

به نام خدا

بردیا صباغ کرمانی

40131411

پروژه درس اصول طراحی پایگاه داده

فاز صفر



در این فاز تمرکز بیشتر بر روی طراحی و ساختار روابط (relations) و جدول های پروژه بود.

طبق قوانین و روش طراحی ای که document پروژه گفته شده، ابتدا یک جدول برای کاربران با نام USERS ایجاد کردیم.

```
CREATE TABLE USERS(  
  id SERIAL PRIMARY KEY,  
  username VARCHAR(25) NOT NULL UNIQUE,  
  firstName VARCHAR(25) NOT NULL,  
  lastName VARCHAR(25),  
  phoneNumber VARCHAR(25) NOT NULL UNIQUE  
);
```

شکل (1)

طبق شکل (1)، رابطه USERS، متشکل از یک id (که auto increment است) به عنوان کلید اصلی، نام کاربری (به صورت یکتا و NOT NULL)، نام (به صورت NOT NULL)، نام خانوادگی و شماره تلفن (به صورت یکتا و NOT NULL) می باشد. بدین صورت که از کاربر یک نام کاربری (که باید خاص باشد)، نام و نام خانوادگی (وارد کردن نام خانوادگی الزامی نمی باشد) و یک شماره تلفن دریافت میکند و آن را در جدول می گذارد.

جدول بعدی CONTACT است.

```
CREATE TABLE CONTACT(  
    userid INT NOT NULL,  
    contactID INT NOT NULL,  
    PRIMARY KEY(userid, contactID),  
    FOREIGN KEY (userid) REFERENCES USERS (id),  
    FOREIGN KEY (contactID) REFERENCES USERS (id)  
);
```

شکل (2)

طبق چیزی که در شکل (2) نشان داده شده، این جدول برای ذخیره مخاطبین کاربران است. دقت کنید که اگر کاربر شماره 1، کاربر شماره 2 را به عنوان مخاطب خود ذخیره کرده باشد، لزوماً برعکس آن صادق نیست. پس برای رسیدگی به این موضوع، دو attribute برای کاربر در نظر گرفتیم (که foreign key از رابطه USERS می باشد).

Userid نشان دهنده کسی است که دارد مخاطب را ذخیره میکند و contactID نشان دهنده کسی است که دارد به عنوان مخاطب ذخیره میشود.

کلید اصلی این رابطه یک کلید ترکیبی از این دو id می باشد. (برای یکتایی)

رابطه تعریف شده ی بعدی، CHAT است.

```
Run | New Tab
CREATE TYPE chatType_t AS ENUM ('pv', 'gp');

Run | New Tab | Copy
CREATE TABLE CHAT (
    chatID SERIAL UNIQUE PRIMARY KEY,
    chatName VARCHAR(25),
    chatType chatType_t NOT NULL
);
```

شکل (3)

طبق شکل (3)، ابتدا یک type جدید (از نوع ENUM)، به نام chatType_t ساخته ایم که دو حالتِ pv و gp دارد. علت ساخت این type برای تبیین نوع چت بین کاربران می باشد که حالت گفتگوی خصوصی یا گروهی را مشخص میکند.

این رابطه chatID را به عنوان کلید اصلی دارد که به صورت auto increment است.

یک chatName برای نام آن chat (که برای چت های خصوصی NULL است) را دارد و یک chatType که بالاتر تعریف شد.

جدول بعدی، جدول CHATMEMBERS است.

```
Run | New Tab | Copy
CREATE TABLE CHATMEMBERS (
  userID INT NOT NULL,
  chatID INT NOT NULL,
  PRIMARY KEY (userID, chatID),
  FOREIGN KEY (userID) REFERENCES USERS(id),
  FOREIGN KEY (chatID) REFERENCES CHAT(chatID)
);
```

شکل (4)

این رابطه ابتدا یک userID از جدول USERS به عنوان foreign key دریافت میکند و یک chatID از جدول CHAT که ترکیب این دو، کلید اصلی این رابطه است.

این جدول برای نشان دادن اعضای چت ها می باشد.

```

> Run | Copy
CREATE OR REPLACE FUNCTION check_chat_constraint()
RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM CHAT
        WHERE chatID = NEW.chatID AND chatType = 'pv'
    ) THEN
        IF (SELECT COUNT(*)
            FROM CHATMEMBERS
            WHERE chatID = NEW.chatID
        ) >= 2 THEN
            RAISE EXCEPTION 'A private chat cannot have more than 2 members';
        END IF;
    END IF;

    IF (NEW.chatId in (
        select c3.chatid from chatmembers as c3 join chat as ch2 on ch2.chatid = c3.chatid
        where ch2.chattype = 'pv' and c3.userid in
            (select c2.userid from chatmembers c2 where c2.userId != NEW.userId and c2.chatid in
              (select c.chatid from chatmembers as c join chat as ch on ch.chatid = c.chatid where c.userId = NEW.userId and ch.chattype = 'pv'))
    )) THEN
        RAISE EXCEPTION 'There can only exist one private chat between 2 users';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

> Run | New Tab
CREATE TRIGGER chat_constraint_trigger
BEFORE INSERT ON CHATMEMBERS
FOR EACH ROW
EXECUTE FUNCTION check_chat_constraint();

```

شکل (5)

برای اینکه در یک چت خصوصی، بیش از دو نفر نتوانند عضو شوند و برای اینکه زمانی بین دو نفر یک چت خصوصی برقرار است، چت دیگری ساخته نشود، یک trigger function تعریف کرده ایم که این دو شرط را چک میکند و در صورت نقض این شروط، عملیات Insert انجام نمیشود.

جدول آخر، جدول Message است که برای ذخیره پیام ها می باشد.

```
Run | New Tab | Copy
CREATE TABLE MESSAGES(
  messageID SERIAL PRIMARY KEY,
  chatID INT NOT NULL,
  senderID INT NOT NULL,
  context TEXT NOT NULL,
  ts TIMESTAMP NOT NULL,
  FOREIGN KEY (chatID) REFERENCES CHAT(chatID),
  FOREIGN KEY (senderID) REFERENCES USERS(id)
);
```

شکل (6)

کلید اصلی این رابطه messageID است که به صورت auto increment است.

سپس chatID که مشخص میکند این پیام برای کدام چت است، foreign key از جدول CHAT می باشد.

senderID برای مشخص کردن این است که فرستنده پیام کدام کاربر است که این attribute نیز foreign key از جدول USERS است.

Attribute بعدی محتوای پیام که از نوع TEXT می باشد و timestamp که نشان دهنده تاریخ و زمان ارسال پیام است.