Course Title: Data Modeling and Normalization with MySQL

Date: 4 August 2020
Topic: Database design and Normalization
Assignment 2 (15%)
Mark: 15

Name – Anu Kulshrestha

Database Design: 'UpClick' - a social photo sharing app for photographers.
Requirements:

1. Users of this system must be registered and should have a unique handle (username). This app is not planning to allow any non-registered users to upload their photo.

2. A voting (Like/Unlike) system where any registered user can like or dislike an uploaded photo.

3. The app should maintain a list of photo categories (e.g., Landscape, Wildlife, Aerial, Sports. Portrait etc..) along with their details.

4. Each photograph can be attached to only one photo category but one category can have many photos associated with it.

5. An user (an photographer) can follow or followed by any number of other users (Photographers)

6. Each user should have their own profile where they can have their facebook. Instagram account link, or any external link they want to share.

** if you want more clarification about the requirements, please ask in piazza forum, can we discuss it there.

Question 1: Identify entities along with their attributes for the above described system? (5)

Note: please be brief as much as possible. The main goal of this question is to determine how critically can think when designing a database.

I found 10 entities required for this website to work.

Photo_category table – This table holds all category names of photo e.g., Landscape, Wildlife, Aerial, Sports. Portrait etc.

Attribute Name	**Attribute data type**	**Mandatory Yes/No**
Photo_category_id (Primary key)	Int	Not Null
Photo_category_name	Varchar	Not Null
Photo_category_added_datetime	DATETIME	Not null
Photo_category_active	TINYINT	Not null

Course Title: Data Modeling and Normalization with MySQL

Date: 4 August 2020

Topic: Database design and Normalization

Assignment 2 (15%)

Mark: 15

Profile_link table – This table holds link type names of photo e.g., facebook. Instagram account link, or any external link

Attribute Name	**Attribute Data Type**	**Mandatory Yes/No**
Profile_link_id (Primary Key)	Int	Not Null
Profile_link_name	Varchar	Not Null
Profile_link_added_datetime	DATETIME	Not null
Profile_link_active	TINYINT	Not null

User_type table – This table holds user types in the system e.g., normal user, user admin

Attribute name	**Attribute Data Type**	**Mandatory Yes/No**
User_type_id (Primary Key)	Int	Not null
User_type_name	Varchar	Not null
User_type_added_datetime	DATETIME	Not null
User_type_active	TINYINT	Not null

User_details table – This table holds the user details of the user

Attribute Name	**Attribute Data Type**	**Mandatory Yes/No**
User_id (Primary Key)	Int	Not Null
User_type_id (Foreign Key)	Int	Not Null
User_first_name	Varchar	Not Null
User_last_name	Varchar	Not Null
User_email_id	Varchar	Not null
User_mobile_number	Int	Not null
User_password	Varchar	Not Null
User_added_datetime	DATETIME	Not null
User_active	TINYINT	Not null

User_profile table – this table holds profile pic and address details if user wants to.

Attribute Name	**Attribute Data Type**	**Mandatory Yes/No**
User_profile_id (Primary Key)	Int	Not null
User_id (Foreign Key)	Int	Not null
User_picture	BLOB	
User_address	Varchar	
User_profile_added_datetime	DATETIME	Not null
User_profile_active	TINYINT	Not null

User_log table – this table holds login details of the user details

Attribute Name	**Attribute Data Type**	**Mandatory Yes/No**
User_log_id (Primary Key)	Int	Not null
User_id (Foreign Key)	Int	Not null
User_login_date	DATE	Not null
User_login_time	Time	Not null

Course Title: Data Modeling and Normalization with MySQL

Date: 4 August 2020

Topic: Database design and Normalization

Assignment 2 (15%)

Mark: 15

User_link table – this table holds user links added by user to his profile.

Attribute Name	Attribute Data Type	Mandatory Yes/No
User_link_id (Primary Key)	Int	Not Null
User_id (Foreign key)	Int	Not Null
Profile_link_id (Foreign Key)	Int	Not Null
User_link_details	Varchar	Not null
User_link_added_datetime	DATETIME	Not null
User_link_active	TINYINT	Not null

User_photo table – this table holds photos details added by the user

Attribute Name	Attribute Data Type	Mandatory Yes/No
Photo_id (Primary Key)	Int	Not Null
User_id (Foreign Key)	Int	Not Null
Photo_category_id (Foreign key)	Int	Not Null
Photo_name	Varchar	Not Null
Photo_image_original	BLOB	Not null
Photo_image_small	LONGBLOB	Not null
Photo_added_day	Int	Not Null
Photo_added_month	Int	Not Null
Photo_added_year	Int	Not Null
Photo_added_hour	Int	Not null
Photo_added_minute	Int	Not null
Photo_added_second	Int	Not Null
Photo_active	TINYINT	Not null

Photo_like table – this table holds the user ids that liked the photo ids.

Attribute Name	Attribute Data Type	Mandatory Yes/No
Photo_like_id (Primary key)	Int	Not null
Photo_id (Foreign Key)	Int	Not null
User_id (Foreign Key)	Int	Not null
Photo_like_added_datetime	DATETIME	Not null
Photo_like_active	TINYINT	Not null

User_follow table – this table holds the details of which user id follows which user id.

Attribute Name	Attribute Data Type	Mandatory Yes/No
User_follow_id (Primary Key)	Int	Not Null
Followed_user_id (Foreign Key)	Int	Not Null
FollowedBy_user_id (Foreign Key)	Int	Not Null
Follow_added_datetime	DATETIME	Not null
Follow_active	TINYINT	Not null

Course Title: Data Modeling and Normalization with MySQL

Date: 4 August 2020

Topic: Database design and Normalization

Assignment 2 (15%)

Mark: 15

Question 2: For each of the entities, please discuss why you have picked certain attributes as the primary key? (3)

Entity/Table Name	Primary Key	Reasons to choose them as primary key
Photo_category_table	**Photo_category_id**	1. This will be a unique count for the details it will carry in the table. 2. We can increment easily for new records to push into the table. 3. This could also be not required much engineering for entering into the table.
Profile_link_table	**Profile_link_id**	
User_type_table	**User_type_id**	
User_details_table	**User_id**	
User_profile_table	**User_profile_id**	
User_log_table	**User_log_id**	
User_link_table	**User_link_id**	
Photo_table	**Photo_id**	
Photo_like_table	**Photo_like_id**	
User_follow_table	**User_follow_id**	

Course Title: Data Modeling and Normalization with MySQL

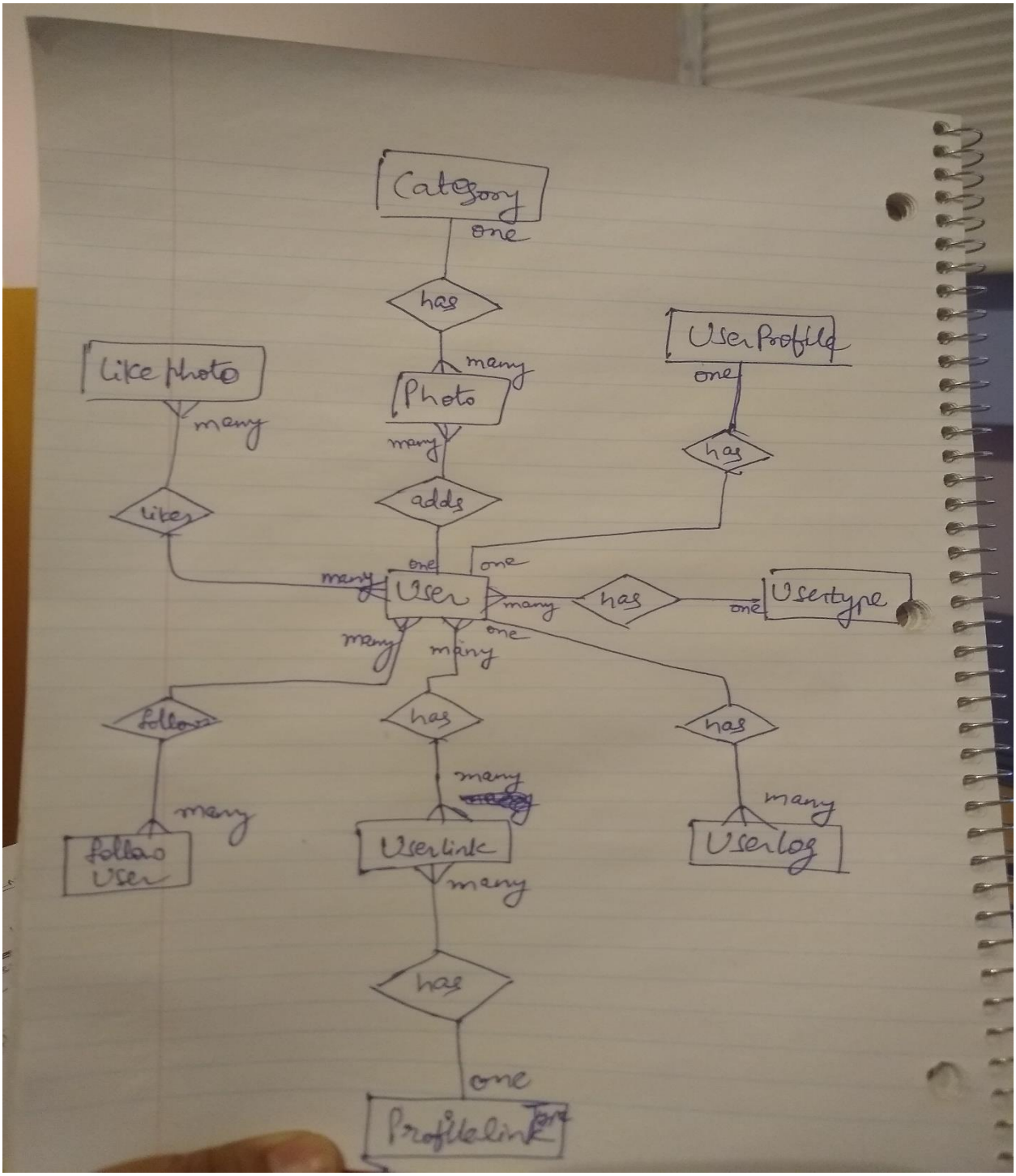
Date: 4 August 2020

Topic: Database design and Normalization

Assignment 2 (15%)

Mark: 15

Question 3: Draw Entity-relationship diagram for the above system (specifying the relationship) (5)



Course Title: Data Modeling and Normalization with MySQL

Date: 4 August 2020
Topic: Database design and Normalization
Assignment 2 (15%)
Mark: 15

Question 4: using an example (two entities), please discuss One-to-One (1on1) relationship? (2)

One-to-one relationship:

One user can have one profile details.

1. after user gets registered, the user can add their picture and contact address to the profile. This will be one for every one user.

So, The **User_details_table** can have one record in **User_profile_table.**

Also, **User_profile_table** can have one user_id from **User_details_table**

Hence both the tables shall have one record that shows relation between both the tables.