SQooL CheatSheet 1

사용된 샘플 데이터

student
학번, 이름, 학과, 지도교수, 학년, 생년월일, 연락처, 주소, 마일리지

subject
과목, 번호, 학과, 번호, 과목명, 이수구분, 학점

scholarship
학번, 성적, 장학금, 근로장학금, 국가장학금

테이블에서 출력하기

```
SELECT (10 + 2) / 2;
```

사칙연산이 가능합니다.

```
SELECT 'hello', 'world';
```

column을 나눠서 출력합니다.

테이블에서 데이터 가져오기

```
SELECT * FROM student;
```

student 테이블에서 모든 column을 가져옵니다.

```
SELECT 이름, 학과 FROM student;
```

student 테이블에서 이름, 학과 column을 가져옵니다.

```
SELECT 학번, 학년, 이름
FROM student
ORDER BY 학년 DESC;
```

student 테이블에서 학번, 학년, 이름 column을 가져오는데, 학년순으로 내림차순하여 가져옵니다.

```
SELECT 학번, 학년, 이름
FROM student
ORDER BY 학년 ASC;
```

student 테이블에서 학번, 학년, 이름 column을 가져오는데, 학년순으로 오름차순하여 가져옵니다.

```
SELECT 학번,
       이름 AS 성명,
       연락처 AS 휴대폰번호
FROM student;
```

student 테이블에서 학번, 이름, 연락처를 가져오는데, 이름은 성명으로, 연락처는 휴대폰번호로 표기합니다.

```
SELECT DISTINCT 학과번호 AS 학과이름
FROM subject;
```

subject 테이블에서 학과번호를 학과이름으로 가져오고, 중복은 허락하지 않습니다.

```
SELECT 이름 AS 성명, 이름
       || '의 지도교수는 '
       || 지도교수
       || '입니다.' AS 지도교수
FROM student;
```

student 테이블에서 이름을 가져오는데 성명으로 가져오고, column을 통해 새로운 문장을 만듭니다. 사칙연산도 들어갈 수 있습니다.

테이블에서 조건에 맞는 데이터 가져오기 -1

```
SELECT 학년, 이름, 연락처
FROM student
WHERE 학년 >= 3;
```

student 테이블에서 학년, 이름, 연락처를 가져오는데, 학년이 3학년 이상인 데이터를 가져옵니다.

```
SELECT 학년, 이름, 연락처, 지도교수
FROM student
WHERE 학년 = 4 and 지도교수 = '일호준';
```

student 테이블에서 학년, 이름, 연락처, 지도교수를 가져오는데, 학년이 4학년이면서 동시에 지도교수가 '일호준' 인 데이터를 가져옵니다.

```
SELECT 학년, 이름, 연락처, 지도교수
FROM student
WHERE 학년 = 4 OR 지도교수 = '일호준';
```

student 테이블에서 학년, 이름, 연락처, 지도교수를 가져오는데, 학년이 4학년이거나 지도교수가 '일호준' 인 데이터를 가져옵니다.

```
SELECT 학년, 이름, 지도교수
FROM student
WHERE 지도교수 LIKE '%호준';
```

student 테이블에서 학년, 이름, 지도교수를 가져오는데, 지도교수가 'O호준' 인 데이터를 가져옵니다. 글자수와 상관없는 % 와 일드카드를 사용했습니다.

SQooL CheatSheet 2

사용된 샘플 데이터

student
학번, 이름, 학과, 지도교수, 학년, 생년월일, 연락처, 주소, 마일리지

subject
과목, 번호, 학과, 번호, 과목명, 이수구분, 학점

scholarship
학번, 성적, 장학금, 근로장학금, 국가장학금

테이블에서 조건에 맞는 데이터 가져오기 -2

```
SELECT 학년, 이름, 연락처, 지도교수
FROM student
WHERE 지도교수 LIKE '김_';
```

student 테이블에서 학년, 이름, 지도교수를 가져오는데, 지도교수가 '김' 으로 시작하는 2글자인 데이터를 가져옵니다. 글자수와 상관있는 _ 와일드카드를 사용했습니다.

```
SELECT 학번, 이름, 연락처, 지도교수
FROM student
WHERE 학번 >= 201900001 AND 학번 <= 201900030;
```

student 테이블에서 학번, 이름, 연락처, 지도교수를 가져오는데, 학번이 201900001부터 201900030까지 해당하는 학생을 모두 가져옵니다.

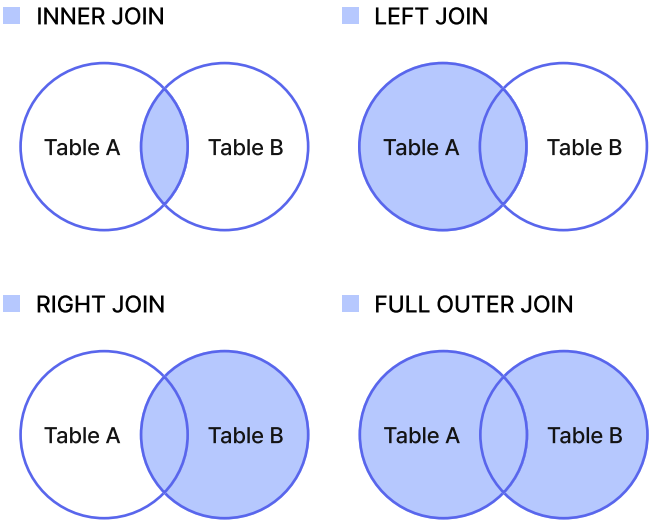
```
SELECT 학번, 국가장학금
FROM scholarship
WHERE 국가장학금 IS 2000000;
--WHERE 국가장학금 IS NOT NULL;
```

scholarship 테이블에서 학번, 국가장학금을 가져오는데, 200만원을 받은 학생만 가져옵니다.

```
SELECT 학번, 이름, 지도교수
FROM student
WHERE 지도교수 IN ('일호준', '삼호준');
```

student 테이블에서 학번, 이름, 지도교수를 가져오는데, 지도교수가 '일호준' 이거나 '삼호준' 인 데이터를 가져옵니다.

다중 테이블에서 데이터 가져오기




```
SELECT student.학번,
       student.이름 AS 학생이름,
       professor.이름 AS 교수이름,
       professor.이메일 AS 교수이메일
FROM student
INNER JOIN professor
--RIGHT JOIN professor
--LEFT JOIN professor
--FULL JOIN professor
ON student.지도교수 = professor.이름;
```

student 테이블과 professor를 교수이름으로 합친 다음, 학번, 학생이름, 교수이름, 교수이메일을 출력합니다. student 테이블에는 교수이메일이 없습니다. 비어 있는 값이 있을 경우, 어느 테이블을 기준으로 값을 채울지 RIGHT, LEFT, FULL로 설정할 수 있습니다.

```
SELECT student.학번,
       student.이름 AS 학생이름,
       professor.이름 AS 교수이름,
       professor.이메일 AS 교수이메일,
       scholarship.국가장학금
FROM student
INNER JOIN professor
ON student.지도교수 = professor.이름
INNER JOIN scholarship
ON student.학번 = scholarship.학번;
```

student, professor, scholarship 테이블에서 먼저 student 테이블과 professor 테이블을 교수 이름으로 합치고, student 테이블과 scholarship 테이블을 학번 기준으로 합칩니다. 학번, 학생이름, 교수이름, 교수이메일, 국가장학금을 출력합니다.

함수 -1

```
SELECT DATE();
--SELECT DATE() + 10;
--SELECT TIME() + 10;
```

현재 날짜와 시간을 출력합니다.

```
SELECT MAX(마일리지) FROM student;
```

마일리지 최댓값을 구합니다.

```
SELECT MIN(마일리지) FROM student;
```

마일리지 최솟값을 구합니다.

SQooL CheatSheet 3

사용된 샘플 데이터

student
학번, 이름, 학과, 지도교수, 학년, 생년월일, 연락처, 주소, 마일리지

subject
과목, 번호, 학과, 번호, 과목명, 이수구분, 학점

scholarship
학번, 성적, 장학금, 근로장학금, 국가장학금

함수 -2

```
SELECT SUM(마일리지) FROM student;
```

마일리지 전체 합계를 구합니다.

```
SELECT AVG(마일리지) FROM student;
```

마일리지 평균을 구합니다.

```
SELECT COUNT(DISTINCT 학과번호) AS 학과갯수
FROM subject;
```

subject 테이블에서 학과의 갯수를 구합니다.

```
SELECT 2023-SUBSTR(생년월일, 1, 4) AS 나이
FROM student;
```

생년월일에서 년도(2023)만 가져와서 뺀 나이를 계산합니다.

```
SELECT 이름,
       LENGTH(이름) AS 이름길이,
       REPLACE(이름, SUBSTR(이름, 2, 1), '*')
       AS 별표채운이름
FROM student;
```

이름과 이름 길이, 이름에서 중간 이름만 *로 표시한 값들을 출력합니다.

그룹핑

```
SELECT 학년, AVG(마일리지)
FROM student
GROUP BY 학년;
```

학년별 마일리지 평균을 출력합니다.

```
SELECT 학과, AVG(마일리지)
FROM student
GROUP BY 학과
HAVING 학년 > 2;
```

학과별 마일리지 평균을 출력하는데 2학년 이상인 학생의 마일리지를 출력합니다.

테이블 관리하기

```
CREATE TABLE 제품 (
    제품번호 INT PRIMARY KEY,
    제품이름 VARCHAR NOT NULL,
    가격 INT DEFAULT 0
);
```

제품 테이블을 생성합니다.

```
INSERT INTO 제품 (제품번호, 제품이름, 가격)
VALUES (1, '버그잡는 개리씨 키링', 12500);
--SELECT * FROM 제품
```

생성한 테이블에 데이터를 넣습니다.

```
UPDATE 제품
SET 제품이름 = '위니브 스티커 팩',
    가격 = 3500
WHERE 제품번호 = 1;
--SELECT * FROM 제품
```

데이터를 수정합니다.

```
DELETE FROM 제품 WHERE 제품번호 = 1;
```

1번째 제품을 삭제합니다.
(주의 : WHERE를 적지 않으면 전부 삭제됩니다.)

```
DROP TABLE 제품;
```

테이블을 제거합니다.