

✨ SQL은 외우는 게 아닙니다!

board.sql

CLI

## 1. “Board” 스키마 생성

```
CREATE DATABASE Board;  
USE Board;
```

## 2. 사용자 테이블 생성

```
CREATE TABLE users (  
    id INT NOT NULL AUTO_INCREMENT,  
    name VARCHAR(30) NOT NULL,  
    job VARCHAR(100),  
    birth DATE, 📌 “YYYY-MM-DD”  
    PRIMARY KEY ( id )  
);
```

## 3. 사용자 데이터 삽입

```
INSERT INTO users (name, job, birth)  
VALUES (“gongu”, “actor”, “800123”);
```

#### 4. 게시글 테이블 생성

```
CREATE TABLE posts (  
    id INT NOT NULL AUTO_INCREMENT,  
    title VARCHAR(100) NOT NULL,  
    content VARCHAR(2000),  
    created_at TIMESTAMP DEFAULT NOW(),  
    PRIMARY KEY ( id )  
);
```

#### 5. 게시글 데이터 삽입

```
INSERT INTO posts (title, content)  
VALUES ("title3", "content3");
```

#### 6. 게시글 테이블에 수정일자 컬럼 추가

```
ALTER TABLE posts  
ADD COLUMN updated_at DATETIME  
DEFAULT NOW()  
ON UPDATE NOW();
```

#### 7. 게시글 테이블 id 2 수정

```
UPDATE posts  
SET content = "updated!"
```

WHERE id = 2;

## 8. 게시물 테이블에 작성자 컬럼 FK 추가

### 1) TABLE 컬럼 추가

ALTER TABLE posts

ADD COLUMN user\_id INT;

### 2) FK 설정

ALTER TABLE posts

ADD FOREIGN KEY(user\_id)

REFERENCES users(id);

-

**[fk 제약조건 이름짓기] : “컨벤션”**

**fk\_기준 테이블명\_참조테이블명\_참조키**

**cartItems.user\_id > users.id :**

**fk\_cartItems\_users\_id**

**likes.user\_id > users.id :**

**fk\_likes\_users\_id**

-

## 9. 🚀조인🚀

SELECT \* FROM posts LEFT

**JOIN users ON posts.user\_id = users.id;**

SELECT posts.id, title, content, created\_at,  
updated\_at,  
name, job, birth FROM posts LEFT  
JOIN users ON posts.user\_id = users.id;

auto\_increment\_lock\_mode

10. 조건을 만족하는 행 개수

SELECT count(\*) FROM likes WHERE  
liked\_book\_id=1

11. 좋아요 개수가 포함된 books 테이블 조회

SELECT \*,  
    (SELECT count(\*) FROM likes WHERE  
liked\_book\_id=books.id) AS likes  
FROM books;

12. 개별 도서 조회 시, 사용자가 좋아요를 했는지  
여부를 포함

