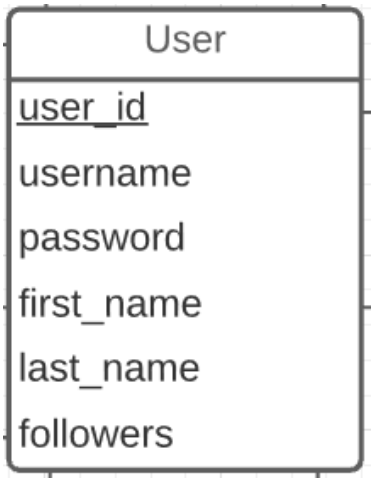
UNIVERSIDAD DE PUERTO RICO
RECINTO UNIVERSITARIO DE MAYAGÜEZ
MAYAGÜEZ, PUERTO RICO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y DE COMPUTADORAS

Documentation:
Mapping from ER to Table

Juan G. Dávila Álvarez
John A. Parks Guzmán
Anthony V. Rivera Torres
Agustin M. Sánchez Cancio
ICOM5016 – CIIC4060
Profesor: Manuel Rodríguez
12 de abril de 2021

Users

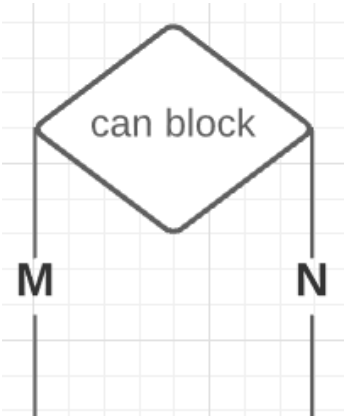


The entity user in the ER Diagram has a primary key calles user_id, it also has the parameters username, password, first_name and last_name which are a combination of characters, lastly followers, which will be an integer with the number of followers the user has.

When mapping this entity, since it only has atomic attributes, we declared each attribute as a parameter in the table creation with their specific data type, the resulting implementation for create table was the next one:

create table users (uid serial primary key, uname varchar(20), upassword varchar(20), fname varchar(20), lname varchar(20), followers int).

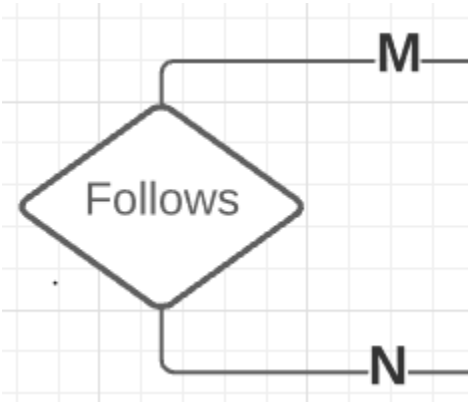
Blocks



In the ER Diagram what will be the table Blocks is a many to many relation with User, this relation is between the same entity, User. When mapping a many to many relation the table needs as attributes the identification of the entities related and a primary key formed by both of the entities identifications, which will be foreign keys. Once implemented, the resulting tables for blocks is:

create table blocks (uid integer references users(uid), buid integer references users(uid), primary key (uid, buid))

Followers



Similar to the table Blocks, the table for Followers in the ER Diagram was a many to many relation, when implemented as a table the attributes that will make up the table are the both user identifiactions and a primary key composed of both foreign keys of the entities related, in this case the both will be user identifications one will represent the user and the other the follower. The table creation will look like:

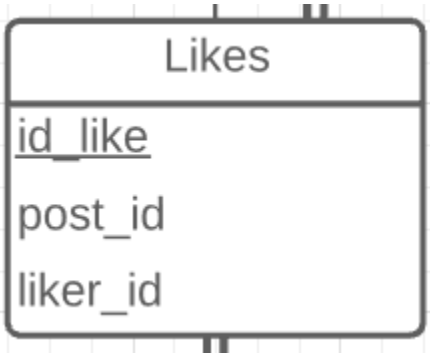
create table followers (uid integer references users(uid), fuid integer references users(uid), primary key (uid, fuid))

Posts

The entity posts in the ER Diagram was composed of several atomic attributes. It has a primary key post_id, a foreign key user_id, and the content which will be a combination of characters. When implementing the table, the attributes will be added according to their data type, the primary key will be declared as such and the foreign key will be declared as a reference of another entities attribute. The date will be implemented with the data type date. The table creation will be:

create table posts (pid serial primary key, uid integer references users(uid), pdate date, content varchar(100))

Likes



The entity Likes will be composed of its primary key id_like and two foreign keys which will be the post_id and the liker_id. The liker_id will be a user identification; it is named liker_id for an easier understanding when implementing the ER Diagram. All the attributes in this entity will be integers, the table for this entity will be:

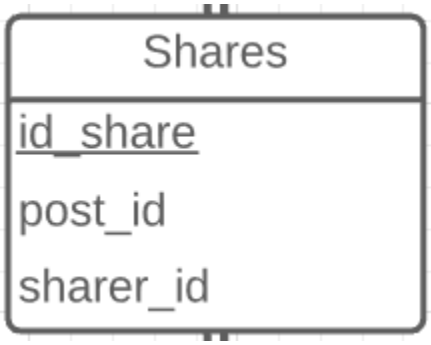
create table likes (pid integer references posts(pid), uid integer references users(uid))

Dislikes

The entity Dislikes will be composed of its primary key id_dislike and two foreign keys which will be the post_id and the disliker_id. The disliker_id will be a user identification; it is named disliker_id for an easier understanding when implementing the ER Diagram. All the attributes in this entity will be integers, the table for this entity will be:

create table dislikes (pid integer references posts(pid), uid integer references users(uid))

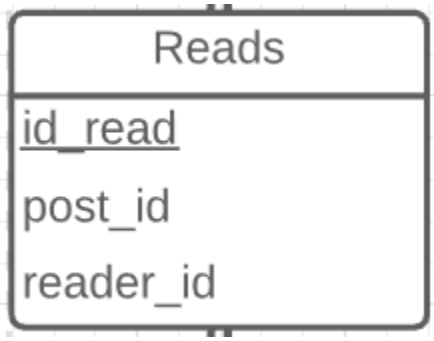
Shares



The entity Shares will be composed of its primary key id_share and two foreign keys which will be the post_id and the sharer_id. The sharer_id will be a user identification; it is named sharer_id for an easier understanding when implementing the ER Diagram. All the attributes in this entity will be integers, the table for this entity will be:

create table shares (pid integer references posts(pid), uid integer references users(uid))

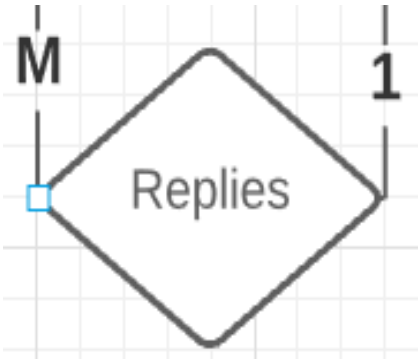
Reads



The entity Reads will be composed of its primary key id_read and two foreign keys which will be the post_id and the reader_id. The reader_id will be a user identification; it is named reader_id for an easier understanding when implementing the ER Diagram. All the attributes in this entity will be integers and the table for this entity will be:

create table reads (pid integer references posts(pid), uid integer references users(uid))

Replies



Replies in the ER Diagram is a One to Many without total participation relationship with the User, which is why it was implemented as a table. A table for this type of relation is composed of two foreign keys of the entities related, which in this case are one post identification for the post that will be replied to, and a user identification, for the user that creates the reply. The primary key for this relationship will be composed of both the foreign keys and it will have another attribute which will be a combination of characters that will be the reply. The table creation would look like:

create table replies (pid integer references posts(pid), uid integer re ferences users(uid), primay key(pid, uid), content varchar(100))