

정보처리 산업기사

조장
서은빈



김종호



윤수빈



김지성

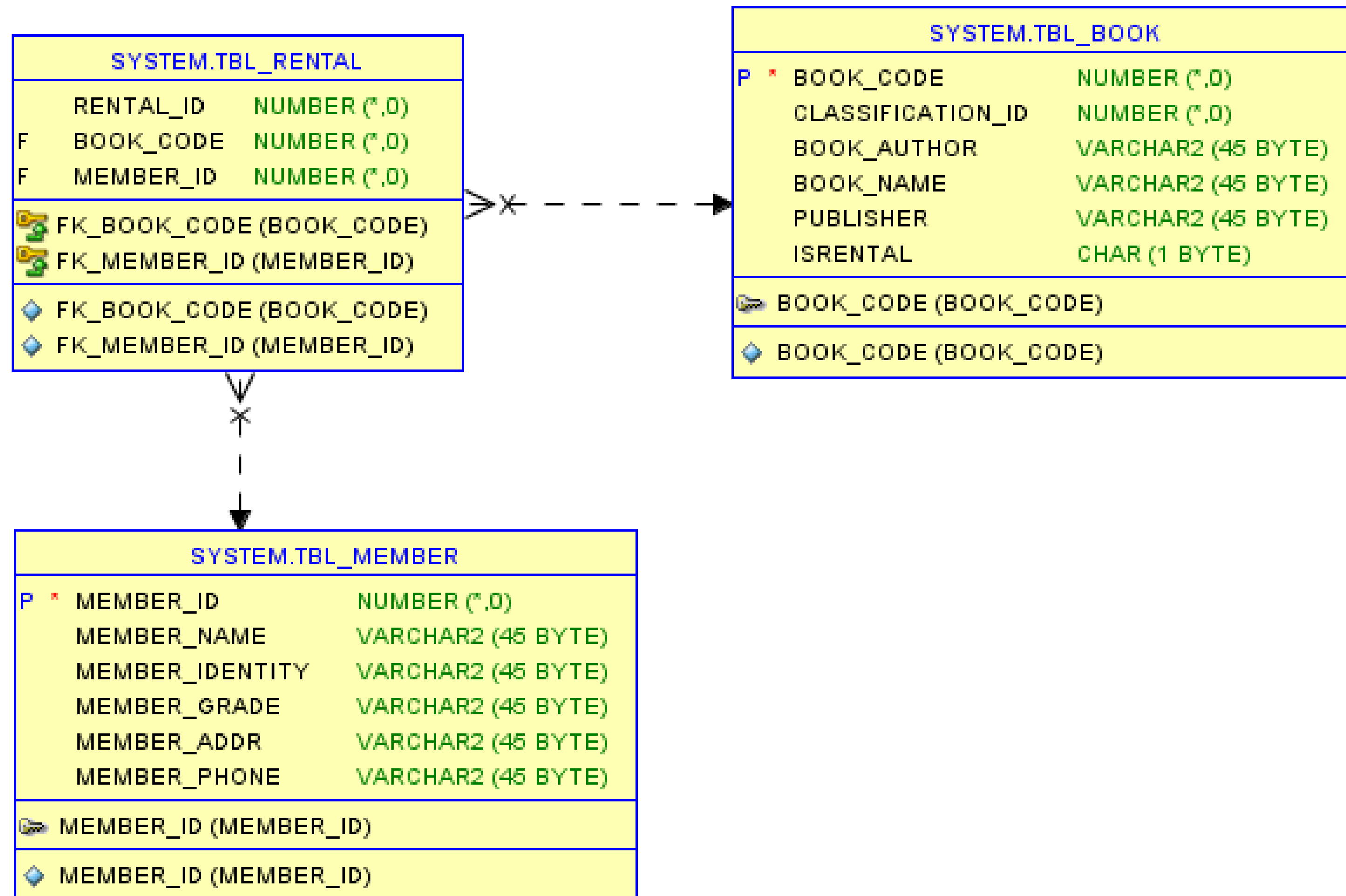


강사님

SQL 활용



전체 ERD



1) tbl_member 의 Member_id 에 Primary key 제약조건 설정하세요. (alter 사용)

Code

```
alter table tbl_member add constraints member_id primary key (member_id);  
desc tbl_member;
```

2)tbl-book의 Book_code 에 primary key 제약조건 설정하세요. (alter 사용)

Code

```
alter table tbl_book add constraints book_code primary key (book_code);  
desc tbl_book;
```

**3)tbl_rental의 Book_code 에 Book_Tbl의 Book_code로 FK 설정하세요(alter사용)
-delete시 cascade 옵션 적용합니다.**

Code

```
alter table tbl_rental  
add constraints fk_book_code foreign key (book_code) references tbl_book (book_code)  
on delete cascade;  
alter table tbl_rental drop constraint fk_book_code;  
desc tbl_rental;
```

4)tbl_rental 의 Member_id에 Member_tbl의 Member_id로 FK 설정하세요 (alter 사용)
-delete시 cascade 옵션 적용합니다.

Code

```
alter table tbl_rental  
add constraints fk_member_id foreign key (member_id) references tbl_member (member_id)  
on delete cascade;  
desc tbl_member;
```

5)각 테이블에 다음 값을 넣어보세요. (tbl_member)

Code

```
insert into tbl_member values(111,'aaa','111','일반','대구','010-111-2222');
insert into tbl_member values(222,'bbb','222','VIP','울산','010-111-2222');
insert into tbl_member values(333,'ccc','333','VIP','인천','010-111-2222');
insert into tbl_member values(444,'ddd','444','일반','부산','010-111-2222');
insert into tbl_member values(555,'eee','555','VIP','서울','010-111-2222');
insert into tbl_member values(666,'fff','666','일반','경기','010-111-2222');
select * from tbl_member;
```

Result

	MEMBER_ID	MEMBER_NAME	MEMBER_IDENTITY	MEMBER_GRADE	MEMBER_ADDR	MEMBER_PHONE
1	111	aaa	111	일반	대구	010-111-2222
2	222	bbb	222	VIP	울산	010-111-2222
3	333	ccc	333	VIP	인천	010-111-2222
4	444	ddd	444	일반	부산	010-111-2222
5	555	eee	555	VIP	서울	010-111-2222
6	666	fff	666	일반	경기	010-111-2222

5)각 테이블에 다음 값을 넣어보세요. (tbl_book)

Code

```
insert into tbl_book values(1010,1,'윤성우','열혈C','오렌지미디어','1');
insert into tbl_book values(1011,1,'남궁성','JAVA의정석','00미디어','1');
insert into tbl_book values(1012,1,'남길동','이것미리눅스다','한빛미디어','1');
insert into tbl_book values(2010,2,'데일카네기','인간관계론','00미디어','1');
insert into tbl_book values(2011,3,'홍길동','미움받을용기','00미디어','1');
select * from tbl_book;
```

Result

	BOOK_CODE	CLASSIFICATION_ID	BOOK_AUTHOR	BOOK_NAME	PUBLISHER	ISRENTAL
1	1010	1	윤성우	열혈C	오렌지미디어	1
2	1011	1	남궁성	JAVA의정석	00미디어	1
3	1012	1	남길동	이것미리눅스다	한빛미디어	1
4	2010	2	데일카네기	인간관계론	00미디어	1
5	2011	3	홍길동	미움받을용기	00미디어	1

5)각 테이블에 다음 값을 넣어보세요. (tbl_book)

Code

```
insert into tbl_rental values (1,1010,111);  
insert into tbl_rental values (2,1011,222);  
insert into tbl_rental values (3,1012,333);  
select * from tbl_rental;
```

Result

	RENTAL_ID	BOOK_CODE	MEMBER_ID
1	1	1010	111
2	2	1011	222
3	3	1012	333

TWO

6) 각 테이블의 제약조건을 확인하세요(사전검색)

- tbl_member의 Primary key 확인이 되어야합니다.
- tbl_book의 Primary key 확인이 되어야합니다.
- tbl_rental의 Foreign key 확인이 되어야합니다.

-tbl_member의 Primary key 확인이 되어야합니다.

Code

```
select constraint_name, constraint_type, search_condition, r_constraint_name
from user_constraints
where table_name = 'TBL_MEMBER';
```

Result

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_CONSTRAINT_NAME
1	MEMBER_ID	P	(null)	(null)

-tbl_book의 Primary key 확인이 되어야합니다.

Code

```
select constraint_name, constraint_type, search_condition, r_constraint_name
from user_constraints
where table_name = 'TBL_BOOK';
```

Result

	⚡ CONSTRAINT_NAME	⚡ CONSTRAINT_TYPE	SEARCH_CONDITION	⚡ R_CONSTRAINT_NAME
1	BOOK_CODE	P	(null)	(null)

-tbl_rental의 Foreign key 확인이 되어야합니다.

Code

```
select constraint_name, constraint_type, search_condition, r_constraint_name
from user_constraints
where table_name = 'TBL_RENTAL';
```

Result

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_CONSTRAINT_NAME
1	FK_BOOK_CODE	R	(null)	BOOK_CODE
2	FK_MEMBER_ID	R	(null)	MEMBER_ID

7) Rental_tbl의 각 FK열에 index를 추가하고 확인해보세요.

Code

```
create index fk_book_code on tbl_rental(book_code);  
create index fk_member_id on tbl_rental(member_id);
```

Result

	INDEX_OWNER	INDEX_NAME	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	COLUMN_LENGTH	CHAR_LENGTH	DESCEND
1	SYSTEM	IDX_RENTAL_MEMBER_ID	SYSTEM	TBL_RENTAL	MEMBER_ID	1	22	0	ASC
2	SYSTEM	IDX_RENTAL_BOOK_CODE	SYSTEM	TBL_RENTAL	BOOK_CODE	1	22	0	ASC

8) 다음과 같이 내용이 나오도록 하는 view테이블을 만들고 사용해 봅니다(조인 + 뷰사용)

Code

```
create view ShowRental_view
as
    select Rental_Id, Member_Name, Book_name
    from tbl_rental
    join tbl_member
    on tbl_rental.member_id = tbl_member.member_id
    join tbl_book
    on tbl_rental.book_code = tbl_book.book_code;
```


8) 다음과 같이 내용이 나오도록 하는 view테이블을 만들고 사용해 봅시다(조인 + 뷰사용)

Result

	RENTAL_ID	MEMBER_NAME	BOOK_NAME
1	1	aaa	열혈C
2	2	bbb	JAVA의정석
3	3	ccc	이것이리눅스다

강사님

서술형

9) 현재 만들어진 ERD에서 1개 이상의 관계형 테이블을 생성하기 위한 시나리오를 서술해보세요.



tbl_grade

시나리오

도서관리 프로그램에서는 도서를 대여할 때마다 적립금이 적립됩니다.
tbl_member의 등급에 따라 적립금률이 정해집니다.
member_grade의 개념을 분리, tbl_grade를 새로 만들어 해당 적립금률과 등급 이름을 관리합니다.

member한명당 하나의 등급 및 적립금률을 가지고, 등급은 하나당 여러 member가 있을 수 있습니다.
tbl_member는 tbl_grade의 grade_id를 외래 키(FK)로 참조하여 N:1 관계를 맺습니다.

등급 테이블은 grade_id를 PK로 가지고, tbl_member는 tbl_grade의 grade_id를 외래키(FK)로 참조합니다.

tbl_grade에서 기존 member_grade의 등급을 grade_name으로 관리합니다.
grade_id에 따라 point_rate가 결정됩니다.

시나리오

tbl_grade

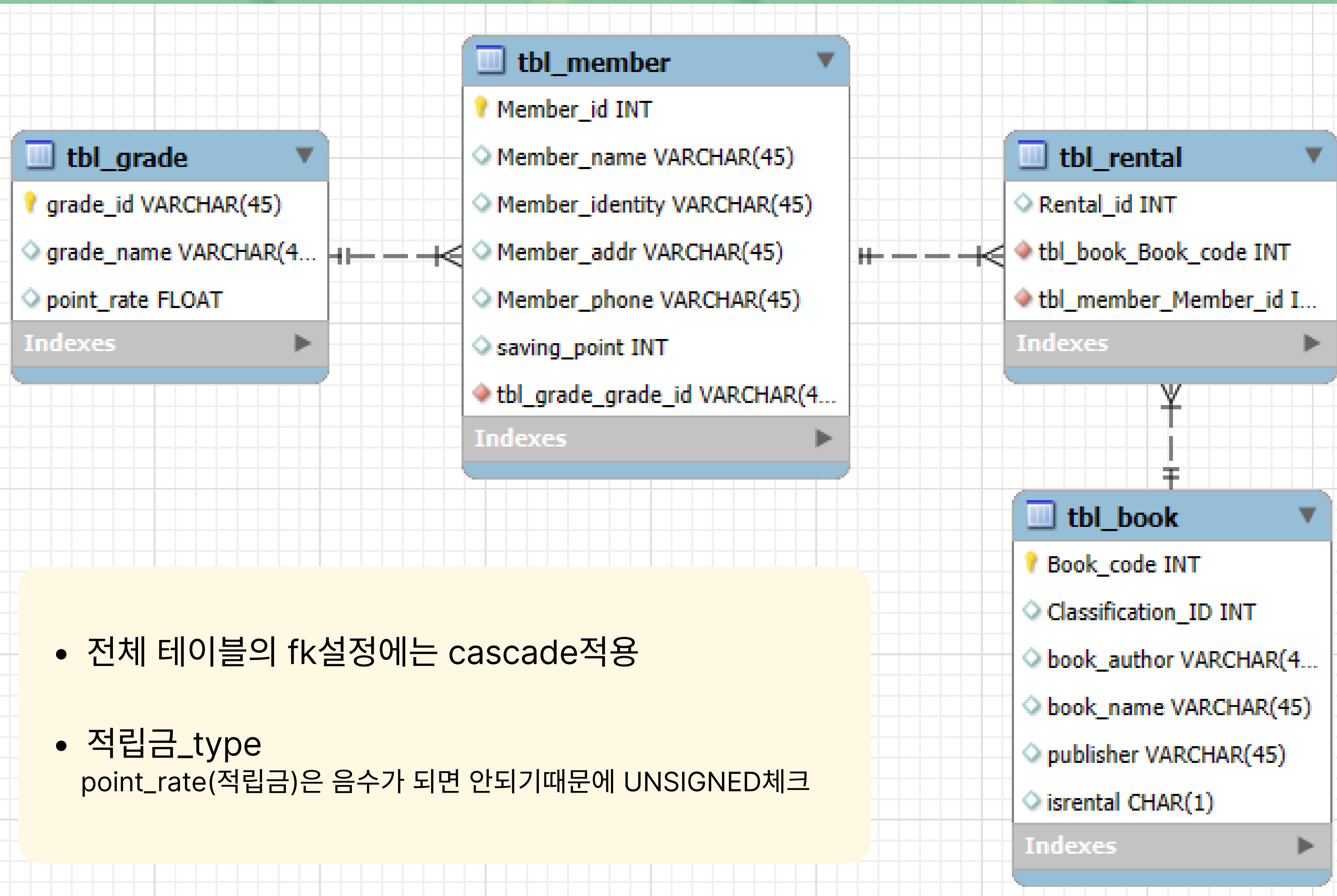
컬럼명	자료형	제약조건
grade_id	varchar(45)	Primary key
grade_name	varchar(45)	Not null
point_rate	float	Not null

tbl_member (추가/참조)

컬럼명	자료형	제약조건	
saving_point	Int	Not null	추가
grade_id	varchar(45)	foreign key	참조

tbl_grade

모델링



- 전체 테이블의 fk설정에는 cascade적용
- 적립금_type
point_rate(적립금)은 음수가 되면 안되기때문에 UNSIGNED체크



감사합니다!

To be continued

여기까지 하고 마치시겠습니까?