|  |  |
| --- | --- |
| **INFO 210 — Database Management Systems Assignment – Milestone 2** |  |

**1. ER diagram modification and description**

# Please submit the modified graph before other modification

**2. Relational Schema**

The following relational schema is presented in bold, and the description of its constraints will be presented in [...] directly below it.

**user (uid: INTEGER, email: CHAR, password: CHAR, profile: CHAR)**

**posts (pid: INTEGER, title: CHAR, content: CHAR, datetime: DATETIME, uid: INTEGER)**

[ uid is the foreign key reference to user and uid is not null. If the user is deleted, the corresponding post would be deleted(uid is cascade)]

**comments (cid: INTEGER, content: CHAR, datetime: DATETIME, uid: INTEGER, pid: INTEGER)**

[ Respectively, uid and pid are foreign key reference to user and posts and both are not null. If a post or a user is deleted，respectively, the corresponding comment got deleted(both uid and pid is cascade) ]

**admin (uid: INTEGER, permission\_level: INTEGER)**

**advertisement (id: INTEGER, content: CHAR, uid: INTEGER)**

[ uid is the foreign key reference to admin. (on deleted no action)]

**categories (cat\_id: INTEGER, cat\_name: CHAR, cat\_desc: CHAR)**

**manage (uid: INTEGER, cat\_id: INTEGER)**

[uid and cat\_id are reference to admins and categories respectively. When the admin or category got deleted, the corresponding row in this table is also got deleted(both uid and cid are cascade)]

**subscribe (uid: INTEGER, cat\_id: INTEGER)**

[uid and cid are reference to users and categories respectively. When the user or category got deleted, the corresponding row in this table is also got deleted(both uid and cid are cascade)]

**classify (pid: INTEGER, cat\_id: INTEGER)**

[pid and cat\_id are reference to posts and categories respectively. When the post or category got deleted, the corresponding row in this table is also got deleted(both pid and cat\_id are cascade)]

**groups ( gid: INTEGER, des: CHAR, uid: INTEGER)**

**join ( gid: INTEGER，uid: INTEGER)**

[uid and gidare reference to users and groups respectively. When the user or group got deleted, the corresponding row in this table is also got deleted(both uid and cid are cascade)]

**announcement (gid: INTEGER, aid: INTEGER, title: CHAR, content: CHAR)**

[ aid is the partial key of announcements and gid is the primary key of groups. The two key combines the primary key of announcements. gid is not null. If a announcement is deleted, delete the corresponding row from the group table( ON DELETE CASCADE ) ]

**Back list(from:INTEGER, to:INTEGER)**

[both from and to is reference to the uid of users table, and both of them comprise the primary key. Plus, when the user in from or in to is deleted, the corresponding term in this table is also deleted(on delete cascade)]

**Friend list(user1,user2)**

[both user1and user1 is reference to the uid of users table, and both of them comprise the primary key. Plus, when the user in user1 or in user2 is deleted, the corresponding term in this table is also deleted(on delete cascade)]

**3.** **Explanation**

**I. 1 to N relations**

Posts: One post could only be owned by one user, so posts and users have a N-to-1 relationship. So we should add uid of user as foreign key to the posts schema and uid should not be null.

Comments: One comment could only be owned by one user and be attached to only one post, while a user could add multiple comments and a post could have multiple comments. So comments has a many-to-one relationship with posts and user. And comments are total participations of both two relations. So we should add uid reference to user and pid reference to posts as foreign keys to the schema and they should not be null.

Announcements: It is a weak entity and it is n to 1 with the groups entity, because each group can generate more than one announcement. The partial key, aid of announcements depends on the groups, because the aid is defined by groups. So, the primary key for announcements is composed of aid and the gid of groups, and when a group is deleted, the group's announcements are also deleted.

**II. N to N relations**

Subscribe: Subscribe is n to n relationship, because one category could be subscribed by multiple user and one user could subscribe multiple categories. So both uid of user and cat\_id of categories comprise the primary key

Manage: Same as subscribe, uid and cat\_id comprise its primary. However, Its uid is admin's uid.

Join: Join is n to n relationship, because many user could join in one group and a user could join in multiple group. So Both gid of groups and uid of user consist the primary key

Classify: It is n to n relationship, because multiple posts could be classified under one category while one post could only be classified into one category. What’s more, posts are total participations in this relation, every posts must be classified. So pid of posts and cat\_id of categories are primary key and cat\_id as the foreign key should not be null.

Friend\_list: Is n to n relationship, one user could add multiple user into friend list and one user could be added by multiple user into friend list, so both from and to are primary key

Black\_list: Is n to n relationship, one user could add multiple user into black list and one user could be added by multiple user into black list, so both user1 and user2 are primary key