

Movie Theater Database

가상의 요구사항 분석부터 테이블 설계 및 추출

한 규 원

목차

01

요구사항 수집 및 분석

02

Database 설계

03

SQL을 활용한 데이터 추출

01 | 요구사항 수집 및 분석

사용자의 요구사항을 수집하고 분석하여 데이터베이스의 용도를 파악

1. 요구사항 수집 및 분석

회원관련 요구사항

- 각 회원에 대해서 회원 번호, 이름, 전화번호, 주민등록번호, 아이디, 비밀번호, 포인트를 저장한다. 주민등록번호 앞자리를 이용하여 영화관람 나이제한을 확인하여 입장 시킬 수 있음.
- 티켓을 예매할 때마다 포인트 적립이 되고 예매 수단, 일자, 방법이 저장되고 관람한 영화에 대해서 평점을 남길 수 있음.

티켓관련 요구사항

- 회원이 예매한 티켓에는 티켓번호, 영화제목, 상영날짜, 상영관, 위치 좌석번호, 가격이 저장되어 있으며, 한 명의 회원이 여러 장의 티켓을 구매할 수 있고 한 영화의 티켓을 여러명이 구매할 수 있음.

영화관련 요구사항

- 영화에는 고유번호, 제목, 장르, 감독, 상영시간, 관람나이가 저장되어 있고 어떠한 영화에 출연하면 배우는 역할을 받으며 배우 정보는 고유번호, 이름, 나이, 키 저장되어 있으며 한 명의 배우는 여러 개의 영화에 출연 가능, 한 영화에 여러 명의 배우가 출연함.

상품관련 요구사항

- 여러 명의 회원이 매점에서 여러 개의 상품을 구매할 수 있는데 상품에는 상품 고유번호, 이름, 가격이 저장되어 있으며 주문할 때 상품의 수량을 결정할 수 있음.

이벤트관련 요구사항

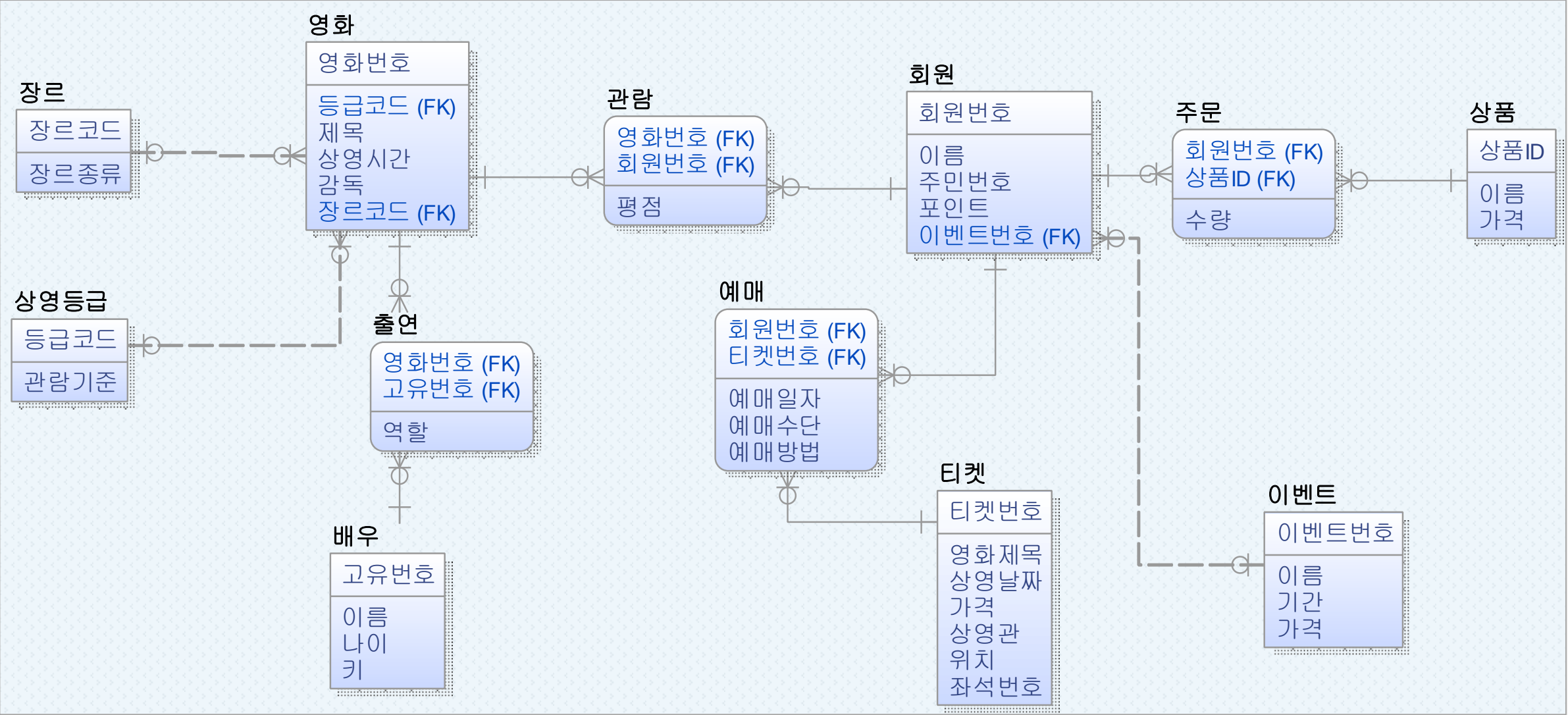
- 이벤트를 진행할 때, 한 명의 회원 당 한 개의 이벤트 상품을 구매하는 것으로 제한 함.
- 이벤트 정보에는 고유번호, 프로모션이름, 프로모션 기간, 가격 정보가 있음.

02 | Database 설계

1. 개념적 설계
2. 논리적 설계
3. 물리적 설계

1. 개념적 설계

요구사항 분석 결과물을 개념적 데이터 모델을 이용해 구조로 표현 (Entity - Relation Diagram)



2. 논리적 설계

개념적 스키마를 논리적 데이터 모델을 이용해 논리적 구조로 표현

티켓 Relation

<u>티켓번호</u>	영화제목	상영날짜	가격	위치	상영관	좌석번호
-------------	------	------	----	----	-----	------

티켓 Relation 데이터 타입 및 크기 설정

티켓번호 char(20) not null

영화제목 char(50)

상영날짜 int

가격 int

위치 char(10)

상영관 char(10)

좌석번호 char(20)

primary key(티켓번호)

이벤트 Relation

<u>이벤트번호</u>	이름	가격	기간
--------------	----	----	----

이벤트 Relation 데이터 타입 및 크기 설정

이벤트번호 char(10) not null

이름 char(50)

가격 int

기간 char(50)

primary key(이벤트번호)

2. 논리적 설계

개념적 스키마를 논리적 데이터 모델을 이용해 논리적 구조로 표현

상품 Relation

<u>상품ID</u>	이름	가격
-------------	----	----

상품 Relation 데이터 타입 및 크기 설정

상품ID char(20) not null

이름 char(20)

가격 int

primary key(상품ID)

배우 Relation

<u>고유번호</u>	이름	나이	키
-------------	----	----	---

배우 Relation 데이터 타입 및 크기 설정

고유번호 char(10) not null

이름 char(20)

나이 int

키 int

Primary key(고유번호)

check(나이 >= 0)

2. 논리적 설계

개념적 스키마를 논리적 데이터 모델을 이용해 논리적 구조로 표현

상영등급 Relation

<u>등급코드</u>	관람기준
-------------	------

상영등급 Relation 데이터 타입 및 크기 설정

등급코드 char(20) not null

관람기준 char(20)

primary key(등급코드)

check(관람기준 in ('전체', '7세', '12세', '15세', '19세'))

장르 Relation

<u>장르코드</u>	장르종류
-------------	------

장르 Relation 데이터 타입 및 크기 설정

장르코드 char(20) not null

장르종류 char(30)

Primary key(장르코드)

2. 논리적 설계

개념적 스키마를 논리적 데이터 모델을 이용해 논리적 구조로 표현

영화 Relation

영화번호	제목	감독	상영시간	등급코드	장르코드
------	----	----	------	------	------

영화 Relation 데이터 타입 및 크기 설정

영화번호 char(20) not null

제목 char(50)

감독 char(50)

상영시간 int

등급코드 char(20)

장르코드 char(20)

primary key(영화번호)

foreign key(등급코드) references 상영등급(등급코드)

foreign key(장르코드) references 장르(장르코드)

회원 Relation

회원번호	이름	주민번호	포인트	이벤트번호
------	----	------	-----	-------

회원 Relation 데이터 타입 및 크기 설정

회원번호 char(20) not null

이름 char(20)

주민번호 int

포인트 int

이벤트번호 char(10)

primary key(회원번호)

foreign key(이벤트번호) references 이벤트(이벤트번호)

2. 논리적 설계

개념적 스키마를 논리적 데이터 모델을 이용해 논리적 구조로 표현

출연 Relation

<u>영화번호</u>	<u>고유번호</u>	역할
-------------	-------------	----

출연 Relation 데이터 타입 및 크기 설정

영화번호 char(20) not null

고유번호 char(10) not null

역할 char(50)

primary key(영화번호, 고유번호)

foreign key(영화번호) references 영화(영화번호)

foreign key(고유번호) references 배우(고유번호)

관람 Relation

<u>회원번호</u>	<u>영화번호</u>	평점
-------------	-------------	----

관람 Relation 데이터 타입 및 크기 설정

회원번호 char(20) not null

영화번호 char(20) not null

평점 int

primary key(회원번호, 영화번호)

foreign key(회원번호) references 회원(회원번호)

foreign key(영화번호) references 영화(영화번호)

check(평점 >= 0 and 평점 <=5)

2. 논리적 설계

개념적 스키마를 논리적 데이터 모델을 이용해 논리적 구조로 표현

주문 Relation

<u>주문번호</u>	회원번호	상품ID	수량
-------------	------	------	----

주문 Relation 데이터 타입 및 크기 설정

주문번호 int not null

회원번호 char(20)

상품ID char(20)

수량 int

primary key(주문번호)

foreign key(회원번호) references 회원(회원번호)

foreign key(상품ID) references 상품(상품ID)

회원 Relation

<u>예매번호</u>	회원번호	티켓번호	예매일자	예매수단	예매방법
-------------	------	------	------	------	------

회원 Relation 데이터 타입 및 크기 설정

예매번호 char(20) not null

회원번호 char(20)

티켓번호 char(20)

예매일자 int

예매수단 char(10)

예매방법 char(10)

primary key(예매번호)

foreign key(회원번호) references 회원(회원번호)

foreign key(티켓번호) references 티켓(티켓번호)

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

티켓 테이블

```
/* 티켓 테이블 생성 */
create table 티켓(
    티켓번호 char(20) not null,
    영화제목 char(50),
    상영날짜 int,
    가격 int,
    위치 char(10),
    상영관 char(10),
    좌석번호 char(20),
    primary key(티켓번호)
);
```

```
insert into 티켓 values('T001','베데랑',200601,6000,'6층','1관','D-14');
insert into 티켓 values('T002','국제시장',200602,6000,'6층','2관','E-01');
insert into 티켓 values('T003','7번방의선물',200528,7000,'5층','컴포트4관','J-23');
insert into 티켓 values('T004','신과함께',200524,6000,'6층','3관','C-07');
insert into 티켓 values('T005','명량',200601,6000,'6층','6관','F-21');
insert into 티켓 values('T006','극한직업',200523,6000,'6층','8관','D-15');
insert into 티켓 values('T007','도둑들',200513,6000,'5층','7관','G-16');
insert into 티켓 values('T008','베데랑',200526,6000,'6층','1관','E-12');
insert into 티켓 values('T009','명량',200529,6000,'6층','6관','C-06');
insert into 티켓 values('T010','명량',200601,6000,'6층','6관','J-18');
```

```
select *
from 티켓;
```

119 %

결과 메시지

	티켓번호	영화제목	상영날짜	가격	위치	상영관	좌석번호
1	T001	베데랑	200601	6000	6층	1관	D-14
2	T002	국제시장	200602	6000	6층	2관	E-01
3	T003	7번방의선물	200528	7000	5층	컴포트4관	J-23
4	T004	신과함께	200524	6000	6층	3관	C-07
5	T005	명량	200601	6000	6층	6관	F-21
6	T006	극한직업	200523	6000	6층	8관	D-15
7	T007	도둑들	200513	6000	5층	7관	G-16
8	T008	베데랑	200526	6000	6층	1관	E-12
9	T009	명량	200529	6000	6층	6관	C-06
10	T010	명량	200601	6000	6층	6관	J-18

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

이벤트 테이블

/* 이벤트 테이블 생성 */

```
CREATE TABLE 이벤트(  
    이벤트번호 CHAR(10) NOT NULL,  
    이름 CHAR(50),  
    가격 INT,  
    기간 CHAR(50),  
    PRIMARY KEY (이벤트번호),  
);
```

```
insert into 이벤트 values( 'E001', '2020다이어리', 10000, '191200-200130');  
insert into 이벤트 values( 'E002', '극한집업통', 13000, '200101-200130');  
insert into 이벤트 values( 'E003', '이순신활', 17000, '200201-200215');
```

```
select *  
from 이벤트;
```

119 %

결과 메시지

	이벤트번호	이름	가격	기간
1	E001	2020다이어리	10000	191200-200130
2	E002	극한집업통	13000	200101-200130
3	E003	이순신활	17000	200201-200215

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

회원 테이블

```
/*회원 테이블 생성*/  
create table 회원(  
    회원번호 char(20) not null,  
    이름 char(20),  
    주민번호 INT,  
    포인트 INT,  
    이벤트번호 char(10),  
    primary key (회원번호),  
);
```

```
insert into 회원 values ('P001', '홍길동', 951123, 9000, NULL);  
insert into 회원 values ('P002', '이수진', 910214, 7900, NULL);  
insert into 회원 values ('P003', '최동혜', 900317, 11000, 'E001');  
insert into 회원 values ('P004', '이경석', 941002, 700, 'E003');  
insert into 회원 values ('P005', '하재강', 960616, 4500, NULL);  
insert into 회원 values ('P006', '안중원', 970823, 6000, 'E002');  
insert into 회원 values ('P007', '신수지', 991101, NULL, 'E002');  
insert into 회원 values ('P008', '박지민', 911225, 11500, NULL);  
insert into 회원 values ('P009', '이수민', 921123, NULL, NULL);  
insert into 회원 values ('P010', '안혜림', 870205, 10000, 'E001');
```

```
select *  
from 회원;
```

119 %

	회원번호	이름	주민번호	포인트	이벤트번호
1	P001	홍길동	951123	9000	NULL
2	P002	이수진	910214	7900	NULL
3	P003	최동혜	900317	11000	E001
4	P004	이경석	941002	700	E003
5	P005	하재강	960616	4500	NULL
6	P006	안중원	970823	6000	E002
7	P007	신수지	991101	NULL	E002
8	P008	박지민	911225	11500	NULL
9	P009	이수민	921123	NULL	NULL
10	P010	안혜림	870205	10000	E001

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

상품 테이블

```
/* 상품 테이블 생성 */  
create table 상품(  
    상품ID CHAR(20) NOT NULL,  
    이름 CHAR(20),  
    가격 INT,  
    PRIMARY KEY(상품ID)  
);
```

```
insert into 상품 values('C001', '일반팝콘', 5000);  
insert into 상품 values('C002', '카라멜팝콘', 6000);  
insert into 상품 values('C003', '갈릭팝콘', 6000);  
insert into 상품 values('C004', '치즈팝콘', 6000);  
insert into 상품 values('C005', '콜라', 2500);  
insert into 상품 values('C006', '사이다', 2500);  
insert into 상품 values('C007', '환타', 2500);  
insert into 상품 values('C008', '통짬어', 10000);  
insert into 상품 values('C009', '순살치킨', 9000);  
insert into 상품 values('C010', '포테이토', 9000);  
  
select *  
from 상품;
```

119 %

결과 메시지

	상품ID	이름	가격
1	C001	일반팝콘	5000
2	C002	카라멜팝콘	6000
3	C003	갈릭팝콘	6000
4	C004	치즈팝콘	6000
5	C005	콜라	2500
6	C006	사이다	2500
7	C007	환타	2500
8	C008	통짬어	10000
9	C009	순살치킨	9000
10	C010	포테이토	9000

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

배우 테이블

```
/* 배우 테이블 생성 */  
create table 배우(  
    고유번호 char(10) not null,  
    이름 char(20),  
    나이 int,  
    키 int,  
    primary key(고유번호),  
    check(나이>=0)  
);
```

```
insert into 배우 values ('A001', '김윤석', 53, 178);  
insert into 배우 values ('A002', '이정재', null, 180);  
insert into 배우 values ('A003', '황정민', null, 180);  
insert into 배우 values ('A004', '김윤진', 48, 168);  
insert into 배우 values ('A005', '최민식', 59, 177);  
insert into 배우 values ('A006', '류승룡', null, null);  
insert into 배우 values ('A007', '이하늬', null, null);  
insert into 배우 values ('A008', '하정우', null, 184);  
insert into 배우 values ('A009', '차태현', null, null);  
insert into 배우 values ('A010', '유아인', null, null);  
insert into 배우 values ('A011', '박신혜', 31, 168);  
insert into 배우 values ('A012', '오달수', 53, 176);
```

```
select*  
from 배우;
```

	고유번호	이름	나이	키
1	A001	김윤석	53	178
2	A002	이정재	NULL	180
3	A003	황정민	NULL	180
4	A004	김윤진	48	168
5	A005	최민식	59	177
6	A006	류승룡	NULL	NULL
7	A007	이하늬	NULL	NULL
8	A008	하정우	NULL	184
9	A009	차태현	NULL	NULL
10	A010	유아인	NULL	NULL
11	A011	박신혜	31	168
12	A012	오달수	53	176

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

상영등급 테이블

```
/* 상영등급 테이블 생성 */  
create table 상영등급(  
    등급코드 char(20) not null,  
    관람기준 char(20),  
    primary key(등급코드),  
    check(관람기준 in('전체', '7세', '12세', '15세', '19세'))  
);
```

```
insert into 상영등급 values ('D1', '전체');  
insert into 상영등급 values ('D2', '7세');  
insert into 상영등급 values ('D3', '12세');  
insert into 상영등급 values ('D4', '15세');  
insert into 상영등급 values ('D5', '19세');
```

```
select *  
from 상영등급;
```

119 %

결과 메시지

	등급코드	관람기준
1	D1	전체
2	D2	7세
3	D3	12세
4	D4	15세
5	D5	19세

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

장르 테이블

```
/* 장르 테이블 생성 */  
create table 장르(  
    장르코드 char(20) not null,  
    장르종류 char(30),  
    primary key(장르코드)  
);
```

```
insert into 장르 values ('J001', '액션');  
insert into 장르 values ('J002', '코미디');  
insert into 장르 values ('J003', '드라마');  
insert into 장르 values ('J004', '판타지');
```

```
select *  
from 장르;
```

119 %

결과 메시지

	장르코드	장르종류
1	J001	액션
2	J002	코미디
3	J003	드라마
4	J004	판타지

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

영화 테이블

```
/* 영화 테이블 생성 */  
create table 영화(  
    영화번호 char(20) not null,  
    제목 char(50),  
    감독 char(50),  
    상영시간 int,  
    등급코드 char(20),  
    장르코드 char(20),  
    primary key(영화번호),  
    foreign key(등급코드) references 상영등급(등급코드),  
    foreign key(장르코드) references 장르(장르코드)  
);
```

```
insert into 영화 values ('M001', '명량', '김한민', 118, 'D4', 'J001');  
insert into 영화 values ('M002', '극한직업', '이병헌', 111, 'D4', 'J002');  
insert into 영화 values ('M003', '신과함께', '김용화', 139, 'D3', 'J004');  
insert into 영화 values ('M004', '7번방의 선물', '이환경', 127, 'D4', 'J002');  
insert into 영화 values ('M005', '국제시장', '윤제균', 126, 'D3', 'J003');  
insert into 영화 values ('M006', '베테랑', '류승완', 123, 'D4', 'J001');  
insert into 영화 values ('M007', '도둑들', '최동훈', 135, 'D4', 'J001');
```

```
select *  
from 영화;
```

119 %

결과 메시지

	영화번호	제목	감독	상영시간	등급코드	장르코드
1	M001	명량	김한민	118	D4	J001
2	M002	극한직업	이병헌	111	D4	J002
3	M003	신과함께	김용화	139	D3	J004
4	M004	7번방의 선물	이환경	127	D4	J002
5	M005	국제시장	윤제균	126	D3	J003
6	M006	베테랑	류승완	123	D4	J001
7	M007	도둑들	최동훈	135	D4	J001

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

출연 테이블

```
/* 출연 테이블 생성 */
create table 출연(
    영화번호 char(20) not null,
    고유번호 char(10) not null,
    역할 char(50),
    primary key (영화번호, 고유번호),
    foreign key (영화번호) references 영화(영화번호),
    foreign key (고유번호) references 배우(고유번호)
);
```

```
insert into 출연 values ('M001', 'A005', '이순신');
insert into 출연 values ('M001', 'A006', '구루지마');
insert into 출연 values ('M002', 'A006', '고반장');
insert into 출연 values ('M002', 'A007', '장형사');
insert into 출연 values ('M003', 'A008', '강림');
insert into 출연 values ('M003', 'A009', '자홍');
insert into 출연 values ('M004', 'A006', '용구');
insert into 출연 values ('M004', 'A011', '큰예승');
insert into 출연 values ('M004', 'A012', '소알호');
insert into 출연 values ('M005', 'A003', '덕수');
insert into 출연 values ('M005', 'A004', '영자');
insert into 출연 values ('M005', 'A012', '탈구');
insert into 출연 values ('M006', 'A003', '서도철');
insert into 출연 values ('M006', 'A010', '오팀장');
insert into 출연 values ('M006', 'A012', '초태오');
insert into 출연 values ('M007', 'A001', '마카오');
insert into 출연 values ('M007', 'A002', '뽀빠이');
```

select * from 출연;

	영화번호	고유번호	역할
1	M001	A005	이순신
2	M001	A006	구루지마
3	M002	A006	고반장
4	M002	A007	장형사
5	M003	A008	강림
6	M003	A009	자홍
7	M004	A006	용구
8	M004	A011	큰예승
9	M004	A012	소알호
10	M005	A003	덕수
11	M005	A004	영자
12	M005	A012	탈구
13	M006	A003	서도철
14	M006	A010	오팀장
15	M006	A012	초태오
16	M007	A001	마카오
17	M007	A002	뽀빠이

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

관람 테이블

```
/* 관람 테이블 생성 */  
create table 관람(  
    회원번호 char(20) not null,  
    영화번호 char(20) not null,  
    평점 int,  
    primary key (회원번호, 영화번호),  
    foreign key (회원번호) references 회원(회원번호),  
    foreign key (영화번호) references 영화(영화번호),  
    check(평점 >= 0 and 평점 <= 5)  
);
```

```
insert into 관람 values ('P001', 'M001', 5);  
insert into 관람 values ('P010', 'M002', 4);  
insert into 관람 values ('P004', 'M003', 5);  
insert into 관람 values ('P006', 'M004', 5);  
insert into 관람 values ('P003', 'M002', 5);  
insert into 관람 values ('P003', 'M007', 4);  
insert into 관람 values ('P001', 'M005', 5);  
insert into 관람 values ('P007', 'M006', 3);  
insert into 관람 values ('P008', 'M003', 4);  
insert into 관람 values ('P009', 'M004', 5);  
  
select *  
from 관람;
```

119 %

결과 메시지

	회원번호	영화번호	평점
1	P001	M001	5
2	P001	M005	5
3	P003	M002	5
4	P003	M007	4
5	P004	M003	5
6	P006	M004	5
7	P007	M006	3
8	P008	M003	4
9	P009	M004	5
10	P010	M002	4

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

주문 테이블

```
/*주문 테이블 생성*/  
create table 주문(  
    주문번호 int not null,  
    회원번호 char(20),  
    상품ID char(20),  
    수량 int,  
    PRIMARY KEY (주문번호),  
    foreign key (회원번호) references 회원(회원번호),  
    foreign key (상품ID) references 상품(상품ID)  
);
```

```
insert into 주문 values (1, 'P001', 'C002', 2);  
insert into 주문 values (2, 'P001', 'C005', 2);  
insert into 주문 values (3, 'P010', 'C010', 1);  
insert into 주문 values (4, 'P007', 'C001', 1);  
insert into 주문 values (5, 'P007', 'C007', 2);  
insert into 주문 values (6, 'P003', 'C005', 4);  
insert into 주문 values (7, 'P002', 'C010', 2);  
insert into 주문 values (8, 'P005', 'C004', 3);  
insert into 주문 values (9, 'P005', 'C005', 3);  
  
select *  
from 주문;
```

119 %

결과 메시지

	주문번호	회원번호	상품ID	수량
1	1	P001	C002	2
2	2	P001	C005	2
3	3	P010	C010	1
4	4	P007	C001	1
5	5	P007	C007	2
6	6	P003	C005	4
7	7	P002	C010	2
8	8	P005	C004	3
9	9	P005	C005	3

3. 물리적 설계

Database Management System(DBMS)로 구현 가능한 물리적 구조 설계

예매 테이블

```
/* 예매 테이블 생성 */  
create table 예매(  
    예매번호 char(20) not null,  
    회원번호 char(20),  
    티켓번호 char(20),  
    예매일자 int,  
    예매수단 char(10),  
    예매방법 char(10),  
    primary key (예매번호),  
    foreign key (회원번호) references 회원(회원번호),  
    foreign key (티켓번호) references 티켓(티켓번호)  
);
```

```
insert into 예매 values ('R001', 'P001', 'T002', 200425, '현장', '현금');  
insert into 예매 values ('R002', 'P010', 'T001', 200523, '온라인', '카드');  
insert into 예매 values ('R003', 'P002', 'T008', 200521, '현장', '카드');  
insert into 예매 values ('R004', 'P009', 'T010', 200525, '온라인', '카드');  
insert into 예매 values ('R005', 'P003', 'T003', 200526, '온라인', '카드');  
insert into 예매 values ('R006', 'P005', 'T005', 200529, '온라인', '카드');  
insert into 예매 values ('R007', 'P006', 'T004', 200530, '온라인', '카드');  
insert into 예매 values ('R008', 'P004', 'T009', 200601, '온라인', '카드');  
insert into 예매 values ('R009', 'P001', 'T006', 200524, '현장', '현금');  
insert into 예매 values ('R010', 'P008', 'T008', 200520, '온라인', '카드');  
insert into 예매 values ('R011', 'P007', 'T007', 200521, '현장', '현금');  
select * from 예매;
```

	예매번호	회원번호	티켓번호	예매일자	예매수단	예매방법
1	R001	P001	T002	200425	현장	현금
2	R002	P010	T001	200523	온라인	카드
3	R003	P002	T008	200521	현장	카드
4	R004	P009	T010	200525	온라인	카드
5	R005	P003	T003	200526	온라인	카드
6	R006	P005	T005	200529	온라인	카드
7	R007	P006	T004	200530	온라인	카드
8	R008	P004	T009	200601	온라인	카드
9	R009	P001	T006	200524	현장	현금
10	R010	P008	T008	200520	온라인	카드
11	R011	P007	T007	200521	현장	현금

03 | SQL을 활용한 데이터 추출

SQL을 활용한 데이터 추출

```
/* 가장 많이 판매된 상품순으로 순위와 상품명, 수량을 구하시오 */  
select rank() over (order by SUM(수량) desc) as 순위, 상품.이름, sum(수량) as '판매된 수량'  
from 주문, 상품  
where 주문.상품ID=상품.상품ID  
group by 상품.이름;
```

119 %

결과 메시지

	순위	이름	판매된 수량
1	1	콜라	9
2	2	포테이토	3
3	2	치즈팝콘	3
4	4	카라멜팝콘	2
5	4	환타	2
6	6	일반팝콘	1

```
/* 평점 5점 이상 받은 영화제목는? */  
select 제목  
from 영화  
where 영화.영화번호 in (select 영화번호  
from 관람  
where 평점 <= 5);
```

119 %

결과 메시지

	제목
1	명량
2	극한직업
3	신과함께
4	7번방의 선물
5	국제시장
6	베테랑
7	도둑들

SQL을 활용한 데이터 추출

```
/* 회원중에 포인트가 많은순으로 3등까지 이름과 포인트, 이벤트참여여부를 출력하시오  
(이벤트참여하면 참여 참여 안하면 불참 출력)*/  
select TOP(3) 이름,포인트,case when 이벤트번호 is null then '불참' else '참여' end as '이벤트참여여부'  
from 회원  
ORDER BY 포인트 DESC;
```

119 %

결과 메시지

	이름	포인트	이벤트참여여부
1	박지민	11500	불참
2	최동해	11000	참여
3	안혜림	10000	참여

```
/* 현금으로 결제한 회원수와 카드로 결제한 회원수는? */  
select 예매방법,count(*) as '회원수'  
from 회원,예매  
where 회원.회원번호=예매.회원번호  
group by 예매방법;
```

119 %

결과 메시지

	예매방법	회원수
1	카드	8
2	현금	3

SQL을 활용한 데이터 추출

```
/* 각회원 별로 구매한 상품의 총가격은? (회원명,총가격을 가격 보기 좋게 내림차순으로 출력하시오) */
select a.이름,sum(가격)*b.수량 as '총 구매 가격'
from 회원 a, 주문 b,상품 c
where a.회원번호=b.회원번호 and b.상품ID = c.상품ID
group by a.이름,b.수량
order by sum(가격)*b.수량 desc;
```

119 %

결과 메시지

	이름	총 구매 가격
1	하재강	25500
2	이수진	18000
3	홍길동	17000
4	최동해	10000
5	안해림	9000
6	신수지	5000
7	신수지	5000

```
/* 홍길동 회원이 이름을 개명해서 안상민이라는 이름으로 바꿨다 데이터를 업데이트 하시오*/
update 회원
set 이름='안상민'
where 이름='홍길동'

select * from 회원;
```

119 %

결과 메시지

	회원번호	이름	주민번호	포인트	이벤트번호
1	P001	안상민	951123	9000	NULL
2	P002	이수진	910214	7900	NULL
3	P003	최동해	900317	11000	E001
4	P004	이경석	941002	700	E003
5	P005	하재강	960616	4500	NULL
6	P006	안종원	970823	6000	E002
7	P007	신수지	991101	NULL	E002
8	P008	박지민	911225	11500	NULL
9	P009	이수민	921123	NULL	NULL
10	P010	안해림	870205	10000	E001

감사합니다