Database Management Systems

Project phase 2

Messaging Application

SQL Specification

Authored by: Martin Ting

May 10, 2014

Summary

This document details the implementation of the schema for our Messaging Application. The schema for the entities, relationships, and all the constraints are defined in the SQL script. Below are some details regarding design considerations, restrictions, and issues faced in this version of the SQL script.

Team & Roles

- Martin Ting - 860997603
  - SQL design and documentation
- Kevin Suh - 860992968

Entities and Relationships

Entities:

```
Users( phoneNum char(13) NOT NULL UNIQUE,
 login char(50) UNIQUE,
 password char(50) not null,
 status char(140),
 PRIMARY KEY(login))
User_List( owner char(50) NOT NULL,
 contact char(50) NOT NULL,
 list_type char(10) NOT NULL,
 PRIMARY KEY(owner, contact, list_type),
 FOREIGN KEY(owner) REFERENCES Users(login)
  ON DELETE CASCADE,
 FOREIGN KEY(contact) REFERENCES Users(login)
  ON DELETE CASCADE );
Chats( cid int,
 login char(50),
 chat_type char(10) NOT NULL,
 PRIMARY KEY(cid),
 FOREIGN KEY(login) REFERENCES Users
  ON DELETE CASCADE);
Notification( login char(50),
 mid int,
 PRIMARY KEY(login, mid),
 FOREIGN KEY(login) REFERENCES Users
  ON DELETE CASCADE,
 FOREIGN KEY(mid) REFERENCES Messages
  ON DELETE CASCADE);
```

```
Messages( mid int,
 text char(140) NOT NULL,
 timestamp timestamp NOT NULL,
 destr_timestamp timestamp,
 status char(30) NOT NULL,
 login char(50),
 cid int,
 PRIMARY KEY(mid),
 FOREIGN KEY(login) REFERENCES Users
  ON DELETE CASCADE,
 FOREIGN KEY(cid) REFERENCES Chats
  ON DELETE CASCADE);
Media( id int,
 media_type char(10),
 URL char(256) NOT NULL,
 mid int,
 PRIMARY KEY(id),
 FOREIGN KEY(mid) REFERENCES Messages
  ON DELETE CASCADE);
```

Relationships:

```
Contains( login char(50),
 owner char(50),
 PRIMARY KEY(login, owner),
 FOREIGN KEY(login) REFERENCES Users);
```

```
ChatList(login char(50),
 cid int,
 PRIMARY KEY(login, cid),
 FOREIGN KEY(login) REFERENCES Users
  ON DELETE NO ACTION,
 FOREIGN KEY(cid) REFERENCES Chats
  ON DELETE CASCADE);
```

Note: Block, Contact, Initial_Sender, Sender, Has, Receiver, Notify, Contains are represented in existing entity tables.

Design Considerations for Entities

Users:

- Phone number and logins are unique to each user.
- A password must be set if a User account exists.

User_List:

- Userlist defines two users and a list_type in each tuple
- The first user, aptly named owner, is the user for which all other users in the tuple (**OWNER**, other users, list type) are associated with. In other words, that user is the owner of the user list that contains all other users associated to the owner.
- The entries of the user_list can be a contact or a blocked user to the owner.

Notification:

- Notifications tell users about new messages.
- A new attribute timestamp is defined to log the time the notification was sent.

Design Consideration for Relationships

Contains: expressed in Contains table

- Participation:
  - Enforced in the SQL design.
- Referential:
  - When the “Owner” of a user list is deleted, the contacts contained in respect to that user is deleted also.

Block: Expressed in User_List table

- Participation:
  - Enforced in the SQL design. The logins for both owner and contact must be defined either in Contact or Block.
- Referential:
  - When the “Owner” of a block list is deleted, the contacts blocked by that user is also deleted

Contact: Expressed in User_List table

- Participation:
  - Enforced in the SQL design. The logins for both owner and contact must be defined either in Contact or Block.
- Referential:
  - When the “Owner” of the contact list is deleted, the contacts of that user is also deleted.

Initial_Sender: Expressed in Chat table

- Participation:
  - Enforced in the SQL Design. The foreign key login must be defined.
- Referential:
  - When the initial sender is deleted, the chat is also deleted. It was designed this way because for our system, the initial sender acts as a moderator for the chat and there is no definition for a reassignment in case that the moderator is deleted so we will delete the chats as well.

Sender: Expressed in Message table

- Participation:
    - Participation is enforced. The foreign key login must be defined.
- Referential:
    - Once a User is deleted, the messages sent by that user is deleted as well.

Contains: Expressed in Message table

- Participation:
    - Participation is enforced. The foreign key cid must be defined. By our system specifications, a chat must be created with the first message so the many constraint is enforced as well.
- Referential:
    - When a Chat is deleted, all the messages in that chat is deleted as well.

Has: Expressed in Media

- Participation:
    - Participation is enforced. The foreign key mid must be defined in media.
- Referential:
    - When message is deleted, the attachment associated with that message is deleted as well.

Receiver: Expressed in Notification

- Participation:
    - Participation is enforced. foreign key login must be defined.
- Referential:
    - Once a user is deleted, so are the notifications associated with that user.

Notify: Expressed in Notification

- Participation:
    - Participation is enforced. foreign key mid must be defined.
- Referential:
    - Once a message is deleted, so are the notifications associated with that message.

ChatList: expressed in ChatList table

- Participation:
    - Participation is enforced.
- Referential:
    - Chats must be cleared before deleting a user. Clearing a chat is a step before deleting the user for our user deletion protocol.
    - Once a chat is deleted, the entry in the chatList is deleted.