

## 과제 6: Delete & Update, 외래키 제약조건, 뷰 관련

1분반 32193430 이재원

이번 과제는 Insert, Delete & Update의 MySQL 확장된 기능들, 테이블 수정, 외래키 제약조건 적용 시 처리 방법, 그리고 뷰에 관한 내용이다. 과제 3의 solution에서 제공된 데이터베이스 생성 스크립트를 기반으로 다음 문제에 답하세요.

1. 다음의 요구사항대로 명령문을 작성하고 이를 실행 후 결과를 보이시오.

(a) Student의 gpa를 모두 0.0으로 수정하시오.

```
update Student
```

```
set gpa = 0;
```

	id	name	major	address	gpa	bdate
▶	930405	한나라	cs	서울 마포구 원효로	0	1974-06-12
	940123	강동희	ss	서울 중구 필동	0	1975-08-07
	950564	허영만	cs	서울 강동구 풍납동	0	1976-12-21
	960157	이동주	cs	서울 서초구 잠원동	0	1977-10-10
	970734	조용필	NULL	서울 영등포구 영등포동	0	1978-07-12
	980115	이미숙	ss	서울 서초구 반포동	0	NULL
	980397	조용기	NULL	서울 서대문구 홍은동	0	NULL
★	NULL	NULL	NULL	NULL	NULL	NULL

(b) 현재 course\_taken의 내용을 기반으로 각 학생의 gpa를 계산 후 수정하는 update문을 쓰시오.

```
create view avg_grade -- 각 학생의 gpa를 계산한 view 생성!
```

```
as (select sid, avg(grade) as aver from course_taken group by sid);
```

```
update student s join avg_grade a
```

```
on s.id = a.sid
```

```
set s.gpa = a.aver;
```

	id	name	major	address	gpa	bdate
▶	930405	한나라	cs	서울 마포구 원효로	3.28571	1974-06-12
	940123	강동희	ss	서울 중구 필동	3.5	1975-08-07
	950564	허영만	cs	서울 강동구 풍납동	1.8	1976-12-21
	960157	이동주	cs	서울 서초구 잠원동	1.5	1977-10-10
	970734	조용필	NULL	서울 영등포구 영등포동	2	1978-07-12
	980115	이미숙	ss	서울 서초구 반포동	3.6	NULL
	980397	조용기	NULL	서울 서대문구 홍은동	2	NULL
★	NULL	NULL	NULL	NULL	NULL	NULL

[illegible]

3. 과목번호는 종종 교육과정의 변화에 따라 수정되며, 그리고 외부 강사는 수급에 따라 수시로 변경된다. 이를 전제로 다음 문제에 답하시오.

(a) 과목번호의 수정 시 다른 테이블에 자동 반영되도록, 그리고 강사가 삭제되었을 때 과목 테이블에서 강사번호에 대한 외래키를 NULL로 설정될 수 있도록 외래키 제약조건들을 수정하시오.

```
alter table course drop
foreign key fk_Prerequisite_Course;
```

```
alter table course add -- course(id)의 update가 prerequisite에 반영되도록
CONSTRAINT fk_Prerequisite_Course
foreign key(prerequisite) references course(id)
on delete no action
on update cascade;
```

```
alter table course_taken drop
foreign key fk_CourseTaken_Course;
```

```
alter table course_taken add -- course(id)의 update가 course_taken에 반영되도록
CONSTRAINT fk_CourseTaken_Course foreign key(cid) references course(id)
on delete no action
on update cascade;
```

```
alter table course drop
foreign key fk_Course_Instructor;
```

```
alter table course add -- instructor pid가 삭제되면 course의 instructor는 null로 바뀐다.
Constraint fk_Course_Instructor
foreign key(instructor) references instructor(pid)
on delete set null;
```

(b) 위의 내용이 제대로 처리되는 경우의 예를 보이시오.

-- course의 id가 변경되는 경우

	id	name	instructor	prerequisite
▶	cs111	기초전산	cs13	NULL
	cs211	수치해석	cs12	cs111
	cs221	자료구조론	cs11	cs111
	cs222	시스템프로그래밍	cs10	cs111
	cs311	컴퓨터 구조론	cs14	cs111
	cs312	알고리즘	cs11	cs221
	cs321	프로그래밍언어론	cs13	cs221
	cs322	운영체제	cs15	cs222
	cs411	데이터베이스	cs13	cs321
course_taken 27		course 28	×	

update course

set id = 'cs123'

where name = '수치해석';

	id	name	instructor	prerequisite
	cs111	기초전산	cs13	NULL
▶	cs123	수치해석	cs12	cs111
	cs221	자료구조론	cs11	cs111
	cs222	시스템프로그래밍	cs10	cs111
	cs311	컴퓨터 구조론	cs14	cs111
	cs312	알고리즘	cs11	cs221
	cs321	프로그래밍언어론	cs13	cs221
	cs322	운영체제	cs15	cs222
	cs411	데이터베이스	cs13	cs321
course_taken 37		course 38	×	

	no	sid	cid	grade	year_taken
	1	930405	cs111	2	1993
▶	2	930405	cs123	4	1996
	3	930405	cs221	3	1996
	4	930405	cs222	3	1996
	5	930405	cs311	3	1997
	6	930405	cs321	4	1997
	7	930405	cs411	4	1998
	8	940123	ss111	2	1994
	9	940123	cs111	4	1997

course\_taken 37 × course 38

-- instructor의 pid가 삭제되는 경우

	pid	name	dept
▶	cs10	구자영	cs
	cs11	우진운	cs
	cs12	유해영	cs
	cs13	이석균	cs
	cs14	조경산	cs
	cs15	조성제	cs
	ss16	이강섭	ss
	ss17	황형태	ss
	ss18	이장택	ss

instructor 41 × course 42

delete from instructor where pid = 'cs10';

	id	name	instructor	prerequisite
	cs_changed	수치해석	cs12	cs111
	cs111	기초전산	cs13	NULL
	cs221	자료구조론	cs11	cs111
▶	cs222	시스템프로그래밍	NULL	cs111
	cs311	컴퓨터 구조론	cs14	cs111
	cs312	알고리즘	cs11	cs221
	cs321	프로그래밍언어론	cs13	cs221
	cs322	운영체제	cs15	cs222
	cs411	데이터베이스	cs13	cs321
instructor 44		course 45	X	

(c) 위의 내용이 제대로 처리되지 않는 경우도 존재한다. 어떤 경우인지 설명하시오.

	id	name	instructor	prerequisite
	cs_이재원	수치해석	cs12	cs111
	cs111	기초전산	cs13	NULL
▶	cs221	자료구조론	cs11	cs111
	cs222	시스템프로그래밍	NULL	cs111
	cs311	컴퓨터 구조론	cs14	cs111
	cs312	알고리즘	cs11	cs221
	cs321	프로그래밍언어론	cs13	cs221
	cs322	운영체제	cs15	cs222
	cs411	데이터베이스	cs13	cs321
course 83		X		

update course

set id = 'cs\_이재원'

where name = '자료구조론';

위와 같이 course table의 id를 update할 때, prerequisite에서 외래 키로서 참조하면 다음과 같은 구문과 함께 오류가 발생한다.

21:08:02 update course set id = 'cs\_이재원' where name = '자료구조론' Error Code: 1451.  
Cannot delete or update a parent row: a foreign key constraint fails (`학사db`.`course`,  
CONSTRAINT `fk\_Prerequisite\_Course` FOREIGN KEY (`prerequisite`) REFERENCES `course` (`id`) ON  
UPDATE CASCADE) 0.015 sec

4. 학생들의 성적 조회 용 뷰를 생성하고 이를 통해 자신의 수강 과목들의 성적을 조회하려고 한다. 뷰의 정의는 courseRecord(student\_id, student\_name, course\_name, course\_grade)으로 이루어진다.

(a) view를 생성하는 명령문을 보이시오.

drop view courseRecord;



```

create algorithm=merge view courseRecord(student_id, student_name, course_name, course_grade)
as
select s.id, s.name, c.name, ct.grade
from student s, course c, course_taken ct
where ct.sid = s.id and ct.cid = c.id
with check option;

```

	student_id	student_name	course_name	course_grade
▶	930405	한나라	기초전산	2
	930405	한나라	수치해석	4
	930405	한나라	자료구조론	3
	930405	한나라	시스템프로그래밍	3
	930405	한나라	컴퓨터 구조론	3
	930405	한나라	프로그래밍언어론	4
	930405	한나라	데이터베이스	4
	940123	강동희	전산통계	2
	940123	강동희	기초전산	4

courseRecord 56 ✕

(b) 이를 통해 특정 학생의 수강 과목들의 성적을 조회하는 예제와 실행 결과를 보이시오.

```
select * from courseRecord where student_id = '930405';
```

	student_id	student_name	course_name	course_grade
▶	930405	한나라	기초전산	2
	930405	한나라	수치해석	4
	930405	한나라	자료구조론	3
	930405	한나라	시스템프로그래밍	3
	930405	한나라	컴퓨터 구조론	3
	930405	한나라	프로그래밍언어론	4
	930405	한나라	데이터베이스	4

courseRecord 57 ✕

(c) courseRecord에 insert문이 실행가능한지 여부를 적고 불가능하면 이유를 쓰고 가능하면 실행 예제를 보이시오.

```
insert into courseRecord(student_id, student_name, course_name, course_grade) values
('32193430', '이재원', '데이터베이스', '4'); -- 불가능하다!
```

이유:

/\*

It's Impossible. Because for a multiple-table updatable view, **INSERT can work if it inserts into single table.**

\*/

(d) courseRecord에 delete문이 실행가능한지 여부를 적고 불가능하면 이유를 쓰고 가능하면 실행 예제를 보이시오.

```
delete from courseRecord
```

```
where student_name = '이재원'; -- 불가능하다!
```

/\*

**It's Impossible. Because Join views not allowed.**

\*/

(e) courseRecord의 각 속성에 대해 update문을 적용하려고 한다. 어떤 속성인 경우 가능한지를 쓰고 예를 보이고 불가능한 속성인 경우는 그 이유를 쓰시오.

/\*

각 속성이 updatable하면 update문을 적용할 수 있다.

속성이 다음의 요소를 포함하지 않으면 update문을 적용할 수 있다.

- 1) Aggregate functions such as min, max, avg and count
- 2) distinct
- 3) group by
- 4) having
- 5) union
- 6) left join or outer join
- 7) Subquery in the select or where clause.
- 8) non-updatable view in the From clause.
- 9) Reference only to literal values
- 10) Multiple references to any column of the base table.

\*/

-- 다음 구문으로 courserrecord view에 updatable한 속성이 포함되어 있다는 것을 알 수 있다.

```
select table_name, is_updatable
```

```
from information_schema.views
```

```
where table_schema = '학사DB';
```



	TABLE_NAME	IS_UPDATABLE
▶	avg_grade	NO
	rand_inst	YES
	courserecord	YES

```
update courseRecord
set student_name = '이재원'
where student_id = '930405';
```

	student_id	student_name	course_name	course_grade
▶	930405	이재원	기초전산	2
	930405	이재원	수치해석	4
	930405	이재원	자료구조론	3
	930405	이재원	시스템프로그래밍	3
	930405	이재원	컴퓨터 구조론	3
	930405	이재원	프로그래밍언어론	4
	930405	이재원	데이터베이스	4
	940123	강동희	전산통계	2
	940123	강동희	기초전산	4

courseRecord 63 ✕