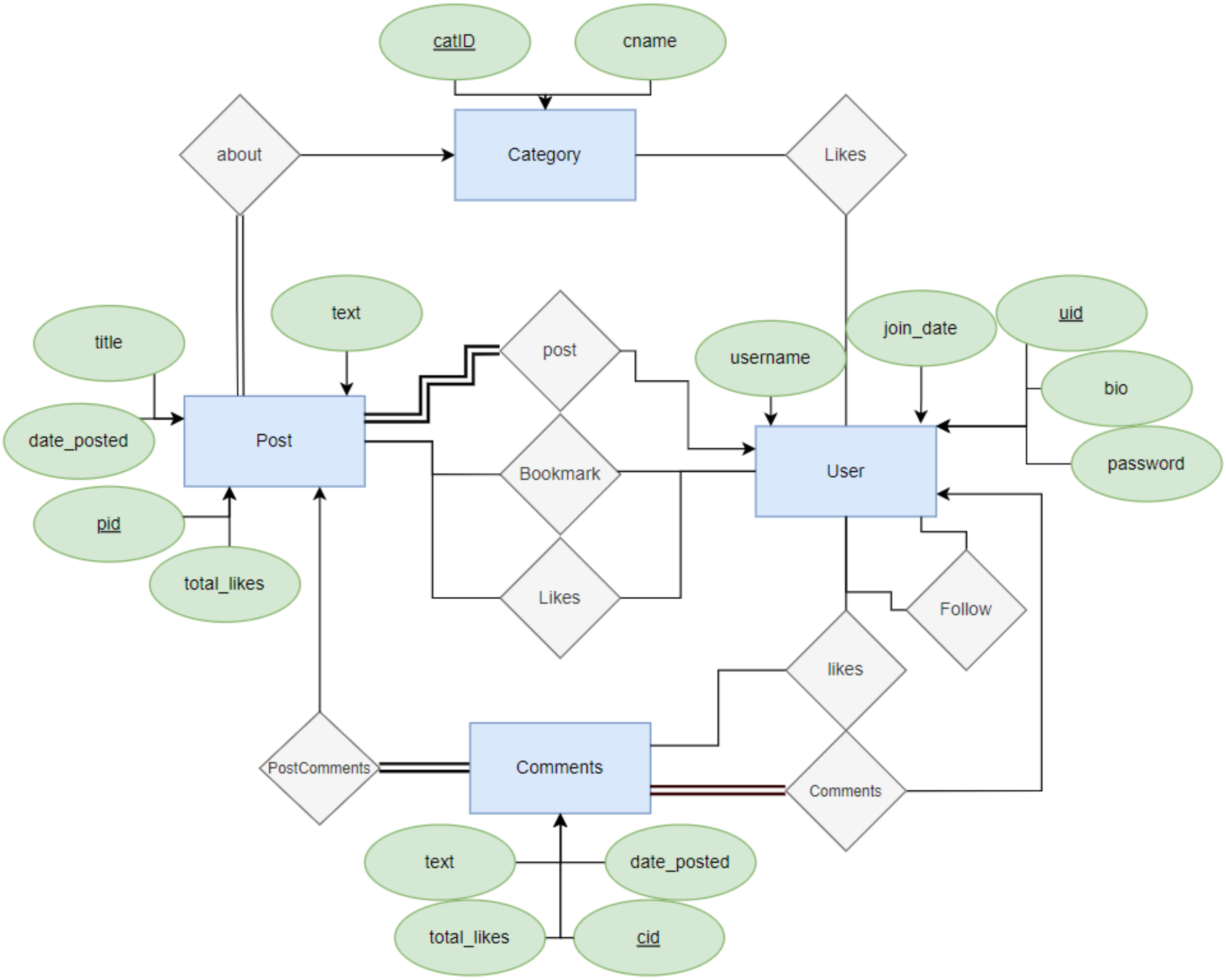
Tech For Dummies - Milestone 1

By: Matthew Reid (mrr7rn) and Dakota Bourne (db2nb)

1)



Previously Mentioned Features:

Bookmarking - Many to Many (i.e. Many Users can bookmark many posts)

Posting - Total Participation Many to One (i.e. All posts are posted by at most one user)

Commenting - Users can interact with comments in multiple ways, meaning an entity was required.

- To Post - Total Participation Many to One (i.e. All comments are on individual posts)
- To User
  - Likes - Many to Many (i.e. Many users can like many comments)

    - Comments - One to Total Participation Many (i.e. One users can comment many times or All comments have to be commented by at most one user)

New Features Created to Meet Table Limit:

Categories for Posts, including users being able to like different categories or setting them as interest

- To Post - One to Total Participation Many (i.e. All posts are required to have at most one category)
- To User - Many to Many (i.e. Many users can like many categories)

Liking Posts and Liking Comments

- Posts - Many to Many (i.e. Many users can like many posts)
- Comments - Many to Many (i.e. Many users can like many comments)

Following others users - Many to Many (i.e. Many users can follow many users)

2)

Tables

User(uid, username, password, join_date, bio)

Post(pid, uid, date_posted, catID, title, text, total_likes)

Comments(cid, pid, cdate_posted, text, total_likes)

Category(catID, cname)

UsersComments(uid, cid)

CatLikes(uid, catID)

UserLikes(uid, pid)

CommentLikes(uid, cid)

Bookmarks(uid, pid)

FollowedUsers(uid, uid)

3)

A->uid[user ID]

B->username

C->password

D->join_date

E->bio

F->pid[post ID]

G->date_posted

H->title

I->text

J->total_likes

K->catID

L->cid [comment ID]

M->text
N->cdate_posted
O->total_likes
P-> cname

Tables With FDs:
User(A, B, C, D, E)
A->B, C, D, E
B->A (username implies uid)
Post(F, A, G, K, H, I, J)
F-> A, G, K, H, I, J
AG->F (user id and when the post was posted implies post id)
Comments(L, F, N, M, O)
L-> F, N, M, O
Category(K, P)
K-> P
P -> K (category name implies category ID (done for ease of maintenance))
UserComments(A, L)
AL-> AL
CatLikes(A, K)
AK -> AK
UserLikes(A, F)
AF -> AF
CommentLikes(A, L)
AL -> AL
Bookmarks(A, F)
AF -> AF
FollowedUsers(A, A)
AA -> AA

BCNF Decomposition for all the tables
1. User(A, B, C, D, E)
A->ABCDE sk
B->ABCDE (original B->A) sk
2. Post(F, A, G, K, H, I, J)
F-> AGKHIJF sk
AG -> AGKHIF (original AG->F) sk
3. Comments(L, F, N, M, O)
L-> FNMOL sk
4. Category(K, P)
K-> KP sk
P-> KP sk
5. UserComments(A, L)
AL->AL sk

6. CatLikes(A, K)

AK -> AK sk

7. UserLikes(A, F)

AF -> AF sk

8. CommentLikes(A, L)

AL -> AL sk

9. Bookmarks(A, F)

AF -> AF sk

10. FollowedUsers(A, A)

AA -> AA sk

For all of the tables above, when considering their FDs and decomposing into BCNF, we put each table into F+, and when we did so, all of the relations were super keys, and as such it was not necessary to split the tables any further. All of the tables are in BCNF.

Final Tables

YellowHighlight - Unique
BlueHighlight - Can be Null

User(uid, username, password, join_date, bio)
Post(pid, uid, date_posted, catID, title, text, total_likes)
Comments(cid, pid, cdate_posted, text, total_likes)
Category(catID, cname)
UsersComments(uid, cid)
CatLikes(uid, catID)
UserLikes(uid, pid)
CommentLikes(uid, cid)
Bookmarks(uid, pid)
FollowedUsers(uid, uid)
Everything else is assumed to be not null, as it generates when posted, created, etc.