

ICE4016 데이터베이스설계

과제2

보고서 작성 서약서

1. 나는 타학생의 보고서를 베끼거나 여러 보고서의 내용을 짜집기하지 않겠습니다.

2. 나는 보고서의 주요 내용을 인터넷사이트 등을 통해 얻지 않겠습니다.

3. 나는 보고서의 내용을 조작하지 않겠습니다.

4. 나는 보고서 작성에 참고한 문헌의 출처를 밝히겠습니다.

5. 나는 나의 보고서를 제출 전에 타학생에게 보여주지 않겠습니다.

나는 보고서 작성시 윤리에 어긋난 행동을 하지 않고 정보통신공학인으로서 나의 명예를 지킬 것을 맹세합니다.

2021년 10월 24일

학부 정보통신공학과

학년 3

성명 김현우

학번 12171774



Student Database를 다음과 같이 script code를 완성 하였습니다.  
전체 코드는 아래에 있습니다.  
텍스트이(가) 표시된 사진

자동 생성된 설명  
다음과 같이 assignment2 db를 만든 뒤에 use로 change 하였습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명  
student table의 PK의 빈칸은 Student\_Number로 설정 하였습니다. (이름, Class, Major는 Unique X)  
텍스트이(가) 표시된 사진

자동 생성된 설명  
course table의 pk는 course\_name이 unique이므로 course\_number로 하였습니다.  
텍스트이(가) 표시된 사진

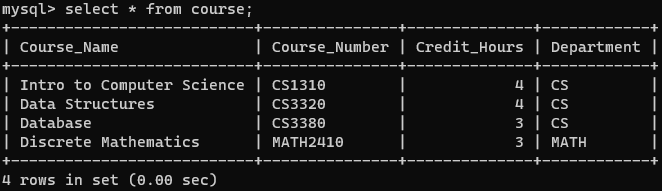
자동 생성된 설명  
section table의 pk는 section\_identifier을 pk로, coure\_number는 COURSE TABLE을 참조 하도록 하였습니다.  
텍스트이(가) 표시된 사진

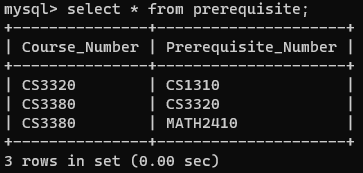
자동 생성된 설명  
grade\_report table의 pk는 pair로 student\_number, section\_identifier 로 하였으며, fk는 각각 STUDENT,  
SECTION에서 참조 하도록 하였습니다.  
텍스트이(가) 표시된 사진

자동 생성된 설명  
prerequisite에서 pk는 pair로 course\_number, prerequisite\_number 로 하였으며, fk는 둘 다 course  
에서 참조 하도록 하였습니다.

CREATE SCHEMA IF NOT EXISTS `assignment2` DEFAULT CHARACTER SET utf8mb4;  
use assignment2;  
CREATE TABLE STUDENT ( Name VARCHAR(30) NOT NULL,   
Student\_Number INTEGER NOT NULL,   
Class CHAR NOT NULL,   
Major CHAR(4),   
PRIMARY KEY (Student\_Number) );  
CREATE TABLE COURSE ( Course\_Name VARCHAR(30) NOT NULL,   
Course\_Number CHAR(8) NOT NULL,   
Credit\_Hours INTEGER,   
Department CHAR(4),   
PRIMARY KEY (Course\_Number),   
UNIQUE (Course\_Name) );  
CREATE TABLE SECTION ( Section\_Identifier INTEGER NOT NULL,   
Course\_Number CHAR(8) NOT NULL,   
Semester VARCHAR(6) NOT NULL,   
Year CHAR(4) NOT NULL,   
Instructor VARCHAR(15),   
PRIMARY KEY (Section\_Identifier),   
FOREIGN KEY (Course\_Number) REFERENCES COURSE (Course\_Number) );  
CREATE TABLE GRADE\_REPORT ( Student\_Number INTEGER NOT NULL,   
Section\_Identifier INTEGER NOT NULL,   
Grade CHAR(2),   
PRIMARY KEY (Student\_Number, Section\_Identifier),   
FOREIGN KEY (Student\_Number) REFERENCES STUDENT (Student\_Number),   
FOREIGN KEY (Section\_Identifier) REFERENCES SECTION (Section\_Identifier) );  
CREATE TABLE PREREQUISITE ( Course\_Number CHAR(8) NOT NULL,   
Prerequisite\_Number CHAR(8) NOT NULL,   
PRIMARY KEY (Course\_Number, Prerequisite\_Number),   
FOREIGN KEY (Course\_Number) REFERENCES COURSE (Course\_Number),   
FOREIGN KEY (Prerequisite\_Number) REFERENCES COURSE (Course\_Number) );

과제2의 data를 insert 하였습니다.  
텍스트이(가) 표시된 사진

자동 생성된 설명  
  
텍스트, 점수판이(가) 표시된 사진

자동 생성된 설명  
  
테이블이(가) 표시된 사진

자동 생성된 설명

insert into student values ('Smith', 17, 1, 'CS');   
insert into student values ('Brown', 8, 2, 'CS');   
insert into student values ('Gauss', 9, 4, 'MATH');   
insert into student values ('Jones', 15, 4, 'CS');   
insert into student values ('Patel', 22, 4, 'CS');  
  
insert into course values   
('Intro to Computer Science', 'CS1310', 4, 'CS'), ('Data Structures', 'CS3320', 4, 'CS'), ('Discrete Mathematics', 'MATH2410', 3, 'MATH'), ('Database', 'CS3380', 3, 'CS');  
  
insert into section values   
(85, 'MATH2410', 'Fall', 2007, 'King'), (92, 'CS1310', 'Fall', 2007, 'Anderson'), (102, 'CS3320', 'Spring', 2008, 'Knuth'), (112, 'MATH2410', 'Fall', 2008, 'Chang'), (119, 'CS1310', 'Fall', 2008, 'Anderson'), (135, 'CS3380', 'Fall', 2008, 'Stone');  
  
insert into grade\_report values   
(17, 112, 'B'), (17, 119, 'C'), ( 8, 85, 'A'), ( 8, 92, 'A'), ( 8, 102, 'B'), ( 8, 135, 'A'), ( 9, 85, 'A'), ( 9, 112, 'A'), (15, 92, 'B+'), (15, 119, 'B'), (22, 102, 'A-'), (22, 119, 'B-'), (22, 135, 'B');  
  
insert into prerequisite values   
('CS3380', 'CS3320'), ('CS3380', 'MATH2410'), ('CS3320', 'CS1310');

1. Retrieve the names of all senior students majoring in 'CS'.

텍스트이(가) 표시된 사진

자동 생성된 설명  
CS가 Major 인 학생들이 모두 출력 되었습니다.  
  
**SELECT NAME  
FROM STUDENT  
WHERE Major='CS';**

2. Retrieve the names of all courses taught by professor King in 2007 or 2008.

텍스트이(가) 표시된 사진

자동 생성된 설명  
**SELECT Course.Course\_Name  
FROM Course, Section  
WHERE Section.Course\_Number = Course.Course\_Number AND  
Section.Instructor = 'King' AND  
(Section.Year = '2007' OR Section.Year = '2008');**

3. For each section taught by professor King, retrieve the course number, semester, year, and number of students who took the section.

텍스트이(가) 표시된 사진

자동 생성된 설명  
 **SELECT Section.Course\_Number, Section.Semester, Section.Year, COUNT(\*) as number\_of\_student  
FROM Section, Grade\_Report   
WHERE Section.Instructor='King' AND Section.Section\_Identifier=Grade\_Report.Section\_Identifier   
GROUP BY Section.Course\_Number, Section.Semester, Section.Year;**

4. Retrieve the name and transcript of each senior student (Class=4) majoring in CS. Transcript includes course name, course number, credit hours, semester, year, and grade for each course completed by the student.

텍스트이(가) 표시된 사진

자동 생성된 설명

**SELECT St.name, C.Course\_Name, C.Course\_Number, C.Credit\_hours, S.semester, S.Year, G.Grade   
FROM Student ST, Course C, Section S, Grade\_Report G   
WHERE St.Class=4 AND St.Major='CS' AND ST.Student\_Number=G.Student\_Number AND  
G.Section\_Identifier=S.Section\_Identifier AND S.Course\_Number=C.Course\_Number**;

5. Retrieve the names and major departments of all straight A students (students who have a grade of A in all their courses).   
NOT IN을 이용한 방식  
텍스트이(가) 표시된 사진

자동 생성된 설명  
 **SELECT Name, Major  
FROM Student  
WHERE Student\_Number NOT IN (  
SELECT Student.Student\_Number  
FROM Student, Grade\_Report  
WHERE NOT(Grade = 'A') && Grade\_Report.Student\_Number = Student.Student\_Number);**  
  
NOT EXISTS을 이용한 방식  
텍스트이(가) 표시된 사진

자동 생성된 설명

**SELECT Name, Major  
FROM Student  
WHERE NOT EXISTS (  
SELECT \*  
FROM Grade\_Report  
WHERE NOT(Grade = 'A') AND Student.Student\_Number = Grade\_Report.Student\_Number);**

6. Retrieve the names and major departments of all students who do not have any grade of A in any of their courses.  
A-도 A에 포함 안되는 것으로 하였습니다. NOT IN을 이용한 방식  
텍스트이(가) 표시된 사진

자동 생성된 설명  
 **SELECT Name, Major  
FROM Student  
WHERE Student\_Number NOT IN (  
SELECT Student.Student\_Number  
FROM Student, Grade\_Report  
WHERE Grade = 'A' && Grade\_Report.Student\_Number = Student.Student\_Number);**  
  
NOT EXISTS를 이용한 방식

텍스트이(가) 표시된 사진

자동 생성된 설명  
**SELECT Name, Major  
FROM Student  
WHERE NOT EXISTS (  
SELECT \*  
FROM Grade\_Report  
WHERE Grade = 'A' AND Grade\_Report.Student\_Number = Student.Student\_Number);**7. Insert a new course <'Knowledge Engineering','CS4390', 3,'CS'>.   
텍스트이(가) 표시된 사진

자동 생성된 설명정상적으로 INSERT된 것을 알 수 있습니다.  
**INSERT INTO COURSE  
VALUES ('Knowledge Engineering','CS4390', 3,'CS');**

8. Change the class of student 'Smith' to 2.

텍스트이(가) 표시된 사진

자동 생성된 설명  
Smith의 Class가 2로 바뀐 것을 알 수 있습니다.  
**UPDATE STUDENT  
SET CLASS = 2  
WHERE NAME = 'Smith';**

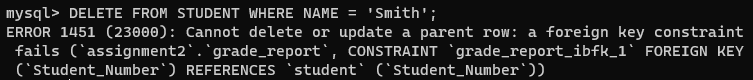
9. Insert a new student <'Johnson', 25, 6, 'MATH'> in the database.   
- 단, class가 1보다 작으면 1로, 4보다 크면 4로 강제로 변경하여 insert하는 trigger를 작성한 후 trigger가 정상 작동하는 것을 보일 것.

텍스트이(가) 표시된 사진

자동 생성된 설명  
다음과 같이 1보다 작으면 CLASS 를 1로, 4보다 크면 4로 설정 하였습니다.  
텍스트이(가) 표시된 사진

자동 생성된 설명  
INSERT 하였을 때 Johnson의 class가 6 인데, trigger에 의해 4로 바뀐 것을 알 수 있습니다.  
텍스트이(가) 표시된 사진

자동 생성된 설명  
0으로 insert 하였을 때는 trigger에 의해 자동으로 1로 변경이 됩니다.

**delimiter //  
CREATE TRIGGER CLASS\_CHECK BEFORE INSERT ON STUDENT  
FOR EACH ROW  
BEGIN  
IF NEW.CLASS < 1 THEN  
SET NEW.CLASS = 1;  
ELSEIF NEW.CLASS > 4 THEN  
SET NEW.CLASS = 4;  
END IF;  
END;//  
delimiter ;  
INSERT INTO STUDENT  
VALUES ('Johnson', 25, 6, 'MATH');**   
  
10. Delete the record for the student whose name is 'Smith' and student number is 17.   
-이때 FK제약조건으로 인하여 delete문이 실행이 안되는 것을 알 수 있음.   
-즉, grade\_report에서 해당 투플을 먼저 delete를 해야 student에서 지울 수 있음   
- trigger를 작성하여 grade\_report에서 먼저 삭제되도록 만들 것.  
   
fk로 인해 delete가 안되는 것 입니다.

텍스트이(가) 표시된 사진

자동 생성된 설명  
정상적으로 삭제가 되는 것을 알 수 있습니다.  
텍스트이(가) 표시된 사진

자동 생성된 설명  
smith 가 삭제 된 것을 알 수 있습니다.

**delimiter //  
CREATE TRIGGER DELETE\_CHECK BEFORE DELETE ON STUDENT  
FOR EACH ROW  
BEGIN  
DELETE FROM GRADE\_REPORT WHERE OLD.STUDENT\_NUMBER;  
END;//  
delimiter ;  
DELETE FROM STUDENT WHERE NAME = 'Smith';**