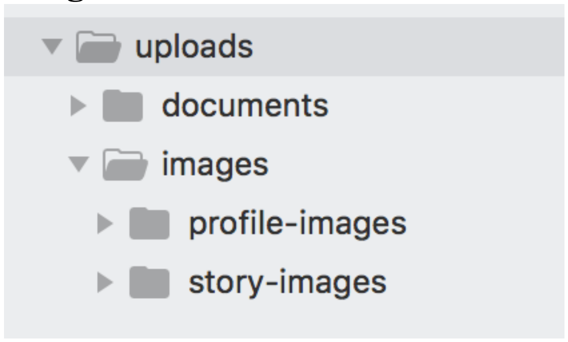
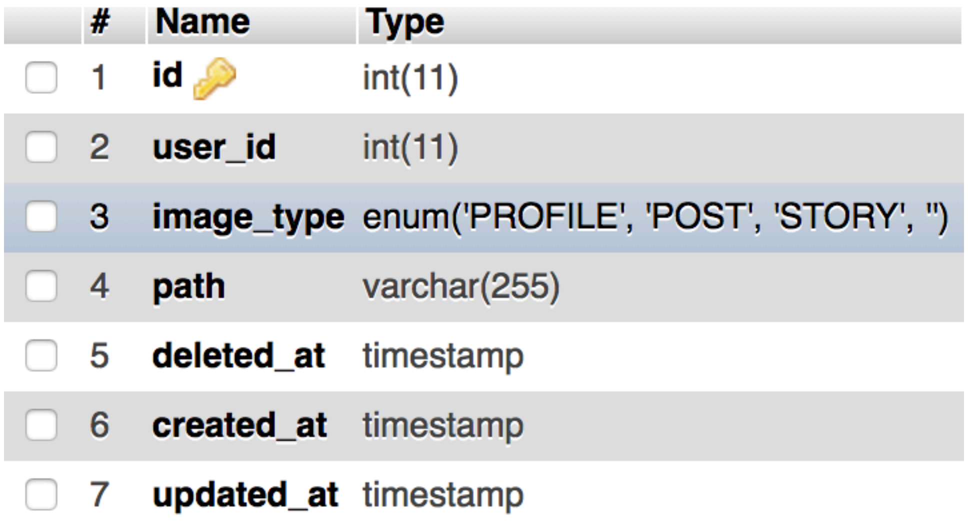
Problem 1 (Theoretical):

A. Given that you have only 32MB of RAM available and the average profile image size is 100KB, therefore all the images cannot fit in the RAM. ***What is the best way you can think of to store these images on the hard disk drive?*** (Assume that unlike RAM, hard drive has no space limitation).

- Initially I would not store any image at ram if not needed.
- While uploading the image generally convention is to append unix timestamp of the time of upload to the start of the filename of image, this is performed after the file is verified to be image file (extension check).(Unix timestamp is appended to prevent duplicate file name uploads.)
- <u>Hard disk setup:</u> I will create a separate folder called **uploads** and then go to uploads directory and create a separate folder called **images**. Under images there will be another folder called **profile-images**.



- This structure enables me to add further relatable files. Suppose tomorrow I have to upload doc files. Then I will create a new folder call **documents** under uploads directory.
- I will create a table/collection in the database, let's assume the name of the table/collection is **images** and whenever we upload profile image I would store the image under the path **/uploads/images/profile-images/** and rename the file to **unixTimeStampFilename** and take the save this path/newFileName in the DB under the images table.



	#	Name	Type
☐	1	id	int(11)
☐	2	user_id	int(11)
☐	3	image_type	enum('PROFILE', 'POST', 'STORY', '')
☐	4	path	varchar(255)
☐	5	deleted_at	timestamp
☐	6	created_at	timestamp
☐	7	updated_at	timestamp

- Whenever the image needs to be displayed with respect to the user, will get the relative path from the db and give the image to the front end.
- When a user is viewing a page, all the path of images, which needs to be there, will be sent and we can make use of lazy loading in the front end.
- Example: https://unsplash.com/. I came across this website 2 years ago which provides license free images which we are free to use. I noticed how the images load as you go down and if you don't go down at all then these images are never loaded.
- Using **lazy loading** I think we can decrease the ram load.
- Compressing the image is also one other way to decrease the load on ram and this also means less storage the image requires. Example: WhatsApp. But this may result in image quality loss as well, which, for me is a total no no as I prefer good quality images over compressed ones.

Problem 2 (Theoretical):

Suppose each image has the same name as the personid ,i.e.,if person id is *23321123* the name of the profile image will be *23321123.jpg*. ***What is the best way to get this image from the hard disk?*** Assume ids and image names are unique and ids are 8-digit numbers, and the platform takes 4 byte to store the integers.

- If image name is same person id and assuming all the images is of the same extension then we don't need to store the image name information anywhere.
- Will create a directory and within this directory I'll store all the images. Since person id is primary key, duplication of image name won't happen.
- Since the path to the directory is constant i.e PATH. Whenever we require a profile image for that particular person, we will just search in PATH directory for the file name with personID.jpg. In this way there won't be searching involved.
- Due to this mechanism you will be generating hard coded image path dynamically.