

```
CREATE SCHEMA letstalk;
```

```
USE letstalk;
```

```
CREATE TABLE user_group  
(group_id CHAR(20),  
  name CHAR(20) NOT NULL,  
  group_manager CHAR(20) NOT NULL,  
  number_of_people INTEGER NOT NULL,  
  PRIMARY KEY(group_id));
```

```
CREATE TABLE user_birthday_and_age  
(birthday CHAR(20),  
  age INTEGER,  
  PRIMARY KEY(birthday));
```

```
CREATE TABLE user  
(user_id CHAR(20),  
  name CHAR(20) NOT NULL,  
  headshot LONGBLOB,  
  birthday CHAR(20),  
  gender CHAR(20),  
  password CHAR(20) NOT NULL,  
  PRIMARY KEY (user_id),  
  FOREIGN KEY (birthday) REFERENCES user_birthday_and_age(birthday)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE);
```

```
CREATE TABLE user_in_group  
(group_id CHAR(20),  
  user_id CHAR(20),  
  PRIMARY KEY(group_id, user_id),  
  FOREIGN KEY (group_id) REFERENCES user_group(group_id)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  FOREIGN KEY (user_id) REFERENCES user(user_id)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE);
```

```
CREATE TABLE emoji
    (emoji_number CHAR(20),
     PRIMARY KEY (emoji_number));
```

```
CREATE TABLE own
    (user_id CHAR(20),
     emoji_number CHAR(20),
     PRIMARY KEY(user_id, emoji_number),
     FOREIGN KEY(user_id) REFERENCES user(user_id)
         ON DELETE CASCADE
         ON UPDATE CASCADE,
     FOREIGN KEY(emoji_number) REFERENCES emoji(emoji_number)
         ON DELETE CASCADE
         ON UPDATE CASCADE);
```

```
CREATE TABLE interest
    (interest_id CHAR(20),
     interest_name CHAR(20),
     type CHAR(20),
     PRIMARY KEY (interest_id));
```

```
CREATE TABLE has
    (user_id CHAR(20),
     interest_id CHAR(20),
     PRIMARY KEY(user_id, interest_id),
     FOREIGN KEY(user_id) REFERENCES user(user_id)
         ON DELETE CASCADE
         ON UPDATE CASCADE,
     FOREIGN KEY(interest_id) REFERENCES interest(interest_id)
         ON DELETE CASCADE
         ON UPDATE CASCADE);
```

```
CREATE TABLE is_friend_of
    (user_id CHAR(20),
     friend_id CHAR(20),
     PRIMARY KEY (user_id, friend_id),
     FOREIGN KEY(user_id) REFERENCES user(user_id)
         ON DELETE CASCADE
         ON UPDATE CASCADE,
     FOREIGN KEY(friend_id) REFERENCES user(user_id)
         ON DELETE CASCADE
```

ON UPDATE CASCADE);

```
CREATE TABLE photo_post
  (user_id CHAR(20),
   photo_id CHAR(20),
   text CHAR(20),
   time CHAR(20),
   content LONGBLOB,
   PRIMARY KEY(user_id, photo_id),
   FOREIGN KEY(user_id) REFERENCES user(user_id)
     ON DELETE CASCADE
     ON UPDATE CASCADE);
```

```
CREATE TABLE communication_event_take
  (user_id CHAR(20),
   event_number CHAR(20),
   date CHAR(20),
   time CHAR(20),
   PRIMARY KEY(user_id, event_number),
   FOREIGN KEY(user_id) REFERENCES user(user_id)
     ON DELETE CASCADE
     ON UPDATE CASCADE);
```

```
CREATE TABLE notification
  (nid INTEGER,
   status CHAR(20),
   sound LONGBLOB,
   visibility BOOLEAN NOT NULL,
   PRIMARY KEY(nid));
```

```
CREATE TABLE contain
  (nid INTEGER,
   user_id CHAR(20),
   event_number CHAR(20),
   PRIMARY KEY(nid, user_id, event_number),
   FOREIGN KEY(user_id, event_number) REFERENCES communication_event_take(user_id,
   event_number)
     ON DELETE CASCADE
     ON UPDATE CASCADE,
```

```
FOREIGN KEY(nid) REFERENCES notification(nid)
    ON DELETE CASCADE
    ON UPDATE CASCADE);
```

```
CREATE TABLE message
    (message_id CHAR(20),
    sender CHAR(20) NOT NULL,
    receiver CHAR(20) NOT NULL,
    read_or_unread BOOLEAN NOT NULL,
    content LONGBLOB NOT NULL,
    PRIMARY KEY(message_id));
```

```
CREATE TABLE text
    (message_id CHAR(20),
    word_count INTEGER NOT NULL,
    PRIMARY KEY(message_id),
    Foreign key (message_id) references message(message_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
    );
```

```
CREATE TABLE image
    (message_id CHAR(20),
    format CHAR(20) NOT NULL,
    PRIMARY KEY(message_id),
    Foreign key(message_id) references message(message_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
    );
```

```
CREATE TABLE voice
    (message_id CHAR(20),
    length INTEGER NOT NULL,
    PRIMARY KEY(message_id),
    Foreign key (message_id) references message(message_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
    );
```

```
CREATE TABLE include
```

```

(message_id CHAR(20),
user_id CHAR(20),
event_number CHAR(20),
PRIMARY KEY(message_id, user_id, event_number),
FOREIGN KEY(message_id) REFERENCES message(message_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
FOREIGN KEY(user_id,event_number) REFERENCES communication_event_take(user_id,
event_number)
        ON DELETE CASCADE
        ON UPDATE CASCADE);

```

```

CREATE TABLE use_message_manager
(manager_id CHAR(20),
user_id CHAR(20) NOT NULL,
number_of_message INTEGER NOT NULL,
PRIMARY KEY(manager_id),
FOREIGN KEY(user_id) REFERENCES user(user_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE);

```

```

CREATE TABLE administrate
(message_id CHAR(20),
manager_id CHAR(20),
PRIMARY KEY(message_id, manager_id),
Foreign key (message_id) references message(message_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
Foreign key (manager_id) references use_message_manager(manager_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE);

```