

REPORT



과 목 명 : 데이터베이스기초

담당교수 : 이석균 교수님

소 속 : 소프트웨어학과

학 번 : 32151671

이 름 : 박민혁



단국대학교
Dankook University

이번 과제는 Insert, Delete & Update의 MySQL 확장된 기능들, 테이블 수정, 외래키 제약 조건 적용 시 처리 방법, 그리고 뷰에 관한 내용이다. 과제 2의 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.

```
# By. Dankook University Software 32151671 ParkMinHyeok

SET SQL_SAFE_UPDATES=0;

USE 학사db;

UPDATE student SET gpa = 0.0;

SELECT *FROM student;
```

	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 불납동	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.

```
# By. Dankook University Software 32151671 ParkMinHyeok

SET SQL_SAFE_UPDATES=0;

USE 학사db;

UPDATE student SET gpa=(
    SELECT AVG(grade) FROM course_taken WHERE id=sid GROUP BY sid
);

SELECT * FROM student;
```

	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

2. 학생 테이블에 Instructor에 대한 외래키인 advisor라는 속성을 추가하고자 한다.

(a) 위의 내용을 추가하는 명령문들을 쓰시오.

```
# By, Dankook University Software 32151671 ParkMinHyeok

ALTER TABLE student ADD
        COLUMN advisor CHAR(10) NULL;

ALTER TABLE Student ADD
        CONSTRAINT FK_Student_Instructor
        FOREIGN KEY (advisor) REFERENCES instructor (PID) ON DELETE SET NULL ;
```

(b) 같은 학과 교수로 advisor(지도교수)를 지정하는데 이는 임의 순서로 처리하고, 전공이 미정인 학생들의 지도교수는 null로 표현하시오. 이를 위한 명령문을 쓰고 처리 결과를 보이시오.

```
# By, Dankook University Software 32151671 ParkMinHyeok

SET SQL_SAFE_UPDATES=0;

USE 학사db;

UPDATE student SET advisor = NULL;

SELECT * FROM student;

UPDATE student t
SET     t.Advisor = (SELECT i.PID FROM instructor i WHERE t.major = i.dept
                    ORDER BY RAND() limit 1)
WHERE  t.Major IS NOT NULL;

SELECT * FROM student;
SELECT * FROM instructor;
```

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

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 강사 번호에 대한 외래키를 NULL로 설정될 수 있도록 외래키 제약조건들을 수정하시오.

```
# By. Dankook University Software 32151671 ParkMinHyeok
SET SQL_SAFE_UPDATES=0;
USE 학사db;

-- 2.(a) 과목의 경우

ALTER TABLE course_taken DROP
    FOREIGN KEY FK_CourseTaken_Course;
ALTER TABLE course_taken ADD
    CONSTRAINT FK_CourseTaken_Course
    FOREIGN KEY (CID) REFERENCES course (ID) on UPDATE CASCADE ;

ALTER TABLE Course DROP
    FOREIGN KEY Prerequisite_Course;
ALTER TABLE Course ADD
    CONSTRAINT Prerequisite_Course
    FOREIGN KEY (prerequisite) REFERENCES course(ID) on UPDATE CASCADE;

SELECT * FROM course;
```

```

-- 선수과목이 아닌 과목 명

SELECT C.NAME
FROM course C
WHERE NOT EXISTS (SELECT * FROM course c1 WHERE c1.prerequisite = C.id);
SELECT * FROM course C JOIN course_taken CT ON C.id = CT.cid
WHERE C.NAME = '회귀분석';    -- 선수과목이 없는 과목의 경우
UPDATE course
SET id = 'ss555'
WHERE NAME = '회귀분석';

SELECT * FROM course C JOIN Course_Taken CT ON C.id = CT.cid
WHERE C.NAME = '기초전산';    -- 선수과목이 있는 과목의 경우
SELECT *
FROM course C
WHERE EXISTS (SELECT * FROM course c1 WHERE c1.id = C.prerequisite AND c1.NAME = '기초전산');
UPDATE course
SET id = 'cs000'
WHERE NAME = '기초전산';

-- 2.(a) 강사의 경우

ALTER TABLE course DROP
    FOREIGN KEY FK_Course_Instructor;
ALTER TABLE course ADD
    CONSTRAINT FK_Course_Instructor
    FOREIGN KEY (Instructor) REFERENCES Instructor (PID) on DELETE SET NULL ;

SELECT * FROM course C JOIN Instructor I ON C.Instructor = I.pid WHERE I.NAME = '황형태';
DELETE FROM Instructor WHERE NAME = '황형태';
SELECT * FROM course C where C.NAME = '응용해석학';

```

(b) 위의 내용이 제대로 처리되는 경우의 예를 보이시오.

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

	id	name	instructor	prerequisite
	cs311	컴퓨터 구조론	cs14	cs111
	cs312	알고리즘	cs11	cs221
	cs321	프로그래밍언어론	cs13	cs221
	cs322	운영체제	cs15	cs222
	cs413	컴퓨터네트워크	cs14	cs311
	cs421	소프트웨어 공학	cs12	cs312
	cs522	데이터베이스	cs13	cs321
	ss111	전산통계	ss18	NULL
	ss311	응용해석학	ss17	ss111
	ss312	통계적 품질관리	ss16	ss111
	ss321	회귀분석	ss18	ss312
•	NULL	NULL	NULL	NULL

	id	name	instructor	prerequisite
	cs211	수치해석	NULL	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
	cs413	컴퓨터네트워크	cs14	cs311
	cs421	소프트웨어 공학	NULL	cs312
	cs522	데이터베이스	cs13	cs321
	ss111	전산통계	ss18	NULL

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

자신을 참조하는 외래키의 경우 업데이트나 삭제가 불가능하다.

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.

```
# By. Dankook University Software 32151671 ParkMinHyeok
USE 학사db;

CREATE view courseRecord(student_id, student_name, course_name, course_grade, instructor_name)
as
    SELECT s.id, s.name, c.name, ct.grade, i.name
    FROM student s JOIN course_taken ct ON s.id = ct.sid
        JOIN course c ON ct.cid = c.id
        JOIN instructor i ON c.instructor = i.pid ;
```

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

```
SELECT * FROM courseRecord WHERE student_name = '이미숙';
```

	학생ID	학생 이름	과목이름	성적
▶	980115	이미숙	기초전산	3
	980115	이미숙	자료구조론	3
	980115	이미숙	시스템프로그래밍	4
	980115	이미숙	컴퓨터 구조론	4
	980115	이미숙	전산통계	4

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

```
SELECT    table_name, is_updatable
FROM      information_schema.views
WHERE     table_schema = '학사DB';

-- *****한 테이블에 insert고 다른 값들이 디폴트나 널이 가능해야한다
INSERT INTO courseRecord(student_id, student_name) VALUES ('0003', 'test');
/* https://dev.mysql.com/doc/refman/5.7/en/view-updatability.html
   For a multiple-table updatable view, INSERT can work if it inserts into a single table. */

-- 나머지 값들은 디폴트나 널값으로 입력이 가능해서 insert 가능하다
INSERT INTO courseRecord(course_name) VALUES('test_course');
-- Error Code: 1423. Field of view '학사db.courserecord' underlying table doesn't have a
default value

-- 디폴트가 정해지지 않았는데 값을 안줘서 입력 불가능
/* With respect to insertability (being updatable with INSERT statements), an updatable view
is insertable
   if it also satisfies these additional requirements for the view columns:
   - There must be no duplicate view column names.
   - The view must contain all columns in the base table that do not have a default value.
   - The view columns must be simple column references. They must not be expressions,
such as these:
*/

INSERT INTO courseRecord(student_id, course_name, course_grade) VALUES('0004', 'test_course',
3);
-- Error Code: 1393. Can not modify more than one base table through a join view '학사
db.courserecord'
-- 세 테이블에 insert라 불가능
```


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

```
DELETE FROM courseRecord WHERE student_id = '0003';
-- Error Code: 1395, Can not delete from join view '학사db.courserecord'
/* DELETE: The table or tables to be deleted from in a DELETE statement must be merged
views.
Join views are not allowed (this differs from INSERT and UPDATE).
두개 이상 조인된 테이블은 삭제 불가능하다.*/
```

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

```
# By, Dankook University Software 32151671 ParkMinHyeok
USE 학사db;

SELECT * FROM courseRecord WHERE student_name = '이미숙';
UPDATE courseRecord
SET student_id = '000000'
WHERE student_name = '이미숙'; #외래키 제약조건 위반. 다른 테이블에서 참조하고 있기 때문
UPDATE courseRecord
SET student_name = 'TEST'
WHERE student_id = '980115';
SELECT * FROM courseRecord WHERE student_id = '980115';
UPDATE courseRecord
SET course_name = 'TEST'
WHERE instructor_name = '이장택';
```