AT82.02

DATA MODELING AND MANAGEMENT

LAB02: SQL PART I

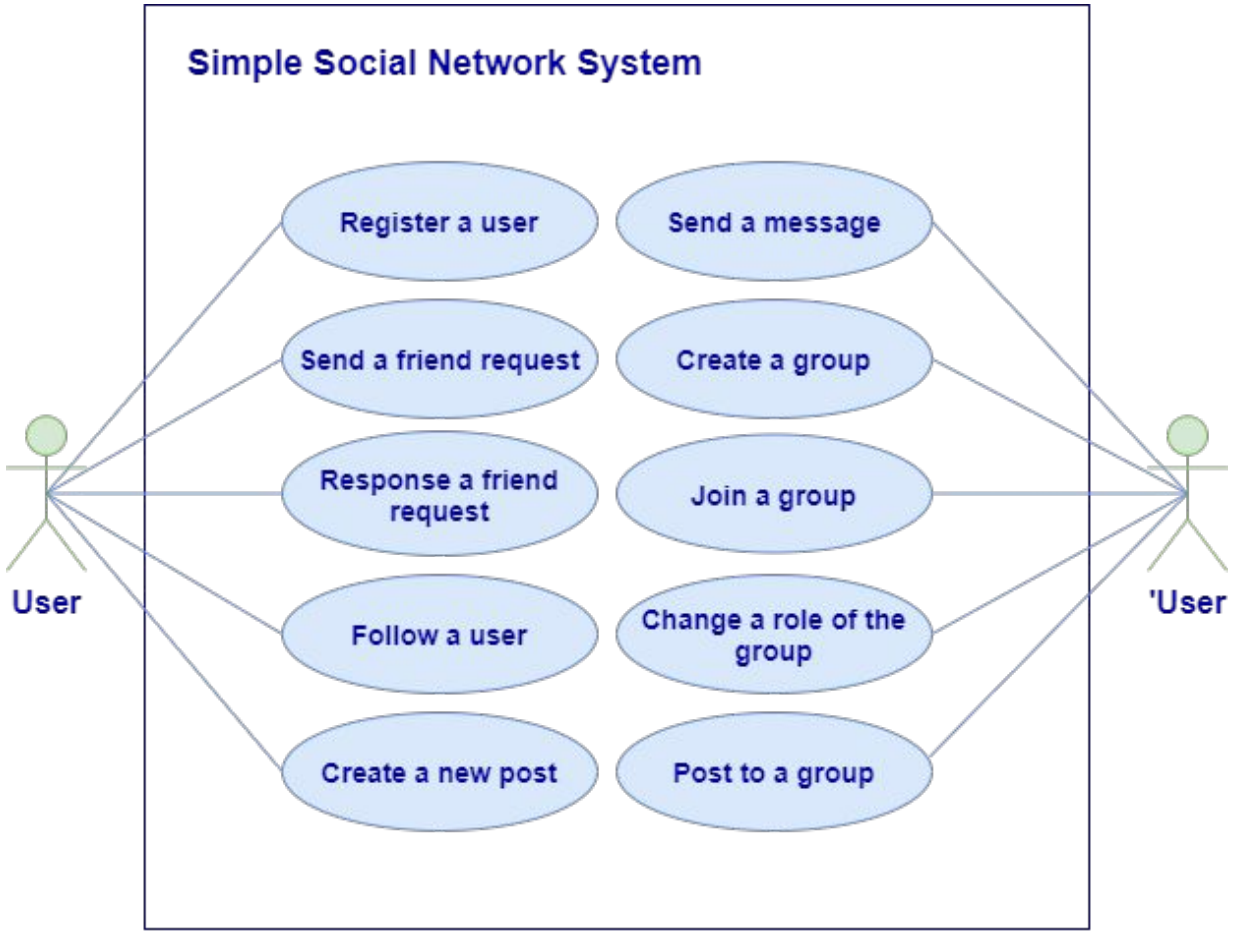
DS&AI

Case Study: Social Network System

- In this lab series (Lab 02 – Lab 04), we will follow the case study of a Social Network System.

- We will cover some parts in the class, a few of them is supposed to be completed by student as part of ***Take Home Exercise.***

Features of Social Networking Site



Lab02: Outline

- Create database objects (database, tables)
  - **`CREATE DATABASE|TABLE`**
- Modify database objects
  - **`ALTER TABLE`**
  - **`ALTER COLUMN`**
  - **`ADD PRIMARY KEY | INDEX | CONSTRAINT`**
- Manage data in the table
  - **`INSERT`**
  - **`UPDATE`**
  - **`DELETE`**

USERS

Column Name	Description
Id	The unique id to identify the user.
First Name	The first name of the user.
Middle Name	The middle name of the user.
Last Name	The last name of the user.
Mobile	The mobile number of the user. It can be used for login and registration purposes.
Email	The email of the user. It can be used for login and registration purposes.
Password Hash	The password hash generated by the appropriate algorithm. We must avoid storing plain or encrypted passwords.
Registered At	This column can be used to calculate the life of the user with the application.
Last Login	It can be used to identify the last login of the user.
Intro	The brief introduction of the User.
Profile	User details.

CREATE DATABASE

```
-- create a new database named sns
CREATE DATABASE sns;
```

CREATE TABLE user

```
CREATE TABLE `sns`.`user` (
  `id` BIGINT NOT NULL AUTO_INCREMENT,
  `firstName` VARCHAR(50) NULL, `middleName` VARCHAR(50) NULL,
  `lastName` VARCHAR(50) NULL, `username` VARCHAR(50) NULL,
  `mobile` VARCHAR(15) NULL, `email` VARCHAR(50) NOT NULL,
  `passwordHash` VARCHAR(32) NOT NULL,
  `registeredAt` DATETIME NOT NULL DEFAULT now(),
  `lastLogin` DATETIME NULL, `intro` TINYTEXT NULL,
  `bio` TEXT NULL,
  PRIMARY KEY (`id`)
);
```

Adding UNIQUE columns

Some user's information must be unique to only ONE user, Therefore, we need to add UNIQUE index to some columns,

```
ALTER TABLE sns.user
   ADD UNIQUE (`username`),
   ADD UNIQUE (`mobile`),
   ADD UNIQUE (`email`);
```

Register a new user

To register a new user, the INSERT command is usually employed.

```
INSERT INTO sns.user
   (firstName, middleName, lastName, username, mobile, email,
passwordHash, registeredAt)
VALUES
   ('Sergio', 'Checo', 'Perez',
'checo11','6415499120','checo.perez@gmail.com',md5('redbull@11'),
now());
```

Questions?

1. Can we insert a user without specifying each column?
   - `middleName`
   - `registeredAt`
   - `email`
2. Can we insert 2 users having the same mobile phone number or email?

Answers

1. Can we insert a user without specifying each column?
   - **`middleName`**, because it is nullable
   - **`registeredAt`**, because it has a default value (from **`now()`** function)
   - **`email`**, because it is required for user registration (**`NOT NULL`**)

   Therefore, when inserting a new row, specify values of ALL **`NOT NULL`** column unless they have no **`DEFAULT`** value.

2. No, because they are **`UNIQUE`** column, meaning that each mobile phone and email can be used to identify only ONE user at a time.

In-class exercise (1)

1. Insert 3 users according to each scenario?
   - User 1: a user who has only first name, last name, username, password, email, and a mobile phone number
   - User 2: a user having his 20-character middle name
   - User 3: a user with around 50-word introduction

Show your commands in an sql file (*.sql). For the invalid scenarios, *specify the reason why it is impossible to insert.*

Adding a column

We have realized that we should store a gender of each user (if he/she is willing to specify). Therefore, we use the following ALTER command to add the gender column.

```
ALTER TABLE sns.user

   ADD COLUMN gender ENUM ('m','f') NULL;
```

Question:

Can we add a column with `NOT NULL` constraint when we have some data in the `user` table?

UPDATE user

Suppose a user wants to update their profile (e.g. change their names or add their bio in their profile), the following UPDATE command will be used to these actions.

```
UPDATE sns.user

SET    gender = 'm',

    intro = 'Formula One | Red Bull Racing',

   bio = 'Sergio Michel Checo Perez is a mexican born Formula One racing driver, driving for Red Bull Racing'

WHERE user.id = 1;
```

Phewww…!

Let’s Take a short break?

Create User Friend table

To allow the users to send a friend request and response to it, we need a table to store who are friends among users in the system.

Column Name	Description
Id	The unique id to identify the friendship.
Source Id	The user id to identify the user who initiated the friendship.
Target Id	The user id of the target friend (user).
Status	The status can be New, Rejected, or Accepted.
Created At	It stores the date and time at which the friend request was initiated.
Updated At	It stores the date and time at which the friend request was updated.

CREATE TABLE user_friend

CREATE command for the table ‘user_friend’

```
CREATE TABLE `sns`.`user_friend` (
  `id` BIGINT NOT NULL AUTO_INCREMENT, `sourceId` BIGINT NOT NULL,
  `targetId` BIGINT NOT NULL,
  `status` ENUM ('new','accepted','rejected') DEFAULT 'new',
  `createdAt` DATETIME NOT NULL,`updatedAt` DATETIME NULL,
  PRIMARY KEY (`id`)
);
```

Question

Is it enough to have only a primary key in user_friend table?

Answer: To maintain the referential integrity, foreign keys should be defined so that we only have the record of existing users in this table, **`sourceId`** and **`targetId`**.

Adding constraints (Foreign Keys)

```
ALTER TABLE sns.user_friend
    ADD CONSTRAINT fk_friend_source
        FOREIGN KEY (sourceId) REFERENCES sns.user(id)
        ON DELETE RESTRICT ON UPDATE RESTRICT,
    ADD CONSTRAINT fk_friend_target
        FOREIGN KEY (targetId) REFERENCES sns.user(id)
        ON DELETE RESTRICT ON UPDATE RESTRICT,
    ADD UNIQUE uq_friends (sourceId, targetId);
```

Send a friend request

Sergio Perez with ID 1, wants to be a friend of Esteban Ocon (ID=5), so he sends a friend request to Ocon.

The equivalent INSERT command to the user_friend table is.

```
INSERT INTO sns.user_friend (sourceId, targetId, createdAt)
VALUES (1, 5, now());
```

Question: *What is the status of this friend request?*

Response to the request

When Ocon sees the friend request from Sergio, let's assume that he wants to accept the request. The equivalent Update command for this would be:

```
UPDATE sns.user_friend

SET status = 'accepted'

WHERE sourceId = 1 AND targetId = 5;
```

In-class exercise (2)

1. Write an SQL command to store **two friend requests** from a user to two other users (define your user as you want).
2. Write another command to respond the above friend requests, one for **acceptation** and another for **rejection**.

Show your commands corresponding the above problems in the sql file.

Terminating an account

Suppose Esteban Ocon wants to quit using social network. He finally terminated his account. The system may execute the following command.

```
DELETE FROM sns.user WHERE id=5;
```

Does the above command work for now? Can you try?

Answer: No, because of the restriction of the FOREIGN KEY of user_friend table (having RESTRICT option).

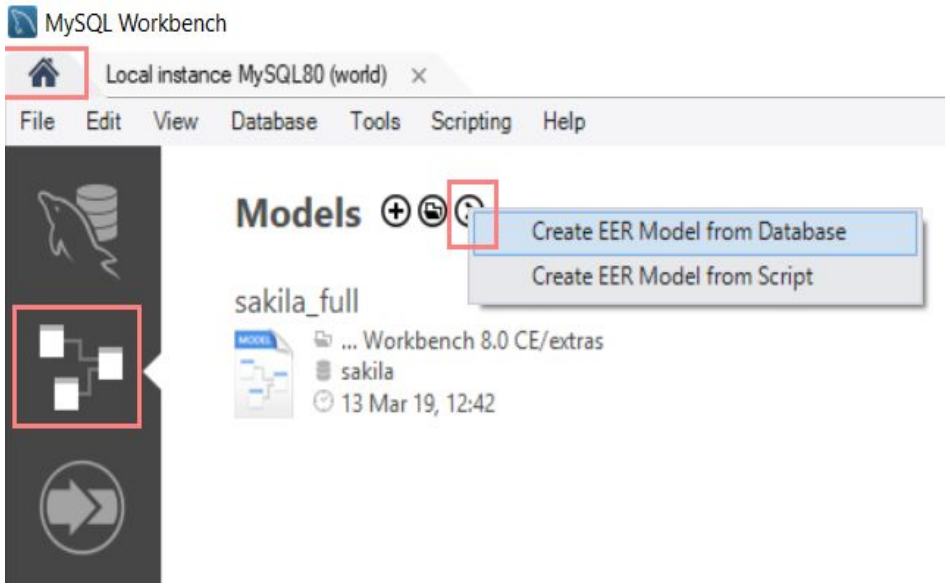
Changing FK constraints

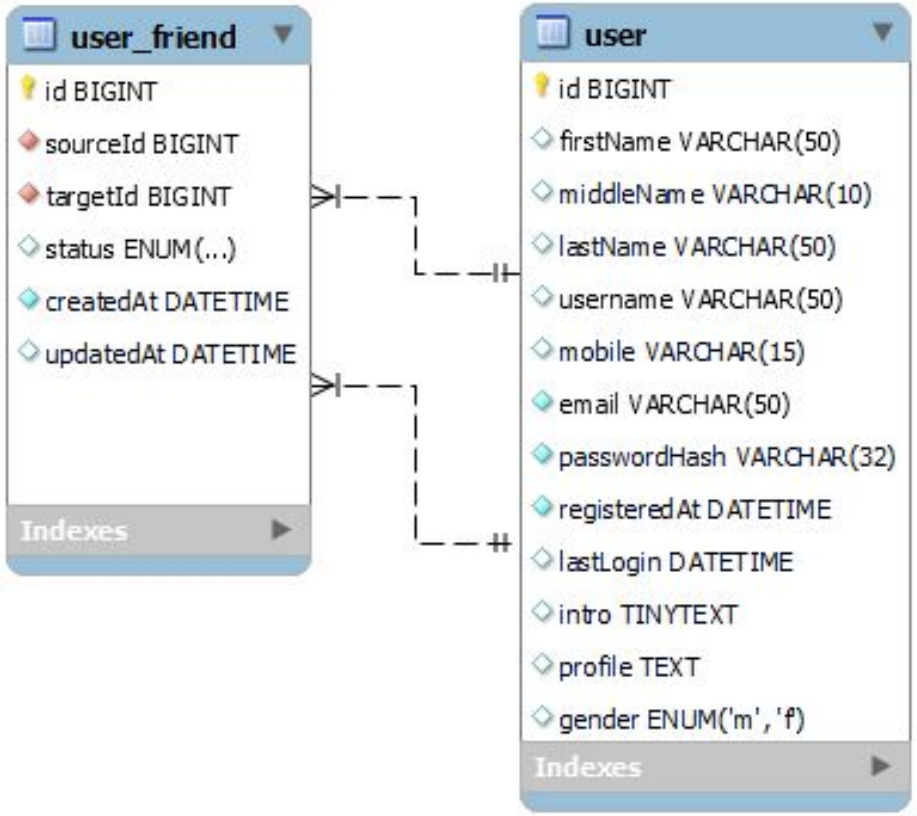
```
-- Drop all FKs and indicies first

ALTER TABLE user_friend

    DROP FOREIGN KEY fk_friend_source,

    DROP FOREIGN KEY fk_friend_target,

    DROP INDEX uq_friends;
```

Changing FK constraints

```
-- Then recreate FKs and a unique key
ALTER TABLE sns.user_friend
    ADD CONSTRAINT fk_friend_source
        FOREIGN KEY (sourceId) REFERENCES sns.user(id)
        ON DELETE CASCADE,
    ADD CONSTRAINT fk_friend_target
        FOREIGN KEY (targetId) REFERENCES sns.user(id)
        ON DELETE CASCADE,
    ADD UNIQUE uq_friends (sourceId, targetId);
```

Creating EER Diagram





Take home exercise

Complete the corresponding tasks in the LAB 02 instruction sheets given in the ***GOOGLE CLASSROOM***.

Due: **Thursday 25 August 2022, 11.00 P.M.**

References

SQL Tutorial on W3 Schools:
https://www.w3schools.com/sql/

Question?