

데이터베이스 기말과제 보고서

컴퓨터공학부 202015007 지유경

•기말 과제 제안서 제출된 내용과 대비 차이점, 발전성

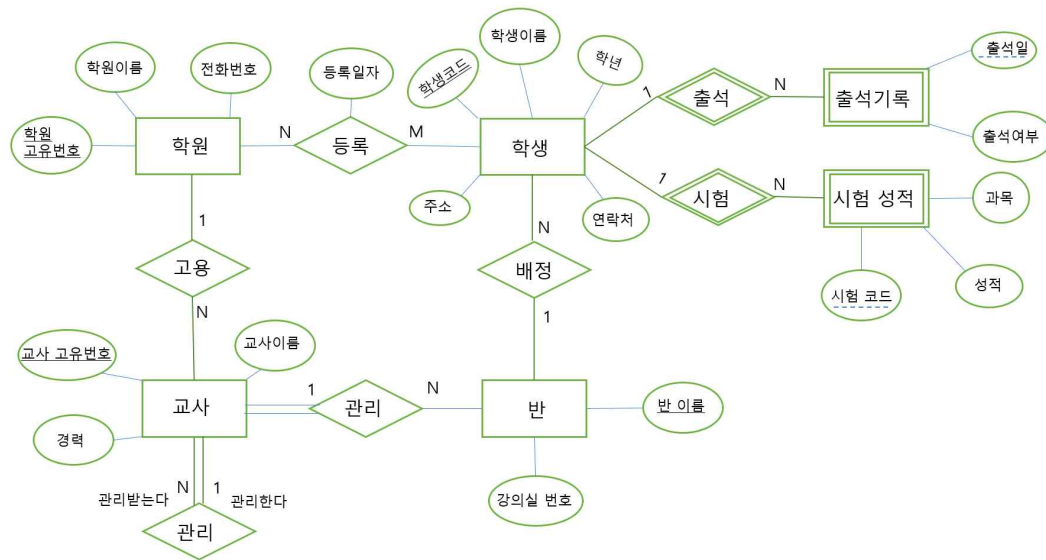
학생의 정보를 추가하여, 테이블을 5개에서 6개로 늘렸다. 학생 한명 당 여러 개의 시험성적을 가질 수 있도록 설정했다. 지난 기말 과제 제안서에서는 출석기록 하나당 여러 명의 학생의 기록을 가질 수 있도록 일대다 관계로 작성하였는데 이상하다는 생각이 들어서 학생 한명 당 여러 개의 출석기록을 가질 수 있도록 수정하였다. 출석기록에 학생코드와 이름, 출석일을 저장하도록 했었는데 한명 당 여러 개의 출석기록을 가질 수 있도록 수정하였기 때문에 학생의 출석여부는 날짜별로 하나씩 여러 개의 출석기록을 가질 수 있도록 설정했다. 따라서 출석기록의 기본 키는 출석일로 설정했다.

•요구사항 분석 및 개념 모델링

[요구사항 명세서]

1. 학원은 학원 고유번호(고유함), 학원 이름, 전화번호의 정보를 갖는다.
2. 학생은 학생코드(고유함), 학생 이름, 학년, 주소, 연락처의 정보를 가지며, 학생이 학원에 등록할 경우에 등록일자 정보가 저장된다.
3. 학원은 여러 명의 학생을 받을 수 있고, 학생도 여러 학원에 등록할 수 있다.
4. 학생이 출석할 경우 출석 기록이 생성되는데 한 학생당 여러 출석기록을 가질 수 있다. 출석기록은 날짜(고유함), 출석여부 정보를 가지며, 하루에 한번만 출석이 가능하기 때문에 날짜는 고유하다.
5. 학생이 학원에 다니지 않아 학생정보를 삭제하면, 이 출석 정보도 더 이상 남아있지 않게 된다.
6. 학생이 시험을 볼 경우, 시험성적이 생성되는데 한 학생당 여러 시험 성적 기록을 가질 수 있다. 시험 성적은 시험 코드(고유함)와 과목, 성적의 정보를 가진다.
7. 학생이 학원에 다니지 않아 학생 정보를 삭제하면, 이 시험 성적 기록도 더 이상 남아있지 않게 된다.
8. 학생은 시험을 통해 하나의 반에 배정이 되며, 반에는 여러 명의 학생이 배정될 수 있다.
9. 학생이 반에 배정될 경우, 학생의 시험 성적이 저장된다.
10. 반은 반 이름(고유함), 강의실 번호의 정보를 갖는다.
11. 학원은 여러 명의 교사를 고용할 수 있으며, 교사는 하나의 학원에서만 근무할 수 있다.
12. 교사는 교사 고유번호(고유함), 교사 이름, 경력의 정보를 가지고 있으며, 가장 직급이 높은 교사가 나머지 교사들을 관리한다.
13. 한 명의 교사는 여러 반을 관리할 수 있으며, 한 반에는 한 명의 교사가 맡는다.
14. 교사는 필수적으로 반을 관리해야 한다.

[ER 다이어그램]



•논리 모델링(관계형 모델)

-관계형모델 변환

학원(학원고유번호, 학원이름, 전화번호)

교사(학원고유번호(FK), 교사고유번호, 교사이름, 경력)

반(교사고유번호(FK), 반이름, 강의실번호)

학생(반이름(FK), 학생코드, 학생이름, 학년, 주소, 연락처)

등록(학원고유번호(FK), 학생코드(KF), 등록일자)

출석기록(학생코드(FK), 출석일, 출석여부)

시험성적(학생코드(FK), 시험코드, 과목, 성적)

[테이블 정의서]

```

CREATE TABLE 학원(
  학원고유번호 NUMBER(6) NOT NULL,
  학원이름 VARCHAR(20) NOT NULL,
  학원연락처 VARCHAR(20) NOT NULL,
  CONSTRAINT PK_학원 PRIMARY KEY(학원고유번호)
);
    
```

//학원 TABLE에는 학원고유번호, 학원이름, 학원연락처라는 속성들이 있고, 학원고유번호는 기본키로 설정되었다. 이때, 제약조건의 이름을 CONSTRAINT를 이용하여 PK_학원으로 지정했다. 제약조건의 이름을 설정하지 않아도 자동으로 생성되지만, 나중에 제약조건을 비활성화하거나 삭제하는 등의 관리를 위해서 이름을 지정해주었다.

```

CREATE TABLE 교사(
  교사고유번호 NUMBER(6) NOT NULL,
  학원_고유번호1 NUMBER(6),
  CONSTRAINT fk_교사_학원고유번호
  FOREIGN KEY(학원_고유번호1) REFERENCES 학원(학원고유번호) ON DELETE CASCADE,
  교사이름 VARCHAR(10) NOT NULL,
  경력 VARCHAR(50) NOT NULL,
  CONSTRAINT PK_교사 PRIMARY KEY(교사고유번호)
);
    
```

//교사 TABLE에는 교사고유번호, 학원_고유번호, 교사이름, 경력이라는 속성들이 있으며 교사 고유번호는 기본키, 학원 TABLE의 학원고유번호를 참조한 학원_고유번호는 외래키이다. 위와 마찬가지로 각각의 제약에 이름을 지정해주었

다. CASCADE는 외래키의 옵션으로 부모 테이블의 데이터가 삭제될 시에 자식 테이블의 데이터도 동시 삭제되도록 함.

```
CREATE TABLE 반(  
반이름 VARCHAR(10) NOT NULL,  
교사_고유번호1 NUMBER(6),  
CONSTRAINT fk_반_교사고유번호  
FOREIGN KEY(교사_고유번호1) REFERENCES 교사(교사고유번호) ON DELETE CASCADE,  
강의실번호 NUMBER(6) NOT NULL,  
CONSTRAINT PK_반 PRIMARY KEY(반이름)  
);
```

//반 TABLE에는 반이름, 교사_고유번호1, 강의실번호의 속성들이 있으며 반이름은 기본키, 교사_고유번호는 교사 TABLE의 교사고유번호를 참조한 외래키이다.

```
CREATE TABLE 학생(  
반_이름 VARCHAR(10),  
학생코드 VARCHAR(10) NOT NULL,  
학생이름 VARCHAR(10) NOT NULL,  
학년 VARCHAR(10) NOT NULL,  
주소 VARCHAR(30) NOT NULL,  
학생연락처 VARCHAR(20) NOT NULL,  
CONSTRAINT fk_학생_반이름  
FOREIGN KEY(반_이름) REFERENCES 반(반이름) ON DELETE CASCADE,  
CONSTRAINT PK_학생 PRIMARY KEY(학생코드)  
);
```

//학생 TABLE에는 반_이름, 학생코드, 학생이름, 학년, 주소, 학생연락처의 속성들이 있으며 학생코드는 기본키, 반_이름은 반 TABLE의 반이름을 참조한 외래키이다.

```
CREATE TABLE 등록(  
학원_고유번호2 NUMBER(6),  
학생_코드1 VARCHAR(10) NOT NULL,  
등록일자 DATE NOT NULL,  
PRIMARY KEY(학원_고유번호2, 학생_코드1),  
CONSTRAINT fk_등록_학원고유번호  
FOREIGN KEY(학원_고유번호2) REFERENCES 학원(학원고유번호) ON DELETE CASCADE,  
CONSTRAINT fk_등록_학생코드  
FOREIGN KEY(학생_코드1) REFERENCES 학생(학생코드) ON DELETE CASCADE  
);
```

//등록 TABLE은 학원_고유번호2, 학생_코드1, 등록일자 속성을 가지며, 기본키는 두 개로 학원_고유번호2와 학생_코드1이다. 이들은 각각 학원 TABLE의 학원고유번호, 학생 TABLE의 학생코드를 참조한 외래키이다.

```
CREATE TABLE 출석기록(  
학생_코드2 VARCHAR(10),  
출석일 DATE NOT NULL,  
출석여부 VARCHAR(10) NOT NULL,  
PRIMARY KEY(학생_코드2, 출석일),  
CONSTRAINT fk_출석기록_학생코드  
FOREIGN KEY(학생_코드2) REFERENCES 학생(학생코드) ON DELETE CASCADE  
);
```

//출석기록 TABLE은 학생_코드2, 출석일, 출석여부의 속성을 가지며, 기본키는 학생_코드2와 출석일이다. 학생_코드2는 학생 TABLE의 학생코드를 참조한 외래키이기도 하다.

```
CREATE TABLE 시험(  

```

```

학생_코드3 VARCHAR(10),
시험코드 VARCHAR(10),
과목 VARCHAR(10) NOT NULL,
성적 NUMBER(6) NOT NULL,
PRIMARY KEY(학생_코드3,시험코드),
FOREIGN KEY(학생_코드3) REFERENCES 학생(학생코드) ON DELETE CASCADE
);

```

//시험 TABLE은 학생_코드3, 시험코드, 과목, 성적의 속성을 가지며, 기본키는 학생_코드3, 시험코드이다. 학생_코드3은 학생 TABLE의 학생코드를 참조한 외래키이기도 하다.

사용된 제약조건으로는 NULL 값을 허용하지 않는 NOT NULL, NOT NULL과 UNIQUE의 의미를 합친 PRIMARY KEY, 다른 테이블의 컬럼을 조회해서 무결성 검사를 하는 FOREIGN KEY가 있다.

•구현

테이블 정의

SQL Worksheet

```

1 CREATE TABLE 학원(
2   학원고유번호 NUMBER(6) NOT NULL,
3   학원이름 VARCHAR(20) NOT NULL,
4   학원연락처 VARCHAR(20) NOT NULL,
5   CONSTRAINT PK_학원 PRIMARY KEY(학원고유번호)
6 );
7 CREATE TABLE 교사(
8   교사고유번호 NUMBER(6) NOT NULL,
9   학원_고유번호1 NUMBER(6),
10  CONSTRAINT fk_교사_학원고유번호
11  FOREIGN KEY(학원_고유번호1) REFERENCES 학원(학원고유번호) ON DELETE CASCADE,
12  교사이름 VARCHAR(10) NOT NULL,
13  경력 VARCHAR(50) NOT NULL,
14  CONSTRAINT PK_교사 PRIMARY KEY(교사고유번호)
15 );
16 CREATE TABLE 반(
17   반이름 VARCHAR(10) NOT NULL,
18   교사_고유번호1 NUMBER(6),
19   CONSTRAINT fk_반_교사고유번호
20   FOREIGN KEY(교사_고유번호1) REFERENCES 교사(교사고유번호) ON DELETE CASCADE,
21   강의실번호 NUMBER(6) NOT NULL

```

Table created.

Table created.

Table created.

Table created.

Table created.

학원 TABLE

Sort most recent last View Statements and Results

Statement 1

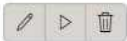


```
CREATE TABLE 학원(  
  학원고유번호 NUMBER(6) NOT NULL,  
  학원이름 VARCHAR(20) NOT NULL,  
  학원연락처 VARCHAR(20) NOT NULL ,  
  CONSTRAINT PK_학원 PRIMARY KEY(학원고유번호)  
)
```

Table created.

교사 TABLE

Statement 2



```
CREATE TABLE 교사(  
  교사고유번호 NUMBER(6) NOT NULL,  
  학원_고유번호1 NUMBER(6),  
  CONSTRAINT fk_교사_학원고유번호  
  FOREIGN KEY(학원_고유번호1) REFERENCES 학원(학원고유번호) ON DELETE CASCADE,  
  교사이름 VARCHAR(10) NOT NULL,  
  경력 VARCHAR(50) NOT NULL,  
  CONSTRAINT PK_교사 PRIMARY KEY(교사고유번호)  
)
```

Table created.

반 TABLE

Statement 3



```
CREATE TABLE 반(  
  반이름 VARCHAR(10) NOT NULL,  
  교사_고유번호1 NUMBER(6),  
  CONSTRAINT fk_반_교사고유번호  
  FOREIGN KEY(교사_고유번호1) REFERENCES 교사(교사고유번호) ON DELETE CASCADE,  
  강의실번호 NUMBER(6) NOT NULL,  
  CONSTRAINT PK_반 PRIMARY KEY(반이름)  
)
```

Table created.

학생 TABLE

Statement 4






```
CREATE TABLE 학생(  
  반_이름 VARCHAR(10),  
  학생코드 VARCHAR(10) NOT NULL,  
  학생이름 VARCHAR(10) NOT NULL,  
  학년 VARCHAR(10) NOT NULL,  
  주소 VARCHAR(30) NOT NULL,  
  학생연락처 VARCHAR(20) NOT NULL,  
  CONSTRAINT fk_학생_반이름  
  FOREIGN KEY(반_이름) REFERENCES 반(반이름) ON DELETE CASCADE,  
  CONSTRAINT PK_학생 PRIMARY KEY(학생코드)  
)
```

Table created.

등록 TABLE

Statement 5



```
CREATE TABLE 등록(  
  학원_고유번호2 NUMBER(6),  
  학생_코드1 VARCHAR(10) NOT NULL,  
  등록일자 DATE NOT NULL,  
  PRIMARY KEY(학원_고유번호2,학생_코드1),  
  CONSTRAINT fk_등록_학원고유번호  
  FOREIGN KEY(학원_고유번호2) REFERENCES 학원(학원고유번호) ON DELETE CASCADE,  
  CONSTRAINT fk_등록_학생코드  
  FOREIGN KEY(학생_코드1) REFERENCES 학생(학생코드) ON DELETE CASCADE  
)
```

Table created.

출석기록 TABLE

Statement 6




```
CREATE TABLE 출석기록(  
  학생_코드2 VARCHAR(10),  
  출석일 DATE NOT NULL,  
  출석여부 VARCHAR(10) NOT NULL,  
  PRIMARY KEY(학생_코드2,출석일),  
  CONSTRAINT fk_출석기록_학생코드  
  FOREIGN KEY(학생_코드2) REFERENCES 학생(학생코드) ON DELETE CASCADE  
)
```

Table created.

시험 TABLE

Statement 7



```
CREATE TABLE 시험(  
  학생_코드3 VARCHAR(10),  
  시험코드 VARCHAR(10),  
  과목 VARCHAR(10) NOT NULL,  
  성적 NUMBER(6) NOT NULL,  
  PRIMARY KEY(학생_코드3,시험코드),  
  FOREIGN KEY(학생_코드3) REFERENCES 학생(학생코드) ON DELETE CASCADE  
)
```

Table created.

데이터 삽입

SQL Worksheet

```
1 INSERT INTO 학원 VALUES('1001','엠코드수학','206-1111');
2
3 INSERT INTO 학원 VALUES('1002','최선어학원','206-1112');
4
5 INSERT INTO 학원 VALUES('1003','수학여행','206-1113');
6
7 INSERT INTO 학원 VALUES('1004','대찬학원','206-1114');
8
9 INSERT INTO 학원 VALUES('1005','공감','206-1115');
10
11 INSERT INTO 학원 VALUES('1006','대성학원','206-1116');
12
13 INSERT INTO 학원 VALUES('1007','러셀학원','206-1117');
14
15 INSERT INTO 학원 VALUES('1008','정철학원','206-1118');
16
17 INSERT INTO 교사
18 VALUES('01','1001','지유경','엠코드수학학원 18개월 근무');
19
20 INSERT INTO 교사
21 VALUES('02','1001','황수현','정철학원 3개월 근무');
22
23 INSERT INTO 교사
24 VALUES('03','1003','이승민','수학여행');
25
26 INSERT INTO 교사
27 VALUES('04','1005','한지유','공감 10개월 근무');
28
29 INSERT INTO 교사
```

1 row(s) inserted.

1 row(s) inserted.

SQL 질의어

1. 학원의 이름과 각 학원에 근무하는 교사 이름을 검색하라.

```
1 SELECT 학원이름,교사이름 FROM 학원,교사
2 WHERE 학원.학원고유번호=교사.학원_고유번호1;
```

학원이름	교사이름
엠코드수학	지유경
엠코드수학	황수현
최선어학원	전수원
수학여행	이승민
공감	한지유
공감	김예하
러셀학원	채민경

Download CSV

7 rows selected.

2. 학원의 이름과 그 학원에 다니는 학생이름, 등록일자를 검색하라.

SQL Worksheet

```
1 SELECT 학원이름,학생이름,등록일자 FROM 학원,학생,등록
2 WHERE 학생.학생코드=등록.학생_코드 1 AND 학원.학원고유번호=등록.학원_고유번호 2;
```

학원이름	학생이름	등록일자
엠코드수학	정채현	01-DEC-21
엠코드수학	조은재	24-FEB-21
엠코드수학	이윤지	11-APR-21
엠코드수학	이은빈	29-SEP-21
엠코드수학	이수미	01-SEP-19
엠코드수학	서동주	18-NOV-20
수학여행	이가인	13-OCT-18
공감	천민채	05-FEB-19
공감	김민종	15-MAY-21
최선어학원	김가영	11-AUG-20
러셀학원	박소현	25-DEC-19

[Download CSV](#)

11 rows selected.

3. 학원의 이름과 그 학원의 교사이름과 맡는 반, 그 반의 학생 이름을 검색하라.

SQL Worksheet

```
1 SELECT 학원이름,교사이름,반이름,학생이름 FROM 학원,교사,반,학생
2 WHERE 학원.학원고유번호=교사.학원_고유번호 1
3 AND 교사.교사고유번호=반.교사_고유번호 1 AND 반.반이름=학생.반_이름;
```

학원이름	교사이름	반이름	학생이름
엠코드수학	지유경	a1	정채현
엠코드수학	지유경	a1	조은재
엠코드수학	지유경	a1	이윤지
엠코드수학	지유경	a1	이은빈
엠코드수학	황수현	a2	이수미
엠코드수학	황수현	a2	서동주
수학여행	이승민	b1	이가인
공감	한지유	c1	천민채
공감	김예하	c2	김민종
최선어학원	전수원	d1	김가영
러셀학원	채민경	e1	박소현

[Download CSV](#)

11 rows selected.

4. 시험코드가 '0001'인 시험을 치른 학생 중에서 성적이 50점 이상인 학생의 이름과 성적을 검색하라.

SQL Worksheet

```
1 SELECT 학생이름, 성적 FROM 학생, 시험
2 WHERE 학생.학생코드=시험.학생_코드3 AND 시험코드='0001' AND 성적 >= 50;
```

학생이름	성적
정채현	70
조은재	80
이수미	99

[Download CSV](#)

3 rows selected.

5. 시험코드가 '0001'인 시험을 치른 학생의 성적중에서 가장 높은 점수를 검색하라.

SQL Worksheet

```
1 SELECT MAX(성적) FROM 시험 WHERE 시험.시험코드='0001';
```

MAX(성적)

99

[Download CSV](#)

6. 21년 12월 1일 출석한 학생의 이름과 출석여부를 검색하라.

SQL Worksheet

```
1 SELECT 출석일, 학생이름, 출석여부 FROM 학생, 출석기록
2 WHERE 학생.학생코드=출석기록.학생_코드2
3 AND 출석일=DATE '21-12-01' AND 출석여부='출석';
```

출석일	학생이름	출석여부
01-DEC-21	정채현	출석
01-DEC-21	조은재	출석
01-DEC-21	이윤지	출석
01-DEC-21	이수미	출석
01-DEC-21	서동주	출석
01-DEC-21	이가인	출석
01-DEC-21	천민채	출석
01-DEC-21	김민종	출석
01-DEC-21	박소현	출석

[Download CSV](#)

9 rows selected.

7. 21년 12월 1일에 출석한 학생의 수를 세어 "21/12/01 출석학생수"라는 이름으로 바꿔서 검색하라.

SQL Worksheet

```
1 SELECT COUNT(출석여부) AS "21/12/01 출석학생수" FROM 출석기록
2 WHERE 출석일=DATE '21-12-01' AND 출석여부='출석';
```

21/12/01 출석학생수

9

[Download CSV](#)

8. 학생연락처가 '010-6661-4442'인 학생의 이름을 '한혜정'으로 변경하고 그 학생의 정보를 검색하라.

SQL Worksheet

```
1 SELECT 반_이름,학생코드,학생이름,학년,주소,학생연락처 FROM 학생 WHERE 학생연락처='010-6661-4442';
2 UPDATE 학생 SET 학생이름 ='한혜정' WHERE 학생연락처='010-6661-4442';
3 SELECT 반_이름,학생코드,학생이름,학년,주소,학생연락처 FROM 학생 WHERE 학생연락처='010-6661-4442';
```

반_이름	학생코드	학생이름	학년	주소	학생연락처
a1	0a15	이은빈	중3	경기도 수원시 영통구	010-6661-4442

[Download CSV](#)

1 row(s) updated.

반_이름	학생코드	학생이름	학년	주소	학생연락처
a1	0a15	한혜정	중3	경기도 수원시 영통구	010-6661-4442

[Download CSV](#)

9. 이름이 '서동주'인 학생의 정보를 삭제하고 출석기록에서 학생의 출석 정보가 사라짐을 확인하라.

SQL Worksheet

```
1 SELECT *FROM 출석기록;
2 DELETE FROM 학생 WHERE 학생이름='서동주';
3 SELECT *FROM 출석기록;
```

학생_코드2	출석일	출석여부	학생_코드2	출석일	출석여부
0a12	01-DEC-21	출석	0a12	01-DEC-21	출석
0a12	03-DEC-21	출석	0a12	03-DEC-21	출석
0a12	04-DEC-21	지각	0a12	04-DEC-21	지각
0a12	06-DEC-21	결석	0a12	06-DEC-21	결석
0a12	08-DEC-21	출석	0a12	08-DEC-21	출석
0a13	01-DEC-21	출석	0a13	01-DEC-21	출석
0a14	01-DEC-21	출석	0a14	01-DEC-21	출석
0a15	01-DEC-21	결석	0a15	01-DEC-21	결석
0a16	01-DEC-21	출석	0a16	01-DEC-21	출석
0a17	01-DEC-21	출석	0a18	01-DEC-21	출석
0a18	01-DEC-21	출석	0a19	01-DEC-21	출석
0a19	01-DEC-21	출석	0a20	01-DEC-21	출석
0a20	01-DEC-21	출석	0a21	01-DEC-21	지각
0a21	01-DEC-21	지각	0a22	01-DEC-21	출석
0a22	01-DEC-21	출석			

Download CSV 14 rows selected.

->'서동주'학생의 학생코드인 '0a17'의 출석정보가 삭제됨을 확인할 수 있다.

10. 출석기록 테이블 삭제

```
1 DROP TABLE 출석기록;
```

Table dropped.

-개체 무결성 제약조건 위배 사례

```
1 INSERT INTO 학생
2 VALUES('a4','김철아','고3','경기도 수원시 장안구','010-2229-9992');
```

ORA-01400: cannot insert NULL into ("SQL_OTMFGVKT SXANUQRPQBRAJBVKU"."학생"."학생코드") ORA-06512: at "SYS.DBMS_SQL", line 1721

:학생 TABLE의 학생코드는 NOT NULL로 설정했기 때문에 NULL값을 넣을 수 없다. 이는 개체 무결성 제약조건을 위배되므로 실행될 수 없다.

-참조 무결성 제약조건 위배 사례

```
1 DROP TABLE 학원;
```

```
ORA-02449: unique/primary keys in table referenced by foreign keys
```

:학원 TABLE은 교사 TABLE과 등록 TABLE의 부모 TABLE로 자식 TABLE에 데이터가 남아있는 경우 부모 TABLE의 데이터를 삭제할 수 없다. 따라서 이는 참조 무결성 제약조건에 위배되므로 실행될 수 없다.