

# DB 모델링 요구사항

## Sample01 - 커뮤니티

기능	설명
회원가입	회원가입을 할 때 <b>이메일, 비밀번호, 이름, 나이</b> 의 정보를 받는다.
로그인	로그인할 때 <b>이메일, 비밀번호</b> 를 활용해서 로그인한다.
게시글 작성	<b>로그인한 사용자만</b> 게시글을 작성할 수 있다. 게시글에는 <b>제목과 내용</b> 을 작성할 수 있고, <b>해시태그</b> 를 달 수 있고, 게시글의 <b>카테고리(정보 게시글, 홍보 게시글 등)</b> 를 고를 수 있다.
게시글 조회	<b>작성자, 게시글 제목, 게시글 내용, 게시글 작성 시간, 좋아요 수, 조회 수</b> 를 조회할 수 있어야 한다.
해시태그로 게시글 조회	<b>특정 해시태그를 가진 모든 게시글을</b> 조회할 수 있어야 한다.
좋아요	로그인한 사용자가 특정 게시글에 <b>좋아요</b> 를 누를 수 있다. 로그인한 사용자가 특정 게시글에 <b>좋아요</b> 취소를 할 수 있다. 특정 게시글에 <b>어떤 사용자들이 좋아요를 눌렀는 지</b> 조회할 수 있어야 한다.
팔로우	특정 사용자가 다른 사용자를 <b>팔로우</b> 할 수 있다. 특정 사용자가 다른 사용자를 <b>언팔로우</b> 할 수 있다. 각 사용자의 <b>팔로우 수를</b> 조회할 수 있어야 한다.
신고	<b>로그인한 사용자가</b> 특정 게시글을 신고할 수 있다. 신고할 때 <b>신고 사유</b> 를 작성해야 한다.

## DB 설계과정

### 1. 저장할 데이터 파악하기

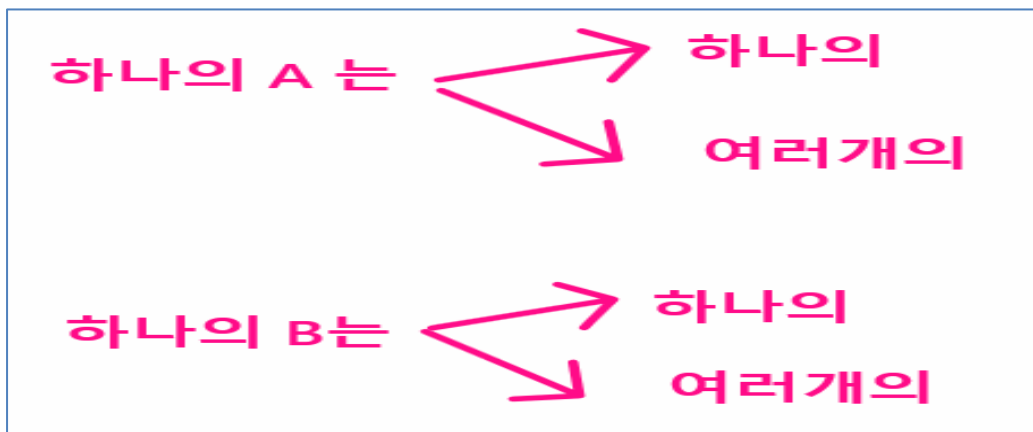
- 이메일, 비밀번호, 이름, 나이
- 게시글 제목, 게시글 내용, 게시글에 달린 해시태그, 게시글의 카테고리, 게시글 작성자, 게시글 작성 시간, 게시글 좋아요 수, 게시글 조회 수
- 어떤 게시글에 어떤 사용자가 좋아요를 눌렀는 지
- 어떤 사용자가 누구를 팔로우 했는 지

- 팔로우 수
- 어떤 사용자가 어떤 게시글을 신고 했는 지, 신고 사유
- 

## 2. 그룹핑해서 분류하기

- 이메일, 비밀번호, 이름, 나이 → **사용자**
- 게시글 제목, 게시글 내용, 게시글에 달린 해시태그, 게시글의 카테고리, 게시글 작성자, 게시글 작성 시간, 게시글 좋아요 수, 게시글 조회 수 → **게시글**
- 어떤 게시글에 어떤 사용자가 좋아요를 눌렀는 지 → **게시글? 사용자?**
- 어떤 사용자가 누구를 팔로우 했는 지 → **사용자?**
- 팔로우 수 → **사용자**
- 어떤 사용자가 어떤 게시글을 신고 했는 지, 신고 사유 → **신고**

## A 와 B 관계 파악 하기



<b>1 : 1</b>	어느쪽의 테이블이든 한쪽의 pk 를 다른 쪽으로 FK 설정한다.
<b>1 : N</b>	1 쪽의 pk 를 다 쪽의 일반속성으로 FK 설정한다.
<b>N : M</b>	새로운 Entity(table)를 만들고 양쪽의 pk 를 새로운 Entity 로 FK 설정한다.

사용자(users)				
id	이메일	비밀번호	이름	나이
1	<a href="mailto:jang@daum.net">jang@daum.net</a>			
2	<a href="mailto:kim@naver.com">kim@naver.com</a>			

### 사용자(1) : 이메일(1) 관계

한명의 사용자는 하나의 이메일을 가진다.

하나의 이메일은 한명의 사용자에게 관리된다.

카테고리(categories)	
id	카테고리명
1	자유게시판
2	자료실게시판

게시글(posts)					
post_no	제목	내용	카테고리_id(FK)	작성자_id(FK)	등록일
1			1	1	
2			1	2	

게시글(N) : 카테고리(1)

하나의 게시글은 하나의 카테고리를 가진다.

하나의 카테고리는 여러 개의 게시글에 사용된다.

게시글(N) : 사용자(1)

하나의 게시글은 한명의 작성자가 등록한다.

한명의 사용자는 여러 개의 게시글을 등록 할 수 있다.

해시태그(hashtags)	
id	이름
1	자바
2	React
3	spring

게시글-해시태그(posts_hashtags)		
id	게시글_no(FK )	해시태그id(FK )
1	1	1
2	1	2

게시글(N) : 해시태그(M)

하나의 게시글은 여러 개의 해시태그를 가진다.

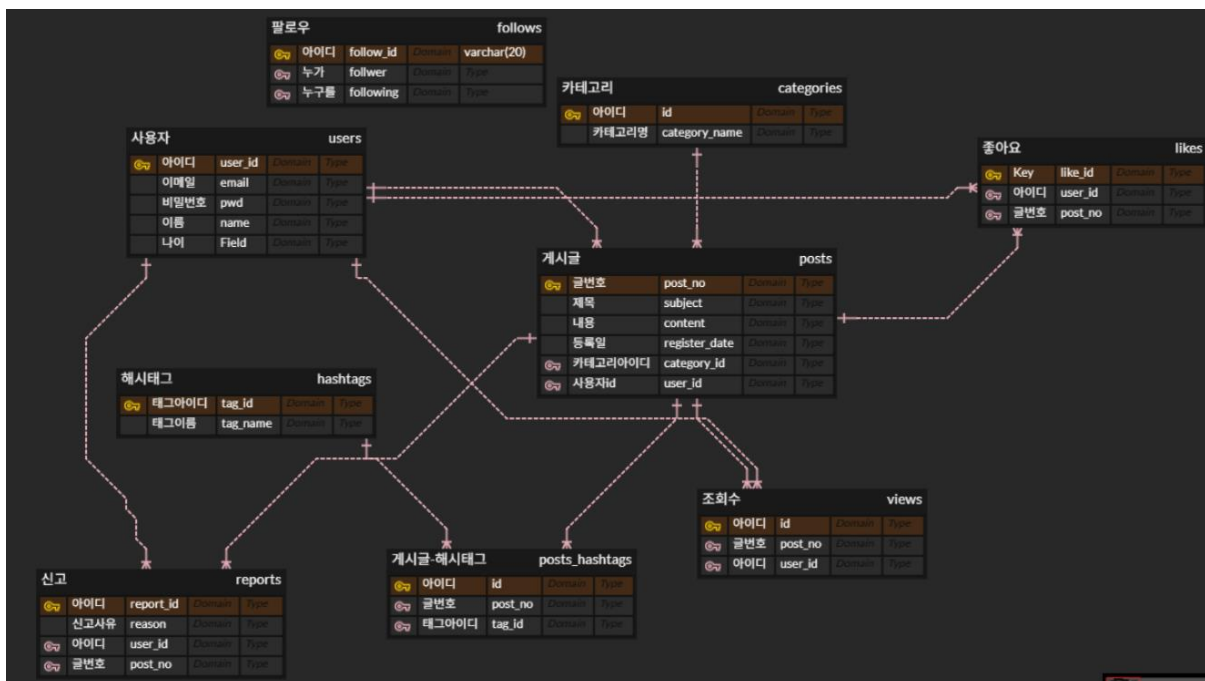
하나의 해시태그는 여러 개의 게시글에 속한다.

조회수(views)		
id	누가(사용자id)	어떤글조회(post_no)
1	1	1
2	2	1

좋아요(likes)		
id	누가좋아요클릭(사용자_id)	어떤게시글(post_no)
1	1	2

팔로우(follows)		
id	누가 팔로우했는지(사용자_id)	누구를 팔로우했는지(사용자_id)

신고(reports)			
id	누가신고했는지(사용자_id)	신고사유	신고한 게시글(post_id)
1	1	욕설이 있어서	1
2	2	비속어가 있어서	1



ENTITY	
Search entity	
Logical Name ▼	Physical Name
게시글	posts
게시글-해시태그	posts_hashtags
사용자	users
신고	reports
조회수	views
좋아요	likes
카테고리	categories
팔로우	follows
해시태그	hashtags

## Sample02 - 쇼핑몰

기능	설명
회원가입	회원가입 시 이메일, 비밀번호, 이름, 주소, 전화번호의 정보를 받는다.
로그인	로그인할 때 이메일과 비밀번호를 활용해서 로그인한다.
상품 등록	로그인한 사용자만 상품을 등록할 수 있다. 상품에는 상품명, 설명, 가격, 재고량, 카테고리(의류, 신발, 가전제품 등)의 정보가 포함된다.
상품 조회	상품명, 설명, 가격, 재고량, 등록 시간을 조회할 수 있다. 특정 카테고리의 상품만 조회할 수 있다.
주문 기능	한 번 주문할 때 여러 개의 상품을 주문할 수 있다. 각 상품의 수량을 정해서 주문할 수 있다. 주문한 여러개의 상품의 가격과 수량을 기록한다. 주문했을 때 상품의 총 가격을 계산할 수 있다. 로그인한 사용자만 주문할 수 있다. 주문 시 배송 정보(이름, 주소, 전화번호)를 입력해야 한다. 주문한 날짜를 조회할 수 있어야 한다.
리뷰 작성	로그인한 사용자만 리뷰를 작성할 수 있다. 상품에 대한 리뷰를 작성하고, 평점(1~5 점)을 매길 수 있다. 다른 사용자들이 작성한 리뷰를 조회할 수 있다.
관리자	관리자 페이지에 접근하려면 관리자용 이메일과 비밀번호를 입력해야 한다.

## DB 설계과정

### 1. 저장할 데이터 파악하기

- 이메일, 비밀번호, 이름, 주소, 전화번호
- 상품명, 상품 설명, 가격, 재고량, 카테고리(의류, 신발, 가전제품 등), 상품 등록 시간, 누가 상품을 등록했는 지

- 특정 장바구니에 포함된 상품들, 장바구니에 포함된 상품의 각 개수, 장바구니의 주인이 누구 지
- 주문 시 배송 정보 (이름, 주소, 전화번호), 주문한 상품 및 수량, 누가 주문했는지
- 어떤 상품에 리뷰를 달았는지, 리뷰 내용, 평점, 누가 리뷰를 달았는지
- 관리자용 이메일, 비밀번호

## 2. 그룹핑해서 분류하기

위의 요구사항을 봤을 때 여러 데이터를 묶어서 표현할 수 있는 상위 개념은 아래와 같다.

- 이메일, 비밀번호, 이름, 주소, 전화번호 → **사용자**
- 상품명, 상품 설명, 가격, 재고량, 카테고리(의류, 신발, 가전제품 등), 상품 등록 시간, 누가 상품을 등록했는지 → **상품**
- 특정 장바구니에 포함된 상품들, 장바구니에 포함된 상품의 각 개수, 장바구니의 주인이 누구 지 → **장바구니**
- 주문 시 배송 정보 (이름, 주소, 전화번호), 주문한 상품 및 수량, 누가 주문했는지 → **주문**
- 어떤 상품에 리뷰를 달았는지, 리뷰 내용, 평점, 누가 리뷰를 달았는지 → **리뷰**
- 관리자용 이메일, 비밀번호 → **관리자**