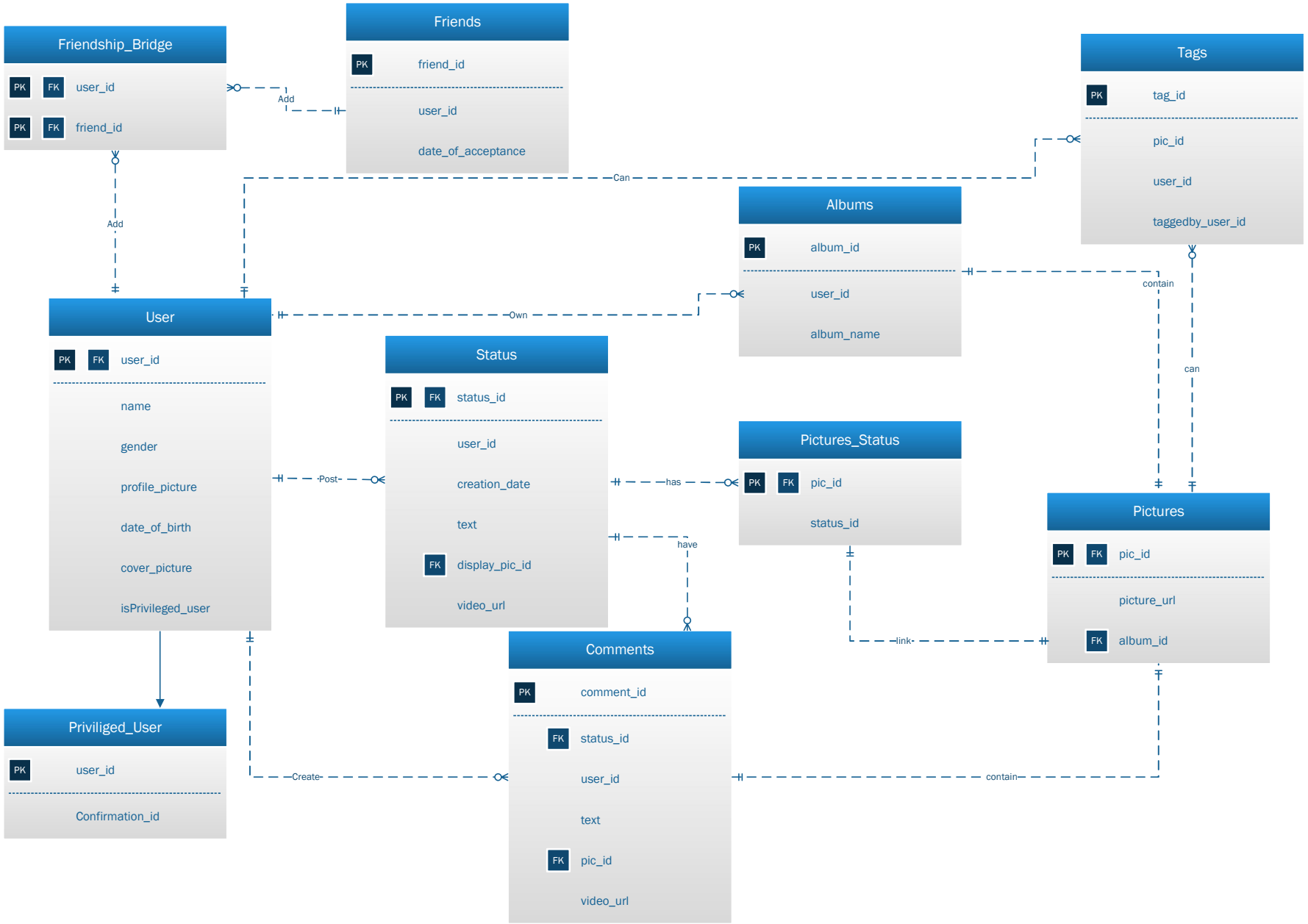
ER DIAGRAM


Friendship_Bridge
PK FK user_id
PK FK friend_id
Add
Friends
PK friend_id
user_id
date_of_acceptance
Tags
PK tag_id
pic_id
user_id
taggedby_user_id
Can
Add
Albums
PK album_id
user_id
album_name
contain
can
User
PK FK user_id
name
gender
profile_picture
date_of_birth
cover_picture
isPrivileged_user
Own
Status
PK FK status_id
user_id
creation_date
text
FK display_pic_id
video_url
Post
has
Pictures_Status
PK FK pic_id
status_id
have
Pictures
PK FK pic_id
picture_url
FK album_id
link
Comments
PK comment_id
FK status_id
user_id
text
FK pic_id
video_url
Create
contain
Priviliged_User
PK user_id
Confirmation_id

REPORT

ENTITIES: -

USER

Attributes: user_id, name, gender, profile_picture, date of birth, cover_picture, isprivilegedUser, password, username, about me, email

- The user entity is connected to multiple entities.
- It is connected to other entities through user_id foreign key.
- All the user profile information can be stored in this entity.
- I have not displayed all in the ER diagram to make it simple.
- Above are some extra entities which can be added to a user.
- isprivilegedUser attribute is used to keep track of privilege users. If it is set to 1 then the user is privilege user else he is a regular user.

Table Example:

User_id	Name	Gender	ProfilePicture	Username	Password	isprivilegedUser
344	James	Male	Pictures/profilepic	james	James12	0
678	Betty	Female	Bettyalbum/pics	betty	Betty56	1

PRIVILEGED_USER

Attributes: user_id, confirmation_id

- Privileged user is inherited from User Entity.
- All Privileged user data is stored in User entity, only the confirmation ID is stored here.

Table Example:

User_id	Confirmation_id
678	47007622

FRIENDSHIP_BRIDGE

Attributes: user_id, frnd_id

- This entity is created to simplify m to n cardinality between users and friends.

Table Example:

User_id	friend_id
UI	F1
U2	F3
U3	F1
U4	F3
U4	F4

FRIENDS

Attributes: friend_id, user_id, date of acceptance

- By combining friendship bridge and friends we can get all the users who are friends with one another.

Table Example:

friend_id	user_id
F1	U4
F2	U2
F3	U1
F4	U3

STATUS

Attributes: status_id, user_id, create_date, text, display_pic_id, video_url

- With an assumption that a status can contain text, pictures and video.
- This entity contains information of all possible combinations that the user can post in a status.
- Before the video_url is added to the status we need to check from user_id if the user is a privileged or not. If the user is privileged then only the video url should be updated.
- Only the displaying picture ID is stored in this entity to get the complete list of pictures we need to check in pictures_status table.

Table Example:

Status_id	User_id	Creation_date	Text	Display_pic_id	Video_url
234	U1	Jan 23rd	Hello	null	null
986	U1	Jan 25th	null	678	null
677	U1	Jan 27th	null	null	Video/myvid.mp4
917	U2	Jan 28th	Hi. How are you?	847	Video/myvid2.mp4

PICTURES_STATUS

Attributes: pic_id, status_id

- This entity contains the status and pictures link i.e. stores the data for which status which picture is linked.

Table Example:

Pic_id	Status_id
678	986
847	917

PICTURES

Attributes: pic_id, picture_url, album_id

- Contains complete information the pictures which are added to the statuses and comments.
- Also contains the album id in which the pictures belong to.
- As mentioned only picture url is stored in our system.

Table Example:

Pic_id	Picture_url	Album_id
678	Pictures/mypic1.pgn	456
874	Pictures/mypic2.pgn	678

COMMENTS

Attributes: comment_id, status_id, user_id, text, pic_id, video_url

- Comments can contain either text, picture or video.
- My assumption is that we can add combination of text, picture or video too.
- A picture with text caption or video with text caption can be added.
- We are limiting 1 picture or 1 video per comment.
- To generalize pictures, we store only the picture id in this entity and connect to pictures entity where complete picture information is present.
- Only privileged user can comment videos.
- Regular users can comment on video status with text or pictures.

Table Example:

Comment_id	Status_id	User_id	Text	Pic_id	Video_url
7684	986	U2	Good Picture	null	null
8768	986	U3	Null	975	null

ALBUM

Attributes: album_id, user_id, album_name

- An album is collection of pictures.
- These pictures can be added to statues or comments.
- This entity also contains the information that who is the owner of these albums.

Table Example:

Album_id	User_id	Album_name
456	U1	Album1
678	U1	Album2

TAGS

Attributes: tag_id, pic_id, user_id, tagged_by_user_id

- Any user can tag people in a picture.
- We can log the information that who is tagging whom on which picture in this entity.

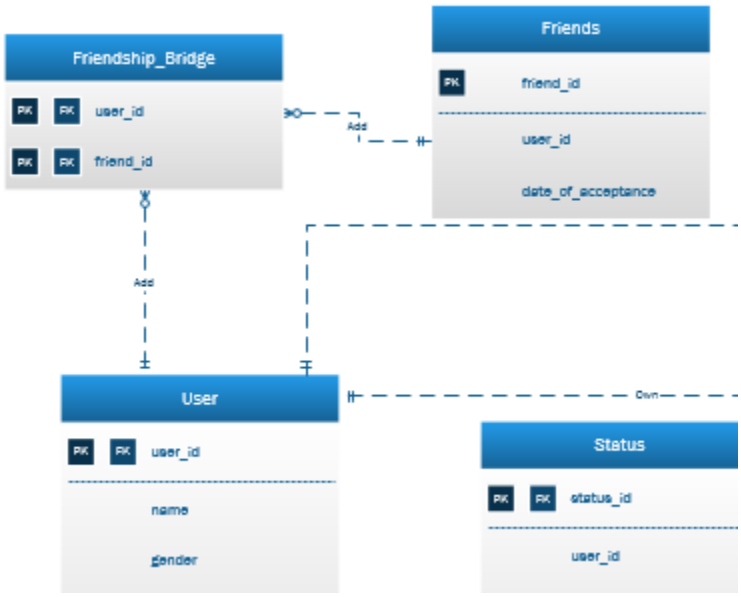
Table Example:

Tag_id	Pic_id	User_id	Tagged_by_user_id
567	546	U2	U3
836	976	U1	U3

Relationships:

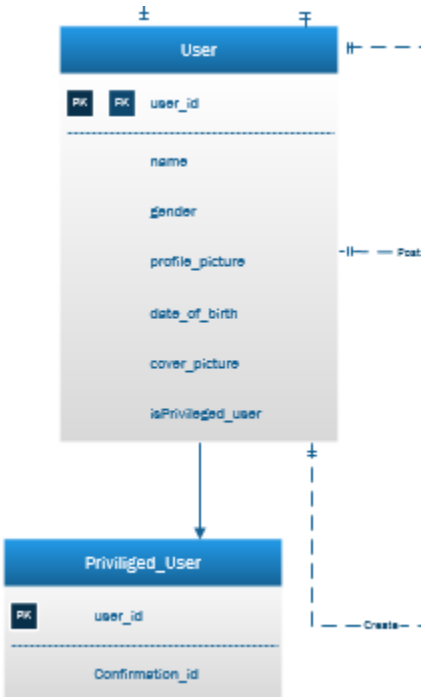
Users – Friends

They have many to many relationship, to solve this relationship to a simpler manner we add bridge entity to our ER diagram.



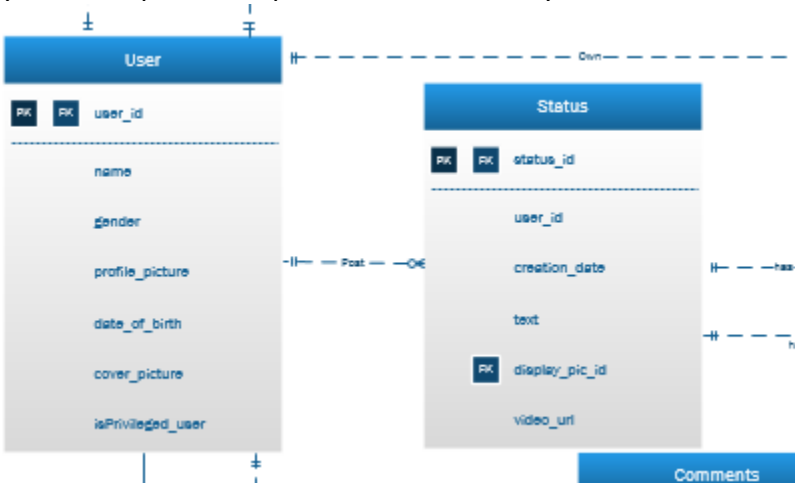
User- Privileged_User

Privileged User inherits the properties from Users. Privileged users have extra confirmation id which needs to be stored in the database hence to accommodate that I have used a separate entity. Privileged user gets the confirmation ID only if he does a one-time payment to the system and the flag isPrivileged_user is set to 1.



User- Status

Every user can post multiple statuses but every status can have only one owner user.

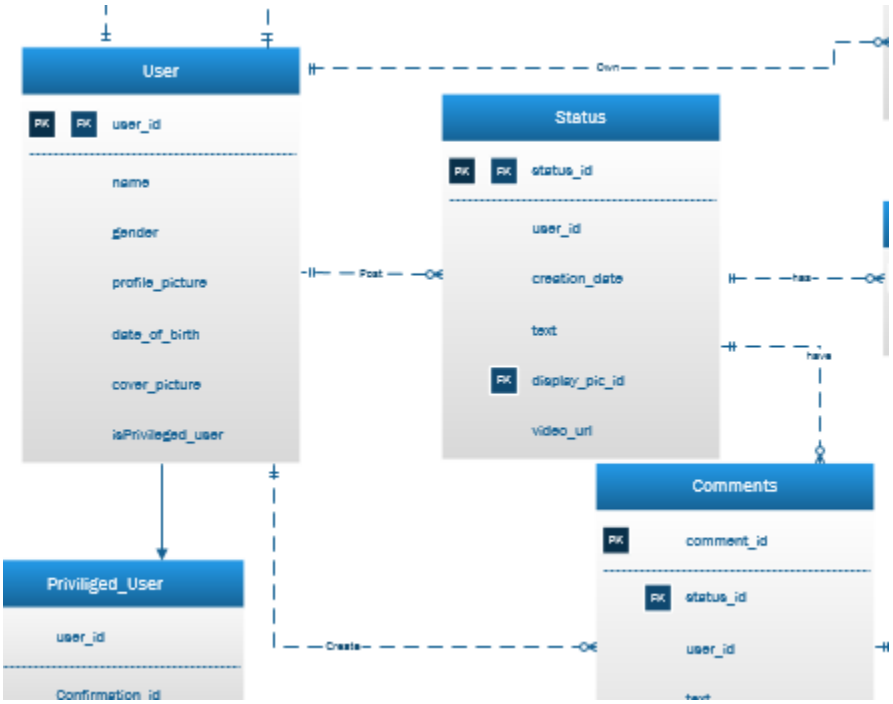


User- Album

Every user can have one or more albums but single album can be owned by only one user.

User- Comments

Every user can do multiple comments on multiple status but each comment can be done by only one user.

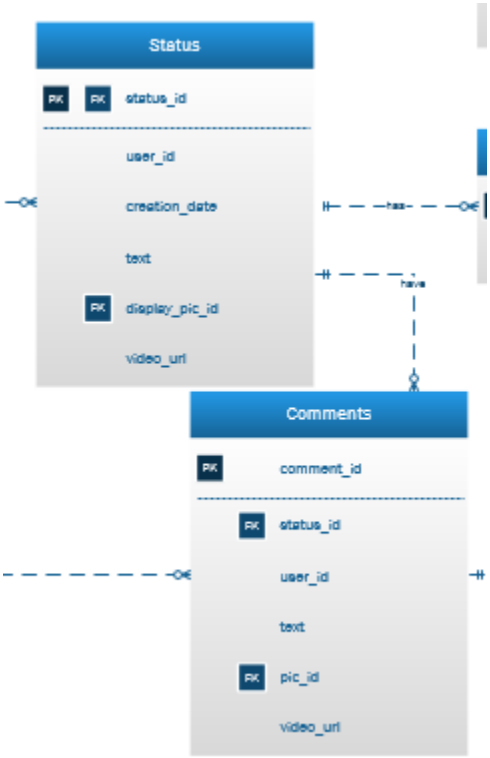


User – Tags

Single user can be tagged in multiple picture, each tag must be linked to only one user.

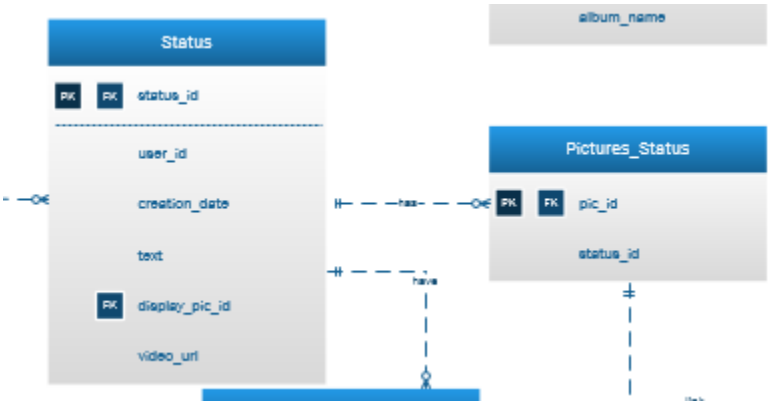
Status- Comment

Every status can have one or more comments but each comment can belong to only one status.

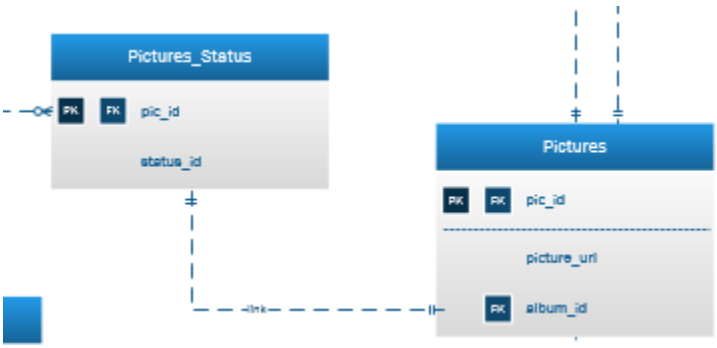


Status- Picture_status

Each status can have multiple pictures but each picture can belong to only one status.
The pictures are tacked through status_id.



Picture_status- Pictures



Pictures – Album

Each picture can belong to only one album. Each album has either one or more pictures.

Pictures – Tags

There can be multiple tags on each picture but each tag can be associated to single picture.
The users tagged can have multiple tags but the tag_id differs.