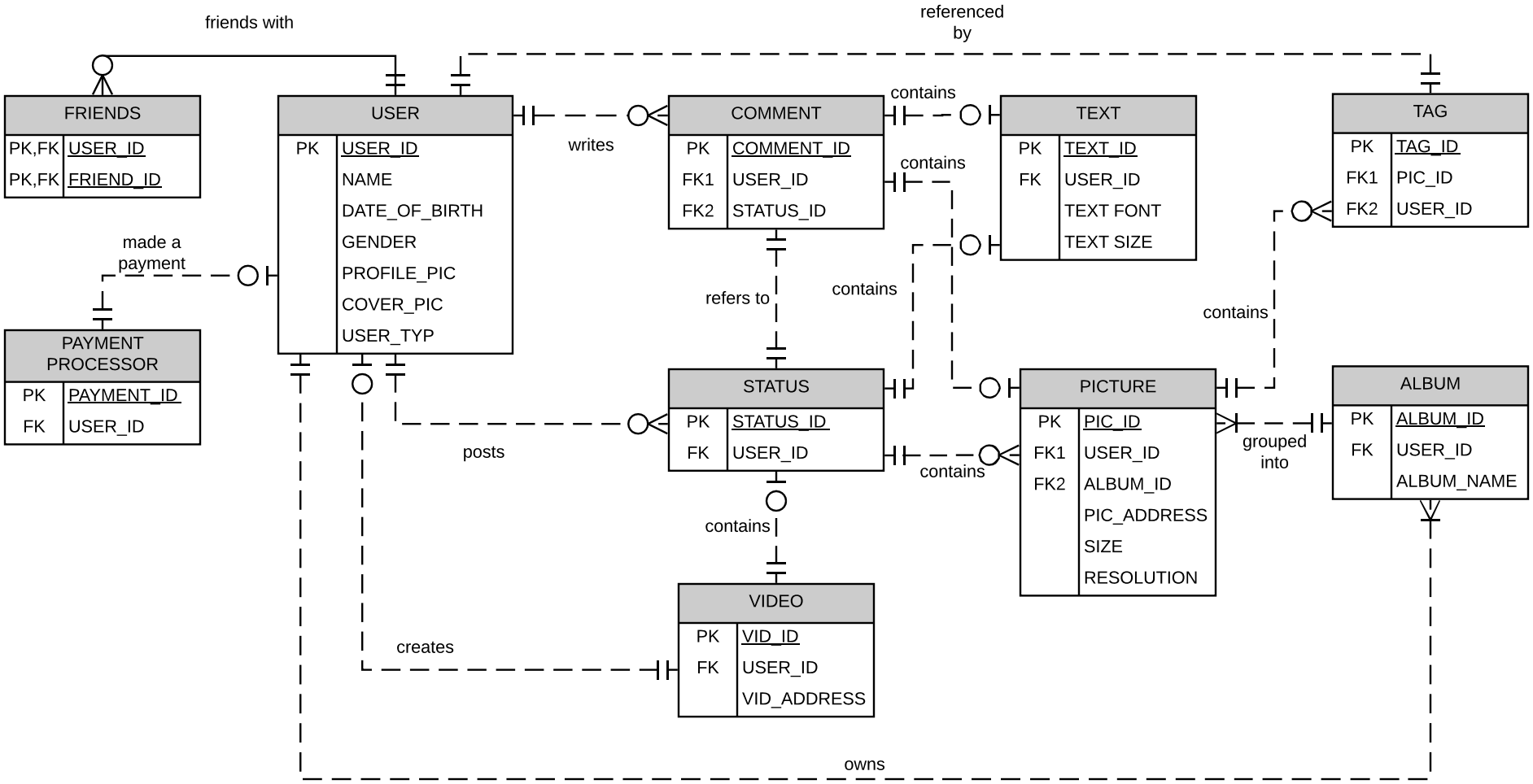
friends with
referenced by
FRIENDS
PK,FK USER_ID
PK,FK FRIEND_ID
USER
PK USER_ID
NAME
DATE_OF_BIRTH
GENDER
PROFILE_PIC
COVER_PIC
USER_TYP
writes
COMMENT
PK COMMENT_ID
FK1 USER_ID
FK2 STATUS_ID
contains
TEXT
PK TEXT_ID
FK USER_ID
TEXT FONT
TEXT SIZE
TAG
PK TAG_ID
FK1 PIC_ID
FK2 USER_ID
made a payment
PAYMENT PROCESSOR
PK PAYMENT_ID
FK USER_ID
refers to
contains
contains
STATUS
PK STATUS_ID
FK USER_ID
posts
contains
PICTURE
PK PIC_ID
FK1 USER_ID
FK2 ALBUM_ID
PIC_ADDRESS
SIZE
RESOLUTION
grouped into
ALBUM
PK ALBUM_ID
FK USER_ID
ALBUM_NAME
contains
VIDEO
PK VID_ID
FK USER_ID
VID_ADDRESS
creates
owns

SBOOK ER DIAGRAM REPORT

The following table shows the relationships between the entities in the ER Diagram:

ENTITY	RELATIONSHIP	CONNECTIVITY	ENTITY
USER	FRIENDS WITH	1 : N	FRIENDS
USER	MADE A PAYMENT	1 : 1	PAYMENT PROCESSOR
USER	POSTS	1 : N	STATUS
USER	WRITES	1 : N	COMMENT
USER	CREATES	1 : 1	VIDEO
USER	OWNS	1 : N	ALBUM
USER	REFERENCED BY	1 : 1	TAG
STATUS	CONTAINS	1 : 1	TEXT
STATUS	CONTAINS	1 : N	PICTURE
STATUS	CONTAINS	1 : 1	VIDEO
COMMENT	CONTAINS	1 : 1	TEXT
COMMENT	CONTAINS	1 : 1	PICTURE
COMMENT	REFERS TO	1 : 1	STATUS
PICTURE	CONTAINS	1 : N	TAG
PICTURE	GROUPED INTO	N : 1	ALBUM

FRIENDSHIPS

A user may have no friends or may be friends with multiple users. (friends with- 1 : N)
A Friends table is maintained with 2 columns : the user himself and his friend(also a user). The combination of the user and his friend together serve as the composite Primary Key.

Example:

USER_ID	FRIEND_ID
User1	User4
User1	User5
User2	User10

UPGRADING FROM REGULAR TO PRIVILEGED USER

The USER has an attribute called USER_TYP. This can be set to '0' for Regular User or '1' for Privileged User.
There is a relation from USER to PAYMENT PROCESSOR called made payment- 1 : 1. This is designed to be an optional relation. The user may or may not make this payment.
In case he does not make a payment, he stays a Regular User and the USER_TYP attribute is set to 0. If he makes a payment, then a PAYMENT_ID is generated in the PAYMENT PROCESSOR and the USER_TYP is set to 1.

POSTING A STATUS:

A User can post zero or many statuses.
Every Status has a unique STATUS_ID and the identification of the User who wrote the status in STATUS_ID.
Status contains a text component(TEXT), multiple pictures or either of them.
Every TEXT has a unique TEXT_ID and every PICTURE has a unique PIC_ID.
If the user is a privileged user, then the status may have a video too and becomes an advanced status. This is depicted by an optional relation between STATUS and VIDEO.
A VIDEO has a unique VID_ID. It would also have the address of the video i.e. VID_ADDRESS.

COMMENTING ON A PICTURE:

A user (regular and privileged) can create zero or many comments.
Every Comment has a unique COMMENT_ID. It stores information about the user who created the comment as well as the status on which it was written.
A comment can have text or a picture or either of them.
A comment made by a user can refer to just one status, depicted by the 1 : 1 relation between comment and status.

TAGGING OF A PICTURE:

A picture can have multiple tags and each tag can refer to one user only (referenced by- 1 : 1).
TAG contains a unique TAG_ID. It also contains information about which picture it belongs to and which user it refers to.

GROUPING INTO ALBUMS:

Pictures can be grouped into Albums.
Each album is owned by a user. A user can have multiple albums belonging to him. (Owns relation 1 : N)

There are no Multivalued Attributes or Self Loops in the ER diagram to facilitate easy querying.