****

**School of Engineering & Technology**

**Asian Institute of Technology**

**AT82.02 - Data Modeling and Management**

**Date: 24th August 2022**

**Lab Report: 2**

**Submitted To Submitted By:**

**Dr. Chutiporn Anutariya Todsavad Tangtortan [st123012]**

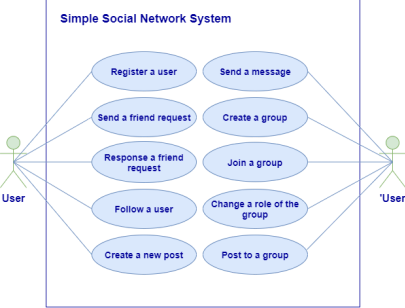
**Lab02: SQL (Part I)**

**Objective:** To study and practice the SQL commands to create database objects and manipulate data in the relational database.

**Estimated Time:** 3 hours

**Due:** Thursday 26 August 2022, 11:00 P.M.

**Description: Simple Social Network System (SNS)**

****

As in the in-class practices, we have created two tables with related keys and constraints to support the use case **‘Register a user’, ‘Send a friend request’ and ‘Response to a friend request’**.

This take-home lab will require you to create more stuff for the use case **‘Create a new post’** and **‘Send a message’**. Please consider the following instructions and do the given practices. Some items will ask you to show the SQL command while others will require you to explain your reason on your design as well as your implemented SQL command for the above system.

**Reference for SQL commands:** https://www.w3schools.com/sql/

**Instruction:**

***Task 1 to 7 refer to the use case ‘Create a new post’.***

## Task 1:

Use the given table description to create a table user\_post. This table can support a post on a public news feed or a post on the other user’s wall. Show your CREATE TABLE command with the appropriate data types, primary key(s) and necessary constraints (foreign keys and unique columns, etc.).

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| id | The unique id to identify the post. |
| posterId | The user id to identify who create the post |
| profileId | The profile of the target user where the poster wants to post to (leave it null for the public post). |
| message | The message body. |
| createAt | It stores the date and time at which the post was created. |
| updatedAt | It stores the date and time at which the post was updated. |

*[Paste your SQL command here]*

*DROP TABLE sns.user\_post;*

-- CREATE TABLE user\_post

CREATE TABLE `sns`.`user\_post` (

`id` BIGINT NOT NULL UNIQUE AUTO\_INCREMENT,

`posterId` BIGINT NOT NULL,

`profileId` BIGINT NULL,

`message` TEXT NULL,

`createAt` DATETIME NOT NULL DEFAULT now(),

`updatedAt` DATETIME NULL ,

PRIMARY KEY(`id`),

FOREIGN KEY(`posterId`) REFERENCES user(`id`) ON DELETE CASCADE,

FOREIGN KEY(`profileId`) REFERENCES user(`id`) ON DELETE CASCADE

);

SELECT \* from sns.user\_post;

## Task 2:

From your *user\_post* table, which columns you have set to be a NOT NULL constraint? List those columns and explain your reason.

*[Put your answer here]*

*`id` because it is a primary key so it cannot be a NULL*

*`posterId` because it is a foreign key from use.id*

*`createAt` because it refers to datetime timestamp*

## Task 3:

Explain the foreign keys of this table. What options you have chosen **(RESTRICT, CASCADE, or SET NULL)** and the reason why you did?

*[Put your answer here]*

I have set CASCADE on posterID and createAt because both of them are set as foreign key and if we want to delete some id from sns.user, it is going to be deleted from sns.user\_post as well.

## Task 4:

Insert two different posts from two different users. The first one is for a public post and the second is posted to the other user’s wall.

*[Paste your SQL command here]*

*-- public post*

*INSERT INTO sns.user\_post*

*(posterId, message)*

*VALUES (1,"Hi");*

*-- other user's wall*

*INSERT INTO sns.user\_post*

*(posterId, profileId, message)*

*VALUES (2,1,"Bye");*

## Task 5:

Suppose a user with userId = 3 wants to remove all the posts that they have ever created. Write an appropriate SQL command to delete all the posts created by the user. (both public and private posts)

*[Paste your SQL command here]*

*DELETE FROM sns.user\_post WHERE posterId=3;*

## Task 6:

Suppose the system allows users to add a photo to the post, how do you modify the user\_post table structure? [Hint: Add a new column called attachment/photos]

*[Paste your SQL command here]*

*ALTER TABLE sns.user\_post*

*ADD COLUMN photos BINARY(1) NULL;*

## Task 7:

All posts on the social network system are monitored. Some of them can be deleted by the administrator if they violate the terms and conditions of usage (e.g. hate speech and fake news, etc.). What is the appropriate **DELETE** command to remove such violating posts (suppose that the administrator knows the id of the post they want to delete).

*[Paste your SQL command here]*

*DELETE FROM sns.user\_post WHERE id=2;*

***Task 8 to 12 refer to the use case ‘Send a message’.***

## Task 8:

Use the given table description to create a table user\_message. This table can support sending a message to the desired user. Show your **CREATE TABLE** command with the appropriate data types, primary key(s) and necessary constraints (foreign keys and unique columns, etc.).

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| id | The unique id to identify the message |
| sourceId | The user id to identify the sender. |
| targetId | The user id to identify the receiver. |
| message | The message body. |
| createdAt | It stores the date and time at which the post was created. Set the default value as now(). |
| updatedAt | It keeps the date and time at which the post was updated. |

*[Paste your SQL command here]*

DROP TABLE sns.user\_message;

CREATE TABLE `sns`.`user\_message` (

`id` BIGINT NOT NULL UNIQUE AUTO\_INCREMENT,

`sourceId` BIGINT NOT NULL,

`targetId` BIGINT NOT NULL,

`message` TEXT NULL,

`createAt` DATETIME NOT NULL DEFAULT now(),

`updatedAt` DATETIME NULL ,

PRIMARY KEY(`id`),

FOREIGN KEY(`sourceId`) REFERENCES user(`id`),

FOREIGN KEY(`targetId`) REFERENCES user(`id`)

);

SELECT \* from sns.user\_message;

## Task 9:

From your user\_message table, are there any columns that can be NULLABLE? Explain all such columns and explain your reason.

*[Put your answer here]*

`message` because user is able to consider to put or not put message

`updatedAt` because when we insert data to data base at first time, it is not involve in updateAt

## Task 10:

Insert data to the user\_message table to store 2-exchange conversation messages between 2 users.

List all INSERT commands.

*[Put your answer here]*

INSERT INTO sns.user\_message

(sourceId, targetId, message)

VALUES (1,2,"1 communicate to 2");

INSERT INTO sns.user\_message

(sourceId, targetId, message)

VALUES (2,1,"2 communicate to 1");

## Task 11:

If the system owner needs to know the status of the message (seen or delivered), how do you modify this table? Show how you update the table structure.

*[Paste your SQL command here]*

*ALTER TABLE sns.user\_message*

*ADD COLUMN status ENUM ('seen','delivered') NULL;*

## Task 12:

Use MySQL Workbench to generate the EER Diagram and put your screenshot of this diagram showing how all tables are designed.

*[Paste your screenshot here]*

