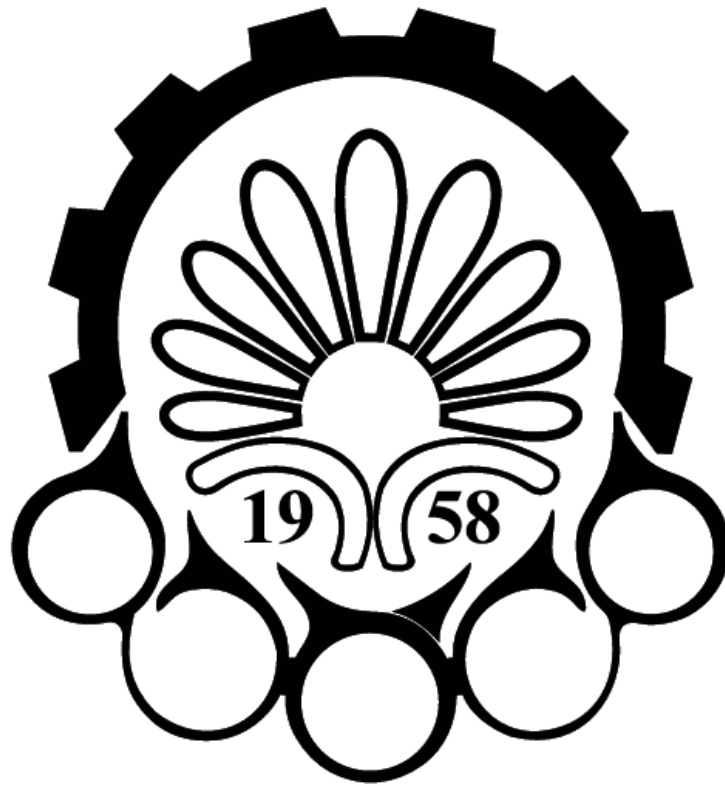


به نام خدا



Amirkabir University of Technology
(Tehran Polytechnic)

فاز صفر پروژه اصول طراحی و مدیریت پایگاه داده

کاوه احمدی 40131904

در این بخش از پروژه قرار است که جداولی که برای پیاده سازی پروژه مورد نیاز است را پیاده سازی کرده و توضیح مختصری راجع به آن ها بدهیم.

• جدول کاربران :

در این جدول ما اطلاعات فردی هر کاربر شامل نام، نام خانوادگی، رمز عبور، شماره تلفن منحصر به فرد هر کاربر و شناسه یکتای هر کاربر را ذخیره کنیم.

```
1 • CREATE TABLE users(  
2   ID VARCHAR(255) UNIQUE NOT NULL,  
3   phoneNumber INT UNIQUE NOT NULL,  
4   pass VARCHAR(255) NOT NULL,  
5   FName VARCHAR(255) NOT NULL,  
6   LName VARCHAR(255) NOT NULL,  
7   PRIMARY KEY(ID, phoneNumber)  
8 );
```

مطابق قطعه کد فوق جدولی برای مشخصات فردی هر شخص می سازیم که در آن برای هر کاربر یک شناسه از جنس رشته داریم که منحصر به فرد است و یک عدد برای شماره تلفن که یکتاست و یک رمز عبور که یک رشته حداکثر به طول 256 کاراکتر و یک نام و نام خانوادگی با طول حداکثر 256 کاراکتر

• جدول اعتبارسنجی :

جدول authentication برای بررسی ورود هر کاربر هست که اطلاعاتی که از کاربر login کرده برای بخش سرور مهم است را نگه می داریم

این اطلاعات شامل : شناسه، شماره تلفن، IP، و یک بیت برای این که بررسی کنیم آیا کاربر دوست دارد login بماند یا هر سری مشخصات خود را وارد کند. شایان ذکر است که باید IP همان باشد و بیت saveLogin مقدار 1 داشته باشد.

```
1 • CREATE TABLE authentication(  
2     ID VARCHAR(255),  
3     phoneNumber INT UNIQUE,  
4     IP VARCHAR(15),  
5     saveLogin BIT,  
6     PRIMARY KEY(ID, phoneNumber),  
7     FOREIGN KEY(ID, phoneNumber) REFERENCES users(ID, phoneNumber)  
8 );
```

• جدول مخاطبین :

این جدول تنها شامل دو ستون هست که شماره تلفن 1 و شماره تلفن 2 می باشد

بدان معنا که شماره تلفن 1 در لیست مخاطبین خود شماره تلفن 2 را دارد. شماره تلفن ها هر کدام به یک user ارتباط دارد که اطلاعات user1 با شماره تلفن 1 را از جدول authentication می آوریم چون لاگین صورت گرفته و اگر نیاز به اطلاعات بیشتر بود با

authentication ای که صورت گرفته از جدول users اطلاعات اضافه برای user1 را می آوریم ولی برای user2 چون authentication لزوما وجود ندارد مستقیم از جدول users اطلاعات مورد نظر را می آوریم

```
1 • CREATE TABLE contacts(  
2     phoneNumber1 INT,  
3     phoneNumber2 INT,  
4     PRIMARY KEY(phoneNumber1, phoneNumber2),  
5     FOREIGN KEY(phoneNumber2) REFERENCES users(phoneNumber)  
6 );
```

• جدول چت ها :

در این جدول اطلاعات هر چت را نگه داری می کنیم به این شکل که در هر رکورد از این جدول یک شماره تلفن 1 وجود دارد که به صورت NOT NULL تعریف شده که نشان دهنده این است که این شماره تلفن این چت را دارد.

برای شما دیگر چت 2 حالت وجود دارد:

1-شماره تلفن 2

groupID-2

یکی از دو مقدار فوق هر سری مقدار NULL گرفته و مورد دیگر قطعا مقداری NOT NULL دارد که هر یک نشان دهنده این است که شماره

تلفن 1 با شماره تلفن 2 یک چت دارد یا این که شماره تلفن 1 در گروهی با شناسه x به نام y هست

```
1 • CREATE TABLE chats(  
2     phoneNumber1 INT NOT NULL,  
3     phoneNumber2 INT,  
4     groupName VARCHAR(255),  
5     groupID INT UNIQUE AUTO_INCREMENT,  
6     PRIMARY KEY(phoneNumber2, groupId),  
7     FOREIGN KEY(phoneNumber1) REFERENCES authentication(phoneNumber)  
8 );
```

• جدول پیام ها :

جدول است مانند جدول چت ها با این تفاوت که کلید های جدول چت را دارد و به هر چت پیام هایی که مربوط به آن ها هست را اختصاص می دهد

```
1 • CREATE TABLE messages(  
2     phoneNumber1 INT,  
3     phoneNumber2 INT,  
4     groupID INT,  
5     groupName VARCHAR(255),  
6     message VARCHAR(2047),  
7     FOREIGN KEY (phoneNumber2, groupId) REFERENCES chats(phoneNumber2, groupId)  
8 );
```

در ادامه روند پروژه قرار است با استفاده از advanced SQL سری تغییرات در ساختمان جداول مورد نیاز پروژه داد مانند اضافه کردن چند trigger function به عنوان مثال trigger function زیر برای این است که هر شخص تنها یک چت خصوصی با یک نفر دیگر داشته باشد :

```

1  DELIMITER //
2  • CREATE TRIGGER chats_check
3  BEFORE INSERT ON chats
4  FOR EACH ROW
5  BEGIN
6      DECLARE flag INT;
7      SELECT COUNT(*) INTO flag
8      FROM chats
9      WHERE (phoneNumber1 = NEW.phoneNumber1 AND phoneNumber2 = NEW.phoneNumber2)
10     OR (phoneNumber2 = NEW.phoneNumber1 AND phoneNumber2 = NEW.phoneNumber1);
11     IF flag > 0 THEN
12         SIGNAL SQLSTATE '45000'
13         SET MESSAGE_TEXT = 'Duplicate record found. Transaction rolled back.';
14     END IF;
15 END //
16 DELIMITER ;

```

در نهایت یک ER Diagram از نمای کلی جداول و روابطی که بین آن ها هست به شکل زیر می شود :

