Database Assignment No. 1
ER Diagram for a social network Sbook
Name: Rahul Shreerang Lele USC ID: 2483165273

Introduction

This assignment consists of making an ER diagram for a social network Sbook. Sbook has two types of users regular and privileged users. Every user has a unique profile. The regular users can upgrade to privileged users through onetime payment. Both the users can post status consisting of text and pictures. Pictures are grouped under albums. One user may have one or more albums. A user can tag other users in the pictures. The privileged users have an additional privilege that they can post video as well in a status. The users can make comments on the status of other users. These comments can include text and pictures but no videos. User have friendships, which are bidirectional.

An Entity Relationship Diagram(ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. ER diagram illustrates the logical structure of databases.
The solution ER diagram is presented using Crow's Foot Notation which represents entities as boxes, and relationships as lines between the boxes. Different shapes at the ends of these lines represent cardinality of the relationship.

1. **Entities**
   There are in total 13 entities in the ER diagram. They are as below :

1. User: This entity stores the profile info of every user. This is a supertype entity having subtypes: Regular User and Privileged User.
   Primary Key: USER_ID
   Other attributes: name, gender, Date_of_Birth, profile_pic, cover_pic, user_type
   Assumption: For the user entity, it is assumed that the profile_pic and cover_pic contain pic_id from the picture entity. The name and user_type attributes cannot be null.

2. Regular_User: This is a subtype of user entity. This entity stores the user_id of the regular users.
   Primary Key: USER_ID (inherited from user, therefore foreign key)

3. Privileged_User: This is a subtype of user entity. This entity stores the user_id of the privileged users.
   Primary Key: USER_ID (inherited from users, therefore foreign key)
   Assumption: Here, every privileged user must have a payment_id.

4. Album: A user can have zero or more albums.
   Primary Key: ALBUM_ID
   USER_ID (inherited from users, therefore Foreign Key)
   Other attributes: album_name, album_address

5. Picture: An album can have zero or more pictures.
   Primary Key: PIC_ID
   ALBUM_ID(inherited from album, so Foreign Key)
   Other attributes: pic_address

6. Status: A regular user can post zero or more status. Status can have text and multiple pictures.
   Primary Key: STATUS_ID
   USER_ID (Foreign Key)
   Other: text_content

7. Tag: A user can tag multiple users in a picture.
   Primary Key: TAG_ID
   PIC_ID (Foreign Key)
   USER_ID(Foreign Key)
   Assumption: user_id is the id of the user who is tagged.

8. Comments: A user can make comments on statuses by other users. The comment can contain text and only one picture.
   Primary Key: COMMENT_ID
   USER_ID (Foreign Key)
   PIC_ID (Foreign Key)
   Other attributes: text_content
   Assumption: Comment can contain text and only one picture. Here, the id of the pic in the comment is provided.

9. Advanced_Status: A privileged user can post advanced status. It can contain text, multiple pictures and video as well.
   Primary Key: ADV_ID
   USER_ID (Foreign Key)
   Other attributes: text_content, video_url

10. Status_Picture: This is a bridge entity for status and picture.
    Attributes: PIC_ID (Primary key and Foreign key)
    STATUS_ID (Primary Key and Foreign Key)

11. Adv_status_picture: This is a bridge entity for advance status and picture.
    Attributes: ADV_ID (Primary Key and Foreign Key)
    PIC_ID (Primary key and Foreign key)

12. status_comments: This is a bridge entity for status and comments.
    Attributes: STATUS_ID (Primary Key and Foreign Key)
    COMMENT_ID (Primary Key and Foreign Key)

13. adv_status_comments: This is a bridge entity for advanced status and comments.
    Attributes: ADV_ID (Primary Key and Foreign Key)
    COMMENT_ID (Primary Key and Foreign Key)

2. Relationship

Majority relations are weak relations that is entities are loosely coupled.

One to zero or many:

1. Between user and album: A user has zero or more albums.
2. Between user and pictures: A user can post zero or more pictures.
3. Between album and pictures: An album contains zero or more pictures.
4. Between picture and tags: A picture has zero or more tags.
5. Between regular_user and status: A regular user can post zero or more statuses.
6. Between privileged_user and advanced_status: A privileged user can post zero or more advanced statuses.
7. Between user and comments: A user can make zero or more comments.
8. Between status and status_picture: A status can have zero or more pictures.
9. Between advanced_status and adv_status_picture: An advanced status can have zero or more pictures.
10. Between user and tag: A user has zero or more tags. But a tag belongs to one user only.

One to Zero or One:

1. Between status_comments and comments: A comment may belong to a status or it may not belong to a status, so it is zero or one.
2. Between advanced status and comments: A comment may belong to an advanced status or it may not belong to an advanced status, so it is zero or one.
3. Between comments and pictures: A comment can have zero or one picture.
4. Between pictures and status_picture: A picture may belong to a status or may not belong to a status, so it is zero or one.
5. Between pictures and advanced_status_picture: A picture may belong to an advanced status or may not belong to an advanced status, so it is zero or one.

Many to Many:

1. Between user to user (unary relationship): A user can have many friends and a user can be friend to many other users.

Assumptions for relationships

1. A status or advanced status can have zero or more comments. Every comment has a unique id. Every comment_id in the comment entity can either belong to status or advanced status. So the relation from comments to status and advanced status is one to zero or one.
2. Any single picture cannot be shared by multiple statuses. Any single picture cannot be shared by multiple advanced statuses. Any single picture can either belong to a status or advanced status or comments. So pictures entity has one to zero or one relation with status_picture, advanced_status_picture and comments.

Strong Relationship

1. Between comments and status_comments.
2. Between comments and adv_status_comment.
3. Between status and status_comments
4. Between adv_status_comment and advanced_status
5. Between pictures and status_picture
6. Between status and status_picture.
7. Between pictures and adv_status_picture
8. Between adv_status_picture and advanced_status
9. Between album and pictures.

Trade-offs

1. The bridge entities like status_picture, adv_status_picture, status_comments and adv_status_comments were added in the ER diagram just to simplify the design. Instead of status_picture, the status_id can be given as foreign key in the pictures entity. The same logic applies to other three bridge entities as well.
2. The friendship relation can also be shown as a separate entity.
3. We can also have a separate payment entity to upgrade from regular to privileged user.

ER DIAGRAM FOR SBOOK