

테이블 생성

```
CREATE TABLE 학생
( 학번 INT,
  성명 CHAR(5) NOT NULL,
  학과 CHAR(10),
  학점 CHAR(1),
  PRIMARY KEY (학번),
  FOREIGN KEY (학과) REFERENCES 수강(학과)
);
```

확인하기

```
INSERT INTO 학생
VALUES ( 20190101, '홍길동', '컴퓨터공학과', 'A');

SELECT * FROM 학생;
```

테이블 속성 추가

```
ALTER TABLE 학생 ADD 학년 INT ;
```

확인하기

```
SELECT * FROM 학생;

UPDATE 학생
SET 학년 = 1
WHERE 성명 = '홍길동';

SELECT * FROM 학생;
```

튜플 추가

[컴퓨터공학과]

```
INSERT INTO 학생
VALUES ( 20190101, '홍길동', '컴퓨터공학과', 'A', 1);
```

```
INSERT INTO 학생
VALUES ( 20190102, '김가이', '컴퓨터공학과', 'A', 1);
```

```
INSERT INTO 학생
VALUES ( 20190103, '김루리', '컴퓨터공학과', 'B', 1);
```

```
INSERT INTO 학생
VALUES ( 20180104, '이효인', '컴퓨터공학과', 'A', 2);
```

```
INSERT INTO 학생
VALUES ( 20170105, '박설리', '컴퓨터공학과', 'A', 3);
```

[영어영문학과]

```
INSERT INTO 학생
VALUES ( 20190201, '박현수', '영어영문학과', 'A', 1);
```

```
INSERT INTO 학생
VALUES ( 20190202, '정환희', '영어영문학과', 'C', 1);
```

```
INSERT INTO 학생
VALUES ( 20190203, '이천수', '영어영문학과', 'B', 1);
```

```
INSERT INTO 학생
VALUES ( 20180204, '한요한', '영어영문학과', 'C', 2);
```

```
INSERT INTO 학생
VALUES ( 20170205, '구수진', '영어영문학과', 'A', 3);
```

```
INSERT INTO 학생
VALUES ( 20180206, '한요한', '영어영문학과', 'C', 2);
```

```
INSERT INTO 학생
VALUES ( 20170207, '박설리', '영어영문학과', 'A', 3);
```

[경영학과]

```
INSERT INTO 학생
VALUES ( 20190301, '박서연', '경영학과', 'A', 1);
```

```
INSERT INTO 학생
VALUES ( 20190302, '홍민서', '경영학과', 'F', 1);
```

```
INSERT INTO 학생
VALUES ( 20190303, '정여름', '경영학과', 'B', 1);
```

```
INSERT INTO 학생
VALUES ( 20160304, '송지우', '경영학과', 'A', 4);
```

```
INSERT INTO 학생
VALUES ( 20160305, '권수아', '경영학과', 'A', 4);
```

```
INSERT INTO 학생
VALUES ( 20180306, '강서윤', '경영학과', 'F', 2);
```

```
INSERT INTO 학생
VALUES ( 20170307, '최하린', '경영학과', 'B', 3);
```

```
INSERT INTO 학생
VALUES ( 20160307, '안아람', '경영학과', 'A', 4);
```

[뮤지컬학과]

```
INSERT INTO 학생
VALUES ( 20160405, '원빈', '뮤지컬학과', 'C', 4);
```

```
INSERT INTO 학생
VALUES ( 20180406, '장동건', '뮤지컬학과', 'C', 2);
```

```
INSERT INTO 학생
VALUES ( 20170407, '전지현', '뮤지컬학과', 'B', 3);
```

```
INSERT INTO 학생
VALUES ( 20160407, '이나영', '뮤지컬학과', 'A', 4);
```

튜플 검색

1) <학생> 테이블에서 성명이 '홍길동'인 학번, 성명, 학과, 학점, 학년을 조회하시오.

```
SELECT 학번, 성명, 학과, 학점, 학년
FROM 학생
WHERE 성명 = '홍길동';
```

2) <학생> 테이블에서 2학년 이상인 모든 튜플을 출력하시오.

```
SELECT *
FROM 학생
WHERE 학년 >= 2;
```

3) <학생> 테이블에서 학생들의 모든 튜플을 조회하되, 학점을 기준으로 오름차순으로 정렬하고 2차 정렬기준으로 학년을 내림차순으로 정렬하시오.

```
SELECT *
FROM 학생
ORDER BY 학점, 학년 DESC;
```

4) <학생> 테이블에서 학과가 '컴퓨터공학과', '경영학과'인 모든 튜플을 조회하시오.

```
SELECT *
FROM 학생
WHERE 학과 IN ('컴퓨터공학과', '경영학과');
```

```
SELECT *
FROM 학생
WHERE 학과 = '컴퓨터공학과'
OR 학과 = '경영학과';
```

5) <학생> 테이블에서 학점이 'A'인 경우, 학과, 튜플의 수를 'A학점자 수'로 검색하되 학과별로 출력하시오.

```
SELECT 학과, COUNT(*) AS 'A학점자 수'
FROM 학생
WHERE 학점 = 'A'
GROUP BY 학과;
```

6) <학생> 테이블에서 학점이 'C'인 경우, 학과, 튜플의 수를 'C학점자 수'로 검색하되 학과별 해당인원이 3명이 상인 학과만 조회하시오.

```
SELECT 학과, COUNT(*) AS 'C학점자 수'
FROM 학생
WHERE 학점 = 'C'
GROUP BY 학과 HAVING COUNT(*) >= 2;
```

7) <학생> 테이블에서 김씨를 조회하시오.

```
SELECT * FROM 학생
WHERE 성명 LIKE '김%';
```

조인

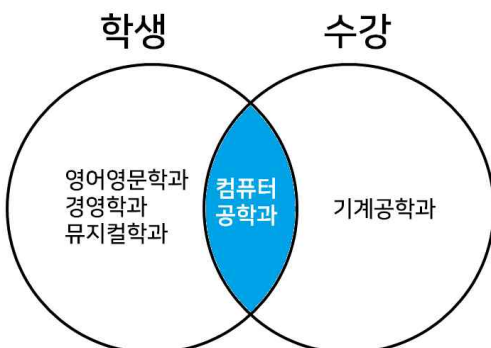
```
CREATE TABLE 수강
(
  학과 CHAR(10),
  과목 CHAR(10),
  교수 CHAR(10),
  학점 INT,
);
```

```
INSERT INTO 수강 (학과, 과목, 교수, 학점)
VALUES ('컴퓨터공학과', 'JAVA', '박강', 3);

INSERT INTO 수강 (학과, 과목, 교수, 학점)
VALUES ('기계공학과', '유체역학', '김준수', 3);
```

```
SELECT *
FROM 학생
JOIN 수강
ON 학생.학과 = 수강.학과;
```

```
SELECT *
FROM 학생, 수강
WHERE 학생.학과 = 수강.학과;
```



조건

```
SELECT *
FROM 학생
WHERE 학과 = '컴퓨터공학과'
OR 학과 = '경영학과';
```

```
SELECT *
FROM 학생
WHERE 학과 IN ('컴퓨터공학과', '경영학과');
```

```
SELECT *
FROM 학생
WHERE 학과 = '컴퓨터공학과'
AND 학점 = 'C';
```

```
SELECT *
FROM 학생
WHERE 학점 BETWEEN 'A' AND 'B' ;
```

VIEW 생성

```
CREATE VIEW A학생
AS SELECT * FROM 학생
WHERE 학점 = 'A';
```

W3Schools : 서버환경에서 직접해보기!

https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all