Agenda :-

i) Requirements

ii) Entities

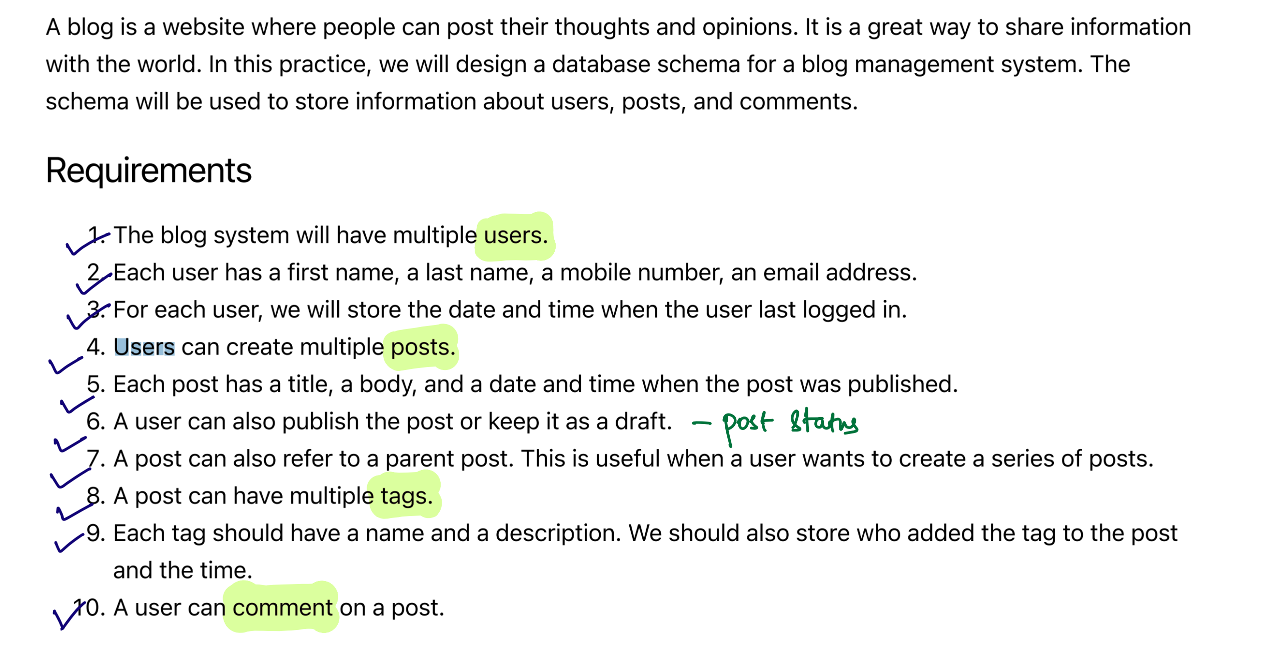
iii) Schema Design

iv) New features

=> BLOGS → Blogspot, Medium, HackerNoon, Dev.to - - -

↓

{ Blog Management System }



A blog is a website where people can post their thoughts and opinions. It is a great way to share information with the world. In this practice, we will design a database schema for a blog management system. The schema will be used to store information about users, posts, and comments.

Requirements

✓ 1. The blog system will have multiple users.

✓ 2. Each user has a first name, a last name, a mobile number, an email address.

✓ 3. For each user, we will store the date and time when the user last logged in.

✓ 4. Users can create multiple posts.

✓ 5. Each post has a title, a body, and a date and time when the post was published.

✓ 6. A user can also publish the post or keep it as a draft. — post Status

✓ 7. A post can also refer to a parent post. This is useful when a user wants to create a series of posts.

✓ 8. A post can have multiple tags.

✓ 9. Each tag should have a name and a description. We should also store who added the tag to the post and the time.

✓ 10. A user can comment on a post.

⇒ User [______________________]

Post [______________________]

– [______________________]

–

relationships -

U–P ⇒ —

P–U - —

[9:26 — 9:38]

↓

time for schema design

→ **USERS**

- id
- firstName
- lastName
- phoneNo.
- emailAddress
- lastActive

→ **POSTS :-**

- id
- title
- body
- publishedAt
- status ← String / boolean (0/1)
- createdAt
- user
- parent post

→ **TAGS**
- id
- name
- description
- addedAt
- addedBy

store the info when a tag was created & by whom

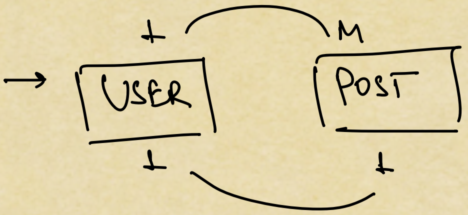
- list<Tag>
- list<comments>
- no of likes

store datetime when it was added to post

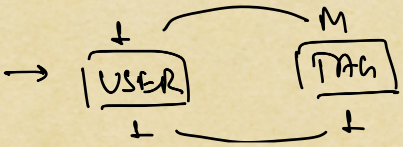
→ **COMMENTS**
- id
- body
- user
- post
- no of likes

⇒ CARDIANALITIES :-

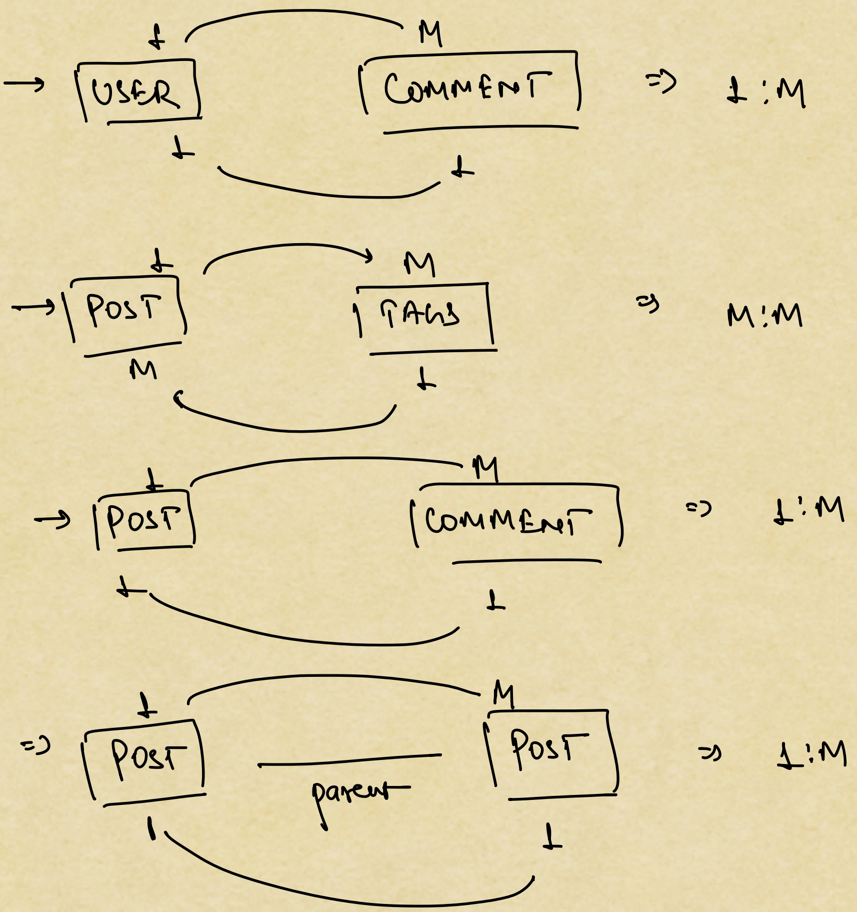
1 : 1	1 : M - M : 1	M : M

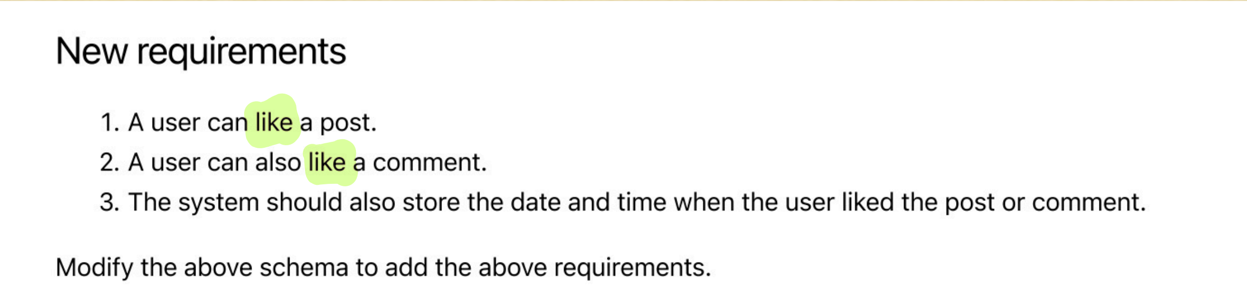


⇒ 1 : M



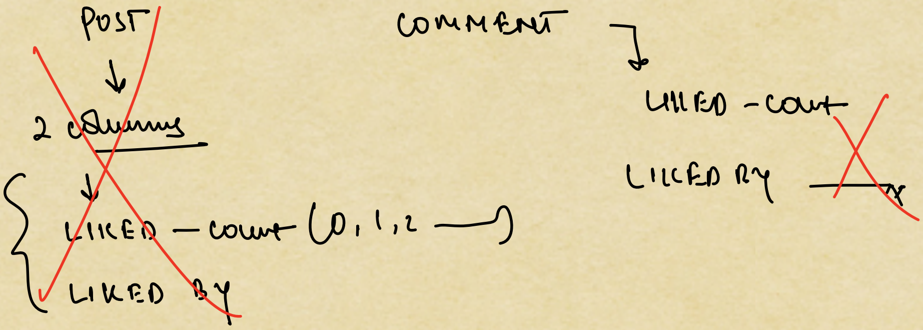
⇒ 1 : M




New requirements

1. A user can like a post.
2. A user can also like a comment.
3. The system should also store the date and time when the user liked the post or comment.

Modify the above schema to add the above requirements.



LIKES

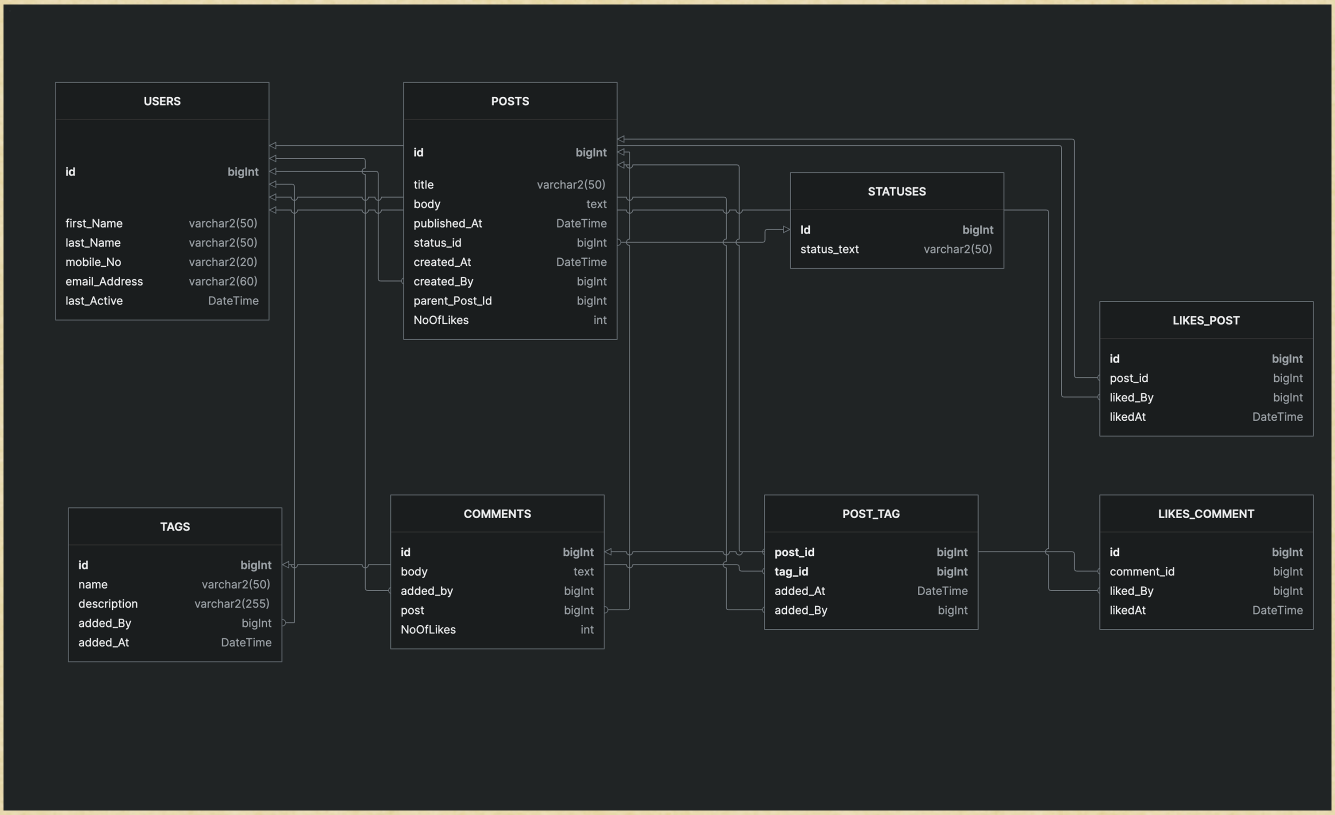
id	likedBy	postId	commentId	datetime
1	U1	null	C1	X
2	U1	P1	null	X
3	U2	P1	null	X
4	U2	null	C2	X

LIKES_POST

id	likedby	postId	likedAt

LIKES_COMMENT

id	likedby	commentId	likedAt



USERS	
id	**bigInt**
first_Name	varchar2(50)
last_Name	varchar2(50)
mobile_No	varchar2(20)
email_Address	varchar2(60)
last_Active	DateTime

POSTS	
id	**bigInt**
title	varchar2(50)
body	text
published_At	DateTime
status_id	bigInt
created_At	DateTime
created_By	bigInt
parent_Post_Id	bigInt
NoOfLikes	int

STATUSES	
Id	**bigInt**
status_text	varchar2(50)

LIKES_POST	
id	**bigInt**
post_id	bigInt
liked_By	bigInt
likedAt	DateTime

TAGS	
id	**bigInt**
name	varchar2(50)
description	varchar2(255)
added_By	bigInt
added_At	DateTime

COMMENTS	
id	**bigInt**
body	text
added_by	bigInt
post	bigInt
NoOfLikes	int

POST_TAG	
post_id	**bigInt**
tag_id	**bigInt**
added_At	DateTime
added_By	bigInt

LIKES_COMMENT	
id	**bigInt**
comment_id	bigInt
liked_By	bigInt
likedAt	DateTime