

[데이터베이스 구축 사례]
청년 음악가 후원 웹사이트를 위한 DB 구축
(최종 보고서)

목차

1. 주제 소개	3
2. 요구 수집 및 분석	4
3. 개념적 설계	15
4. 논리적 설계	30
5. 물리적 설계	39
6. 구현 및 테스트	50
7. 별첨	78

1. 주제 소개¹

목차로

(1) 목적

청년 음악가를 소개하고 후원자를 연결해 주며, 공연, 시설물 대여, 강의, 중고거래 등의 기능을 통해 청년 음악가의 수익 창출 및 홍보 기회를 제공해 주는 청년 음악가 후원 웹사이트를 구축하는 데 필요한 데이터베이스를 설계 및 구현한다.

(2) 주요 기능

청년 음악가 후원 웹사이트에서 다음과 같은 기능을 제공할 수 있어야 한다.

- ① 청년 음악가가 본인의 곡과 앨범을 등록해서 홍보할 수 있는 기능
- ② 공연을 기획해서 개최할 수 있는 기능
- ③ 공연장 및 연습실 대여 기능
- ④ 회원 등급을 고려한 시설물 대여 기능
- ⑤ 중고물품을 등록해서 회원끼리 상호 거래하는 기능
- ⑥ 일반회원을 대상으로 청년 음악가가 음악강의를 개설할 수 있는 기능
- ⑦ 활동 점수에 따라 회원 등급이 자동 갱신되는 기능
- ⑧ 후원하고 싶은 청년 음악가의 멤버십 가입을 통해서 정기 후원할 수 있는 기능
- ⑨ 청년 음악가의 멤버십에 가입한 회원에게 멤버십 혜택을 제공하는 기능
- ⑩ 청년 음악가를 지정하여 비정기 후원할 수 있는 기능
- ⑪ 공연 기획 및 앨범 발매 비용 마련을 위한 펀딩 기능
- ⑫ 청년 음악가의 곡과 앨범의 월별 조회 순위를 제공하는 기능
- ⑬ 시설물 대여나 중고물품 거래 후 평점 남기는 기능 등

1 : 이 내용은 데이터베이스설계 교과목을 수강한 메모리팀(팀원 여승준, 강원용, 조원석, 심규재, 최수연)의 팀 프로젝트 최종보고서를 정리한 것으로, 일부 내용에 오류가 있을 수 있습니다.

2. 요구 수집 및 분석

[목차로](#)

(1) 요구 수집 명세서

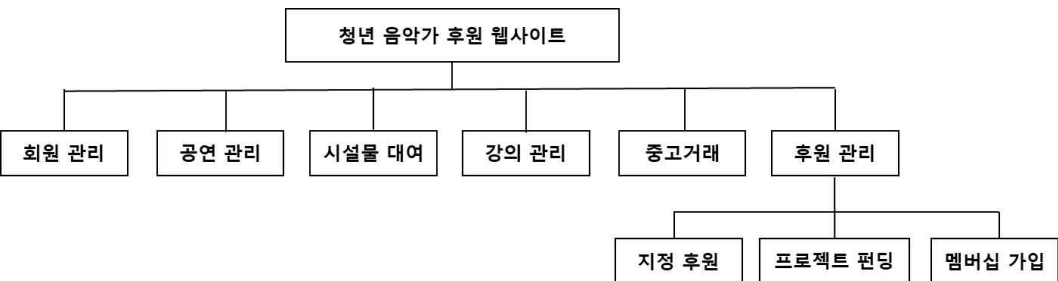
청년 음악가 후원 웹사이트를 위한 DB 구축을 위해 수집한 요구사항은 <표 1>과 같다.

<표 1> 청년 음악가 후원 웹사이트의 DB 구축을 위한 요구 수집 명세서

No	요구 수집 내용	수집 일자
1	운영책임자인 김철수님과 웹사이트 운영 등에 대해 인터뷰함 [별첨 1] 운영책임자와 인터뷰 내용 요약	2024.06.20
2	시설물 대여관리자인 안대영님과 시설물 대여업무 흐름 등에 대해 인터뷰함 [별첨 2] 시설물 대여관리자와 인터뷰 내용 요약	2024.06.20
3	후원 기능이 있는 기존 플랫폼 조사 [별첨 3] 후원 기능 관련 플랫폼 조사 및 분석	2024.06.20
4	시설물 대여 기능이 있는 기존 플랫폼 조사 [별첨 4] 시설물 대여 관련 플랫폼 조사 및 분석	2024.06.21
5	크라우드 펀딩 기능이 있는 기존 플랫폼 조사 [별첨 5] 크라우드 펀딩 플랫폼 조사 및 분석	2024.06.21
6	중고 거래 기능이 있는 기존 플랫폼 조사 [별첨 6] 중고 거래 플랫폼 조사 및 분석	2024.06.22
7	온라인 강의 기능이 있는 기존 플랫폼 조사 [별첨 7] 온라인 강의 플랫폼 조사 및 분석	2024.06.22

(2) 업무 영역 분할도

수집한 요구사항을 기초로 업무 영역을 [그림 1]과 같이 분할하였다.



[그림 1] 청년 음악가 후원 웹사이트의 DB 구축을 위한 업무 영역 분할도

(3) 업무별 데이터 및 트랜잭션 요구 분석 명세서

[그림 1]의 각 업무 영역과 관련한 데이터 및 트랜잭션 요구 분석 결과는 다음과 같다.

1) 회원 관리 업무를 위한 데이터 및 트랜잭션 요구 분석 명세서

<표 2> 회원 관리 업무 관련 데이터 요구 분석 명세서

No.	요구 분석 내용
1	회원 가입 시 id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호 정보가 필요하다.
2	회원은 일반회원, 음악가회원, 업체회원으로 구분된다.
3	음악가회원은 학력, 작업실주소 정보가 추가로 필요하다.
4	학력은 중졸, 고졸, 대졸로 구분한다.
5	업체회원은 id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 사업자등록번호, 업체명 정보가 필요하다.
6	업체회원은 회원 가입 시 자신이 소유한 시설물을 등록할 수 있다.
7	일반회원과 음악가회원은 접속내역, 등급 정보를 갖는다.
8	회원 등급에는 VWIP, VIP, GOLD, SILVER, BEGIN 등급으로 구분한다.
9	활동점수가 250점 이상이면 VWIP등급, 200점 이상이면 VIP등급, 150점 이상이면 GOLD등급, 100점 이상이면 SILVER등급, 50점 이상이면 BEGIN 등급이 된다.
10	일반회원은 포인트를 가진다.
11	일반회원과 음악가회원은 자신의 프로필을 가질 수 있다.
12	프로필에는 이름, 생년월일, 등급, 평판점수, 공개여부 정보가 필요하다.
13	회원의 활동점수와 평판점수의 초기값은 0점이다.
14	활동점수는 강의 수강, 강의 개설, 공연 기획, 공연 참여, 프로젝트 기획, 프로젝트 펀딩, 지정 후원, 멤버십 가입을 통해 10점씩 획득한다.
15	회원이 한 달 이상 접속하지 않으면 활동점수를 10점씩 감소한다.
16	평판점수는 한 번의 중고거래를 통해 최대 5점, 시설물 대여를 통해 최대 20점까지 감소될 수 있다.
17	평판점수가 10점씩 차감될 때마다 활동점수 5점씩 감소한다.
18	활동점수와 평판점수의 최하 점수는 0점이다.
19	활동점수가 10점씩 증가할 때마다 평판점수는 5점씩 증가한다.
20	평판점수가 0점이 되면 영구적으로 대여 및 거래가 불가능하다.
21	음악가회원은 앨범을 발매할 수 있다.
22	각 앨범은 고유한 번호를 가지며, 제목, 발매일, 수록 곡, 월별조회수, 누적조회수 정보가 필요하다.
23	각 수록곡은 고유한 번호를 가지며, 제목, 장르, 작곡일, 월별조회수, 누적조회수 정보가 필요하다.
24	음악가회원이 수록곡을 작곡할 수 있다.
25	장르는 힙합, 발라드, 클래식, 트로트, 재즈, 락, K-POP, 댄스, 성악, 블루스, 락발라드, 인디로 구분한다.

<표 3> 회원 관리 업무 관련 트랜잭션 요구 분석 명세서

No.	요구 분석 내용
1	새로 가입한 회원 정보를 삽입한다.
2	회원의 활동점수에 따라 등급이 변경된다.
3	회원은 프로필을 공개한 다른 회원의 프로필을 검색할 수 있다.
4	회원은 계좌정보를 변경할 수 있다.
5	회원은 프로필 정보 공개여부를 변경할 수 있다.
6	회원은 다른 회원의 평판점수를 검색할 수 있다.
7	음악가회원은 자신의 앨범과 곡 정보를 삽입할 수 있다.
8	음악가회원은 자신의 앨범과 곡 정보를 삭제할 수 있다.
9	음악가회원은 자신의 앨범과 곡 정보를 변경할 수 있다.
10	일반회원은 음악가회원으로 변경될 수 있다.
11	회원은 음악가회원이 발매한 앨범을 검색할 수 있다.
12	회원은 음악가회원이 등록한 곡을 검색할 수 있다.
13	회원은 누적조회수 가장 높은 곡을 수록하는 앨범을 검색할 수 있다.
14	회원은 희망하는 장르의 곡을 누적조회수 높은 순으로 검색할 수 있다.
15	회원은 앨범을 누적조회수 높은 순으로 검색할 수 있다.
16	회원은 월별조회수 가장 높은 앨범에 수록된 곡을 검색할 수 있다.
17	회원은 월별조회수 높은 순으로 곡을 검색할 수 있다.
18	회원은 등급이 높은 음악가회원의 곡을 누적조회수 높은 순으로 검색할 수 있다.
19	회원은 어떤 곡을 작곡한 음악가회원을 검색할 수 있다.
20	회원은 발매한 앨범 수의 순으로 음악가회원을 검색할 수 있다.
21	매월 1일 앨범과 곡의 월별조회수 0으로 변경한다.
22	관리자는 평판점수가 0점인 회원을 검색할 수 있다.
23	관리자는 마지막으로 접속한 지 30일이 지난 회원을 검색할 수 있다.

2) 공연 관리 업무를 위한 데이터 및 트랜잭션 요구 분석 명세서

<표 4> 공연 관리 업무 관련 데이터 요구 분석 명세서

No.	요구 분석 내용
1	각 공연은 고유한 번호를 가지며, 제목, 장르, 개최일시, 마감일시, 시간, 모집인원수, 최소관람객수, 최대관람객수, 가격, 관람등급 정보가 필요하다.
2	회원은 공연장을 대여해서 공연을 개최할 수 있다.
3	공연은 VIP 등급 이상의 회원만 기획할 수 있다.
4	회원이 공연을 기획할 때, 기획일시 정보가 필요하다.
5	공연기획 시 최대 관람객 수는 공연장의 최대 좌석 수를 넘을 수 없다.
6	공연이 끝나면 공연 기획자는 공개/비공개 여부를 결정할 수 있다.
7	회원은 공연을 관심 목록에 담을 수 있다.
8	회원은 공연을 관람하기 위해 예매할 수 있다.
9	회원이 공연을 예매할 때 예약 일시와 수량 정보가 필요하다.
10	회원은 공연 참여 신청을 할 수 있다.
11	회원이 공연 참여 신청하면, 신청일시와 참여역할, 승인여부 정보가 필요하다.
12	공연 장르는 '힙합', '발라드', '클래식', '트로트', '재즈', '락', 'K-POP', '댄스', '성악', '블루스', '락발라드', '인디', '기타'로 구분한다.

<표 5> 공연 관리 업무 관련 트랜잭션 요구 분석 명세서

No.	요구 분석 내용
1	회원은 자신이 개최하는 공연 정보를 삽입할 수 있다.
2	회원은 자신이 개최하는 공연 정보를 변경할 수 있다.
3	회원은 자신이 개최하는 공연 정보를 삭제할 수 있다.
4	회원은 공연 정보를 검색할 수 있다.
5	회원은 공연을 관심 목록에 삽입할 수 있다.
6	회원은 공연을 관심 목록에서 삭제할 수 있다.
7	회원은 잔여 좌석 수가 적은 순으로 공연을 검색할 수 있다.
8	공연 기획자는 회원의 공연 참여 신청을 승인할 수 있다.
9	공연 기획자는 회원의 공연 참여 신청을 거절할 수 있다.
10	공연 기획 시 회원의 활동점수를 자동으로 10점 증가한다.
11	공연 참여 시 회원의 활동점수를 자동으로 10점 증가한다.
12	공연이 끝나면 기획자는 공개/비공개 여부를 변경할 수 있다.
13	관리자는 일자별로 공연을 검색할 수 있다.
14	관리자는 일자별로 공연 예매 현황을 검색할 수 있다.
15	회원은 장르별로 가장 최근에 기획된 순으로 공연을 검색할 수 있다.
16	회원은 희망하는 기획자의 공연을 가격이 낮은 순으로 검색할 수 있다.
17	회원은 지역별로 가장 최근에 기획된 순으로 공연을 검색할 수 있다.

3) 시설물 대여 업무를 위한 데이터 및 트랜잭션 요구 분석 명세서

<표 6> 시설물 대여 업무 관련 데이터 요구 분석 명세서

No.	요구 분석 내용
1	각 시설물은 고유한 시설물 번호를 가지며, 등급, 대여제한 등급 정보가 필요하다.
2	업체회원은 시설물을 등록할 때 시설물명과 주소 정보가 필요하다.
3	시설물은 공연장과 연습실로 구분한다.
4	VIP 등급 이상 회원만 공연장을 대여할 수 있다.
5	SILVER 등급 이상 회원만 연습실을 대여할 수 있다.
6	회원이 시설물 대여를 신청할 때 신청일시, 시작일시, 종료일시, 사용자수 정보가 필요하다.
7	업체회원은 여러 개의 시설물을 등록할 수 있다.
8	시설물의 등급은 크기별로 A, B, C 등급으로 나뉜다.
9	공연장의 최대 좌석 수는 A등급의 경우 2500명, B등급의 경우 1000명, C등급의 경우 500명이며 등급별 최대 좌석 수의 5%는 휠체어석이다.
10	연습실의 최대 수용 인원은 A등급의 경우 70명, B등급의 경우 30명, C등급의 경우 10명으로 제한되어 있다.
11	회원이 시설물 대여 신청하면, 관리자는 회원의 등급과 평판점수를 기초로 승인 여부를 결정한다.
12	회원은 시설물 대여 시 공연장 하나, 연습실 하나 최대 두 개의 시설물만 동시에 대여할 수 있다.
13	하나의 공연장 대여 중에는 다른 공연장을 대여할 수 없다.
14	하나의 연습실 대여 중에는 다른 연습실을 대여할 수 없다.
15	대여 기간은 최대 한 달을 넘을 수 없다.
16	시설물 대여 승인 과정에서 회원의 평판점수가 0점이면 승인 요청이 거절된다.
17	회원의 대여 신청 내용 변경 및 신청 취소는 관리자가 승인 여부를 결정하기 전까지만 가능하다.
18	회원이 시설물을 반납한 후 청결 상태를 평가한다.
19	청결 상태는 P, D, E, F등급으로 구분해서 평가한다. (P등급: 시설물을 청결하게 사용하고 업체 물건이 손상되지 않은 경우 D등급: 시설물에 설치된 장비 중 하나가 손상되었으나 고의성이 없는 경우 E등급: 시설물에 설치된 장비 두 개 이상 손상되었으나 고의성이 없는 경우 또는 시설물을 불결하게 사용한 경우 F등급: 시설물에 설치된 장비 중 하나 이상을 고의로 망가뜨린 경우)
20	청결 상태에 따라 회원의 평판점수를 가감한다. (F등급: 20점 감소, E등급: 10점 감소, D등급: 5점 감소, P등급: 5점 증가)

<표 7> 시설물 대여 업무 관련 트랜잭션 요구 분석 명세서

No.	요구 분석 내용
1	회원은 관리자에게 시설물 대여를 신청할 수 있다.
2	회원이 시설물 대여 신청 시 신청일시, 시작일시, 종료일시, 사용자수 정보가 삽입된다.
3	회원은 시설물 대여 신청이 승인되기 전까지 시작일시, 종료일시, 사용자수 정보를 변경할 수 있다.
4	회원은 시설물 대여 신청이 승인되기 전까지 신청을 취소할 수 있다.
5	업체회원은 시설물 정보를 삽입할 수 있다.
6	업체회원은 시설물 정보를 변경할 수 있다.
7	업체회원은 더 이상 존재하지 않는 시설물 정보를 삭제할 수 있다.
8	회원은 시설물을 시설물의 등급별로 검색할 수 있다.
9	회원은 시설물을 지역별로 검색할 수 있다.
10	회원은 본인의 등급을 기준으로 대여할 수 있는 시설물을 지역별로 검색할 수 있다.
11	회원은 월별 가장 많이 대여된 시설물을 검색할 수 있다.
12	회원은 현재 대여 가능한 시설물을 검색할 수 있다.
13	회원은 현재 대여 가능한 시설물 중 가장 많이 대여된 시설물 순으로 검색할 수 있다.
14	관리자는 시설물 대여를 신청한 회원의 평판점수를 검색할 수 있다.
15	관리자는 시설물 대여를 신청한 회원의 등급을 검색할 수 있다.
16	관리자는 시설물 대여를 신청한 회원 중 대여 기간이 한 달을 초과한 회원을 검색할 수 있다.
17	관리자는 회원의 시설물 대여 신청을 승인할 수 있다.
18	관리자는 회원의 시설물 대여 신청을 거절할 수 있다.
19	관리자는 회원이 대여한 시설물 상태에 따라 청결상태의 등급을 삽입할 수 있다.
20	시설물 반납 후 청결 상태 평가 결과에 따라 해당 시설물을 대여한 회원의 평판점수가 변경된다.
21	관리자는 월별로 가장 많이 대여된 시설물을 검색할 수 있다.
22	관리자는 시설물의 등급을 검색할 수 있다.
23	관리자는 현재 시설물을 대여 중인 모든 회원을 검색할 수 있다.

4) 강의 관리 업무를 위한 데이터 및 트랜잭션 요구 분석 명세서

<표 8> 강의 관리 업무 관련 데이터 요구 분석 명세서

No.	요구 분석 내용
1	음악가회원은 하나 이상의 음악강의를 개설할 수 있다.
2	일반회원은 하나 이상의 음악강의를 수강 신청할 수 있다.
3	일반회원이 수강 신청한 음악강의는 개설한 음악가회원이 승인해야 수강할 수 있다.
4	일반회원이 음악강의를 수강 신청할 때는 신청일시 정보가 필요하다.
5	음악강의는 고유한 번호를 가지며, 강의명, 내용, 장소, 개강일시, 종강일시, 강의요일, 시작시각, 종료시각, 수강료, 대면여부, 수강정원 정보가 필요하다.
6	강의요일은 '월', '화', '수', '목', '금', '토' 가운데 하나의 값을 갖는다.
7	대면여부는 '대면' 또는 '비대면'으로 구분한다.
8	일반회원은 음악강의를 수강할 때 보유한 할인쿠폰과 포인트를 사용해서 할인받을 수 있다.
9	음악가회원은 본인의 음악강의에 대해 하나 이상의 할인쿠폰을 발행할 수 있다.
10	할인쿠폰은 고유한 번호를 가지며, 할인율과 발행일시, 유효기간 정보가 필요하다.
11	일반회원은 여러 할인쿠폰을 보유할 수 있다.

<표 9> 강의 관리 업무 관련 트랜잭션 요구 분석 명세서

No.	요구 분석 내용
1	음악가회원은 자신이 개설한 음악강의 정보를 삽입할 수 있다.
2	음악가회원은 자신이 개설한 음악강의 정보를 삭제할 수 있다.
3	음악가회원은 자신이 개설한 음악강의 정보를 변경할 수 있다.
4	일반회원은 음악강의를 수강 신청할 수 있다.
5	음악가회원은 일반회원이 수강 신청한 음악강의의 수강을 승인할 수 있다.
6	일반회원은 음악강의 수강신청 정보를 음악가회원이 승인하기 전까지만 삭제할 수 있다.
7	음악가회원은 자신이 개설한 음악강의를 수강하는 일반회원을 검색할 수 있다.
8	음악가회원은 자신이 개설한 음악강의에 대한 할인쿠폰을 삽입할 수 있다.
9	음악가회원은 자신이 개설한 음악강의에 대한 할인쿠폰을 삭제할 수 있다.
10	음악가회원은 자신이 개설한 음악강의에 대한 할인쿠폰을 변경할 수 있다.
11	일반회원은 특정 음악강의에 적용할 수 있는 할인쿠폰을 검색할 수 있다.
12	일반회원은 자신이 보유하고 있는 할인쿠폰을 검색할 수 있다.
13	일반회원이 사용한 할인쿠폰은 자동 삭제된다.
14	발행일로부터 유효기간이 지난 할인쿠폰은 자동으로 삭제된다.
15	음악가회원이 강의를 개설하면 활동점수를 자동으로 10점 증가시킨다.
16	회원이 음악강의를 수강하면 활동점수를 자동으로 10점 증가시킨다.
17	일반회원은 수강료가 낮은 순으로 음악강의를 검색할 수 있다.
18	일반회원은 수강정원이 많이 남은 순으로 음악강의를 검색할 수 있다.
19	일반회원은 할인율이 높은 순서대로 음악강의를 검색할 수 있다.
20	일반회원은 특정 요일에 개설되는 음악강의를 검색할 수 있다.

5) 중고거래 업무를 위한 데이터 및 트랜잭션 요구 분석 명세서

<표 10> 중고거래 업무 관련 데이터 요구 분석 명세서

No.	요구 분석 내용
1	회원은 하나 이상의 중고물품을 판매할 수 있다.
2	중고물품은 고유한 번호를 가지며, 물품명, 카테고리, 판매가, 구매가, 물품상태, 조회수 정보가 필요하다.
3	중고물품의 카테고리는 '관악기', '타악기', '국악기', '현악기', '기타 물품'으로 구분한다.
4	회원이 중고물품을 판매할 때는 등록일시, 배송비, 판매진행상태, 거래유형 정보가 필요하다.
5	중고물품의 판매진행상태는 '판매중', '예약중', '판매취소', '판매완료'로 구분한다.
6	거래유형은 '직거래'와 '택배거래'로 구분한다.
7	회원만 중고물품을 구매할 수 있다.
8	회원이 중고물품을 구매할 때는 구매일시, 배송지, 구매확정여부 정보가 필요하다.
9	회원은 하나 이상의 중고물품을 찜할 수 있다.
10	중고거래 후 구매자와 판매자는 서로의 평판점수를 한 번에 0점 ~ 5점까지 평가할 수 있다.
11	중고거래 후 평판점수를 남길 때 이유도 함께 남길 수 있다.
12	거래 후 평가된 평판점수는 회원의 프로필에 등록된 평판점수에 합산되어 저장된다.

<표 11> 중고거래 업무 관련 트랜잭션 요구 분석 명세서

No.	요구 분석 내용
1	회원은 판매할 중고물품 정보를 삽입할 수 있다.
2	회원은 자신이 등록한 중고물품 정보를 변경할 수 있다.
3	회원은 자신이 등록한 중고물품 정보를 삭제할 수 있다.
4	회원은 등록된 중고물품을 검색할 수 있다.
5	회원은 자신이 찜한 중고물품 목록을 검색할 수 있다.
6	회원은 신뢰할 수 있는 중고거래를 위해서 다른 회원의 평판점수를 검색할 수 있다.
7	회원이 중고물품을 조회할 때마다 자동으로 해당 물품의 조회수가 1씩 증가된다.
8	회원은 조회수가 높은 순으로 중고물품을 검색할 수 있다.
9	회원은 최신 등록일 순으로 중고물품을 검색할 수 있다.
10	회원은 가격이 낮은 순으로 중고물품을 검색할 수 있다.
11	회원은 찜한 회원이 많은 순으로 중고물품을 검색할 수 있다.
12	중고물품 구매회원이 구매를 결정하면 자동으로 판매진행상태가 '예약중'으로 변경된다.
13	거래를 완료한 후에 회원이 상대 회원의 평판점수를 평가하면, 자동으로 상대 회원의 평판점수가 변경된다.
14	거래 완료되면 거래한 회원의 활동점수는 자동으로 5점 증가된다.
15	관리자는 일/월/연도 별로 가장 많이 판매된 중고물품 카테고리를 검색할 수 있다.
16	관리자는 평판점수가 높은 회원이 등록한 중고물품을 검색할 수 있다.
17	회원은 평판점수가 높은 회원이 등록한 중고물품을 검색할 수 있다.

6) 지정 후원 업무를 위한 데이터 및 트랜잭션 요구 분석 명세서

<표 12> 지정 후원 업무 관련 데이터 요구 분석 명세서

No.	요구 분석 내용
1	일반회원은 음악가회원을 지정해서 후원할 수 있다.
2	음악가회원을 후원한 일반회원의 활동점수는 10점 증가된다.
3	음악가회원을 후원할 때 금액과 후원일시 정보가 필요하다.
4	후원금액은 웹사이트를 통해서 음악가회원에게 전달된다.

<표 13> 지정 후원 업무 관련 트랜잭션 요구 분석 명세서

No.	요구 분석 내용
1	음악가회원이 후원을 받으면 후원내역이 자동으로 변경된다.
2	음악가회원은 자신이 후원받은 총 후원누적금액을 검색할 수 있다.
3	음악가회원은 자신에게 후원한 일반회원을 검색할 수 있다.
4	음악가회원은 자신에게 후원한 일반회원 중 후원 횟수가 많은 회원 순으로 검색할 수 있다.
5	음악가회원은 자신에게 후원한 일반회원 중 후원 금액이 많은 회원 순으로 검색할 수 있다.
6	일반회원은 자신이 음악가회원들에게 후원한 총 후원누적금액을 검색할 수 있다.
7	일반회원은 자신이 특정 음악가회원에게 후원한 후원누적금액을 검색할 수 있다.
8	일반회원은 일별로 자신이 후원한 금액을 검색할 수 있다.
9	일반회원은 자신의 후원내역을 최신순으로 검색할 수 있다.
10	관리자는 음악가회원에게 후원한 회원목록을 검색할 수 있다.
11	관리자는 월별로 가장 많은 후원을 받은 음악가회원을 검색할 수 있다.
12	관리자는 음악가회원별로 후원누적금액을 검색할 수 있다.
13	음악가회원에게 후원한 일반회원의 활동점수는 10점 증가된다.

7) 프로젝트 펀딩 업무를 위한 데이터 및 트랜잭션 요구 분석 명세서

<표 14> 프로젝트 펀딩 업무 관련 데이터 요구 분석 명세서

No.	요구 분석 내용
1	회원은 프로젝트를 기획할 수 있다.
2	각 프로젝트는 고유한 번호를 가지며, 제목, 분야, 세부설명, 목표후원금액 정보가 필요하다.
3	프로젝트를 기획할 때 시작일자, 종료일자 정보가 필요하다.
4	회원은 특정 프로젝트를 펀딩할 수 있다.
5	프로젝트를 펀딩할 때 금액과 일시 정보가 필요하다.
6	프로젝트를 펀딩하는 최소 금액은 10,000원이다.
7	프로젝트 분야는 '곡'과 '앨범', '공연'으로 구분한다.
8	일반회원은 프로젝트 분야 중 공연만 프로젝트를 기획할 수 있다.
9	현재 진행 중인 프로젝트는 삭제할 수 없다.

<표 15> 프로젝트 펀딩 업무 관련 트랜잭션 요구 분석 명세서

No.	요구 분석 내용
1	회원은 자신이 기획하는 프로젝트 정보를 삽입한다.
2	프로젝트 종료 이후 10년이 지난 프로젝트는 자동으로 삭제한다.
3	회원은 자신이 기획한 프로젝트의 정보를 변경할 수 있다.
4	회원은 진행 중인 프로젝트를 검색할 수 있다.
5	회원은 자신이 기획한 프로젝트에 펀딩한 회원들을 검색할 수 있다.
6	회원은 현재 진행 중인 프로젝트 가운데 누적펀딩금액이 가장 큰 프로젝트를 검색할 수 있다.
7	회원은 자신이 기획한 프로젝트 정보를 검색할 수 있다.
8	회원은 자신이 펀딩한 프로젝트 정보를 검색할 수 있다.
9	관리자는 월별로 누적펀딩금액이 가장 큰 프로젝트 정보를 검색할 수 있다.
10	회원이 기획한 프로젝트가 완료되면 회원의 활동점수는 10점 증가시킨다.
11	프로젝트에 펀딩한 회원의 활동점수를 10점 증가시킨다.
12	회원은 종료일자 순으로 프로젝트를 검색할 수 있다.
13	회원은 특정 분야에 속하는 프로젝트를 검색할 수 있다.
14	회원은 특정 분야에 속하는 프로젝트를 높은 목표후원금액 순으로 검색할 수 있다.
15	회원은 펀딩된 금액이 목표후원금액에 가까운 순으로 프로젝트를 검색할 수 있다.

8) 멤버십 가입 업무를 위한 데이터 및 트랜잭션 요구 분석 명세서

<표 16> 멤버십 가입 업무 관련 데이터 요구 분석 명세서

No.	요구 분석 내용
1	음악가회원은 자신의 멤버십을 등록할 수 있다.
2	멤버십을 등록할 때 등록일자와 가입금액, 제공포인트 정보가 필요하다.
3	각 멤버십은 고유한 번호를 갖는다.
4	일반회원은 특정 음악가회원의 멤버십에 가입할 수 있다.
5	멤버십에 가입할 때 가입일자 정보가 필요하다.
6	일반회원이 멤버십에 가입하면 가입일시로부터 매달 한 번씩 자동으로 멤버십 결제가 된다.
7	멤버십에 가입한 일반회원은 매달 한 번씩 멤버십별 제공포인트가 지급된다.
8	멤버십에 가입한 일반회원에게 멤버십의 제공포인트가 매달 지급된다.
9	멤버십에 가입한 일반회원은 해당 멤버십을 등록한 음악가회원의 공연을 선예매할 수 있다.
10	멤버십에 가입한 일반회원은 해당 멤버십을 등록한 음악가회원의 특정 강의를 무료로 볼 수 있다.
11	일반회원이 멤버십에 가입하면 활동점수가 매달 10점씩 증가한다.
12	음악가회원이 멤버십을 등록하면 활동점수가 매달 10점씩 증가한다.
13	음악가회원이 멤버십을 삭제하면 활동점수는 20점 감소한다.

<표 17> 멤버십 가입 업무 관련 트랜잭션 요구 분석 명세서

No.	요구 분석 내용
1	음악가회원은 자신의 멤버십 정보를 삽입할 수 있다.
2	음악가회원은 자신의 멤버십 정보를 삭제할 수 있다.
3	음악가회원은 자신의 멤버십 정보를 변경할 수 있다.
4	멤버십을 등록한 음악가회원의 활동점수를 자동으로 변경(증가)한다.
5	멤버십에 가입한 일반회원의 활동점수를 자동으로 변경(증가)한다.
6	멤버십을 삭제한 음악가회원의 활동점수를 자동으로 변경(감소)한다.
7	멤버십에 가입한 일반회원에게 제공되는 포인트를 자동으로 변경(증가)한다.
8	음악가회원은 자신의 멤버십에 가입한 일반회원을 검색할 수 있다.
9	관리자는 가장 많은 멤버십 회원을 보유한 음악가회원을 검색할 수 있다.
10	일반회원은 보유 회원수가 많은 순으로 음악가회원을 검색할 수 있다.
11	일반회원은 자신이 가입한 멤버십 목록을 검색할 수 있다.
12	일반회원은 제공포인트를 많이 주는 멤버십을 최신순으로 검색할 수 있다.
13	일반회원은 가입금액이 낮은 순으로 멤버십을 검색할 수 있다.

3. 개념적 설계

목차로

(1) 개념적 스키마 모델링

1) 개체 후보 식별

데이터 요구 분석 명세서에서 주로 주어나 목적어로 표현된 개체 후보를 식별한 결과는 <표 18>과 같다.

<표 18> 개체 후보 목록

no.	개체 후보	no.	개체 후보	no.	개체 후보
1	회원	12	수록곡	23	음악강의
2	일반회원	13	장르	24	할인쿠폰
3	음악가회원	14	공연	25	포인트
4	업체회원	15	공연장	26	중고물품
5	학력	16	관심목록	27	중고물품카테고리
6	프로필	17	공연기획자	28	판매진행상태
7	회원등급	18	시설물	29	거래유형
8	활동점수	19	연습실	30	프로젝트
9	접속내역	20	대여제한등급	31	프로젝트분야
10	평판점수	21	시설물등급	32	멤버십
11	앨범	22	청결상태		

2) 개체 후보 검토

각 개체 후보의 실제 데이터 수와 대표 속성을 파악해서 개체 선정 여부를 결정한 결과는 <표 19>와 같다.

<표 19> 개체 후보 검토

No	개체 후보	데이터 수	주요 속성	비고	개체 선정 여부
1	회원	N	id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호		O
2	일반회원	N	id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호, 평판점수, 활동점수, 등급		O
3	업체회원	N	id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호, 사업자등록번호, 업체명		O
4	음악가회원	N	id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호, 평판점수, 활동점수, 등급		O

			곡, 앨범, 학력, 작업실주소		
5	학력	-	(없음)	속성이 없음	X
6	프로필	N	이름, 생년월일, 등급, 평판점수, 공개여부	공개여부 속성을 제외한 나머지 속성이 일반회원 및 음악가 회원과 동일	X
7	회원등급	-	(없음)	속성이 없음	X
8	활동점수	-	(없음)	속성이 없음	X
9	접속내역	N	순서, 접속일시	회원의 약한개체	O
10	평판점수	-	(없음)	속성이 없음	X
11	앨범	N	번호, 제목, 발매일, 누적조회수, 월별조회수		O
12	수록곡	N	번호, 제목, 장르, 작곡일, 누적조회수, 월별조회수		O
13	장르	-	(없음)		O
14	공연	N	번호, 제목, 장르, 개최일시, 마감일시, 시간, 모집인원수, 최소관람객수, 최대관람객수, 가격, 관람등급		O
15	공연장	N	번호, 시설물명, 주소, 등급, 대여제한등급	시설물과 동일	X
16	관심목록		번호, 공연제목	관계로 표현	X
17	공연기획자	N	(없음)	회원과 동일	X
18	시설물	N	번호, 시설물명, 주소, 등급, 대여제한등급		O
19	연습실	N	번호, 시설물명, 주소, 등급, 대여제한등급	시설물과 동일	X
20	대여제한등급	-	(없음)	속성이 없음	X
21	시설물등급	-	(없음)	속성이 없음	X
22	청결상태	-	(없음)	속성이 없음	X
23	음악강의	N	번호, 강의명, 내용, 장소, 개강일시, 종강일시, 강의요일, 시작시각, 종료시각, 수강료, 대면여부, 수강정원		O
24	할인쿠폰	N	번호, 할인율, 발행일시, 유효기간		O
25	포인트	-	(없음)	일반회원 개체의 속성	X
26	중고물품	N	번호, 물품명, 카테고리, 판매가, 구매가, 물품상태, 조회수		O
27	물품카테고리	5	(없음)		X
28	판매진행상태	4	(없음)		X
29	거래유형	-	(없음)	속성이 없음	X
30	프로젝트	N	번호, 제목, 분야, 세부설명, 목표후원금액, 시작일자, 종료일자		O
31	프로젝트분야	-	(없음)	속성이 없음	X
32	멤버십	N	번호, 등록일시, 가입금액, 제공포인트		O
35	등록일시	-	(없음)	속성이 없음	X
36	결제여부	-	(없음)	속성이 없음	X

3) 개체 선정

식별된 개체 후보를 검토해서 속성이 없거나 관계로 표현하는 것이 더 적합한 것 등을 제외하고
선정한 개체는 <표 20>과 같다.

<표 20> 개체 목록

No	개체	데이터수	주요 속성	비고	개체 선정 여부
1	회원	N	id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호	일반회원과 음악가 회원의 슈퍼 개체	O
2	일반회원	N	id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호, 접속기록, 평판점수, 활동점수, 등급, 포인트, 프로필공개여부	포인트, 프로필공개 여부 속성 추가	O
3	업체회원	N	id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호, 사업자등록번호, 업체명		O
4	음악가회원	N	id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호, 접속기록, 평판점수, 활동점수, 등급, 학력, 작업실주소, 프로필공개여부	곡, 앨범 속성 제거 (관계로 표현) 프로필공개여부 속 성 추가	O
5	앨범	N	번호, 제목, 발매일, 누적조회수, 월별조회수		O
6	곡	N	번호, 제목, 장르, 작곡일, 누적조회수, 월별조회수	개체 이름(수록곡)을 변경함	O
7	접속내역	N	순서, 접속일시	회원의 약한개체	O
8	공연	N	번호, 제목, 장르, 개최일시, 마감일시, 시간, 모집인원수, 최소관람객수, 최대관람객수, 가격, 관람등급		O
9	시설물	N	번호, 시설물명, 주소, 등급, 대여제한등급, 시설물구분	공연장과 연습실 구 분을 위한 시설물구 분 속성 추가	O
10	음악강의	N	번호, 강의명, 내용, 장소, 개강일시, 종강일시, 강의요일, 시작시각, 종료시각, 수강료, 대면여부, 수강정원		O
11	할인쿠폰	N	번호, 할인율, 발행일시, 유효기간		O
12	중고물품	N	번호, 물품명, 카테고리, 판매가, 구매가, 물품상태, 조회수		O
15	프로젝트	N	번호, 제목, 분야, 세부설명, 목표후원금액, 시작일자, 종료일자		O
17	멤버십	N	번호, 등록일자, 가입금액, 제공포인트		O

4) 관계 후보 식별

데이터 요구 분석 명세서에서 주로 동사로 표현된 관계 후보를 검토한 결과는 <표 21>과 같다.

<표 21> 관계 후보 검토

No.	관계 후보	관련된 개체 목록	참고	비고	선정 여부
1	접속하다 (식별관계)	회원, 접속내역	<표 2>의 7번		O
2	발매하다	음악가회원, 앨범	<표 2>의 21번		O
3	수록하다	앨범, 곡	<표 2>의 22번		O
4	작곡하다	음악가회원, 곡	<표 2>의 24번		O
5	대여하다	회원, 시설물	<표 4>의 2번	12번과 동일함	X
6	공연_개최하다	회원, 공연	<표 4>의 2번		O
7	공연_기획하다	회원, 공연	<표 4>의 3번	6번과 중복됨	X
8	관심_등록하다	회원, 공연	<표 4>의 7번		O
9	예매하다	회원, 공연	<표 4>의 8번		O
10	참여_신청하다	회원, 공연	<표 4>의 10번		O
11	시설물_등록하다	업체회원, 시설물	<표 6>의 2번		O
12	대여_신청하다	회원, 시설물	<표 6>의 6번		O
13	개설하다	음악가회원, 음악강의	<표 8>의 1번		O
14	수강_신청하다	일반회원, 음악강의	<표 8>의 2번		O
15	발행하다	할인쿠폰, 음악강의	<표 8>의 9번		O
16	소유하다	일반회원, 할인쿠폰	<표 8>의 11번		O
17	판매하다	회원, 중고물품	<표 10>의 1번		O
18	구매하다	회원, 중고물품	<표 10>의 7번		O
19	찜하다	회원, 중고물품	<표 10>의 9번		O
20	평판점수_매기다	회원, 중고물품	<표 10>의 10번		O
21	후원하다	일반회원, 음악가회원	<표 12>의 1번		O
22	기획하다	회원, 프로젝트	<표 14>의 1번		O
23	편당하다	회원, 프로젝트	<표 14>의 4번		O
24	멤버십_등록하다	음악가회원, 멤버십	<표 16>의 1번		O
25	가입하다	일반회원, 멤버십	<표 16>의 3번		O

5) 관계 식별 목록

<표 21>에서 선정된 관계 후보에 대해 관계 유형과 관계 설명 속성까지 결정하면 <표 22>와 같다.

<표 22> 관계 목록

No.	관계	관련된 개체 목록	관계 유형	관계 설명 속성
1	접속하다 (식별관계)	회원, 접속내역	1:N	-
2	발매하다	음악가회원, 앨범	1:N	-
3	수록하다	앨범, 곡	M:N	-
4	작곡하다	음악가회원, 곡	1:N	-
5	공연 개최하다	회원, 공연	1:N	기획일, 추후공개여부
6	관심_등록하다	회원, 공연	M:N	
7	예매하다	회원, 공연	M:N	예매일시, 수량
8	참여_신청하다	회원, 공연	M:N	신청일시, 역할, 승인여부
9	시설물_등록하다	업체회원, 시설물	1:N	-
10	대여_신청하다	회원, 시설물	M:N	승인여부, 신청일시, 시작일시, 종료일시, 사용자수, 청결상태
11	개설하다	음악가회원, 음악강의	1:N	일시
12	수강_신청하다	일반회원, 음악강의	M:N	신청일시, 강사승인여부
13	발행하다	음악강의, 할인쿠폰	1:N	
14	소유하다	일반회원, 할인쿠폰	M:N	
15	판매하다	회원, 중고물품	1:N	등록일시, 배송비, 진행상태, 거래유형
16	구매하다	회원, 중고물품	1:N	구매일시, 배송지, 구매확정여부
17	찜하다	회원, 중고물품	M:N	-
18	평판점수_매기다	회원, 중고물품	M:N	평가자구분, 점수, 이유
19	후원하다	일반회원, 음악가회원	M:N	금액, 후원일시
20	기획하다	회원, 프로젝트	1:N	-
21	펀딩하다	회원, 프로젝트	M:N	금액, 일시
22	멤버십_등록하다	음악가회원, 멤버십	1:N	
23	가입하다	일반회원, 멤버십	M:N	가입일자

6) 최종 개체 목록

<표 20>에서 선정된 개체의 식별자를 결정하고, 일부 속성 이름을 적절히 변경하였으며, 일부 속성 값을 간단히 표현하는데 필요한 코드 개체까지 식별한 최종 개체 목록은 <표 23>과 같다. 여기서 슈퍼타입 개체인 회원 개체와 서브타입 개체인 일반회원 및 음악가회원 개체를 대상으로 하는 응용 프로그램의 수가 비슷하다고 가정하고, 슈퍼타입 개체와 서브타입 개체를 각각 분리해서 정제하였다.

<표 23> 최종 개체 목록

No	개체	속성 목록	비고	식별자
1	회원	회원_id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호, 평판점수, 활동점수, 등급, 프로필공개여부	일반회원과 음악가회원의 슈퍼타입 개체	회원_id
2	일반회원	회원_id, 포인트	회원의 서브타입 개체	회원_id
3	업체회원	id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 사업자등록번호, 업체명		id
4	음악가회원	회원_id, 학력, 작업실주소	회원의 서브타입 개체	회원_id
5	앨범	번호, 제목, 발매일, 누적조회수, 월별조회수		번호
6	곡	번호, 제목, 장르, 작곡일, 누적조회수, 월별조회수		번호
7	장르	코드, 장르구분	코드 개체	코드
8	접속내역	순서, 접속일시	회원의 약한 개체	{회원_id, 순서}
9	공연	번호, 제목, 장르, 개최일시, 마감일시, 시간, 모집인원수, 최소관람객수, 최대관람객수, 가격, 관람등급		번호
10	시설물	시설물_id, 시설물명, 주소, 등급, 대여제한등급, 시설물구분		시설물_id
11	음악강의	번호, 강의명, 내용, 장소, 개강일시, 종강일시, 강의요일, 시작시각, 종료시각, 수강료, 대면여부, 수강정원		번호
12	할인쿠폰	번호, 할인율, 발행일시, 유효기간		번호
13	중고물품	번호, 물품명, 카테고리, 판매가, 구매가, 물품상태, 조회수		번호
14	중고물품_카테고리	코드, 카테고리	코드 개체	코드
15	프로젝트	번호, 제목, 분야, 세부설명, 목표후원금액, 시작일자, 종료일자		번호
16	멤버십	번호, 등록일자, 가입금액, 제공포인트		번호
17	판매진행상태	코드, 판매상태	코드 개체 (판매하다 관계의 진행 상태 속성값 표현)	코드

7) 최종 관계 목록

<표 22>의 관계 식별 목록을 기초로 식별자를 결정하고, 일부 속성 이름을 적절히 변경한 최종 관계 목록은 <표 24>와 같다.

<표 24> 최종 관계 목록

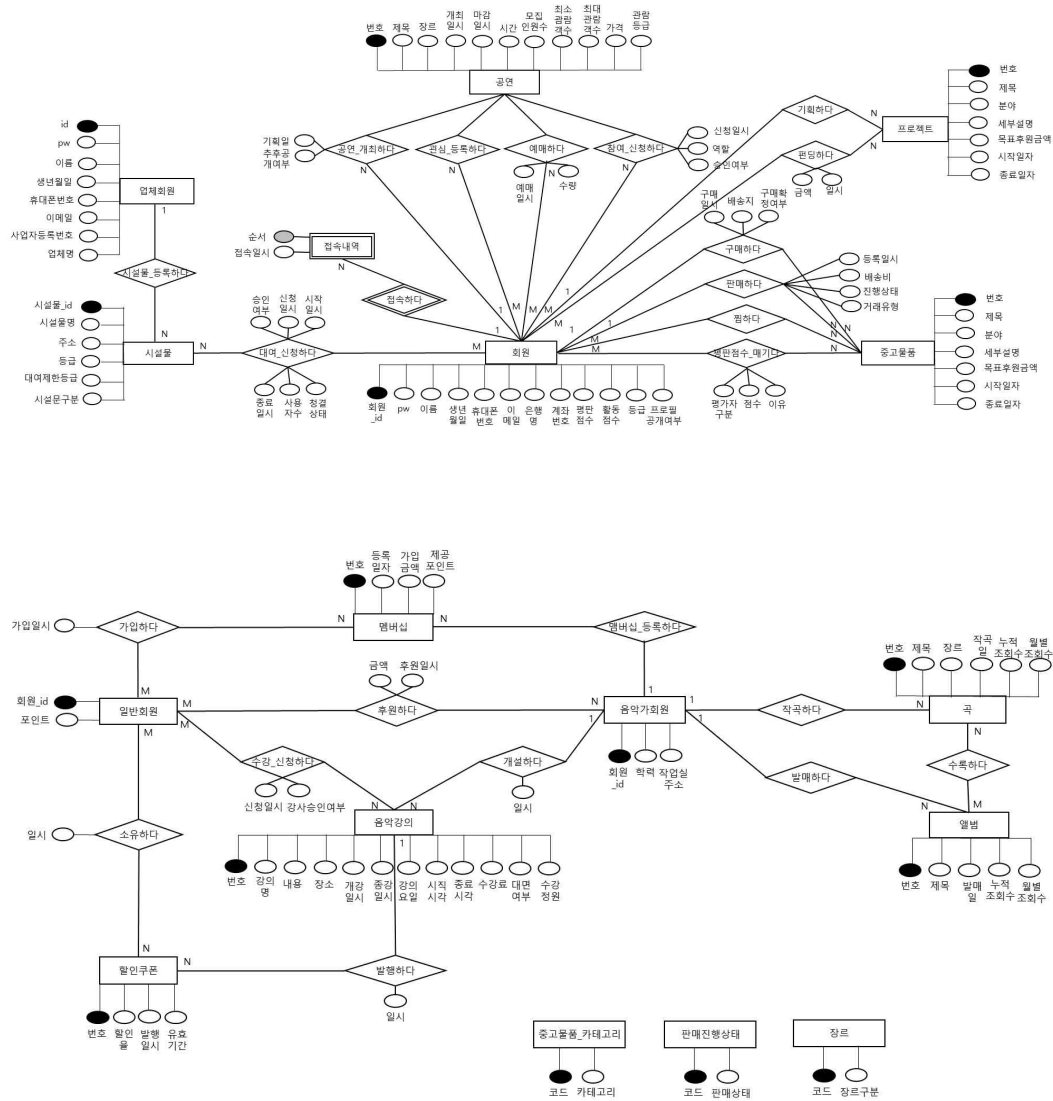
No.	관계	관련된 개체 목록	관계 유형	관계 설명 속성	식별자
1	접속하다 (식별관계)	회원, 접속내역	1:N	-	-
2	발매하다	음악가회원, 앨범	1:N	-	앨범_번호
3	수록하다	앨범, 곡	M:N	-	{앨범_번호, 곡_번호}
4	작곡하다	음악가회원, 곡	1:N	-	곡_번호
5	공연_개최하다	회원, 공연	1:N	기획일, 추후공개여부	공연_번호
6	관심_등록하다	회원, 공연	M:N	-	{회원_id, 공연_번호}
7	예매하다	회원, 공연	M:N	예매일시, 수량	{회원_id, 공연_번호}
8	참여_신청하다	회원, 공연	M:N	신청일시, 역할, 승인여부	{회원_id, 공연_번호}
9	시설물_등록하다	업체회원, 시설물	1:N	-	시설물_id
10	대여_신청하다	회원, 시설물	M:N	승인여부, 신청일시, 시작일시, 종료일시, 사용자수, 청결상태	{회원_id, 시설물_id, 시작일시}
11	개설하다	음악가회원, 음악강의	1:N	일시	음악강의번호
12	수강_신청하다	일반회원, 음악강의	M:N	신청일시, 강사승인여부	{회원_id, 음악강의번호}
13	발행하다	음악강의, 할인쿠폰	1:N	일시	할인쿠폰번호
14	소유하다	일반회원, 할인쿠폰	M:N	일시	{회원_id, 할인쿠폰번호}
15	판매하다	회원, 중고물품	1:N	등록일시, 배송비, 진행상태, 거래유형	중고물품번호
16	구매하다	회원, 중고물품	1:N	구매일시, 배송지, 구매확정여부	중고물품번호
17	찜하다	회원, 중고물품	M:N	-	{회원_id, 중고물품번호}
18	평판점수_매기다	회원, 중고물품	M:N	평가자구분, 점수, 이유	{회원_id, 중고물품번호}
19	후원하다	일반회원, 음악가회원	M:N	금액, 후원일시	{일반회원_id, 음악가회원_id, 후원일시}
20	기획하다	회원, 프로젝트	1:N	-	프로젝트번호
21	편당하다	회원, 프로젝트	M:N	금액, 일시	{회원_id, 프로젝트번호}
22	멤버십_등록하다	음악가회원, 멤버십	1:N		멤버십번호
23	가입하다	일반회원, 멤버십	M:N	가입일시	{회원_id, 멤버십번호, 가입일자}

(2) ER 다이어그램 작성 전 체크리스트

<표 25> DB 구축을 위한 ER 다이어그램 작성 전 체크리스트

구분	No.	내용	확인 (예/아니오)
개체 관련	1	실제 데이터가 1개만 존재하는 개체가 식별되었는가?	아니오
	2	속성이 1개인 개체가 식별되었는가?	아니오
	3	개체 이름이 중복되지 않았는가?	아니오
	4	각 개체의 식별자가 정확히 결정되었는가?	예
	5	각 개체를 설명하는 속성이 모두 식별되었는가?	예
	6	각 개체를 설명하는데 필요한 속성이 추가로 식별되었는가?	예
	7	개체 이름을 부여하는 일정한 규칙이 있는가?	예
관계 관련	1	개체 가운데 관계로 표현하는 것이 더 적합한 것이 있는가?	아니오
	2	관계를 설명하는데 필요한 속성이 모두 식별되었는가?	예
	3	관계 이름이 동사로 표현되었는가?	예
	4	관계 이름이 현재 시제로 표현되었는가?	예
	5	관계 이름이 개체들 간의 관련성을 명확히 표현하고 있는가?	예
	6	관계 이름이 중복되지 않았는가?	아니오
	7	관계 유형을 정확히 결정하였는가?	예
	8	관계 이름을 부여하는 일정한 규칙이 있는가?	예
속성 관련	1	속성 이름에 개체나 관계 이름이 접두어로 사용되지 않았는가?	아니오
	2	속성 이름으로 의미가 불분명한 약어를 사용하지 않았는가?	아니오
	3	속성의 성격을 명확히 이해하기 어려운 속성 이름이 부여되지는 않았는가?	아니오
	4	속성 이름을 부여하는 일정한 규칙이 있는가?	예

(3) ER 다이어그램



[그림 2] 청년 음악가 후원 웹사이트의 DB 구축을 위한 ER 다이어그램

(4) 트랜잭션 모델링

앞에서 작성한 트랜잭션 요구 분석 명세서를 기초로 업무별로 트랜잭션의 유형을 구분해서 작성하면 <표 26> ~ <표 33>과 같다.

<표 26> 회원 관리 업무 관련 유형별 트랜잭션 명세서

No.	유형	트랜잭션
1	삽입	새로 가입한 회원 정보를 삽입한다.
2		새로 가입한 업체회원 정보를 삽입한다.
3		음악가회원은 자신의 앨범 정보를 삽입할 수 있다.
4		음악가회원은 자신의 곡 정보를 삽입할 수 있다.
5	삭제	음악가회원은 자신의 앨범 정보를 삭제할 수 있다.
6		음악가회원은 자신의 곡 정보를 삭제할 수 있다.
7	갱신	회원의 활동점수에 따라 등급을 갱신한다.
8		회원은 계좌정보를 갱신할 수 있다.
9		회원은 프로필 정보 공개 여부를 갱신할 수 있다.
10		음악가회원은 자신의 앨범과 곡 정보를 갱신할 수 있다.
11		일반회원은 음악가회원으로 변경할 수 있다.
12	검색	매월 1일 앨범과 곡의 월별조회수를 0으로 변경한다.
13		회원은 프로필을 공개한 다른 회원의 프로필을 검색할 수 있다.
14		회원은 다른 회원의 평판점수를 검색할 수 있다.
15		회원은 음악가회원이 발매한 앨범을 검색할 수 있다.
16		회원은 음악가회원이 등록한 곡을 검색할 수 있다.
17		회원은 누적조회수가 높은 곡을 수록하는 앨범을 검색할 수 있다.
18		회원은 희망하는 장르의 곡을 누적조회수가 높은 순으로 검색할 수 있다.
19		회원은 앨범을 누적조회수가 높은 순으로 검색할 수 있다.
20		회원은 월별조회수가 높은 앨범에 수록된 곡을 검색할 수 있다.
21		회원은 월별조회수가 높은 순으로 곡을 검색할 수 있다.
22		회원은 등급이 높은 음악가회원의 곡을 누적조회수가 높은 순으로 검색할 수 있다.
23		회원은 어떤 곡을 작곡한 음악가회원을 검색할 수 있다.
24		회원은 발매한 앨범 수의 순으로 음악가회원을 검색할 수 있다.
25		관리자는 평판점수가 0점인 회원을 검색할 수 있다.
26		관리자는 마지막으로 접속한 지 30일이 경과한 회원을 검색할 수 있다.

<표 27> 공연 관리 업무 관련 유형별 트랜잭션 명세서

No.	유형	트랜잭션
1	삽입	회원은 자신이 개최하는 공연 정보를 삽입할 수 있다.
2		회원은 공연을 관심 목록에 삽입할 수 있다.
3	삭제	회원은 자신이 개최하는 공연 정보를 삭제할 수 있다.
4		회원은 공연을 관심 목록에서 삭제할 수 있다.
5	갱신	회원은 자신이 개최하는 공연 정보를 갱신할 수 있다.
6		공연 기획자는 회원의 공연 참여 신청을 승인할 수 있다.
7		공연 기획자는 회원의 공연 참여 신청을 거절할 수 있다.
8		공연기획 시 회원의 활동점수를 자동으로 10점 증가한다.
9		공연 참여 시 회원의 활동점수를 자동으로 10점 증가한다.
10	검색	공연이 끝나면 기획자는 공개/비공개 여부를 변경할 수 있다.
11		회원은 공연 정보를 검색할 수 있다.
12		회원은 잔여 좌석 수가 적은 순으로 공연을 검색할 수 있다.
13		관리자는 일자별로 공연을 검색할 수 있다.
14		관리자는 일시별로 공연 예매 현황을 검색할 수 있다.
15		회원은 장르별로 가장 최근에 기획된 순으로 공연을 검색할 수 있다.
16		회원은 희망하는 기획자의 공연을 가격이 낮은 순으로 검색할 수 있다.
17		회원은 지역별로 가장 최근에 기획된 순으로 공연을 검색할 수 있다.

<표 28> 시설물 대여 업무 관련 유형별 트랜잭션 명세서

No.	유형	트랜잭션
1	삽입	회원은 관리자에게 시설물 대여를 신청할 수 있다.
2		회원이 시설물 대여 신청 시 신청일시, 시작일시, 종료일시, 사용자수 정보가 삽입된다.
3		업체회원은 시설물 정보를 삽입할 수 있다.
4	삭제	회원은 시설물 대여 신청이 승인되기 전까지 신청을 취소할 수 있다.
5		업체회원은 더 이상 존재하지 않는 시설물 정보를 삭제할 수 있다.
6	갱신	회원은 시설물 대여 신청이 승인되기 전까지 시작일시, 종료일시, 사용자수 정보를 갱신할 수 있다.
7		업체회원은 시설물 정보를 갱신할 수 있다.
8		관리자는 회원의 시설물 대여 신청을 승인할 수 있다.
9		관리자는 회원의 시설물 대여 신청을 거절할 수 있다.
10		관리자는 회원이 대여한 시설물 상태에 따라 청결상태의 등급을 변경할 수 있다.
11	검색	청결상태에 따라 해당 시설물을 대여한 회원의 평판점수를 자동으로 갱신한다.
12		회원은 시설물을 시설물의 등급별로 검색할 수 있다.
13		회원은 시설물을 지역별로 검색할 수 있다.
14		회원은 본인의 등급을 기준으로 대여할 수 있는 시설물을 지역별로 검색할 수 있다.
15		회원은 월별 가장 많이 대여된 시설물을 검색할 수 있다.
16		회원은 현재 대여 가능한 시설물을 최신순으로 검색할 수 있다.
17		회원은 현재 대여 가능한 시설물 중 가장 많이 대여된 시설물 순으로 검색할 수 있다.
18		관리자는 시설물 대여를 신청한 회원의 평판점수를 검색할 수 있다.
19		관리자는 시설물 대여를 신청한 회원의 등급을 검색할 수 있다.
20		관리자는 시설물 대여를 신청한 회원 중 대여 기간이 한 달을 초과한 회원을 검색할 수 있다.

21		관리자는 월별 가장 많이 대여된 시설물을 검색할 수 있다.
22		관리자는 시설물의 등급을 검색할 수 있다.
23		관리자는 현재 시설물을 대여 중인 모든 회원을 검색할 수 있다.

<표 29> 강의 관리 업무 관련 유형별 트랜잭션 명세서

No.	유형	트랜잭션
1	삽입	음악가회원은 자신이 개설한 음악강의 정보를 삽입할 수 있다.
2		일반회원은 음악강의를 수강 신청할 수 있다.
3		음악가회원은 자신이 개설한 음악강의에 대한 할인쿠폰을 삽입할 수 있다.
4	삭제	음악가회원은 자신이 개설한 음악강의 정보를 삭제할 수 있다.
5		일반회원은 음악강의 수강신청 정보를 음악가회원이 승인하기 전까지만 삭제할 수 있다.
6		음악가회원은 자신이 개설한 음악강의에 대한 할인쿠폰을 삭제할 수 있다.
7		일반회원이 사용한 할인쿠폰은 자동 삭제된다.
8		발행일로부터 유효기간이 지난 할인쿠폰은 자동으로 삭제된다.
9	갱신	음악가회원은 자신이 개설한 음악강의 정보를 변경할 수 있다.
10		음악가회원은 일반회원이 수강 신청한 음악강의의 수강을 승인할 수 있다.
11		음악가회원은 자신이 개설한 음악강의에 대한 할인쿠폰을 변경할 수 있다.
12		음악가회원이 강의를 개설하면 활동점수를 자동으로 10점 증가시킨다.
13		회원이 음악강의를 수강하면 활동점수를 자동으로 10점 증가시킨다.
14	검색	음악가회원은 자신이 개설한 음악강의를 수강하는 일반회원을 검색할 수 있다.
15		일반회원은 특정 음악강의에 적용할 수 있는 할인쿠폰을 검색할 수 있다.
16		일반회원은 자신이 소유하고 있는 할인쿠폰을 검색할 수 있다.
17		일반회원은 수강료가 낮은 순으로 음악강의를 검색할 수 있다.
18		일반회원은 수강정원이 많이 남은 순으로 음악강의를 검색할 수 있다.
19		일반회원은 할인율이 높은 순서대로 음악강의를 검색할 수 있다.
		일반회원은 특정 요일에 개설되는 음악강의를 검색할 수 있다.

<표 30> 중고거래 업무 관련 유형별 트랜잭션 명세서

No.	유형	트랜잭션
1	삽입	회원은 중고물품 정보를 삽입할 수 있다.
2	삭제	회원은 자신이 등록한 중고물품 정보를 삭제할 수 있다.
3	갱신	회원은 자신이 등록한 중고물품 정보를 변경할 수 있다.
4		회원이 중고물품을 조회할 때마다 자동으로 해당 물품의 조회수가 1씩 증가된다.
5		중고물품 구매회원이 구매를 결정하면 자동으로 판매진행상태가 예약중으로 갱신된다.
6		거래를 완료한 후에 회원이 상대 회원의 평판점수를 평가하면, 자동으로 상대 회원의 평판점수가 갱신된다.
7		거래 완료하면 거래한 회원의 활동점수는 자동으로 5점 추가된다.
8	검색	회원은 등록된 중고물품을 검색할 수 있다.
9		회원은 자신이 찜한 중고물품 목록을 검색할 수 있다.
10		회원은 신뢰할 수 있는 중고거래를 위해서 다른 회원의 거래 내역을 검색할 수 있다.
11		회원은 조회수가 높은 순으로 중고물품을 검색할 수 있다.
12		회원은 최신 등록일 순서로 중고물품을 검색할 수 있다.
13		회원은 가격이 낮은 순으로 중고물품을 검색할 수 있다.
14		회원은 찜한 회원이 많은 중고물품을 검색할 수 있다.
15		관리자는 일/월/연도 별로 가장 많이 판매된 물품 카테고리를 검색할 수 있다.
16		관리자는 평판점수가 높은 회원이 등록한 물품을 검색할 수 있다.
17		회원은 평판점수가 높은 회원이 등록한 물품을 검색할 수 있다.

<표 31> 지정 후원 업무 관련 유형별 트랜잭션 명세서

No	유형	트랜잭션
1	변경	음악가회원이 후원을 받으면 후원내역이 자동으로 변경된다.
2		음악가회원에게 후원한 일반회원의 활동점수는 10점 증가된다.
3	검색	음악가회원은 자신이 후원받은 총 후원누적금액을 검색할 수 있다.
4		음악가회원은 자신에게 후원한 일반회원을 검색할 수 있다.
5		음악가회원은 자신에게 후원한 일반회원 중 후원 횟수가 많은 회원 순으로 검색할 수 있다.
6		음악가회원은 자신에게 후원한 일반회원 중 후원 금액이 가장 큰 회원 순으로 검색할 수 있다.
7		일반회원은 자신이 음악가회원들에게 후원한 총 후원누적금액을 검색할 수 있다.
8		일반회원은 자신이 특정 음악가회원에게 후원한 후원누적금액을 검색할 수 있다.
9		일반회원은 일별로 자신이 후원한 총 금액을 검색할 수 있다.
10		일반회원은 자신의 후원내역을 최신순으로 검색할 수 있다.
11		관리자는 음악가회원에게 후원한 회원목록을 검색할 수 있다.
12		관리자는 월별로 가장 많은 후원을 받은 음악가회원을 검색할 수 있다.
13		관리자는 후원누적금액을 음악가회원별로 검색할 수 있다.

<표 32> 프로젝트 펀딩 업무 관련 유형별 트랜잭션 명세서

No	유형	트랜잭션
1	삽입	회원은 자신이 기획하는 프로젝트 정보를 삽입한다.
2	삭제	프로젝트 종료 이후 10년이 지난 프로젝트는 자동으로 삭제한다.
3	갱신	회원은 자신이 기획한 프로젝트의 정보를 변경할 수 있다.
4		회원이 기획한 프로젝트가 완료되면 회원의 활동점수는 10점 증가한다.
5		프로젝트에 펀딩한 회원의 활동점수를 10점 증가시킨다.
6	검색	회원은 진행 중인 프로젝트를 검색할 수 있다.
7		회원은 자신이 기획한 프로젝트에 펀딩한 회원들을 검색할 수 있다.
8		회원은 현재 진행 중인 프로젝트 가운데 누적펀딩금액이 가장 프로젝트를 검색할 수 있다.
9		회원은 자신이 기획한 프로젝트의 정보를 검색할 수 있다.
10		회원은 자신이 펀딩한 프로젝트의 정보를 검색할 수 있다.
11		관리자는 월별로 누적펀딩금액이 가장 큰 프로젝트 정보를 검색할 수 있다.
12		회원은 마감일 순으로 프로젝트를 검색할 수 있다.
13		회원은 특정 분야에 속하는 프로젝트를 검색할 수 있다.
14		회원은 특정 분야에 속하는 프로젝트를 높은 금액 순으로 검색할 수 있다.
15		회원은 목표후원금액에 가까운 순서대로 프로젝트를 검색할 수 있다.

<표 33> 멤버십 가입 업무 관련 유형별 트랜잭션 명세서

No	유형	트랜잭션
1	삽입	음악가회원은 자신의 멤버십 정보를 삽입할 수 있다.
2	삭제	음악가회원은 자신의 멤버십 정보를 삭제할 수 있다.
3	갱신	음악가회원은 자신의 멤버십 정보를 변경할 수 있다.
4		멤버십을 등록한 음악가회원의 활동점수를 자동으로 갱신(증가)한다.
5		멤버십에 가입한 일반회원의 활동점수를 자동으로 갱신(증가)한다.
6	검색	멤버십을 삭제한 음악가회원의 활동점수를 자동으로 갱신(감소)한다.
7		음악가회원은 자신의 멤버십에 가입한 일반회원들을 검색할 수 있다.
8		관리자는 가장 많은 멤버십 회원을 보유한 음악가회원을 검색할 수 있다.
9		일반회원은 보유회원수가 많은 순서대로 음악가회원을 검색할 수 있다.
10		일반회원은 자신이 가입한 멤버십 목록을 검색할 수 있다.
11		일반회원이 제공포인트를 많이 주는 멤버십을 최신순으로 검색할 수 있다.
12		일반회원이 가입금액이 낮은 순으로 멤버십을 검색할 수 있다.

(5) 운영상의 제약사항

요구 분석 내용 가운데 ER 다이어그램이나 트랜잭션 명세서에 표현하기 어려운 운영상의 제약사항은 <표 34>와 같다.

<표 34> 운영상의 제약사항 명세서

No	제약사항	관련 업무
1	활동점수는 강의 수강, 강의 개설, 공연 기획, 공연 참여, 프로젝트 기획, 프로젝트 펀딩, 지정 후원, 멤버십 가입을 통해 10점씩 획득한다.	회원 관리
2	회원이 한 달 이상 접속하지 않으면 활동점수를 10점씩 감소한다.	회원 관리
3	평판점수는 한 번의 중고거래를 통해 최대 5점, 시설물 대여를 통해 최대 20점까지 감소될 수 있다.	회원 관리
4	평판점수가 10점씩 차감될 때마다 활동점수 5점씩 감소한다.	회원 관리
5	활동점수와 평판점수의 최하 점수는 0점이다.	회원 관리
6	활동점수가 10점씩 증가할 때마다 평판점수는 5점씩 증가한다.	회원 관리
7	평판점수가 0점이 되면 영구적으로 대여 및 거래가 불가능하다.	회원 관리
8	공연은 VIP 등급 이상의 회원만 기획할 수 있다.	공연 관리
9	회원이 시설물 대여 신청하면, 관리자는 회원의 등급과 평판점수를 기초로 승인 여부를 결정한다.	시설물 대여
10	시설물 대여 승인 과정에서 회원의 평판점수가 0점이면 승인 요청이 거절된다.	시설물 대여
11	회원의 대여 신청 내용 변경 및 신청 취소는 관리자가 승인여부를 결정하기 전까지만 가능하다.	시설물 대여
12	하나의 공연장 대여 중에는 다른 공연장을 대여할 수 없다.	시설물 대여
13	하나의 연습실 대여 중에는 다른 연습실을 대여할 수 없다.	시설물 대여
14	대여 기간은 최대 한 달을 넘을 수 없다.	시설물 대여
15	회원의 대여 신청 내용 변경 및 신청 취소는 관리자가 승인여부를 결정하기 전까지만 가능하다.	시설물 대여
16	일반회원은 음악강의 수강신청 정보를 음악가회원이 승인하기 전까지만 삭제할 수 있다.	강의 관리
17	하나의 중고물품 거래가 완료되면 판매 회원과 구매 회원이 각각 상대 회원의 평판 점수를 평가할 수 있다.	중고거래
18	음악가회원을 후원한 일반회원의 활동점수는 10점 증가된다.	지정 후원
19	일반회원은 프로젝트 분야 중 공연만 프로젝트를 기획할 수 있다.	프로젝트 펀딩
20	현재 진행 중인 프로젝트는 삭제할 수 없다.	프로젝트 펀딩
21	회원이 프로젝트를 펀딩하면 활동점수가 10점 증가한다.	프로젝트 펀딩
22	회원이 기획한 프로젝트가 완료되면 회원의 활동점수는 10점 증가한다.	프로젝트 펀딩
23	멤버십에 가입한 일반회원은 해당 멤버십을 등록한 음악가회원의 공연을 선예매 할 수 있다.	멤버십 가입
24	일반회원이 멤버십에 가입하면 활동점수가 매달 10점씩 증가된다.	멤버십 가입
25	음악가회원이 멤버십을 등록하면 활동점수가 매달 10점씩 증가된다.	멤버십 가입
26	음악가회원이 멤버십을 삭제하면 활동점수는 20점 감소한다.	멤버십 가입

4. 논리적 설계

목차로

(1) 개체 릴레이션 스키마

<표 35> 개체 릴레이션 스키마

No.	개체	릴레이션 스키마	비고
1	회원	회원(id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호, 활동점수, 평판점수, 등급, 프로필공개여부)	
2	일반회원	일반회원(회원_id(FK), 포인트)	회원(id) 참조
3	음악가회원	음악가회원(회원_id(FK), 학력, 작업실주소)	회원(id) 참조
4	업체회원	업체회원(업체회원_id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 사업자등록번호, 업체명)	
5	접속내역	접속내역(회원_id(FK), 순서, 접속일시)	회원의 약한개체
6	앨범	앨범(번호, 제목, 발매일, 누적조회수, 월별조회수)	
7	곡	곡(번호, 제목, 장르, 작곡일, 누적조회수, 월별조회수)	
8	공연	공연(번호, 제목, 장르, 개최일시, 마감일시, 시간, 모집인원수, 최대관람객수, 최소관람객수, 가격, 관람등급)	
9	장르	장르(코드, 장르구분)	코드개체
10	시설물	시설물(시설물_id, 시설물명, 주소, 등급, 대여제한등급, 시설물구분)	
11	음악강의	음악강의(번호, 강이명, 내용, 장소, 개강일시, 종강일시, 강의요일, 시작시각, 종료시각, 수강료, 대면여부, 수강정원)	
12	할인쿠폰	할인쿠폰(번호, 할인율, 유효기간)	
13	중고물품	중고물품(번호, 물품명, 카테고리, 판매가, 구매가, 물품상태, 조회수)	
14	중고물품_카테고리	물품카테고리(코드, 카테고리)	코드개체
15	판매진행상태	판매진행상태(코드, 판매상태)	코드개체
16	멤버십	멤버십(번호, 등록일자, 가입금액, 제공포인트)	
17	프로젝트	프로젝트(번호, 제목, 분야, 세부설명, 목표후원금액, 시작일자, 종료일자)	

(2) 관계 릴레이션 스키마

<표 36> 관계 릴레이션 스키마

No.	관계	릴레이션 스키마	비고
1	작곡하다	작곡하다(곡_번호(FK), 회원_id(FK))	
2	발매하다	발매하다(앨범_번호(FK), 회원_id(FK))	
3	수록하다	수록하다(앨범_번호 (FK), 곡_번호(FK))	
4	공연_개최하다	공연_개최하다(공연_번호(FK), 회원_id(FK), 기획일, 추후공개여부)	
5	예매하다	예매하다(공연_번호(FK), 회원_id(FK), 예매일시, 수량)	
6	참여_신청하다	참여_신청하다(공연_번호(FK), 회원_id(FK), 신청일시, 역할, 승인여부)	
7	관심_등록하다	관심_등록하다(회원_id(FK), 공연_번호(FK))	
8	시설물_등록하다	시설물_등록하다(시설물_id(FK), 업체회원_id(FK))	
9	대여_신청하다	대여_신청하다(회원_id(FK), 시설물_id(FK), 시작일시, 승인여부, 신청일시, 종료일시, 사용자수, 청결상태)	한 회원이 같은 시설물을 여러 번 대여 신청할 수 있으므로 기본키에 '시작일시' 추가
10	개설하다	개설하다(음악강의번호(FK), 회원_id(FK), 일시)	
11	수강_신청하다	수강_신청하다(회원_id(FK), 음악강의번호(FK), 신청일시, 강사승인여부)	한 회원이 같은 강의를 여러 번 수강신청할 수 있으므로 기본키에 '신청일시' 추가
12	발행하다	발행하다(할인쿠폰번호(FK), 음악강의번호(FK), 일시)	
13	소유하다	소유하다(회원_id(FK), 할인쿠폰번호(FK), 일시)	
14	판매하다	판매하다(중고물품번호(FK), 판매회원_id(FK), 등록일시, 배송비, 진행상태, 거래유형)	
15	구매하다	구매하다(중고물품번호(FK), 구매회원_id(FK), 구매일시, 배송지, 구매확정여부)	
16	찜하다	찜하다(회원_id(FK), 중고물품번호(FK))	
17	평판점수_매기다	평판점수_매기다(중고물품번호(FK), 회원_id(FK), 평가자구분, 점수)	
18	후원하다	후원하다(후원번호, 일반회원_id(FK), 음악가회원_id(FK), 후원일시, 금액)	기본키인 {일반회원_id, 음악가회원_id, 후원일시}를 대체하기 위해 '후원번호' 속성 추가
19	기획하다	기획하다(프로젝트번호(FK), 회원_id(FK))	
20	펀딩하다	펀딩하다(프로젝트번호(FK), 회원_id(FK), 금액, 일시)	
21	멤버십_등록하다	멤버십_등록하다(멤버십번호(FK), 회원_id(FK))	
22	가입하다	가입하다(가입번호, 회원_id(FK), 멤버십번호(FK), 가입일자)	기본키인 {회원_id, 멤버십번호, 가입일시}를 대체하기 위해 '가입번호' 속성 추가

(3) 단순화된 릴레이션 스키마

<표 37> 단순화된 릴레이션 스키마

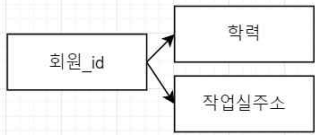
No.	개체 및 관계	릴레이션 스키마	비고
1	회원	회원(id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호, 활동점수, 평판점수, 등급, 프로필공개여부)	
2	일반회원	일반회원(회원_id(FK), 포인트)	
3	음악가회원	음악가회원(회원_id(FK), 학력, 작업실주소)	
4	업체회원	업체회원(업체회원_id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 사업자등록번호, 업체명)	
5	접속내역	접속내역(회원_id(FK), 순서, 접속일시)	
6	앨범	앨범(번호, 발매회원_id(FK), 제목, 발매일, 누적조회수, 월별조회수)	'발매하다' 관계 단순화 -> 외래키로 '발매회원_id' 추가
7	곡	곡(번호, 작곡자_id(FK), 작곡일, 제목, 장르, 누적조회수, 월별조회수)	'작곡하다' 관계 단순화 -> 외래키로 '작곡자_id' 추가
8	공연	공연(번호, 제목, 장르, 개최일시, 마감일시, 시간, 모집인원수, 최대관람객수, 최소관람객수, 가격, 관람등급, 개최회원_id(FK), 기획일, 추후공개여부)	'공연_개최하다' 관계 단순화 -> '개최회원_id(FK)', '기획일', '추후공개여부' 추가
9	장르	장르(코드, 장르구분)	
10	시설물	시설물(시설물_id, 시설물명, 주소, 등급, 대여제한등급, 시설물구분, 등록업체회원_id(FK))	'시설물_등록하다' 관계 단순화 -> '등록업체회원_id' 추가
11	음악강의	음악강의(번호, 강의명, 내용, 장소, 개강일시, 종강일시, 강의요일, 시작시각, 종료시각, 수강료, 대면여부, 수강정원, 음악가회원_id(FK), 개설일시)	'개설하다' 관계 단순화 -> '음악가회원_id', '개설일시' 속성 추가
12	할인쿠폰	할인쿠폰(번호, 할인율, 유효기간, 음악강의번호(FK), 발행일시)	'발행하다' 관계 단순화 -> '음악강의번호', '발행일시' 속성 추가
13	중고물품	중고물품(번호, 물품명, 카테고리, 판매가, 구매가, 물품상태, 조회수)	
14	중고물품_카테고리	물품카테고리(코드, 카테고리)	
15	멤버십	멤버십(번호, 등록일자, 가입금액, 제공포인트, 멤버십_등록자_id(FK))	'멤버십_등록하다' 관계 단순화 -> '멤버십_등록자_id', 속성 추가
16	프로젝트	프로젝트(번호, 제목, 분야, 세부설명, 목표후원금액, 프로젝트기획자_id(FK), 시작일시, 종료일시)	'기획하다' 관계 단순화 -> '프로젝트기획자_id' 속성 추가
17	수록하다	수록하다(곡_번호(FK), 앨범_번호(FK))	
18	예매하다	예매하다(공연_번호(FK), 회원_id(FK), 예매일시, 수량)	
19	참여	참여_신청하다(공연_번호(FK), 회원_id(FK), 신청일시,	

	<u>_신청하다</u>	역할, 승인여부)	
20	<u>관심_등록하다</u>	관심_등록하다(<u>회원_id(FK)</u> , <u>공연_번호(FK)</u>)	
21	<u>대여_신청하다</u>	대여_신청하다(<u>회원_id(FK)</u> , <u>시설물_id(FK)</u> , <u>시작일시</u> , 승인여부, 신청일시, 종료일시, 사용자수, 청결상태)	
22	<u>수강_신청하다</u>	수강_신청하다(<u>회원_id(FK)</u> , <u>음악강의번호(FK)</u> , <u>신청일시</u> , <u>강사승인여부</u>)	
23	소유하다	소유하다(<u>회원_id(FK)</u> , <u>할인쿠폰번호(FK)</u> , 일시)	
24	판매하다	판매하다(<u>중고물품번호(FK)</u> , <u>판매회원_id(FK)</u> , <u>등록일시</u> , 배송비, 진행상태, 거래유형)	관계 설명 속성이 많아서 단순화시키지 않았음
25	진행상태	진행상태(<u>코드</u> , 판매상태)	
26	구매하다	구매하다(<u>중고물품번호(FK)</u> , <u>구매회원_id(FK)</u> , <u>구매일시</u> , 배송지, 구매확정여부)	관계 설명 속성이 많아서 단순화시키지 않았음
27	찜하다	찜하다(<u>회원_id(FK)</u> , <u>중고물품번호(FK)</u>)	
28	<u>평판점수_매기다</u>	평판점수_매기다(<u>중고물품번호(FK)</u> , <u>구매회원_id(FK)</u> , 평가자구분, 점수, 이유)	
29	후원하다	후원하다(<u>후원번호</u> , <u>일반회원_id(FK)</u> , <u>음악가회원_id(FK)</u> , 후원일시, 금액)	
30	펀딩하다	펀딩하다(<u>프로젝트번호(FK)</u> , <u>펀딩회원_id(FK)</u> , 금액, 펀딩일시)	
31	가입하다	가입하다(<u>가입번호</u> , <u>가입자_id(FK)</u> , <u>멤버십번호(FK)</u> , 가입일자)	

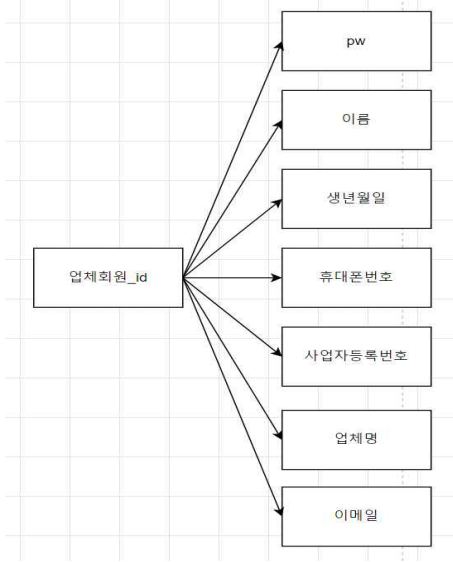
(4) 릴레이션 정규화

<표 37>의 릴레이션 가운데 몇 가지 릴레이션의 정규화 과정을 소개하면 <표 38> ~ <표 40>과 같다.

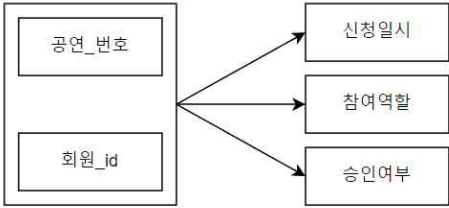
<표 38> 음악가회원 릴레이션의 정규화 내용

릴레이션 스키마	음악가회원(회원_id(FK), 학력, 작업실주소)	
기본 키	회원_id	
대체 키	없음	
함수적 종속	회원_id -> (학력, 작업실주소)	
FDD	 <pre> graph LR A[회원_id] --> B[학력] A --> C[작업실주소] </pre>	
정규형	1NF	모든 속성의 도메인이 원자 값이므로 1NF를 만족함
	2NF	모든 속성이 기본 키에 완전 함수적 종속이므로 2NF를 만족함
	3NF	이행적 함수적 종속이 존재하지 않으므로 3NF를 만족함

<표 39> 업체회원 릴레이션의 정규화 내용

릴레이션 스키마	업체회원(업체회원_id, pw, 이름, 생년월일, 휴대폰번호, 이메일, 사업자등록번호, 업체명)	
기본 키	업체회원_id	
대체 키	휴대폰번호, 이메일, 사업자등록번호	
함수적 종속	업체회원_id -> (pw, 이름, 생년월일, 휴대폰번호, 이메일, 사업자등록번호, 업체명)	
FDD	 <pre> graph LR A[업체회원_id] --> B[pw] A --> C[이름] A --> D[생년월일] A --> E[휴대폰번호] A --> F[사업자등록번호] A --> G[업체명] A --> H[이메일] </pre>	
정규형	1NF	모든 속성의 도메인이 원자 값이므로 1NF를 만족함
	2NF	모든 속성이 기본 키에 완전 함수적 종속이므로 2NF를 만족함
	3NF	이행적 함수적 종속이 존재하지 않으므로 3NF를 만족함

<표 40> 참여_신청하다 릴레이션의 정규화 내용

릴레이션 스키마	참여_신청하다(공연_번호(FK), 회원_id(FK), 신청일시, 역할, 승인여부)	
기본 키	{공연_번호, 회원_id}	
대체 키	없음	
함수적 종속	{공연_번호, 회원_id} → (신청일시, 역할, 승인여부)	
FDD		
정규형	1NF	모든 속성의 도메인이 원자 값이므로 1NF를 만족함
	2NF	모든 속성이 기본 키에 완전 함수적 종속이므로 2NF를 만족함
	3NF	이행적 함수적 종속이 존재하지 않으므로 3NF를 만족함

(5) 정규화된 최종 릴레이션 스키마

정규화 과정을 통해서 분해된 릴레이션이 없으므로 최종 릴레이션 스키마는 <표 37>과 동일하다.

(6) 무결성 제약조건 정의

<표 41> 무결성 제약조건

No	제약조건	제약 유형	관련 릴레이션
1	pw, 이름, 생년월일, 휴대폰번호, 이메일, 은행명, 계좌번호는 널 값을 가질 수 없다.	NOT NULL	회원
2	휴대폰번호, 이메일, 계좌번호는 고유한 값을 가져야 한다.	UNIQUE	회원
3	등급의 초기값은 'BEGIN' 이다.	DEFAULT	회원
4	활동점수와 평판점수의 초기값은 0이다.	DEFAULT	회원
5	활동점수와 평판점수는 0 이상의 값을 가져야 한다.	CHECK	회원
6	등급은 'VIP', 'VIP', 'GOLD', 'SILVER', 'BEGIN' 가운데 하나의 값을 가져야 한다.	CHECK	회원
7	프로필_공개여부는 'Y' 또는 'N' 값을 가져야 한다.	CHECK	회원

8	프로필_공개여부의 초기값은 'N' 이다.	DEFAULT	회원
9	포인트의 초기값은 0 이다.	DEFAULT	일반회원
10	포인트는 0 이상의 값을 가져야 한다.	CHECK	일반회원
11	학력은 '중졸', '고졸', '대졸' 가운데 하나의 값을 가져야 한다.	CHECK	음악가회원
12	pw, 이름, 휴대폰번호, 사업자등록번호, 업체명, 이메일은 널 값을 가질 수 없다.	NOT NULL	업체회원
13	휴대폰번호, 사업자등록번호, 이메일은 유일한 값을 가져야 한다.	UNIQUE	업체회원
14	순서의 초기값은 1이다.	DEFAULT	접속내역
15	접속일시의 초기값은 현재 일시(SYSDATE)이다.	DEFAULT	접속내역
16	제목은 NULL 값을 가질 수 없다.	NOT NULL	앨범
17	발매회원_id는 NULL 값을 가질 수 없다.	NOT NULL	앨범
18	누적조회수과 월별조회수 초기값은 0이다.	DEFAULT	앨범
19	제목과 장르는 NULL 값을 가질 수 없다.	NOT NULL	곡
20	작곡자_id NULL 값을 가질 수 없다.	NOT NULL	곡
21	누적조회수과 월별조회수 초기값은 0이다.	DEFAULT	곡
22	제목, 장르, 개최일시, 마감일시, 시간, 모집인원수, 최대관람객수, 최소관람객수, 가격, 관람등급은 널 값을 가질 수 없다.	NOT NULL	공연
23	최대관람객수는 최소관람객수보다 큰 값을 가져야 한다.	CHECK	공연
24	마감일시는 개최일시보다 큰 값을 가져야 한다.	CHECK	공연
25	추후공개여부는 'Y' 또는 'N' 값을 가져야 한다.	CHECK	공연
26	시설물명, 주소, 등급, 대여제한등급, 시설물구분, 등록업체회원_id는 널 값을 가질 수 없다.	NOT NULL	시설물
27	주소는 유일한 값을 가져야 한다.	UNIQUE	시설물
28	등급은 'A' 또는 'B' 또는 'C' 값을 가져야 한다.	CHECK	시설물
29	대여제한등급은 'VIP' 또는 'SILVER' 값을 가져야 한다.	CHECK	시설물
30	시설물구분은 '공연장' 또는 '연습실' 값을 가져야 한다.	CHECK	시설물
31	대면여부는 '대면' 또는 '비대면' 값을 가져야 한다.	CHECK	음악강의
32	대면여부의 초기값은 '비대면'이다.	DEFAULT	음악강의
33	강의명과 내용, 장소, 개강일시, 종강일시, 강의요일, 시작시각, 종료시각, 음악가회원_id는 널 값을 가질 수 없다.	NOT NULL	음악강의
34	수강료의 초기값은 0(무료)이다.	DEFAULT	음악강의
35	강의요일은 '월', '화', '수', '목', '금', '토' 가운데 하나의 값을 가져야 한다.	CHECK	음악강의
36	할인율은 NULL 값을 가질 수 없다.	NOT NULL	할인쿠폰
37	유효기간의 초기값은 30이다.	DEFAULT	할인쿠폰
38	물품명과 판매가, 구매가는 널 값을 가질 수 없다.	NOT NULL	중고물품
39	조회수 초기값은 0이다.	DEFAULT	중고물품
40	제공포인트는 0 이상의 값을 가져야 한다.	CHECK	멤버십
41	등록일자의 초기값은 오늘 날짜(SYSDATE)로 설정한다.	DEFAULT	멤버십
42	가입금액은 1,000 이상의 값을 가져야 한다.	CHECK	멤버십
43	등록일자와 가입금액은 널 값을 가질 수 없다.	NOT NULL	멤버십
44	제목, 분야, 세부설명, 목표후원금액은 널 값을 가질 수 없다.	NOT NULL	프로젝트

45	종료일자는 시작일자보다 큰 값을 가져야 한다.	CHECK	프로젝트
46	분야는 '곡' '앨범', '공연' 가운데 하나의 값을 가져야 한다.	CHECK	프로젝트
47	예매일시는 현재 일시(SYSDATE) 값을 갖는다.	DEFAULT	예매하다
48	수량의 기초값은 1이다.	DEFAULT	예매하다
49	신청일시는 현재 일시(SYSDATE) 값을 갖는다.	DEFAULT	참여_신청하다
50	승인여부는 'Y' 또는 'N' 값을 가져야 한다.	CHECK	참여_신청하다
51	신청일시는 현재 일시(SYSDATE) 값을 갖는다.	DEFAULT	대여_신청하다
52	승인여부는 'Y' 또는 'N' 값을 가져야 한다.	CHECK	대여_신청하다
53	승인여부의 초기값은 'N'이다.	DEFAULT	대여_신청하다
54	시작일시는 신청일시보다 큰 값을 가져야 한다.	CHECK	대여_신청하다
55	종료일시는 시작일시보다 큰 값을 가져야 한다.	CHECK	대여_신청하다
56	청결상태는 'P' 또는 'D' 또는 'E' 또는 'F' 값을 가져야 한다.	CHECK	대여_신청하다
57	시작일시, 종료일시, 사용자수는 널 값을 가질 수 없다.	NOT NULL	대여_신청하다
58	청결상태의 초기값은 'P'이다.	DEFAULT	대여_신청하다
59	강사승인여부는 'Y' 또는 'N' 값을 가져야 한다.	CHECK	수강_신청하다
60	강사승인여부의 초기값은 'N'이다.	DEFAULT	수강_신청하다
61	신청일시는 현재 일시(SYSDATE) 값을 갖는다.	DEFAULT	수강_신청하다
62	진행상태의 초기값은 1(판매중)이다.	DEFAULT	판매하다
63	거래유형의 초기값이 '택배거래'이다.	DEFAULT	판매하다
64	판매회원_id와 배송비는 NULL 값을 가질 수 없다.	NOT NULL	판매하다
65	구매회원_id와 배송지는 NULL 값을 가질 수 없다.	NOT NULL	구매하다
66	구매일시는 현재 일시(SYSDATE) 값을 갖는다.	DEFAULT	구매하다
67	구매확정여부의 초기값은 'N'이다.	DEFAULT	구매하다
68	구매확정여부는 'Y' 또는 'N' 값을 가져야 한다.	CHECK	구매하다
69	점수는 -5에서 +5 사이 정수 값을 가져야 한다.	CHECK	평판점수_매기다
70	평가자구분은 '판매자' 또는 '구매자' 값을 가져야 한다.	CHECK	평판점수_매기다
71	평가자구분과 점수는 널 값을 가질 수 없다.	NOT NULL	평판점수_매기다
72	후원일시와 금액은 널 값을 가질 수 없다.	NOT NULL	후원하다
73	후원금액은 1,000 이상의 값을 가져야 한다.	CHECK	후원하다
74	후원일시는 현재 일시(SYSDATE) 값을 갖는다.	DEFAULT	후원하다
75	금액은 널 값을 가질 수 없다.	NOT NULL	편당하다
76	금액은 10,000 이상의 값을 가져야 한다.	CHECK	편당하다
77	편당일시는 현재 일시(SYSDATE) 값을 갖는다.	DEFAULT	편당하다
78	가입일시는 현재 일시(SYSDATE) 값을 갖는다.	DEFAULT	가입하다

5. 물리적 설계

[목차로](#)

(1) 내부 스키마 정의

1) 회원(member) 릴레이션

<표 42> 회원(member) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	id	id	VARCHAR2	20		PK	
2	pw	pw	VARCHAR2	20	N		
3	이름	name	VARCHAR2	10	N		
4	생년월일	birthday	NUMBER	8	N		
5	휴대폰번호	cellphone	VARCHAR2	13	N		UNIQUE
6	이메일	email	VARCHAR2	40	N		UNIQUE
7	은행명	bank	VARCHAR2	10	N		
8	계좌번호	account	VARCHAR2	20	N		UNIQUE
9	활동점수	activity_score	NUMBER	8			DEFAULT(0), CHECK (>=0)
10	평판점수	reputation_score	NUMBER	8			DEFAULT(0), CHECK (>=0)
11	등급	class	VARCHAR2	10			DEFAULT('BEGIN'), CHECK('VIP' or 'VIP' or 'GOLD' or 'SILVER' or 'BEGIN')
12	프로필 공개여부	profile_open	CHAR	1			DEFAULT('N'), CHECK ('Y', or 'N')

2) 일반회원(normal_member) 릴레이션

<표 43> 일반회원(normal_member) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	회원_id	id	VARCHAR2	20		PK, FK	FK: member(id)
2	포인트	point	NUMBER	8			DEFAULT(0), CHECK (>=0)

3) 음악가회원(musician_member) 릴레이션

<표 44> 음악가회원(musician_member) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	회원_id	id	VARCHAR2	20		PK, FK	FK: member(id)
2	학력	education	CHAR	4			CHECK('중졸' or '고졸' or '대졸')
3	작업실주소	office_addr	VARCHAR2	50			

4) 업체회원(company_member) 릴레이션

<표 45> 업체회원(company_member) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	업체회원_id	id	VARCHAR2	20		PK	
2	pw	pw	VARCHAR2	20	N		
3	이름	name	VARCHAR2	10	N		
4	생년월일	birthday	NUMBER	8			
5	휴대폰번호	cellphone	VARCHAR2	13	N		UNIQUE
6	이메일	email	VARCHAR2	40	N		UNIQUE
7	사업자등록 번호	comp_reg_num	CHAR	12	N		UNIQUE
8	업체명	comp_name	VARCHAR2	40	N		

5) 접속내역(connect_record) 릴레이션

<표 46> 접속내역(connect_record) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	회원_id	id	VARCHAR2	20		PK, FK	FK: member(id)
2	순서	seq	NUMBER	8		PK	DEFAULT(1)
3	접속일시	connection_date	DATE				DEFAULT(SYSDATE)

6) 앨범(album) 릴레이션

<표 47> 앨범(album) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	번호	num	NUMBER	8		PK	
2	발매회원_id	mus_id	VARCHAR2	20	N	FK	FK: musician_member(id)
3	제목	alb_title	VARCHAR2	50	N		
4	발매일	alb_date	DATE		N		
5	누적조회수	alb_acc_views	NUMBER	10			DEFAULT(0)
6	월별조회수	alb_month_views	NUMBER	10			DEFAULT(0)

7) 곡(song) 릴레이션

<표 50> 곡(song) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	번호	num	NUMBER	8		PK	
2	작곡자_id	mus_id	VARCHAR2	20	N	FK	FK: musician_member(id)
3	작곡일	song_date	DATE				
4	제목	song_title	VARCHAR2	50	N		
5	장르	genre	NUMBER	2	N		
6	누적조회수	song_acc_views	NUMBER	8			DEFAULT(0)
7	월별조회수	song_month_views	NUMBER	8			DEFAULT(0)

8) 공연(show) 릴레이션

<표 51> 공연(show) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	번호	num	NUMBER	8		PK	SEQUENCE: seq_show
2	제목	title	VARCHAR2	100	N		
3	장르	genre	NUMBER	2	N		
4	개최일시	start_date	DATETIME		N		
5	마감일시	end_date	DATETIME		N		CHECK (> start_date)
6	시간	running_time	NUMBER	3	N		
7	모집인원수	recruit_count	NUMBER	3	N		
8	최대관람객수	max_visitor	NUMBER	3	N		CHECK (> min_visitor)
9	최소관람객수	min_visitor	NUMBER	3	N		
10	가격	price	NUMBER	8	N		
11	관람등급	rating	VARCHAR2	20	N		
12	개최회원_id	hold_mem_id	VARCHAR2	20		FK	FK : member(id)

13	기획일	plan_date	DATE				
14	추후공개여부	is_open	CHAR	1			CHECK('Y' or 'N')

9) 장르(genre) 릴레이션

<표 49> 장르(genre) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	코드	code	NUMBER	2		PK	
2	장르구분	category	VARCHAR2	20			1: 힙합, 2: 발라드, 3: 클래식, 4: 트로트, 5: 재즈, 6: 락, 7: K-POP, 8: 댄스, 9: 성악, 10: 블루스, 11: 락발라드, 12: 인디, 13: 기타

10) 시설물(facility) 릴레이션

<표 52> 시설물(facility) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	시설물_id	id	NUMBER	8		PK	
2	시설물명	name	VARCHAR2	30	N		
3	주소	addr	VARCHAR2	50	N		UNIQUE
4	등급	rating	CHAR	1	N		CHECK ('A' or 'B' or 'C')
5	대여제한등급	rent_limit_level	VARCHAR2	6	N		CHECK ('VIP' or 'SILVER')
6	시설물구분	type	CHAR	6	N		CHECK ('공연장' or '연습실')
7	등록업체회원_id	regist_comp_id	VARCHAR2	20	N	FK	FK: company_member(id)

11) 음악강의(lecture) 릴레이션

<표 53> 음악강의(lecture) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	번호	num	NUMBER	10		PK	
2	강의명	title	VARCHAR2	50	N		
3	내용	contents	VARCHAR2	400	N		
4	수강료	tuition	NUMBER	8			DEFAULT(0)
5	장소	place	VARCHAR2	50	N		
6	대면여부	on_off	VARCHAR2	6			DEFAULT('비대면'), CHECK('대면' or '비대면')
7	개강일시	begin_day	DATE		N		

8	종강일시	end_day	DATE		N		
9	수강인원	capacity	NUMBER	2			
10	강의요일	day	CHAR	2	N		CHECK ('월' or '화' or '수' or '목' or '금' or '토')
11	시작시각	start_time	DATE		N		
12	종료시각	end_time	DATE		N		
13	음악가회원_id	musician_id	VARCHAR2	20	N	FK	FK: musician_member(id)
14	개설일시	released_time	DATE				

12) 할인쿠폰(coupon) 릴레이션

<표 54> 할인쿠폰(coupon) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	번호	num	NUMBER	10		PK	
2	할인율	disc_rate	NUMBER	3	N		
3	유효기간	expire_day	NUMBER	3			DEFAULT(30)
4	음악강의번호	lect_id	NUMBER	10		FK	FK: lecture(num)
5	발행일시	rel_time	DATE				

13) 중고물품(usedgoods) 릴레이션

<표 55> 중고물품(usedgoods) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	번호	num	NUMBER	10		PK	SEQUENCE: seq_num
2	물품명	name	VARCHAR2	50	N		
3	카테고리	category	NUMBER	1			
4	판매가	sell_price	NUMBER	10	N		
5	구매가	buy_price	NUMBER	10	N		
6	물품상태	condition	VARCHAR2	100			
7	조회수	views	NUMBER	10			DEFAULT(0)

14) 중고물품_카테고리(ug_category) 릴레이션

<표 56> 중고물품_카테고리(ug_category) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	코드	code	NUMBER	1		PK	
2	물품분류	category	VARCHAR2	20			1: 관악기, 2: 타악기, 3: 국악기, 4: 현악기, 5: 기타물품

15) 멤버십(membership) 릴레이션

<표 57> 멤버십(membership) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	번호	mbs_id	NUMBER	8		PK	
2	등록일자	regist_time	DATE		N		DEFAULT(SYSDATE)
3	가입금액	mbs_fee	NUMBER	6	N		CHECK (>= 1000)
4	제공포인트	point	NUMBER	4	N		CHECK(>=0)
5	멤버십등록자_id	regist_id	VARCHAR2	20	N	FK	FK: musician_member(id)

16) 프로젝트(project) 릴레이션

<표 58> 프로젝트(project) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	번호	proj_id	NUMBER	8		PK	
2	제목	title	VARCHAR2	40	N		
3	분야	field	VARCHAR2	4	N		CHECK('곡' or '앨범' or '공연')
4	세부설명	detail	VARCHAR2	400	N		
5	목표후원금액	target_amount	NUMBER	13	N		
6	프로젝트기획자_id	pj_id	VARCHAR2	20		FK	FK: member(id)
7	시작일자	start_day	DATE				
8	종료일자	end_day	DATE				CHECK(> start_day)

17) 수록하다(song_in_album) 릴레이션

<표 59> 수록하다(song_in_album) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	곡_번호	song_num	NUMBER	8		PK, FK	FK: song(song_num)
2	앨범_번호	album_num	NUMBER	8		PK, FK	FK: album(alb_num)

18) 예매하다(show_reserve) 릴레이션

<표 60> 예매하다(show_reserve) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	공연_번호	show_num	NUMBER	8		PK, FK	FK : show(num)
2	회원_id	mem_id	VARCHAR2	20		PK, FK	FK : member(id)
3	예매일시	reserve_date	DATE				DEFAULT(SYSDATE)
4	수량	quantity	NUMBER	2			DEFAULT(1)

19) 참여_신청하다(show_apply) 릴레이션

<표 61> 참여_신청하다(show_apply) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	공연_번호	show_num	NUMBER	8		PK, FK	FK : show(num)
2	회원_id	mem_id	VARCHAR2	20		PK, FK	FK : member(id)
3	신청일시	appli_date	DATE				DEFAULT(SYSDATE)
4	역할	role	VARCHAR2	40			
5	승인여부	approval	CHAR	1			CHECK('Y' or 'N')

20) 관심_등록하다(show_like) 릴레이션

<표 62> 관심_등록하다(show_like) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	회원_id	mem_id	VARCHAR2	20		PK, FK	FK : member(id)
2	공연_번호	show_num	NUMBER	8		PK, FK	FK : show(num)

21) 대여_신청하다(rental_apply) 릴레이션

<표 63> 대여_신청하다(rental_apply) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	회원_id	mem_id	VARCHAR2	20		PK, FK	FK: member(id)
2	시설물_id	facil_id	NUMBER	8		PK, FK	FK: facility(id)
3	시작일시	rent_start	DATE		N	PK	CHECK(>= 신청일시)
4	승인여부	approval	CHAR	1			DEFAULT('N') CHECK('Y' or 'N')
5	신청일시	rent_apply	DATE				DEFAULT(SYSDATE)
6	종료일시	rent_end	DATE		N		CHECK(> 시작일시)
7	사용자수	user_num	NUMBER	3	N		
8	청결상태	clean	CHAR	1			DEFAULT('P'), CHECK('P' or 'D' or 'E' or 'F')

22) 수강_신청하다(lecture_apply) 릴레이션

<표 64> 수강_신청하다(lecture_apply) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	회원_id	mem_id	VARCHAR2	20		PK, FK	FK : normal_member(id)
2	음악강의번호	lect_num	NUMBER	10		PK, FK	FK : lecture(num)
3	신청일시	apply_date	DATE			PK	DEFAULT(SYSDATE)
4	강사승인여부	approval	CHAR	1			DEFAULT('N') CHECK('Y' or 'N')

23) 소유하다(get_coupon) 릴레이션

<표 65> 소유하다(get_coupon) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	회원_id	mem_id	VARCHAR2	20		PK, FK	FK : normal_member(id)
2	할인쿠폰번호	coupon_num	NUMBER	10		PK, FK	FK : coupon(num)
3	일시	get_date	DATE				DEFAULT(SYSDATE)

24) 판매하다(sell) 릴레이션

<표 66> 판매하다(sell) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	중고물품번호	ugoods_num	NUMBER	10		PK, FK	FK: usedgoods(num)
2	판매회원_id	seller_id	VARCHAR2	20	N	FK	FK: member(id)
3	등록일시	reg_date	DATE				DEFAULT(SYSDATE)
4	배송비	delivery_fee	NUMBER	5	N		DEFAULT(0)
5	진행상태	sell_status	NUMBER	1			
6	거래유형	trans_type	VARCHAR2	10			DEFAULT('택배거래')

25) 진행상태(sales_status) 릴레이션

<표 67> 진행상태(sales_status) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	코드	code	NUMBER	1		PK	
2	판매상태	status	VARCHAR2	20			1: 판매중, 2: 예약중, 3: 판매취소, 4: 판매완료

26) 구매하다(buy) 릴레이션

<표 68> 구매하다(buy) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	중고물품번호	ugoods_id	NUMBER	10		PK, FK	FK: usedgoods(num)
2	구매회원_id	buyer_id	VARCHAR2	20	N	FK	FK: member(id)
3	구매일시	buy_date	DATE				DEFAULT(SYSDATE)
4	배송지	ship_addr	VARCHAR2	100	N		
5	구매확정여부	confirm	CHAR	1			DEFAULT('N') CHECK('Y' or 'N')

27) 찜하다(wishlist) 릴레이션

<표 69> 찜하다(wishlist) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	회원_id	mem_id	VARCHAR2	20		PK, FK	FK : member(id)
2	중고물품번호	ugoods_id	NUMBER	10		PK, FK	FK: usedgoods(id)

28) 평판점수_매기다(score_reputation) 릴레이션

<표 70> 평판점수_매기다(score_reputation) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	중고물품번호	ugoods_id	NUMBER	10		PK, FK	FK: usedgoods(id)
2	구매회원_id	buyer_id	VARCHAR2	20		PK, FK	FK: member(id)
3	평가자구분	eval_class	CHAR	6	N		CHECK('판매자' or '구매자')
4	점수	score	NUMBER	1	N		CHECK(>= -5 and <= 5)
5	이유	reason	VARCHAR2	100			

29) 후원하다(sponsor) 릴레이션

<표 71> 후원하다(sponsor) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	후원번호	spon_num	NUMBER	8		PK	SEQUENCE: seq_spon
2	일반회원_id	norm_id	VARCHAR2	20		FK	FK: normal_member(id)
3	음악가회원_id	music_id	VARCHAR2	20		FK	FK: musician_memberid)
4	후원일시	spon_time	DATE		N		DEFAULT(SYSDATE)
5	금액	spon_amount	NUMBER	8	N		CHECK(>= 1000)

30) 펀딩하다(funding) 릴레이션

<표 72> 펀딩하다(funding) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	프로젝트번호	proj_num	NUMBER	8		PK, FK	FK: project(proj_id)
2	펀딩회원_id	mem_id	VARCHAR2	20		PK, FK	FK: member(id)
3	금액	fund_amount	NUMBER	8	N		CHECK(>= 10000)
4	펀딩일시	fund_time	DATE		N		DEFAULT(SYSDATE)

31) 가입하다(join) 릴레이션

<표 73> 가입하다(join) 릴레이션의 내부 스키마

No.	속성	칼럼 이름	데이터타입	크기	NULL 허용	키	기타
1	가입번호	join_num	NUMBER	8		PK	SEQUENCE: seq_join
2	가입자_id	join_id	VARCHAR2	20		FK	FK: normal_member(id)
3	멤버십번호	mbs_num	NUMBER	8		FK	FK: membership(mbs_id)
4	가입일자	join_time	DATE		N		DEFAULT(SYSDATE)

(2) 인덱스 정의

1) 인덱스 정의서

<표 74> 청년 음악가 후원 웹사이트용 DB의 인덱스 정의서

릴레이션명	속성명	테이블명	칼럼명	인덱스명	데이터 타입	정렬
업체회원	업체명	company_member	comp_name	company_idx	VARCHAR2(40)	ASC
공연	장르	show	genre	genre_idx	NUMBER(1)	ASC
중고물품	물품명	usedgoods	name	ugname_idx	VARCHAR2(50)	ASC

2) 인덱스 정의의 적정성 평가

<표 75> 청년 음악가 후원 웹사이트용 DB의 인덱스 적정성 평가

인덱스명	분포도	용도	적정 여부
company_idx	$1 / 47 * 100 = 1.06 / 50 * 100 \approx 2.12\%$ (업체회원 수가 50개이고, 서로 다른 회사명이 47개라고 가정함)	업체명이 SQL문의 WHERE절과 ORDER BY절에서 자주 사용됨	○
genre_idx	$1 / 13 * 100 = 9.2 / 120 * 100 \approx 7.7\%$ (공연 수가 총 120개라고 가정하고, 서로 다른 장르 수는 13개임)	장르가 SQL문의 GROUP BY절에 서 자주 사용됨	○
ugname_idx	$1 / 36 * 100 = 1.11 / 40 * 100 \approx 2.78\%$ (중고물품 수가 총 40개이고, 서로 다른 물품명이 36개라고 가정함)	물품명이 SQL문의 WHERE절에서 자주 사용됨	○

6. 구현 및 테스트

[목차로](#)

(1) DB 생성 및 초기 데이터 삽입

1) 회원 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE member(
  id VARCHAR2(20) PRIMARY KEY,
  pw VARCHAR2(20) NOT NULL,
  name VARCHAR2(10) NOT NULL,
  birthday NUMBER(8) NOT NULL,
  cellphone VARCHAR2(13) NOT NULL,
  email VARCHAR2(40) NOT NULL,
  bank VARCHAR2(10) NOT NULL,
  account VARCHAR2(20) NOT NULL,
  activity_score NUMBER(8) DEFAULT(0),
  reputation_score NUMBER(8) DEFAULT(0),
  class VARCHAR2(10) DEFAULT('BEGIN'),
  profile_open CHAR(1) DEFAULT('N'),
  CONSTRAINT member_unique_column UNIQUE(cellphone, email, account),
  CONSTRAINT member_check_act_score CHECK(activity_score >= 0),
  CONSTRAINT member_check_rep_score CHECK(reputation_score >= 0),
  CONSTRAINT member_check_class CHECK(class IN ('VIP', 'VIP', 'GOLD', 'SILVER', 'BEGIN')),
  CONSTRAINT member_check_profile_open CHECK(profile_open IN ('Y', 'N'))
);

INSERT INTO member VALUES ('kwy', '@kwy11', '강원용', 19990218, '010-1234-5678',
'wy@koreatech.ac.kr', '농협', '111-1111-1111-11', 70, 30, 'VIP', 'Y');
INSERT INTO member VALUES ('ysj', '@ysj11', '여승준', 19980603, '010-1111-5678', 'ysj@koreatech.ac.kr',
' 신한', '222-2222-2222-22', 100, 50, 'VIP', 'Y');
INSERT INTO member VALUES ('chws', '@chws123', '조원석', 19981226, '010-2222-5678',
'chws@koreatech.ac.kr', '수협', '333-3333-3333-33', DEFAULT, DEFAULT, DEFAULT, DEFAULT);
INSERT INTO member VALUES ('sgj', '@sgj1235', '심규재', 20001016, '010-3333-5678',
'sgj@koreatech.ac.kr', '기업', '444-4444-4444-44', 40, 50, 'VIP', 'Y');
INSERT INTO member VALUES ('csy', '@csy1513', '최수연', 20010101, '010-5555-5678',
'csy@koreatech.ac.kr', ' 신한', '555-5555-5555-55', 50, 40, 'SILVER', DEFAULT);
INSERT INTO member VALUES ('mus_kwy', '@mus_kwy14', '이원용', 19990218, '010-1234-7777',
'mus_kwy@koreatech.ac.kr', '농협', '666-6666-6666-66', 30, 30, 'SILVER', DEFAULT);
INSERT INTO member VALUES ('mus_ysj', '@mus_ysj1342', '박승준', 19980603, '010-1234-9999',
'mus_ysj@koreatech.ac.kr', '국민', '777-7777-7777-77', DEFAULT, DEFAULT, 'GOLD', 'Y');
INSERT INTO member VALUES ('mus_chws', '@mus_chws123', '이원미', 19990218, '010-1256-1278',
'mus_chws@koreatech.ac.kr', '농협', '888-8888-8888-88', 50, 0, 'GOLD', DEFAULT);
INSERT INTO member VALUES ('mus_sgj', '@mus_sgj63', '송규재', 19990218, '010-1124-5688',
'mus_sgj@koreatech.ac.kr', '기업', '999-9999-9999-99', 100, 70, 'VIP', 'Y');
INSERT INTO member VALUES ('mus_csy', 'mus_csy56', '김수연', 20010101, '010-1584-5238',
'mus_csy@koreatech.ac.kr', ' 신한', '101-0101-0101-10', 70, 50, 'VIP', DEFAULT)
```

2) 일반회원 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE normal_member(
    id VARCHAR2(20) PRIMARY KEY,
    point NUMBER(8) DEFAULT(0) CHECK(point >=0),
    FOREIGN KEY(id) REFERENCES member(id) ON DELETE CASCADE,
);

INSERT INTO normal_member VALUES('kwy', 1000);
INSERT INTO normal_member VALUES('ysj', 3000);
INSERT INTO normal_member VALUES('chws', 5000);
INSERT INTO normal_member VALUES('sgj', 2000);
INSERT INTO normal_member VALUES('csy', 2000);

```

3) 음악가회원 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE musician_member(
    id VARCHAR2(20),
    education VARCHAR2(5) NOT NULL,
    office_addr VARCHAR2(50),
    PRIMARY KEY(id),
    FOREIGN KEY(id) REFERENCES member(id) ON DELETE CASCADE,
    CONSTRAINT check_mus_education CHECK(education IN ('중졸', '고졸', '대졸'))
);

INSERT INTO musician_member VALUES('mus_kwy', '고졸', '오산시 오산대로역로 100');
INSERT INTO musician_member VALUES('mus_ysj', '대졸', '청주시 상당구 사직대로 40');
INSERT INTO musician_member VALUES('mus_chws', '대졸', '대전시 유성구 한밭대로 320');
INSERT INTO musician_member VALUES('mus_sgj', '고졸', '서울 종로구 북촌로 20-2');
INSERT INTO musician_member VALUES('mus_csy', '대졸', '천안시 서북구 불당2로 10');

```

4) 업체회원 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE company_member(
    id VARCHAR2(20),
    pw VARCHAR2(20) NOT NULL,
    name VARCHAR2(10) NOT NULL,
    birthday NUMBER(8) NOT NULL,
    cellphone VARCHAR2(13) NOT NULL,
    email VARCHAR2(40) NOT NULL,
    comp_reg_num CHAR(12) NOT NULL,
    comp_name VARCHAR2(40) NOT NULL,
    CONSTRAINT pk_company_id PRIMARY KEY(id),
    CONSTRAINT unique_comp UNIQUE(cellphone, email, comp_reg_num)
);

INSERT INTO company_member VALUES ('rarepeace1', 'FWISadh7l9GY', '장도윤', 19990218, '010-6580-7789', 'nalton0@naver.com', '913-90-79313', '레어피스');
INSERT INTO company_member VALUES ('rarepeace2', 'jQrpVX', '김지유', 19890603, '010-3237-8607', 'gagular1@daum.net', '913-90-79313', '레어피스');
INSERT INTO company_member VALUES ('rarepeace3', 'AKJfWqP', '박채원', 20000620, '010-4769-5489', 'kvaldes2@gmail.com', '913-90-79313', '레어피스');

```

```

INSERT INTO company_member VALUES ('doublerosen1', 'l0P8vbjn0f', '정현우', 19950818,
'010-5825-4490', 'hwallage3@gmail.com', '186-54-03951', '더블로젠');
INSERT INTO company_member VALUES ('doublerosen2', 'Uy8bmFvjWAL', '송지훈', 20011212,
'010-6564-6464', 'somahoney4@koreatech.ac.kr', '186-54-03951', '더블로젠');
INSERT INTO company_member VALUES ('doublerosen3', 'Y6DNlCR', '차은우', 19921031,
'010-5293-0908', 'tbenkin5@naver.com', '1865403951', '더블로젠');
INSERT INTO company_member VALUES ('doublerosen4', 'sb24jq8', '인유찬', 19880829, '010-8457-3223',
'kbarnaclough6@gmail.com', '186-54-03951', '더블로젠');
INSERT INTO company_member VALUES ('dreamcompany1', 'oQqElfKekgf', '이서영', 19740912,
'010-2575-6715', 'xforrestor7@naver.com', '457-444-9671', '드림컴퍼니');
INSERT INTO company_member VALUES ('dreamcompany2', 'fZ9Ox1DK', '김한결', 19980317,
'010-9266-8705', 'emarques8@koreatech.ac.kr', '457-44-49671', '드림컴퍼니');
INSERT INTO company_member VALUES ('dreamcompany3', 'k6ldf5CPiLck', '신재민', 20020927,
'010-3200-2256', 'nbraznell9@naver.com', '457-44-49671', '드림컴퍼니');
INSERT INTO company_member VALUES ('nj1', 'wq0Tgs', '안유빈', 20020809, '010-3368-1935',
'tcorresa@naver.com', '710-03-46731', '앤제이');
INSERT INTO company_member VALUES ('nj2', 'FnEXmj3yA4', '장준호', 19990130, '010-4722-7953',
'rbarbieb@gmail.com', '710-03-46731', '앤제이');
INSERT INTO company_member VALUES ('nj3', '1tpHfXxh99h', '이지한', 19950510, '010-6437-1995',
'kdombrellc@koreatech.ac.kr', '7100-34-6731', '앤제이');
INSERT INTO company_member VALUES ('nj4', 'qYca9G', '윤채아', 20020929, '010-1858-6392',
'jsnowballd@daum.net', '710-03-46731', '앤제이');
INSERT INTO company_member VALUES ('aishus1', 'ys2o23Hc', '전지현', 19960610, '010-2174-2909',
'jdownse@naver.com', '2029729030', '에이셔널');
INSERT INTO company_member VALUES ('aishus2', 'akwJ6E', '최우주', 20000715, '010-4745-7789',
'mfranchionif@gmail.com', '202-97-29030', '에이셔널');
INSERT INTO company_member VALUES ('aishus3', 'vmlk19ZPd4rQ', '심민찬', 19940131,
'010-8239-6382', 'ceversg@koreatech.ac.kr', '202-97-29030', '에이셔널');
INSERT INTO company_member VALUES ('aishus4', 'NZDVZHep5Kv', '유다인', 20010525,
'010-4832-9169', 'zpithcockh@koreatech.ac.kr', '202-97-29030', '에이셔널');
INSERT INTO company_member VALUES ('aishus5', 'hODTCKgys', '김시후', 19850118, '010-1563-6563',
'twilsherei@daum.net', '202-97-29030', '에이셔널');
INSERT INTO company_member VALUES ('aishus6', '1pIKWIHevY6', '이주연', 19970325, '010-2648-3384',
'daistonj@gmail.com', '202-97-29030', '에이셔널');

```

5) 접속내역 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE connect_record(
    id VARCHAR2(20),
    seq NUMBER(8) DEFAULT(1),
    connection_date DATE DEFAULT SYSDATE,
    PRIMARY KEY(id, seq),
    FOREIGN KEY(id) REFERENCES member(id)
);

INSERT INTO connect_record VALUES ('kwy', 1, TO_DATE('2023-03-10 16:00','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('kwy', 2, TO_DATE('2023-03-12 09:10','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('kwy', 3, TO_DATE('2023-03-15 19:05','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('ysj', 1, TO_DATE('2023-03-15 08:10','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('ysj', 2, TO_DATE('2023-03-17 18:20','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('chws', 1, TO_DATE('2023-03-15 09:15','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('chws', 2, TO_DATE('2023-03-17 08:40','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('chws', 3, TO_DATE('2023-03-20 19:20','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('sgj', 1, TO_DATE('2023-03-16 12:10','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('sgj', 2, TO_DATE('2023-03-17 12:15','YYYY-MM-DD HH24:MI'));

```

```

INSERT INTO connect_record VALUES ('sgj', 3, TO_DATE('2023-03-18 13:20','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('csy', 1, TO_DATE('2023-03-20 14:20','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('csy', 2, TO_DATE('2023-03-20 19:45','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('mus_kwy', 1, TO_DATE('2023-03-19 11:20','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('mus_ysj', 1, TO_DATE('2023-03-22 14:55','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('mus_ysj', 2, TO_DATE('2023-03-22 16:08','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('mus_ysj', 3, TO_DATE('2023-03-24 20:00','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('mus_ysj', 4, TO_DATE('2023-03-25 05:10','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('mus_chws', 1, TO_DATE('2023-03-20 12:05','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('mus_sgj', 1, TO_DATE('2023-04-01 12:00','YYYY-MM-DD HH24:MI'));
INSERT INTO connect_record VALUES ('mus_csy', 1, TO_DATE('2023-04-02 14:15','YYYY-MM-DD HH24:MI'));

```

6) 앨범 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE album(
    num NUMBER(8) PRIMARY KEY,
    mus_id VARCHAR2(20) NOT NULL,
    alb_title VARCHAR2(50) NOT NULL,
    alb_date DATE NOT NULL,
    alb_acc_views NUMBER(10) DEFAULT(0),
    alb_month_views NUMBER(10) DEFAULT(0),
    FOREIGN KEY(mus_id) REFERENCES musician_member(id) ON DELETE CASCADE
);

INSERT INTO album VALUES (1, 'mus_kwy', '고향', TO_DATE('03/01/2023', 'MM/DD/YYYY'), 2000, 120);
INSERT INTO album VALUES (2, 'mus_ysj', '가을 향기', TO_DATE('05/10/2023', 'MM/DD/YYYY'), 2200, 300);
INSERT INTO album VALUES (3, 'mus_chws', '꽃피는 봄', TO_DATE('06/12/2023', 'MM/DD/YYYY'), 1800, 150);
INSERT INTO album VALUES (4, 'mus_sgj', '책갈피', TO_DATE('07/20/2023', 'MM/DD/YYYY'), 1550, 105);
INSERT INTO album VALUES (5, 'mus_csy', '가요 100년', TO_DATE('08/10/2023', 'MM/DD/YYYY'), 1200, 85);

```

7) 장르 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE genre(
    code    NUMBER(2) PRIMARY KEY,
    category VARCHAR2(20) UNIQUE NOT NULL
);

INSERT INTO genre VALUES (1, '힙합');
INSERT INTO genre VALUES (2, '발라드');
INSERT INTO genre VALUES (3, '클래식');
INSERT INTO genre VALUES (4, '트로트');
INSERT INTO genre VALUES (5, '재즈');
INSERT INTO genre VALUES (6, '락');
INSERT INTO genre VALUES (7, 'K-POP');
INSERT INTO genre VALUES (8, '댄스');
INSERT INTO genre VALUES (9, '성악');
INSERT INTO genre VALUES (10, '블루스');
INSERT INTO genre VALUES (11, '락발라드');
INSERT INTO genre VALUES (12, '인디');
INSERT INTO genre VALUES (13, '기타');
```

8) 곡 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE song(
    num    NUMBER(8) PRIMARY KEY,
    mus_id VARCHAR2(20) NOT NULL,
    song_title VARCHAR2(50) NOT NULL,
    song_date DATE,
    genre  NUMBER(2) NOT NULL,
    song_acc_views NUMBER(8) DEFAULT(0),
    song_month_views NUMBER(8) DEFAULT(0),
    FOREIGN KEY(mus_id) REFERENCES member(id) ON DELETE CASCADE
);

INSERT INTO song VALUES (1, 'mus_kwy', 'tomboy', TO_DATE('03/10/2023', 'MM/DD/YYYY'), 7, 250, 30);
INSERT INTO song VALUES (2, 'mus_ysj', 'YOU', TO_DATE('03/12/2023', 'MM/DD/YYYY'), 2, 155, 10);
INSERT INTO song VALUES (3, 'mus_chws', '찐이야', TO_DATE('03/15/2023', 'MM/DD/YYYY'), 4, 130, 10);
INSERT INTO song VALUES (4, 'mus_sgji', 'square', TO_DATE('03/20/2023', 'MM/DD/YYYY'), 5, 50, 10);
INSERT INTO song VALUES (5, 'mus_csy', 'lalala lovesong', TO_DATE('03/25/2023', 'MM/DD/YYYY'), 5, 100, 20);
INSERT INTO song VALUES (6, 'mus_kwy', 'maria', TO_DATE('04/05/2023', 'MM/DD/YYYY'), 8, 80, 30);
INSERT INTO song VALUES (7, 'mus_kwy', 'tomboy', TO_DATE('04/10/2023', 'MM/DD/YYYY'), 9, 50, 5);
INSERT INTO song VALUES (8, 'mus_ysj', '봄여름가을겨울', TO_DATE('04/15/2023', 'MM/DD/YYYY'), 12, 100, 20);
INSERT INTO song VALUES (9, 'mus_chws', 'lie', TO_DATE('04/20/2023', 'MM/DD/YYYY'), 11, DEFAULT, DEFAULT);
INSERT INTO song VALUES (10, 'mus_sgji', '니가 뭔데', TO_DATE('04/22/2023', 'MM/DD/YYYY'), 3, 80, 10);
INSERT INTO song VALUES (11, 'mus_csy', 'lalala lovesong', TO_DATE('05/01/2023', 'MM/DD/YYYY'), 2, 100, 20);
INSERT INTO song VALUES (12, 'mus_kwy', 'let it go', TO_DATE('05/10/2023', 'MM/DD/YYYY'), 10, 90, 12);
INSERT INTO song VALUES (13, 'mus_kwy', 'pain killer', TO_DATE('05/14/2023', 'MM/DD/YYYY'), 7, 100, 10);
INSERT INTO song VALUES (14, 'mus_ysj', '빠빠빠', TO_DATE('05/15/2023', 'MM/DD/YYYY'), 8, 50, 5);
INSERT INTO song VALUES (15, 'mus_chws', '땡땡땡', TO_DATE('05/20/2023', 'MM/DD/YYYY'), 4, 100, 30);
INSERT INTO song VALUES (16, 'mus_sgji', '곰세마리', TO_DATE('06/11/2023', 'MM/DD/YYYY'), 3, 150, 20);
INSERT INTO song VALUES (17, 'mus_csy', '호랑나비', TO_DATE('06/12/2023', 'MM/DD/YYYY'), 6, 55, 15);
INSERT INTO song VALUES (18, 'mus_kwy', '남만고양이', TO_DATE('06/17/2023', 'MM/DD/YYYY'), 1, 40, 15);
INSERT INTO song VALUES (19, 'mus_kwy', '눈', TO_DATE('06/20/2023', 'MM/DD/YYYY'), 8, 50, 10);
INSERT INTO song VALUES (20, 'mus_ysj', '작은별', TO_DATE('06/25/2023', 'MM/DD/YYYY'), 11, 80, 10);
INSERT INTO song VALUES (21, 'mus_chws', '알루미늄', TO_DATE('07/10/2023', 'MM/DD/YYYY'), 6, 50, 10);
INSERT INTO song VALUES (22, 'mus_sgji', '무제', TO_DATE('4/10/2023', 'MM/DD/YYYY'), 2, 50, 20);
INSERT INTO song VALUES (23, 'mus_csy', '박하사탕', TO_DATE('07/20/2023', 'MM/DD/YYYY'), 9, 70, 30);
INSERT INTO song VALUES (24, 'mus_kwy', '이밤이지나면', TO_DATE('07/20/2023', 'MM/DD/YYYY'), 1, 60, 25);
```

9) 공연 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE show(
    num NUMBER(8) PRIMARY KEY,
    title VARCHAR2(100) NOT NULL,
    genre NUMBER(2) NOT NULL,
    start_date DATETIME NOT NULL,
    end_date DATETIME NOT NULL,
    running_time NUMBER(3) NOT NULL,
    recruit_count NUMBER(3) NOT NULL,
    rating VARCHAR2(20) NOT NULL,
    price NUMBER(8) NOT NULL,
    max_visitor NUMBER(3) NOT NULL,
    min_visitor NUMBER(3) NOT NULL,
    hold_mem_id VARCHAR2(20),
    plan_date DATE,
    is_open CHAR(1),
    CONSTRAINT check_end_date CHECK(end_date >= start_date),
    CONSTRAINT check_max_visitor CHECK(max_visitor >= min_visitor),
    CONSTRAINT check_show_open CHECK(is_open in ('Y', 'N')),
    CONSTRAINT fk_hold_mem_id FOREIGN KEY(hold_mem_id) REFERENCES member(id)
    ON DELETE CASCADE
);

INSERT INTO show VALUES (1, '다 같이 놀아봐요', 2, TO_DATE('2023-05-20 20:00','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-05-30 20:00','YYYY-MM-DD HH24:MI'), 60, 5, '7세 이상', 10000, 200, 20, 'kwy',
TO_DATE('2023-05-01','YYYY-MM-DD '), 'Y');
INSERT INTO show VALUES (2, '신데렐라', 1, TO_DATE('2023-03-09 12:30','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-03-10 23:00','YYYY-MM-DD HH24:MI'), 120, 5, '12세 이상', 7000, 120, 10, 'mus_kwy',
TO_DATE('2023-01-01','YYYY-MM-DD'), 'N');
INSERT INTO show VALUES (3, '올드 송', 2, TO_DATE('2023-02-12 14:40','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-02-14 18:00','YYYY-MM-DD HH24:MI'), 100, 5, '15세 이상', 8000, 200, 30, 'mus_csy',
TO_DATE('2023-01-01','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (4, '한국 대표 가요', 2, TO_DATE('2023-05-14 22:50','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-05-20 14:00','YYYY-MM-DD HH24:MI'), 70, 5, '전체관람가', 12000, 300, 30, 'csy',
TO_DATE('2023-01-01','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (5, '달려라 대한민국', 5, TO_DATE('2023-07-01 10:00','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-07-10 16:00','YYYY-MM-DD HH24:MI'), 150, 5, '7세 이상', 14000, 500, 25, 'csy',
TO_DATE('2023-05-12','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (6, '다같이 파이팅!', 6, TO_DATE('2023-10-31 14:00','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-12-01 18:00','YYYY-MM-DD HH24:MI'), 200, 5, '19세 이상', 20000, 700, 35, 'sgj',
TO_DATE('2023-07-01','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (7, '아, 대한민국', 5, TO_DATE('2023-10-20 18:20','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-11-20 20:00','YYYY-MM-DD HH24:MI'), 140, 5, '15세 이상', 30000, 800, 40, 'mus_sgj',
TO_DATE('2023-03-15','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (8, '우리 함께', 10, TO_DATE('2023-02-12 17:40','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-03-30 20:00','YYYY-MM-DD HH24:MI'), 200, 5, '12세 이상', 8000, 900, 50, 'kwy',
TO_DATE('2023-01-01','YYYY-MM-DD'), 'N');
INSERT INTO show VALUES (9, '아자, 아자, 파이팅', 11, TO_DATE('2023-01-14 16:30','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-02-20 20:00','YYYY-MM-DD HH24:MI'), 90, 5, '7세 이상', 9000, 100, 10, 'mus_chws',
TO_DATE('2022-12-01','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (10, '노래는 나의 인생', 12, TO_DATE('2023-04-01 18:30','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-04-10 20:00','YYYY-MM-DD HH24:MI'), 60, 5, '12세 이상', 12000, 100, 10, 'sgj',
TO_DATE('2023-03-10','YYYY-MM-DD'), 'N');
INSERT INTO show VALUES (11, '다같이 차차차', 2, TO_DATE('2023-02-05 18:40','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-03-20 20:00','YYYY-MM-DD HH24:MI'), 120, 5, '12세 이상', 14000, 100, 10, 'kwy',
TO_DATE('2023-01-10','YYYY-MM-DD'), 'N');
INSERT INTO show VALUES (12, '푸른 하늘', 2, TO_DATE('2023-01-07 22:50','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-02-20 20:00','YYYY-MM-DD HH24:MI'), 60, 5, '15세 이상', 10000, 200, 30, 'kwy',
TO_DATE('2022-12-25','YYYY-MM-DD'), 'Y');
```

```

INSERT INTO show VALUES (13, '평화, 사랑', 9, TO_DATE('2023-12-27 23:40','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-12-31 20:00','YYYY-MM-DD HH24:MI'), 120, 5, '19세 이상', 12000, 300, 30, 'chws',
TO_DATE('2023-06-01','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (14, '전쟁과 평화', 8, TO_DATE('2023-06-16 05:50','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-06-20 20:00','YYYY-MM-DD HH24:MI'), 140, 5, '전체관람가', 14000, 150, 15, 'mus_chws',
TO_DATE('2023-05-30','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (15, '사랑과 믿음', 8, TO_DATE('2023-07-18 13:40','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-08-20 20:00','YYYY-MM-DD HH24:MI'), 30, 5, '전체관람가', 17000, 200, 20, 'chws',
TO_DATE('2023-06-12','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (16, '노래로 위로를', 7, TO_DATE('2023-04-10 12:20','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-05-20 20:00','YYYY-MM-DD HH24:MI'), 40, 5, '전체관람가', 8000, 300, 30, 'mus_chws',
TO_DATE('2023-03-02','YYYY-MM-DD'), 'N');
INSERT INTO show VALUES (17, '사랑, 소망, 믿음', 7, TO_DATE('2023-10-20 21:00','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-10-22 20:00','YYYY-MM-DD HH24:MI'), 60, 5, '전체관람가', 9000, 400, 50, 'kwy',
TO_DATE('2023-09-27','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (18, '소리 높여 불러요', 6, TO_DATE('2023-11-11 22:10','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-11-20 20:00','YYYY-MM-DD HH24:MI'), 60, 5, '전체관람가', 10000, 600, 50, 'kwy',
TO_DATE('2023-10-30','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (19, '어디로 가나요', 6, TO_DATE('2023-11-01 23:20','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-11-20 20:00','YYYY-MM-DD HH24:MI'), 90, 5, '전체관람가', 10000, 700, 10, 'mus_kwy',
TO_DATE('2023-08-20','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (20, '푸른 하늘', 10, TO_DATE('2023-07-01 10:30','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-07-20 20:00','YYYY-MM-DD HH24:MI'), 90, 5, '전체관람가', 500, 100, 10, 'mus_kwy',
TO_DATE('2023-05-12','YYYY-MM-DD'), 'Y');
INSERT INTO show VALUES (21, '쇼, 쇼, 쇼', 10, TO_DATE('2023-09-05 08:50','YYYY-MM-DD HH24:MI'),
TO_DATE('2023-09-20 20:00','YYYY-MM-DD HH24:MI'), 90, 5, '12세 이상', 6000, 500, 10, 'mus_chws',
TO_DATE('2023-07-11','YYYY-MM-DD'), 'Y');

```

10) 시설물 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE facility(
    id NUMBER(8),
    name VARCHAR2(30) NOT NULL,
    addr VARCHAR2(50) NOT NULL UNIQUE,
    rating CHAR(1) NOT NULL,
    rent_limit_level VARCHAR2(6) NOT NULL,
    type CHAR(6) NOT NULL CHECK(type IN ('공연장', '연습실')),
    regist_comp_id VARCHAR2(20) NOT NULL,
    CONSTRAINT pk_facility PRIMARY KEY(id),
    CONSTRAINT fk_regist_comp_id FOREIGN KEY(regist_comp_id)
        REFERENCES company_member(id) ON DELETE CASCADE,
    CONSTRAINT check_rating_facility CHECK(rating IN ('A', 'B', 'C')),
    CONSTRAINT check_rent_limit_level CHECK(rent_limit_level IN ('VIP', 'SILVER')),
);

INSERT INTO facility VALUES (1, '한빛관', '한빛관 1동', 'A', 'SILVER', '연습실', 'rarepeace1');
INSERT INTO facility VALUES (2, '한빛관', '한빛관 2동', 'B', 'SILVER', '연습실', 'rarepeace2');
INSERT INTO facility VALUES (3, '한빛관', '한빛관 3동', 'C', 'SILVER', '연습실', 'rarepeace3');
INSERT INTO facility VALUES (4, '함지관', '함지관 1동', 'A', 'VIP', '공연장', 'doublerosen1');
INSERT INTO facility VALUES (5, '예술관', '예술관 1동', 'A', 'SILVER', '연습실', 'doublerosen1');
INSERT INTO facility VALUES (6, '예술관', '예술관 2동', 'B', 'SILVER', '연습실', 'doublerosen2');
INSERT INTO facility VALUES (7, '예술관', '예술관 3동', 'C', 'SILVER', '연습실', 'doublerosen3');
INSERT INTO facility VALUES (8, '은술관', '은술관 1동', 'A', 'VIP', '공연장', 'doublerosen2');
INSERT INTO facility VALUES (9, '은술관', '은술관 2동', 'B', 'VIP', '공연장', 'doublerosen3');
INSERT INTO facility VALUES (10, '은술관', '은술관 3동', 'C', 'VIP', '공연장', 'doublerosen4');
INSERT INTO facility VALUES (11, '청술관', '청술관 1동', 'A', 'VIP', '공연장', 'dreamcompany1');

```

```

INSERT INTO facility VALUES (12, '청솔관', '청솔관 2동', 'B', 'VIP', '공연장', 'dreamcompany2');
INSERT INTO facility VALUES (13, '청솔관', '청솔관 3동', 'C', 'VIP', '공연장', 'dreamcompany3');
INSERT INTO facility VALUES (14, '해솔관', '해솔관 1동', 'A', 'SILVER', '연습실', 'nj1');
INSERT INTO facility VALUES (15, '해솔관', '해솔관 2동', 'B', 'SILVER', '연습실', 'nj2');
INSERT INTO facility VALUES (16, '해솔관', '해솔관 3동', 'C', 'SILVER', '연습실', 'nj3');
INSERT INTO facility VALUES (17, '별빛관', '별빛관 1동', 'A', 'VIP', '공연장', 'nj3');
INSERT INTO facility VALUES (18, '별빛관', '별빛관 2동', 'B', 'VIP', '공연장', 'nj3');
INSERT INTO facility VALUES (19, '별빛관', '별빛관 3동', 'C', 'VIP', '공연장', 'nj4');
INSERT INTO facility VALUES (20, '솔지관', '솔지관 1동', 'A', 'VIP', '공연장', 'aishus1');
INSERT INTO facility VALUES (21, '솔지관', '솔지관 2동', 'B', 'VIP', '공연장', 'aishus2');
INSERT INTO facility VALUES (22, '디비관', '디비관 1동', 'A', 'SILVER', '연습실', 'aishus3');
INSERT INTO facility VALUES (23, '디비관', '디비관 2동', 'B', 'SILVER', '연습실', 'aishus4');
INSERT INTO facility VALUES (24, '디비관', '디비관 3동', 'B', 'SILVER', '연습실', 'aishus5');
INSERT INTO facility VALUES (25, '디비관', '디비관 4동', 'C', 'SILVER', '연습실', 'aishus6');
INSERT INTO facility VALUES (26, '예빛관', '예빛관 1동', 'A', 'SILVER', '연습실', 'dreamcompany1');
INSERT INTO facility VALUES (27, '예빛관', '예빛관 2동', 'A', 'SILVER', '연습실', 'dreamcompany2');
INSERT INTO facility VALUES (28, '예빛관', '예빛관 3동', 'B', 'SILVER', '연습실', 'dreamcompany3');
INSERT INTO facility VALUES (29, '예빛관', '예빛관 4동', 'B', 'SILVER', '연습실', 'dreamcompany1');
INSERT INTO facility VALUES (30, '예빛관', '예빛관 5동', 'C', 'SILVER', '연습실', 'dreamcompany2');

```

11) 음악강의 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE lecture (
    num          NUMBER(10) PRIMARY KEY,
    title        VARCHAR2(50) NOT NULL,
    contents     VARCHAR2(400) NOT NULL,
    tuition      NUMBER(8) DEFAULT 0,
    place        VARCHAR2(50) NOT NULL,
    on_off       VARCHAR2(6) DEFAULT '비대면',
    begin_day    DATE NOT NULL,
    end_day      DATE NOT NULL,
    capacity     NUMBER(2),
    day          CHAR(2) NOT NULL,
    start_time   DATE NOT NULL,
    end_time     DATE NOT NULL,
    musician_id  VARCHAR2(20) NOT NULL,
    released_time DATE,
    CONSTRAINT check_on_off CHECK(on_off IN ('대면', '비대면')),
    CONSTRAINT check_day CHECK(day IN ('월', '화', '수', '목', '금', '토')),
    CONSTRAINT fk_lecture_musician_id FOREIGN KEY(musician_id)
        REFERENCES musician_member(id) ON DELETE CASCADE
);

INSERT INTO lecture VALUES (1, '플룻으로 만드는 조수미', '플룻의 묘미를 가르쳐드립니다.', 12000,
    '서울특별시 영등포구 관악로 20-2번지', DEFAULT, TO_DATE('2023/02/01', 'YYYY-MM-DD'),
    TO_DATE('2023/02/28', 'YYYY-MM-DD'), 30, '화', TO_DATE('20:00:00', 'HH24:MI:SS'),
    TO_DATE('21:00:00', 'HH24:MI:SS'), 'mus_kwy', TO_DATE('2023/01/01', 'YYYY/MM/DD'));

INSERT INTO lecture VALUES (2, '관악구 모차르트', '피아노 체르니 300을 기가막히게 잘 가르쳐드립니다.',
    DEFAULT, '서울특별시 관악구', '대면', TO_DATE('2023/02/01', 'YYYY-MM-DD'),
    TO_DATE('2023/02/28', 'YYYY-MM-DD'), 30, '목', TO_DATE('19:00:00', 'HH24:MI:SS'),
    TO_DATE('21:00:00', 'HH24:MI:SS'), 'mus_kwy', TO_DATE('2023/01/01', 'YYYY/MM/DD'));

INSERT INTO lecture VALUES (3, '제스트 기타의 신! 심규재가 왔다!', '니들이 베이스를 알아?', 50000,
    '병천면 정글', DEFAULT, TO_DATE('2023/03/15', 'YYYY-MM-DD'), TO_DATE('2023/05/28', 'YYYY-MM-DD'),
    30, '월', TO_DATE('10:00:00', 'HH:MI:SS'), TO_DATE('14:00:00', 'HH24:MI:SS'), 'mus_ysj',
    TO_DATE('2023/02/01', 'YYYY/MM/DD'));

```

```

INSERT INTO lecture VALUES (4, '제스트 기타의 신! 심규재가 왔다!', '니들이 베이스를 알아?', 50000,
'병천면 정글', '대면', TO_DATE('2023/03/15','YYYY-MM-DD'), TO_DATE('2023/05/28','YYYY-MM-DD'), 30,
'화', TO_DATE('10:00:00', 'HH:MI:SS'), TO_DATE('14:00:00', 'HH24:MI:SS'), 'mus_ysj',
TO_DATE('2023/02/01','YYYY/MM/DD'));

INSERT INTO lecture values (5, '정겨운 알프스의 산골노래', '누구나 쉽게 따라부를 수 있는 알프스
동요를 가르쳐 드립니다. 남녀노소 누구나 지원 가능!', 12000, '경기도 부천시 소사동로 72번길 55-1',
'대면', TO_DATE('2023/03/21','YYYY-MM-DD'), TO_DATE('2023/05/22','YYYY-MM-DD'), 10, '금',
TO_DATE('08:00:00', 'HH:MI:SS'), TO_DATE('10:00:00', 'HH24:MI:SS'), 'mus_csy',
TO_DATE('2023/03/01','YYYY/MM/DD'));

INSERT INTO lecture VALUES (6, '힙합해서 돈벌자고', '난 가격도 싸게 안 해. 돈 벌고 싶으면 와서 내
가르침 귀에 때려 박어~', 300000, '서울특별시 영등포 타임스퀘어 지하 단칸방', '대면', TO_DATE('
2023/04/03','YYYY-MM-DD'), TO_DATE('2023/06/22','YYYY-MM-DD'), 5, '수', TO_DATE('23:00:00',
'HH24:MI:SS'), TO_DATE('23:59:00', 'HH24:MI:SS'), 'mus_chws', TO_DATE('2023/03/10','YYYY/MM/DD'));

INSERT INTO lecture VALUES (7, '클래식은 영원하다', '고전은 죽지않는다. 체르니 30', 100000,
'서울특별시 종로구 1번가 지하1층', DEFAULT, TO_DATE('2023/05/10','YYYY-MM-DD'),
TO_DATE('2023/05/30','YYYY-MM-DD'), 5, '수', TO_DATE('12:00:00', 'HH24:MI:SS'), TO_DATE('13:30:00',
'HH24:MI:SS'), 'mus_sgj', TO_DATE('2023/04/10', 'YYYY/MM/DD'));

```

12) 할인쿠폰 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE coupon (
    num          NUMBER(10) PRIMARY KEY,
    disc_rate    NUMBER(3) NOT NULL,
    expire_day   NUMBER(3) DEFAULT 30,
    lect_id      NUMBER(10),
    rel_time     DATE,
    CONSTRAINT fk_coupon_lect_id FOREIGN KEY(lect_id) REFERENCES lecture(num)
    ON DELETE CASCADE
);

INSERT INTO coupon(num, disc_rate, expire_day, lect_id) VALUES (1, 30, 14, 1);
INSERT INTO coupon(num, disc_rate, expire_day, lect_id) VALUES (2, 20, 21, 1);
INSERT INTO coupon(num, disc_rate, expire_day, lect_id) VALUES (3, 10, 30, 2);
INSERT INTO coupon(num, disc_rate, expire_day, lect_id) VALUES (4, 50, 60, 3);
INSERT INTO coupon(num, disc_rate, expire_day, lect_id) VALUES (5, 10, 15, 3);
INSERT INTO coupon(num, disc_rate, expire_day, lect_id) VALUES (6, 40, 5, 4);
INSERT INTO coupon(num, disc_rate, expire_day, lect_id) VALUES (7, 20, 7, 5);

```

13) 중고물품_카테고리 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE ug_category (
    code      NUMBER(1) PRIMARY KEY,
    category  VARCHAR2(30)
);

INSERT INTO ug_category VALUES (1, '관악기');
INSERT INTO ug_category VALUES (2, '타악기');
INSERT INTO ug_category VALUES (3, '국악기');
INSERT INTO ug_category VALUES (4, '현악기');
INSERT INTO ug_category VALUES (5, '기타물품');
```

14) 중고물품 테이블 생성 및 초기 데이터 삽입

```
CREATE SEQUENCE SEQ_usedgoods START WITH 1 CACHE 10;

CREATE TABLE usedgoods(
    num      NUMBER(10) PRIMARY KEY,
    name     VARCHAR2(50) NOT NULL,
    category NUMBER(1),
    sell_price NUMBER(10) NOT NULL,
    buy_price NUMBER(10) NOT NULL,
    condition VARCHAR2(100),
    views    NUMBER(10) DEFAULT 0
);

INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '가시나무로 만든 바이올린', 4, 450000, 800000, '새것처럼 깨끗함. 여전히 오래 쓸 수 있음.', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '손에 딱 맞는 리코더', 1, 200000, 400000, '보관하기 용이함. 최고의 상태고 개봉씰이 붙어있음. 원하시면 방문하셔서 보셔도 됩니다.', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '호른', 1, 300000, 500000, '호른 팔아요~', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '클라리넷', 1, 100000, 600000, '클라리넷 싸게 팔아요', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '플룻', 1, 200000, 400000, '플룻 비싼 거 싸게 팔아요', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '뿔가리', 2, 30000, 40000, '뿔가리 살래요? 직거래합니다.', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '목탁', 2, 35000, 70000, '목탁 팔아요. 스님 아닙니다.', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '북', 2, 40000, 80000, '사용 거의 안 했음', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '젬베', 2, 20000, 70000, '상태 양호', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '가야금', 3, 600000, 800000, '상태 최상', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '장', 3, 100000, 300000, '전문가용', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '거문고', 3, 150000, 300000, '거의 사용 안함', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '기타', 4, 50000, 300000, '일렉기타, 상태 양호', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '바이올린', 4, 100000, 200000, '초보자용',
```

```

DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '순목 바이올린', 4, 150000, 300000,
'어린이용', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '징맨이 치던 징', 2, 33000, 50000, '상태
양호', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '바이올린', 4, 150000, 500000, '흠집 약간
있음', DEFAULT);
INSERT INTO usedgoods VALUES (SEQ_usedgoods.NEXTVAL, '마을회관 장구', 2, 20000, 70000, '10년
사용, 상태 양호', DEFAULT);

```

15) 멤버십 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE membership (
    mbs_id          NUMBER(8) PRIMARY KEY,
    point           NUMBER(4) CHECK(point >= 0),
    regist_id       VARCHAR2(20),
    regist_time     DATE   DEFAULT SYSDATE NOT NULL,
    mbs_fee         NUMBER(6) NOT NULL CHECK(mbs_fee >= 1000),
    CONSTRAINT membership_FK FOREIGN KEY(regist_id)
        REFERENCES musician_member(id) ON DELETE CASCADE
);

INSERT INTO membership VALUES (1, 110, 'mus_kwy', TO_DATE('2023/05/19', 'YYYY/MM/DD'), 5000);
INSERT INTO membership VALUES (2, 150, 'mus_ysj', TO_DATE('2023/05/18', 'YYYY/MM/DD'), 10000);
INSERT INTO membership VALUES (3, 130, 'mus_ysj', TO_DATE('2023/04/14', 'YYYY/MM/DD'), 3000);
INSERT INTO membership VALUES (4, 160, 'mus_kwy', TO_DATE('2023/01/23', 'YYYY/MM/DD'), 5000);
INSERT INTO membership VALUES (5, 150, 'mus_csy', TO_DATE('2023/05/25', 'YYYY/MM/DD'), 8000);
INSERT INTO membership VALUES (6, 120, 'mus_chws', TO_DATE('2023/05/14', 'YYYY/MM/DD'), 3500);
INSERT INTO membership VALUES (7, 170, 'mus_chws', TO_DATE('2023/05/01', 'YYYY/MM/DD'), 6000);
INSERT INTO membership VALUES (8, 250, 'mus_csy', TO_DATE('2023/02/20', 'YYYY/MM/DD'), 9000);
INSERT INTO membership VALUES (9, 220, 'mus_kwy', TO_DATE('2023/05/19', 'YYYY/MM/DD'), 2000);
INSERT INTO membership VALUES (10, 160, 'mus_kwy', TO_DATE('2023/01/26', 'YYYY/MM/DD'), 8000);
INSERT INTO membership VALUES (12, 190, 'mus_sgj', TO_DATE('2023/05/16', 'YYYY/MM/DD'), 12000);
INSERT INTO membership VALUES (13, 220, 'mus_ysj', TO_DATE('2023/03/22', 'YYYY/MM/DD'), 3000);
INSERT INTO membership VALUES (14, 150, 'mus_sgj', TO_DATE('2023/05/09', 'YYYY/MM/DD'), 5500);
INSERT INTO membership VALUES (15, 250, 'mus_chws', TO_DATE('2023/01/19', 'YYYY/MM/DD'), 4000);
INSERT INTO membership VALUES (16, 140, 'mus_csy', TO_DATE('2023/05/17', 'YYYY/MM/DD'), 3500);
INSERT INTO membership VALUES (17, 210, 'mus_kwy', TO_DATE('2023/02/11', 'YYYY/MM/DD'), 5000);
INSERT INTO membership VALUES (18, 350, 'mus_chws', TO_DATE('2023/03/21', 'YYYY/MM/DD'), 12000);
INSERT INTO membership VALUES (19, 140, 'mus_ysj', TO_DATE('2023/04/03', 'YYYY/MM/DD'), 13000);
INSERT INTO membership VALUES (20, 170, 'mus_chws', TO_DATE('2023/02/07', 'YYYY/MM/DD'), 55000);
INSERT INTO membership VALUES (21, 250, 'mus_sgj', TO_DATE('2023/03/15', 'YYYY/MM/DD'), 6000);
INSERT INTO membership VALUES (22, 330, 'mus_ysj', TO_DATE('2023/02/21', 'YYYY/MM/DD'), 4500);
INSERT INTO membership VALUES (23, 210, 'mus_kwy', TO_DATE('2023/04/23', 'YYYY/MM/DD'), 3000);
INSERT INTO membership VALUES (24, 180, 'mus_sgj', TO_DATE('2023/01/28', 'YYYY/MM/DD'), 10000);
INSERT INTO membership VALUES (25, 240, 'mus_ysj', TO_DATE('2023/02/18', 'YYYY/MM/DD'), 3000);

```

16) 프로젝트 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE project(
    proj_id          NUMBER(8) PRIMARY KEY,
    title            VARCHAR2(40) NOT NULL,
    field            VARCHAR2(4) NOT NULL,
    detail           VARCHAR2(400) NOT NULL,
    target_amount    NUMBER(13) NOT NULL,
    pj_id           VARCHAR2(20),
    start_day        DATE,
    end_day          DATE, CHECK (end_day > start_day)
    CONSTRAINT project_fk FOREIGN KEY (pj_id) REFERENCES member(id)
        ON DELETE CASCADE,
    CONSTRAINT check_pj_field CHECK(field IN ('앨범', '곡', '공연')),
);

INSERT INTO project VALUES(20000001, '강태풍 미발매곡', '곡', '숨겨둔 그의 신곡 디비디비답이 돌아온다.', 500000,
'mus_ysj', TO_DATE('2023/05/29', 'YYYY/MM/DD'), TO_DATE('2023/06/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000002, '기타리스트 조민우 앨범', '앨범', '새로운 감각의 핑거스타일 연주', 100000,
'mus_kwy', TO_DATE('2023/05/29', 'YYYY/MM/DD'), TO_DATE('2023/06/30', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000003, '추억의 드라마 OST', '곡', '메타버스에서 시작되는 드라마의 어울리는 그
OST', 1000000, 'mus_chws', TO_DATE('2023/04/29', 'YYYY/MM/DD'), TO_DATE('2023/05/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000004, '피아니스트 최연수 3집', '앨범', '그녀의 콘서트 곡 모음', 1200000, 'mus_sgj',
TO_DATE('2023/04/29', 'YYYY/MM/DD'), TO_DATE('2023/05/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000005, '첼로니스트 박기범 1집', '앨범', '그의 콘서트 솔로 모음', 500000, 'mus_ysj',
TO_DATE('2023/04/29', 'YYYY/MM/DD'), TO_DATE('2023/05/29', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000006, 'ZESTIVAL 3회', '공연', '밴드동아리 ZEST 10번째 공연', 700000, 'kwy',
TO_DATE('2023/04/29', 'YYYY/MM/DD'), TO_DATE('2023/05/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000007, '소울메이트 5회 정기공연', '공연', '소울가득한 그들이 돌아온다.', 800000,
'kwy', TO_DATE('2023/06/20', 'YYYY/MM/DD'), TO_DATE('2023/07/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000008, '베이스리스트 솔로곡', '곡', '폭풍 베이스 솔로', 900000, 'mus_csy',
TO_DATE('2023/04/20', 'YYYY/MM/DD'), TO_DATE('2023/05/10', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000009, '강중구 정규 3집', '앨범', '다시 돌아온 그의 전설적인 노래 모음', 1000000,
'mus_csy', TO_DATE('2022/12/29', 'YYYY/MM/DD'), TO_DATE('2023/01/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000010, '조민우 단독콘서트', '공연', '그의 단독콘서트가 시작된다.', 2000000,
'mus_sgj', TO_DATE('2023/05/10', 'YYYY/MM/DD'), TO_DATE('2023/08/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000011, '트롯 X 발라드 콜라보', '공연', '환상적인 콜라보레이션', 1500000, 'sgj',
TO_DATE('2023/03/20', 'YYYY/MM/DD'), TO_DATE('2023/04/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000012, '로맨스 드라마 OST', '곡', '어쩔디비, 저쩔디비 어쩔저쩔입니다.', 1400000,
'mus_sgj', TO_DATE('2023/05/10', 'YYYY/MM/DD'), TO_DATE('2023/06/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000013, '쇼팽 모음집', '앨범', '다아아아비 모음입니다만', 1000000, 'mus_chws',
TO_DATE('2023/04/29', 'YYYY/MM/DD'), TO_DATE('2023/05/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000014, '아름다운 영화 BGM', '앨범', '영화 음악 추천합니다', 1200000, 'mus_kwy',
TO_DATE('2023/05/01', 'YYYY/MM/DD'), TO_DATE('2023/05/30', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000015, '돌아온 트롯', '곡', '트롯 열풍을 선도할 신인 발굴', 2000000, 'mus_kwy',
TO_DATE('2023/03/25', 'YYYY/MM/DD'), TO_DATE('2023/04/25', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000016, '트롯 REMIX', '곡', 'REMIX 뽕짝, 디스코 다 좋아요!', 1500000, 'mus_csy',
TO_DATE('2023/03/20', 'YYYY/MM/DD'), TO_DATE('2023/04/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000017, '강민수 REMIX 모음', '앨범', '환상적인 그의 목소리 다시듣기', 1000000,
'mus_sgj', TO_DATE('2023/05/20', 'YYYY/MM/DD'), TO_DATE('2023/06/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000018, '아이돌 Kwy 콘서트', '공연', '메타버스에서 듣는 그의 꿀보이스', 1100000,
'csy', TO_DATE('2023/04/20', 'YYYY/MM/DD'), TO_DATE('2023/05/20', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000019, '초청 가수 YS 초청 공연', '공연', '당신은 어디서 왔죠 이젠 사람 목소리가
아니예요!', 1200000, 'ysj', TO_DATE('2023/07/01', 'YYYY/MM/DD'), TO_DATE('2023/07/31', 'YYYY/MM/DD'));
INSERT INTO project VALUES(20000020, '다시 도약하자', '공연', '정신줄 놓고 다같이 놀아보자!', 1400000, 'mus_ysj',
TO_DATE('2023/06/01', 'YYYY/MM/DD'), TO_DATE('2023/06/14', 'YYYY/MM/DD'));

```

17) 수록하다 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE song_in_album (
    song_num    NUMBER(8),
    album_num   NUMBER(8),
    CONSTRAINT pk_song_in_album PRIMARY KEY(song_num, album_num),
    FOREIGN KEY(song_num) REFERENCES song(num) ON DELETE CASCADE,
    FOREIGN KEY(album_num) REFERENCES album(num) ON DELETE CASCADE
);

INSERT INTO song_in_album VALUES (1, 1);
INSERT INTO song_in_album VALUES (2, 1);
INSERT INTO song_in_album VALUES (3, 1);
INSERT INTO song_in_album VALUES (4, 1);
INSERT INTO song_in_album VALUES (5, 1);
INSERT INTO song_in_album VALUES (6, 2);
INSERT INTO song_in_album VALUES (7, 2);
INSERT INTO song_in_album VALUES (8, 2);
INSERT INTO song_in_album VALUES (9, 2);
INSERT INTO song_in_album VALUES (10, 2);
INSERT INTO song_in_album VALUES (11, 3);
INSERT INTO song_in_album VALUES (12, 3);
INSERT INTO song_in_album VALUES (13, 3);
INSERT INTO song_in_album VALUES (14, 3);
INSERT INTO song_in_album VALUES (15, 3);
INSERT INTO song_in_album VALUES (16, 4);
INSERT INTO song_in_album VALUES (17, 4);
INSERT INTO song_in_album VALUES (18, 4);
INSERT INTO song_in_album VALUES (19, 4);
INSERT INTO song_in_album VALUES (20, 4);
INSERT INTO song_in_album VALUES (21, 5);
INSERT INTO song_in_album VALUES (22, 5);
INSERT INTO song_in_album VALUES (23, 5);
INSERT INTO song_in_album VALUES (24, 5);
```

18) 예매하다 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE show_reserve(
    show_num    NUMBER(8),
    mem_id      VARCHAR2(20),
    reserve_date DATE DEFAULT SYSDATE,
    quantity    NUMBER(2) DEFAULT 1,
    CONSTRAINT pk_show_reserve PRIMARY KEY(show_num, mem_id),
    CONSTRAINT fk_reserve_show_num FOREIGN KEY(show_num)
        REFERENCES show(num) ON DELETE CASCADE,
    CONSTRAINT fk_reserve_mem_id FOREIGN KEY(mem_id)
        REFERENCES member(id) ON DELETE CASCADE
);

INSERT INTO show_reserve values(1, 'chws', TO_DATE('20230501','YYYYMMDD'), 2);
INSERT INTO show_reserve values(1, 'sgj', TO_DATE('20230504','YYYYMMDD'), 2);
INSERT INTO show_reserve values(2, 'mus_csy', TO_DATE('20230201','YYYYMMDD'), 4);
INSERT INTO show_reserve values(2, 'sgj', TO_DATE('20230112','YYYYMMDD'), 3);
INSERT INTO show_reserve values(3, 'mus_csy', TO_DATE('20230112','YYYYMMDD'), 2);
INSERT INTO show_reserve values(3, 'ysj', TO_DATE('20230120','YYYYMMDD'), 1);
```

```

INSERT INTO show_reserve values(3, 'kwy', TO_DATE('20230114','YYYYMMDD'), 2);
INSERT INTO show_reserve values(4, 'kwy', TO_DATE('20230112','YYYYMMDD'), 2);
INSERT INTO show_reserve values(4, 'chws', TO_DATE('20230120','YYYYMMDD'), 1);
INSERT INTO show_reserve values(5, 'ysj', TO_DATE('20230420','YYYYMMDD'), 4);
INSERT INTO show_reserve values(5, 'mus_chws', TO_DATE('20230421','YYYYMMDD'), 2);
INSERT INTO show_reserve values(6,'csy', TO_DATE('20230320','YYYYMMDD'), 1);
INSERT INTO show_reserve values(6,'kwy', TO_DATE('20230420','YYYYMMDD'), 2);
INSERT INTO show_reserve values(6,'mus_chws', TO_DATE('20230412','YYYYMMDD'), 4);
INSERT INTO show_reserve values(6,'sgj', TO_DATE('20230415','YYYYMMDD'), 3);

```

19) 참여_신청하다 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE show_apply(
    show_num    NUMBER(8),
    mem_id      VARCHAR2(20),
    appli_date  DATE  DEFAULT SYSDATE,
    role        VARCHAR2(40),
    approval    CHAR(1)  CHECK(approval IN ('Y', 'N')),
    CONSTRAINT pk_show_apply PRIMARY KEY(show_num, mem_id),
    CONSTRAINT fk_show_num FOREIGN KEY(show_num)
        REFERENCES show(num) ON DELETE CASCADE,
    CONSTRAINT fk_mem_id FOREIGN KEY(mem_id)
        REFERENCES member(id) ON DELETE CASCADE,
);

INSERT INTO show_apply values(1,'mus_kwy', TO_DATE('20220501','YYYYMMDD'),'보컬','Y');
INSERT INTO show_apply values(1,'mus_csy', TO_DATE('20220504','YYYYMMDD'),'보컬','Y');
INSERT INTO show_apply values(2,'mus_csy', TO_DATE('20220201','YYYYMMDD'),'보컬','Y');
INSERT INTO show_apply values(2,'chws', TO_DATE('20220112','YYYYMMDD'),'보컬','Y');
INSERT INTO show_apply values(3,'sgj', TO_DATE('20220112','YYYYMMDD'),'보컬','Y');
INSERT INTO show_apply values(3,'mus_sgj', TO_DATE('20220120','YYYYMMDD'),'코러스','N');
INSERT INTO show_apply values(3,'mus_ysj', TO_DATE('20220114','YYYYMMDD'),'코러스','N');
INSERT INTO show_apply values(4,'ysj', TO_DATE('20220112','YYYYMMDD'),'보컬','N');
INSERT INTO show_apply values(4,'sgj', TO_DATE('20220120','YYYYMMDD'),'코러스','N');
INSERT INTO show_apply values(5,'mus_kwy', TO_DATE('20220620','YYYYMMDD'),'피아노','N');
INSERT INTO show_apply values(5,'mus_csy', TO_DATE('20220621','YYYYMMDD'),'기타','N');
INSERT INTO show_apply values(6,'ysj', TO_DATE('20220920','YYYYMMDD'),'보컬','N');
INSERT INTO show_apply values(6,'mus_chws', TO_DATE('20220820','YYYYMMDD'),'드럼','N');
INSERT INTO show_apply values(6,'csy', TO_DATE('20220812','YYYYMMDD'),'기타','N');
INSERT INTO show_apply values(6,'kwy', TO_DATE('20220815','YYYYMMDD'),'베이스','N');

```

20) 관심_등록하다 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE show_like(
    show_num    NUMBER(8),
    mem_id      VARCHAR2(20),
    CONSTRAINT pk_show_like PRIMARY KEY(show_num, mem_id),
    CONSTRAINT fk_show_like_show_num FOREIGN KEY(show_num)
        REFERENCES show(num) ON DELETE CASCADE,
    CONSTRAINT fk_show_like_mem_id FOREIGN KEY(mem_id)
        REFERENCES member(id) ON DELETE CASCADE,
);

INSERT INTO show_like VALUES(1, 'chws');

```

```

INSERT INTO show_like VALUES(1, 'ysj');
INSERT INTO show_like VALUES(1, 'mus_csy');
INSERT INTO show_like VALUES(2, 'mus_ysj');
INSERT INTO show_like VALUES(2, 'chws');
INSERT INTO show_like VALUES(3, 'ysj');
INSERT INTO show_like VALUES(3, 'chws');
INSERT INTO show_like VALUES(4, 'mus_kwy');
INSERT INTO show_like VALUES(5, 'mus_kwy');
INSERT INTO show_like VALUES(6, 'mus_csy');
INSERT INTO show_like VALUES(6, 'mus_kwy');
INSERT INTO show_like VALUES(7, 'mus_csy');
INSERT INTO show_like VALUES(7, 'chws');
INSERT INTO show_like VALUES(7, 'ysj');
INSERT INTO show_like VALUES(8, 'mus_ysj');
INSERT INTO show_like VALUES(8, 'ysj');
INSERT INTO show_like VALUES(8, 'chws');
INSERT INTO show_like VALUES(9, 'mus_kwy');
INSERT INTO show_like VALUES(9, 'chws');
INSERT INTO show_like VALUES(10, 'chws');
INSERT INTO show_like VALUES(10, 'ysj');
INSERT INTO show_like VALUES(11, 'ysj');
INSERT INTO show_like VALUES(12, 'mus_kwy');
INSERT INTO show_like VALUES(12, 'mus_ysj');
INSERT INTO show_like VALUES(13, 'mus_kwy');
INSERT INTO show_like VALUES(14, 'mus_csy');
INSERT INTO show_like VALUES(15, 'ysj');
INSERT INTO show_like VALUES(16, 'mus_kwy');
INSERT INTO show_like VALUES(18, 'mus_kwy');
INSERT INTO show_like VALUES(19, 'mus_kwy');
INSERT INTO show_like VALUES(21, 'mus_kwy');

```

21) 대여_신청하다 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE rental_apply (
    mem_id    VARCHAR2(20),
    facil_id  NUMBER(8),
    rent_start DATE,
    approval  CHAR(1) DEFAULT 'N',
    rent_apply DATE DEFAULT SYSDATE,
    rent_end  DATE NOT NULL,
    user_num  NUMBER(3) NOT NULL,
    clean     CHAR(1) DEFAULT 'P',
    CONSTRAINT pk_rental_apply PRIMARY KEY(mem_id, facil_id, rent_start),
    CONSTRAINT fk_rental_apply_mem_id FOREIGN KEY(mem_id) REFERENCES member(id)
        ON DELETE CASCADE,
    CONSTRAINT fk_rental_apply_facil_id FOREIGN KEY(facil_id) REFERENCES facility(id)
        ON DELETE CASCADE,
    CONSTRAINT check_rent_start CHECK(rent_start >= rent_apply),
    CONSTRAINT check_approval_rent CHECK(approval IN('Y', 'N')),
    CONSTRAINT check_rent_end CHECK(rent_end > rent_start),
    CONSTRAINT check_clean CHECK(clean IN('P', 'D', 'E', 'F'))
);

INSERT INTO rental_apply VALUES ('kwy', 1, TO_DATE('2023-03-15','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-15','YYYY-MM-DD'), TO_DATE('2023-03-28','YYYY-MM-DD'), 35, 'P');

```

```

INSERT INTO rental_apply VALUES ('kwy', 1, TO_DATE('2023-03-16','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-16','YYYY-MM-DD'), TO_DATE('2023-03-24','YYYY-MM-DD'), 28, 'P');
INSERT INTO rental_apply VALUES ('mus_kwy', 2, TO_DATE('2023-03-20','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-19','YYYY-MM-DD'), TO_DATE('2023-03-28','YYYY-MM-DD'), 24, 'D');
INSERT INTO rental_apply VALUES ('mus_kwy', 2, TO_DATE('2023-03-21','YYYY-MM-DD'), 'N',
TO_DATE('2023-03-20','YYYY-MM-DD'), TO_DATE('2023-04-06','YYYY-MM-DD'), 30, 'P');
INSERT INTO rental_apply VALUES ('mus_ysj', 2, TO_DATE('2023-03-22','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-21','YYYY-MM-DD'), TO_DATE('2023-03-26','YYYY-MM-DD'), 28, 'P');
INSERT INTO rental_apply VALUES ('mus_ysj', 4, TO_DATE('2023-03-28','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-25','YYYY-MM-DD'), TO_DATE('2023-03-29','YYYY-MM-DD'), 60, 'E');
INSERT INTO rental_apply VALUES ('ysj', 5, TO_DATE('2023-03-25','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-24','YYYY-MM-DD'), TO_DATE('2023-04-03','YYYY-MM-DD'), 23, 'P');
INSERT INTO rental_apply VALUES ('ysj', 6, TO_DATE('2023-03-26','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-25','YYYY-MM-DD'), TO_DATE('2023-04-05','YYYY-MM-DD'), 27, 'F');
INSERT INTO rental_apply VALUES ('chws', 7, TO_DATE('2023-03-28','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-27','YYYY-MM-DD'), TO_DATE('2023-04-03','YYYY-MM-DD'), 5, 'P');
INSERT INTO rental_apply VALUES ('chws', 8, TO_DATE('2023-03-29','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-28','YYYY-MM-DD'), TO_DATE('2023-04-04','YYYY-MM-DD'), 7, 'P');
INSERT INTO rental_apply VALUES ('mus_chws', 9, TO_DATE('2023-03-28','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-28','YYYY-MM-DD'), TO_DATE('2023-04-06','YYYY-MM-DD'), 15, 'D');
INSERT INTO rental_apply VALUES ('mus_chws', 13, TO_DATE('2023-04-01','YYYY-MM-DD'), 'Y',
TO_DATE('2023-03-30','YYYY-MM-DD'), TO_DATE('2023-04-10','YYYY-MM-DD'), 10, 'P');
INSERT INTO rental_apply VALUES ('sgj', 15, TO_DATE('2023-04-05','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-01','YYYY-MM-DD'), TO_DATE('2023-04-13','YYYY-MM-DD'), 29, 'D');
INSERT INTO rental_apply VALUES ('sgj', 17, TO_DATE('2023-04-03','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-02','YYYY-MM-DD'), TO_DATE('2023-04-14','YYYY-MM-DD'), 22, 'E');
INSERT INTO rental_apply VALUES ('mus_sgj', 18, TO_DATE('2023-04-06','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-05','YYYY-MM-DD'), TO_DATE('2023-04-09','YYYY-MM-DD'), 8, 'D');
INSERT INTO rental_apply VALUES ('mus_sgj', 18, TO_DATE('2023-04-07','YYYY-MM-DD'), 'N',
TO_DATE('2023-04-05','YYYY-MM-DD'), TO_DATE('2023-04-16','YYYY-MM-DD'), 16, 'P');
INSERT INTO rental_apply VALUES ('csy', 18, TO_DATE('2023-04-10','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-07','YYYY-MM-DD'), TO_DATE('2023-04-15','YYYY-MM-DD'), 7, 'P');
INSERT INTO rental_apply VALUES ('csy', 19, TO_DATE('2023-04-09','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-08','YYYY-MM-DD'), TO_DATE('2023-04-19','YYYY-MM-DD'), 8, 'D');
INSERT INTO rental_apply VALUES ('mus_csy', 20, TO_DATE('2023-04-13','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-12','YYYY-MM-DD'), TO_DATE('2023-04-16','YYYY-MM-DD'), 27, 'D');
INSERT INTO rental_apply VALUES ('mus_csy', 22, TO_DATE('2023-04-15','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-12','YYYY-MM-DD'), TO_DATE('2023-04-20','YYYY-MM-DD'), 4, 'P');
INSERT INTO rental_apply VALUES ('kwy', 23, TO_DATE('2023-04-16','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-14','YYYY-MM-DD'), TO_DATE('2023-04-28','YYYY-MM-DD'), 19, 'E');
INSERT INTO rental_apply VALUES ('mus_kwy', 24, TO_DATE('2023-04-16','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-16','YYYY-MM-DD'), TO_DATE('2023-04-20','YYYY-MM-DD'), 5, 'F');
INSERT INTO rental_apply VALUES ('ysj', 24, TO_DATE('2023-04-19','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-17','YYYY-MM-DD'), TO_DATE('2023-04-27','YYYY-MM-DD'), 3, 'P');
INSERT INTO rental_apply VALUES ('mus_chws', 25, TO_DATE('2023-04-20','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-18','YYYY-MM-DD'), TO_DATE('2023-04-22','YYYY-MM-DD'), 9, 'E');
INSERT INTO rental_apply VALUES ('mus_sgj', 26, TO_DATE('2023-04-20','YYYY-MM-DD'), 'N',
TO_DATE('2023-04-18','YYYY-MM-DD'), TO_DATE('2023-04-21','YYYY-MM-DD'), 6, 'P');
INSERT INTO rental_apply VALUES ('csy', 27, TO_DATE('2023-04-23','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-20','YYYY-MM-DD'), TO_DATE('2023-04-27','YYYY-MM-DD'), 45, 'D');
INSERT INTO rental_apply VALUES ('mus_csy', 28, TO_DATE('2023-04-24','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-23','YYYY-MM-DD'), TO_DATE('2023-05-01','YYYY-MM-DD'), 24, 'D');
INSERT INTO rental_apply VALUES ('mus_sgj', 28, TO_DATE('2023-04-27','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-24','YYYY-MM-DD'), TO_DATE('2023-05-04','YYYY-MM-DD'), 21, 'P');
INSERT INTO rental_apply VALUES ('kwy', 29, TO_DATE('2023-04-28','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-25','YYYY-MM-DD'), TO_DATE('2023-05-03','YYYY-MM-DD'), 19, 'D');
INSERT INTO rental_apply VALUES ('mus_ysj', 30, TO_DATE('2023-04-27','YYYY-MM-DD'), 'Y',
TO_DATE('2023-04-26','YYYY-MM-DD'), TO_DATE('2023-05-06','YYYY-MM-DD'), 4, 'P');

```

22) 수강_신청하다 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE lecture_apply (
    mem_id    VARCHAR2(20),
    lect_num   NUMBER(10),
    apply_date DATE    DEFAULT SYSDATE,
    approval   CHAR(1) DEFAULT 'N' CHECK(approval IN ('Y', 'N')),
    CONSTRAINT pk_lecture_apply PRIMARY KEY(mem_id, lect_num, apply_date),
    CONSTRAINT fk_lecture_apply_mem_id FOREIGN KEY(mem_id)
        REFERENCES normal_member(id) ON DELETE CASCADE,
    CONSTRAINT fk_lecture_apply_lect_num FOREIGN KEY(lect_num)
        REFERENCES lecture(num) ON DELETE CASCADE,
);

INSERT INTO lecture_apply VALUES ('kwy', 2, DEFAULT, DEFAULT);
INSERT INTO lecture_apply VALUES ('kwy', 3, DEFAULT, DEFAULT);
INSERT INTO lecture_apply VALUES ('ysj', 3, DEFAULT, DEFAULT);
INSERT INTO lecture_apply VALUES ('ysj', 1, DEFAULT, 'Y');
INSERT INTO lecture_apply VALUES ('csy', 2, DEFAULT, DEFAULT);
INSERT INTO lecture_apply VALUES ('ysj', 2, DEFAULT, 'Y');
INSERT INTO lecture_apply VALUES ('chws', 4, DEFAULT, DEFAULT);
INSERT INTO lecture_apply VALUES ('sgj', 5, DEFAULT, DEFAULT);
INSERT INTO lecture_apply VALUES ('sgj', 4, DEFAULT, 'Y');
INSERT INTO lecture_apply VALUES ('sgj', 3, DEFAULT, DEFAULT);
```

23) 소유하다 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE get_coupon (
    mem_id    VARCHAR2(20),
    coupon_num NUMBER(10),
    get_date   DATE    DEFAULT SYSDATE,
    CONSTRAINT pk_get_coupon PRIMARY KEY(mem_id, coupon_num),
    CONSTRAINT fk_get_coupon_mem_id FOREIGN KEY(mem_id)
        REFERENCES normal_member(id) ON DELETE CASCADE,
    CONSTRAINT fk_get_coupon_coupon_num FOREIGN KEY(coupon_num)
        REFERENCES coupon(num) ON DELETE CASCADE,
);

INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('kwy', 2);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('sgj', 2);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('kwy', 3);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('chws', 3);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('kwy', 4);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('csy', 5);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('kwy', 1);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('ysj', 4);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('ysj', 5);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('sgj', 5);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('csy', 2);
INSERT INTO get_coupon(mem_id, coupon_num) VALUES ('kwy', 5);
```

24) 진행상태 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE sales_status (  
    code    NUMBER(1) PRIMARY KEY,  
    status  VARCHAR2(20)  
);  
  
INSERT INTO sales_status VALUES (1, '판매중');  
INSERT INTO sales_status VALUES (2, '예약중');  
INSERT INTO sales_status VALUES (3, '판매취소');  
INSERT INTO sales_status VALUES (4, '판매완료');
```

25) 판매하다 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE sell (  
    ugoods_num  NUMBER(10) PRIMARY KEY,  
    seller_id   VARCHAR2(20) NOT NULL,  
    sell_status NUMBER(1) DEFAULT 1,  
    reg_date    DATE DEFAULT SYSDATE,  
    delivery_fee NUMBER(5) DEFAULT 0 NOT NULL,  
    trans_type  VARCHAR2(10) DEFAULT '택배거래',  
    CONSTRAINT fk_sell_ugoods_id FOREIGN KEY(ugoods_num)  
        REFERENCES usedgoods(num) ON DELETE CASCADE,  
    CONSTRAINT fk_sell_seller_id FOREIGN KEY(seller_id)  
        REFERENCES member(id) ON DELETE CASCADE  
);  
  
INSERT INTO sell VALUES (1, 'kwy', DEFAULT, DEFAULT, DEFAULT, DEFAULT);  
INSERT INTO sell VALUES (2, 'ysj', DEFAULT, DEFAULT, 2500, DEFAULT);  
INSERT INTO sell VALUES (3, 'kwy', 2, DEFAULT, DEFAULT, '직거래');  
INSERT INTO sell VALUES (4, 'ysj', DEFAULT, DEFAULT, DEFAULT, DEFAULT);  
INSERT INTO sell VALUES (5, 'mus_kwy', DEFAULT, DEFAULT, 2500, DEFAULT);  
INSERT INTO sell VALUES (6, 'mus_ysj', 2, DEFAULT, 2500, '직거래');  
INSERT INTO sell VALUES (7, 'ysj', DEFAULT, DEFAULT, DEFAULT, DEFAULT);  
INSERT INTO sell VALUES (8, 'mus_kwy', 2, DEFAULT, 3000, '직거래');  
INSERT INTO sell VALUES (9, 'kwy', DEFAULT, DEFAULT, DEFAULT, DEFAULT);  
INSERT INTO sell VALUES (10, 'ysj', DEFAULT, DEFAULT, DEFAULT, DEFAULT);  
INSERT INTO sell VALUES (11, 'mus_ysj', DEFAULT, DEFAULT, DEFAULT, '직거래');  
INSERT INTO sell VALUES (12, 'mus_sgj', DEFAULT, DEFAULT, 2500, DEFAULT);  
INSERT INTO sell VALUES (13, 'mus_csy', 3, DEFAULT, 2500, '직거래');  
INSERT INTO sell VALUES (14, 'kwy', DEFAULT, DEFAULT, 3000, DEFAULT);  
INSERT INTO sell VALUES (15, 'mus_sgj', 3, DEFAULT, 2500, DEFAULT);  
INSERT INTO sell VALUES (16, 'mus_kwy', DEFAULT, DEFAULT, DEFAULT, DEFAULT);  
INSERT INTO sell VALUES (17, 'mus_csy', DEFAULT, DEFAULT, DEFAULT, DEFAULT);
```

26) 구매하다 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE buy (  
    ugoods_id  NUMBER(10) PRIMARY KEY,  
    buyer_id   VARCHAR2(20) NOT NULL,  
    buy_date   DATE DEFAULT SYSDATE,
```

```

ship_addr  VARCHAR2(100) NOT NULL,
confirm    CHAR(1)  DEFAULT 'N'  CHECK(confirm IN ('Y', 'N')),
CONSTRAINT fk_buy_ugoods_id FOREIGN KEY(ugoods_id)
            REFERENCES usedgoods(num) ON DELETE CASCADE,
CONSTRAINT fk_buy_buyer_id FOREIGN KEY(buyer_id)
            REFERENCES member(id) ON DELETE CASCADE,
);

INSERT INTO buy VALUES (1, 'mus_csy', DEFAULT, '서울특별시 종로1가', DEFAULT);
INSERT INTO buy VALUES (2, 'mus_ysj', DEFAULT, '울산광역시 종로1가', DEFAULT);
INSERT INTO buy VALUES (3, 'csy', DEFAULT, '광주광역시 종로1가', DEFAULT);
INSERT INTO buy VALUES (4, 'mus_ysj', DEFAULT, '광주광역시 종로1가', 'Y');
INSERT INTO buy VALUES (5, 'csy', DEFAULT, '서울특별시 종로1가', DEFAULT);
INSERT INTO buy VALUES (6, 'mus_csy', DEFAULT, '대전광역시 종로1가', DEFAULT);
INSERT INTO buy VALUES (7, 'mus_ysj', DEFAULT, '대전광역시 종로1가', DEFAULT);
INSERT INTO buy VALUES (8, 'ysj', DEFAULT, '대전광역시 종로1가', 'Y');
INSERT INTO buy VALUES (9, 'csy', DEFAULT, '서울특별시 종로1가', DEFAULT);
INSERT INTO buy VALUES (10, 'mus_csy', DEFAULT, '강원도 인제군 병사리', 'Y');
INSERT INTO buy VALUES (11, 'ysj', DEFAULT, '강원도 인제군 병사리', DEFAULT);

```

27) 찜하다 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE wishlist (
    mem_id    VARCHAR2(20),
    ugoods_id NUMBER(10),
    CONSTRAINT pk_wishlist PRIMARY KEY (mem_id, ugoods_id),
    CONSTRAINT fk_wishlist_mem_id FOREIGN KEY(mem_id) REFERENCES member(id),
    CONSTRAINT fk_wishlist_ugoods_id FOREIGN KEY(ugoods_id)
            REFERENCES usedgoods(num) ON DELETE CASCADE
);

INSERT INTO wishlist VALUES ('mus_ysj', 1);
INSERT INTO wishlist VALUES ('mus_sgj', 1);
INSERT INTO wishlist VALUES ('mus_kwy', 2);
INSERT INTO wishlist VALUES ('kwy', 2);
INSERT INTO wishlist VALUES ('mus_csy', 3);
INSERT INTO wishlist VALUES ('mus_sgj', 2);
INSERT INTO wishlist VALUES ('ysj', 1);
INSERT INTO wishlist VALUES ('csy', 3);
INSERT INTO wishlist VALUES ('mus_csy', 4);
INSERT INTO wishlist VALUES ('kwy', 4);
INSERT INTO wishlist VALUES ('mus_ysj', 4);
INSERT INTO wishlist VALUES ('mus_csy', 5);
INSERT INTO wishlist VALUES ('ysj', 5);
INSERT INTO wishlist VALUES ('mus_sgj', 5);
INSERT INTO wishlist VALUES ('ysj', 4);
INSERT INTO wishlist VALUES ('csy', 6);
INSERT INTO wishlist VALUES ('mus_kwy', 6);
INSERT INTO wishlist VALUES ('mus_ysj', 7);
INSERT INTO wishlist VALUES ('ysj', 8);
INSERT INTO wishlist VALUES ('mus_sgj', 8);
INSERT INTO wishlist VALUES ('mus_kwy', 7);

```

28) 평판점수_매기다 테이블 생성 및 초기 데이터 삽입

```
CREATE TABLE score_reputation (
    buyer_id    VARCHAR2(20),
    ugoods_id   NUMBER(10),
    eval_class  CHAR(6) NOT NULL,
    score       NUMBER(1) NOT NULL CHECK(score >= -5 AND score <= 5),
    reason      VARCHAR2(100),
    CONSTRAINT pk_score_reputation PRIMARY KEY(buyer_id, ugoods_id),
    CONSTRAINT fk_score_reputation_buyer_id FOREIGN KEY(buyer_id)
        REFERENCES member(id) ON DELETE CASCADE,
    CONSTRAINT fk_score_reputation_ugoods_id FOREIGN KEY(ugoods_id)
        REFERENCES usedgoods(num) ON DELETE CASCADE,
    CONSTRAINT check_eval_class CHECK(eval_class IN ('판매자', '구매자')),
);

INSERT INTO score_reputation VALUES ('kwy', 1, '판매자', 5, '친절하고 좋았음');
INSERT INTO score_reputation VALUES ('mus_csy', 5, '구매자', 4, NULL);
INSERT INTO score_reputation VALUES ('ysj', 1, '구매자', 5, '상냥하고 좋았음');
INSERT INTO score_reputation VALUES ('mus_kwy', 3, '판매자', 5, '친절함');
INSERT INTO score_reputation VALUES ('sgj', 6, '구매자', 4, 'GOOD');
INSERT INTO score_reputation VALUES ('ysj', 2, '구매자', 5, 'VERY GOOD');
INSERT INTO score_reputation VALUES ('mus_kwy', 15, '판매자', 4, '집앞까지 와주셨어요~ 친절! 근데
네고 요청하시더라고요..');
INSERT INTO score_reputation VALUES ('mus_csy', 6, '판매자', 0, '불친절함');
INSERT INTO score_reputation VALUES ('kwy', 2, '판매자', 0, '반말했어요');
INSERT INTO score_reputation VALUES ('mus_csy', 11, '판매자', 5, '매너 굿');
INSERT INTO score_reputation VALUES ('mus_chws', 4, '구매자', 3, '보통');
INSERT INTO score_reputation VALUES ('mus_csy', 15, '구매자', 3, '');
INSERT INTO score_reputation VALUES ('mus_sgj', 11, '판매자', 5, 'GOOD');
INSERT INTO score_reputation VALUES ('sgj', 8, '판매자', 4, '나쁘지 않았어요');
INSERT INTO score_reputation VALUES ('mus_kwy', 4, '판매자', 3, 'so so');
INSERT INTO score_reputation VALUES ('mus_chws', 10, '판매자', 4, '');
INSERT INTO score_reputation VALUES ('mus_ysj', 8, '판매자', 5, '멋있게 거래해서 좋아요');
INSERT INTO score_reputation VALUES ('kwy', 3, '구매자', 5, '깔끔하게 거래완료');
INSERT INTO score_reputation VALUES ('mus_ysj', 14, '판매자', 4, 'GOOD');
INSERT INTO score_reputation VALUES ('mus_csy', 10, '구매자', 0, '늦게 오시고도 당당함');
INSERT INTO score_reputation VALUES ('sgj', 14, '판매자', 2, '10분 늦게 옴');
INSERT INTO score_reputation VALUES ('mus_kwy', 5, '판매자', 5, '신속하게 거래 완료');
```

29) 후원하다 테이블 생성 및 초기 데이터 삽입

```
CREATE SEQUENCE SEQ_spon START WITH 1 CACHE 10;

CREATE TABLE sponsor (
    spon_num    NUMBER(8) PRIMARY KEY,
    norm_id     VARCHAR2(20),
    music_id    VARCHAR2(20),
    spon_time   DATE DEFAULT SYSDATE,
    spon_amount NUMBER(8) NOT NULL CHECK (spon_amount >= 1000),
    CONSTRAINT normal_FK FOREIGN KEY(norm_id) REFERENCES normal_member(id),
    CONSTRAINT music_FK FOREIGN KEY(music_id) REFERENCES musician_member(id),
);

INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'kwy', 'mus_sgj', TO_DATE('20230720','YYYYMMDD'),
```

```

20000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'kwy', 'mus_chws',
TO_DATE('20230725','YYYYMMDD'), 50000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'kwy', 'mus_ysj', TO_DATE('20230830','YYYYMMDD'),
50000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'ysj', 'mus_sgj', DEFAULT, 30000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'ysj', 'mus_chws', TO_DATE('20230822','YYYYMMDD'),
20000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'ysj', 'mus_kwy', TO_DATE('20230830','YYYYMMDD'),
25000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'chws', 'mus_chws', DEFAULT, 20000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'chws', 'mus_ysj', TO_DATE('20230815','YYYYMMDD'),
20000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'chws', 'mus_kwy',
TO_DATE('20230824','YYYYMMDD'), 50000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'sgj', 'mus_chws', DEFAULT, 50000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'sgj', 'mus_chws', TO_DATE('20230821','YYYYMMDD'),
10000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'sgj', 'mus_chws', TO_DATE('20230812','YYYYMMDD'),
30000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'csy', 'mus_sgj', DEFAULT, 40000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'csy', 'mus_chws', TO_DATE('20230813','YYYYMMDD'),
50000);
INSERT INTO sponsor VALUES(SEQ_spon.NEXTVAL, 'csy', 'mus_kwy', TO_DATE('20230821','YYYYMMDD'),
30000);

```

30) 편당하다 테이블 생성 및 초기 데이터 삽입

```

CREATE TABLE funding (
    proj_num    NUMBER(8),
    mem_id      VARCHAR2(20),
    fund_amount NUMBER(8) NOT NULL CHECK (fund_amount >= 10000),
    fund_time   DATE DEFAULT SYSDATE,
    CONSTRAINT funding_pk PRIMARY KEY (proj_num, mem_id),
    CONSTRAINT funding_proj_fk FOREIGN KEY (proj_num)
        REFERENCES project(proj_id) ON DELETE CASCADE,
    CONSTRAINT funding_mem_fk FOREIGN KEY (mem_id)
        REFERENCES member(id),
);

INSERT INTO funding VALUES(20000001, 'mus_chws', 10000, TO_DATE('2023/05/30 09:03:30', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000001, 'mus_csy', 30000, TO_DATE('2023/05/31 17:03:40', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000001, 'kwy', 40000, TO_DATE('2023/06/01 09:10:30', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000002, 'kwy', 50000, TO_DATE('2023/06/01 12:05:20', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000002, 'ysj', 40000, TO_DATE('2023/06/01 19:10:30', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000011, 'mus_chws', 50000, TO_DATE('2023/04/01 08:05:00', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000010, 'mus_chws', 40000, TO_DATE('2023/07/01 19:04:20', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000010, 'chws', 430000, TO_DATE('2023/07/02 10:03:30', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000014, 'mus_ysj', 30000, TO_DATE('2023/02/01 14:23:10', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000015, 'mus_kwy', 430000, TO_DATE('2023/04/01 11:20:30', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000015, 'mus_chws', 450000, TO_DATE('2023/04/01 15:30:10', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000020, 'mus_chws', 50000, TO_DATE('2023/05/01 11:10:30', 'YYYY/MM/DD HH24:MI:SS'));
INSERT INTO funding VALUES(20000020, 'kwy', 30000, TO_DATE('2023/05/04 20:30:40', 'YYYY/MM/DD HH24:MI:SS'));

```

31) 가입하다 테이블 생성 및 초기 데이터 삽입

```
CREATE SEQUENCE SEQ_join START WITH 1 CACHE 10;

CREATE TABLE join (
    join_num    NUMBER(8),
    join_id     VARCHAR2(20),
    mbs_num     NUMBER(8),
    join_time   DATE DEFAULT SYSDATE,
    CONSTRAINT join_join_id_fk FOREIGN KEY (join_id)
        REFERENCES normal_member(id),
    CONSTRAINT join_mbs_num_fk FOREIGN KEY (mbs_num)
        REFERENCES membership(mbs_num) ON DELETE CASCADE
);

INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'kwy', 23, TO_DATE('2023/05/22', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'kwy', 15, TO_DATE('2023/03/10', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'kwy', 6, TO_DATE('2023/01/21', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'ysj', 3, TO_DATE('2023/04/28', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'ysj', 2, TO_DATE('2023/03/23', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'ysj', 1, TO_DATE('2023/01/12', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'chws', 9, TO_DATE('2023/02/22', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'chws', 17, TO_DATE('2023/05/13', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'chws', 16, TO_DATE('2023/02/13', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'sgj', 14, TO_DATE('2023/04/24', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'sgj', 3, TO_DATE('2023/01/04', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'sgj', 12, TO_DATE('2023/03/11', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'csy', 16, TO_DATE('2023/02/12', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'csy', 15, TO_DATE('2023/04/20', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'csy', 7, TO_DATE('2023/02/21', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'kwy', 9, TO_DATE('2023/04/26', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'kwy', 22, TO_DATE('2023/02/21', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'ysj', 17, TO_DATE('2023/01/22', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'ysj', 4, TO_DATE('2023/04/28', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'ysj', 13, TO_DATE('2023/02/10', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'chws', 1, TO_DATE('2023/01/03', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'sgj', 4, TO_DATE('2023/03/16', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'csy', 25, TO_DATE('2023/02/22', 'YYYY/MM/DD'));
INSERT INTO join VALUES(SEQ_join.NEXTVAL, 'csy', 22, TO_DATE('2023/04/13', 'YYYY/MM/DD'));
```

(2) 인덱스 및 시퀀스 생성

1) 인덱스 생성

```
CREATE INDEX company_idx ON company_member(comp_name);
CREATE INDEX genre_idx ON show(genre);
CREATE INDEX uername_idx ON usedgoods(name);
```

2) 시퀀스 생성

시퀀스는 반드시 해당 시퀀스를 사용하는 테이블을 생성하기 전에 생성해야 한다.

```
// 중고물품 시퀀스 생성
CREATE SEQUENCE SEQ_usedgoods START WITH 1 CACHE 10;

// 후원하다 시퀀스 생성
CREATE SEQUENCE SEQ_spon START WITH 1 CACHE 10;

// 가입하다 시퀀스 생성
CREATE SEQUENCE SEQ_join START WITH 1 CACHE 10;
```

(3) DB 테스트용 SQL문

1) 삽입 테스트용 SQL문

새로운 중고물품을 삽입하는 방법에는 두 가지가 있는데, 그중 하나는 다음과 같이 각 속성의 값을 직접 삽입하는 것이다. 이때 중고물품번호를 자동으로 삽입하기 위해 'SEQ_usedgoods'라는 시퀀스를 생성했으므로 다음에 사용 가능한 시퀀스 값을 생성하는 의사열(pseudo column)인 NEXTVAL을 활용해서 다음과 같이 삽입할 수 있다.

```
INSERT INTO usedgoods
VALUES (SEQ_usedgoods.NEXTVAL, '바이올린', 4, 500000, 800000, '상태 매우 양호함', DEFAULT);
```

또 다른 방법은 &대체변수를 사용해서 다음과 같이 사용자가 직접 입력한 값을 삽입하는 것이다.

```
INSERT INTO usedgoods
VALUES(SEQ_usedgoods.NEXTVAL, '&물품명', &카테고리, &판매가격, &구매당시가격,
      '&물품상태설명', DEFAULT);
```

2) 삭제 테스트용 SQL문

기존 프로젝트 정보를 삭제하는 방법 역시 두 가지가 있는데, 그중 하나는 다음과 같이 삭제할 프로젝트_id를 직접 지정해서 삭제하는 것이다.

```
DELETE FROM project
WHERE proj_id = 20000001;
```

또 다른 방법은 다음과 같이 &대체변수를 사용해서 사용자가 입력한 프로젝트_id를 갖는 데이터를 삭제하는 것이다.

```
DELETE FROM project
WHERE proj_id = &프로젝트_id;
```

3) 갱신 테스트용 SQL문

기존 할인쿠폰 정보를 갱신하는 방법도 두 가지로 구분할 수 있는데, 그중 하나는 다음과 같이 번호(num)를 직접 지정해서 유효기간을 변경하는 것이다.

```
UPDATE coupon
SET expire_day = 28
WHERE num = 1;
```

또 다른 방법은 다음과 같이 &대체변수를 사용해서 갱신할 속성의 값을 사용자가 입력한 값으로 변경하는 것이다.

```
UPDATE coupon
SET expire_day = &유효기간
WHERE num = 1;
```

4) 검색 테스트용 SQL문

① 등급이 VIP인 일반회원 가운데 활동점수가 50점 이상인 회원의 이름과 휴대폰번호, 포인트, 그리고 활동점수를 검색한다.

```
SELECT name 회원명, cellphone 휴대폰번호, point 포인트, activity_score 활동점수
FROM member, normal_member
WHERE class = 'VIP'
AND activity_score >= 50
AND member.id = normal_member.id;
```

② 사용자가 입력한 제목의 앨범에 수록된 모든 곡의 제목과 작곡자 이름, 그리고 누적조회수를 검색한다.

```
SELECT song_title 제목, name 작곡자, song_acc_views 누적조회수
FROM album, song, song_in_album, member
WHERE alb_title = '&앨범제목'
AND album.num = song_in_album.album_num
AND song_in_album.song_num = song.num
AND song.mus_id = member.id;
```

- ③ 음악가회원의 이름과 누적후원금액을 누적후원금액의 내림차순으로 검색한다.

```
SELECT  M.name 음악가회원, S.sum_fee 총후원금액
FROM    (SELECT music_id, SUM(spon_amount) sum_fee FROM sponsor GROUP BY music_id) S
        INNER JOIN member M ON M.id = S.music_id
ORDER BY S.sum_fee DESC;
```

- ④ 펀딩된 프로젝트에 대해 각 프로젝트의 제목과 기획자 이름, 누적된 펀딩금액을 누적된 펀딩금액의 내림차순으로 검색한다.

```
SELECT  P.title 제목, M.name 기획자, F.sum_fund 펀딩금액
FROM    (SELECT proj_num, SUM(fund_amount) sum_fund
        FROM funding GROUP BY proj_num) F
        INNER JOIN project P ON P.proj_id = F.proj_num
        INNER JOIN member M ON P.pj_id = M.id
ORDER BY F.sum_fund DESC;
```

- ⑤ 2023년 9월 1일 이후에 개최되는 공연의 제목과 시작 날짜, 기획자 이름, 장르를 시작 날짜의 오름차순으로 검색한다.

```
SELECT  S.title 제목, start_date 날짜, M.name 기획자, G.category 장르
FROM    show S, member M, genre G
WHERE   start_date >= TO_DATE('2023/09/01', 'YYYY/MM/DD')
AND     S.hold_mem_id = M.id
AND     S.genre = G.code
ORDER BY start_date;
```

- ⑥ 중고물품 가운데 국악기의 이름과 판매가격을 판매가격이 저렴한 것부터 검색한다.

```
SELECT  name 악기이름, sell_price 판매가격
FROM    usedgoods, ug_category
WHERE   ug_category.category = '국악기'
AND     code = usedgoods.category
ORDER BY sell_price;
```

(4) 테이블 및 시퀀스 삭제

```
ALTER TABLE member
    DROP PRIMARY KEY CASCADE;
DROP TABLE member;
ALTER TABLE normal_member
    DROP PRIMARY KEY CASCADE;
DROP TABLE normal_member;
ALTER TABLE musician_member
    DROP PRIMARY KEY CASCADE;
DROP TABLE musician_member;
ALTER TABLE company_member
    DROP PRIMARY KEY CASCADE;
DROP TABLE company_member;
DROP TABLE connect_record;
ALTER TABLE album
    DROP PRIMARY KEY CASCADE;
DROP TABLE album;
DROP TABLE genre;
ALTER TABLE song
    DROP PRIMARY KEY CASCADE;
DROP TABLE song;
ALTER TABLE show
    DROP PRIMARY KEY CASCADE;
DROP TABLE show;
ALTER TABLE facility
    DROP PRIMARY KEY CASCADE;
DROP TABLE facility;
ALTER TABLE lecture
    DROP PRIMARY KEY CASCADE;
DROP TABLE lecture;
ALTER TABLE coupon
    DROP PRIMARY KEY CASCADE;
DROP TABLE coupon;
DROP TABLE ug_category;
ALTER TABLE usedgoods
    DROP PRIMARY KEY CASCADE;
DROP TABLE usedgoods;
ALTER TABLE membership
    DROP PRIMARY KEY CASCADE;
DROP TABLE membership;
ALTER TABLE project
    DROP PRIMARY KEY CASCADE;
DROP TABLE project;
DROP TABLE song_in_album;
DROP TABLE show_reserve;
DROP TABLE show_apply;
DROP TABLE show_like;
DROP TABLE rental_apply;
DROP TABLE lecture_apply;
DROP TABLE get_coupon;
DROP TABLE sales_status;
DROP TABLE sell;
DROP TABLE buy;
DROP TABLE wishlist;
DROP TABLE score_reputation;
```

```
DROP TABLE sponsor;  
DROP TABLE funding;  
DROP TABLE join;  
  
DROP SEQUENCE SEQ_usedgoods;  
DROP SEQUENCE SEQ_spon;  
DROP SEQUENCE SEQ_join;  
  
PURGE RECYCLEBIN;
```

7. 별첨

[목차로](#)

[별첨 1] 운영책임자와 인터뷰 내용 요약

No.	질문내용	응답 내용
1	회원 등급은 일반회원들만 가질 수 있나요?	아닙니다. 회원의 등급은 회원의 활동 점수에 기초하여 음악가 회원을 포함한 모든 회원이 가질 수 있습니다.
2	회원 등급은 몇 개로 나눌 수 있을까요?	회원 등급은 다섯 단계로 나눌 수 있습니다.
3	활동점수와 평판점수의 차이는 무엇인가요?	활동점수는 강의 참여, 강의 등록, 공연 등록, 공연 참여, 펀딩 기획, 펀딩 후원, 지정 후원, 멤버십 가입, 중고 거래를 통해, 평판점수는 중고 거래를 통해 갱신되는 점수입니다.
4	그렇다면 활동점수와 평판점수 사이에 어떤 상관관계가 존재하나요?	평판점수가 활동점수에 영향을 주는 관계입니다. 예를 들어, 중고 거래를 통해 평판점수가 0보다 작아지면 떨어진 점수만큼 활동점수가 작아지는 식으로 구현되길 바랍니다.
5	음악가 회원의 작품활동은 크게 앨범과 공연으로 구분할 수 있을까요?	네, 앨범에 수록되는 곡 작업도 포함됩니다.
6	음악가 회원의 작품활동을 어떻게 홍보하길 원하시나요?	월로별 조회수가 높은 순으로 작품활동을 나열하는 방식으로 음악가를 홍보했으면 좋겠습니다.
7	음악가 회원만 할 수 있는 활동이 있을까요?	강의 등록, 작품활동, 펀딩 등록 등의 전문성이 필요한 활동들만 가능하고, 그 외의 것들은 일반회원과 같으면 좋겠습니다.
8	음악강의를 할인할 수 있는 수단이 할인쿠폰과 포인트만 있는 것이 맞나요?	네, 그렇습니다.
9	음악강의의 수강 대상은 일반회원에 한정되나요?	네, 그렇습니다. 음악강의의 수강 대상은 일반회원만 가능합니다.
10	음악강의를 등록한 음악가는 모든 강의의 할인쿠폰을 발행할 수 있나요?	아닙니다. 음악가는 본인의 음악강의에만 할인쿠폰을 발행할 수 있습니다.

[별첨 2] 시설물 대여관리자와 인터뷰 내용 요약

[목차로](#)

No.	질문내용	응답 내용
1	시설물 대여를 위해 지역의 검색 단위를 어디까지 구분할까요?	시/군/구/동까지로 생각하고 있습니다.
2	회원에게 어떤 시설물을 대여하나요?	공연장과 연습실입니다.
3.	회원이 같은 시설물을 하루에 여러 번 대여할 수 있을까요?	아닙니다. 한 회원은 하루에 최대 1개의 시설물만 대여할 수 있습니다.
4	회원들이 공연장 혹은 연습장을 빌릴 때 대여비를 내야 하는 경우가 발생할 수도 있을까요?	그렇지 않습니다. 음악가 후원의 취지와 타 플랫폼과의 차별점을 고려해 시설물 대여는 무료로 생각하고 있습니다.
5	시설물을 무료로 대여해준다고 했는데 시설물이 파손될 경우, 대처 방법이 있나요?	활동점수와 등급을 통해 가능합니다. 활동점수가 차감되어 회원 등급이 낮아지면 시설물을 대여할 수 없도록 하길 원합니다.
6	회원의 시설물 대여 내역은 얼마동안 보관해야 하나요?	최소 5년 정도는 보관하면 좋겠습니다.
7	시설물마다 이미 예약이 찬 시간대의 정보를 알 수 있어야 할까요?	네, 그렇습니다.
8	업체와의 계약을 통해 예약 가능한 시설물이지만, 보수공사/재해 등의 돌발상황 발생 시 표현할 방법이 필요할까요?	필요합니다. 시설물 대여가 불가능하다는 정보를 알려줄 수 있으면 좋겠습니다.
9	시설물들의 등급이 따로 있을까요?	시설물의 크기에 따라 세 단계로 구분할 생각입니다.
10	시설물들의 상태는 어떤 기준으로는 판단하실 예정인가요?	청결도, 시설물 크기, 방음수준 등의 정보로 판단할 예정입니다.

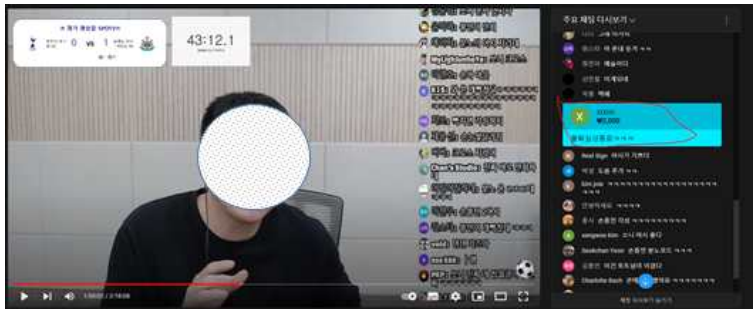
[별첨 3] 후원 기능 관련 플랫폼 조사 및 분석

[목차로](#)

- 조사 플랫폼 :유튜브(<https://www.youtube.com/>)

1) 후원

유튜브 크리에이터가 실시간 스트리밍을 할 때, 사용자가 원하는 금액만큼 채널에 후원할 수 있다.



2) 멤버십

다양한 등급의 멤버십에 가입하여 비용을 지급하고, 자신의 정보를 변경할 수 있고, 닉네임을 노출해 줌으로써 홍보 효과를 올릴 수 있다.



[별첨 4] 시설물 대여 관련 플랫폼 조사 및 분석

목차로

- 조사 플랫폼 : SpaceCloud(<https://www.spacecloud.kr/>)

1) 공간 대여

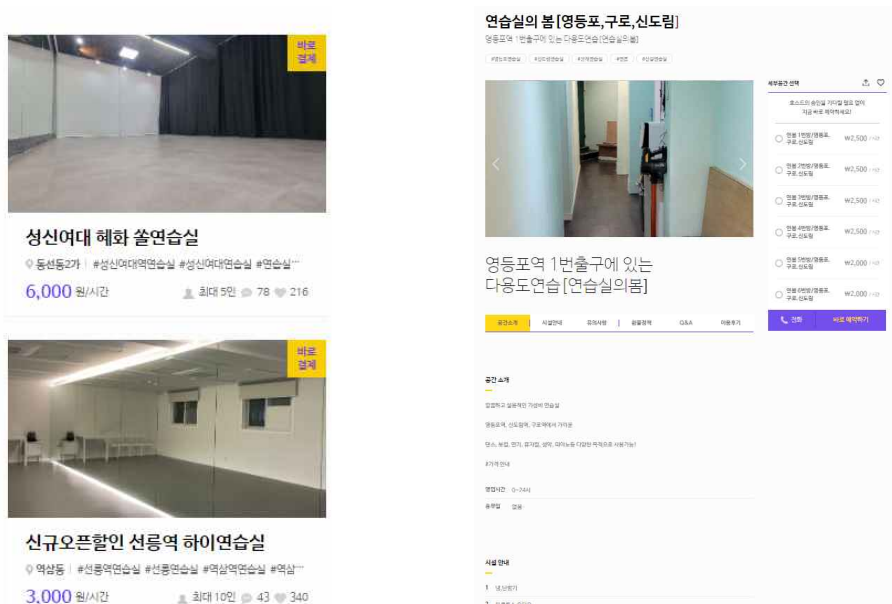
원하는 공간 유형을 선택하여 사용자가 필요한 유형의 장소를 대여할 수 있다.

어떤 공간을 찾고 있나요?



2) 공간 조회

원하는 유형을 고르면, 장소의 이름, 이용 정보, 가격, 주소 정보를 확인할 수 있다.



[별첨 5] 크라우드 펀딩 플랫폼 조사 및 분석

[목차로](#)

- 조사 플랫폼 : tumbbug(<https://tumbbug.com/>)

1) 프로젝트 등록

- 분야를 선택하고 해당 분야의 프로젝트를 진행하여 펀딩을 등록할 수 있다.
- 간단한 소개 글을 작성하여 어떤 프로젝트를 진행할지 등록할 수 있다.

멋진 아이디어가 있으시군요!
어떤 프로젝트를 계획 중이신가요?

나중에 변경 가능하니 너무 걱정하세요.

보드게임 · TRPG

디지털 게임

웹툰 · 만화

웹툰 리소스

디자인 용구

캐릭터 · 코스

종 · 라벨

대크 · 개천

만화종물

무드

장수 · 책지

먹물

잡화

주얼리

출판

디자인

예술

사진

음악

영화 · 비디오

공백

프로젝트를 간단하게 소개해주세요.

나중에 수정 가능하니 편하게 적어주세요.

프로젝트 요약글 입력해주세요.

최소 10자 이상 입력해주세요.

0/50

2) 프로젝트 펀딩

- 프로젝트에 원하는 금액을 후원하고 선물을 전달받을 수 있다.

✓ 0명이 선택

50개 선물

41,000원 +

[얼리버드] 7차 혼자서 매혹하기 <배송비 모함>

· 아홉번 매혹되는 황기2 (Nine Tailed Fox 2) perfume [50ml] (x 1)

✓ 4명이 선택

196개 선물

42,000원 +

혼자서 매혹하기 <배송비모함>

· 아홉번 매혹되는 황기2 (Nine Tailed Fox 2) perfume [50ml] (x 1)

▶ 아홉번 매혹되는 황기2 [Nine Tailed Fox 2]

A. 혼자서 매혹하기 (슈퍼얼리버드) - 36,000 < 한정수량 60개 > 조기 마감

B. 혼자서 매혹하기 (얼리버드) - 38,000 < 한정수량 100개 > 조기 마감

B-1. 혼자서 매혹하기 (얼리버드 2차) - 38,500 < 한정수량 50개 추가 : >

C. 혼자서 매혹하기 - 49,000 → 42,000

D. 둘이서 매혹하기 (슈퍼얼리버드) - 69,000 < 한정수량 60개 >

E. 둘이서 매혹하기 (얼리버드) - 72,000 < 한정수량 60개 >

F. 둘이서 매혹하기 - 98,000 → 82,000

3) 분야/상태별 프로젝트 조회

- 분야별로 등록된 프로젝트를 조회할 수 있다.
- 프로젝트 상태 : 공개 예정, 진행 중, 종료
- 조회 수, 후원금액, 남은 기간 등의 조건으로 펀딩 프로젝트를 조회할 수 있다.

The screenshot shows a web interface for searching projects. At the top, there are various category icons like '문자', '동영상', '책', etc. Below these, there's a '공연' (Performance) section with filters for '상태' (Status) and '에디터 추천' (Editor's Choice). A list of projects is displayed, each with a thumbnail, title, and funding progress bar. On the right, there's a sidebar with '인기순' (Popularity) and '최신순' (Latest) sorting options, and a list of project names like '최다 후원순', '최다 금액순', and '마감 임박순'.

4) 응원권(할인 쿠폰) 발행

- 본인이 등록한 프로젝트의 응원권을 발행하여, 프로젝트의 펀딩 비용 할인을 적용하고, 이를 통해 펀딩을 장려할 수 있다.

The screenshot shows the '응원권 발급' (Supporter's Right Issuance) interface. It includes a section for '응원권 발행' (Supporter's Right Issuance) with a description of how it works. Below that, there's a '응원권 제목' (Supporter's Right Title) field. On the right, there's a notification box that says '사용 가능한 응원권이 있습니다.' (You have supporter's rights that can be used.) and '3,000원 응원권을 사용하여 이 프로젝트를 후원해 보세요! 결제 단계에서 사용할 수 있습니다.' (Try supporting this project with a 3,000 won supporter's right! You can use it at the payment stage.)

[별첨 6] 중고 거래 플랫폼 조사 및 분석

[목차로](#)

- 조사 플랫폼 : 당근마켓(<https://www.daangn.com/>)

1) 중고물품 등록

자신이 팔고자 하는 상품의 사진, 설명, 주소를 등록하여 거래를 진행한다.

중고거래



기타 책 통기타 책 독학 꿀템...
서울 강서구 동촌동
8,000원



기타 7단 거치대 (통기타/알...
충남 천안시 서북구 불당동
40,000원



기타가방+클래식기타+발판...
경기도 김포시 구래동
130,000원

2) 주변 지역 조회

- 내 주변 지역에 등록된 중고 거래를 조회할 수 있다.
- 내 주변 지역 동네 정보를 통해 주변 지역 사람들과 소통할 수 있다.

동네정보



요며칠 원성천 벚꽃으로 사람도 장사꾼도 많아요 저도 이제 나이가먹긴했나봐요
꽃이 예쁜거보니..
충남 천안시 동남구 원성2동

2



천안천 벚꽃 개나리 만개했어요 백석동에서 살다 이사왔는데 너무좋아요 공기도
좋고,조용하고, 걷기운동하기도 좋아요.
충남 천안시 동남구 용곡동

3

3) 회원 정보 조회

- 회원별 이름, 사진, 거래 목록을 조회할 수 있다.
- 정상적인 거래를 진행하면 매너 온도를 증가시켜 신뢰도를 표현할 수 있다.
- 거래 후기 등을 등록할 수 있다.



진우 천안시 서북구 불당동
매너온도 31.9°C 재거래희망률 100%

[별첨 7] 온라인 강의 플랫폼 조사 및 분석

[목차로](#)

- 조사 플랫폼 : 레슨올(<https://www.lessonall.com/renewal>)

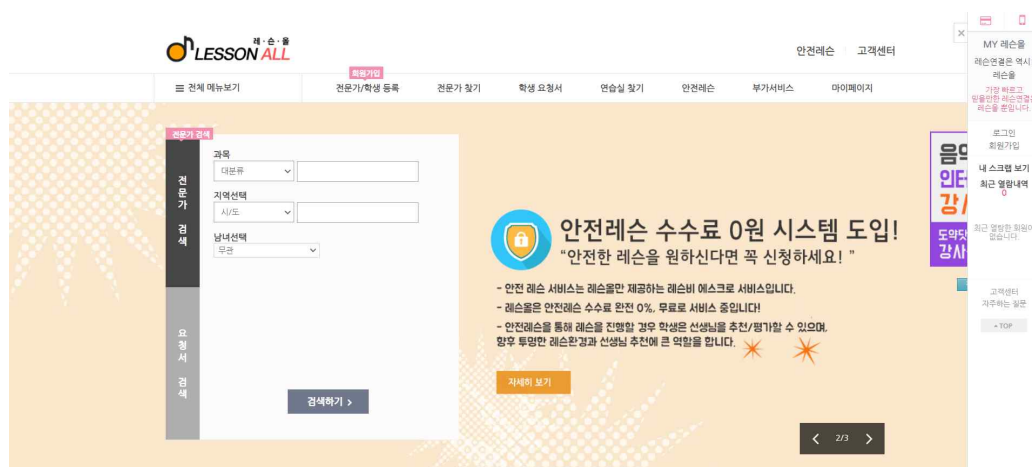
1) 일반회원의 레슨 요청 기능

- 회원이 레슨 희망 과목을 작성하면 해당 능력을 지닌 전문가가 요청서를 보고 레슨을 진행할 수 있고, 오늘의 요청서를 확인할 수 있다.



2) 전문가 검색 기능

- 회원이 배우고 싶은 강의와 거주지, 원하는 성별 등의 정보를 검색하여 원하는 전문가를 선택할 수 있다.



3) 강의실 제공 기능

- 레슨을 원활하게 진행할 수 있도록 전국적으로 655개의 연습실 정보를 연습실 종류, 주차 여부, 연습실 수, 인근 지하철역, 지역 정보 등과 함께 제공한다.

The screenshot displays a web application for finding music practice rooms. The main content area is titled '전체 연습실' (All Practice Rooms) and shows a list of 655 rooms. The list is organized into columns: '사진' (Photo), '연습실 명' (Practice Room Name), '종류' (Type), '주차여부' (Parking), '연습실 수' (Number of Rooms), '지하철 역' (Subway Station), and '지역' (Region).

사진	연습실 명	종류	주차여부	연습실 수	지하철 역	지역
	예술실용음악학원	음악종합	주차가능	8개	[서울지하철] 홍대입구역	경기 부천시
	한국생활음악협회계통지부	음악종합	주차가능	11개	[대전지하철] 대전역	충남 계룡시
	24시 방음 부스 연습실 임대 (서대문구 영지전문대)	음악종합	주차불가	9개	[서울지하철] 새결역	서울 영등포구

The sidebar on the left provides filters for '지역선택' (Region Selection) and '지하철 노선별 검색' (Search by Subway Line). The right-hand panel includes a 'MY 레슨을' (My Lesson) section, a '로그인' (Login) button, and a '내 스크린 보기' (View My Screen) button.