



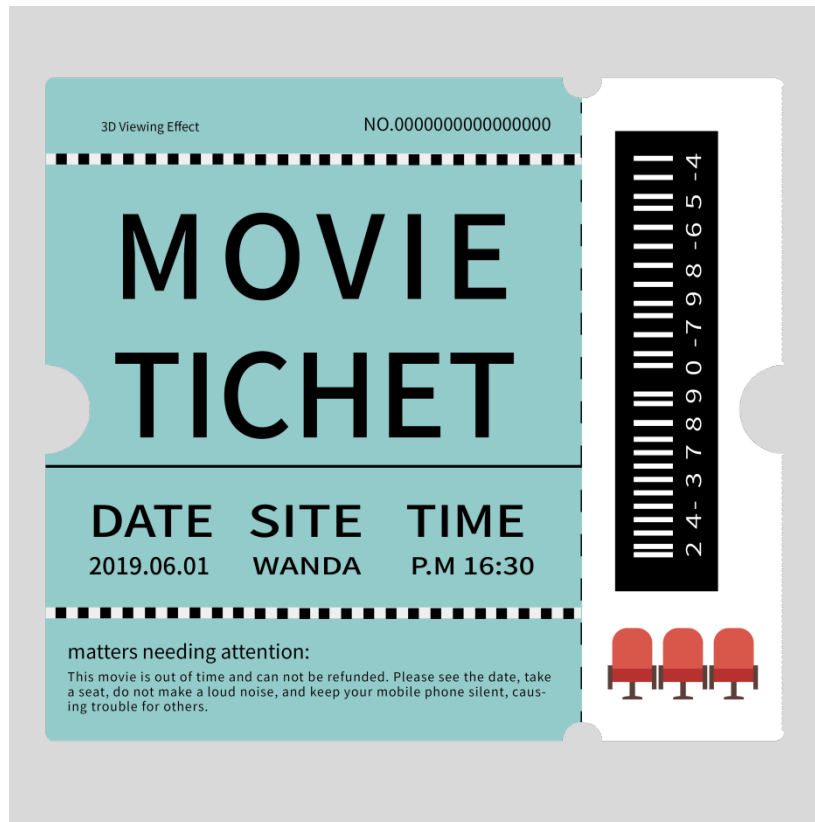
김가을 / 진소원 / 김영균 / 고동현

질문하지말아조



CONTENTS

- 01 DB 주제 설명
- 02 기능과 요구사항 정의
- 03 ERD
- 04 테이블 명세서
- 05 동작 테스트
- 06 기능 구현
- 07 추가 기능
- 08 프로젝트 후기



회원제 영화예매 데이터베이스 개요

회원제로만 운영하는 영화 예매 프로그램에 기반이 되는 데이터베이스를 콘셉트로 여러 기능을 구현하였습니다. 영화 예매라고 한다면 크게 회원관리, 영화예매, 결제 관리로 3가지 기본 기능으로 나누어집니다. 기본 제공 기능에서 추가 기능을 확장하는 콘셉트로 목표하였습니다. 회원 관리에는 회원정보와 회원별 포인트 제도를 바탕으로 예매 시 자동 포인트 합산까지 되는 기능이 제공하게 구현했습니다. 영화 예매는 영화 정보와 등록된 리뷰를 참고할 수 있도록 하고 예매 진행과 결제 진행 시 자동으로 잔여 좌석에 변동이 될 수 있도록 했습니다. 결제 방식은 최신 트렌드에 맞춰 여러 가지 결제 방법을 제안하였고 결제 완료 시 발급되는 고유 번호를 기준으로 티켓 출력 확인을 할 수 있는 프로시저를 구현하였습니다. 추가적으로 결제 취소를 고려하여 취소 시 지급된 포인트 자동반과 예약한 좌석수에 취소된 만큼 원복 될 수 있도록 하였습니다

1. 회원정보

회원고유번호, ID, 비밀번호, 이름, 전화번호, 포인트
회원고유번호와 ID는 중복 불가이며, 포인트의 기본값은 0이다

2. 영화정보

영화 번호, 영화제목, 장르, 영화 감독, 주인공의 정보
영화 번호는 중복 불가하며 같은 영화여도 다른 번호를 부여한다.

3. 리뷰

영화 번호, 이미 관람한 관람객들의 리뷰가 있다.

4. 상영관

상영관 번호, 총 좌석, 잔여 좌석
상영관은 총 5개이며, 개별적 총 좌석은 각각 다르지만 최대 200석 까지만 가능하다.
잔여 좌석은 결제와 취소에 따라 차감되거나 복구된다

5. 상영일정

상영 일정 번호, 상영 시간, 상영관 번호, 영화 번호

6. 예매

예매 번호, 상영 일정 번호, 회원 번호, 인원수
인원 수 만큼 자동으로 포인트 적용

7. 결제

예매 번호, 결제 방법, 티켓 가격
티켓 가격 기본값은 15,000원 이다

기능과 요구 사항 정의



01
02
03
04
05
06
07
08

ERD Movie Ticket Database

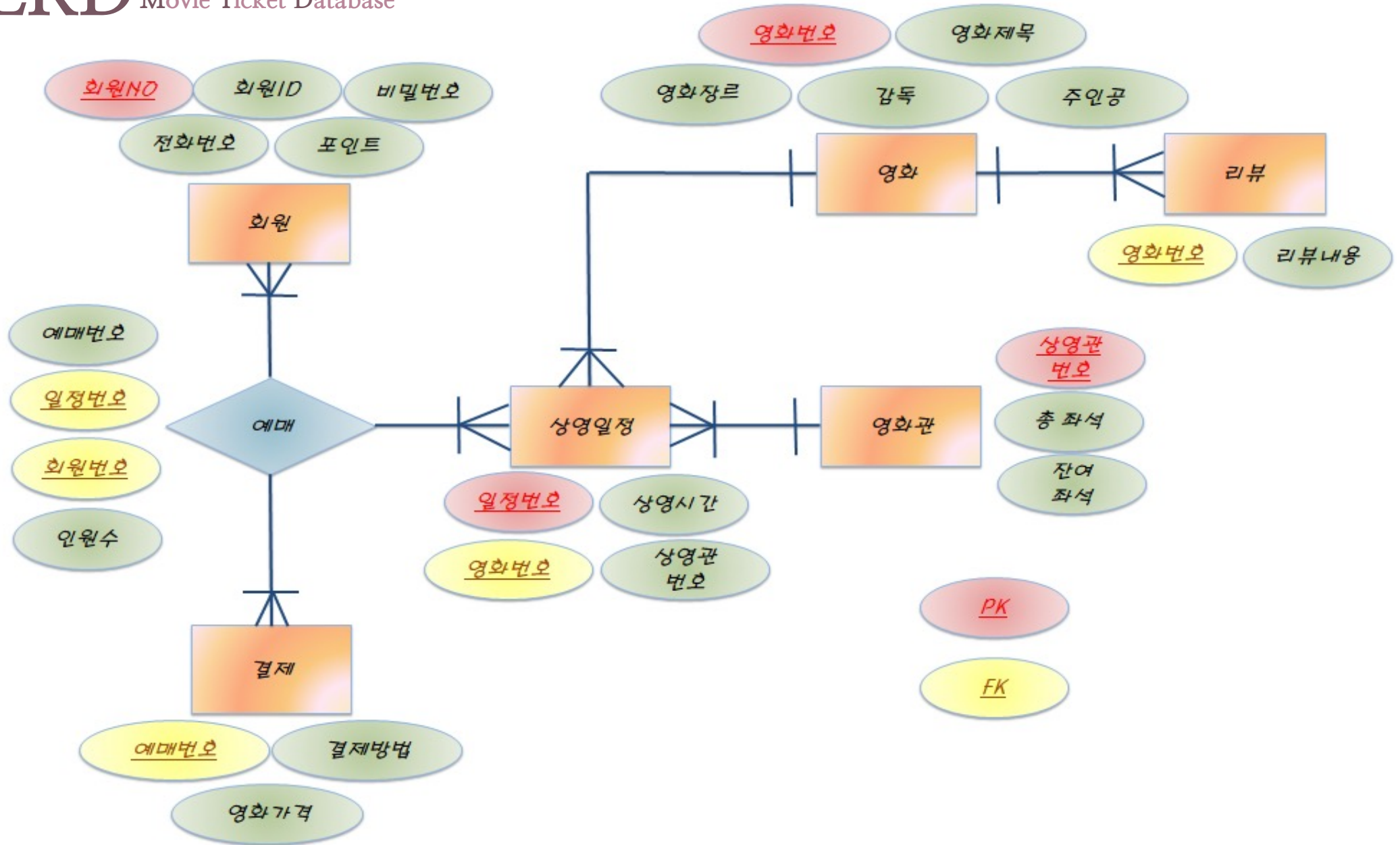


Table Statement & Query

테이블명		회원	데이터베이스명	ORCL			
테이블ID		CUSTOMER	테이블 설명	회원에 대한 정보를 기술하는 테이블			
순번	컬럼명	컬럼ID	타입 및 길이	NULL	PK	FK	설명
1	회원NO	cos_num	number	NOT NULL	yes	no	Primary key
2	회원ID	cos_id	varchar2(15)	NOT NULL	no	no	Unique
3	비밀번호	cos_pass	varchar2(15)	NOT NULL	no	no	
4	전화번호	cos_phone	varchar2(13)		no	no	
5	포인트	cos_point	number	NOT NULL	no	no	Default 0
6	회원이름	cos_name	varchar2(10)	NOT NULL	no	no	

TABLE 생성 쿼리문

```
CREATE TABLE CUSTOMER (
    cos_num NUMBER
    , cos_id varchar2(15) NOT NULL
    , cos_name varchar2(10) NOT NULL
    , cos_pass varchar2(15) NOT NULL
    , cos_phone varchar2(13)
    , cos_point NUMBER DEFAULT 0
    , PRIMARY KEY (cos_num)
    , UNIQUE (cos_id)
);
```

회원 포인트, 기본값은 0으로 함
 회원번호를 기본키로 설정
 회원의 아이디가 중복되지 않도록 unique 제약조건 설정

Table Statement & Query

테이블명 `		영화	데이터베이스명	ORCL			
테이블ID		MOVIE	테이블 설명	영화에 대한 상세한 정보를 기술하는 테이블			
순번	컬럼명	컬럼ID	타입 및 길이	NULL	PK	FK	설명
1	영화번호	mo_num	number	NOT NULL	yes	no	Primary key
2	영화제목	mo_title	varchar2(50)	NOT NULL	no	no	
3	주인공	mo_actor	varchar2(10)	NOT NULL	no	no	
4	영화장르	mo_gen	varchar2(15)	NOT NULL	no	no	Check
5	감독	mo_super	Varchar2(15)	NOT NULL	no	no	

TABLE 생성 쿼리문

```
CREATE TABLE MOVIE (
    mo_num NUMBER PRIMARY KEY
    , mo_title varchar2(50) NOT NULL
    , mo_actor varchar2(20) NOT NULL
    , mo_gen varchar2(15) CHECK
        (mo_gen IN('로맨스', 'SF', '액션', '스릴러'
        , '드라마', '코미디', '판타지', '애니'))
    , mo_super varchar2(20) NOT NULL
);
```

영화정보
영화번호
영화 제목
영화 배우
영화 장르

장르의 입력 값 제한
영화 감독

Table Statement & Query

테이블명		상영일정	데이터베이스명	ORCL			
테이블ID		SHOW_DAY	테이블 설명	각 영화에 상영 일정 및 시간, 상영관에 대한 정보를 기술하는 테이블			
순번	컬럼명	컬럼ID	타입 및 길이	NULL	PK	FK	설명
1	일정번호	Sh_num	Number	NOT NULL	yes	no	Primary key
2	상영시간	Sh_time	date		no	no	
3	상영관번호	Sh_moiven	Number	NOT NULL	no	no	Unique
4	영화번호	Mo_num	number	NOT NULL	no	yes	Reference moive

TABLE 생성 쿼리문

```
CREATE TABLE SHOW_DAY (
    sh_num NUMBER
    , sh_time DATE
    , sh_movien NUMBER NOT NULL
    , mo_num NUMBER
    , CONSTRAINT show_pk

    PRIMARY KEY (sh_num)
    , FOREIGN KEY (mo_num) REFERENCES MOVIE(mo_num)
    , UNIQUE (sh_movien)
);
```

상영정보
일정 번호
상영 시간
상영관 번호
영화 번호

일정 번호를 기본키로 지정
영화 번호를 영화테이블의 영화 번호를 참조하는 외래키로 지정
상영관 번호는 중복되어선 안됨 됨

Table Statement & Query

테이블명		영화관	데이터베이스명	ORCL			
테이블ID		MOVIE_ROUNDGE	테이블 설명	회원에 대한 정보를 기술하는 테이블			
순번	컬럼명	컬럼ID	타입 및 길이	NULL	PK	FK	설명
1	상영관번호	ro_num	number	NOT NULL	no	yes	Reference show_day(sh_movien)
2	총좌석	ro_seet	number	NOT NULL	no	no	Check (ro_seet<=200)
3	잔여좌석	re_seet	number	NOT NULL	no	no	

TABLE 생성 쿼리문

```
CREATE TABLE MOVIE_ROUNDGE (
    ro_num NUMBER
    , ro_seet NUMBER CHECK ( ro_seet <= 200)
    , re_seet NUMBER NOT NULL

    , CONSTRAINT rounge_fk
    FOREIGN KEY (ro_num) REFERENCES SHOW_DAY(sh_movien)
);
```

영화관
상영관 번호
총좌석, 좌석은 200개가 넘지 않도록 제한
잔여 좌석

부모테이블인 SHOW_DAY의 sh_movien과
자식 테이블인 MOVIE_ROUNDGE 외래키 지정

Table Statement & Query

테이블명		예매	데이터베이스명	ORCL			
테이블ID		ticketting	테이블 설명	예매할 때 필때 필요한 정보를 기술하는 테이블			
순번	컬럼명	컬럼ID	타입 및 길이	NULL	PK	FK	설명
1	예매번호	Ti_num	number	NOT NULL	yes	no	Primary key
2	회원ID	Sh_num	varchar2(15)	NOT NULL	no	yes	Reference show_day
3	비밀번호	Cos_num	varchar2(15)	NOT NULL	no	yes	Reference customer
4	전화번호	Ti_nop	varchar2(13)	NOT NULL	no	no	

TABLE 생성 쿼리문

```
CREATE TABLE TICKETTING (
    ti_num      number
, sh_num      number
, cos_num     number
, ti_nop      NUMBER      NOT NULL
, CONSTRAINT ticket_pk
    PRIMARY KEY (ti_num)
, FOREIGN KEY (sh_num) REFERENCES SHOW_DAY (sh_num)
    ON DELETE SET NULL
, FOREIGN KEY (cos_num) REFERENCES CUSTOMER (cos_num)
    ON DELETE SET NULL
);
```

예매정보
예매번호
일정번호
회원번호
인원수

예매번호를 기본키로 지정

일정번호 외래외래키로 지정하며 부모 테이블인
SHOW_DAY 테이블에 정보 삭제 시 NULL로 보존
회원번호 외래키 지정
부모 테이블인 CUSTOMER에 정보 삭제되더라도
NULL값으로 처리되며 정보유지 옵션지정

Table Statement & Query

테이블명		결제	데이터베이스명	ORCL			
테이블ID		PAY	테이블 설명	결제에 대한 정보를 기술하는 테이블			
순번	컬럼명	컬럼ID	타입 및 길이	NULL	PK	FK	설명
1	예약번호	ti_num	number	NOT NULL	no	yes	Reference ticketting, unique
2	결제방법	Py_type	varchar2(7)	NOT NULL	no	no	Check
3	기본금액	Py_tot	number	NOT NULL	no	no	Default 15000

TABLE 생성 쿼리문

```
CREATE TABLE PAY (
    ti_num number
    , py_type varchar2(20) CHECK
        (py_type IN('카드', '삼성페이', '카카오페이', '애플페이', '네이버페이', '상품권'))
    , py_tot number DEFAULT 15000
        CONSTRAINT pay_fk FOREIGN KEY (ti_num)
            REFERENCES TICKETTING (ti_num)
            ON DELETE CASCADE
);
```

결제정보
예약번호
결제방법
결제방법 입력 값 제한
1인당 기본금, 기본값 15000으로

부모 테이블인 TICKETTING 테이블에 예약번호
삭제 시 자식 테이블인 PAY 테이블에도 정보가 삭제
될 수 있도록 옵션 추가

Table Statement & Query

테이블명		리뷰	데이터베이스명	ORCL			
테이블ID		Review	테이블 설명	이미 영화를 본 사람들이 볼 예정인 사람들에게 후기를 보여주기 위한 재료 테이블			
순번	컬럼명	컬럼ID	타입 및 길이	NULL	PK	FK	설명
1	영화번호	mo_num	number	NOT NULL	No	Yes	Reference movie / On Delete Cascade
2	리뷰내용	re_text	varchar2(15)		No	No	

TABLE 생성 쿼리문

```
CREATE TABLE REVIEW (
    mo_num    number
, re_text    varchar2(100)
, CONSTRAINT re_fk
FOREIGN KEY (mo_num) REFERENCES MOVIE (mo_num)
ON DELETE CASCADE
);
```

리뷰정보

부모 테이블인 MOVIE테이블에 영화 넘버가 삭제 시 = 영화 삭제
자녀 테이블인 Review테이블에 자료도 삭제될 수 있도록 옵션지정

동작 테스트

Q. 포인트가 높은 순서대로 4~6위까지 회원의 정보를 출력하시오

```
SELECT *  
FROM(  
    select rownum rn,c.*  
    from(  
        select * from customer c  
        order by cos_point desc)c  
    )c1  
WHERE rn between 4 and 6;
```



등수	회원번호	이름	포인트
4	11	하도녀	400
5	10	솔방울	400
6	15	옥도시	300

동작 테스트

Q. 회원의 정보를 출력하시오 단 비밀번호는 *표로 출력하시오

```
SELECT  
  
    cos_name,cos_id,  
    rpad(substr(cos_pass,0,1),  
    length(cos_pass),'*')  
  
FROM customer;
```



	⚡ COS_NAME	⚡ COS_ID	⚡ 비밀번호
1	김나라	abccddfg	*****
2	밤나라	a42141	*****
3	강나라	abcsdfafg	*****
4	집가라	aasd124ddfg	*****
5	최춘식	233fscbg	*****
6	이도용	ab5dssg	*****
7	홍도도	abccssdkkdfg	*****
8	추마을	aSdfsddfg	*****
9	서마은	Gsfdddfg	*****

동작 테스트

Q. 회원중 포인트를 가장 많이 소유하고 있는 회원의 정보를 출력하시오

```
SELECT c.*  
FROM customer c  
  
WHERE cos_point =  
    (SELECT max(cos_point)  
     FROM customer);
```



	이름	전화번호	포인트
1	밤나라	010-0101-0202	1200

01
02
03
04
05
06
07
08

동작 테스트

Q. 잔여 좌석이 제일 적은 상영관의 정보를 출력하시오

```
SELECT *  
FROM ( SELECT *  
        FROM movie_rounge  
        ORDER BY re_seet)  
WHERE rownum = 1;
```



	RO_NUM	RO_SEET	RE_SEET
1	2	150	136

기능 구현1

1. 예매 이벤트 발생 시 구매 회원의 포인트 증가 및 상영관 잔여 좌석 차감 자동 진행 및 기존 데이터 반영



트리거의 필요 이유

회원이 예매를 하면
회원의 포인트 컬럼의 값이 update 되어 하고
상영관의 잔여 좌석 컬럼값이 update를 개별 처리 해야한다.

예매 테이블에 insert 될때마다 마다
update를 수동적으로 해주어야 하므로
트리거가 없다면. 수작업을 해야 하는
매우 불편한 부분이 발생 된다.

이를 개선 하기 위해 자동으로 처리 되는 트리거를 사용 한다

기능 구현1

1. 예매 이벤트 발생 시 구매 회원의 포인트 증가 및 상영관 잔여 좌석 차감 자동 진행 및 기존 데이터 반영

트리거

```
CREATE OR REPLACE TRIGGER upd_cos_point
AFTER
INSERT ON TICKETTING
FOR EACH ROW

DECLARE
    cnt number;
    s_num number;

BEGIN
    cnt := :NEW.ti_nop;
    s_num := :NEW.sh_num;

    UPDATE CUSTOMER
    SET cos_point = cos_point + 100 * cnt
    WHERE cos_num = :NEW.cos_num ;

    UPDATE MOVIE_ROUNDGE
    SET re_seet = re_seet - cnt
    WHERE ro_num=
        (select sh_movien from show_day where sh_num=s_num) ;

END;
```

upd_cos_point

예매 테이블에 insert가 실행 된 후
변수 선언부 선언을 한다

데이터 타입이 number인
cnt와 s_num이라는 변수를 선언
cnt에 insert 된 튜플의 인원수 값을 할당한다
cnt 는 인원수 이다.

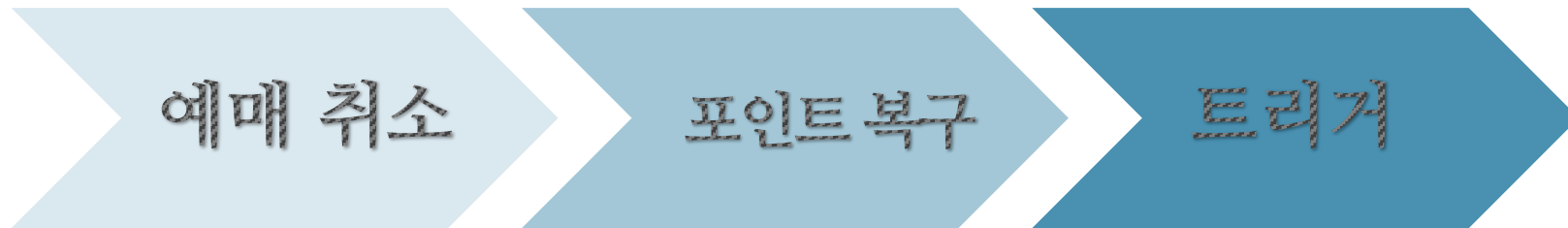
s_num에 insert 된 튜플의 일정번호를 할당한다 s_num은 일정번호
이다.
회원의 포인트 컬럼 값은
update = 포인트 적립 기능이다.
회원의 포인트에 100 * 인원수만큼 더해서
변경하고 저장 한다.

회원 번호가 새로 insert된 튜플의
회원번호와 같은 튜플을 선정 한다
상영관의 잔여좌석 값을
update = 잔여좌석 차감 기능을 한다
잔여좌석을 인원수만큼 빼서 변경하고 저장 하고
새로 들어온 상영관 번호와 같은 튜플을 선정 한다

기능 구현2

2. 예매 취소 시 회원의 포인트와 상영관의 잔여좌석 복구

KEY POINT



트리거의 필요 이유

회원이 예매를 하면
회원의 포인트 컬럼의 값이 update 되어 하고
상영관의 잔여 좌석 컬럼값이 update를 개별 처리 해야한다.

예매 테이블에 DELETE 될때마다 마다
update를 수동적으로 해주어야 하므로
트리거가 없다면, 수작업을 해야 하는
매우 불편한 부분이 발생 된다.

이를 개선 하기 위해 자동으로 처리 되는 트리거를 사용 한다

기능 구현2

2. 예매 취소 시 회원의 포인트와 상영관의 잔여좌석 복구

트리거

CREATE OR REPLACE TRIGGER upd_reset

AFTER DELETE ON TICKETTING
FOR EACH ROW

DECLARE

cnt number;
s_num number;

BEGIN

cnt := :NEW.ti_nop;
s_num := :NEW.sh_num;

UPDATE CUSTOMER

SET cos_point = cos_point - 100 * cnt
WHERE cos_num = :OLD.cos_num ;

UPDATE MOVIE_ROUNDGE

SET re_seet = re_seet + cnt
WHERE ro_num=
(select sh_movien from show_day where sh_num=s_num) ;

END;

upd_reset

예매 테이블에 insert 이벤트가 발생 되었을 때 실행된다.

변수 선언부에 기재된 데이터 타입이
number인 cnt와 s_num이라는 변수를 선언 해 놓는다.

cnt에 DELETE 된 튜플의 인원수 값을 할당한다
cnt 는 인원수이다.

s_num에 DELETE 된 튜플의 일정번호를 할당 한다
s_num는 일정번호이다

회원의 포인트 컬럼 값 update 하여 포인트 복구를 한다

회원의 포인트에 100 * 인원수만큼 뺄셈 연산 하여
변경하고 변경된 값을 회원 정보 포인트에 저장한다

회원 번호가 DELETE된 튜플의 회원번호와
같은 튜플을 선정 하여

상영관의 잔여좌석 값을 update 실행, 잔여좌석 복구 한다
잔여좌석을 예매한 인원수만큼 더해서 변경하고 저장 한다
상영관 번호와 같은 튜플을 선정한다.

기능 구현 확인

예매 이벤트 전 포인트 값 확인

이벤트 발생 후 포인트 값 확인

회원번호	회원이름	포인트	인원수
1	김나라	200	2
2	밤나라	1800	3
2	밤나라	1800	5
2	밤나라	1800	5
2	밤나라	1800	5
4	집가라	600	2
4	집가라	600	2
4	집가라	600	2
5	최춘식	600	6
10	슬방울	400	4
11	하도녀	400	4
14	최범죄	200	2
15	옥도시	300	3

회원번호	회원이름	포인트	인원수
1	김나라	700	5
1	김나라	700	2
2	밤나라	1800	3
2	밤나라	1800	5
2	밤나라	1800	5
2	밤나라	1800	5
4	집가라	600	2
4	집가라	600	2
4	집가라	600	2
5	최춘식	600	6
10	슬방울	400	4
11	하도녀	400	4
14	최범죄	200	2
15	옥도시	300	3

```
INSERT INTO TICKETTING VALUES (ti_no.nextval,1,1,5); (예매번호, 1번상영관,1번회원,인원수5명)
SELECT c.cos_num 회원번호, c.cos_name 회원이름, c.cos_point 포인트, t.ti_nop 인원수
FROM TICKETTING t, CUSTOMER c
WHERE t.cos_num=c.cos_num;
```

회원 정보 조회 조인으로 인원수까지 구현

기능 구현 확인

예매 이벤트 발생 전 좌석

상영관번호	총좌석	잔여좌석
1	200	198
2	150	136
3	180	173
4	150	141
5	200	187

예매 이벤트 발동 후 좌석

상영관번호	총좌석	잔여좌석
1	200	193
2	150	136
3	180	173
4	150	141
5	200	187

```
INSERT INTO TICKETTING VALUES (ti_no.nextval,1,1,5);  
      (예매번호, 1번상영관,1번회원,인원수5명)  
SELECT ro_num 상영관번호, ro_seet 총좌석, re_seet 잔여좌석  
FROM movie_rounge;
```

상영관 테이블 좌석 조회 구현

기능 구현 확인

예매 취소 이벤트 전 포인트 확인

회원번호	회원이름	포인트	인원수
1	김나라	700	5
1	김나라	700	2
2	밤나라	1800	3
2	밤나라	1800	5
2	밤나라	1800	5
2	밤나라	1800	5
4	집가라	600	2
4	집가라	600	2
4	집가라	600	2
5	최춘식	600	6
10	솔방울	400	4
11	하도녀	400	4
14	최범죄	200	2
15	옥도시	300	3

예매 취소 이벤트 발생 후 포인트 확인

회원번호	회원이름	포인트	인원수
1	김나라	200	2
2	밤나라	1800	3
2	밤나라	1800	5
2	밤나라	1800	5
2	밤나라	1800	5
4	집가라	600	2
4	집가라	600	2
4	집가라	600	2
5	최춘식	600	6
10	솔방울	400	4
11	하도녀	400	4
14	최범죄	200	2
15	옥도시	300	3

```
DELETE TICKETTING WHERE cos_num=1 and ti_nop=5;
```

(회원1번 인원수5명 예약 건 취소)

```
SELECT c.cos_num 회원번호, c.cos_name 회원이름, c.cos_point 포인트, t.ti_nop 인원수  
FROM ticketting t, customer c  
WHERE t.cos_num=c.cos_num;
```

(회원 정보 조회 조인으로 인원수까지 나오게 구현)

01
02
03
04
05
06
07
08

기능 구현 확인

예매 취소 이벤트 전 좌석 확인

상영관번호	총좌석	잔여좌석
1	200	193
2	150	136
3	180	173
4	150	141
5	200	187

예매 취소 이벤트 발생 후 좌석 확인

상영관번호	총좌석	잔여좌석
1	200	198
2	150	136
3	180	173
4	150	141
5	200	187

```
DELETE TICKETING WHERE cos_num=1 and ti_nop=5;  
(회원1번 인원수5명 예약건 취소)  
SELECT ro_num 상영관번호, ro_seet 총좌석, re_seet 잔여좌석  
FROM movie_rounge;
```

상영관 테이블 좌석 조회 구현

기능 구현 확인

회원 탈퇴 전 예매 정보 확인

예매번호	상영일...	회원번호	인원수
1	1	1	2
2	2	2	5
4	3	4	2
5	5	5	6
6	2	10	4
7	5	11	4
8	4	14	2
9	5	15	3
11	2	2	5
13	3	4	2
14	4	2	5
15	4	4	2
16	3	2	3

회원 탈퇴 후 예매정보 확인

예...	상영일정번호	회원번호	인원수
1	1	(null)	2
2	2	2	5
4	3	4	2
5	5	5	6
6	2	10	4
7	5	11	4
8	4	14	2
9	5	15	3
11	2	2	5
13	3	4	2
14	4	2	5
15	4	4	2
16	3	2	3

DELETE CUSTOMER WHERE cos_num=1;
1번회원을 삭제 한다.

*FOREIGN KEY (cos_num) REFERENCES CUSTOMER (cos_num) ON DELETE SET NULL
탈퇴된 회원의 예매정보는 남아있음

01
02
03
04
05
06
07
08

기능 구현 확인

영화 삭제 전

영화번호	영화제목	배우	장르	감독	내용
5	쿵푸허슬	주성치	코미디	주성치	생각없이 봤는데 재밌네요
5	쿵푸허슬	주성치	코미디	주성치	다큐영화 같았네요 지루함.
5	쿵푸허슬	주성치	코미디	주성치	다 똑같은 세상을 살고있겠지?
5	쿵푸허슬	주성치	코미디	주성치	핸드폰에 우리의 모든 삶이 다 있다.
5	쿵푸허슬	주성치	코미디	주성치	배우들 연기가 좋아서 몰입이 잘됐어요.
5	쿵푸허슬	주성치	코미디	주성치	코믹하면서도 생각 해볼만한 내용이 많은 영화!
5	쿵푸허슬	주성치	코미디	주성치	재미로만 보면 이 영화가 천만 영화가 되어야 마땅하다.
6	박쥐	송강호	드라마	박찬욱	비범한 사람이 평범해지는 과정
6	박쥐	송강호	드라마	박찬욱	지난 주말 극장가서 잘 보고왔다
6	박쥐	송강호	드라마	박찬욱	음악과 연출이 좋았어요.
6	박쥐	송강호	드라마	박찬욱	드디어 인간의 감정을 다룰줄 알게 된 누렁지
6	박쥐	송강호	드라마	박찬욱	인간이 얼마나 쓰레기인지 잘 보여준 영화
6	박쥐	송강호	드라마	박찬욱	마치 3시간이 30분 같았다.
6	박쥐	송강호	드라마	박찬욱	소원 배우의 연기력 너무너무 좋았어요

```
SELECT m.mo_num 영화번호,m.mo_title 영화제목,m.mo_actor
배우,m.mo_gen 장르,m.mo_super 감독,r.re_text 내용
FROM movie m, review r
WHERE m.mo_num=r.mo_num order by m.mo_num;
```

영화 정보 조회 쿼리

영화 삭제 후

영...	영화제목	배우	장르	감독	내용
4	너의 이름은	카미키 애니	신카이	엄마랑 편하게 재밌게 봤어요	
4	너의 이름은	카미키 애니	신카이	남자친구와 처음 본 영화인데 재밌었어요	
4	너의 이름은	카미키 애니	신카이	긴장감있고 쫄나게 보았네요	
4	너의 이름은	카미키 애니	신카이	꿀잼입니당 완전 꿀꿀잼이에용	
4	너의 이름은	카미키 애니	신카이	보는 내내 불안불안함	
4	너의 이름은	카미키 애니	신카이	어후 스트레스 받아....	
4	너의 이름은	카미키 애니	신카이	좁은공간에서 어떻게 이런 연출을 하는지 대단.. 꿀잼	
5	쿵푸허슬	주성치	코미디	주성치	핸드폰에 우리의 모든 삶이 다 있다.
5	쿵푸허슬	주성치	코미디	주성치	누구나 감추고싶은 비밀이 있다
5	쿵푸허슬	주성치	코미디	주성치	코믹하면서도 생각 해볼만한 내용이 많은 영화!
5	쿵푸허슬	주성치	코미디	주성치	생각없이 봤는데 재밌네요
5	쿵푸허슬	주성치	코미디	주성치	다큐영화 같았네요 지루함.
5	쿵푸허슬	주성치	코미디	주성치	다 똑같은 세상을 살고있겠찌?
5	쿵푸허슬	주성치	코미디	주성치	배우들 연기가 좋아서 몰입이 잘됐어요.
5	쿵푸허슬	주성치	코미디	주성치	재미로만 보면 이 영화가 천만 영화가 되어야 마땅하다.

```
DELETE MOVIE WHERE mo_num=6;
6번영화 '박쥐' 종영하여 삭제 진행 하였습니다.
리뷰테이블에 mo_num(영화번호)는 외래키로
ON DELETE CASCADE 되어 리뷰조회시
6번영화 리뷰 전부 같이 삭제 된걸 확인하였습니다.
```

추가 기능

결제 정보 뷰 생성 쿼리

```
CREATE OR REPLACE VIEW EW_pay_info
AS
    SELECT p.ti_num T_NUM
           , c.cos_name T_NAMES
           , s.sh_time T_TIMES
           , m.mo_title M_TITLES
           , p.py_type P_TYPES
           , (t.ti_nop)*p.py_tot P_TOPAY
    FROM pay p
    JOIN ticketing t ON p.ti_num = t.ti_num
    JOIN customer c ON t.cos_num = c.cos_num
    JOIN show_day s ON t.sh_num = s.sh_num
    JOIN movie m ON s.sh_movien = m.mo_num ;

SELECT * FROM EW_pay_info;
```

결제 정보 뷰 확인

⚡ T_NUM	⚡ T_NAMES	⚡ T_TIMES	⚡ M_TITLES	⚡ P_TYPES	⚡ P_TOPAY
10	김나라	23/09/13	통증	상품권	60000
1	김나라	23/09/13	통증	카드	60000
13	밤나라	23/09/13	천박사 퇴마 연구소	삼성페이	30000
11	밤나라	23/09/13	베니스 유령 살인사건	네이버페이	75000
2	밤나라	23/09/13	베니스 유령 살인사건	카드	75000
12	강나라	23/09/13	너의 이름은	카드	15000
3	강나라	23/09/13	너의 이름은	삼성페이	15000
4	집가라	23/09/13	천박사 퇴마 연구소	삼성페이	30000
5	최춘식	23/09/13	쿵푸허슬	카드	90000
6	솔방울	23/09/13	베니스 유령 살인사건	카카오페이	60000
7	하도녀	23/09/13	쿵푸허슬	애플페이	60000
8	최범죄	23/09/13	너의 이름은	삼성페이	30000
9	옥도시	23/09/13	쿵푸허슬	상품권	45000

추가 기능

프로시저 쿼리

```
CREATE OR REPLACE PROCEDURE payProc
(
    pay_no IN EW_pay_info.T_NUM%TYPE
)
AS
    v_payNUMBER NUMBER;
    v_payNAME varchar2(20);
    v_payTIME DATE;
    v_payTITLE varchar2(50);
    v_payTYPE varchar2(20);
    v_payTOPAY NUMBER;

begin
    SELECT T_NUM into v_payNUMBER from
        EW_pay_info WHERE T_NUM=pay_no;
    SELECT T_NAMES into v_payNAME from
        EW_pay_info WHERE T_NUM=pay_no;
    SELECT T_TIMES into v_payTIME from
        EW_pay_info WHERE T_NUM=pay_no;
    SELECT M_TITLES into v_payTITLE from
        EW_pay_info WHERE T_NUM=pay_no;
```

프로시저 쿼리

```
SELECT P_TYPES into v_payTYPE from
    EW_pay_info WHERE T_NUM=pay_no;
SELECT P_TOPAY into v_payTOPAY from
    EW_pay_info WHERE T_NUM=pay_no;

DBMS_OUTPUT.PUT_LINE('----- 결제 완료 -----');
DBMS_OUTPUT.PUT_LINE('티켓번호 : ' || v_payNUMBER);
DBMS_OUTPUT.PUT_LINE('회원이름 : ' || v_payNAME);
DBMS_OUTPUT.PUT_LINE('상영날짜 : ' || v_payTIME);
DBMS_OUTPUT.PUT_LINE('영화이름 : ' || v_payTITLE);
DBMS_OUTPUT.PUT_LINE('결제타입 : ' || v_payTYPE);
DBMS_OUTPUT.PUT_LINE('결제총액 : ' || v_payTOPAY);

END payProc ;

프로시저 호출

EXEC payProc(7);
```

01
02
03
04
05
06
07
08

추가 기능

프로시저 기능 결과

----- 결제 완료 -----

티켓번호 : 5

회원이름 : 최춘식

상영날짜 : 23/09/13

영화이름 : 쿵푸허슬

결제타입 : 카드

결제총액 : 90000

PL/SQL 프로시저가 성공적으로 완료되었습니다.

프로시저 설명

CREATE 구문중 ()괄호 안에 있는
pay_no는 입력 파라미터 입니다.
특정 프로시저를 호출 할 때 입력 받을 파라미터를
먼저 지정해 주어야 됩니다.
AS 구문 뒤에는 출력할 매개변수들의
데이터 타입과 변수명을 지정해 줍니다.
begin ~ end 사이의 구문은 실제 처리할 내용들을 적습니다.
select ~ into 구문을 사용하여 EM_pay_info 뷰에
컬럼 별칭 T_NUM, T_NAMES, T_TIMES, M_TITLES, P_TYPE,
P_TOPAY를 각각 선언한 변수에 넣어줍니다.
DBMS_OUTPUT.PUT_LINE 을 사용 하여
해당 변수에 들어간 값을 출력하도록 지정해 놓습니다.
해당 프로시저를 사용할 때엔
지정해놓은 매개변수를 이용하여 뷰의 T_NUM을 입력 시킵니다.

프로시저는 일치된 NUM의 결제정보를 화면에 출력해줍니다.
학원에서 배운 영역은 아니지만,
트리거와 자료를 찾다보니 알게되어 사용해 보았습니다.
저장 프로시저같은 경우 다른 외부 값을 받아
함수처럼 처리도 할 수 있고
리턴값을 출력해주는 기능 등 다양한 기능이 있어
알아두면 좋을 것 같습니다
아직 해당 프로시저라는 개념을 100% 핸들링 하지는 못하지만
이번 기회에 활용해보아서 너무 뿌듯합니다
-가을 외 3명-

01
02
03
04
05
06
07
08

CHUNSIC

트리거 및 프로시저를 구현
하며 DB공부에
좋은 기회가 되었습니다.
팀 워크 또한
훌륭해서 너무 좋았습니다
다음 조는 다음에...

SOWON

프로젝트를 진행 하면서
트리거를 구현 해 볼 수
있어서 좋은 기회가 되었습
니다.
팀워크가 너무 좋아서
수월하게 진행 할 수
있었습니다

DONG

주제를 같이 정하고
회의 하며 여러 기능 구현을
해보며 협업을 배웠습니다
쿼리를 작성 하며
테스트를 해보면서
배웠던 부분을 복기하여
너무 좋았습니다.

V12

팀 워크가 훌륭하여
너무 좋았으며
배워보지 못한 기능들을
함께 공부 할 수 있는
계기가 되어 너무
즐거웠습니다.

질 문 하 지 말 아 조



THANK YOU

감사합니다.