

.....

2017.06.07

.....

데이터 베이스 모델링 프로젝트

최종혁

씨쥬브2조

이형호

함태훈



서희정

문세진

이창범





1. 프로젝트 개요

● 소개



제목

CGV 웹사이트 데이터 베이스 모델링



선정 웹사이트

<http://www.cgvl.co.kr/>

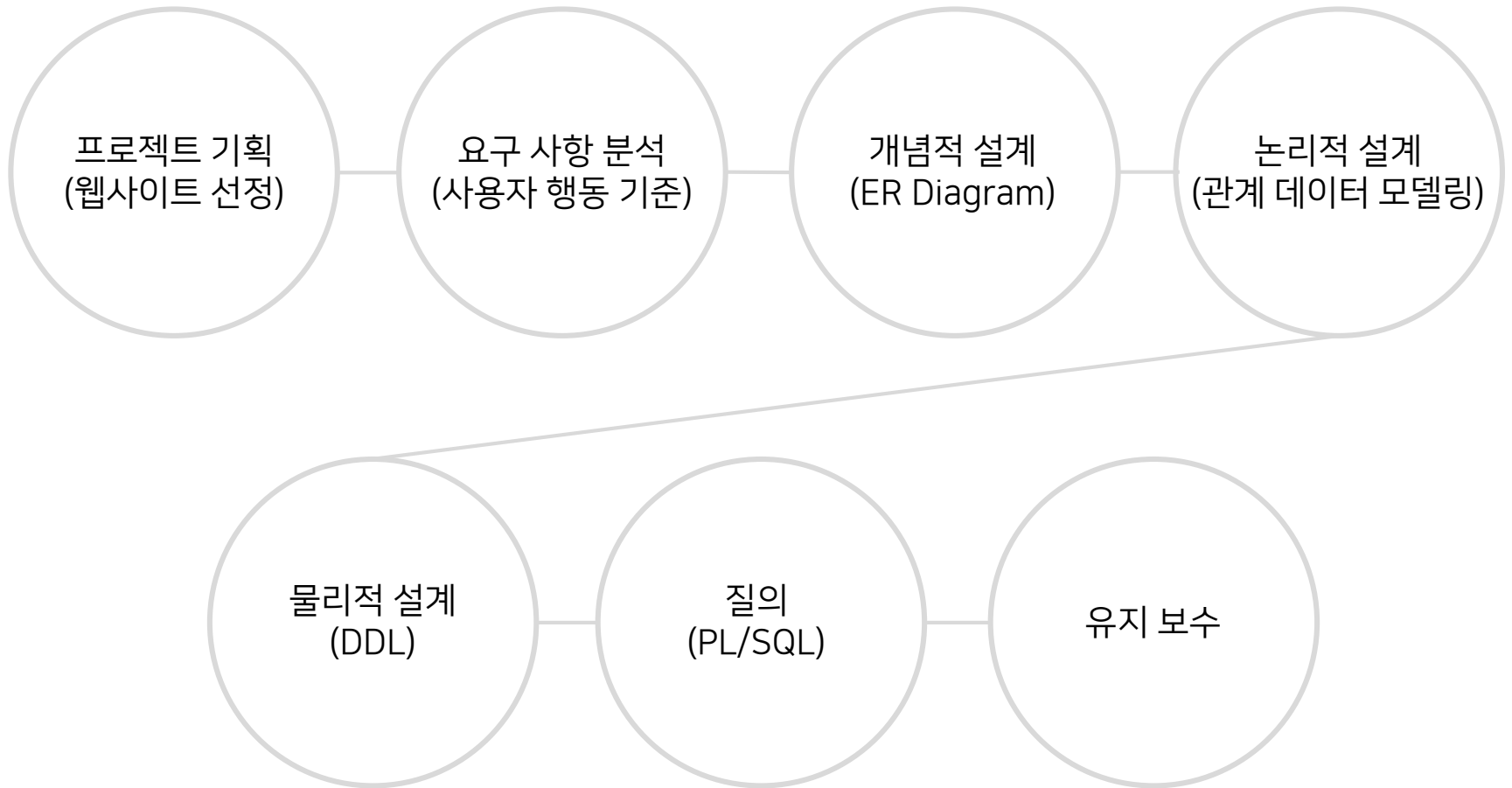


목적

영화를 보는 고객들의 행동 패턴에 맞추어 그에 상응하는 서비스와 콘텐츠를 제공하기 위해 프로젝트 진행

1. 프로젝트 개요

● 구축과정



1. 프로젝트 개요

● 웹사이트 소개



영화 예매뿐만 아니라 차트, 영화 정보, 극장 정보, 이벤트, 티켓/팝콘 스토어 등 다양한 서비스 제공

1. 프로젝트 개요

- 개발환경(S/W)



요구 분석서

Microsoft Power Point, Word



ER Diagram

Adobe Illustrator CS6



DB모델링

eXERD, Sql Developer



데이터 작업

Microsoft Excel



2. 요구사항 분석

● 사용자 요구분석

• 영화 정보

1. 개봉 날짜
2. 차트
3. 상영 극장
4. 상영 시간
5. 평점
6. 예매율
7. 누적 관객 수
8. 가격
9. 관람 등급

• 극장 정보

1. 지역(시,도)
2. 지점명
3. 주소
4. 상영관 수
5. 전체 좌석 수
6. 특별관 여부
7. 상영중인 영화
8. 상영 시간

• 이벤트 정보

1. 멤버십
2. 영화/예매
3. CGV아트하우스
4. 제휴/할인
5. 시사/무대인사
6. CGV극장별

• 스낵 정보

1. 스낵 이름
2. 가격
 - 1) 결제 취소 시 -1000원 된 가격 환불
3. 판매 극장 지점명
4. 구매 시 휴대폰으로 기프티콘 전송

• 후기 작성

1. 로그인 한 회원만 작성 가능
2. 관람한 영화만 작성 가능

2. 요구사항 분석

● 사용자 요구분석

• 예매

1. 현재 상영중인 영화만 가능
2. 영화가 상영중인 극장 필요
3. 날짜는 현재 날짜와 같거나 다음 날짜
4. 시간은 지금 시간보다 뒤
5. 인원 최대 8명
6. 좌석은 이미 예매한 좌석은 선택 X
7. 결제
 - 1) 신용카드일 경우 카드종류, 카드번호, 비밀번호 앞 두자리, 카드 유효기간(월,년), 법정생년월일(앞 6자리) 필요
 - 2) 카카오페이일 경우 휴대폰번호, 생년월일 필요
 - 3) 휴대폰결제일 경우 이동통신사, 휴대폰번호, 주민등록번호 앞 6자리(생년월일) + 뒤 한자리(성별) 필요
 - 4) 페이코일 경우 이메일 또는 휴대폰아이디, 비밀번호 필요
8. 예매취소
 - 1) 신용카드는 결제 후 3일 이내 취소 시 승인 취소 가능, 3일 이후 취소 시 3~5일 소요
 - 2) 체크카드는 결제 후 3일 이내 취소 시 당일 카드사에서 환불 처리, 3일 이후 취소 시 3~10일 소요
 - 3) 부분 취소 불가능

2. 요구사항 분석

● 사용자 요구분석

• 회원 정보

1. 예매내역 관리
 - 1) 예매내역
 - 지난 1개월까지의 예매내역을 확인 가능, 영수증은 신용카드 결제 내역만 출력 가능
 - 2) 취소내역
 - 상영일 기준 지난 7일 동안의 취소내역만 확인 가능
2. 프로필 관리
 - 1) 닉네임
 - 한글, 영문, 숫자 혼용 가능 (한글 기준 10자 이내)
 - 2) 프로필 이미지
 - jpg, gif, BMP 파일만 등록 가능 (최대 3M)
3. 정보 관리
 - 1) 필수 정보
 - 성명, 아이디, 비밀번호, 생년월일, 성별, 휴대폰번호, 이메일주소
 - 2) 선택 정보
 - 선호장르, 주중선호요일, 선호 시간대, 자주 가는 CGV(최대 5개), 기념일, 주소, 성별, 생년월일 등
4. 쿠폰 관리
 - 1) CGV 영화관람권 보유(등록) 내역
 - 2) CGV 영화관람권 사용내역
 - 3개월 이전의 내역은 자동 삭제
5. 나의 문의내역(Q&A)
6. 관람한 영화 내역

2. 요구사항 분석

● 관리자 요구분석

• 회원 관리

1. 회원 가입 전 회원 가입 여부로 가입한 회원인지 관리 한 후 회원가입 할 수 있게 관리
 - 1) 이름, 생년월일, 휴대전화번호 뒤 7~8자리 필요
 - 2) 가입한 회원일 경우 로그인
 - 3) 가입 안된 회원일 경우 회원 가입
2. 회원 가입 시 필수 정보 입력 받아야만 가입 완료되게 관리
 - 1) 회원 ID, 비밀번호, 비밀번호 확인, 이름, 생년월일, 휴대전화번호, 이메일 등 필요
 - 2) 회원 ID는 중복 검사 필요
 - 3) 비밀번호는 비밀번호 확인과 일치 필요
3. 회원은 아이디, 비밀번호로 로그인할 수 있게 관리
 - 1) 아이디, 비밀번호 틀릴 시 재입력
 - 2) 아이디 모를 경우 아이디 찾기
 - 3) 비밀번호 모를 경우 비밀번호 찾기

• 비회원 관리

1. 비회원으로 예매가 가능하게 비회원 로그인 관리
 - 1) 이름, 휴대폰번호, 법정생년월일, 비밀번호, 비밀번호 확인 필요
 - 2) 비밀번호는 비밀번호 확인과 일치 필요
2. 비회원인 경우 청소년 관람불가 영화 예매 할 수 없게 관리
3. 비회원인 경우 영화 예매 시 신용카드만 결제 가능하게 관리
4. 비회원인 경우 회원만 이용 가능한(평점, 댓글, 인정 등) 서비스 이용 할 수 없게 관리

2. 요구사항 분석

● 관리자 요구분석

- 영화 관리

1. 사이트 접속자가 원하는 영화를 찾을 수 있게 무비 파인더 제공
2. 영화 예매율, 관람객, 평점에 따른 순위를 볼 수 있는 무비 차트 제공
3. 영화의 주요내용, 트레일러, 스틸컷 등을 보여주는 영화 상세 제공
4. 극장마다 상영 시간표 관리

- 영화 예매 관리

1. 로그인한 회원이거나, 비회원 로그인을 한 비회원만 예매가 가능하게 관리
2. 현재 상영중인 영화만 예매가 가능하게 관리
3. 이미 지난 날짜는 예매를 할 수 없게 관리
4. 결제가 되지 않으면 예매가 완료되지 않게 관리

2. 요구사항 분석

● 관리자 요구분석

• 포인트 관리

1. 회원이 영화 예매 시 구매 금액의 5%가 누적되게 포인트 관리
2. 누적 포인트에 따라 등급이 달라지므로 등급에 따른 혜택 관리
 - 1) VIP : 일반 고객 중 최근 1년 구매누적 14,000P달성 고객
 - 혜택 : 영화 무료쿠폰 7매,매점 무료쿠폰 2매,매점 할인쿠폰 3매
 - 2) RVIP : VIP 고객 중 1년간 14,000P 달성 고객 VIP 승급 후 1년 이내에 28,000P 달성 고객
 - 혜택 : 영화 무료쿠폰 10매,매점 무료쿠폰 5매,매점 할인쿠폰 4매,
포토티켓 무료쿠폰 2매

• 지점 관리

1. 지점의 상영관, 좌석, 좌석 파손여부 등 관리
2. 좌석에 따른 가격 관리
 - 1) STANDARD ZONE
 - 2) ECONOMY ZONE
 - 3) PRIME ZONE
 - 4) SWEETBOX
 - 5) 4DX
 - 6) 장애인석

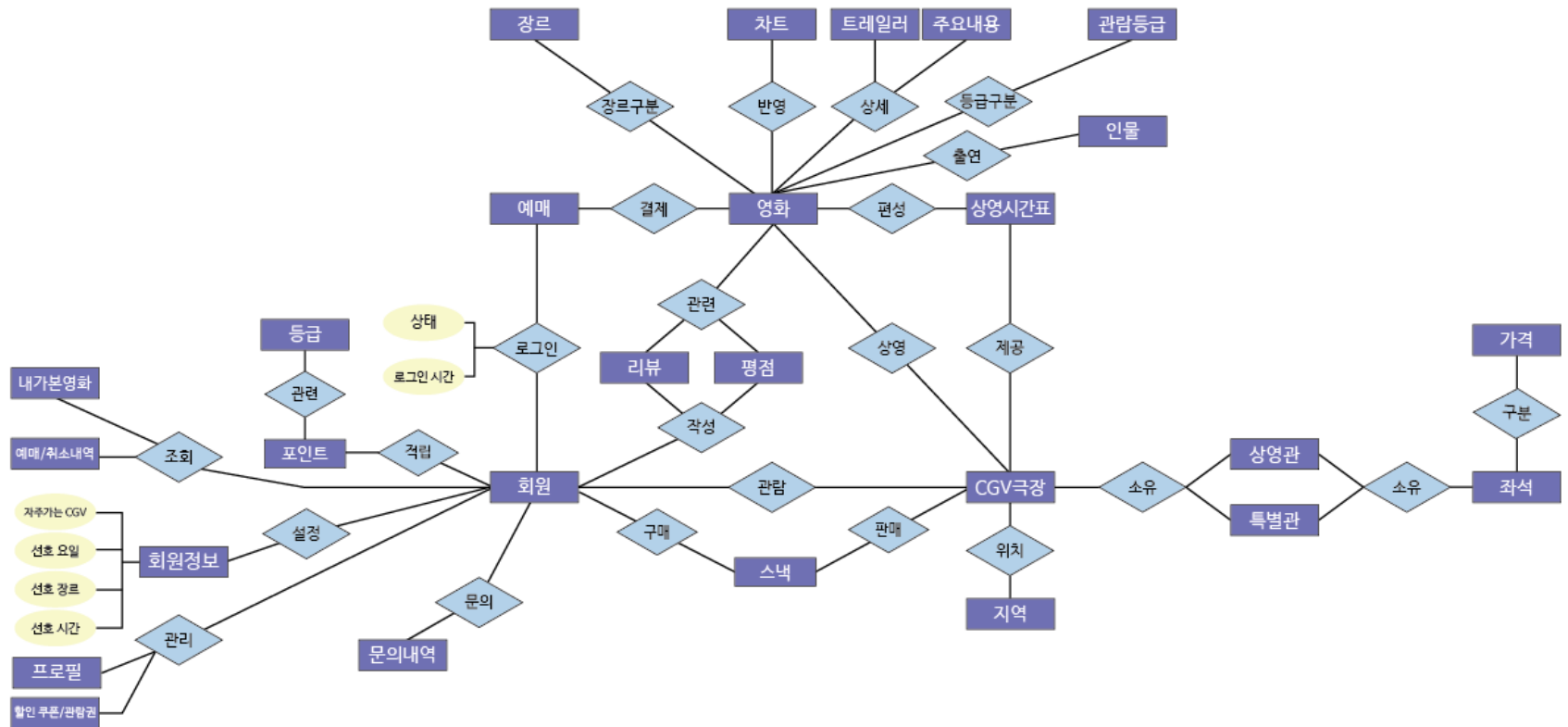
• 게시판 관리

1. 로그인한 회원만 문의를 할 수 있게 관리



3. 개념적 모델링

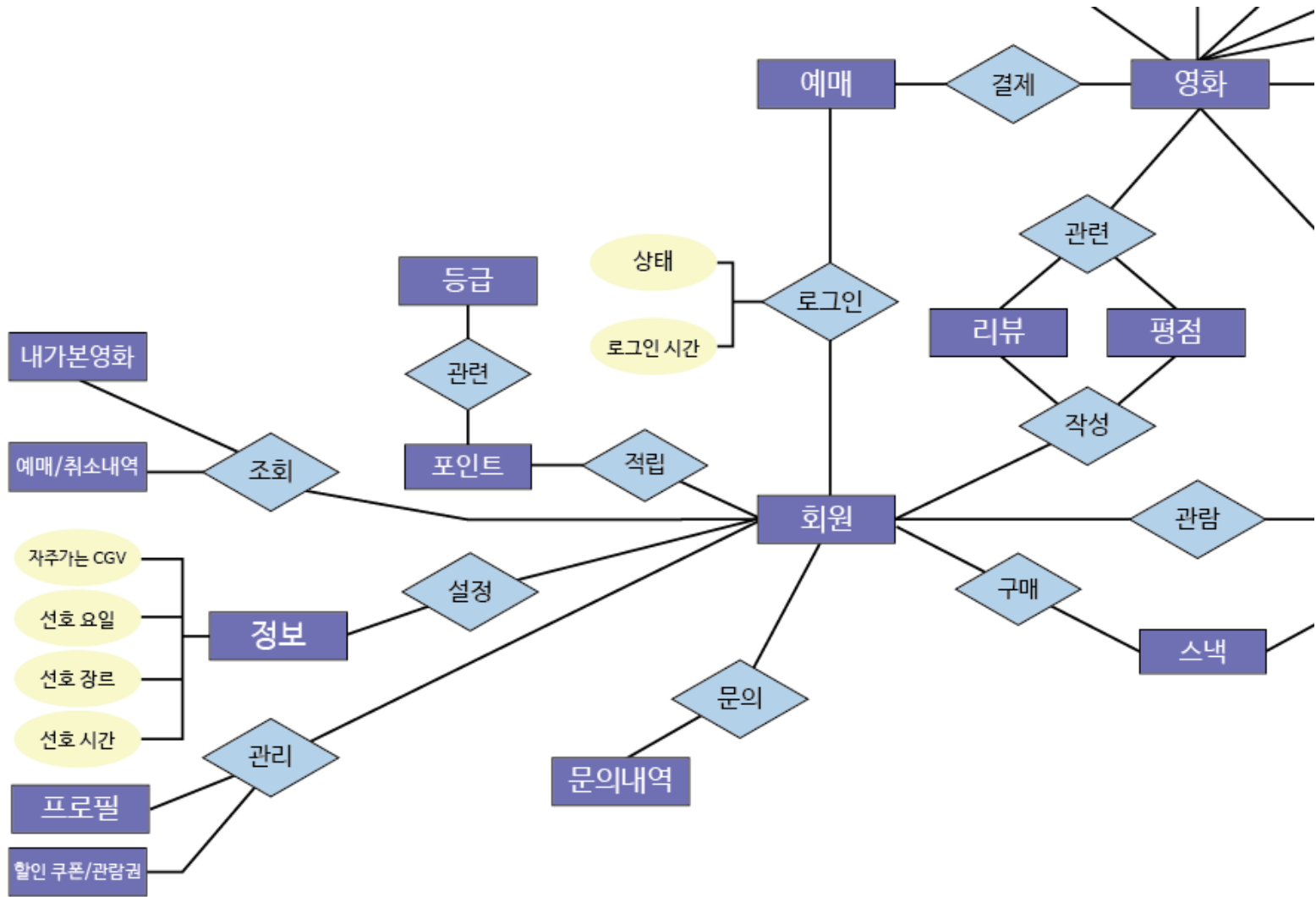
- 전체 ER Diagram



개체(Entity) 관계(Relation) 속성(Attribute)

3. 개념적 모델링

● 회원 부분



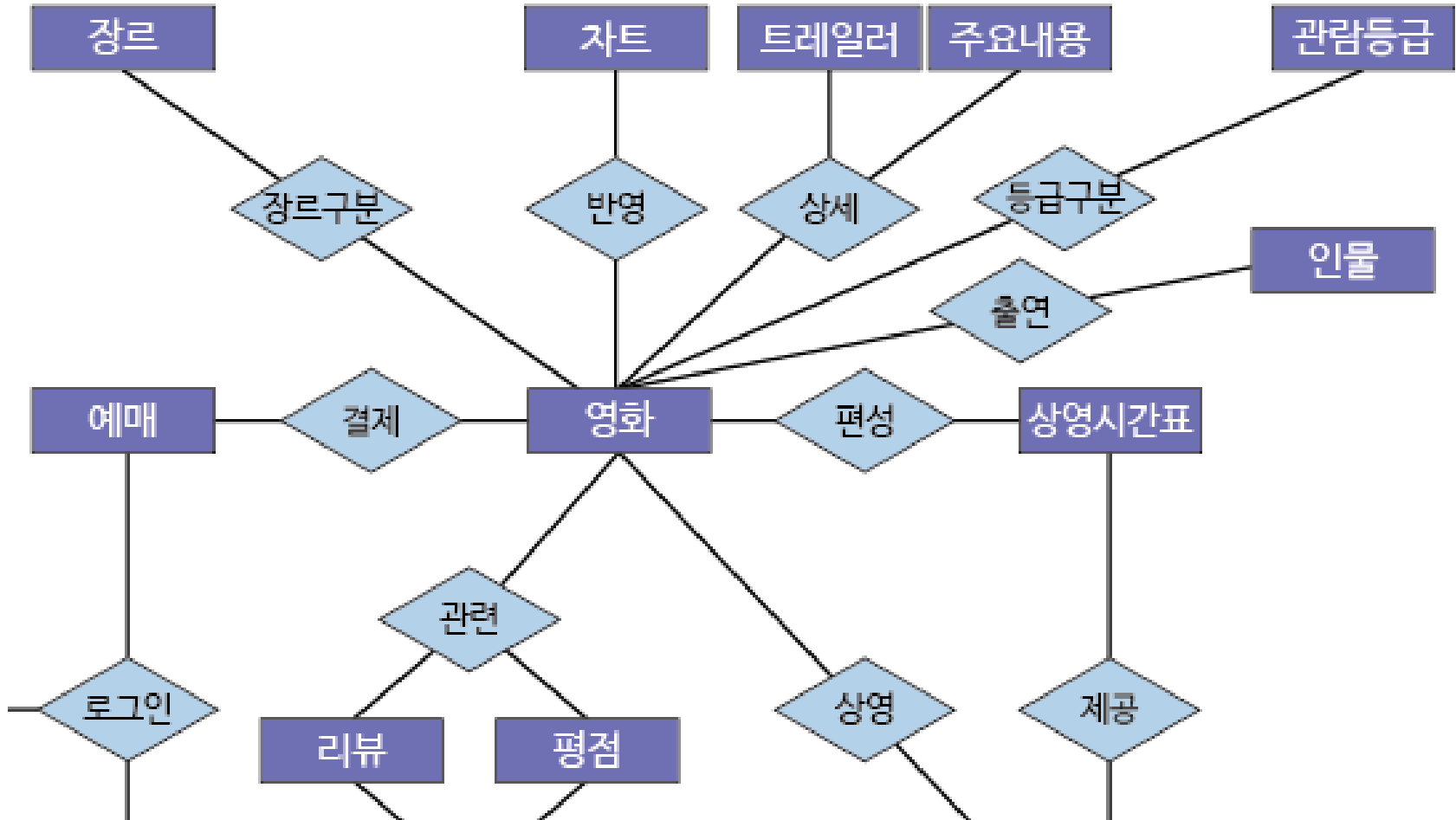
3. 개념적 모델링

● 회원 부분 속성

ENTITY	ATTRIBUTE
회원	(회원ID), 이름, 비밀번호, 생년월일, 휴대폰번호, 이메일, CJ ONE 마케팅 정보 수신 동의
예매	(예매번호), 상영관, 상영시간, 영화제목, 예매날짜, 인원, 좌석, 가격
결제	(결제코드), 예매번호, 결제수단명, 결제날짜, 결제금액
포인트	(카드번호), 누적포인트
등급	(등급코드), 등급명, 혜택
예매/취소내역	(내역순번), 관람영화, 관람 CGV, 관람 일시, 취소 일시, 결제 취소 금액
문의내역	(문의번호), 문의유형, 제목, 내용, 등록일, 상태, 만족도
프로필	닉네임, 프로필이미지
할인쿠폰/관람권	(할인쿠폰코드), 할인쿠폰명
리뷰	호감리뷰, 비호감리뷰, 무비매니아여부, 추천수
평점	감독연출 점수, OST점수, 스토리점수, 영상미점수, 배우연기점수, 전체 평가

3. 개념적 모델링

● 영화 부분



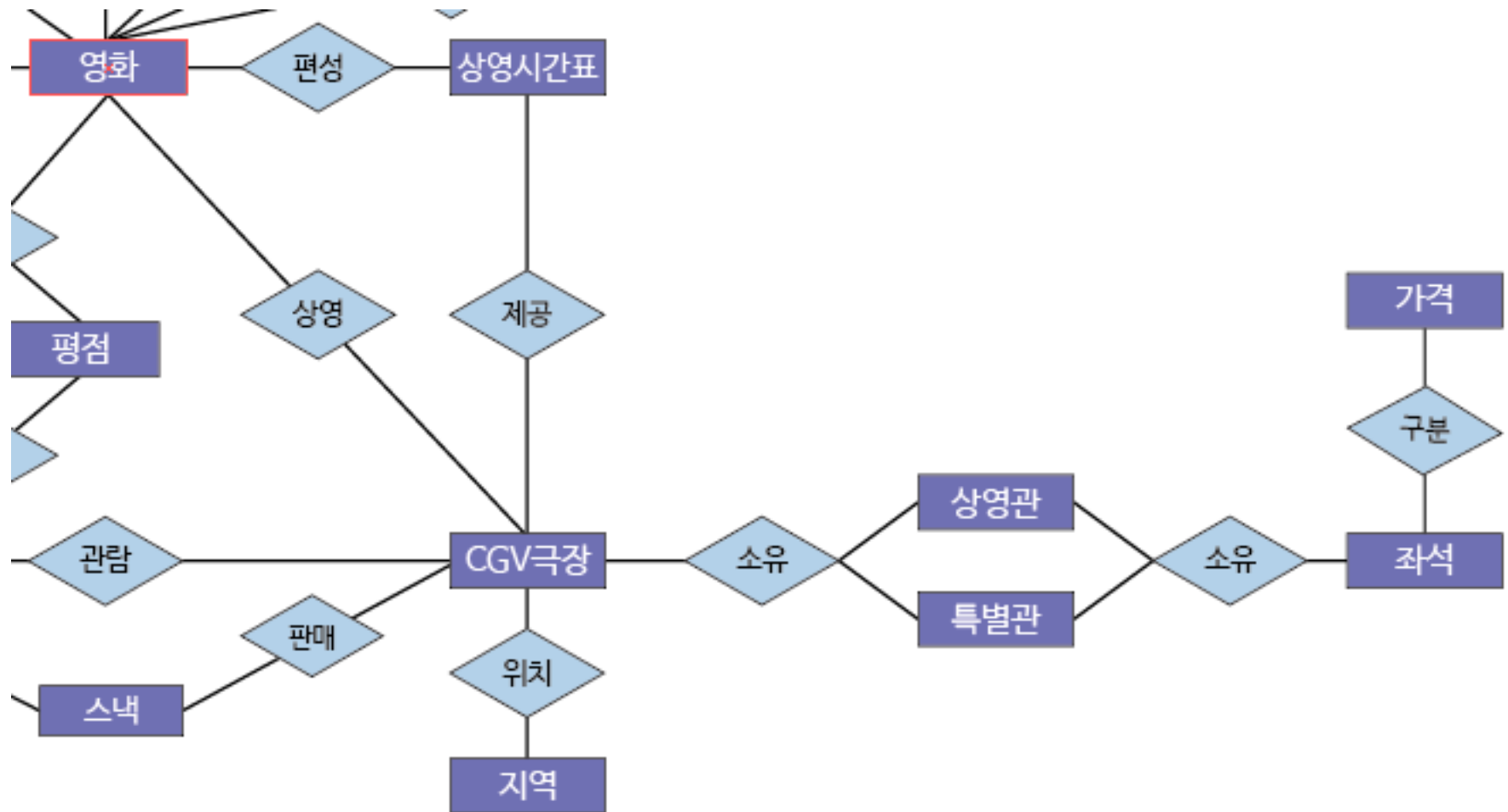
3. 개념적 모델링

- 영화 부분 속성

ENTITY	ATTRIBUTE
영화	(영화코드), 영화제목, 등급, 장르, 개봉날짜, 누적관람객, 주요내용, 감독, 출연진
장르	(장르코드), 장르명
차트	(차트코드), 영화제목, 좋아요 수, 현재 상영 여부
주요내용	(주요내용코드), 제작노트, 포토, 배우명대사
트레일러	(예고편코드), 예고편동영상
인물	(인물코드), 이름, 출생, 국적, 역할, 바이오그래피, 연관작품
관람등급	(등급코드), 등급명, 등급기준
상영시간표	(상영시간코드), 상영관이름, 영화제목, 날짜, 시간, 자막/더빙, 영상기술, 러닝타임

3. 개념적 모델링

● 극장 부분



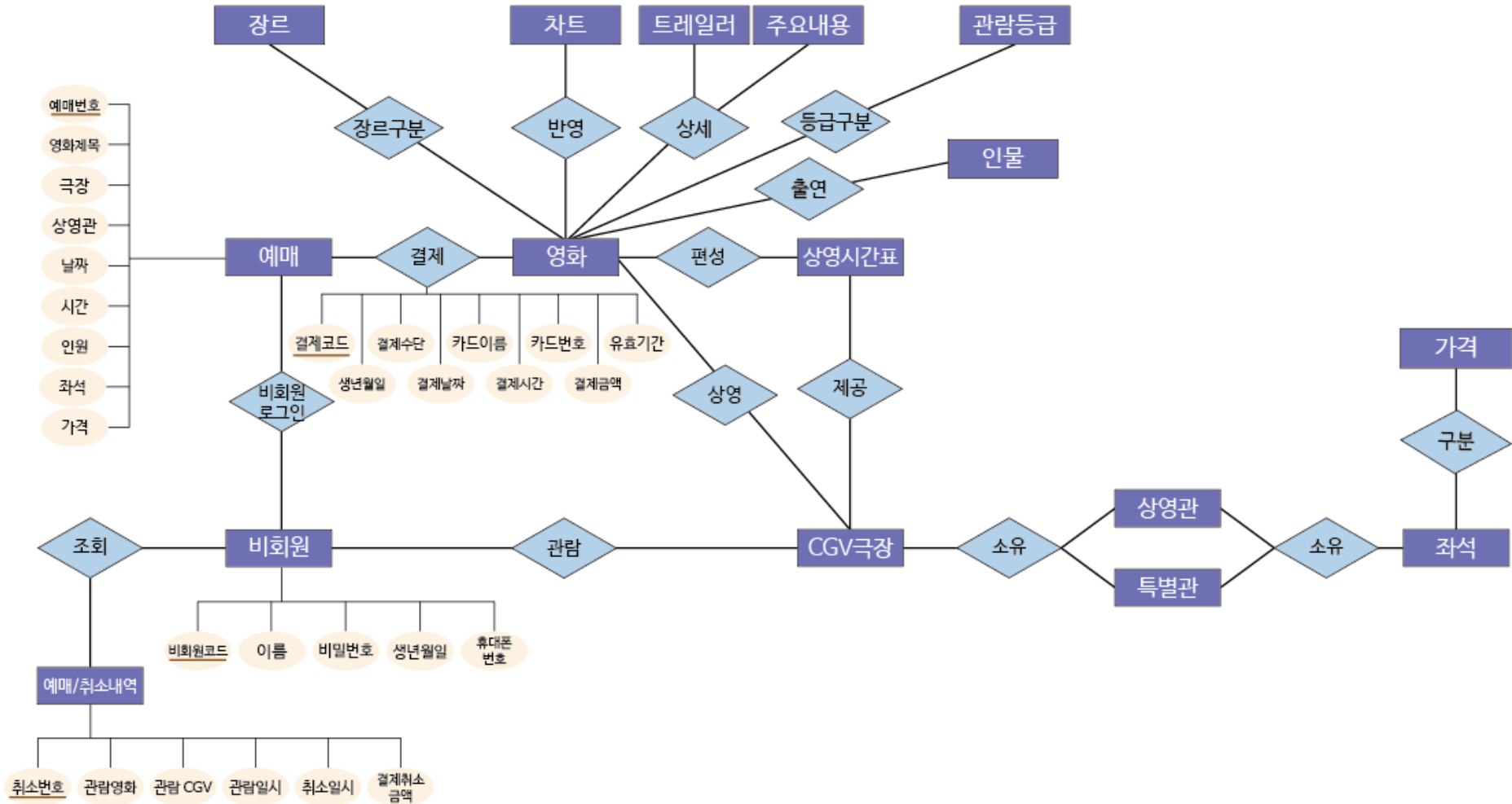
3. 개념적 모델링

● 극장 부분 속성

ENTITY	ATTRIBUTE
CGV 극장	(극장코드), 지역, 지점명, 지점위치, 상영관 수, 전체좌석 수, 특별관 여부
지역	(지역코드), 지역명
스낵	(스낵코드), 종류, 스낵명, 상품설명, 스낵가격
상영관/특별관	(상영관코드), 상영관 이름, 총 좌석, 특별관 코드
좌석	(좌석코드), 좌석명, 파손여부
가격	(가격코드), 좌석코드, 성인가격, 청소년가격, 어린이가격, 우대가격

3. 개념적 모델링

● 비회원 ER Diagram

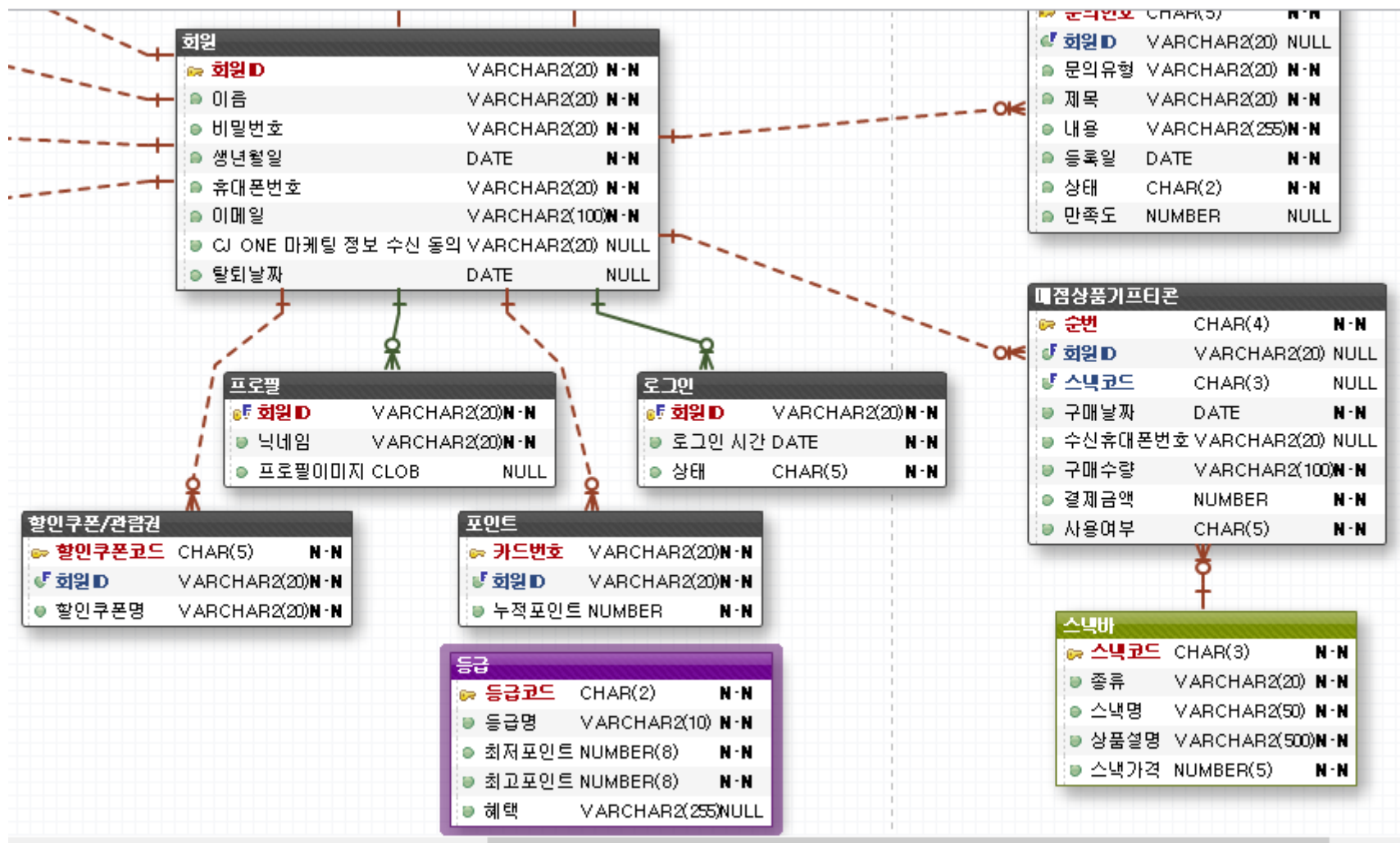




4. 논리적 모델링

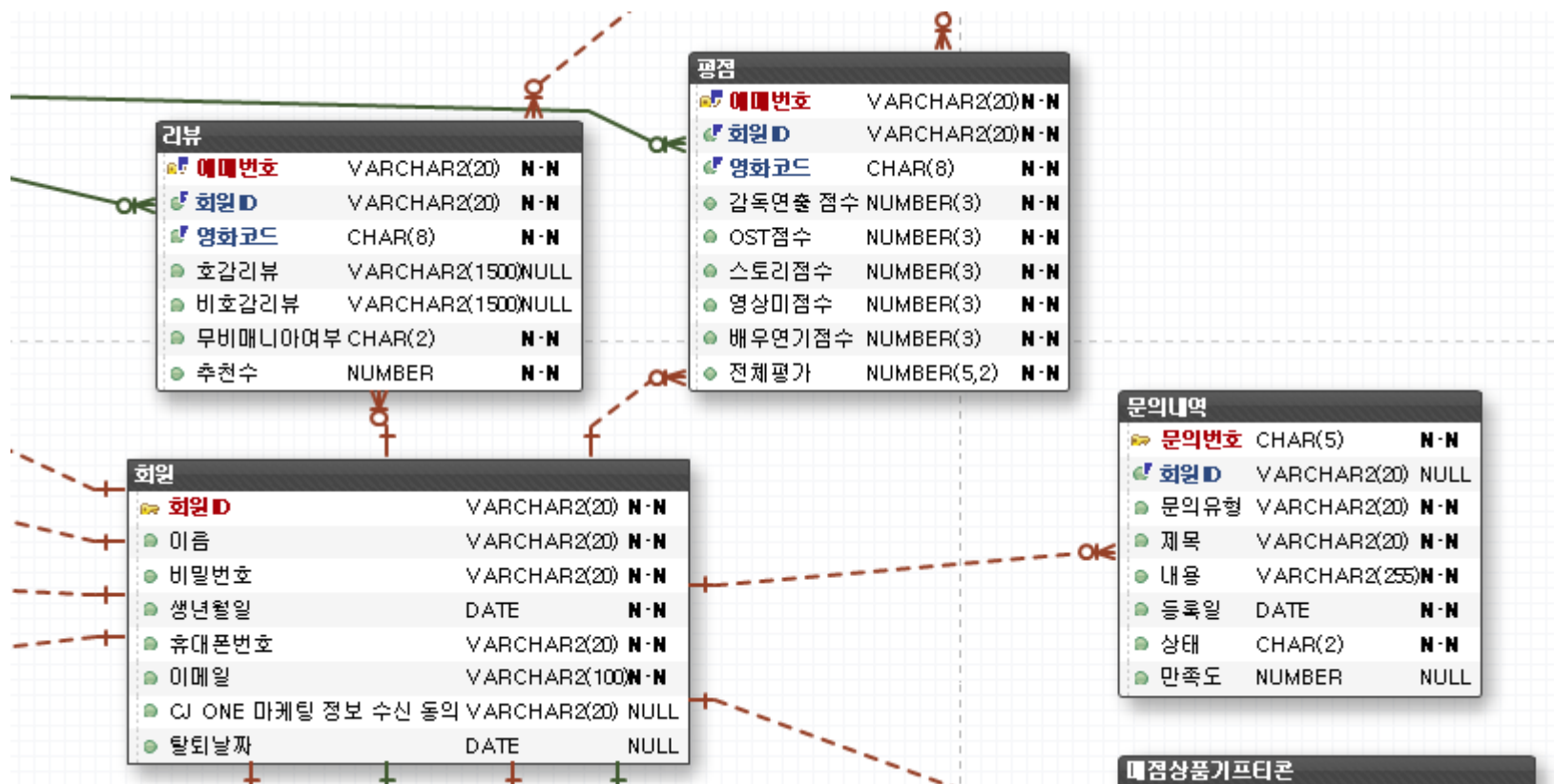
4. 논리적 모델링

● 회원 부분



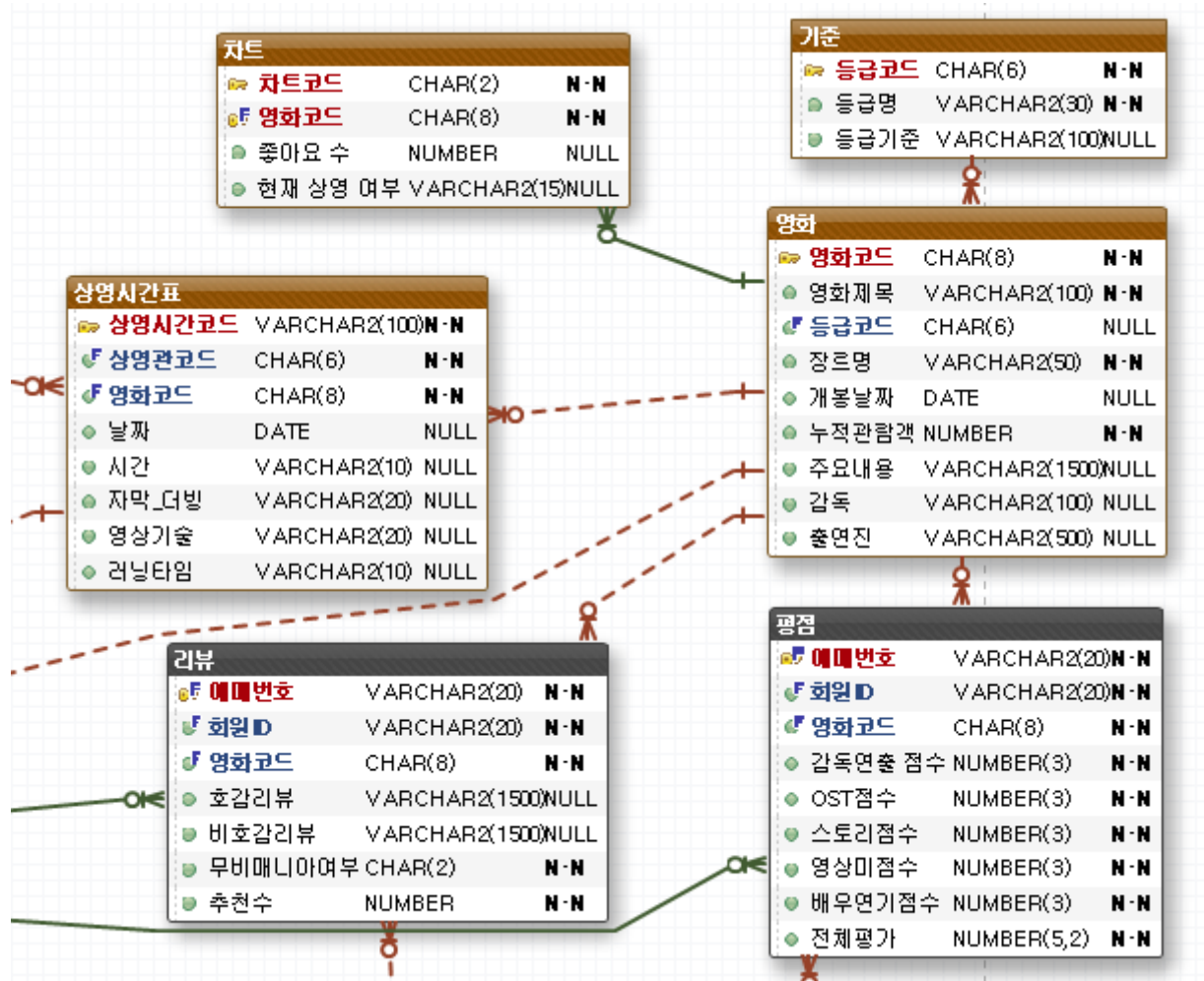
4. 논리적 모델링

● 회원 부분



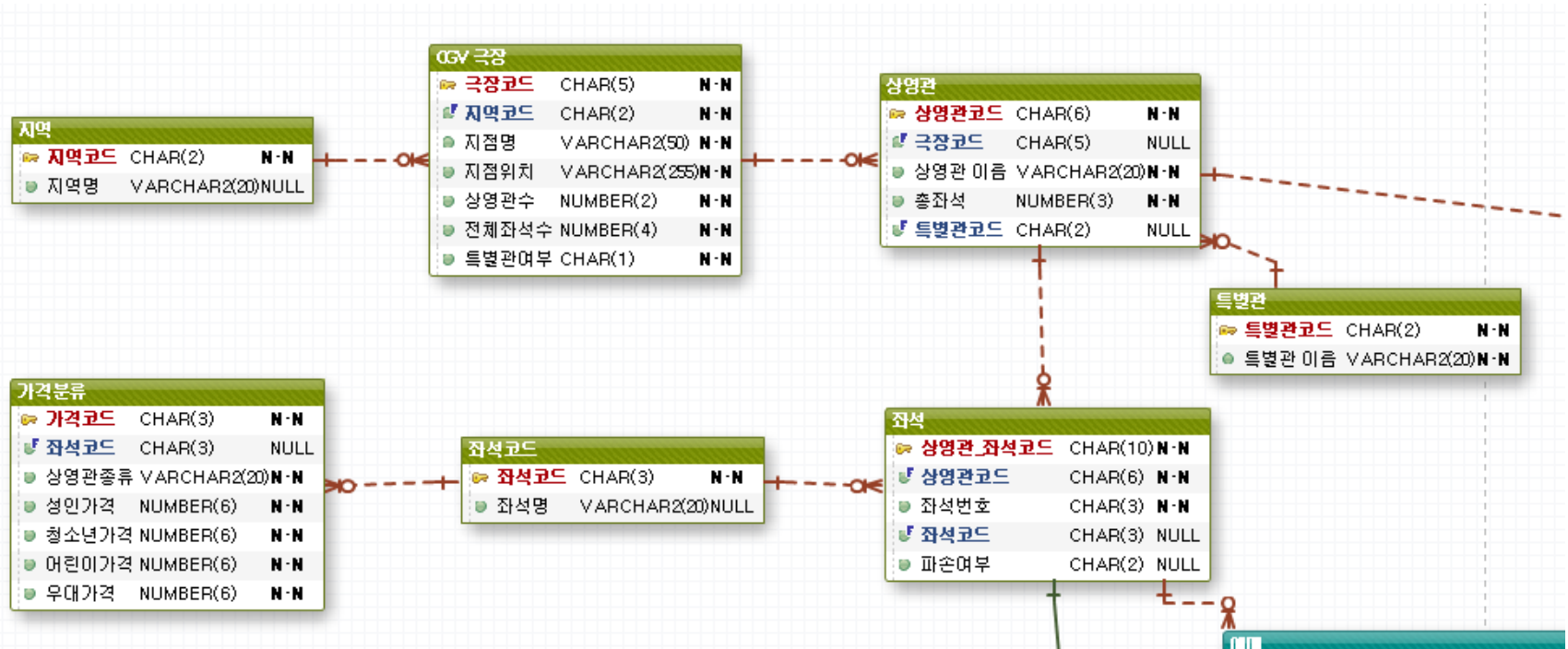
4. 논리적 모델링

● 영화 부분



4. 논리적 모델링

극장 부분



4. 논리적 모델링

예매 / 결제 부분

비회원 결제		
결제코드	CHAR(5)	N-N
예매번호	VARCHAR2(20)	N-N
비회원코드	CHAR(5)	NULL
결제수단	VARCHAR2(20)	NULL
카드이름	VARCHAR2(20)	NULL
카드번호	VARCHAR2(20)	NULL
비밀번호	VARCHAR2(20)	NULL
유효기간	DATE	NULL
생년월일	DATE	NULL
결제날짜	DATE	NULL
결제시간	VARCHAR2(20)	NULL
결제금액	NUMBER(6)	NULL

비회원		
비회원코드	CHAR(5)	N-N
이름	VARCHAR2(20)	NULL
생년월일	DATE	NULL
비밀번호	VARCHAR2(20)	NULL
휴대폰번호	VARCHAR2(20)	NULL

예매		
예매번호	VARCHAR2(20)	N-N
상영시간코드	VARCHAR2(100)	NULL
회원ID	VARCHAR2(20)	NULL
영화코드	CHAR(8)	NULL
상영관 좌석코드	CHAR(10)	NULL
예매날짜	DATE	N-N
인원	NUMBER(1)	N-N
가격	NUMBER(6)	NULL

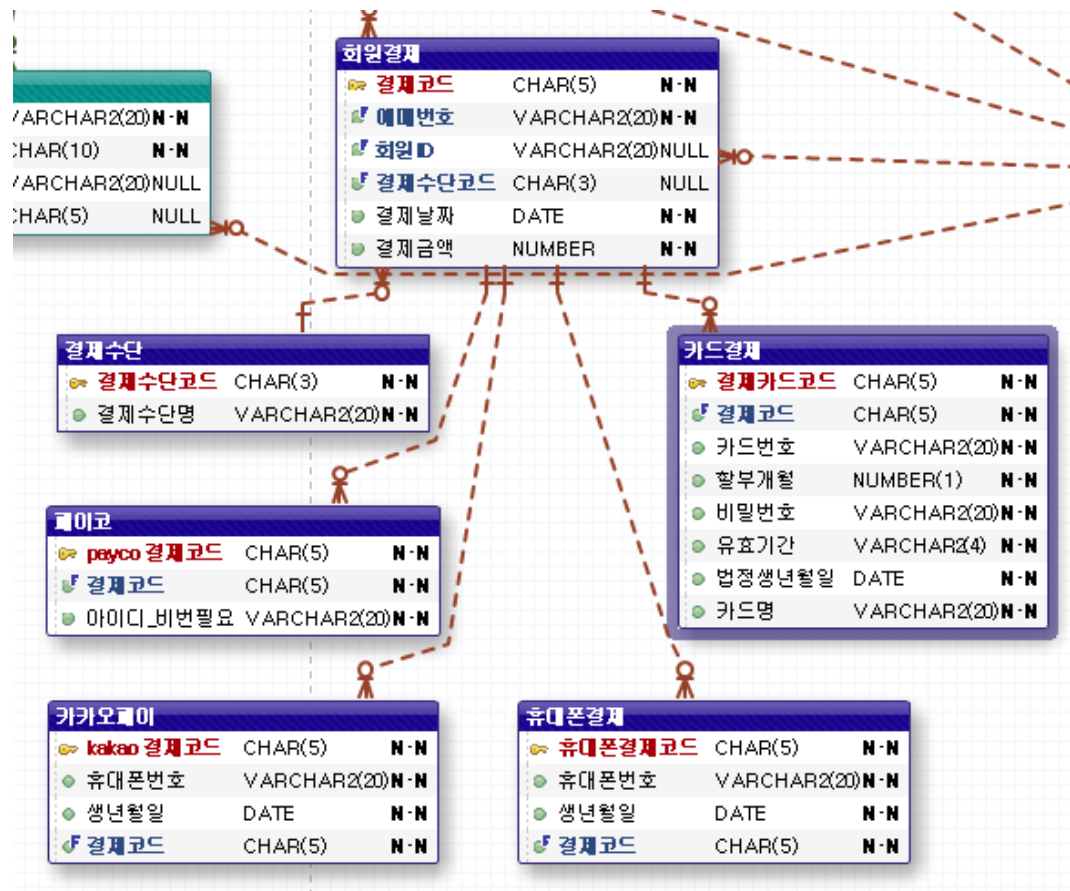
예매 좌석		
예매번호	VARCHAR2(20)	N-N
상영관 좌석코드	CHAR(10)	N-N
회원ID	VARCHAR2(20)	NULL
비회원코드	CHAR(5)	NULL

예매 취소내역		
내역순번	CHAR(4)	N-N
예매번호	VARCHAR2(20)	N-N
회원ID	VARCHAR2(20)	N-N
관람일시	DATE	N-N
취소일시	DATE	NULL
금액	NUMBER	N-N

회원결제		
결제코드	CHAR(5)	N-N
예매번호	VARCHAR2(20)	N-N
회원ID	VARCHAR2(20)	NULL
결제수단코드	CHAR(3)	NULL
결제날짜	DATE	N-N
결제금액	NUMBER	N-N

4. 논리적 모델링

예매 / 결제 부분

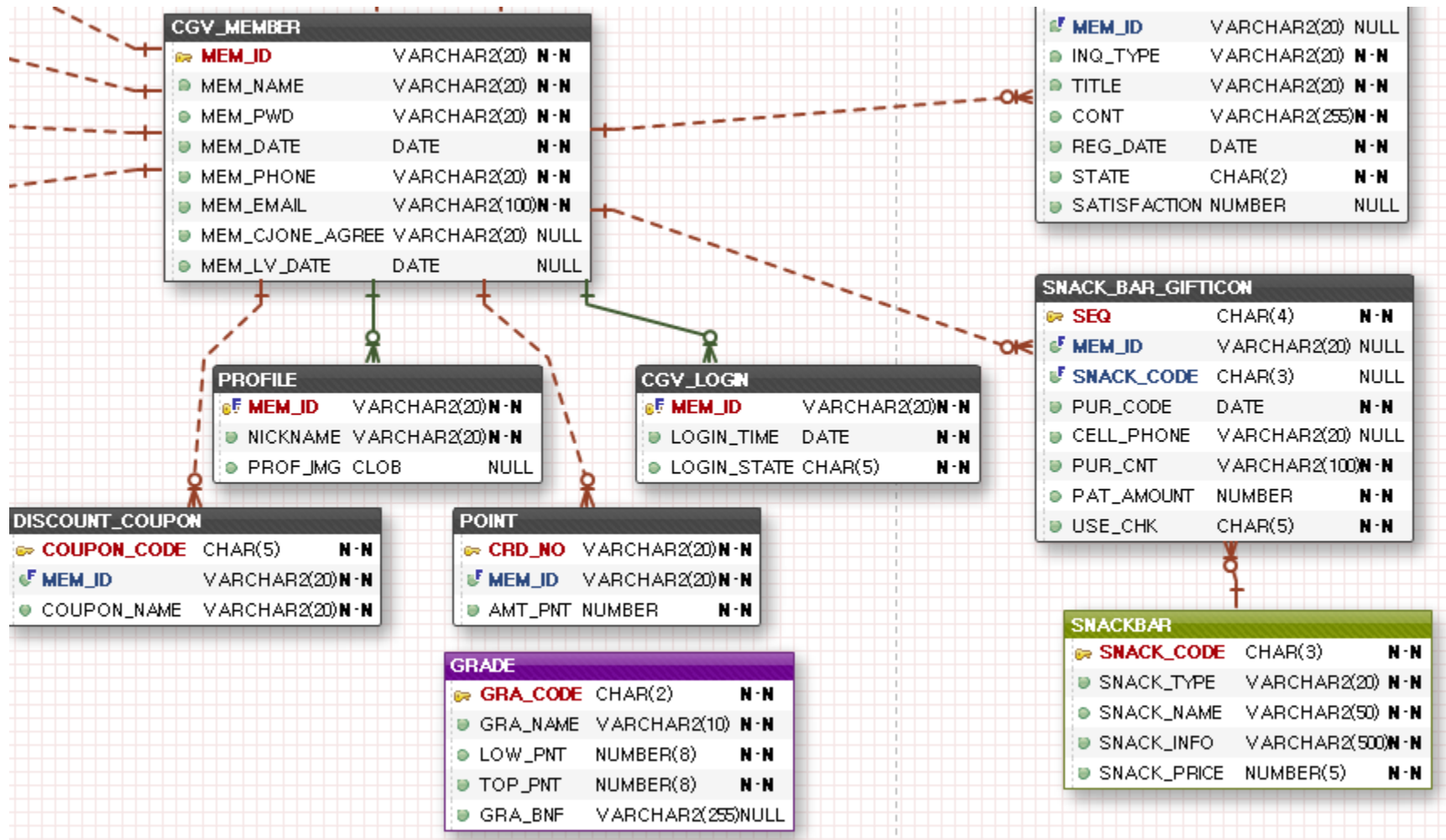




5. 물리적 모델링

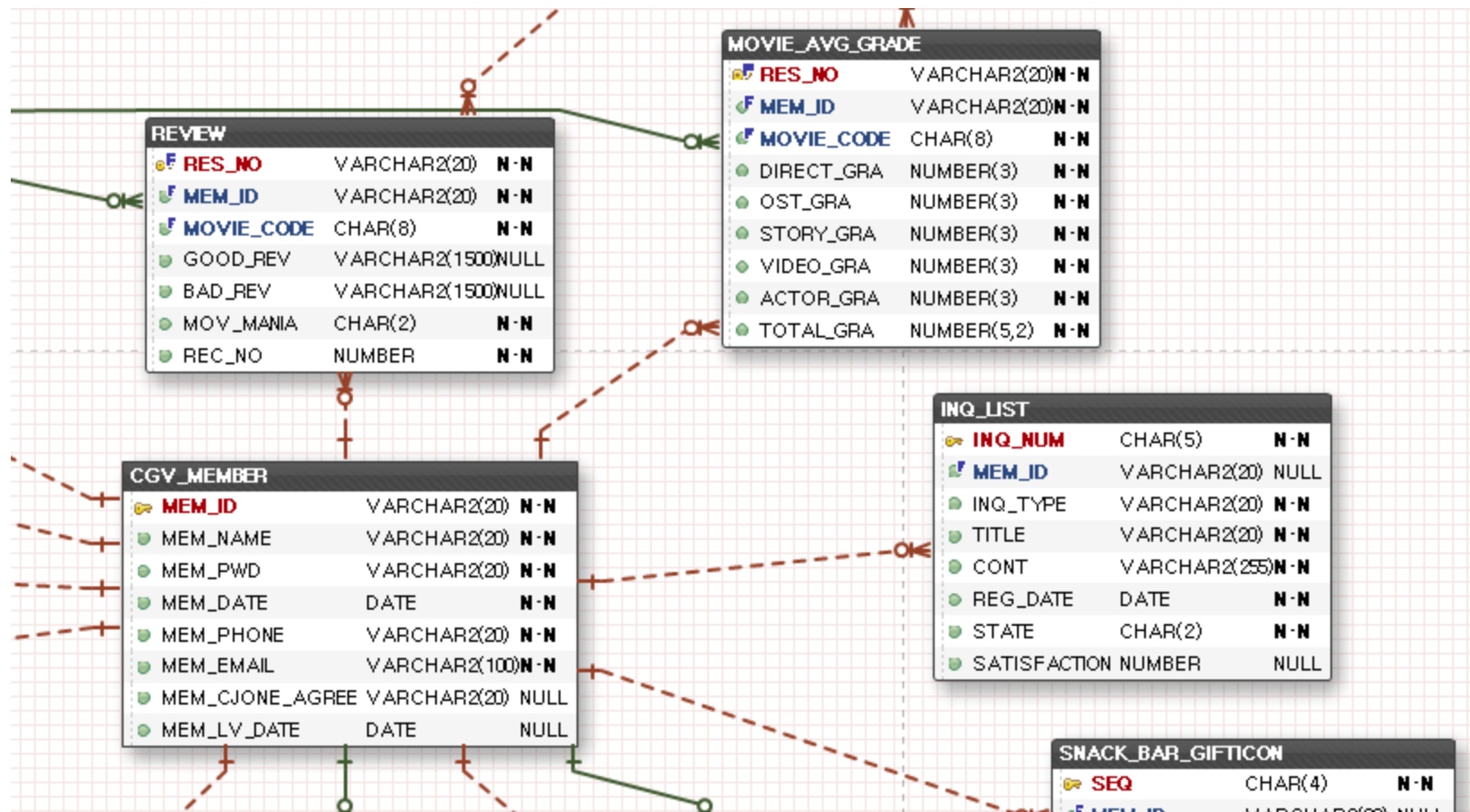
5. 물리적 모델링

• 회원 부분



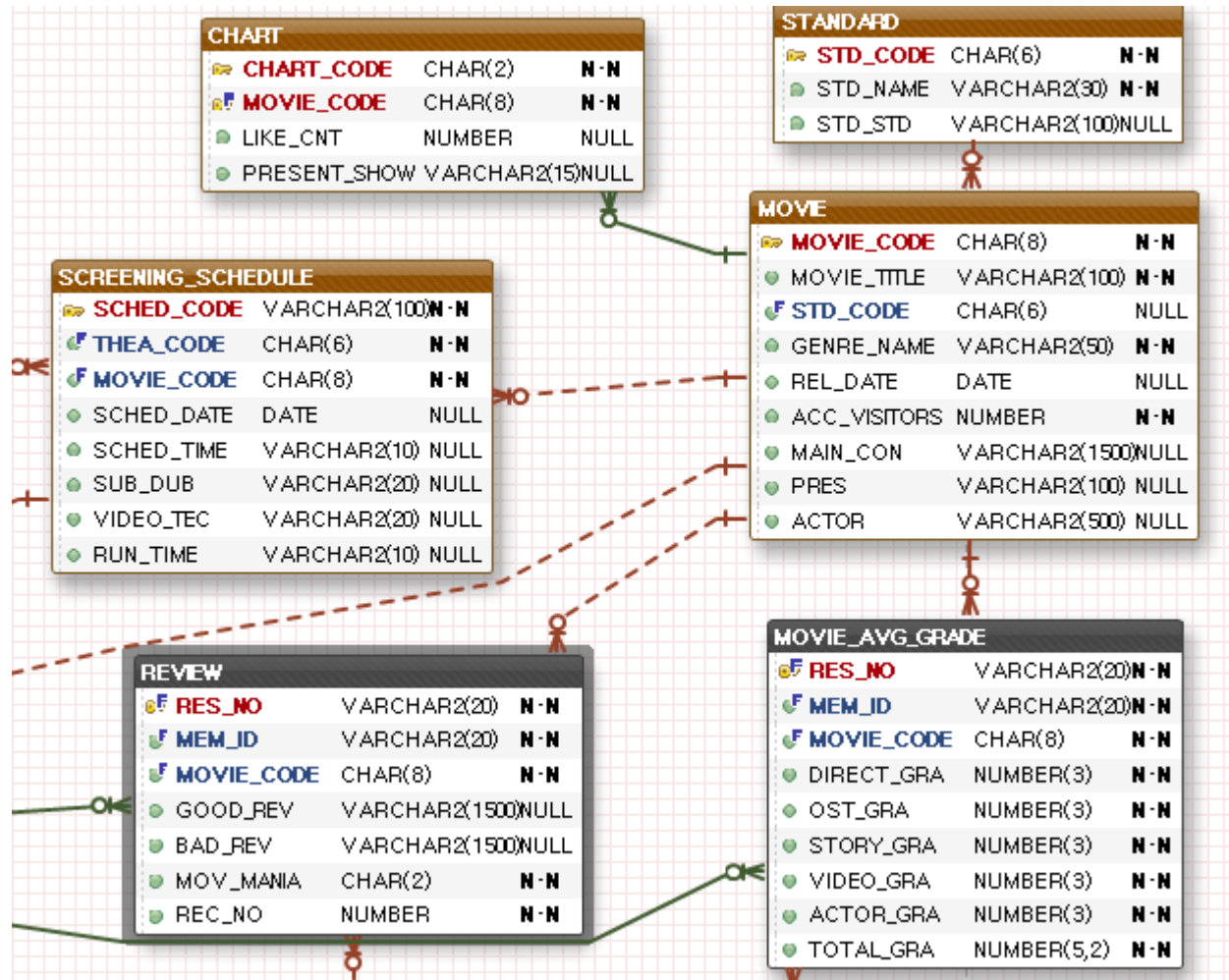
5. 물리적 모델링

● 회원 부분



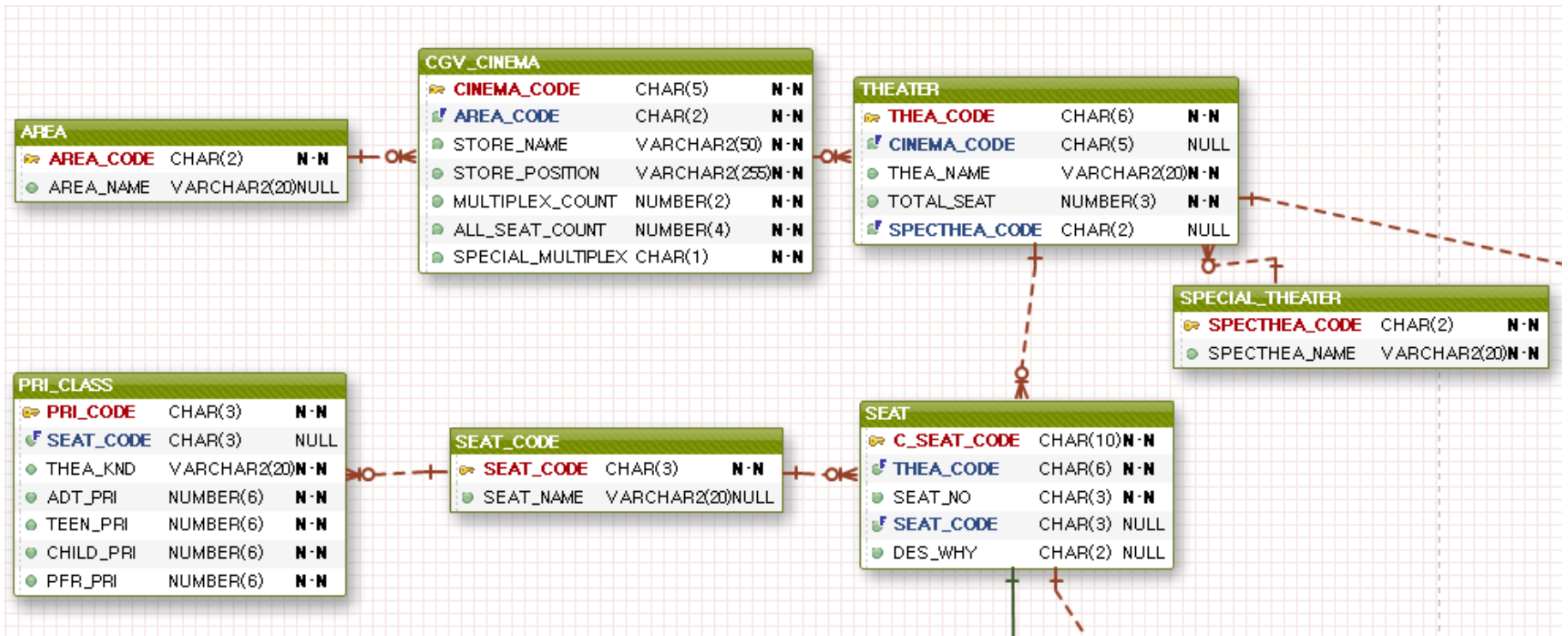
5. 물리적 모델링

영화 부분



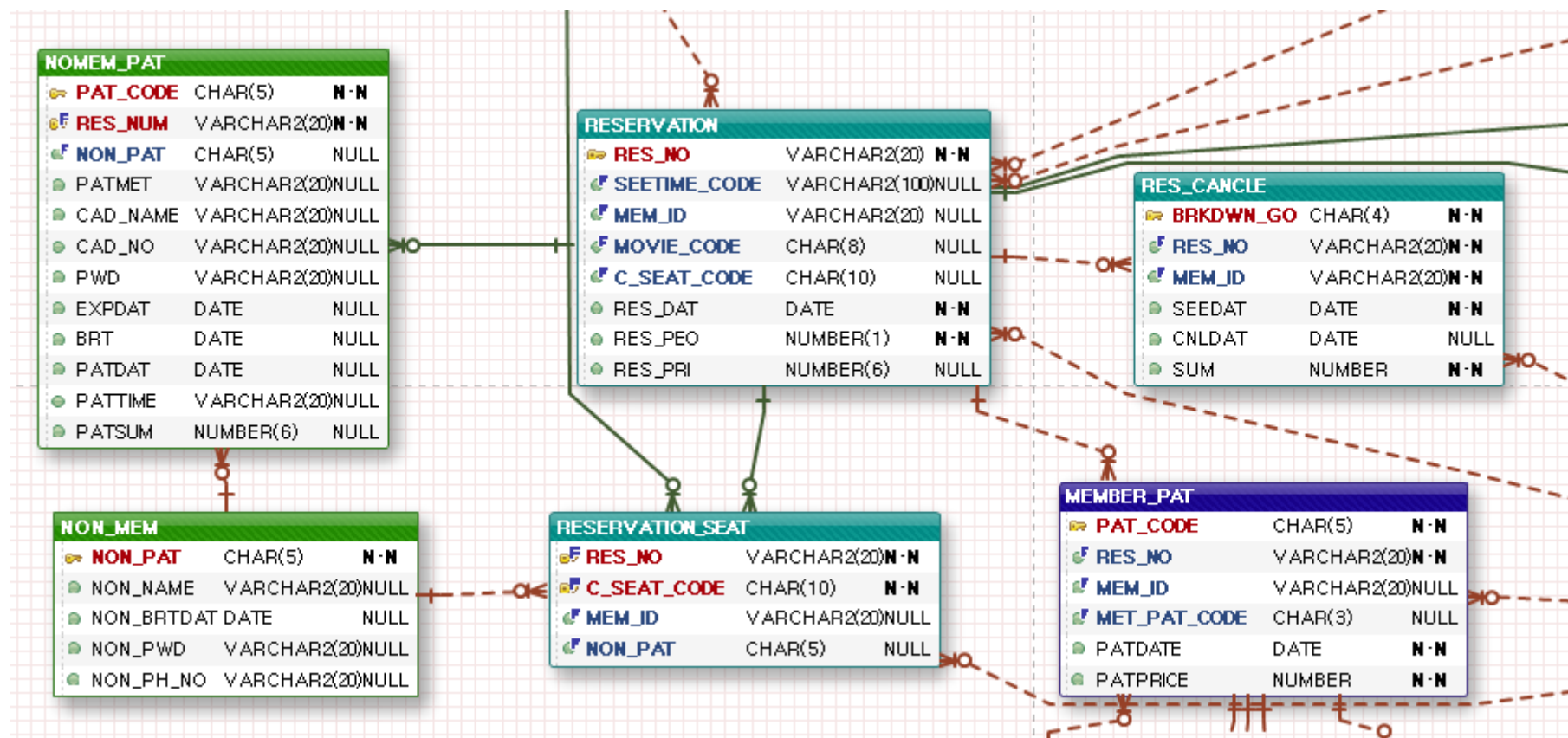
5. 물리적 모델링

극장 부분



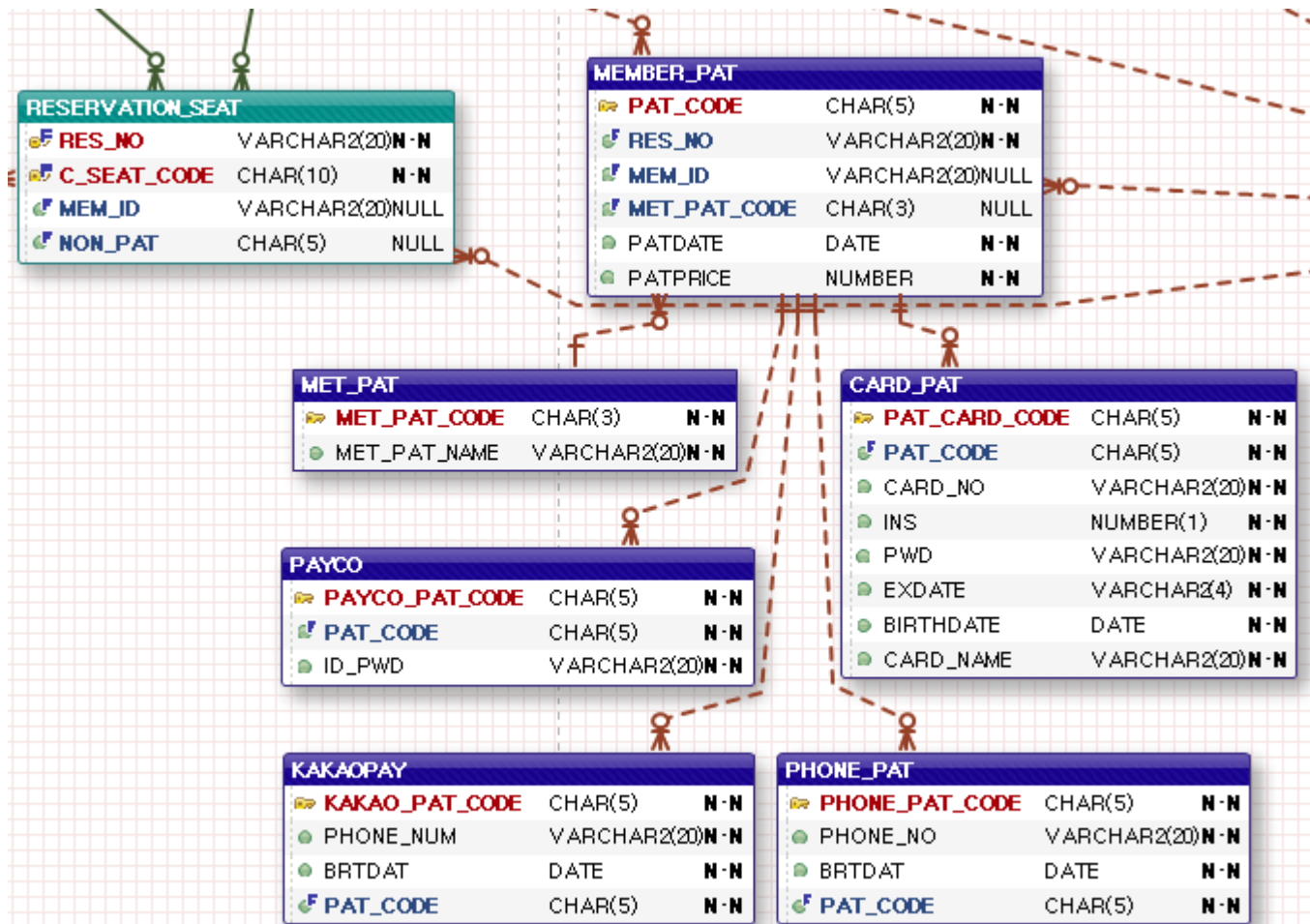
5. 물리적 모델링

- 예매 / 결제 부분



5. 물리적 모델링

예매 / 결제 부분





6. PL/SQL

● CGV 화면 / 회원 여부 확인

회원가입 여부 안내

- 기존 회원가입 정보와 일치하는 정보를 입력하셔야 회원가입 여부를 정확하게 확인하실 수 있습니다.
- CJmall(CJ오쇼핑), CJ온마트, CJ월디스, CJ문화재단, CJ올리브영 에서는 전자상거래에 의거하여 만 14세 미만의 어린이/학생의 회원가입을 제한합니다.
- 법정생년월일은 주민등록번호 앞 6자리, 휴대전화번호는 국번 3자리(01X)를 제외하 휴대전화번호 뒤 7~8자리를 입력해주세요

이름을 입력해주세요.

법정생년월일 6자리를 입력해주세요.

휴대전화번호 뒤 7~8자리를 입력해주세요.

가입여부 확인

6. PL/SQL

● 회원 여부 확인 PL/SQL

--1)회원가입 여부 프로시저

```
CREATE OR REPLACE PROCEDURE UP_MEMBER_JOIN_CHECK
(
  P_MEM_NAME IN CGV_MEMBER.MEM_NAME%TYPE,
  P_MEM_DATE IN CGV_MEMBER.MEM_DATE%TYPE,
  P_MEM_PHONE IN CGV_MEMBER.MEM_PHONE%TYPE
)
IS
  V_COUNT NUMBER:=0;
BEGIN
  SELECT COUNT(*)
  INTO V_COUNT
  FROM CGV_MEMBER WHERE MEM_NAME=P_MEM_NAME AND MEM_DATE=P_MEM_DATE AND (SUBSTR(MEM_PHONE,4,11))=P_MEM_PHONE;

  IF V_COUNT > 0 THEN
    DBMS_OUTPUT.PUT_LINE('회원가입 불가능합니다.');
```

--확인(불가능)

```
EXEC UP_MEMBER_JOIN_CHECK ('함태훈', '89/05/22', '99448905');
회원가입 불가능합니다.
```

--확인(가능)

```
EXEC UP_MEMBER_JOIN_CHECK('서희정', '94/05/20', '99374928');
회원가입 가능합니다.
```

MEM_ID	MEM_NAME	MEM_PWD	MEM_DATE	MEM_PHONE	MEM_EMAIL	MEM_CJONE_AGREE	MEM_LV_DATE
1 vietto	함태훈	U"ilD	89/05/22	01099448905	hamzzange@naver.com	NO	(null)
2 zzang	최종혁	a	93/02/16	01063409513	adcd515@naver.com	YES	(null)
3 kenik	이창익	x!8	00/01/01	01091253653	kenik@naver.com	YES	(null)
4 test01	테스트	+EpnTpi	50/01/01	01012341234	test@navercom	NO	(null)
5 airfos	이형호	EE	91/07/23	01022317629	jsalhh3@naver.com	NO	(null)
6 lyg3399	이창범	(m)	92/07/12	01041250098	lyg3399@naver.com	NO	(null)
7 saejinchu	문세진	j6	94/12/01	01048548025	saejinchu@naver.com	NO	(null)

6. PL/SQL

● CGV 화면 / 회원가입

기본정보

✓ 표시는 필수 입력 항목입니다.

✓ 이름	김영민
✓ 아이디	<input type="text" value="아이디를 입력해주세요."/> <input type="button" value="중복확인"/> <input type="button" value="CJ ONE 통합 아이디 안내 >"/> <small>CGV, CJmail, Mnet 등 기존의 CJ ONE 제휴 브랜드 회원님의 경우는 기존에 사용하고 계신 아이디가 새롭게 생성하신 통합 아이디로 변경됩니다.</small>
✓ 비밀번호	<input type="text" value="비밀번호를 입력해주세요."/> <small>영문자, 숫자, 특수문자 조합하여 8~12자리. 아이디와 4자리이상 동일, 반복 문자숫자 불가 사용 가능 특수 문자 : !"#%&'()*+,-./:;<=>?@[^_`{}~</small>
✓ 비밀번호 확인	<input type="text" value="비밀번호를 재입력해주세요."/>
✓ 생년월일	1967 년 10 월 10 일 <small>설정하신 생일을 기준으로 15일 전 생일쿠폰이 발행됩니다.</small>
✓ 휴대전화	010 - <input type="text"/> - <input type="text"/> <small>주문 및 배송, 쿠폰, 이벤트 정보 등을 제공 받으실 수 있습니다.</small>
✓ 이메일	<input type="text" value="이메일 아이디"/> @ <input type="text" value="naver.com"/> <input type="text" value="네이버"/> <input type="button" value="v"/> <small>이메일 주소 입력 시 사용 가능 특수 문자: - . _</small>

6. PL/SQL

● 회원가입 PL/SQL

--2) 회원가입 프로시저

```
CREATE OR REPLACE PROCEDURE UP_MEMBERJOIN
(
  p_MEM_ID          CGV_MEMBER.MEM_ID%TYPE
  , p_MEM_NAME      CGV_MEMBER.MEM_NAME%TYPE
  , p_MEM_PWD       CGV_MEMBER.MEM_PWD%TYPE
  , p_MEM_DATE      CGV_MEMBER.MEM_DATE%TYPE
  , p_MEM_PHONE     CGV_MEMBER.MEM_PHONE%TYPE
  , p_MEM_EMAIL     CGV_MEMBER.MEM_EMAIL%TYPE
  , p_MEM_CJONE_AGREE CGV_MEMBER.MEM_CJONE_AGREE%TYPE
)
IS
  v_MEM_ID CGV_MEMBER.MEM_ID%TYPE;
BEGIN
  SELECT MEM_ID INTO v_MEM_ID
  FROM CGV_MEMBER
  WHERE MEM_ID = p_MEM_ID;
  IF NOT (v_MEM_ID is null) THEN
    DBMS_OUTPUT.PUT_LINE(v_MEM_ID || '는 이미 사용중인 ID 입니다');
  END IF;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    IF NOT (LENGTH(p_MEM_PWD) BETWEEN 8 AND 16) THEN
      DBMS_OUTPUT.PUT_LINE('비밀번호 형식에 맞지 않습니다');
    ELSIF p_MEM_EMAIL NOT LIKE('%@%.%') THEN
      DBMS_OUTPUT.PUT_LINE('이메일 형식에 맞지 않습니다');
    ELSE
      INSERT INTO CGV_MEMBER VALUES( P_MEM_ID, P_MEM_NAME, CRYPTIT.ENCRYPT(P_MEM_PWD, 'pwd'),
        p_MEM_DATE, p_MEM_PHONE, p_MEM_EMAIL, p_MEM_CJONE_AGREE, NULL);
      DBMS_OUTPUT.PUT_LINE('회원 가입이 완료 되었습니다');
    END IF;
  COMMIT;
END;
```

6. PL/SQL

● 회원가입 PL/SQL 실행결과

--이메일 형식

```
EXEC UP_MEMBERJOIN('seohj0520', '서희정', 'heej1234', '94/05/20', '01099374928', 'heejung0520naver.com', 'NO');
```

이메일 형식에 맞지 않습니다

--비밀번호 형식

```
EXEC UP_MEMBERJOIN('seohj0520', '서희정', 'hj1234', '94/05/20', '01099374928', 'heejung0520naver.com', 'NO');
```

비밀번호 형식에 맞지 않습니다

--회원가입 완료

```
EXEC UP_MEMBERJOIN('seohj0520', '서희정', 'heej1234', '94/05/20', '01099374928', 'heejung0520naver.com', 'YES');
```

회원 가입이 완료 되었습니다

--이미 사용중인 아이디

```
EXEC UP_MEMBERJOIN('seohj0520', '서희정', 'heej1234', '94/05/20', '01099374928', 'heejung0520naver.com', 'YES');
```

seohj0520는 이미 사용중인 ID 입니다

6. PL/SQL

- CGV 화면 / 로그인

로그인

비회원로그인

아이디 비밀번호를 입력하신 후, 로그인 버튼을 클릭해 주세요.

⋈ |

🔒 |

로그인

☐ 아이디 저장 [아이디 찾기 >](#) [비밀번호 찾기 >](#)

6. PL/SQL

로그인 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MEMBER_LOGIN
(
  P_MEM_ID IN CGV_MEMBER.MEM_ID%TYPE,
  P_MEM_PWD IN CGV_MEMBER.MEM_PWD%TYPE
)
IS
  V_COUNT NUMBER:=0;
  V_MEM CGV_MEMBER.MEM_ID%TYPE;
BEGIN
  SELECT COUNT(*) INTO V_COUNT FROM CGV_MEMBER WHERE MEM_ID=P_MEM_ID AND CRYPTIT.DECRYPT(MEM_PWD,'pwd') = P_MEM_PWD;
  SELECT NVL(max(MEM_ID), null) INTO V_MEM FROM CGV_LOGIN WHERE MEM_ID=P_MEM_ID;

  IF V_COUNT > 0 AND NOT(V_MEM IS NULL) THEN
    UPDATE CGV_LOGIN SET LOGIN_STATE = 'Y' WHERE MEM_ID=P_MEM_ID;
    DBMS_OUTPUT.PUT_LINE(P_MEM_ID || '님 환영합니다.');
```

```
  ELSIF V_COUNT > 0 AND V_MEM IS NULL THEN
    INSERT INTO CGV_LOGIN(MEM_ID, LOGIN_TIME, LOGIN_STATE) VALUES (P_MEM_ID, SYSDATE, 'Y');
    DBMS_OUTPUT.PUT_LINE(P_MEM_ID || '님 환영합니다.');
```

```
  ELSE
    DBMS_OUTPUT.PUT_LINE('로그인을 실패하였습니다. 아이디와 비밀번호를 확인해주세요.');
```

```
  END IF;
END;
```

MEM_ID	LOGIN_TIME	LOGIN_STATE
1 zzang	17/05/21	Y
2 vietto	17/05/21	N
3 kenik	17/05/22	Y

MEM_ID	LOGIN_TIME	LOGIN_STATE
1 zzang	17/05/21	Y
2 vietto	17/05/21	N
3 seohj0520	17/05/31	Y
4 kenik	17/05/22	Y

--로그인 실패

```
EXEC UP_MEMBER_LOGIN ('shj0520', 'heej12');
```

로그인을 실패하였습니다. 아이디와 비밀번호를 확인해주세요.

--로그인 성공

```
EXEC UP_MEMBER_LOGIN('seohj0520', 'heej1234');
```

seohj0520님 환영합니다.

6. PL/SQL

- CGV 화면 / 로그아웃 PL/SQL

로그아웃

```
CREATE OR REPLACE PROCEDURE UP_MEMBER_LOGOUT
(
  P_MEM_ID IN CGV_MEMBER.MEM_ID%TYPE
)
IS
  V_STATE CGV_LOGIN.LOGIN_STATE%TYPE;
BEGIN
  SELECT LOGIN_STATE INTO V_STATE FROM CGV_LOGIN WHERE MEM_ID = P_MEM_ID;
  IF V_STATE = 'Y' THEN
    UPDATE CGV_LOGIN SET LOGIN_STATE='N', LOGIN_TIME = SYSDATE WHERE MEM_ID=P_MEM_ID;
    DBMS_OUTPUT.PUT_LINE('로그아웃 되었습니다');
  ELSE
    DBMS_OUTPUT.PUT_LINE('로그인을 하지 않았습니다.');
```

```
--로그아웃
EXEC UP_MEMBER_LOGOUT('seohj0520');

로그아웃 되었습니다
```

	MEM_ID	LOGIN_TIME	LOGIN_STATE
1	zzang	17/05/21	Y
2	vietto	17/05/21	N
3	seohj0520	17/05/31	N
4	kenik	17/05/22	Y

6. PL/SQL

● CGV 화면 / 비회원 로그인

STEP 2 개인정보(이름,휴대폰번호,법정생년월일,비밀번호) 입력 후 로그인

비회원 로그인 정보 오입력 시, 예매내역 확인/취소 및 티켓 발권이 어려울 수 있으니, 입력하신 정보를 다시 한번 확인해주시기 바랍니다.

비회원 로그인 입력정보	비회원 로그인 등록정보분실 안내
<p>모든 항목은 필수 입력사항입니다.</p> <p>이름 <input type="text"/></p> <p>법정생년월일 (6자리) <input type="text"/> - * * * * *</p> <p>휴대폰번호 <input type="text"/> 010 <input type="text"/> - <input type="text"/> - <input type="text"/></p> <p>비밀번호 (4자리) <input type="text"/></p> <p>비밀번호확인 <input type="text"/></p> <p>비회원 로그인</p>	<p>비회원 등록정보 찾기를 통해 비회원 등록정보를 확인하실 수 있습니다.</p> <p>비밀번호 분실 시 휴대폰 인증을 통해 확인</p> <p>휴대폰 번호 분실 시 이름 + 법정생년월일 + 비밀번호 입력 후 확인</p> <p>비회원 등록정보 찾기</p>

6. PL/SQL

비회원 로그인 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MEMBER_NON_LOGIN
(
  P_NON_NAME IN NON_MEM.NON_NAME%TYPE,
  P_NON_BRDAT IN NON_MEM.NON_BRDAT%TYPE,
  P_NON_PWD IN NON_MEM.NON_PWD%TYPE,
  P_NON_PWD_OK IN NON_MEM.NON_PWD%TYPE,
  P_NON_PH_NO IN NON_MEM.NON_PH_NO%TYPE
)
IS
BEGIN
  IF (P_NON_PWD = P_NON_PWD_OK AND LENGTH(P_NON_PWD) = 4) THEN
    INSERT INTO NON_MEM(NON_PAT, NON_NAME, NON_BRDAT, NON_PWD, NON_PH_NO)
    VALUES (CGV.FN_NON_MEM(), P_NON_NAME, P_NON_BRDAT, CryptIT.encrypt(P_NON_PWD, 'pwd'), P_NON_PH_NO);
    DBMS_OUTPUT.PUT_LINE('비회원 로그인이 되었습니다.');

```
-- 비회원테이블의 비회원코드 랜덤 발생(5자리 영숫자 혼합)
CREATE OR REPLACE FUNCTION fnNON_MEM
RETURN char
IS
 vcode NON_MEM.NON_PAT%TYPE;
BEGIN
 SELECT DBMS_RANDOM.STRING('X', 5) INTO vcode FROM DUAL;
RETURN (vcode);
END;
```


```

--비회원 로그인 실패
EXEC UP_MEMBER_NON_LOGIN('서희정', '94/05/05', '05255', '052255', '01099374928'); 비밀번호는 4자리입니다.

--비회원 로그인 실패
EXEC UP_MEMBER_NON_LOGIN('서희정', '94/05/05', '0520', '0522', '01099374928'); 비밀번호가 일치하지 않습니다.

--비회원 로그인 성공
EXEC UP_MEMBER_NON_LOGIN('서희정', '94/05/05', '0520', '0520', '01099374928'); 비회원 로그인이 되었습니다.

NO...	NON_NAME	NON_BRDAT	NON_PWD	NON_PH_NO
1 31HWA	문세진	94/12/01	*Rn	01048548025
2 KQOVA	서희정	94/05/05	—	01099374928

6. PL/SQL

- CGV 화면 / 회원 정보



서희정님 heejung0520 닉네임 : 닉네임을 설정해주세요. 

고객님은 일반 입니다.
MY 지난등급이력 보기

SPECIAL MEMBERSHIP

MY COUPON

CGV VIP에 도전하세요! [CGV VIP 혜택보기](#)

CGV 할인쿠폰

0 개

영화관람권

0 개

CJ ONE POINT +

CJ ONE 사용가능 포인트 1,388 점

VIP선정 포인트 ① 3,790 점

CGV수원 1

CGV북수원 2

3

4

5

자주가는 CGV
설정하기 

6. PL/SQL

● 회원 정보 등록 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MEM_INFOR
(
  P_MEM_ID IN CGV_MEMBER.MEM_ID%TYPE,
  P_MEM_NIC IN PROFILE.NICKNAME%TYPE,
  P_MEM_CARDNO IN POINT.CRD_NO%TYPE
)
IS
  V_COUNT NUMBER:=0;
  V_STATE CGV_LOGIN.LOGIN_STATE%TYPE;
BEGIN
  SELECT COUNT(*) INTO V_COUNT FROM PROFILE PR, POINT PO WHERE PR.MEM_ID = PO.MEM_ID AND PR.MEM_ID = P_MEM_ID;
  SELECT LOGIN_STATE INTO V_STATE FROM CGV_LOGIN WHERE MEM_ID=P_MEM_ID AND LOGIN_STATE = 'Y';

  IF V_COUNT > 0 THEN
    UPDATE PROFILE SET NICKNAME = P_MEM_NIC WHERE MEM_ID=P_MEM_ID;
    UPDATE POINT SET CRD_NO = P_MEM_CARDNO WHERE MEM_ID=P_MEM_ID;
    DBMS_OUTPUT.PUT_LINE('정보가 수정되었습니다');
  ELSE
    INSERT INTO PROFILE (MEM_ID, NICKNAME ) VALUES (P_MEM_ID, P_MEM_NIC );
    INSERT INTO POINT VALUES (P_MEM_CARDNO, P_MEM_ID, 0 );
    DBMS_OUTPUT.PUT_LINE('정보가 등록되었습니다');
  END IF;
EXCEPTION
  when NO_DATA_FOUND then
    DBMS_OUTPUT.PUT_LINE('로그인한 회원만 정보 수정이 가능합니다.');
```

--포인트 카드 번호 16자리 임시 생성 함수

```
CREATE OR REPLACE FUNCTION fn_POINT
RETURN VARCHAR2
IS rnd VARCHAR2(20);
BEGIN
  rnd:=ROUND(DBMS_RANDOM.VALUE(1000, 9999),0)||'-'||
  ROUND(DBMS_RANDOM.VALUE(1000, 9999),0)||'-'||
  ROUND(DBMS_RANDOM.VALUE(1000, 9999),0)||'-'||
  ROUND(DBMS_RANDOM.VALUE(1000, 9999),0);
  RETURN rnd;
END;
```

6. PL/SQL

● 회원 정보 등록 실행결과

--정보 등록

```
EXEC UP_MEM_INFOR('seohj0520','shj', CGV.FN_POINT());
```

MEM_ID	LOGIN_TIME	LOGIN_STATE
1 zzang	17/05/21	Y
2 vietto	17/05/21	N
3 seohj0520	17/05/31	N
4 kenik	17/05/22	Y

로그인한 회원만 정보 수정이 가능합니다.

MEM_ID	LOGIN_TIME	LOGIN_STATE
1 zzang	17/05/21	Y
2 vietto	17/05/21	N
3 seohj0520	17/05/31	Y
4 kenik	17/05/22	Y

정보가 등록되었습니다

CRD_NO	MEM_ID	AMT_PNT
1 8841-8335-9382-3731	kenik	0
2 9808-1196-2513-9447	vietto	0
3 2170-7938-9031-4791	seohj0520	0

MEM_ID	NICKNAME	PROF_IMG
1 vietto	mam	(null)
2 kenik	king	(null)
3 seohj0520	shj	(null)

--정보 수정

```
EXEC UP_MEM_INFOR('seohj0520','jong', CGV.FN_POINT());
```

정보가 수정되었습니다

MEM_ID	NICKNAME	PROF_IMG
1 vietto	mam	(null)
2 kenik	king	(null)
3 seohj0520	jong	(null)

6. PL/SQL

- 회원 정보 출력 PL/SQL

```
-- 회원정보 출력 뷰
CREATE or REPLACE VIEW VI_MEM_INFO
AS
SELECT PRO.PROF_IMG "프로필사진", C_M.MEM_NAME "이름", C_M.MEM_ID 아이디
      , NVL(PRO.NICKNAME, '닉네임을 설정해주세요') 닉네임,
      (SELECT DISTINCT G.GRA_NAME FROM POINT, GRADE G WHERE AMT_PNT BETWEEN G.LOW_PNT AND G.TOP_PNT) 등급,
      PO.AMT_PNT "포인트"
from CGV_MEMBER c_m JOIN PROFILE PRO ON PRO.MEM_ID = c_m.MEM_ID
      JOIN POINT PO ON PO.MEM_ID = c_m.MEM_ID;

SELECT * FROM VI_MEM_INFO WHERE 아이디 = 'seohj0520';
```

	프로필사진	이름	아이디	닉네임	등급	포인트
1	(null)	서희정	seohj0520	jong	일반	0

6. PL/SQL

● 할인 쿠폰 등록 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MEM_SALE
(
  P_MEM_ID CGV_MEMBER.MEM_ID%TYPE,
  P_COUNPON_CODE DISCOUNT_COUPON.COUPON_CODE%TYPE,
  P_COUNPON_NAME DISCOUNT_COUPON.COUPON_NAME%TYPE
)
IS
  V_STATE CGV_LOGIN.LOGIN_STATE%TYPE;
BEGIN
  SELECT LOGIN_STATE INTO V_STATE FROM CGV_LOGIN WHERE MEM_ID=P_MEM_ID AND LOGIN_STATE = 'Y';

  IF (LENGTH(P_COUNPON_CODE) = 5) THEN
    INSERT INTO DISCOUNT_COUPON ( MEM_ID, COUPON_CODE, COUPON_NAME)
    VALUES (P_MEM_ID, P_COUNPON_CODE, P_COUNPON_NAME);
    DBMS_OUTPUT.PUT_LINE('쿠폰을 등록하였습니다.');
```

```
--쿠폰 등록 실패
EXEC UP_MEM_SALE('seohj0520', '123456', '5%할인쿠폰');      쿠폰 번호를 확인해주세요

--쿠폰 등록
EXEC UP_MEM_SALE('seohj0520', '13438', '5%할인쿠폰');      쿠폰을 등록하였습니다.
```

	COUPON_CODE	MEM_ID	COUPON_NAME
1	12345	kenik	임시쿠폰
2	13438	seohj0520	5%할인쿠폰

6. PL/SQL

● CGV 화면 / 스낵 기프티콘 구매

주문 정보 입력

	상품	정상가	할인 적용 금액 (온라인전용)
	빅에이드콤보 팝콘(L)1+에이드(L)2 (오렌지, 자몽, 블루베리 중 택 2)	11,500원	11,000원

02 연락처를 입력해 주세요

구매하는 분

이름 **서희정** | 휴대폰 번호 010 - 9937 - 4928 * 번호 변경시에도 회원정보에 반영되지 않습니다.

사용하는 분

1	휴대폰 번호	010 - 9937 - 4928	수량	1	초기화
2	휴대폰 번호	선택하세요 - -	수량	1	초기화
3	휴대폰 번호	선택하세요 - -	수량	1	초기화
4	휴대폰 번호	선택하세요 - -	수량	1	초기화
5	휴대폰 번호	선택하세요 - -	수량	1	초기화

6. PL/SQL

● 스낵 기프티콘 구매 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_BYU_GIFTICON
(
  P_MEM_ID CGV_MEMBER.MEM_ID%TYPE,
  P_SNACK_NAME SNACKBAR.SNACK_NAME%TYPE,
  P_CELL_PHONE SNACK_BAR_GIFTICON.CELL_PHONE%TYPE,
  P_PUR_CNT SNACK_BAR_GIFTICON.PUR_CNT%TYPE
)
IS
  V_STATE CGV_LOGIN.LOGIN_STATE%TYPE;
  V_PAT_AMOUNT SNACK_BAR_GIFTICON.PAT_AMOUNT%TYPE;
  V_SNACK_CODE SNACKBAR.SNACK_CODE%TYPE;
  V_SNACK_PRICE SNACKBAR.SNACK_PRICE%TYPE;
BEGIN
  SELECT LOGIN_STATE INTO V_STATE FROM CGV_LOGIN WHERE MEM_ID=P_MEM_ID AND LOGIN_STATE = 'Y';
  SELECT SNACK_CODE INTO V_SNACK_CODE FROM SNACKBAR WHERE SNACK_NAME = P_SNACK_NAME;
  SELECT SNACK_PRICE INTO V_SNACK_PRICE FROM SNACKBAR WHERE SNACK_CODE = V_SNACK_CODE;

  INSERT INTO SNACK_BAR_GIFTICON
  VALUES (SEQ_SNACK_BAR_GIFTICON.NEXTVAL, P_MEM_ID, V_SNACK_CODE, SYSDATE, P_CELL_PHONE,
  P_PUR_CNT, V_SNACK_PRICE*P_PUR_CNT, 'N');
  DBMS_OUTPUT.PUT_LINE('구매가 완료되었습니다.');
```

```
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('로그인한 회원만 상품 구매가 가능합니다.');
```

```
END;
```

```
EXEC UP_BYU_GIFTICON('seohj0520', '오렌지환타(L)', '01099374928', '2');
```

구매가 완료되었습니다.

SEQ	MEM_ID	SNACK_CODE	PUR_CODE	CELL_PHONE	PUR_CNT	PAT_AMOUNT	USE_CHK
14	kenik	103	17/05/22	01091253653	22	242000	N
221	seohj0520	310	17/06/06	01099374928	2	5000	N

6. PL/SQL

- CGV 화면 / 무비차트

[홈](#) > [영화](#) > [무비차트](#) > [무비차트](#)

VIP LOUNGE Club서비스 포토티켓


무비차트

▶ 무비차트 상영예정작 CGV아트하우스

☐ 현재 상영작만 보기

예매순 GO

No.1




에이리언: 커버넌트

예매율 23.6% | 90%

2017.05.09 개봉

7,301 [예매](#)

No.2




보스 베이비

예매율 23.2% | 94%

2017.05.03 개봉

8,342 [예매](#)

No.3




가디언즈 오브 갤럭시 VOL. 2

예매율 17.9% | 98%


2017.05.02 개봉

15,427 [예매](#)



씨티카드 특별 혜택!

CGV 영화예매권 5매 증정



[자세히보기](#)

AD 씨타카드 이벤트

영화예매권 5매 증정

6. PL/SQL

● 무비차트 출력 PL/SQL

```
-- 무비차트 출력 뷰(기준 17/05/18)
CREATE or REPLACE VIEW VI_CHART_MOVIE
AS
select C.CHART_CODE 순위, M.MOVIE_TITLE 영화제목, fnCHART(C.MOVIE_CODE) 예매율,
M.REL_DATE 개봉날짜, C.LIKE_CNT 좋아요, C.PRESENT_SHOW "현재상영하는영화여부", M.ACC_VISITORS "관객 수"
from MOVIE M JOIN CHART C ON M.MOVIE_CODE = C.MOVIE_CODE;

SELECT * FROM VI_CHART_MOVIE;

-- 예매율 구하기 함수(누적 관람객 / 차트 테이블의 누적 관람객 SUM )
CREATE or REPLACE FUNCTION fnCHART
(
    pMovie_code CHAR
)
RETURN varchar2
IS
    ticket_sales varchar2(10);
BEGIN
    SELECT to_char(ROUND(ACC_VISITORS/
        (select SUM(M.ACC_VISITORS) from MOVIE M JOIN CHART C
            ON M.MOVIE_CODE = C.MOVIE_CODE) * 100,1), 'FM9990D99') || '%' 예매율
    INTO ticket_sales
    FROM MOVIE M JOIN CHART C ON M.MOVIE_CODE = C.MOVIE_CODE
    WHERE M.MOVIE_CODE like pMovie_code ;

RETURN (ticket_sales);
END;
```

순위	영화제목	예매율	개봉날짜	좋아요	현재상영하는영화여부	관객 수
1	보안관	16.7%	17/05/03	6937	현재상영중	2271025
2	에이리언: 커버넌트	7.5%	17/05/09	8216	현재상영중	1018734
3	가디언즈 오브 갤럭시 VOL. 2	18.1%	17/05/03	16287	현재상영중	2463653
4	보스 베이비	14.4%	17/05/03	9415	현재상영중	1959882
5	석조저택 살인사건	2.3%	17/05/09	4761	현재상영중	313530
6	목소리의 형태	1.6%	17/05/09	6281	현재상영중	214322
7	임금님의 사건수첩	11.9%	17/04/26	7630	현재상영중	1619325
8	킹 아서: 제왕의 검	0.4%	17/05/17	4778	현재상영중	54177
9	분노의 질주: 더 엑스트림	26.9%	17/04/12	12769	현재상영중	3648620
10	불한당: 나쁜 놈들의 세상	0.2%	17/05/17	4928	현재상영중	21444

6. PL/SQL

- CGV 화면 / 영화 상세

영화상세



보스 베이비

현재상영중

(Sub)The Boss Baby

예매율 19.6% | 🏆 94% (실관람객 : 17,673명)

감독 : 톰 맥그래스 / 프로듀서 : 데니스 놀런 카쉬노, 램지 앤 나이트 / 배우 : 스티브 부세미, 알렉 볼드윈, 리사 쿠드로, 토비 맥과이어
장르 : 애니메이션, 코미디, 가족 / 기본 : 전체, 097분, 미국
개봉 : 2017.05.03

3D



8,376



예매

예매 이벤트

📄 주요정보 | 🎬 트레일러 | 🖼️ 스틸컷 | ⭐ 평점/리뷰 | 🕒 상영시간표

12세 이상 관람가

3면에서 펼쳐지는
판타지 액션의 신세계!

6. PL/SQL

● 영화 상세 출력 PL/SQL

```
-- 영화 상세 출력 뷰  
CREATE OR REPLACE VIEW VI_MOV_CHA_GRA  
( 영화코드, 영화명, 등급코드, 장르명, 개봉날짜, 누적관람객, 주요내용, 감독, 출연진, 차트코드,  
  좋아요수, 현재상영여부, 등급명, 등급기준 )  
AS  
SELECT M.MOVIE_CODE, MOVIE_TITLE, M.STD_CODE, GENRE_NAME, REL_DATE, ACC_VISITORS, MAIN_CON, PRES, ACTOR,  
       C.CHART_CODE, LIKE_CNT, PRESENT_SHOW, STD_NAME, STD_STD  
FROM MOVIE M JOIN CHART C ON M.MOVIE_CODE=C.MOVIE_CODE  
  JOIN STANDARD S ON M.STD_CODE= S.STD_CODE;  
  
SELECT * FROM VI_MOV_CHA_GRA where 영화코드 = '20174061';
```

영화코드	영화명	등급코드	장르명	개봉날짜	누적관람객
1 20174061	보스 베이비	전체	애니메이션, 코미디, 가족	17/05/03	1959882

주요내용
시크한 눈빛, 짝 빠진 수트핏 귀여운 얼굴 뒤 숨겨진 반전포스 세상에 없던 베이비가 온다!

감독	출연진	차트코드	좋아요수
툼 맥그라스	알렉 볼드윈, 리사 쿠드로, 스티브 부세미, 토비 맥과이어, 지미 키멜	4	9415

현재상영여부	등급명	등급기준
현재상영중	전체 관람가	모든 연령의 고객님께서 관람 가능

6. PL/SQL

- CGV 화면 / 빠른 예매

[ENGLISH](#)
[예매 가이드](#)
[제휴 할인혜택](#)
[상영시간표](#)
[예매 다시하기](#)

영화	극장	날짜	시간
<div>전체 아드하투스 특별관</div> <div>예매 중인 가나다순</div> <div>2차 보스베이비</div> <div>12 가디언즈 오브 갤럭시 VOL. 2</div> <div>15 보안관</div> <div>2차 목소리의 달래</div> <div>15 석조지옥살인사건</div> <div>12 외계인의 사건 수첩</div> <div>15 8마일</div> <div>2차 스머프-비밀의 숲</div> <div>2차 극장판 토트-로봇군단의 습격</div> <div>15 분노의 질주-더 엑스트라</div> <div>2차 슈퍼헤븐-스파이대작전</div> <div>2차 서서벌, 천천히 걸쎄하게</div> <div>15 세일즈맨</div>	<div>전체 아드하투스 특별관</div> <div>서둘(26) 부원익</div> <div>걸기(34) 복수권</div> <div>인천(9) 상봉</div> <div>강진(4) 서현</div> <div>대전/충청(16) 소들</div> <div>대구(9) 수권</div> <div>부산/동산(10) 시흥</div> <div>경남(13) 안산</div> <div>광주/전라/제주(16) 야탑</div> <div>익곡</div> <div>수원</div>	<div>2017 5</div> <div>화 9</div> <div>수 10</div> <div>목 11</div> <div>금 12</div> <div>토 13</div> <div>일 14</div> <div>월 15</div> <div>화 16</div> <div>수 17</div> <div>목 18</div> <div>금 19</div> <div>토 20</div> <div>일 21</div>	<div>조조 심야</div> <div>2D(더빙) 6관 (총 234석)</div> <div>14:55 6석</div> <div>2D(자막) 6관 [수원여자대학교] (총 267석)</div> <div>15:20 6석</div> <div>17:20 39석</div> <div>19:20 126석</div> <div>21:25 260석</div> <div>23:30 266석</div>

보스베이비
 2D(자막)
 전체 관람가

극장 CGV 수원 >

일시 2017.6.9(화) 19:20

상영관 6관 [수원여자대학교]

인원

[좌석선택](#)
[결제](#)

좌석선택

6. PL/SQL

- 빠른 예매(상영하는 영화 출력) PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MOVIE_FAST
IS
BEGIN
FOR I IN (
    SELECT DISTINCT M.STD_CODE , MOVIE_TITLE
    FROM STANDARD ST
    JOIN MOVIE M
    ON ST.STD_CODE = M.STD_CODE
    JOIN SCREENING_SCHEDULE S
    ON M.MOVIE_CODE = S.MOVIE_CODE
    JOIN THEATER T
    ON S.THEA_CODE = T.THEA_CODE
    JOIN CGV_CINEMA C
    ON T.CINEMA_CODE = C.CINEMA_CODE
    JOIN AREA A
    ON C.AREA_CODE = A.AREA_CODE
)
LOOP
    DBMS_OUTPUT.PUT_LINE( TRIM(I.STD_CODE) || ' : ' || I.MOVIE_TITLE );
END LOOP;
END;

EXEC UP_MOVIE_FAST();
```

전체 : 보스 베이비
전체 : 스머프: 비밀의 숲
전체 : 목소리의 형태
15 : 석조저택 살인사건
청불 : 8 마일
전체 : 극장판 토봇: 로봇군단의 습격
12 : 킹 아서: 제왕의 검
청불 : 불한당: 나쁜 놈들의 세상
전체 : 청불복
15 : 에이리언: 커버넌트
15 : 보안관
12 : 가디언즈 오브 갤럭시 VOL. 2

6. PL/SQL

- 빠른 예매 (지역 선택 후 선택한 지역의 극장목록만을 출력) PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MOVIE_INFO_SEARCH
(
    p_SEARCH_AREA IN AREA.AREA_CODE%TYPE
)
IS
BEGIN
    FOR I IN
    (
        SELECT DISTINCT AREA_NAME, STORE_NAME
        FROM AREA A
        JOIN CGV_CINEMA C
        ON A.AREA_CODE = C.AREA_CODE
        WHERE A.AREA_CODE = p_SEARCH_AREA
    )
    LOOP
        DBMS_OUTPUT.PUT_LINE(I.AREA_NAME || ' > ' || I.STORE_NAME );
    END LOOP;
END;

-- 서울 극장 목록 출력
EXEC UP_MOVIE_INFO_SEARCH('10');
```

서	>	CGV	상명	북동	
서	>	CGV	명강	동남	
서	>	CGV	하계	구정	
서	>	CGV	압신	아트레온	
서	>	CGV	영등	강포	
서	>	CGV	불광	동강	
서	>	CGV	강상	암로	
서	>	CGV	구여	의도	
서	>	CGV	성신	신대입구	
서	>	CGV	왕대	동역	
서	>	CGV	홍명	프린	
서	>	CGV	대학	신도림	
서	>	CGV	근로	자문	
서	>	CGV	목동	파계	
서	>	CGV	송파	계담	
서	>	CGV	충청	씨네시티	
서	>	CGV	청포	카디리	1958
서	>	CGV	천호	미아	
서	>	CGV	강변		

6. PL/SQL

- 빠른 예매 (원하는 지역, 지점 명 입력해서 상영 시간표 출력) PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MOVIE_INFO2_SEARCH
(
  p_SEARCH_AREA IN AREA.AREA_CODE%TYPE
, p_SEARCH_THEA IN CGV_CINEMA.STORE_NAME%TYPE
)
IS
BEGIN
  FOR I IN
  (
    SELECT *
    FROM STANDARD ST
    JOIN MOVIE M
    ON ST.STD_CODE = M.STD_CODE
    JOIN SCREENING_SCHEDULE S
    ON M.MOVIE_CODE = S.MOVIE_CODE
    JOIN THEATER T
    ON S.THEA_CODE = T.THEA_CODE
    JOIN CGV_CINEMA C
    ON T.CINEMA_CODE = C.CINEMA_CODE
    JOIN AREA A
    ON C.AREA_CODE = A.AREA_CODE
    WHERE A.AREA_CODE = p_SEARCH_AREA AND STORE_NAME = p_SEARCH_THEA
  )
  LOOP
    DBMS_OUTPUT.PUT_LINE(I.MOVIE_TITLE || ' > ' || I.STORE_NAME || ' '
      || I.THEA_NAME || ' ' || I.TOTAL_SEAT || '석'
      || I.GENRE_NAME || ' ' || I.ACC_VISITORS || '명' || I.SCHED_DATE || ' '
      || I.SCHED_TIME || ' ' || I.RUN_TIME || '분' || I.STD_NAME );
  END LOOP;
END;
```

6. PL/SQL

- 빠른 예매 (원하는 지역, 지점 명 입력해서 상영 시간표 출력) 실행결과

```
-- 서울 강남지점 상영시간표 출력
```

```
EXEC UP_MOVIE_INFO2_SEARCH('10', 'CGV 강남');
```

보스 베이비	>	CGV	강남	1관	158석	애니메이션, 코미디, 가족	1959882명	17/05/18	02:35	97분	전체	관람가
보스 베이비	>	CGV	강남	1관	158석	애니메이션, 코미디, 가족	1959882명	17/05/18	07:40	97분	전체	관람가
보스 베이비	>	CGV	강남	1관	158석	애니메이션, 코미디, 가족	1959882명	17/05/19	03:35	97분	전체	관람가
보스 베이비	>	CGV	강남	2관	124석	애니메이션, 코미디, 가족	1959882명	17/05/18	08:30	97분	전체	관람가
보스 베이비	>	CGV	강남	2관	124석	애니메이션, 코미디, 가족	1959882명	17/05/18	10:35	97분	전체	관람가
보스 베이비	>	CGV	강남	2관	124석	애니메이션, 코미디, 가족	1959882명	17/05/18	12:40	97분	전체	관람가
보스 베이비	>	CGV	강남	2관	124석	애니메이션, 코미디, 가족	1959882명	17/05/18	05:10	97분	전체	관람가
보스 베이비	>	CGV	강남	2관	124석	애니메이션, 코미디, 가족	1959882명	17/05/19	12:30	97분	전체	관람가
보스 베이비	>	CGV	강남	1관	158석	애니메이션, 코미디, 가족	1959882명	17/05/19	08:30	97분	전체	관람가
보스 베이비	>	CGV	강남	1관	158석	애니메이션, 코미디, 가족	1959882명	17/05/19	10:45	97분	전체	관람가
보스 베이비	>	CGV	강남	1관	158석	애니메이션, 코미디, 가족	1959882명	17/05/19	03:35	97분	전체	관람가
보스 베이비	>	CGV	강남	1관	158석	애니메이션, 코미디, 가족	1959882명	17/05/19	05:40	97분	전체	관람가
보스 베이비	>	CGV	강남	1관	158석	애니메이션, 코미디, 가족	1959882명	17/05/19	07:45	97분	전체	관람가
보스 베이비	>	CGV	강남	1관	158석	애니메이션, 코미디, 가족	1959882명	17/05/20	12:20	97분	전체	관람가
보스 베이비	>	CGV	강남	3관	172석	애니메이션, 코미디, 가족	1959882명	17/05/20	02:25	97분	전체	관람가
킹어스 오브 더 힐	>	CGV	강남	4관	124석	판타지, 액션, 어드벤처	54177명	17/05/18	03:05	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	4관	124석	판타지, 액션, 어드벤처	54177명	17/05/18	05:50	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	4관	124석	판타지, 액션, 어드벤처	54177명	17/05/18	08:25	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	1관	158석	판타지, 액션, 어드벤처	54177명	17/05/18	04:55	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	1관	158석	판타지, 액션, 어드벤처	54177명	17/05/19	12:50	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	2관	124석	판타지, 액션, 어드벤처	54177명	17/05/18	07:15	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	4관	124석	판타지, 액션, 어드벤처	54177명	17/05/18	09:40	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	4관	124석	판타지, 액션, 어드벤처	54177명	17/05/18	11:00	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	5관	172석	판타지, 액션, 어드벤처	54177명	17/05/19	03:50	126분	12세 이상	관람가
가디언즈 오브 갤럭시	>	CGV	강남	1관	158석	액션, SF	2463653명	17/05/18	09:20	136분	12세 이상	관람가
가디언즈 오브 갤럭시	>	CGV	강남	1관	158석	액션, SF	2463653명	17/05/18	09:45	136분	12세 이상	관람가
가디언즈 오브 갤럭시	>	CGV	강남	2관	124석	액션, SF	2463653명	17/05/19	02:40	136분	12세 이상	관람가
가디언즈 오브 갤럭시	>	CGV	강남	6관	124석	액션, SF	2463653명	17/05/18	06:05	136분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	4관	124석	판타지, 액션, 어드벤처	54177명	17/05/19	06:00	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	4관	124석	판타지, 액션, 어드벤처	54177명	17/05/19	08:40	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	2관	124석	판타지, 액션, 어드벤처	54177명	17/05/19	04:50	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	2관	124석	판타지, 액션, 어드벤처	54177명	17/05/19	07:25	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	2관	124석	판타지, 액션, 어드벤처	54177명	17/05/20	01:10	126분	12세 이상	관람가
킹어스 오브 the 힐	>	CGV	강남	4관	124석	판타지, 액션, 어드벤처	54177명	17/05/19	10:00	126분	12세 이상	관람가

- CGV 화면 / 빠른 예매(선택 가능 좌석 출력)



6. PL/SQL

- 빠른 예매(선택 가능 좌석 출력) PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MOVIE_INFO_SEARCH_SEAT
(
  P_SEARCH_AREA      AREA.AREA_CODE%TYPE
  , P_SEARCH_THEA     CGV_CINEMA.STORE_NAME%TYPE
  , P_MOVIE_TITLE     MOVIE.MOVIE_TITLE%TYPE
  , P_THEA_NAME       THEATER.THEA_NAME%TYPE
  , P_SCHED_DATE      SCREENING_SCHEDULE.SCHED_DATE%TYPE
)
IS
BEGIN
  FOR I IN
  (
    SELECT *
    FROM SEAT_CODE SC
    JOIN SEAT_S
    ON SC.SEAT_CODE = S.SEAT_CODE
    JOIN THEATER T
    ON S.THEA_CODE = T.THEA_CODE
    JOIN CGV_CINEMA C
    ON T.CINEMA_CODE = C.CINEMA_CODE
    JOIN AREA A
    ON C.AREA_CODE = A.AREA_CODE
    JOIN SCREENING_SCHEDULE SS
    ON T.THEA_CODE = SS.THEA_CODE
    JOIN MOVIE M
    ON SS.MOVIE_CODE = M.MOVIE_CODE
    JOIN STANDARD STD
    ON M.STD_CODE = STD.STD_CODE
    WHERE A.AREA_CODE = P_SEARCH_AREA
    AND C.STORE_NAME = P_SEARCH_THEA
    AND M.MOVIE_TITLE = P_MOVIE_TITLE
    AND T.THEA_NAME = P_THEA_NAME
    AND S.C_SEAT_CODE NOT IN (SELECT C_SEAT_CODE FROM RESERVATION_SEAT)
    AND SS.SCHED_DATE = P_SCHED_DATE
    ORDER BY SCHED_TIME ASC, SEAT_NO ASC
  )
  LOOP
    DBMS_OUTPUT.PUT_LINE(I.THEA_NAME || ' ' || I.SEAT_NO || ' > ' || I.SEAT_NAME || ' '
      || I.MOVIE_TITLE || ' ' || I.SCHED_DATE || ' ' || I.SCHED_TIME || ' '
      || I.STD_NAME);
  END LOOP;
END;
```

6. PL/SQL

- 빠른 예매(선택 가능 좌석 출력) 실행결과

8관	A1	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A10	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A11	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A12	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A13	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A14	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A15	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A16	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A17	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A18	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A19	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A2	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A20	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A21	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A3	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A4	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A5	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A6	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A7	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A8	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	A9	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B1	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B10	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B11	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B12	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B13	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B14	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B15	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B16	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B17	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B18	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B19	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B2	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B20	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B21	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가
8관	B3	>	ECONOMY	ZONE	보스	베이비	17/05/18	01:30	전체	관람가

6. PL/SQL

● 회원예매 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_RESERVATION
(
    P_RES_NO IN RESERVATION.RES_NO%TYPE
  , P_MEM_ID IN CGV_MEMBER.MEM_ID%TYPE
  , P_SEARCH_AREA IN AREA.AREA_CODE%TYPE
  , P_SEARCH_THEA IN CGV_CINEMA.STORE_NAME%TYPE
  , P_MOVIE_TITLE IN MOVIE.MOVIE_TITLE%TYPE
  , P_THEA_NAME IN THEATER.THEA_NAME%TYPE
  , P_SCHED_DATE IN SCREENING_SCHEDULE.SCHED_DATE%TYPE
  , P_SCHED_TIME IN SCREENING_SCHEDULE.SCHED_TIME%TYPE
  , P_RES_PEO IN RESERVATION.RES_PRI%TYPE
  , P_SEAT_NO IN SEAT.SEAT_NO%TYPE
)
IS
    V_SEETIME_CODE SCREENING_SCHEDULE.SCHED_CODE%TYPE
    := CGV_FN_SCHED_CODE(P_SEARCH_AREA, P_SEARCH_THEA, P_MOVIE_TITLE, P_THEA_NAME, P_SCHED_DATE, P_SCHED_TIME);
    V_COUNT NUMBER:=0;
    V_STATE CGV_LOGIN.LOGIN_STATE%TYPE;
    V_MOVIE_CODE MOVIE.MOVIE_CODE%TYPE;
    V_C_SEAT_CODE SEAT.C_SEAT_CODE%TYPE;
    V_VIDEO_TEC SCREENING_SCHEDULE.VIDEO_TEC%TYPE;
    V_RES_PRI RESERVATION.RES_PRI%TYPE;
    V_RES_C_SEAT_CODE NUMBER:=0;
BEGIN
    SELECT COUNT(*) INTO V_COUNT FROM RESERVATION WHERE RES_NO = P_RES_NO ;
    SELECT NVL(max(LOGIN_STATE), null) INTO V_STATE FROM CGV_LOGIN WHERE MEM_ID=P_MEM_ID AND LOGIN_STATE = 'Y';
    --영화 코드 가져오기
    SELECT NVL(max(MOVIE_CODE), null) INTO V_MOVIE_CODE FROM MOVIE WHERE MOVIE_TITLE = P_MOVIE_TITLE;
    --상영관 좌석코드 가져오기
    SELECT C_SEAT_CODE INTO V_C_SEAT_CODE FROM SEAT S JOIN THEATER T ON S.THEA_CODE = T.THEA_CODE
    JOIN CGV_CINEMA CC ON T.CINEMA_CODE = CC.CINEMA_CODE
    WHERE STORE_NAME = P_SEARCH_THEA AND T.THEA_NAME = P_THEA_NAME AND SEAT_NO = P_SEAT_NO;
    --영상기술 가져오기
    SELECT DISTINCT NVL(VIDEO_TEC, '2D') INTO V_VIDEO_TEC FROM SCREENING_SCHEDULE WHERE SCHED_CODE = V_SEETIME_CODE;
    --가격 가져오기
    SELECT DISTINCT P_C.ADT_PRI INTO V_RES_PRI FROM PRI_CLASS P_C JOIN SEAT S ON S.SEAT_CODE = P_C.SEAT_CODE
    WHERE S.C_SEAT_CODE = V_C_SEAT_CODE AND THEA_KND = V_VIDEO_TEC;
    --예매된 좌석인지 가져오기
    SELECT COUNT(*) INTO V_RES_C_SEAT_CODE FROM RESERVATION_SEAT WHERE C_SEAT_CODE = V_C_SEAT_CODE;

    IF V_COUNT != 0 OR V_STATE <> 'Y' OR V_MOVIE_CODE IS NULL OR V_C_SEAT_CODE IS NULL THEN
        DBMS_OUTPUT.PUT_LINE('예매를 할 수 없습니다.');
```

```
    ELSE
        IF(V_RES_C_SEAT_CODE = 0) THEN
            INSERT INTO RESERVATION VALUES (P_RES_NO, V_SEETIME_CODE, P_MEM_ID, V_MOVIE_CODE, V_C_SEAT_CODE
            , SYSDATE, P_RES_PEO, V_RES_PRI);
            INSERT INTO RESERVATION_SEAT VALUES (P_RES_NO, V_C_SEAT_CODE, P_MEM_ID, NULL);
            COMMIT;
            DBMS_OUTPUT.PUT_LINE('예매가 완료되었습니다. 결제 창으로 이동합니다.');
```

```
        ELSE
            DBMS_OUTPUT.PUT_LINE('이미 예매된 좌석입니다.');
```

```
        END IF;
    END IF;
END;
```

6. PL/SQL

● 회원예매 실행결과

--로그인한 회원이 아닐 경우 예매 불가능

```
EXEC UP_RESERVATION(CGV.FN_RES_NO(), 'shj0520', '41', 'CGV 수원', '보스 베이비', '8관', '17/05/18', '11:30', '1', 'J8');
```

예매를 할 수 없습니다

--이미 예매된 좌석 예매 불가능

```
EXEC UP_RESERVATION(CGV.FN_RES_NO(), 'seohj0520', '41', 'CGV 수원', '보스 베이비', '8관', '17/05/18', '11:30', '1', 'A1');
```

예매를 할 수 없습니다

RES_NO	C_SEAT_CODE	MEM_ID	NON_PAT
1 7021-1970-3363-8562	200608G4	kenik	(null)
2 7207-3241-8396-7125	200608I7	zzang	(null)
3 8586-2618-2873-5435	200608A1	kenik	(null)

--예매하기

```
EXEC UP_RESERVATION(CGV.FN_RES_NO(), 'seohj0520', '41', 'CGV 수원', '보스 베이비', '8관', '17/05/18', '11:30', '1', 'H5');
```

예매가 완료되었습니다. 결제 창으로 이동합니다.

RES_NO	SEETIME_CODE	MEM_ID	MOVIE_CODE	C_SEAT_CODE	RES_DAT	RES_PEO	RES_PRI
1 7625-4145-5815-7119	200608_20174061_0518_02	seohj0520	20174061	200608H5	17/06/07	1	10000
2 7207-3241-8396-7125	200608_20174061_0518_02	zzang	20174061	200608I7	17/05/22	1	10000
3 3262-3349-9856-3998	200608_20174061_0518_02	kenik	20174061	200608H4	17/05/22	1	9000
4 7021-1970-3363-8562	200608_20174061_0518_02	kenik	20174061	200608G4	17/05/22	1	9000
5 8586-2618-2873-5435	200608_20174061_0518_02	kenik	20174061	200608A1	17/06/07	1	8000

6. PL/SQL

- 회원예매 시 필요한 함수 (지역, 지점, 영화제목, 상영관, 날짜, 시간을 입력하면 상영시간표 코드 반환 함수)

```
CREATE OR REPLACE FUNCTION FN_SCHED_CODE
(
  P_SEARCH_AREA      AREA.AREA_CODE%TYPE
, P_SEARCH_THEA      CGV_CINEMA.STORE_NAME%TYPE
, P_MOVIE_TITLE       MOVIE.MOVIE_TITLE%TYPE
, P_THEA_NAME         THEATER.THEA_NAME%TYPE
, P_SCHED_DATE        SCREENING_SCHEDULE.SCHED_DATE%TYPE
, P_SCHED_TIME        SCREENING_SCHEDULE.SCHED_TIME%TYPE
)
RETURN VARCHAR2
IS F_SCHED_CODE SCREENING_SCHEDULE.SCHED_CODE%TYPE;
BEGIN
```

```
  SELECT DISTINCT SCHED_CODE
  INTO F_SCHED_CODE
  FROM SEAT_CODE SC JOIN SEAT S
    ON SC.SEAT_CODE = S.SEAT_CODE
  JOIN THEATER T
    ON S.THEA_CODE = T.THEA_CODE
  JOIN CGV_CINEMA C
    ON T.CINEMA_CODE = C.CINEMA_CODE
  JOIN AREA A
    ON C.AREA_CODE = A.AREA_CODE
  JOIN SCREENING_SCHEDULE SS
    ON T.THEA_CODE = SS.THEA_CODE
  JOIN MOVIE M
    ON SS.MOVIE_CODE = M.MOVIE_CODE
  JOIN STANDARD STD
    ON M.STD_CODE = STD.STD_CODE
  WHERE A.AREA_CODE = P_SEARCH_AREA
    AND C.STORE_NAME = P_SEARCH_THEA
    AND M.MOVIE_TITLE = P_MOVIE_TITLE
    AND T.THEA_NAME = P_THEA_NAME
    AND SS.SCHED_DATE = P_SCHED_DATE
    AND SS.SCHED_TIME = P_SCHED_TIME;

RETURN F_SCHED_CODE;
END;
```

```
SELECT FN_SCHED_CODE('41', 'CGV 수원', '보스 베이비',
                     '8관', '17/05/18', '11:30') 상영시간표
FROM DUAL;
```

	상영시간표
1	200608_20174061_0518_02

6. PL/SQL

- 회원예매 시 필요한 함수 (임시 예매번호 랜덤 발생 함수)

```
CREATE OR REPLACE FUNCTION FN_RES_NO
RETURN VARCHAR2
IS RND_RES VARCHAR2(20);
BEGIN
    RND_RES:=ROUND(DBMS_RANDOM.VALUE(1000, 9999),0)||'-'||
        ROUND(DBMS_RANDOM.VALUE(1000, 9999),0)||'-'||
        ROUND(DBMS_RANDOM.VALUE(1000, 9999),0)||'-'||
        ROUND(DBMS_RANDOM.VALUE(1000, 9999),0);
RETURN RND_RES;
END;
```

```
SELECT CGV.FN_RES_NO() FROM DUAL;
```

CGV.FN_RES_NO()
1 1409-8301-7495-3277

6. PL/SQL

● CGV 화면 / 회원결제

기타 포인트

↑

선택 불가능한 기타 포인트 항목은 비활성화 처리됩니다.

현대카드 M포인트 >

위비꿀머니 >

OK캐쉬백 >

문화상품권 컬처캐쉬 >

하나머니 >

현대오일뱅크 보너스포인트 >

아시아나마일리지 >

S-Oil포인트 >

삼성 U-Point >

KIA RED MEMBERS 포인트 >

E1 오렌지포인트 >

CJ카드적립금 >

● 현대카드 M포인트

카드번호 - - -

유효기간 월 년

비밀번호 * *

법정생년월일 (6자리) - * * * * * * * *

조회

사용가능한 포인트 0 원 적용매수

선택

▼

할인금액: 0 원

이용안내

매주 금요일/토요일 5,000M포인트 사용, 그 외 요일은 2,000M포인트 사용
조조, 청 소년, 우대 및 특별관(IMAX, 4DX, GOLDCLASS 등) 중복적용 불가

6. PL/SQL

● 회원결제 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MEMBER_PAT
(
    PRES_NO IN MEMBER_PAT.RES_NO%TYPE,
    PMEM_ID IN MEMBER_PAT.MEM_ID%TYPE,
    PMET_PAT_NAME IN MET_PAT.MET_PAT_NAME%TYPE,

    pCARD_NO IN CARD_PAT.CARD_NO%TYPE, -- 카드
    pINS IN CARD_PAT.INS%TYPE,
    pPWD IN CARD_PAT.PWD%TYPE,
    pEXDATE IN CARD_PAT.EXDATE%TYPE,
    pBIRTHDATE IN CARD_PAT.BIRTHDATE%TYPE,
    pCARD_NAME IN CARD_PAT.CARD_NAME%TYPE,
    pPHONE_NO IN PHONE_PAT.PHONE_NO%TYPE, --휴대폰, 카카오
    pBRTDAT IN PHONE_PAT.BRTDAT%TYPE,
    pID_PWD IN PAYCO.ID_PWD%TYPE --페이코
)
IS
    VCOUNT NUMBER := 0;
    VPATPRICE MEMBER_PAT.PATPRICE%TYPE;
    VMET_PAT_CODE MET_PAT.MET_PAT_CODE%TYPE;
BEGIN
    SELECT COUNT(*) INTO VCOUNT FROM MEMBER_PAT WHERE RES_NO = PRES_NO;
    SELECT NVL(R.RES_PRI, R.RES_PEO*9000) INTO VPATPRICE FROM RESERVATION R
    WHERE R.MEM_ID = PMEM_ID AND R.RES_NO = PRES_NO;
    SELECT MET_PAT_CODE INTO VMET_PAT_CODE FROM MET_PAT
    WHERE MET_PAT_NAME LIKE '%' || PMET_PAT_NAME || '%';
    IF (VCOUNT = 0) THEN
        INSERT INTO MEMBER_PAT VALUES
        (seq_MEMBER_PAT.NEXTVAL, PRES_NO, PMEM_ID, VMET_PAT_CODE, SYSDATE, VPATPRICE);
        IF (VMET_PAT_CODE = '101') THEN
            INSERT INTO CARD_PAT VALUES
            (SEQ_CARD_PAT.NEXTVAL, (SELECT MAX(PAT_CODE) FROM MEMBER_PAT),
            PCARD_NO, PINS, PPWD, PEXDATE, PBIRTHDATE, PCARD_NAME);
            DBMS_OUTPUT.PUT_LINE('카드결제가 완료되었습니다');
        ELSIF (VMET_PAT_CODE = '201') THEN
            INSERT INTO PHONE_PAT VALUES
            (SEQ_PHONE_PAT.NEXTVAL, PPHONE_NO, PBRTDAT, (SELECT MAX(PAT_CODE) FROM MEMBER_PAT));
            DBMS_OUTPUT.PUT_LINE('휴대폰 결제가 완료되었습니다');
        ELSIF (VMET_PAT_CODE = '301') THEN
            INSERT INTO KAKAOPAY VALUES
            (SEQ_KAKAOPAY.NEXTVAL, PPHONE_NO, PBRTDAT, (SELECT MAX(PAT_CODE) FROM MEMBER_PAT));
            DBMS_OUTPUT.PUT_LINE('카카오페이 결제가 완료되었습니다');
        ELSIF (VMET_PAT_CODE = '401') THEN
            INSERT INTO PAYCO VALUES
            (SEQ_PAYCO.NEXTVAL, (SELECT MAX(PAT_CODE) FROM MEMBER_PAT), PID_PWD);
            DBMS_OUTPUT.PUT_LINE('페이코 결제가 완료되었습니다');
        ELSE
            DBMS_OUTPUT.PUT_LINE('잘못된 결제코드입니다');
        END IF;
    ELSE
        DBMS_OUTPUT.PUT_LINE('이미 결제가 되었습니다');
    END IF;
    COMMIT;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('결제를 할 수 없습니다');
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('결제코드를 확인해주세요');
    END;
```

6. PL/SQL

● 회원결제 실행결과

--에매코드가 없을 시 결제 불가능

```
EXEC UP_MEMBER_PAT ('', 'seohj0520', '카드', '0123-4567-8901-2345', 0, '1234', '1709', '00/01/01', '신한카드', NULL, NULL, NULL);
```

결제를 할 수 없습니다

--결제 가능

```
EXEC UP_MEMBER_PAT ('7625-4145-5815-7119', 'seohj0520', '카드', '0123-4567-8901-2345', 0, '1234', '1709', '00/01/01', '신한카드', NULL, NULL, NULL);
```

카드결제가 완료되었습니다

PAT_CODE	RES_NO	MEM_ID	MET_PAT_CODE	PATDATE	PATPRICE
1 2	7207-3241-8396-7125	zzang	101	17/05/22	10000
2 4	3262-3349-9856-3998	kenik	101	17/05/22	9000
3 9	7625-4145-5815-7119	seohj0520	101	17/06/07	10000

PAT_CARD_CODE	PAT_CODE	CARD_NO	INS	PWD	EXDATE	BIRTHDATE	CARD_NAME
1 4	4	0123-4567-8901-2345	0	1234	1709	00/01/01	신한카드
2 21	9	0123-4567-8901-2345	0	1234	1709	00/01/01	신한카드

--이미 결제를 한 경우 결제 불가능

```
EXEC UP_MEMBER_PAT ('7625-4145-5815-7119', 'seohj0520', '휴대폰', NULL, NULL, NULL, NULL, NULL, NULL, '01099374928', '94/05/20', NULL);
```

이미 결제가 되었습니다

6. PL/SQL

● 비회원 예매 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_NOMEM_RESERVATION
```

```
(  
    P_RES_NO IN RESERVATION.RES_NO%TYPE  
    --비회원 로그인  
    , P_NON_NAME IN NON_MEM.NON_NAME%TYPE  
    , P_NON_BRDAT IN NON_MEM.NON_BRDAT%TYPE  
    , P_NON_PWD IN NON_MEM.NON_PWD%TYPE  
    , P_NON_PWD_OK IN NON_MEM.NON_PWD%TYPE  
    , P_NON_PH_NO IN NON_MEM.NON_PH_NO%TYPE  
  
    , P_SEARCH_AREA IN AREA.AREA_CODE%TYPE  
    , P_SEARCH_THEA IN CGV_CINEMA.STORE_NAME%TYPE  
    , P_MOVIE_TITLE IN MOVIE.MOVIE_TITLE%TYPE  
    , P_THEA_NAME IN THEATER.THEA_NAME%TYPE  
    , P_SCHED_DATE IN SCREENING_SCHEDULE.SCHED_DATE%TYPE  
    , P_SCHED_TIME IN SCREENING_SCHEDULE.SCHED_TIME%TYPE  
    , P_RES_PEO IN RESERVATION.RES_PRI%TYPE  
    , P_SEAT_NO IN SEAT.SEAT_NO%TYPE  
)
```

```
IS
```

```
V_SEETIME_CODE SCREENING_SCHEDULE.SCHED_CODE%TYPE  
:= CGV.FN_SCHED_CODE(P_SEARCH_AREA, P_SEARCH_THEA, P_MOVIE_TITLE, P_THEA_NAME, P_SCHED_DATE, P_SCHED_TIME);  
V_COUNT NUMBER:=0;  
V_NON_PAT NON_MEM.NON_PAT%TYPE;  
V_MOVIE_CODE MOVIE.MOVIE_CODE%TYPE;  
V_C_SEAT_CODE SEAT.C_SEAT_CODE%TYPE;  
V_VIDEO_TEC SCREENING_SCHEDULE.VIDEO_TEC%TYPE;  
V_RES_PRI RESERVATION.RES_PRI%TYPE;  
V_RES_C_SEAT_CODE NUMBER:=0;  
FK_EXIST EXCEPTION;  
PRAGMA EXCEPTION_INIT(FK_exist,-2291);
```

6. PL/SQL

● 비회원 예매 PL/SQL

```
BEGIN
SELECT COUNT(*) INTO V_COUNT FROM RESERVATION WHERE RES_NO = P_RES_NO ;
IF (P_NON_PWD = P_NON_PWD_OK AND LENGTH(P_NON_PWD) = 4) THEN
    SELECT NVL(MAX(NON_PAT), NULL) INTO V_NON_PAT FROM NON_MEM WHERE NON_NAME=P_NON_NAME
    AND NON_BRTDAT = P_NON_BRTDAT AND CRYPTIT.DECRYPT(NON_PWD, 'pwd') = P_NON_PWD AND NON_PH_NO = P_NON_PH_NO ;

    --영화 코드 가져오기
    SELECT NVL(MAX(MOVIE_CODE), NULL) INTO V_MOVIE_CODE FROM MOVIE WHERE MOVIE_TITLE = P_MOVIE_TITLE;
    --상영관 좌석코드 가져오기
    SELECT C_SEAT_CODE INTO V_C_SEAT_CODE FROM SEAT S JOIN THEATER T ON S.THEA_CODE = T.THEA_CODE
    JOIN CGV_CINEMA CC ON T.CINEMA_CODE = CC.CINEMA_CODE
    WHERE STORE_NAME = P_SEARCH_THEA AND T.THEA_NAME = P_THEA_NAME AND SEAT_NO = P_SEAT_NO;
    --영상기술 가져오기
    SELECT DISTINCT NVL(VIDEO_TEC, '2D') INTO V_VIDEO_TEC FROM SCREENING_SCHEDULE
    WHERE SCHED_CODE = V_SEETIME_CODE;
    --가격 가져오기
    SELECT DISTINCT P_C.ADT_PRI INTO V_RES_PRI FROM PRI_CLASS P_C JOIN SEAT S ON S.SEAT_CODE = P_C.SEAT_CODE
    WHERE S.C_SEAT_CODE = V_C_SEAT_CODE AND THEA_KND = V_VIDEO_TEC;
    --예매된 좌석인지 가져오기
    SELECT COUNT(*) INTO V_RES_C_SEAT_CODE FROM RESERVATION_SEAT WHERE C_SEAT_CODE = V_C_SEAT_CODE;
    IF V_COUNT != 0 OR V_NON_PAT IS NULL OR V_MOVIE_CODE IS NULL OR V_C_SEAT_CODE IS NULL THEN
        DBMS_OUTPUT.PUT_LINE('예매를 할 수 없습니다');
    ELSE
        IF(V_RES_C_SEAT_CODE = 0) THEN
            INSERT INTO RESERVATION VALUES (P_RES_NO, V_SEETIME_CODE, V_NON_PAT, V_MOVIE_CODE, V_C_SEAT_CODE
            , SYSDATE, P_RES_PEO, V_RES_PRI);
            INSERT INTO RESERVATION_SEAT VALUES (P_RES_NO, V_C_SEAT_CODE, NULL, V_NON_PAT);
            COMMIT;
            DBMS_OUTPUT.PUT_LINE('예매가 완료되었습니다. 결제 창으로 이동합니다.');
```

6. PL/SQL

● 비회원 예매 실행결과

--비밀번호가 일치하지 않을 경우 예매 불가능

```
EXEC UP_NOMEM_RESERVATION(CGV.FN_RES_NO(), '서희정', '94/05/05', '0522', '0520', '01099374928', '41', 'CGV 수원',  
                           '보스 베이비', '8관', '17/05/18', '11:30', '1', 'I1');
```

비밀번호가 일치하지 않습니다.

--이미 예매된 좌석 예매 불가능

```
EXEC UP_NOMEM_RESERVATION(CGV.FN_RES_NO(), '서희정', '94/05/05', '0520', '0520', '01099374928', '41', 'CGV 수원',  
                           '보스 베이비', '8관', '17/05/18', '11:30', '1', 'I1');
```

이미 예매된 좌석입니다

--예매하기

```
EXEC UP_NOMEM_RESERVATION(CGV.FN_RES_NO(), '서희정', '94/05/05', '0520', '0520', '01099374928', '41', 'CGV 수원',  
                           '보스 베이비', '8관', '17/05/18', '11:30', '1', 'I10');
```

비회원 예매는 준비중입니다.

6. PL/SQL

● CGV 화면 / 비회원 결제

비회원결제 비회원 예매는 신용카드 결제만 가능합니다.

일반 신용카드

카드종류

카드를 선택하세요

카드번호 - - -

비밀번호 * *

유효기간 월 년 예) 2015년 9월 -> 09월 15년

법정생년월일 (6자리) - * * * * *

우리은행

위비꿀머니 포인트

CGV에서 현금처럼 사용하자!

* 즉시할인혜택이 적용되는 카드는, 즉시할인 신용카드 탭에서 결제하셔야 할인이 적용됩니다.

※ 신용카드 결제 가능 최소 금액은 1,000원 이상입니다.

삼성U포인트 적립 OK캐쉬백 적립 KIA RED MEMBERS 적립 (삼성U포인트, OK캐쉬백, KIA RED MEMBERS 포인트는 포인트 중복 적립 불가)

결제하실 금액

20,000 원

할인내역

총 할인금액

0 원

결제내역

신용카드 20,000원

남은 결제금액

20,000 원

OK캐쉬백 10포인트부터 티켓 전액 결제가능!

현대카드 M포인트 사용하고 즉시 할인받자

위비꿀머니 우리은행 위비꿀머니 사용가능!

←

좌석선택

보스베이비

2D(자막)

전체 관람가

극장 CGV 수원

일시 2017.5.9(화) 19:20

상영관 5관[수원여자대학교]

인원 일반 2명

좌석명 Standard석

좌석번호 D13,D14

✓ 결제하기

CGV²

6. PL/SQL

● 비회원 결제 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_NOMEM_PAT
(
    PRES_NO IN NOMEM_PAT.RES_NUM%TYPE,
    PNON_PAT IN NON_MEM.NON_PAT%TYPE,
    PCARD_NAME IN NOMEM_PAT.CAD_NAME%TYPE,
    PCARD_NO IN NOMEM_PAT.CAD_NO%TYPE,
    PPWD IN NOMEM_PAT.PWD%TYPE,
    PEXPDAT IN NOMEM_PAT.EXPDAT%TYPE,
    PBRT IN NOMEM_PAT.BRT%TYPE
)
IS
    VPATSUM NOMEM_PAT.PATSUM%TYPE;
    VPAT_CODE CHAR(5);
    FK_EXIST EXCEPTION;
    PRAGMA EXCEPTION_INIT(FK_EXIST,-2291);
BEGIN
    SELECT NVL(R.RES_PRI,R.RES_PEO*9000) INTO VPATSUM FROM RESERVATION R
    WHERE R.MEM_ID = PNON_PAT AND PRES_NO = R.RES_NO;
    INSERT INTO NOMEM_PAT VALUES (SEQ_NOMEM_PAT.NEXTVAL, PRES_NO, PNON_PAT, '신용카드', PCARD_NAME,
    PCARD_NO, PPWD, PEXPDAT, PBRT, SYSDATE, TO_CHAR(SYSDATE,'hh24:mi'),VPATSUM);
    DBMS_OUTPUT.PUT_LINE('결제가 완료되었습니다');
    COMMIT;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('결제를 할 수 없습니다');
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('이미 결제가 되었습니다');
    WHEN FK_EXIST THEN
        --foreign key를 잘못 설정한 탓에 에러로 처리
        DBMS_OUTPUT.PUT_LINE('비회원 결제는 준비중입니다.');
```

```
--예매를 할 수 없기 때문에 결제 불가능
EXEC UP_NOMEM_PAT('','KQOVA','신한카드','1234-1234-1234-1234','1234','21/07/01','94/05/20');
```

결제를 할 수 없습니다

6. PL/SQL

● 예매 취소 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_RESERVATION_CANCEL
(
  PRES_NO IN RES_CANCEL.RES_NO%TYPE,
  PMEM_ID IN MEMBER_PAT.MEM_ID%TYPE
)
IS
  VSEEDAT DATE;
  VSUM NUMBER;
  vBRKDOWN_GO CHAR(4);
  VRES_NO RES_CANCEL.RES_NO%TYPE;
  VCOUNT NUMBER := 0;
BEGIN
  SELECT COUNT(*) INTO VCOUNT FROM RES_CANCEL WHERE RES_NO = PRES_NO;
  SELECT TO_DATE(TO_CHAR(SYSDATE, 'YYYY') || SUBSTR(R.SEETIME_CODE, 17, 4), 'yyyymmdd'), NVL(R.RES_PRI, R.RES_PEO * 9000)
  INTO VSEEDAT, VSUM FROM RESERVATION R WHERE R.MEM_ID = PMEM_ID AND R.RES_NO = PRES_NO;

  IF (VCOUNT = 0) THEN
    INSERT INTO RES_CANCEL VALUES (SEQ_RES_CANCEL.NEXTVAL, PRES_NO, PMEM_ID, VSEEDAT, SYSDATE, VSUM);
    DELETE FROM RESERVATION_SEAT WHERE RES_NO = PRES_NO;
    DBMS_OUTPUT.PUT_LINE('예매가 취소되었습니다');
  ELSE
    DBMS_OUTPUT.PUT_LINE('이미 취소를 하셨습니다. ');
  END IF;
  COMMIT;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('예매 취소를 할 수 없습니다');
  WHEN DUP_VAL_ON_INDEX THEN
    DBMS_OUTPUT.PUT_LINE('내역순번을 확인해주세요');
END;

--예매 취소 불가능
EXEC UP_RESERVATION_CANCEL('예매번호X', 'seohj0520'); 예매 취소를 할 수 없습니다

--예매 취소
EXEC UP_RESERVATION_CANCEL('7625-4145-5815-7119', 'seohj0520'); 예매가 취소되었습니다
```

	BRKDOWN_GO	RES_NO	MEM_ID	SEEDAT	CNLDAT	SUM
1	5	3262-3349-9856-3998	kenik	17/05/18	17/05/22	9000
2	21	7625-4145-5815-7119	seohj0520	17/05/18	17/06/07	10000

6. PL/SQL

● 평점 작성 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MOVIE_AVG_GRADE
(
  PRES_NO IN MOVIE_AVG_GRADE.RES_NO%TYPE,
  PMEM_ID IN MEMBER_PAT.MEM_ID%TYPE,
  pDIRECT_GRA IN MOVIE_AVG_GRADE.DIRECT_GRA%TYPE,
  pOST_GRA IN MOVIE_AVG_GRADE.OST_GRA%TYPE,
  pSTORY_GRA IN MOVIE_AVG_GRADE.STORY_GRA%TYPE,
  pVIDEO_GRA IN MOVIE_AVG_GRADE.VIDEO_GRA%TYPE,
  pACTOR_GRA IN MOVIE_AVG_GRADE.ACTOR_GRA%TYPE
)
IS
  vTOTAL_GRA MOVIE_AVG_GRADE.TOTAL_GRA%TYPE;
  vCOUNT NUMBER := 0;
  vMOVIE_CODE MOVIE_AVG_GRADE.MOVIE_CODE%TYPE;
BEGIN
  SELECT COUNT(*) INTO vCOUNT FROM RES_CANCEL WHERE RES_NO = PRES_NO;
  SELECT R.MOVIE_CODE INTO vMOVIE_CODE FROM RESERVATION R
  WHERE R.MEM_ID = PMEM_ID AND R.RES_NO = PRES_NO;

  vTOTAL_GRA := (pDIRECT_GRA + pOST_GRA + pSTORY_GRA + pVIDEO_GRA + pACTOR_GRA) / 5;

  IF( vCOUNT = 0 ) then
    IF (PRES_NO IS NULL OR PDIRECT_GRA IS NULL OR POST_GRA IS NULL OR PSTORY_GRA IS NULL
      OR PVIDEO_GRA IS NULL OR PACTOR_GRA IS NULL OR VTOTAL_GRA IS NULL) THEN
      DBMS_OUTPUT.PUT_LINE('평점을 모두 작성해주세요');
    ELSE
      INSERT INTO MOVIE_AVG_GRADE VALUES(PRES_NO, PMEM_ID, vMOVIE_CODE,
        PDIRECT_GRA, POST_GRA, PSTORY_GRA, PVIDEO_GRA, PACTOR_GRA, VTOTAL_GRA);
      DBMS_OUTPUT.PUT_LINE('평점이 등록 되었습니다');
    END IF;
  ELSE
    DBMS_OUTPUT.PUT_LINE('평점을 작성하실 수 없습니다');
  END IF;
  COMMIT;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('평점을 작성하실 수 없습니다');
END;
```

6. PL/SQL

● 평점 작성 실행결과

--예매하지 않은 영화의 평점을 입력할 경우

```
EXEC UP_MOVIE_AVG_GRADE('7625-4145-5815-7119','seohj0520', 66,77,44,90,null);
```

평점을 작성하실 수 없습니다

RES_NO	SEETIME_CODE	MEM_ID	MOVIE_CODE	C_SEAT_CODE	RES_DAT	RES_PEO	RES_PRI
1 5965-9483-9900-8218	200608_20174061_0518_02	seohj0520	20174061	200608H7	17/06/07	1	10000

--평점을 모두 입력하지 않을 경우

```
EXEC UP_MOVIE_AVG_GRADE('5965-9483-9900-8218','seohj0520', 80,40,50,27,NULL);
```

평점을 모두 작성해주세요

--평점입력

```
EXEC UP_MOVIE_AVG_GRADE('5965-9483-9900-8218','seohj0520', 80,40,50,27,80);
```

평점이 등록 되었습니다

RES_NO	MEM_ID	MOVIE_CODE	DIRECT_GRA	OST_GRA	STORY_GRA	VIDEO_GRA	ACTOR_GRA	TOTAL_GRA
1 7207-3241-8396-7125	zzang	20174061	66	77	44	90	80	71.4
2 7021-1970-3363-8562	kenik	20174061	66	77	44	90	80	71.4
3 5965-9483-9900-8218	seohj0520	20174061	80	40	50	27	80	55.4

6. PL/SQL

- 리뷰 작성 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_REVIEW
(
  PRES_NO IN REVIEW.RES_NO%TYPE,
  PMEM_ID IN MEMBER_PAT.MEM_ID%TYPE,
  pGOOD_REV IN REVIEW.GOOD_REV%TYPE,
  pBAD_REV IN REVIEW.BAD_REV%TYPE
)
IS
  VCOUNT NUMBER := 0;
  VMOVIE_CODE REVIEW.MOVIE_CODE%TYPE;
BEGIN
  SELECT COUNT(*) INTO VCOUNT FROM RES_CANCELE WHERE RES_NO = PRES_NO;
  SELECT R.MOVIE_CODE INTO VMOVIE_CODE FROM RESERVATION R
  WHERE R.MEM_ID = PMEM_ID AND R.RES_NO = PRES_NO;

  IF( VCOUNT = 0 ) THEN
    IF (PBAD_REV IS NULL) THEN
      INSERT INTO REVIEW(RES_NO, MEM_ID, MOVIE_CODE, GOOD_REV) VALUES(PRES_NO, PMEM_ID, VMOVIE_CODE, PGOOD_REV);
      DBMS_OUTPUT.PUT_LINE('리뷰가 등록 되었습니다');
    ELSIF (PGOOD_REV IS NULL) THEN
      INSERT INTO REVIEW(RES_NO, MEM_ID, MOVIE_CODE, BAD_REV) VALUES(PRES_NO, PMEM_ID, VMOVIE_CODE, PBAD_REV);
      DBMS_OUTPUT.PUT_LINE('리뷰가 등록 되었습니다');
    ELSIF NOT (PGOOD_REV IS NULL AND PBAD_REV IS NULL) THEN
      DBMS_OUTPUT.PUT_LINE('호감 / 비호감 리뷰 중 하나만 선택해주세요');
    END IF;
  ELSE
    DBMS_OUTPUT.PUT_LINE('리뷰를 작성하실 수 없습니다');
  END IF;
  COMMIT;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('리뷰를 작성하실 수 없습니다');
END;
```

6. PL/SQL

● 리뷰 작성 실행결과

--예매하지 않은 영화의 리뷰를 작성할 경우

```
EXEC UP_REVIEW('7625-4145-5815-7119','seohj0520',NULL,'별로예요');
```

리뷰를 작성하실 수 없습니다

--둘다 쓸 경우

```
EXEC UP_REVIEW('5965-9483-9900-8218','seohj0520','즐겁게 봤습니다','별로예요');
```

호감 / 비호감 리뷰 중 하나만 선택해주세요

--리뷰 작성

```
EXEC UP_REVIEW('5965-9483-9900-8218','seohj0520','감동적이었습니다~',NULL);
```

리뷰가 등록 되었습니다

RES_NO	MEM_ID	MOVIE_CODE	GOOD_REV	BAD_REV	MOV_MANIA	REC_NO	
1	7021-1970-3363-8562	kenik	20174061	즐겁게 봤습니다. 최고예요 ㅎㅎ	(null)	N	0
2	5965-9483-9900-8218	seohj0520	20174061	감동적이었습니다~	(null)	N	0

6. PL/SQL

- 관리자 / 영화등록 PL/SQL

```
CREATE OR REPLACE PROCEDURE UP_MOVIE
(
  pID IN VARCHAR2,
  pPwd IN VARCHAR2,
  pMOVIE_CODE IN MOVIE.MOVIE_CODE%TYPE,
  pMOVIE_TITLE IN MOVIE.MOVIE_TITLE%TYPE,
  pSTD_CODE IN MOVIE.STD_CODE%TYPE,
  pGENRE_NAME IN MOVIE.GENRE_NAME%TYPE,
  pREL_DATE IN MOVIE.REL_DATE%TYPE,
  pMAIN_CON IN MOVIE.MAIN_CON%TYPE,
  pPRES IN MOVIE.PRES%TYPE,
  pACTOR IN MOVIE.ACTOR%TYPE
)
IS
  VMSG VARCHAR2(50);
  VACC_VISITORS MOVIE.ACC_VISITORS%TYPE := 0;
BEGIN
  SELECT '관리자로 접속하셨습니다.' INTO VMSG FROM DUAL
  WHERE PID = 'admin' AND PPWD = 'password';
  DBMS_OUTPUT.PUT_LINE(VMSG);
  IF (PMOVIE_CODE IS NULL) THEN
    DBMS_OUTPUT.PUT_LINE('영화 코드를 입력해주세요');
  ELSIF (PMOVIE_TITLE IS NULL) THEN
    DBMS_OUTPUT.PUT_LINE('영화 제목을 입력해주세요');
  ELSIF (PGENRE_NAME IS NULL) THEN
    DBMS_OUTPUT.PUT_LINE('관람 등급을 입력해주세요');
  ELSE
    INSERT INTO MOVIE VALUES
      (pMOVIE_CODE, pMOVIE_TITLE, pSTD_CODE, pGENRE_NAME, pREL_DATE, 0, pMAIN_CON, pPRES, pACTOR);
  END IF;
  COMMIT;
  DBMS_OUTPUT.PUT_LINE(pMOVIE_TITLE || ' 영화를 등록하셨습니다');
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('관리자가 아닙니다');
  WHEN DUP_VAL_ON_INDEX THEN
    DBMS_OUTPUT.PUT_LINE('영화코드가 중복됩니다. ');
END;
```

6. PL/SQL

● 관리자 / 영화등록 실행결과

--관리자가 아닐 경우

```
EXEC UP_MOVIE('seohj0520','pass','20174602','노무현입니다','12','다큐멘터리','17/05/25',  
'지지율 2%의 꼴찌 후보에서 대선후보 1위, 국민의 대통령이 되기까지! 2002년 전국을 뒤흔들었던 노무현,  
그 기적의 역전 드라마','이창재','노무현','이화춘','유시민');
```

관리자가 아닙니다

--이미 있는 영화를 등록할 경우

```
EXEC UP_MOVIE('admin','password','20162722','보안관','15','범죄,코미디','17/05/03','과잉 수사로 잘리고 낙향한 전직 형사 대호(이성민).  
보안관을 자처하며 바다만큼 드넓은 오지랖으로 고향 '기장'을 수호한다.  
평화롭던 동네에 비치타운 건설을 위해 성공한 사업가 종진(조진웅)이 서울에서 내려온 그때, 인근 해운대에 마약이 돌기 시작한다.  
두고 봐라, 이래 당하고만 있겠나. 게임은 인자 시작이다!','김형주','이성민,조진웅,김성균,김종수,조우진,임현성,배정남');
```

관리자로 접속하셨습니다.
영화코드가 중복됩니다.

--관리자일 경우

```
EXEC UP_MOVIE('admin','password','20174602','노무현입니다','12','다큐멘터리','17/05/25',  
'지지율 2%의 꼴찌 후보에서 대선후보 1위, 국민의 대통령이 되기까지! 2002년 전국을 뒤흔들었던 노무현,  
그 기적의 역전 드라마','이창재','노무현','이화춘','유시민');
```

관리자로 접속하셨습니다.
노무현입니다 영화를 등록하셨습니다

	MOVIE_CODE	MOVIE_TITLE	STD_CODE
37	20060264	악마는 프라다를 입는다	12
38	20174165	극장판 또봇: 로봇군단의 습격	전체
39	20162508	오발탄	15
40	20164721	미녀와 야수	전체
41	20129511	전송가	미정
42	20167363	히든 피겨스	12
43	20173881	런던 프라이드	15
44	20163026	광야의 40일	12
45	19698042	미워도 다시 한번 2	미정
46	20174501	오즈의 마법사 2	전체
47	20170741	지니어스	12
48	20170381	매직키드 보보	전체
49	20179442	밤의 해변에서 혼자	청불
50	20167148	컴, 투게더	15
51	20174602	노무현입니다	12

