

실습과제 지문

간단한 학사관리 시스템을 구축하려 합니다.

고객과의 상담을 통해서 관리할 필요가 있는 다음 정보들을 파악했습니다.

1. 관계

- 학과와 학생은 1 : 다 관계
학과는 많은 학생을 가질 수 있다.
학생은 한 학과의 소속된다.
- 학과와 교수는 1 : 다 관계
학과는 많은 교수를 가질 수 있다.
교수는 한 학과의 소속된다.
- 교수와 개설과목은 1 : 다 관계
교수는 많은 과목을 가르칠 수 있다.
과목은 강의하는 교수 한 명이 지정된다.
- 과목과 학생은 다 : 다 관계
과목은 수강하는 많은 학생을 가진다.
학생은 많은 과목을 수강할 수 있다.

관계형 데이터베이스는 다:다 관계를 허용하지 않는다. 과목과 학생의 다:다 관계를 해소하기 위해서 조인테이블 역할을 수행할 수강 엔티티를 추가한다.

- 과목과 수강은 1 : 다 관계
- 학생과 수강은 1 : 다 관계

2. 데이터

학생(Student)

학번(student_id), 성명(student_name), 키(height), 학과코드(department_id)

학과(Department)

학과코드(department_id), 학과명(department_name)

교수(Professor)

교수코드(professor_id), 교수명(professor_name), 학과코드(department_id)

개설과목(Course)

과목코드(course_id), 과목명(course_name), 교수코드(professor_id),
시작일(start_date), 종료일(end_date)

수강(Student_Course)

학번(student_id), 과목코드(course_id)

외부참조 키(FK) : 해당 엔티티와 관련된 데이터를 갖고 있는 프로퍼티가 아닌 다른 엔티티와의 관계를 설정하기 위한 키

3. 테스트 데이터 생성

다음 작업은 테스트를 위한 가상데이터 등록입니다. 다음 쿼리를 실행하여 데이터를 입력합니다.

학과

```
insert into department values(1, '수학');  
insert into department values(2, '국문학');  
insert into department values(3, '정보통신공학');  
insert into department values(4, '모바일공학');
```

학생

```
insert into student values(1, '가길동', 177, 1);  
insert into student values(2, '나길동', 178, 1);  
insert into student values(3, '다길동', 179, 1);  
insert into student values(4, '라길동', 180, 2);  
insert into student values(5, '마길동', 170, 2);  
insert into student values(6, '바길동', 172, 3);  
insert into student values(7, '사길동', 166, 4);  
insert into student values(8, '아길동', 192, 4);
```

교수

```
insert into professor values(1, '가교수' ,1);  
insert into professor values(2, '나교수' ,2);  
insert into professor values(3, '다교수' ,3);  
insert into professor values(4, '빌게이츠' ,4);  
insert into professor values(5, '스티브잡스' ,3);
```

개설과목

```
insert into course values(1, '교양영어', 1, '2016/9/2', '2016/11/30');  
insert into course values(2, '데이터베이스 입문', 3, '2016/8/20','2016/10/30');  
insert into course values(3, '회로이론', 2, '2016/10/20', '2016/12/30');
```

```
insert into course values(4, '공업수학', 4, '2016/11/2', '2017/1/28');
insert into course values(5, '객체지향프로그래밍', 3, '2016/11/1', '2017/1/30');
```

수강

```
insert into student_course values(1, 1);
insert into student_course values(2, 1);
insert into student_course values(3, 2);
insert into student_course values(4, 3);
insert into student_course values(5, 4);
insert into student_course values(6, 5);
insert into student_course values(7, 5);
```

문제 1.

학생번호, 학생명, 키높이, 학과번호, 학과명 정보를 출력합니다.

STUDENT_ID	STUDENT_NAME	HEIGHT	DEPARTMENT_ID	DEPARTMENT_NAME
1	1 가길동	177	1	수학
2	2 나길동	178	1	수학
3	3 다길동	179	1	수학
4	4 라길동	180	2	국문학
5	5 마길동	170	2	국문학
6	6 바길동	172	3	정보통신공학
7	7 사길동	166	4	모바일공학
8	8 아길동	192	4	모바일공학

문제 2.

'가교수' 교수의 교수아이디를 출력하세요.

PROFESSOR_ID
1

문제 3.

학과이름별 교수의 수를 출력하세요.

DEPARTMENT_NAME	COUNT(P.PROFESSOR_ID)
1 국문학	1
2 정보통신공학	2
3 수학	1
4 모바일공학	1

문제 4.

'정보통신공학'과의 학생정보를 출력하세요.

STUDENT_ID	STUDENT_NAME	HEIGHT	DEPARTMENT_ID	DEPARTMENT_NAME
1	6 바길동	172	3	정보통신공학

문제 5.

'정보통신공학'과의 교수명을 출력하세요.

PROFESSOR_ID	PROFESSOR_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	3 다교수	3	정보통신공학
2	5 스티브잡스	3	정보통신공학

과제 6.

학생 중 성이 '아'인 학생이 속한 학과명과 학생명을 출력하세요.

STUDENT_NAME	DEPARTMENT_NAME
1 아길동	모바일공학

문제 7.

키가 180~190 사이에 속하는 학생 수를 출력하세요.

COUNT(STUDENT_ID)
1 1

문제 8.

학과이름별 키의 최고값, 평균값을 출력하세요.

DEPARTMENT_NAME	MAX(HEIGHT)	AVG(HEIGHT)
1 국문학	180	175
2 정보통신공학	172	172
3 수학	179	178
4 모바일공학	192	179

문제 9.

'다길동' 학생과 같은 학과에 속한 학생의 이름을 출력하세요.

STUDENT_NAME
1 가길동
2 나길동
3 다길동

문제 10.

2016 년 11 월 시작하는 과목을 수강하는 학생의 이름과 수강과목을 출력하세요.

STUDENT_NAME	COURSE_NAME
1 마길동	공업수학
2 바길동	객체지향프로그래밍
3 사길동	객체지향프로그래밍

문제 11.

'데이터베이스 입문' 과목을 수강신청한 학생의 이름은?

STUDENT_NAME
1 다길동

문제 12.

'빌게이츠' 교수의 과목을 수강신청한 학생수는?

COUNT(STUDENT_ID)
1 1