구현부:  
 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

Table contents 의 column:

contents\_id int primary key auto\_increment : 게시물들 고유 번호  
bbs\_id int : 게시물이 속한 게시판의 id <-bbs table과 연결됨, FK  
bbs\_title varchar(40) : 게시물의 제목  
bbs\_content varchar(2048) : 게시물의 본문  
bbs\_date datetime : 작성일시  
nickname varchar(20) : 닉네임 <- user table과 밀접함, (FK 아님!)   
bbs\_likes int : 좋아요 수  
bbs\_stars int : 스크랩 수  
bbs\_comments int : 댓글 수  
i\_user int : 유저 고유번호 <- user table과 연결됨, FK  
bbs\_available boolean default 1 : 삭제된 글인지 아닌지 여부 판단

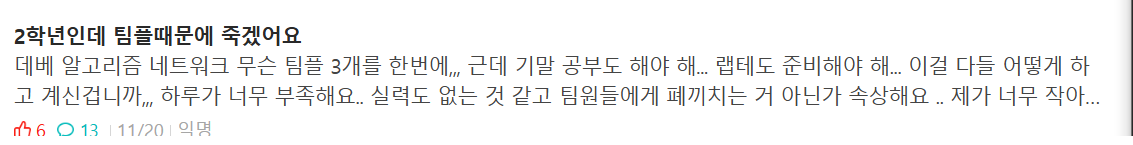
이 column 들 중, bbs\_title, bbs\_content, bbs\_date, nickname, bbs\_llikes, bbs\_comments record 들이 사용자에게 보여진다.

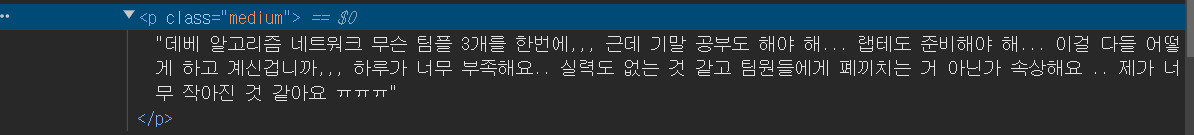
Contents table은 게시판 종류 상관 없이 모든 게시물의 정보를 가지고 있는 Table이다.

**추가설명**

**게시물 고유번호 (contents\_id)**말 그대로 게시물의 고유 번호입니다. PK로 설정되어 있으며, auto\_increment 입니다.  
때문에, 작성 순서대로 contents\_id가 부여되며, 이 값이 높을수록 최근에 생성된 Data입니다.  
이 점을 이용하여, 시간순서대로 정렬할 때 bbs\_date 보단 contents\_id로 정렬하는게 속도가 더 빠릅니다.

**작성일시(bbs\_date)**   
현재 시간으로부터 60초 이내 : ‘방금’  
현재 시간으로부터 60분 이내 : ‘X분 전’  
오늘 작성됨 : ‘Hour:Minutes’  
그 이외 : ‘Month/Date’ 의 형식을 따른다.

**게시물의 본문(bbs\_content)**겉으로만 보면 특정 글자 수, 행 수에 따라 record를 가공하여 가져오는 것 처럼 보이지만,  




놀랍게도 record 전체를 가져온 후 프론트엔드에게 맡겨버림. (실화)  
조금 개선해서 글자 수가 50을 넘어갈 때, Lorem ipsum dolor sit amet… , ‘…’을 추가  
(솔직히 어떤게 더 나은 선택인지는 모르겠습니다. 쿼리에서 string 처리하면서 걸리는 시간이 더 걸릴거 같기도 합니다.)

**닉네임(nickname)**글을 작성할 때, 익명여부를 결정할 수 있습니다. 물론 DB에 익명여부 판단 column을 만든 후,  
data를 가져올 때 해당 column에 따라 nickname을 익명으로 보이게 만들 수도 있습니다.   
nickname, ‘Anonymous’ column이 필요합니다.

그러나, 익명으로 글을 작성하여 post를 할 때, nickname을 ‘익명’으로 하여 post 해도 됩니다.  
DB에 nickname이 익명으로 떠도, i\_user 를 통해 유저 정보를 알 수 있습니다.  
-> nickname, i\_user column 사용

(어떤 글의 작성자 정보를 알아보려고 할 때, nickname으로도 사용자를 찾을 수도 있겠지만, int 기반 query가 varchar보다 빠릅니다. 또한, 해당 row의 anonymous column이 1인지 0인지 판단하는 데에 시간이 소요됩니다.)

**좋아요, 댓글 수(bbs\_likes, comments)**좋아요와 댓글 수가 1개 이상일때만 표시가 되는데, 이건 프론트에서 해결해야 할 문제가 아닌가 싶습니다. (보임/안보임 이 2가지 상태이기에)

**삭제된 글인지? (bbs\_available)**사용자가 글을 삭제하는 경우가 있습니다. 하지만 대부분의 서비스는 사용자가 글을 삭제한다고 해도, DB에서 삭제하지는 않습니다. (관리하는 측면에서 편함)  
이를 위해 bbs\_available column이 있으며, default를 1로 설정하여 글이 작성되면 자동으로 1이 할당됩니다. 또한, bbs\_available의 값이 1인 경우에만 사용자에게 노출됩니다.  
( 0의 경우 사용자의 삭제, 관리자의 삭제 등으로 이루어짐)

**코드**사실 이 부분도 구현할까 말까 고민했는데, 만들면 API가 아니라 webpage 만들기에 가까울 거 같아서 설명만 적어 놨습니다.이걸 발표에 넣어도 될지 말지는 잘 모르겠습니다…. 설명 하신다면 실제로 구현하면 이렇게도 바뀔 수 있다고 하는 게 좋지 않을까 싶습니다..

쿼리에 limit 20 구문이 있습니다. Contents\_id 로 내림차순 정렬한 후, 20개의 게시물만 가져옵니다.

그런데 한 페이지에 20개의 data만 가져온다는 부분을 생각해봐야 합니다.  
contents\_id가 1~ 60까지 존재한다면, 1페이지에는 41~60번, 2페이지에는 21~40번, 3페이지에는 1~20번의 data들을 가져와야 합니다.

어떤 방법으로 현재 보고있는 page\_number을 알아 낸 후,

SELECT \* FROM BBS WHERE contents\_id < ? AND bbs\_available = 1 ORDER BY contents\_id DESC LIMIT 20;

? 에는 가장 최신글의 (contents\_id +1) – (현재 page\_number -1 ) \* 20이 들어갑니다.  
가장 최신글은 60번이고, 현재 페이지가 2번이라면, 61 – (2-1) \* 20 = 41이 들어가게되고,   
1~40 까지의 data를 가져온 후 20개의 data만 가져옵니다. (21~40)

페이지 별로 데이터를 가져오고 싶다면, 이렇게 하지 않을까 싶습니다.

실행 결과:

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

중략

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 번호, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

테스트 한다고 값을 바꿨었습니다. 결과가 다를 수도 있을 듯 합니다.