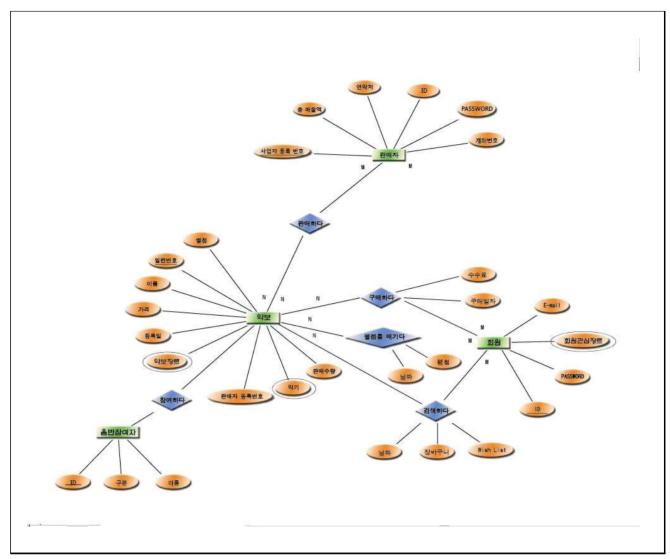
번호	개체 후보	대표 속성	관련 업무
1	악보	일련번호, 이름, 가격, 등록일, 장르, 판매수량, 판매자 등록번호, 제작자, 별점, 악기	악보 관리
2	음반 참여자	ID, 구분, 이름	악보 관리
3	회원	ld, Password, 관심 장르, E-mail	회원 관리
4	판매자	사업자 등록번호, 연락처, ld, 총 매출액, Password, 계좌번호	판매자 관리

(6) 최종 관계 식별

번호	관계 이름	관계된 개체 목록	관계 유형	관계 설명 속성
1	참여하다	악보, 음반참여자	1:N	-
2	검색하다	회원, 악보	1:N	등록일자
3	구매하다	회원, 악보	M:N	구매일자
4	평점을 매기다	회원, 악보	M:N	날짜, 평점
5	판매하다	판매자, 악보	M:N	악보 등록 번호, 악보 등록 일자



(7) ER 다이어그램 작성



(8) 트랜잭션 모델링

◎ 업무별, 유형별 트랜잭션 명세서

ⓐ '악보 관리' 관련 트랜잭션 요구 분석 명세서

유형	'악보 관리' 관련 트랜잭션 요구 분석 명세서
검색	판매자가 판매하는 악보의 이름, 장르, 악기, 가수, 작곡가, 작사자를 통해 악보를 검색한다.
검색	악보의 가격을 통해 순위를 검색한다.
삽입	새로운 악보 정보를 삽입한다.
삭제	판매자가 없는 악보 정보는 삭제한다.
	기존 악보의 가격을 변경한다.
변경	악보의 판매 수량을 변경한다.
	악보의 평점을 변경한다.



ⓑ '회원 관리' 관련 트랜잭션 요구 분석 명세서

유형	'회원 관리' 관련 트랜잭션 요구 분석 명세서	
검색	회원의 구매 내역 정보를 검색한다.	
	회원의 Wish List를 조회한다.	
	회원의 장바구니를 조회한다.	
	새로운 회원 정보를 삽입한다.	
삽입	회원의 Wish List에 악보를 삽입한다.	
48	회원의 장바구니에 악보를 삽입한다.	
	회원의 최근 검색 악보에 자동으로 최근 검색한 악보 정보 3개를 삽입한다.	
삭제	기존 회원 정보를 삭제한다.	
	기존 회원의 Password를 변경한다.	
변경	기존 회원의 관심 장르를 변경한다.	
	기존 회원의 E-mail을 변경한다.	

ⓒ '판매자 관리' 관련 트랜잭션 요구 분석 명세서

유형	'판매자 관리' 관련 트랜잭션 요구 분석 명세서	
	판매자의 총 매출액을 검색한다.	
거새	판매자의 판매 수량을 검색한다.	
검색	판매자가 판매한 악보를 검색한다.	
	판매자가 등록한 악보를 검색한다.	
삭제	기존 판매자가 정보를 삭제한다.	
	악보 판매 수량을 자동으로 변경한다.	
변경	판매자의 총 매출액을 자동으로 계산한다.	
	판매자의 연락처를 변경한다.	

ⓓ '수익 관리' 관련 트랜잭션 요구 분석 명세서

유형	'수익 관리' 관련 트랜잭션 요구 분석 명세서	
מו רב	관리자는 수익 정보를 조회한다.	
검색	관리자는 총 수익 정보를 검색한다.	
ᇤ건	회원이 악보를 구입하면 해당 악보의 수수료가 자동으로 총 수익에 더하여진다.	
변경	판매자가 등록한 악보 가격의 10%가 자동으로 수수료 가격으로 정해진다.	



6. 논리적 설계 단계

(1) 개체 릴레이션 스키마

개체	릴레이션 스키마
악보	악보(<u>일련번호</u> , 별점, 이름, 가격, 등록일, 판매수량, 판매자 등록번호(FK))
악보장르	악보장르(<u>이름, 악보일련번호(FK)</u>)
음반참여자	음반참여자(<u>ID</u> , 구분, 이름)
악기	악기(<u>이름, 악보일련번호(FK)</u>)
판매자	판매자(<u>사업자 등록번호</u> , 총 매출액, 연락처, ID, PASSWORD, 계좌번호)
회원	회원(<u>ID</u> , PASSWORD, E-mail)
회원관심장르	회원관심장르(<u>이름, 회원 ID(FK)</u>)

(2) 관계 릴레이션 스키마

관계	릴레이션 스키마
참여하다	참여하다(<u>음반참여자 ID(FK)</u> , 악보 <u>일련번호(FK)</u> , ^② 구분, 이름)
검색하다	검색하다(<u>회원 ID(FK), 악보일련번호(FK)</u> , 날짜, 장바구니 저장 유무, WishList 저장 유무)
구매하다	구매하다(<u>회원 ID(FK), 악보일련번호(FK)</u> , 구매일자, 수수료)
평점을 매기다	평점을 매기다(<u>ID(FK), 악보일련번호(FK)</u> , 날짜, 평점)
판매하다	판매하다(판매자 등록번호(FK), 악보일련번호(FK))

⁽²⁾ 구분은 '작곡가', '작사자', '가수'로 구성된 음반참여자를 구분짓는 속성이다.

(3) 단순화된 릴레이션 스키마

개체 및	릴레이션 스키마
관계	크네이는기미
악보	악보(<u>일련번호</u> , 별점, 이름, 가격, 등록일, 판매수량, 판매자 등록번호(FK))
악보장르	악보장르(<u>이름, 악보일련번호(FK)</u>)
음반참여자	음반참여자(<u>ID</u> , 구분, 이름)
악기	악기(<u>이름, 악보일련번호(FK)</u>)
판매자	판매자(<u>사업자 등록번호</u> , 총 매출액, 연락처, ID, PASSWORD, 계좌번호)
회원	회원(<u>ID</u> , PASSWORD, E-mail)
회원관심장르	회원관심장르(<u>이름, 회원 ID(FK)</u>)
참여하다	참여하다(음반참여자 <u>ID(FK), 악보일련번호(FK)</u> , ^② 구분, 이름)
검색하다	검색하다(회원 ID(FK), 악보일련번호(FK), 날짜, 장바구니 저장 유무, WishList 저장 유무)
구매하다	구매하다(<u>회원 ID(FK), 악보일련번호(FK)</u> , 구매일자, 수수료)
평점을 매기다	평점을 매기다(<u>ID(FK), 악보일련번호(FK)</u> , 날짜, 평점)
판매하다	판매하다(<u>판매자 등록번호(FK),</u> 악보일련번호(FK))



(4) 무결성 제약조건

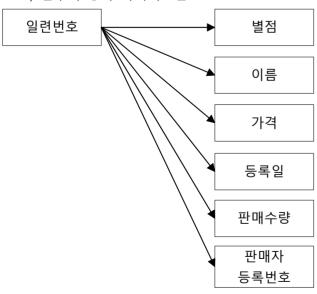
No.	제약조건	제약 유형	관련 릴레이션
1	별점은 1~5사이의 값을 갖는다.	CHECK	악보
2	이름, 가격, 등록일, 수수료는 널 값을 가질 수 없다.	NOT NULL	악보
3	구분, 이름은 널 값을 가질 수 없다.	NOT NULL	음반참여자
4	연락처, ID, PASSWORD, 계좌번호는 널 값을 가질 수 없다.	NOT NULL	판매자
5	ID는 유일한 값을 가져야 한다.	UNIQUE	판매자
6	ID는 널 값을 가질 수 없다.	NOT NULL	판매자
7	ID는 유일한 값을 가져야 한다.	UNIQUE	회원
8	ID는 널 값을 가질 수 없다.	NOT NULL	회원
9	E-mail은 널 값을 가질 수 없다.	NOT NULL	회원
10	E-mail은 유일한 값을 가져야 한다.	UNIQUE	회원
11	PASSWORD는 널 값을 가질 수 없다.	NOT NULL	회원
12	구매 일자는 널 값을 가질 수 없다.	NOT NULL	구매하다
13	날짜는 널 값을 가질 수 없다.	NOT NULL	평점을매기다
14	날짜는 널 값을 가질 수 없다.	NOT NULL	검색하다

(5) 릴레이션 정규화

@ 악보

<u>일련번호</u>	별점	이름	가격	등록일	판매수량	판매자 등록번호(FK)	

1) 함수적 종속 다이어그램



2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.

3) 2NF: 기본 키가 복합속성으로 이루어져있지 않으므로 제 2 정규형을 만족한다.

4) 3NF: 이행적 함수 종속을 포함하고 있지 않으므로 제 3 정규형을 만족한다.

5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.



(b) 장르

1) 함수적 종속 다이어그램



2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.

3) 2NF: 릴레이션에 복합속성으로 이루어진 기본 키만 가지므로 제 2 정규형을 만족한다.

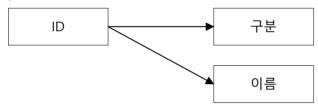
4) 3NF: 이행적 함수 종속을 포함하고 있지 않으므로 제 3 정규형을 만족한다.

5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.

ⓒ 음반참여자

ID	구분	이름

1) 함수적 종속 다이어그램



2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.

3) 2NF: 기본 키가 복합속성으로 이루어져있지 않으므로 제 2 정규형을 만족한다.

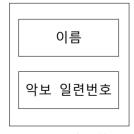
4) 3NF: 이행적 함수 종속을 포함하고 있지 않으므로 제 3 정규형을 만족한다.

5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.

@ 악기

이르	악보 일련번호(FK)
<u> </u>	<u> </u>

1) 함수적 종속 다이어그램



2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.

3) 2NF : 릴레이션에 복합속성으로 이루어진 기본 키만 가지므로 제 2 정규형을 만족한다.

4) 3NF: 이행적 함수 종속을 포함하고 있지 않으므로 제 3 정규형을 만족한다.

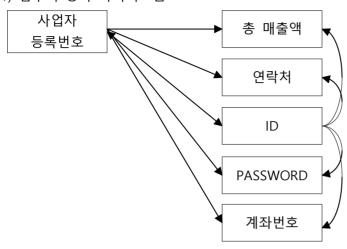
5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.



● 판매자

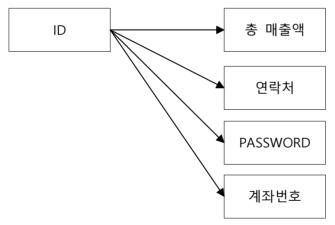
사업자 등록번호 총 매출액	연락처 ID	PASSWORD	계좌번호
----------------	--------	----------	------

1) 함수적 종속 다이어그램



- 2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.
- 3) 2NF: 기본 키가 복합속성으로 이루어져있지 않으므로 제 2 정규형을 만족한다.
- 4) 3NF : 이행적 함수 종속을 포함한다.





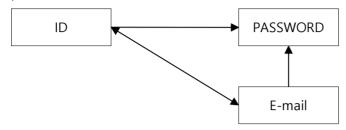
- 5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.
- → 이행적 함수 종속으로 인해 제 3 정규형을 만족하지 않기 때문에 릴레이션을 분해했으나, 성능에 대한 수행속도를 높이기 위하여 역정규화를 수행한다. 또한 갱신이상의 문제가 발생하지 않는다. (의도적으로 ID없이 판매자 등록이 불가능하도록 설계)



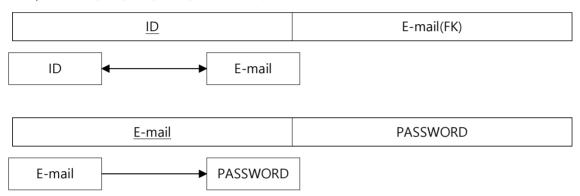
(f) 회원

<u>ID</u>	PASSWORD	E-mail
-----------	----------	--------

1) 함수적 종속 다이어그램



- 2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.
- 3) 2NF: 기본 키가 복합속성으로 이루어져있지 않으므로 제 2 정규형을 만족한다.
- 4) 3NF : 이행적 함수 종속을 포함한다.



- 5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.
- → 이행적 함수 종속으로 인해 제 3 정규형을 만족하지 않기 때문에 릴레이션을 분해했으나, 성능에 대한 수행속도를 높이기 위하여 역정규화를 수행한다. 또한 갱신이상의 문제가 발생하지 않는다. (의도적으로 ID, E-mail없이 회원 등록이 불가능하도록 설계)



g 관심장르

이름	회원 ID(FK)

1) 함수적 종속 다이어그램



2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.

3) 2NF: 릴레이션에 복합속성으로 이루어진 기본 키만 가지므로 제 2 정규형을 만족한다.

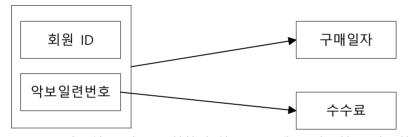
4) 3NF : 이행적 함수 종속을 포함하고 있지 않으므로 제 3 정규형을 만족한다.

5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.

(h) 구매하다

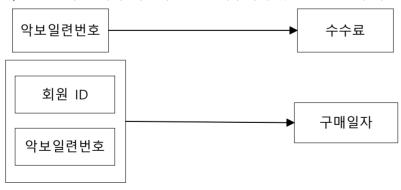
<u> </u>	<u>회원 ID(FK)</u>	<u>악보일련번호(FK)</u>	구매일자	수수료
----------	------------------	-------------------	------	-----

1) 함수적 종속 다이어그램



2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.

3) 2NF: 기본 키가 복합속성으로 이루어져 있고 완전 함수적 종속을 만족하지 않는다.



- → 악보 일련번호가 수수료를 함수적으로 결정한다.
- → 악보 개체에 수수료 속성을 '추가'한다.

4) 3NF: 이행적 함수 종속을 포함하고 있지 않으므로 제 3 정규형을 만족한다.

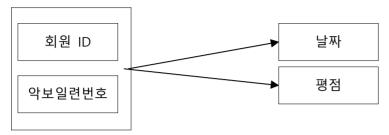
5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.



(i) 평점을 매기다

<u>회원 ID(FK)</u>	<u>악보일련번호(FK)</u>	날짜	평점	
				П

1) 함수적 종속 다이어그램

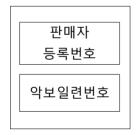


- 2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.
- 3) 2NF: 기본 키가 복합속성으로 이루어져 있고 완전 함수적 종속을 만족하므로 제 2 정규형을 만족한다.
- 4) 3NF: 이행적 함수 종속을 포함하고 있지 않으므로 제 3 정규형을 만족한다.
- 5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.

(j) 판매하다

<u>판매자등록번호(FK)</u>	<u>악보 일련번호(FK)</u>
--------------------	--------------------

1) 함수적 종속 다이어그램



- 2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.
- 3) 2NF: 릴레이션에 복합속성으로 이루어진 기본 키만 가지므로 제 2 정규형을 만족한다.
- 4) 3NF: 이행적 함수 종속을 포함하고 있지 않으므로 제 3 정규형을 만족한다.
- 5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.



(k) 참여하다

악보일련번호(FK)	참여자 ID(FK)

1) 함수적 종속 다이어그램



2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.

3) 2NF: 릴레이션에 복합속성으로 이루어진 기본 키만 가지므로 제 2 정규형을 만족한다.

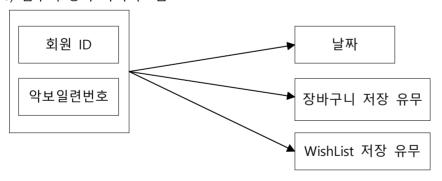
4) 3NF: 이행적 함수 종속을 포함하고 있지 않으므로 제 3 정규형을 만족한다.

5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.

① 검색하다

회원 ID(FK) 악보일련번호(FK)	날짜	장바구니 저장 유무	WishList 저장 유무
----------------------	----	------------	----------------

1) 함수적 종속 다이어그램



2) 1NF: 다중치 속성을 포함하지 않으므로 제 1 정규형을 만족한다.

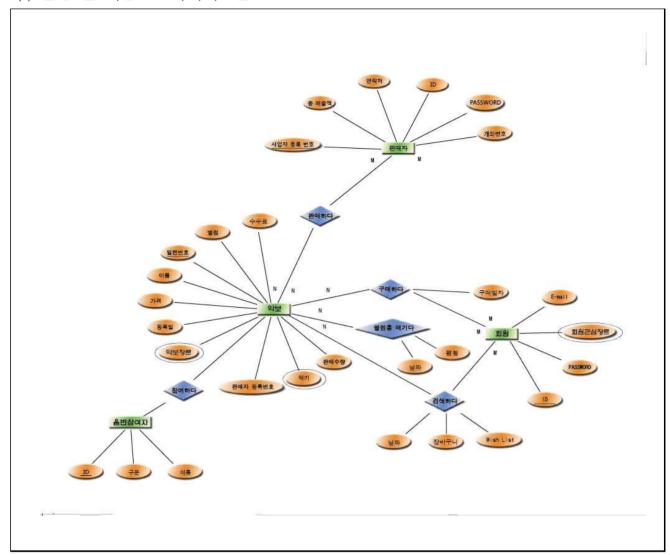
3) 2NF: 기본 키가 복합속성으로 이루어져 있고 완전 함수적 종속을 만족하므로 제 2 정규형을 만족한다.

4) 3NF: 이행적 함수 종속을 포함하고있지 않으므로 제 3 정규형을 만족한다.

5) BCNF: 후보 키가 아닌 결정자가 없으므로 보이스/코드 정규형을 만족한다.



(6) 변경 된 최종 ER 다이어그램





(7) 최종 릴레이션 스키마

개체 및 관계	릴레이션 스키마
악보 악보장르 음반참여자	악보(<u>일련번호</u> , 별점, 이름, 가격, 등록일, 판매수량, 수수료) 장르(<u>이름, 악보일련번호(FK)</u>) 음반참여자(<u>ID</u> , 구분, 이름)
악기	악기(<u>이름, 악보일련번호(FK)</u>)
판매자	판매자(<u>사업자 등록번호</u> , 총 매출액, 연락처, ID, PASSWORD, 계좌번호)
회원	회원(<u>ID</u> , PASSWORD, E-mail)
회원관심장르	관심장르(<u>이름, 회원 ID(FK)</u>)
구매하다	구매하다(<u>회원 ID(FK), 악보일련번호(FK)</u> , 구매일자)
평점을 매기다	평점을 매기다(<u>회원 ID(FK), 악보일련번호(FK)</u> , 날짜, 평점)
판매하다	판매하다(<u>판매자 등록번호(FK), 악보일련번호(FK</u>))
참여하다	참여하다(<u>악보일련번호(FK), 음반참여자 ID(FK)</u>)
검색하다	검색하다(회원 ID(FK), 악보 일련번호(FK), 날짜, 장바구니 저장 유무, WishList 저장 유무)



7. 물리적 설계 단계

(1) 내부 스키마

ⓐ 악보(일련번호, 별점, 이름, 가격, 등록일, 판매수량, 수수료)

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	7
1	일련번호	일련번호	NUMBER	10	N	PK
2	별점	별점	NUMBER	1		
3	이름	악보명	VARCHAR2	20	N	
4	가격	가격	NUMBER	10	N	
5	등록일	등록일	DATE		N	
6	판매수량	판매수량	NUMBER	6		
7	수수료	수수료	NUMBER	10	N	

★ 별점 CHECK: 0 <= 별점 <= 5

ⓑ 악보장르(이름, 악보일련번호(FK))

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	7
1	이름	이름	VARCHAR2	20	N	PK
2	악보일련번호	악보일련번호	NUMBER	10	N	PK, FK

★ 악보일련번호 FK : Reference 악보(일련번호)

ⓒ 음반참여자(ID, 구분, 이름)

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	키
1	ID	ID	VARCHAR2	20	N	PK
2	구분	구분	NUMBER	1	N	
3	이름	참여자명	VARCHAR2	20	N	

ⓓ 악기(이름, 악보일련번호(FK))

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	7
1	이름	악기명	VARCHAR2	20	N	PK
2	악보일련번호	악보일련번호	NUMBER	10	N	PK, FK

★ 악보일련번호 FK : Reference 악보(일련번호)



@ 판매자(사업자등록번호, 총매출액, 연락처, ID, PASSWORD, 계좌번호)

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	키
1	사업자등록번호	사업자등록번호	NUMBER	10	Ν	PK
2	총매출액	총매출액	NUMBER	10		
3	연락처	연락처	VARCHAR2	15	N	
4	아이디	아이디	VARCHAR2	16	Ν	
5	비밀번호	비밀번호	VARCHAR2	16	N	
6	계좌번호	계좌번호	NUMBER	15	N	

★ ID: UNIQUE

f) 회원(ID, PASSWORD, E-mail)

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	7
1	아이디	ID	VARCHAR2	16	Ν	PK
2	비밀번호	PASSWORD	VARCHAR2	16	N	
3	이메일	EMail	VARCHAR2	30	N	

★ E-mail: UNIQUE

⑨ 회원관심장르(이름, 회원ID(FK))

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	7
1	이름	이름	VARCHAR2	20	N	PK
2	회원ID	회원ID	VARCHAR2	16	N	PK, FK

★ 회원ID FK : Reference 회원(아이디)

ⓑ 구매하다(회원ID(FK), 악보일련번호(FK), 구매일자)

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	7
1	회원ID	회원ID	VARCHAR2	16	Ν	PK, FK
2	악보일련번호	악보일련번호	NUMBER	10	N	PK, FK
3	구매일자	구매일자	DATE		N	

★ 회원ID FK : Reference 회원(아이디)

★ 악보일련번호 FK : Reference 악보(일련번호)



(i) 평점을매기다(회원ID(FK), 악보일련번호(FK), 날짜, 평점)

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	키
1	회원ID	ID	VARCHAR2 16		N	PK, FK
2	악보일련번호	악보일련번호	NUMBER	10	N	PK, FK
3	날짜	날짜	DATE		N	
4	평점	평점	NUMBER	1		

★ 회원ID FK : Reference 회원(아이디)

★ 악보일련번호 FK: Reference 악보(일련번호)

① 판매하다(판매자등록번호(FK), 악보일련번호(FK))

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	키
1	판매자등록번호	판매자등록번호	NUMBER	10	N	PK, FK
2	악보일련번호	악보일련번호	NUMBER	10	N	PK, FK

★ 판매자등록번호 FK: Reference 판매자(사업자등록번호)

★ 악보일련번호 FK : Reference 악보(일련번호)

® 참여하다(악보일련번호(FK), 참여자ID(FK))

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	7
1	악보일련번호	악보일련번호	NUMBER	10	N	PK, FK
2	참여자ID	참여자ID	VARCHAR2	20	N	PK, FK

★ 악보일련번호 FK: Reference 악보(일련번호)

★ 참여자ID FK: Reference 음반참여자(ID)

① 검색하다(회원ID(FK), 악보일련번호(FK), 날짜, 장바구니, WishList)

No.	속성	칼럼이름	데이터타입	크기	NULL 허용	7
1	회원ID	회원ID	VARCHAR2	16	N	PK, FK
2	악보일련번호	악보일련번호	NUMBER	10	N	PK, FK
3	날짜	날짜	DATE		N	
4	장바구니 저장 유무	장바구니저장유무	NUMBER	1		
5	WishList 저장 유무	WishList저장유무	NUMBER	1		

★ 회원ID FK: Reference 회원(아이디)

★ 악보일련번호 FK : Reference 악보(일련번호)

★ 장바구니: 0 - 포함하지 않음, 1 - 포함

★ WishList: 0 - 포함하지 않음, 1 - 포함



- (2) 인덱스 정의서
- a 인덱스 대상 선정
- 1. 악보(일련번호, 별점, 이름, 가격, 등록일, 판매수량, 수수료)
- 2. 음반참여자(ID, 구분, 이름)
- 3. 판매자(사업자등록번호, 총매출액, 연락처, ID, PASSWORD, 계좌번호)
- 4. 회원(ID, PASSWORD, E-mail)
- 5. 구매하다(회원ID(FK), 악보일련번호(FK), 구매일자)
- 6. 평점을매기다(회원ID(FK), 악보일련번호(FK), 날짜)
- 7. 검색하다(회원ID(FK), 악보일련번호(FK), 날짜, 장바구니 저장 유무, WishList 저장 유무)

기본 키로만 구성된 테이블은 모두 제외

- (1) "회원" 테이블은 검색 대상에 해당하지 않으므로 인덱스 대상에서 제외한다.
- (2) "평점을매기다" 테이블은 검색 대상에 해당하지 않으므로 인덱스 대상에서 제외한다.
- 1. 악보(일련번호, 별점, 이름, 가격, 등록일, 판매수량, 수수료)
- 2. 음반참여자(ID, 구분, 이름)
- 3. 판매자(사업자등록번호, 총매출액, 연락처, ID, PASSWORD, 계좌번호)
- 4. 구매하다(회원ID(FK), 악보일련번호(FK), 구매일자)
- 5. 검색하다(회원ID(FK), 악보일련번호(FK), 날짜, 장바구니 저장 유무, WishList 저장 유무)
- (3) 자주 사용되는 칼럼
- ① 악보
 - 별점, 이름, 가격
- ② 참여자
 - 구분, 이름
- ③ 판매자
 - ID
- ④ 구매하다
 - 구매일자
- ⑤ 검색하다
 - 날짜, 장바구니 저장 유무, WishList 저장 유무
- (4) 평균 분포도가 10~15% 이내인 칼럼 선정
- ① 악보 : 별점(20%), 이름(1%이하), 가격(1%이하),
- ② 참여자: 구분(33%), 이름(1%이하)
- ③ 판매자 : ID(1%이하)
- ④ 구매하다 : **구매일자(1%이하)**
- ⑤ 검색하다 : **날짜(1%이하)**, 장바구니(50%), WishList(50%)



2) 인덱스 최적화

(1) 인덱스의 효율성 검토

- 인덱스로 설정한 칼럼들이 모두 자주 수정되거나 삭제되지 않으므로 효율성은 좋다고 볼 수 있다.

(2) 인덱스의 데이터타입 검토

- '구매하다'와 '검색하다'테이블의 구매일자, 날짜는 DATE 타입으로 정의했으나, 시, 분, 초까지 처리하지 않고 날짜 단위로 처리하므로 VARCHAR2(8) 형식으로 변경

(3) 인덱스 정렬 및 결합 인덱스 검토

① 악보: 이름(ASC), 가격(ASC)

② 참여자 : 이름(ASC)③ 판매자 : ID(ASC)

④ 구매하다 : **구매일자(DESC)** ⑤ 검색하다 : **날짜(DESC)**

3) 인덱스 정의서 작성

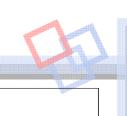
개체 타입명	테이블명	인덱스명	칼럼명	데이터 타입	인덱스 유형 (UNIQUE 여부)	정렬
악보 악보		name_idx	name	VARCHAR2(20)	No	ASC
7 -	7 -	price_idx	price	VARCHAR2(10)	No	ASC
참여자	참여자	name_idx	name	VARCHAR2(20)	No	ASC
판매자	판매자	id_idx	id	VARCHAR2(16)	Yes	ASC
구매하다	구매하다	date_idx	date	VARCHAR2(8)	No	DESC
검색하다	검색하다	date_idx	date	VARCHAR2(8)	No	DESC



8. 테이블 생성 및 초기 삽입용 스크립트 파일 내용

(1) create Table.sql

```
CREATE TABLE 악보 (
 일련번호
                 NUMBER(10),
 별점
                 NUMBER(1)
악보명
                 VARCHAR2(20) NOT NULL,
가격
                 VARCHAR2(10) NOT NULL,
 등록일
                 DATE NOT NULL.
 판매수량
                 NUMBER(6),
수수료
                 NUMBER(10) NOT NULL,
CONSTRAINT 악보 PK PRIMARY KEY (일련번호),
CONSTRAINT 별점 CK CHECK(별점 BETWEEN 0 AND 5));
INSERT INTO 악보
VALUES ('1010101', 4, 'Rolling in the deep', 500, TO_DATE('2003/4/25','yyyy/mm/dd'), 5, 500);
INSERT INTO 악보
VALUES ('2010101', 2, '길을 막지마', 1000, TO_DATE('2003/4/25','yyyy/mm/dd'), 12 ,500);
INSERT INTO 악보
VALUES ('3010101', 3, '소나기', 500, TO DATE('2003/4/25','yyyy/mm/dd'),18,600);
INSERT INTO 악보
VALUES ('4010101', 4, '크게 라디오를 켜고', 1000, TO_DATE('2003/4/25','yyyy/mm/dd'), 54, 1000);
INSERT INTO 악보
VALUES ('5010101', 1, 'Faint', 500, TO_DATE('2003/4/25','yyyy/mm/dd'), 5, 500);
INSERT INTO 악보
VALUES ('6010101', 4, '남자를 몰라', 1000, TO_DATE('2003/4/25','yyyy/mm/dd'), 67, 500);
INSERT INTO 악보
VALUES ('7010101', 5, 'She is', 1500, TO_DATE('2003/4/25','yyyy/mm/dd'), 254, 1000);
INSERT INTO 악보
VALUES ('8010101', 4, 'GoodLuck', 1500, TO DATE('2003/4/25','yyyy/mm/dd'), 63, 500);
INSERT INTO 악보
VALUES ('9010101', 4, 'Hello', 1200, TO_DATE('2003/4/25','yyyy/mm/dd'), 124, 1000);
INSERT INTO 악보
VALUES ('1010100', 4, 'Ladies', 500, TO_DATE('2003/4/25','yyyy/mm/dd'), 15, 500);
commit;
CREATE TABLE 악보장르(
이름
            VARCHAR2(20),
악보일련번호 NUMBER(10),
CONSTRAINT 악보장르 PK PRIMARY KEY (이름, 악보일련번호),
CONSTRAINT 악보장르 FK FOREIGN KEY(악보일련번호)
                REFERENCES 악보(일련번호));
INSERT INTO 악보장르 VALUES ('Pop', '1010101');
INSERT INTO 악보장르 VALUES ('가요','2010101');
INSERT INTO 악보장르 VALUES ('Pop','3010101');
INSERT INTO 악보장르 VALUES ('클래식','4010101');
INSERT INTO 악보장르 VALUES ('가요','5010101');
INSERT INTO 악보장르 VALUES ('가요','6010101');
INSERT INTO 악보장르 VALUES ('가요','7010101');
INSERT INTO 악보장르 VALUES ('가요','8010101');
INSERT INTO 악보장르 VALUES ('가요','9010101');
INSERT INTO 악보장르 VALUES ('가요','1010100');
commit;
```



```
CREATE TABLE 음반참여자(
           VARCHAR2(20).
ID
구분
           NUMBER(1),
참여자명
         VARCHAR2(20)
CONSTRAINT 음반참여자_PK PRIMARY KEY (ID));
INSERT INTO 음반참여자 VALUES ('0000001', '3', 'Adele');
INSERT INTO 음반참여자 VALUES ('0000002', '3', '다이나믹 듀오');
INSERT INTO 음반참여자 VALUES ('0000003', '3', '버스커 버스커');
INSERT INTO 음반참여자 VALUES ('0000004', '3', '시나위');
INSERT INTO 음반참여자 VALUES ('0000005', '3', 'Linkin Park');
INSERT INTO 음반참여자 VALUES ('0000006', '3', 'Buzz');
INSERT INTO 음반참여자 VALUES ('0000007', '3', '클래지콰이');
INSERT INTO 음반참여자 VALUES ('0000008', '1', '김광진');
INSERT INTO 음반참여자 VALUES ('0000009', '3', '허각');
INSERT INTO 음반참여자 VALUES ('0000010', '3', '델리스파이스');
commit:
CREATE TABLE 악기(
악기명
             VARCHAR2(20),
악보일련번호 NUMBER(10),
CONSTRAINT 악기 PK PRIMARY KEY (악기명, 악보일련번호),
CONSTRAINT 악기 FK FOREIGN KEY(악보일련번호)
                REFERENCES 악보(일련번호));
INSERT INTO 악기 VALUES ('기타','1010101');
INSERT INTO 악기 VALUES ('피아노','2010101');
INSERT INTO 악기 VALUES ('기타','3010101');
INSERT INTO 악기 VALUES ('기타','4010101');
INSERT INTO 악기 VALUES ('피아노','5010101');
INSERT INTO 악기 VALUES ('기타','6010101');
INSERT INTO 악기 VALUES ('기타','7010101');
INSERT INTO 악기 VALUES ('피아노','8010101');
INSERT INTO 악기 VALUES ('피아노','9010101');
INSERT INTO 악기 VALUES ('피아노','1010100');
commit;
CREATE TABLE 회원(
            VARCHAR2(16),
PASSWORD VARCHAR2(16) NOT NULL,
            VARCHAR2(30) NOT NULL,
CONSTRAINT 회원_PK PRIMARY KEY (ID),
CONSTRAINT 회원 UK UNIQUE(EMail));
INSERT INTO 회원 VALUES ('Ish4580', 'egwr114', 'Ish4580@hanmail.net');
INSERT INTO 회원 VALUES ('love','gwtgwt124','iop2480@naver.net');
INSERT INTO 회원 VALUES ('lyngirl','dsgsg5235','bgo2480@hanmir.net');
INSERT INTO 회원 VALUES ('hello','dfhdfh5325','number@hanmail.net');
INSERT INTO 회원 VALUES ('linkin', 'bsbqwet123', 'jkl2421@hanmail.net');
INSERT INTO 회원 VALUES ('iop2480', 'bigno23', 'klk5655@naver.net');
INSERT INTO 회원 VALUES ('kyh8890','qwrtq241','njm2141@hanmail.net');
INSERT INTO 회원 VALUES ('zzang', 'grwsfq435', 'tyu2415@hanmail.net');
INSERT INTO 회원 VALUES ('hik8580', 'ritopo422', 'nim454@naver.net');
INSERT INTO 회원 VALUES ('ert3580','grgwr231','ios9009@naver.net');
commit;
```



```
CREATE TABLE 판매자 (
사업자등록번호 NUMBER(10),
총매출액
              NUMBER(10),
연락처
              VARCHAR2(15) NOT NULL,
아이디
              VARCHAR2(16) NOT NULL.
비밀번호
              VARCHAR2(16) NOT NULL,
계촤번호
              VARCHAR2(15) NOT NULL,
CONSTRAINT 판매자 PK PRIMARY KEY (사업자등록번호),
CONSTRAINT 판매자 UK UNIQUE(아이디));
INSERT INTO 판매자
VALUES ('783784', '52000', '010-7711-4508', 'lsh4580', 'idwfqfq204', '110-264-386102');
INSERT INTO 판매자
VALUES ('883784', '52000', '010-7711-4508', 'sh4580', 'idwfq4244', '110-263-314502');
INSERT INTO 판매자
VALUES ('983784', '22000', '010-5637-4253', 'iop4580', 'iddfqfq214', '110-274-246102');
INSERT INTO 판매자
VALUES ('283784', '1000', '010-7711-4508', 'lh4580', 'idwfqfq21414', '110-245-383462');
INSERT INTO 판매자
VALUES ('763784', '60000', '010-6362-4534', 'zxc4580', 'iasfqfq5645', '110-294-334602');
INSERT INTO 판매자
VALUES ('783782', '100000', '010-6786-1242', 'xcv4580', 'idwfqfq7457', '110-256-375726');
INSERT INTO 판매자
VALUES ('783783', '2000', '010-8084-4508', 'bnm4580', 'idwfqfq4573', '110-214-253622');
INSERT INTO 판매자
VALUES ('783684', '6000', '010-7474-4508', 'fer4580', 'idwfqfq5235', '110-245-252352');
INSERT INTO 판매자
VALUES ('785784', '8000', '010-8282-4508', 'gwe4580', 'idwfqfq2352',' 110-254-526432');
INSERT INTO 판매자
VALUES ('782784', '9000', '010-9292-4508', 'ijb4580', 'idwfqf3452', '110-145-646736');
commit;
CREATE TABLE 회원관심장르(
이름
       VARCHAR2(20),
회원ID VARCHAR2(16),
CONSTRAINT 회원관심장르 PK PRIMARY KEY (이름, 회원ID),
CONSTRAINT 회원관심장르 FK FOREIGN KEY(회원ID)
                REFERENCES 회원(ID));
INSERT INTO 회원관심장르 VALUES ('Pop', 'Ish4580');
INSERT INTO 회원관심장르 VALUES ('가요','love');
INSERT INTO 회원관심장르 VALUES ('Pop','lyngirl');
INSERT INTO 회원관심장르 VALUES ('클래식','hello');
INSERT INTO 회원관심장르 VALUES ('가요','linkin');
INSERT INTO 회원관심장르 VALUES ('가요','iop2480');
INSERT INTO 회원관심장르 VALUES ('가요','kyh8890');
INSERT INTO 회원관심장르 VALUES ('가요','zzang');
INSERT INTO 회원관심장르 VALUES ('가요','hjk8580');
INSERT INTO 회원관심장르 VALUES ('가요','ert3580');
commit;
```



```
CREATE TABLE 구매하다(
회원ID
           VARCHAR2(16),
악보일련번호 NUMBER(10),
구매일자
           DATE NOT NULL.
CONSTRAINT 구매하다 PK PRIMARY KEY (회원ID,악보일련번호),
CONSTRAINT 구매하다 FK1 FOREIGN KEY(악보일련번호)
                REFERENCES 악보(일련번호),
CONSTRAINT 구매하다 FK2 FOREIGN KEY(회원ID)
                REFERENCES 회원(ID));
INSERT INTO 구매하다 VALUES ('Ish4580', '1010101', TO_DATE('2003/4/15', 'yyyy/mm/dd'));
INSERT INTO 구매하다 VALUES ('love', '2010101', TO DATE('2003/4/10', 'yyyy/mm/dd'));
INSERT INTO 구매하다 VALUES ('lyngirl', '4010101', TO_DATE('2003/4/11', 'yyyy/mm/dd'));
INSERT INTO 구매하다 VALUES ('linkin', '5010101', TO_DATE('2003/4/13', 'yyyy/mm/dd'));
INSERT INTO 구매하다 VALUES ('hello', '5010101', TO DATE('2003/4/14', 'yyyy/mm/dd'));
INSERT INTO 구매하다 VALUES ('iop2480', '6010101', TO DATE('2003/4/16','yyyy/mm/dd'));
INSERT INTO 구매하다 VALUES ('zzang', '7010101', TO DATE('2003/4/17','yyyy/mm/dd'));
INSERT INTO 구매하다 VALUES ('hjk8580', '5010101', TO_DATE('2003/4/23','yyyy/mm/dd'));
INSERT INTO 구매하다 VALUES ('kyh8890', '9010101', TO_DATE('2003/4/13','yyyy/mm/dd'));
INSERT INTO 구매하다 VALUES ('ert3580', '5010101', TO_DATE('2003/4/13','yyyy/mm/dd'));
commit;
CREATE TABLE 평점을매기다(
           VARCHAR2(16),
악보일련번호 NUMBER(10),
날짜
           DATE NOT NULL,
평점
           NUMBER(1),
CONSTRAINT 평점을매기다 PK PRIMARY KEY (ID,악보일련번호),
CONSTRAINT 평점을매기다 FK1 FOREIGN KEY(악보일련번호)
                REFERENCES 악보(일련번호),
CONSTRAINT 평점을매기다 FK2 FOREIGN KEY(ID)
                REFERENCES 회원(ID));
INSERT INTO 평점을매기다 VALUES ('Ish4580', '1010101', TO DATE('2003/4/15','yyvy/mm/dd'), '5');
INSERT INTO 평점을매기다 VALUES ('love', '2010101', TO_DATE('2003/4/10','yyyy/mm/dd'), '5');
INSERT INTO 평점을매기다 VALUES ('lyngirl', '4010101', TO DATE('2003/4/11','yyyy/mm/dd'), '5');
INSERT INTO 평점을매기다 VALUES ('linkin', '5010101', TO_DATE('2003/4/13','yyyy/mm/dd'), '4');
INSERT INTO 평점을매기다 VALUES ('hello', '5010101', TO_DATE('2003/4/14','yyyy/mm/dd'), '2');
INSERT INTO 평점을매기다 VALUES ('iop2480', '6010101', TO_DATE('2003/4/16','yyyy/mm/dd'), '5');
INSERT INTO 평점을매기다 VALUES ('zzang', '7010101', TO_DATE('2003/4/17','yyyy/mm/dd'), '4');
INSERT INTO 평점을매기다 VALUES ('hjk8580', '5010101', TO_DATE('2003/4/23','yyyy/mm/dd'), '3');
INSERT INTO 평점을매기다 VALUES('kyh8890', '9010101', TO DATE('2003/4/13','yyyy/mm/dd'), '4');
INSERT INTO 평점을매기다 VALUES ('ert3580', '5010101', TO_DATE('2003/4/13','yyyy/mm/dd'), '1');
commit:
```



```
CREATE TABLE 판매하다(
판매자등록번호 NUMBER(10),
악보일련번호
              NUMBER(10).
CONSTRAINT 판매하다 PK PRIMARY KEY (판매자등록번호, 악보일련번호),
CONSTRAINT 판매하다 FK1 FOREIGN KEY(판매자등록번호)
                REFERENCES 판매자(사업자등록번호)
CONSTRAINT 판매하다 FK2 FOREIGN KEY(악보일련번호)
                REFERENCES 악보(일련번호));
INSERT INTO 판매하다 VALUES ('763784','1010101');
INSERT INTO 판매하다 VALUES ('783782','1010101');
INSERT INTO 판매하다 VALUES ('783783','1010101');
INSERT INTO 판매하다 VALUES ('785784','2010101');
INSERT INTO 판매하다 VALUES ('783684','1010101');
INSERT INTO 판매하다 VALUES ('782784','1010101');
INSERT INTO 판매하다 VALUES ('783784'.'5010101'):
INSERT INTO 판매하다 VALUES ('983784','1010101');
INSERT INTO 판매하다 VALUES ('883784','7010101');
INSERT INTO 판매하다 VALUES ('283784','1010101');
commit:
CREATE TABLE 참여하다(
악보일련번호 NUMBER(10),
참여자ID
           VARCHAR2(20),
CONSTRAINT 참여하다 PK PRIMARY KEY (악보일련번호, 참여자ID),
CONSTRAINT 참여하다 FK1 FOREIGN KEY(악보일련번호)
                REFERENCES 악보(일련번호),
CONSTRAINT 참여하다 FK2 FOREIGN KEY(참여자ID)
               REFERENCES 음반참여자(ID));
INSERT INTO 참여하다 VALUES ('1010101','0000001');
INSERT INTO 참여하다 VALUES ('2010101','0000002');
INSERT INTO 참여하다 VALUES ('3010101','0000003');
INSERT INTO 참여하다 VALUES ('4010101','0000004');
INSERT INTO 참여하다 VALUES ('5010101','00000005');
INSERT INTO 참여하다 VALUES ('6010101','0000006');
INSERT INTO 참여하다 VALUES ('7010101','0000007');
INSERT INTO 참여하다 VALUES ('8010101','00000008');
INSERT INTO 참여하다 VALUES ('9010101','0000009');
INSERT INTO 참여하다 VALUES ('1010100','0000010');
commit:
CREATE TABLE 검색하다(
회원ID VARCHAR2(16),
악보일련번호 NUMBER(10),
날짜 DATE NOT NULL,
장바구니저장유무 NUMBER(1),
WishList저장유무 NUMBER(1),
CONSTRAINT 검색하다 PK PRIMARY KEY (회원ID,악보일련번호),
CONSTRAINT 검색하다 FK1 FOREIGN KEY(악보일련번호)
                REFERENCES 악보(일련번호)
CONSTRAINT 검색하다 FK2 FOREIGN KEY(회원ID)
                REFERENCES 회원(ID));
INSERT INTO 검색하다 VALUES ('Ish4580','1010101',TO_DATE('2003/4/15','yyyy/mm/dd'),0,1);
INSERT INTO 검색하다 VALUES ('love','2010101',TO_DATE('2003/4/10','yyyy/mm/dd'),1,1);
INSERT INTO 검색하다 VALUES ('lyngirl','4010101',TO_DATE('2003/4/11','yyyy/mm/dd'),0,1);
INSERT INTO 검색하다 VALUES ('linkin','5010101',TO_DATE('2003/4/13','yyyy/mm/dd'),1,1);
INSERT INTO 검색하다 VALUES ('hello','5010101',TO_DATE('2003/4/14','yyyy/mm/dd'),0,1);
INSERT INTO 검색하다 VALUES ('iop2480','6010101',TO_DATE('2003/4/16','yyyy/mm/dd'),1,1);
INSERT INTO 검색하다 VALUES ('zzang','7010101',TO DATE('2003/4/17','yyyy/mm/dd'),1,1);
INSERT INTO 검색하다 VALUES ('hjk8580','5010101',TO_DATE('2003/4/23','yyyy/mm/dd'),0,1);
INSERT INTO 검색하다 VALUES ('kyh8890','9010101',TO_DATE('2003/4/13','yyyy/mm/dd'),1,1);
INSERT INTO 검색하다 VALUES ('ert3580','5010101',TO_DATE('2003/4/13','yyyy/mm/dd'),1,0);
commit:
```



(2) drop_Table.sql

```
ALTER TABLE 악보
   DROP PRIMARY KEY CASCADE;
ALTER TABLE 악보장르
   DROP PRIMARY KEY CASCADE;
ALTER TABLE 음반참여자
   DROP PRIMARY KEY CASCADE:
ALTER TABLE 악기
   DROP PRIMARY KEY CASCADE;
ALTER TABLE 판매자
   DROP PRIMARY KEY CASCADE;
ALTER TABLE 회원
   DROP PRIMARY KEY CASCADE;
ALTER TABLE 회원관심장르
   DROP PRIMARY KEY CASCADE;
ALTER TABLE 구매하다
   DROP PRIMARY KEY CASCADE;
ALTER TABLE 평점을매기다
   DROP PRIMARY KEY CASCADE;
ALTER TABLE 판매하다
   DROP PRIMARY KEY CASCADE;
ALTER TABLE 참여하다
   DROP PRIMARY KEY CASCADE;
ALTER TABLE 검색하다
   DROP PRIMARY KEY CASCADE:
DROP TABLE 악보장르;
DROP TABLE 음반참여자;
DROP TABLE 악기;
DROP TABLE 회원관심장르;
DROP TABLE 구매하다;
DROP TABLE 평점을매기다;
DROP TABLE 판매하다;
DROP TABLE 참여하다;
DROP TABLE 검색하다;
DROP TABLE 회원:
DROP TABLE 판매자;
DROP TABLE 악보;
```



9. 트랜잭션 처리를 위한 스크립트 파일 내용 및 화면 캡쳐

- (1) 회원 정보 삽입.sql
 - 회원정보를 삽입하기 전에 회원 정보를 출력하는 '00.회원_정보_리스트.sql'을 실행합니다.

```
SET FEEDBACK OFF
TTITLE '회원 정보 리스트'
                                                       '회원 ID'
COLUMN
                 id
                                     HEADING
                                                      '비밀번호'
COLUMN
                 password
                                     HEADING
                                                                              FORMAT A15
                                                       '이메일 주소'
COLUMN
                  email
                                    HEADING
                                                                             FORMAT A25
                  이름
                                                      '관심장르'
COLUMN
                                    HEADING
                                                                      FORMAT 99999
select 회원.id, 회원.password, 회원.email, 회원관심장르.이름 from 회원, 회원관심장르
where 회원.id=회원관심장르.회원id
order by id
TTITLE OFF
CLEAR COLUMN
<실행 결과>
일 6월 09
                                                              회원 전보 리스트
                                                                                                 MINIXI 1
                                  비밀번호
                                                                           미메일 주소
                                                                                                                    관심장르
ert3580
                      qrqwr231
dfhdfh5325
                                                    ios9009@naver.net
number@hanmail.net
njm454@naver.net
klk5655@naver.net
njm2141@hanmail.net
iop2480@naver.net
lsh4580@hanmail.net
                                                                                                          글래식
                      ritopo422
bigno23
qwrtq241
                                                                                                          가요
가요
가요
가요
hjk8580
kyh8890
                      bsbgwet123
qwtqwt124
eqwr114
                                                                                                          가요
                                                                                                          가요
Ish4580
                       dsgsg5235
arwsfa435
                                                     bgo2480@hanmir.net
tvu2415@hanmail.net
```

- 회원 삽입 코드인 '01.회원_정보_삽입.sql'을 실행하고 그 결과를 확인한다.

```
SET ECHO OFF
SET VERIFY OFF
define id='&회원id';
insert into 회원 values ('&id', '&비밀번호', '&이메일주소');
commit:
insert into 회원관심장르 values('&관심장르명', '&id');
undefine id:
SET VERIFY ON
SET ECHO ON
SET FEEDBACK OFF
TTITLE '회원 정보 리스트'
                                    '회원 ID'
'비밀번호'
COLUMN
           id
                         HEADING
COLUMN
            password
                        HEADING
                                                    FORMAT A15
                                    '이메일 주소'
'관심장르' F(
COLUMN
                        HEADING
            email
                                                    FORMAT A25
COLUMN
            이름
                        HEADING
                                               FORMAT 99999
select 회원.id, 회원.password, 회원.email, 회원관심장르.이름
from 회원, 회원관심장르
where 회원.id=회원관심장르.회원id
order by id
TTITLE OFF
CLEAR COLUMN
```

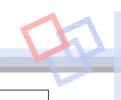


	<실	행 과정	J 및 결과>		
1. ID 삽입			2. PASSWORD 삽입		
① 입력 필요			(i) 입력 필요		
회원id의 값을 입력하십시오	qwert 12345		비밀번호의 값을 입력하십시오: 12345		
3. E-Mail 삽입			4. 관심장르 삽입		
Name of the last of the last					
(i) 입력 필요			(i)입력 필요		
이메일주소의 값을 입력하십/			관심장르명의 값을 입력하십시오: 클래식		
5. 회원 가입 후 회	의원 정보 리스트				
일 6월 09 회원 ID	비밀번호	회원 6	행보리스트 페이지 1 미메일 주소	관심장르	
ert3580	qrqwr231	ios9009@naver.net		가요	
hello	dfhdfh5325	number@hanmail.n	et	클래식	
hjk8580 iop2480	ntopo422	njm454@naver.net		가요 가요	
kyh8890	bigno23 qwrtq241	klk5655@naver.net njm2141@hanmail.		가요	
linkin bsbgwet123 jkl2421@hanmail.					
love qwtqwt124 iop2480@naver.net				가요	
Ish4580 eqwr114 Ish4580@hanmail. lyngirl dsgsg5235 bgo2480@hanmir.				Pop	
qwert12345	12345	qwert12345@naver		클래식	
zzang	qrwsfq435	tyu2415@hanmail.r		가요	
- qwert12345 계	정이 회원, 회원관심	장르 DB	에 잘 삽입 된 것을 알 수 🤉	있다.	

(2) 회원 정보 변경.sql

- 위의 리스트에서 회원 'qwert12345'의 정보를 변경하려고 한다.
- '02.회원_정보_변경.sql'을 실행하고 결과를 확인한다.

```
SET ECHO OFF
SET VERIFY OFF
define id='&회원id';
update 회원 set password='&비밀번호', email='&이메일주소' where id='&id';
commit;
update 회원관심장르 set 이름='&관심장르명' where 회원id='&id';
undefine id='&id';
SET VERIFY ON
SET ECHO ON
TTITLE '회원 정보 리스트'
                                     '회원 ID'
'비밀번호' FORMAT A15
'이메일 주소' FORMAT A25
'관심장르' FORMAT 99999
COLUMN
           id
                          HEADING
COLUMN
                         HEADING
            password
COLUMN
            email
                         HEADING
            이름
COLUMN
                         HEADING
select 회원.id, 회원.password, 회원.email, 회원관심장르.이름
from 회원, 회원관심장르
where 회원.id=회원관심장르.회원id
order by id
TTITLE OFF
CLEAR COLUMN
```



	<실형	행 과정 및 결과>				
1. ID 선택		2 DASSWORD 변경 /기조	2. PASSWORD 변경 (기존 : 12345)			
		CONTRACTOR AND ADVANCED	CONTRACTOR OF THE CONTRACTOR O			
(i)입력 필요		(i)입력 필요	(i) 입력 필요			
MAJ ST						
회원id의 값을 입력하십시오	gwert12345	비밀번호의 값을 입력하십시오: modify				
	4.0.1.120.10					
2 г Mail HI 74 /712	<u> </u>					
3. E-IVIdII 원경 (기급	존 : qwert12345@nave	^{er.com)} 4. 관심장르 변경 (기존 : 를	·래식)			
(i) 입력 필요		(i) 입력 필요				
W B I E E						
미메일주소의 값을 입력하십시	I오: modify@naver.com	관심장르명의 값을 입력하십시오: 가요	관심장르명의 값을 입력하십시오: 가요			
5. 회원 변경 후 회	원 정보 리스트	·				
	_ = -	회원 정보 리스트 페이지 1				
회원 ID	비밀번호	O[메일 주소	관심장르			
ert3580 hello	grqwr231 dfhdfh5325	ios9009@naver.net number@hanmail.net	가요 클래식			
hjk8580	ritopo422	njm454@naver.net	가요			
iop2480	bigno23	klk5655@naver.net	가요			
kyh8890	qwrtq241	njm2141@hanmail.net	가요			
linkin	bsbgwet123	jkl2421@hanmail.net	가요			
lsh4580	qwtqwt124	iop2480@naver.net	가요 Pop			
10000000000000000000000000000000000000	egwr114	Ish4580@hanmail.net	Pop			
lyngirl qwert12345	dsgsg5235 modify	bgo2480@hanmir.net modify@naver.com	가요			
zzang	grwsfq435	tyu2415@hanmail.net	가요			
	정의 비밀번호, E-ma	il, 관심장르 값이 변경 된 것을 알	수 있다.			

(3) 회원 정보 삭제.sql

- 위의 리스트에서 회원 'qwert12345'의 정보를 삭제하려고 한다.
- '03.회원_정보_삭제.sql'을 실행하고 결과를 확인한다.

```
SET ECHO OFF
SET VERIFY OFF
define id='&id';
delete from 회원관심장르 where 회원id='&id';
commit:
delete from 검색하다 where 회원id='&id';
commit;
delete from 검색하다 where 회원id='&id';
commit;
delete from 평점을매기다 where id='&id';
commit;
delete from 회원 where id='&id';
undefine id;
SET VERIFY ON
SET ECHO ON
TTITLE '회원 정보 리스트'
                                       '회원 ID'
'비밀번호' FORMAT A1
'이메일 주소' FORMAT A2
'관심장르' FORMAT 99999
COLUMN
                          HEADING
            id
                                                       FORMAT A15
FORMAT A25
COLUMN
            password
                          HEADING
COLUMN
                         HEADING
            email
            이름
COLUMN
                          HEADING
select 회원.id, 회원.password, 회원.email, 회원관심장르.이름
from 회원, 회원관심장르
where 회원.id=회원관심장르.회원id
order by id
TTITLE OFF
CLEAR COLUMN
```



<실행 과정 및 결과>									
1. ID 선택									
j)입력 필요	2								
	2	.9							
www.man.u.s.ess.									
2. 회원 삭제 후 ^{일 6월 09}	회원 정보 리스트	회원 정보 리스트	期01八 1						
2. 회원 삭제 후 ^{일6월 09}	후 회원 정보 리스트	회원 정보 리스트 이메일 주소	관심장르						
. 회원 삭제 후 ^{16월 09} ^{회원 ID}	후 회원 정보 리스트 비밀번호	회원 정보 리스트 미메일 주소 ios9009@naver.net	관심장로 가요						
. 회원 삭제 후 일월 09 ^{회원 ID} n3530	후 회원 정보 리스트 비밀변호 qqqv231 dhdh6325	회원 정보 리스트 이해일 주소 ios9009@naver.net number@hanmail.net	관심장로 가요 클래식						
2. 회원 삭제 후 ^{회원 ID} ^{회원 ID} ^{회원 ID} ^{회원 ID} (No.8580	후 회원 정보 리스트 ####################################	회원 정보 리스트 이때일 주소 ias9009@naver.net number@hanmail.net njm454@naver.net	관심장르 가요 클래식 가요						
. 회원 삭제 후 일 6월 09 제월 ID vr3580 sello _{sp2480}	후 회원 정보 리스트 비밀변호 qrqwr231 dfldff65325 ritopo422 bigno23	회원 정보 리스트 Of배일 주소 ios9009@naver net numbe@hammail.net njm454@naver.net klk5656@naver net	관심장로 가요 클래식 가요 가요						
. 회원 삭제 후 ^{12 6월 09} ¹³⁵⁸⁰ ₁₈₆₈₀ ₁₉ 2480 _y 188890	후 회원 정보 리스트 바일번호 qrqwr231 ritop422 bigno23 qwtq241	회원 정보 리스트 ios9009@naver net number@hanmail net njm454@naver.net HJK5656pnaver.net njm2141@hanmail net	관심장로 가요 클래식 가요 가요 가요 가요						
2. 회원 삭제 후 일 6월 09 mr3580 mr3580 pp2480 pp2480 mrkin	후 회원 정보 리스트 ####################################	회원 정보 리스트 ios9009@naver net numbe@hammail.net njm454@naver.net HIS665-@naver.net HIS665-@naver.net HIS6424(@hammail.net J#2424(@hammail.net	관심장로 가요 클래식 가요 가요 가요 가요 가요						
2. 회원 삭제 후 일 6월 09 회원 ID nd3580 njk8580 npcp480 nyh8890 nrikin	후 회원 정보 리스트 바일번호 qrqwr231 dhdh65325 ritopo422 bigno23 qwrtq241 babgwet123 qwrtqv124	회원 정보 리스트 ios9009@naver net numbe@hammail.net njm454@naver.net klk5656@naver.net njm2141@hammail.net iop2430@naver.net	관심장로 가요 클래식 가요 가요 가요 가요 가요 가요 가요 가요						
일 6월 09	후 회원 정보 리스트 ####################################	회원 정보 리스트 ios9009@naver net numbe@hammail.net njm454@naver.net HIS665-@naver.net HIS665-@naver.net HIS6424(@hammail.net J#2424(@hammail.net	관심장로 가요 클래식 가요 가요 가요 가요 가요						

(4) 회원_관심장르검색.sql

- 등록된 회원 안에 관심 장르가 일치하는 회원에게 관심 장르에 맞는 메일을 검색할 수 있다.
- 보안상의 이유로 비밀번호는 출력하지 않는다.
- '04.회원_관심장르검색.sql을 실행하고, 관심장르가 '가요'인 회원을 검색한 뒤 결과를 확인한다.

```
TTITLE '회원 관심장르 검색 List'
COLUMN 회원ID HEADING '회원 ID' FORMAT A10
COLUMN 이름 HEADING 관심장르 FORMAT 99999
COLUMN EMail HEADING '이메일 주소' FORMAT A20

SELECT 회원관심장르.회원ID, 회원관심장르.이름, 회원.EMail
FROM 회원관심장르, 회원
WHERE 회원관심장르.이름 = '&관심장르명'
AND 회원관심장르.회원ID = 회원.ID
ORDER BY 회원ID
/
TTITLE OFF
CLEAR COLUMN
```





(5) 악보 이름검색.sal

- 등록된 악보 중에 찾는 악보가 있는지 이름을 통해 검색 할 수 있다.
- '05.악보 이름검색.sql을 실행하고, 악보 이름이 '소나기'인 악보를 검색한 후 값을 확인한다.

```
TTITLE '악보 이름 검색 List'
COLUMN 악보명 HEADING 악보 FORMAT A20
COLUMN 하여자명 HEADING '작곡가/작사자/가수' FORMAT A30
COLUMN 참여자명 HEADING '작곡가/작사자/가수' FORMAT A30
COLUMN 당록일 FORMAT A20
COLUMN 등록일 FORMAT A20
COLUMN 등록일 FORMAT A20
COLUMN 판매수량 HEADING '판매 수'

SELECT 악보.악보명, 악보장르.이름, 음반참여자.참여자명, 악기.악기명, 악보.가격, 악보.등록일, 악보.판매수량, 악보.별점
FROM 악보. 악보장르, 음반참여자, 참여하다, 악기
WHERE 악보명 = '&악보명'
AND 악보장르.악보일련번호 = 악보.일련번호
AND 참여하다.착여자ID = 음반참여자.ID
AND 참여하다.악보일련번호 = 악보.일련번호
ORDER BY 판매수량
/

TTITLE OFF
CLEAR COLUMN
```

			<실행 고	과정 및	결과>			
1. 검색힐	날 악보 (이름 '소나기'	를 입력 한다.					
<u>(</u>) 일	력 필요	2						
악보명의 같	났을 입력하	십시오: 소나기						
2. 악보	이름이 '	소나기'인 악브	보의 정보를 획	학인할 수	있다.			
월 6월 10				악보 검색 List		페이지 1		
약보	장르		H'작사자/가수	막기	가격	등록말	판매 수	별점 2
소나기	Pop Pop	버스커 버스커	1/3小小刀子	기타	500	03/04/25	판매 주 18	명집 3

(6) 악기 이름검색.sal

- 등록된 악보 중에 찾는 악보가 있는지 악기를 통해 검색 할 수 있다.
- '06.악기_이름검색.sql을 실행하고, 악기 이름이 '기타'인 악보를 검색한 후 값을 확인한다.

```
TTITLE '악보 검색 List'
COLUMN 악보명 HEADING 악보 FORMAT A20
COLUMN 이름 HEADING 장르 FORMAT A20
COLUMN 참여자명 HEADING '작곡가/작사자/가수' FORMAT A30
COLUMN 악기명 HEADING '악기'
COLUMN 등록일 FORMAT A20
COLUMN 판매수량 HEADING '판매 수'

SELECT 악보.악보명, 악보장르.이름, 음반참여자.참여자명, 악기.악기명, 악보.가격, 악보.등록일, 악보.판매수량, 악보.별점
FROM 악보, 악보장르, 음반참여자, 참여하다, 악기
WHERE 악기명 = '&악기명'
AND 악보장르.악보일련번호 = 악보.일련번호
AND 참여하다.참여자ID = 음반참여자.ID
AND 참여하다.악보일련번호 = 악보.일련번호
AND 악기.악보일련번호 = 악보.일련번호
ORDER BY 판매수량
/
TTITLE OFF
CLEAR COLUMN
```



(7) 악보 및 악기 이름검색.sal

- 등록된 악보의 이름과 악기를 입력받아 검색할 수 있게 한다.
- '07.악보_및_악기_이름검색.sql을 실행하고, 악보 '남자를 몰라', 악기 '기타'인 악보를 검색한 후에 결과를 확인한다.

```
TTITLE '악보 검색 List'

COLUMN 악보명 HEADING 악보 FORMAT A20
COLUMN 이름 HEADING '작곡가/작사자/가수' FORMAT A30
COLUMN 참여자명 HEADING '작곡가/작사자/가수' FORMAT A30
COLUMN 악기명 HEADING '작곡가/작사자/가수' FORMAT A30
COLUMN 등록일 FORMAT A20
COLUMN 판매수량 HEADING '판매 수'

SELECT 악보.악보명, 악보장르.이름, 음반참여자.참여자명, 악기.악기명, 악보.가격, 악보.등록일, 악보.판매수량, 악보 별점
FROM 악보, 악보장르, 음반참여자, 참여하다, 악기
WHERE 악보명 = '&악보명'
AND 악기명 = '&악기명'
AND 악기명 = '&악기명'
AND 참여하다.착여자ID = 음반참여자.ID
AND 참여하다.악보일련번호 = 악보.일련번호
AND 참여하다.악보일련번호 = 악보.일련번호
ORDER BY 판매수량
/

TTITLE OFF
CLEAR COLUMN
```





10. 팀원들의 프로젝트 소감

(1) 0000000 김00

1, 2, 3 학년 수업을 들어오면서 팀장을 맡아온 적이 손에 꼽을 정도였고, 발표력, 리더쉽 등이 부족하다고 느껴서 데이터베이스 수업을 시작으로 팀장을 맡아보는 경험을 의도적으로 시작했습니다. 항상 팀원의 입장에서 팀 활동을 해왔기 때문인지, 어떻게 팀을 이끌어야 하고, 혹은 팀을 지원해야 하는지 잘모르는 상태에서 데이터베이스 실습 및 팀 프로젝트를 시작했고, 벌써 최종 보고서를 쓰고, 소감문을 쓰게 되는 시간이 찾아왔습니다. 물론 DB를 처음 배우는 입장인지라 모든 팀원이 하나하나 각각의 프로젝트 단계를 밟아가는 과정에서 많은 시행착오를 겪었습니다. 하지만 모든 팀원이 한명도 빠짐없이 열심히 이 팀 프로젝트에 참여함으로서 그 시행착오들을 차근차근 풀어나갔고, 지금에 다다른 것 같습니다.

사실 처음 데이터베이스를 배우기 시작할 때, 왜 데이터베이스를 배우는 것이고, 왜 이를 사용하는 것인 지 잘 이해하지 못했지만, 인터넷 강의와 교수님의 열혈한 강의 덕에 데이터베이스에 관한 내용을 거의 꽤 차게 되었고, 지금은 데이터베이스를 사용함은 물론 왜 필요한지 절실히 느끼고 있습니다. 뿐만 아니라 항상 중요하게 생각했던 설계의 중요성! 이번 데이터베이스 수업과 팀 프로젝트를 통해서 한번 더되새김질 할 기회가 되었습니다.

이러한 속담이 있습니다. "입에 쓴 약이 몸에 좋다". 이를 잘 살펴보면 시행착오를 겪으면서 어려운 문제들을 해결해 나가고, 시간이 부족함에도 이를 해결하고 일어서면 결국 무엇이든 의미 없는 시간은 없었다는 것을 의미하는 것 같습니다. 처음 팀장을 맡아 부족함에도 불구하고 서로의 격려와 협력을 통해도와준 팀원들이 너무나도 고맙고, 데이터베이스에 대한 기본적인 내용을 꾀찰 수 있도록 도와주신 교수님께도 정말 감사하다는 말을 전하면서 소감을 마칩니다.

(2) 2008160056 문찬호

이번 한 학기동안 '데이터베이스 및 실습' 수업을 통해 많은 것을 배웠습니다.

첫 번째로는 이야기로만 듣던 '데이터베이스'에 대해 알 수 있었습니다. 학기가 시작하기 전에도 '데이터베이스'에 대해서 인터넷에서나 지인들에게 들었지만 이번 학기 수업만큼 자세하게 공부하기는 어려웠다고 생각합니다.

두 번째로는 단순히 수업으로 그치지 않고 '팀 활동'을 통해 간단하게 설계해보면서 많은 것을 배웠다고 생각합니다. 그리고 '팀 활동' 내에서의 경험도 소중하다고 생각합니다. 프로젝트를 할 때는 힘들었지만, 지나고 나서 생각해보니 좋은 기억이 많이 남았던 시간이었습니다. 이런 경험을 토대로 앞으로 팀활동을 하는데 있어서 이번 학기의 경험이 많은 도움이 될 것 같습니다.

세 번째로는 '프로 마인드'입니다. 교수님께서 수업시간에 늘 강조하신 내용이 와 닿았습니다. 앞으로 남은 학부 생활뿐만 아니라, 어떤 직장에서도 '나는 프로다'라는 마인드를 가지고 '프로'다운 일 처리를 할 수 있도록 노력할 것입니다.

위에서 언급한 이야기에 말은 못했지만 여러 가지로 많은 것들을 배울 수 있었고, 자신에 대해 깊이 생각할 수 있었던 한 학기였습니다. 교수님 감사합니다.



(3) 0000000 0100

B318에서 처음 '데이터베이스' 수업을 들었을 때, 교수님께서 많이 힘들 것이라고 말씀하셔서 처음에는 조금 겁먹었습니다. 하지만, 수업을 들으면서 우리가 해야 할 일이 많은 만큼, 교수님께서 신경을 많이 써주신다는 생각이 들었습니다. 매주 내는 보고서에 피드백을 9개 조 모두 성의껏 해주시는 모습을 보면서 그런 생각이 들었습니다. 그리고 이번 DB수업에서 가장 좋았던 것은 팀 활동을 통해서 얻는 게 많았다는 것입니다. 처음엔 팀 활동이 순조롭게 잘 이루어지다가 어느 순간부터 조금 힘들어진 때가 있었습니다. 그럴 때 마다 어떻게 하면 잘 극복해 나갈 수 있을까 많은 고민을 하게 되었고, 극복하면서 '나 자신이 많이 발전했다'라고 느꼈습니다. 그리고 팀원들이 저보다 다들 잘해서 제가 모르는 것이 있을 때는 많이 물어보면서, DB에 대해 부족한 지식을 좀 더 보충할 수 있었습니다. 마지막으로, 제 친구중에 이제 2학년 애들이 많은데, 데이터베이스를 듣고 싶어 하는 친구가 있으면 교수님 데이터베이스 수업을 추천하고 싶습니다. 한 학기 동안 정말 수고하셨습니다. 진심으로 감사드립니다.

(4) 0000000 홍0000000

학기 전 겨울 방학 때 산학협력관에서 교육받으면서 데이터베이스라는 것을 먼저 실무에서 공부해보고 직접 써보기도 하면서 어떤 것인지 알게 되었습니다. 단순히 실무용으로 데이터베이스 sql문을 배우다 보니, 이전부터 데이터베이스에 관심이 있었기도 하고 실무용 데이터베이스가 아닌 데이터베이스의 이 론에서는 어떤 것을 배우는지 알고 싶어 신청하게 되었습니다. 수업을 들으면서 실무 쪽에서 배울 때 컨설턴트나 클라이언트들이 미리 데이터베이스를 설계해서 나눠주면 단순히 이를 보고 코딩했던 것과는 달리 그 것을 직접 해보면서 진짜 데이터베이스가 어떤 것이고, 데이터베이스를 보는 폭이 넓어진 것 같습니다.

단순히 전문가가 설계해주었던 것을 코딩할 때와는 달리 설계라는 것이 상당히 고차원적이고, 최종 스 크립트까지 짜보면서 요구분석이 상당히 중요하다는 것도 깨달을 수 있었습니다.

지금까지 학교에서 해왔던 팀 활동 중에서 가장 팀원들이 열심히들 해주시고, 교수님도 팀 활동 결과물에 대해 피드백을 매주 잘해주셔서 많은 것을 배울 수 있었습니다. 감사합니다.