

데이터베이스 프로그래밍

과제2

교수명: 이인규

제출일: 10월 18일

학과: 컴퓨터공학과

학번: 5645640

이름: 제갈건

내용

SQL문	2
결과 이미지	6

SQL문

```
USE myboard;
```

```
-- 학생(Student) 테이블을 생성
```

```
-- student_id: 학생의 학번을 나타내는 고유 번호로 기본 키로 설정
```

```
-- name: 학생의 이름을 저장
```

```
CREATE TABLE Student (
```

```
    student_id INT PRIMARY KEY, -- 학번 (기본 키)
```

```
    name VARCHAR(50) -- 이름
```

```
);
```

```
-- 과목(Course) 테이블을 생성
```

```
-- course_id: 과목을 식별하는 고유 번호로 자동 증가(AUTO_INCREMENT) 설정
```

```
-- course_name: 과목명을 저장
```

```
-- professor_name: 과목을 담당하는 교수의 이름을 저장
```

```
CREATE TABLE Course (
```

```
    course_id INT PRIMARY KEY AUTO_INCREMENT, -- 과목 ID (기본 키, 자동 증가)
```

```
    course_name VARCHAR(50), -- 과목명
```

```
    professor_name VARCHAR(50) -- 담당 교수명
```

```
);
```

```
-- 성적(Enrollment) 테이블을 생성
```

```
-- 학생과 과목 간의 관계를 나타내며 학생이 어떤 과목에서 어떤 성적을 받았는지 저장
```

```
-- student_id: 학생 테이블의 학번과 외래 키로 연결
```

```
-- course_id: 과목 테이블의 과목 ID와 외래 키로 연결
```

-- grade: 학생의 성적을 문자로 저장 (A, B, C 등등)

-- 이 테이블은 학생-과목 관계를 나타내는 다대다(N:N) 관계를 표현하며, 성적도 함께 저장

CREATE TABLE Enrollment (

 student_id INT, -- 학생 ID (외래 키)

 course_id INT, -- 과목 ID (외래 키)

 grade CHAR(1), -- 성적 (한 글자)

 PRIMARY KEY (student_id, course_id), -- 학생 ID와 과목 ID의 복합 기본 키

 FOREIGN KEY (student_id) REFERENCES Student(student_id), -- Student 테이블의 student_id와 외래 키 관계

 FOREIGN KEY (course_id) REFERENCES Course(course_id) -- Course 테이블의 course_id와 외래 키 관계

);

-- 학생(Student) 테이블에 데이터 삽입

-- 학번과 이름을 삽입

INSERT INTO Student (student_id, name) VALUES

(4407153, '이인규'), -- 학번 4407153의 학생 이름은 '이인규'

(4407155, '김동수'), -- 학번 4407155의 학생 이름은 '김동수'

(4408100, '김철수'); -- 학번 4408100의 학생 이름은 '김철수'

-- 과목(Course) 테이블에 데이터 삽입

-- 과목명과 담당 교수명을 삽입

INSERT INTO Course (course_name, professor_name) VALUES

('데이터베이스', '홍길동'), -- '데이터베이스' 과목을 담당하는 교수는 '홍길동'

('C언어', '아무개'), -- 'C언어' 과목을 담당하는 교수는 '아무개'

('자바스크립트', '천재'); -- '자바스크립트' 과목을 담당하는 교수는 '천재'

-- 성적(Enrollment) 테이블에 데이터 삽입

-- 학생이 수강한 과목과 받은 성적을 기록

INSERT INTO Enrollment (student_id, course_id, grade) VALUES

(4407153, 1, 'A'), -- 학번 4407153의 학생(이인규)이 과목 ID 1(데이터베이스)에서 'A'를
받음

(4407153, 2, 'A'), -- 학번 4407153의 학생이 과목 ID 2(C언어)에서 'A'를 받음

(4407155, 1, 'B'), -- 학번 4407155의 학생(김동수)이 과목 ID 1(데이터베이스)에서 'B'를
받음

(4407155, 3, 'C'), -- 학번 4407155의 학생이 과목 ID 3(자바스크립트)에서 'C'를 받음

(4408100, 1, 'B'), -- 학번 4408100의 학생(김철수)이 과목 ID 1(데이터베이스)에서 'B'를
받음

(4408100, 2, 'A'), -- 학번 4408100의 학생이 과목 ID 2(C언어)에서 'A'를 받음

(4408100, 3, 'B'); -- 학번 4408100의 학생이 과목 ID 3(자바스크립트)에서 'B'를 받음

-- Student 테이블의 모든 데이터를 조회

SELECT * FROM Student;

-- Course 테이블의 모든 데이터를 조회

SELECT * FROM Course;

-- Enrollment 테이블의 모든 데이터를 조회

SELECT * FROM Enrollment;

-- 학생의 학번, 이름, 수강 과목, 담당 교수, 성적을 조회하는 쿼리

-- Enrollment 테이블과 Student, Course 테이블을 조인하여 원하는 데이터를 가져옴

-- 각 학생이 수강한 과목과 성적을 알 수 있음

```
SELECT S.student_id, S.name, C.course_name, C.professor_name, E.grade
```

```
FROM Enrollment E
```

```
JOIN Student S ON E.student_id = S.student_id -- Enrollment 테이블과 Student 테이블을  
학번(student_id)으로 조인
```

```
JOIN Course C ON E.course_id = C.course_id -- Enrollment 테이블과 Course 테이블을 과  
목 ID(course_id)로 조인
```

```
ORDER BY S.student_id; -- 학번 순으로 결과 정렬
```

결과 이미지

student_id	name
4407153	이인규
4407155	김동수
4408100	김철수
NULL	NULL

student

course_id	course_name	professor_name
1	데이터베이스	홍길동
2	C언어	아무개
3	자바스크립트	천재
NULL	NULL	NULL

Course

student_id	course_id	grade
4407153	1	A
4407153	2	A
4407155	1	B
4407155	3	C
4408100	1	B
4408100	2	A
4408100	3	B
NULL	NULL	NULL

Enrollment

Output			
Action Output			
#	Time	Action	
✓ 1	13:38:43	use myboard	
✓ 2	13:38:43	select * from post LIMIT 0, 1000	
✓ 3	13:38:43	CREATE TABLE Student (student_id INT PRIMARY KEY, name VARCHAR(50))	
✓ 4	13:38:43	CREATE TABLE Course (course_id INT PRIMARY KEY AUTO_INCREMENT, course_name VA...	
✓ 5	13:38:43	CREATE TABLE Enrollment (student_id INT, course_id INT, grade CHAR(1), PRIMARY KE...	
✓ 6	13:38:43	INSERT INTO Student (student_id, name) VALUES (4407153, '이인규'), (4407155, '김동수'), (440810...	
✓ 7	13:38:43	INSERT INTO Course (course_name, professor_name) VALUES ('데이터베이스', '홍길동'), ('C언어', '...	
✓ 8	13:38:44	INSERT INTO Enrollment (student_id, course_id, grade) VALUES (4407153, 1, 'A'), (4407153, 2, 'A'), (...	
✓ 9	13:38:44	SELECT S.student_id, S.name, C.course_name, C.professor_name, E.grade FROM Enrollment E JOIN...	
✓ 10	13:39:52	use myboard	
✓ 11	13:39:52	select * from post LIMIT 0, 1000	
✓ 12	13:41:20	use myboard	
✓ 13	13:41:20	SHOW TABLES	
✓ 14	13:42:33	use myboard	
✓ 15	13:42:33	SELECT * FROM Student LIMIT 0, 1000	
✓ 16	13:42:33	SELECT * FROM Course LIMIT 0, 1000	
✓ 17	13:42:33	SELECT * FROM Enrollment LIMIT 0, 1000	
✓ 18	13:45:58	USE myboard	
✓ 19	13:45:58	SELECT S.student_id, S.name, C.course_name, C.professor_name, E.grade FROM Enrollment E JOIN...	

Log

Result Grid					
Filter Rows: Export: Wrap Cell Content:					
	student_id	name	course_name	professor_name	grade
▶	4407153	이인규	데이터베이스	홍길동	A
	4407153	이인규	C언어	아무개	A
	4407155	김동수	데이터베이스	홍길동	B
	4407155	김동수	자바스크립트	천재	C
	4408100	김철수	데이터베이스	홍길동	B
	4408100	김철수	C언어	아무개	A
	4408100	김철수	자바스크립트	천재	B

Result 8 ×
Read Only

result