About Referential Integrity Constraints

(프로젝트 1-2 수행할때부터 읽어보세요)

Referential Integrity Constraints는 foreign key를 정의함으로서 발생한다. 이는 참조하는 칼럼에 존재하는 값은 반드시 참조받는 테이블에 존재하는 값 중 하나여야 한다는 것을 의미한다. 예를 들어, 아래와 같이 정의된 두 개의 테이블이 있다고 가정하자.

```
Create table students
(
        id int,
        primary key (id)
)

create table enrolls
(
        id int,
        lecture_name char(50),
        foreign key(id) references students (id)
)
```

enrolls 테이블의 id 칼럼이 students 테이블의 id 칼럼을 참조하고 있다. 그렇다면, enrolls의 id 칼럼에 123이 있다면 이 값은 students의 id에도 있어야 한다는 것이다. 아래와 같은 상태는 제약조건이 지켜지고 있는 상태라고 할 수 있다.

students

id
1001
1002

enrolls

id	lecture_name
1001	Database
1001	Algorithm

만일 위 상황에서, enrolls에 (1003, Database)를 insert하려고 한다면 DBMS는 받아들이지 말아야 한다. 왜냐하면 1003이 students의 id 칼럼에 존재하지 않는 값이기 때문이다. 직관적으로 설명하면 enrolls는 학생과 수업간의 관계를 저장하는 테이블로, 존재하지 않는 학생에 대한 값을 가지면 안된다. 그렇기 때문에 students 테이블에 있는 값만을 가져야 하는 것이다. 이는 실제 응용

시스템에서 빈번히 발생하는 제약조건으로, 이를 제약조건을 표현하는 것이 foreign key인 것이다. Foreign key는 일반적으로 아래와 같은 특성을 갖는다. 본 프로젝트에서도 아래와 같은 조건을 만족하도록 구현해야 한다.

- 참조'받는' 칼럼은 반드시 해당 테이블의 primary key여야 한다. 예에서도 id 칼럼은 student 테이블의 primary key이다.
- Primary key와는 달리 foreign key는 null 값을 가질 수 있도록 정의될 수 있다. 이 경우 foreign key는 null 값, 혹은 참조하는 칼럼의 값을 가질 수 있다.
- 하나의 테이블은 복수의 foreign key를 가질 수 있다. 앞의 enrolls 테이블에 lecture_name을 추가로 foreign key로 지정할 수 있다.
- 다른 테이블에 의해 참조받는 칼럼을 가진 테이블은 drop 될 수 없다.

Cascade Deletion Requirements in Our Project

만일 앞의 예제에서 student 테이블의 id가 삭제되면 어떻게 처리해야 할까? 이는 당연한 문제가 아니며, 실제 DBMS에서는 이를 위해서 여러 가지 cascade update 옵션을 제공한다. 하지만 이번 프로젝트에서는 아래와 같은 두 가지 종류의 cascade update를 구현한다.

Case1; 만일 참조하는 foreign key들이 모두 nullable이라면? -> null 값으로 대체
즉, 앞의 예제에서 student 테이블의 (1001) 튜플을 지울 때 enrolls 을 어떻게 수정할까? Enrolls 테이블의 foreign key(즉 id 칼럼)는 nullable이므로, 1001 대신 null으로 채운다. 즉 아래와 같은 상황이 된다.

tudents

id
1002

enrolls

id	lecture_name
null	Database
null	Algorithm

Case2; 만일 참조하는 foreign key들 중 하나라도 not nullable이라면? -> 삭제 취소
같은 상황에서, enrolls table의 id 칼럼이 not null이라는 것만 다르다고 생각해보자. 이러한 경우에는 삭제 자체를 발생하지 못하도록 막는다. 즉 delete 질의를 하더라도 삭제가 발생하지 않게 된다.

위에서 보듯이, cascade update는 참조하는 측의 테이블의 정의에 따라 다르게 발생한다. 즉 하나

의 튜플이 삭제될 때에는 그 테이블을 참조하는 다른 여러 테이블들을 하나하나 살펴보며 업데이트 해주어야 한다. 만일 참조하는 참조 키 중 하나라도 not null인 것이 있다면 본래 참조받는 튜플의 삭제 오퍼레이션 자체가 취소되므로, nullable한 다른 참조 키가 있더라도 null값으로 대체해서는 안 된다는 점에 유의하자.

※위의 프로세스는 본 프로젝트를 위해 가정한 옵션으로, 실제로는 다양한 업데이트 옵션을 설정할 수 있다.

Additional Assumption for Our Project

- 어떤 테이블도 자기 자신을 참조할 수는 없다.
- 같은 foreign key가 여러 테이블의 칼럼들을 참조할 수 있다. 즉 아래와 같은 정의가 가능하다. 이는 양쪽 각각 모두에 대해서 referential integrity를 만족해야 함을 의미한다.

```
create table enrolls
(
        id int,
        lecture_name char(50),
        foreign key(id) references students(id)
        foreign key(id) references students_other(id)
)
```