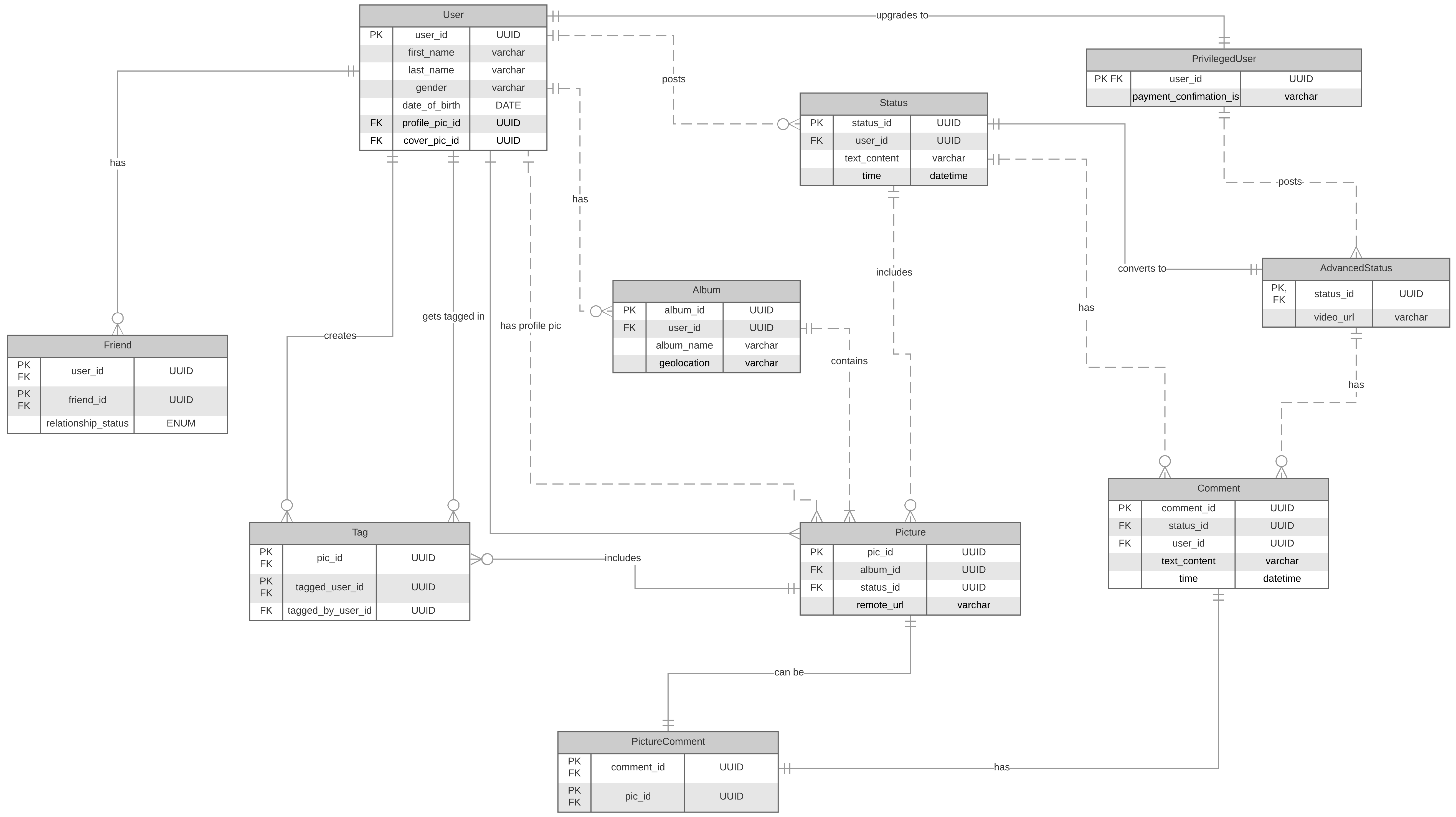
Database Systems CSCI-585
Homework 01
Sbook Report

Submitted by:
Amit Kumar
USC ID: 7088752227

SBook ER Diagram



Components of the ERM

ENTITY (X)	CARDINALITY X	RELATIONSHIP	CONNECTIVITY	ENTITY (Y)	CARDINALITY Y	DESCRIPTION
User	(1, 1)	posts	1:M	Status	(0, M)	A user can have many posts, but a post can only have 1 user.
User	(1, 1)	upgrades to	1:1	PrivilegedUser	(0, 1)	The PrivilegedUser table inherits all the attributes of the User table. Also the Primary key in PrivilegedUser table is the Foreign Key which references the primary key of the User table. So a privileged user can belong to only one user and a user can be only one privileged user.
User	(1, 1)	has	1:M	Album	(0, M)	A user can create many albums, but an album can be owned by a single user. A user can also have 0 albums.
User	(1, 1)	has	1:M	Friend	(0, M)	One user can have many friends.
Status	(1, 1)	has	1:M	Comment	(0, M)	A status can have 0 or more comments, but a comment can have only one user. A comment cannot exist without a user.
Status	(1, 1)	includes	1:M	Picture	(0, M)	A status can have 0 or more pictures in it. A picture will always be associated with only one status because an image will always be added to a user account via a status update or a comment on a status.
Status	(1, 1)	converts to	1:1	AdvancedStatus	(0, 1)	The AdvancedStatus inherits all attributes from the Status table. The primary key of the AdvancedStatus table references the primary key of the Status table. So basically primary keys in both the tables are same. Thus a status can be a advanced status but an AdvancedStatus has to be a Status.
Picture	(1, 1)	Includes	1:M	Tag	(0, M)	A picture can contain 0 or many tags. But a tag will always have only one picture associated with it. A tag cannot appear in more than one picture.
Picture	(1, M)	contained in	M:1	Album	(1, 1)	A picture will always belong to one and only one album. An album can contain 1 and more pictures in it.
Picture	(1, 1)	can be	1:1	PictureComment	(0, 1)	A picture can be posted as a comment on a status and that picture can be part of only one comment. So a picture can or cannot be a considered as a single comment.
User	(1, 1)	creates	1:M	Tag	(0, M)	A user can create a tag to tag his/her friends and a tag can only be created by a single user.
User	(1, 1)	gets tagged in	1:M	Tag	(0, M)	A user can be tagged in multiple tags(multiple pictures), but a tag can have only one user for whom the tag was created.
User	(1, 1)	has profile pic	1:1	Picture	(0, 1)	A user can have no picture or only one picture as a profile picture. Two users cannot have same profile picture.
User	(1, 1)	has cover pic	1:1	Picture	(0, 1)	A user can have no picture or only one picture as a cover picture. Two users cannot have same cover picture.

Assumptions:

1. Every picture will be uploaded to profile via a status update like as we do in Facebook. So even the profile picture and cover picture update will be considered as a status update.
2. A user can have different relationship status with other users like “Following”, “Pending”, “Blocked”, “Friends”.
3. Users will make more textual comments as compared to the image comments.

Schema Definitions:

1. User

Attribute Name	Data Type	Constraints
user_id	UUID	Primary Key, Not Null
first_name	varchar	Not Null
last_name	varchar	Not Null
gender	varchar	Not Null
date_of_birth	DATE	Not Null
profile_pic_id	UUID	
cover_pic_id	UUID	

Description:
1. User table will have the profile information for a user.
2. This entity is the parent entity for the PrivilegedUser entity

2. Privileged User

Attribute Name	Data Type	Constraints
user_id	UUID	Primary Key Not Null, Foreign Key References user_id in User table
payment_confirmation_id	UUID	NOT NULL

Description:
1. This entity is derived from the User entity.
2. The user_id attribute is totally derived from the User entity.
3. This is a weak entity and its existence is dependent on the User entity.

3. Status

Attribute Name	Data Type	Constraints
status_id	UUID	Primary Key Not Null
user_id	UUID	Not NULL, Foreign Key References user_id in User table
text	varchar	
time	datetime	Not NULL

Description:
1. This entity is parent of the AdvancedStatus entity.
2. The “text” field is an optional field because a status update can be just a simple picture upload without any content.

4. Picture

Attribute Name	Data Type	Constraints
pic_id	UUID	Primary Key Not Null
album_id	UUID	NOT NULL, Foreign Key references album_id in Album table
status_id	UUID	NOT NULL, Foreign Key references status_id in Status table
remote_url	varchar	NOT NULL

Description:

1. A picture will always be part of a status, either by comment or by regular status post. So status_id must be present in the picture entity.
2. The url cannot be null because it is the most important attribute of the entity.

5. Comment

Attribute Name	Data Type	Constraints
comment_id	UUID	Primary Key Not Null
status_id	UUID	NOT NULL, Foreign Key references status_id in Status table
user_id	UUID	NOT NULL, Foreign Key references user_id in User table
text_content	varchar	
time	datetime	NOT NULL

Description:

1. This entity is having "text_content" attribute as optional field because a comment can also have only image without text. This field will be null in these cases. Since these cases are going to happen very rare as people mostly comment in text. So we can have few rows with null values.
2. The other option is to create a separate entity named "TextComments". But then we have to do join of three table (Comment, TextComment and PictureComment) to get all comments for a status, will have performance issue.

6. Album

Attribute Name	Data Type	Constraints
album_id	UUID	Primary Key Not Null
user_id	UUID	NOT NULL, Foreign Key references user_id in User table
album_name	varchar	NOT NULL
geolocation	varchar	

Description:

1. This entity can have geolocation as an optional field because user might not chose to enter the geo location of the album.

7. Tag

Attribute Name	Data Type	Constraints
pic_id	UUID	Primary Key Not Null, Foreign Key references image_id in Image table
tagged_user_id	UUID	Primary Key Not Null, Foreign Key references user_id in User table
tagged_by_user_id	UUID	Not Null, Foreign Key references user_id in User table

Description:
1. This entity is a weak entity as its primary key is a composite key derived from Picture entity and User entity.
2. This entity has a strong relationship with User table and Picture table.

8. PictureComment

Attribute Name	Data Type	Constraints
comment_id	UUID	Primary Key Not Null, Foreign Key references comment_id in Comment table
pic_id	UUID	Primary Key Not Null, Foreign Key references pic_id in Picture table

Description:
1. This entity is weak entity as its primary keys are derived from the primary keys of Comment entity and Picture entity.

9. Friend

Attribute Name	Data Type	Constraints
user_id	UUID	Primary Key Not Null, Foreign Key references user_id in User table
friend_id	UUID	Primary Key Not Null, Foreign Key references user_id in User table
relationship_status	enum(‘Pending’, ‘Accepted’, ‘Following’, ‘Blocked’)	NOT NULL

Description:
1. This entity is a weak entity as its primary key is composed of primary keys from User entity.

10. AdvancedStatus

Attribute Name	Data Type	Constraints
status_id	UUID	Primary Key Not Null, Foreign Key references status_id in Status table
video_url	varchar	NOT NULL

Description:
1. This is a weak entity as its primary key is totally derived from the primary key of the Status entity.

Design Choices:

1. We can add indexes to all the primary keys so that the lookup operation can be made fast.

Other Possible Designs:

1. I have tried to break down the entities in smaller entities so as to prevent null values in the rows. For example, I have extracted out the picture attribute from the comment entity because the users will not always comment with a picture. So this design prevents the null values in the rows where no picture was used as a comment. But this reduces the performance of the query as now to get all the comments of a post we need to join two table (Comments and PictureComments). But that we can overlook as we are saving lot of space at very minimal sacrifice in performance.