# Relational Databases with MySQL Week 3 Coding Assignment

**Points possible:** 70

|  |  |  |
| --- | --- | --- |
| Category | Criteria | % of Grade |
| Functionality | Does the code work? | 25 |
| Organization | Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear. | 25 |
| Creativity | Student solved the problems presented in the assignment using creativity and out of the box thinking. | 25 |
| Completeness | All requirements of the assignment are complete. | 25 |

**Instructions:** Using a text editor of your choice, write the queries that accomplishes the objectives listed below. Take screenshots of the queries and results and paste them in this document where instructed below. Create a new repository on GitHub for this week’s assignments and push this document to the repository. Additionally, push an .sql file with all your queries and your ERD to the same repository. Add the URL for this week’s repository to this document where instructed and submit this document to your instructor when complete.

**Coding Steps:**

You have been asked to create a database for a new social media application that your company is developing.

The database must store user data such as username, email, password, etc...

Users are able to post and comment. So, your database must also store post and comment data.

We need to know which user made which posts.

We also need to know which user made which comments, and which post a comment is on.

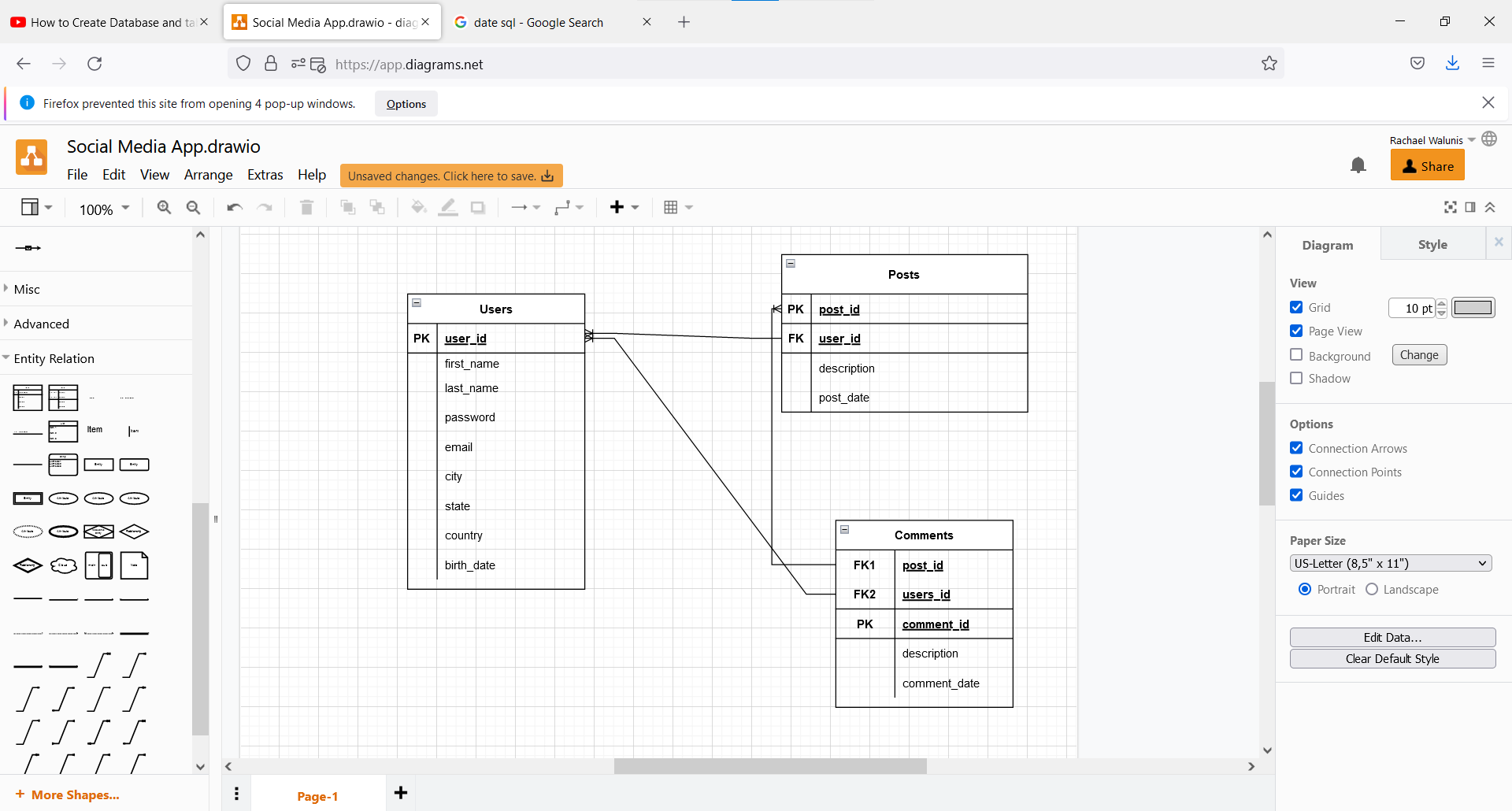
Posts and comments should both include the time they were created, and what the content of the post or comment is.

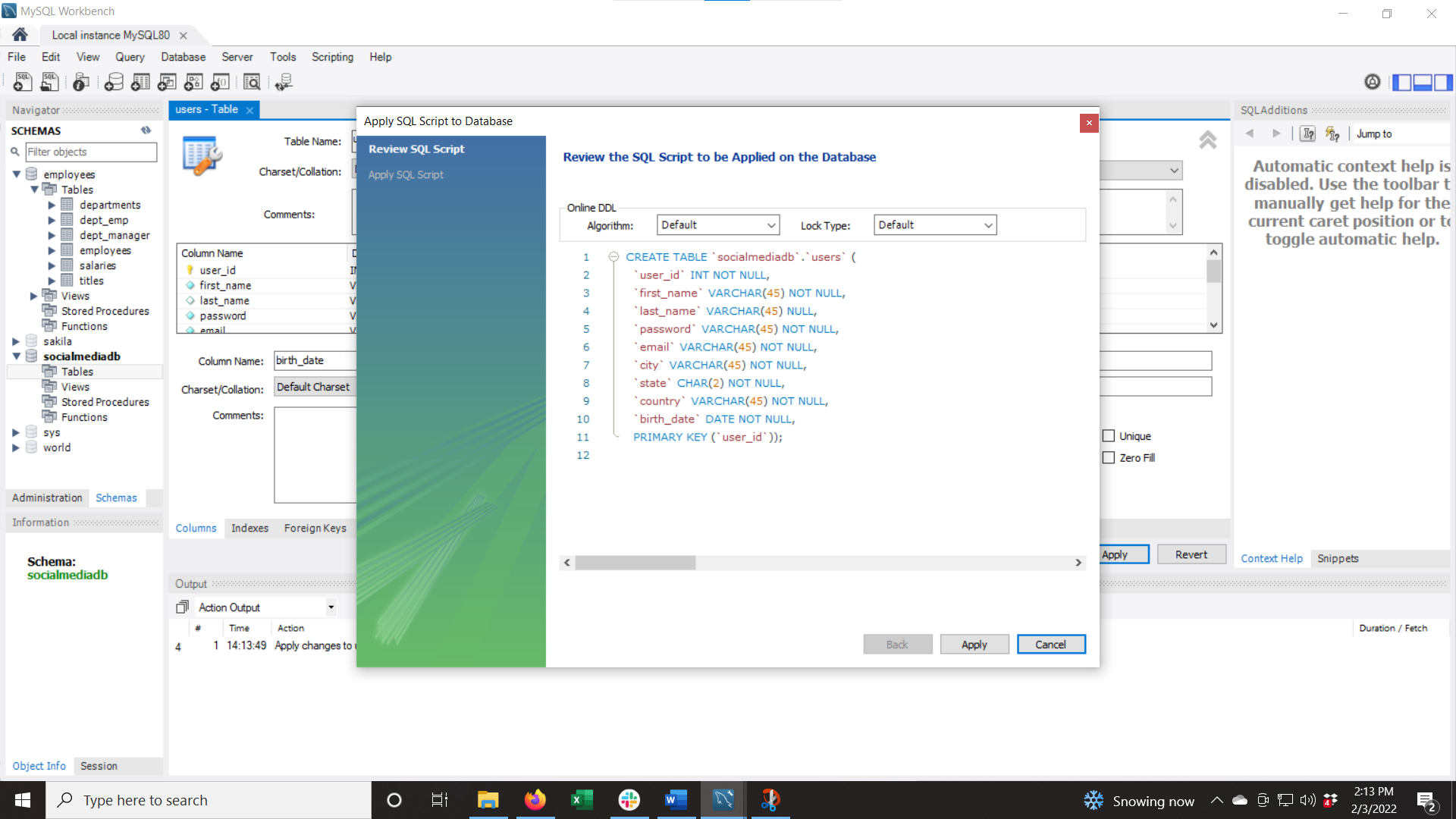
Create an Entity Relationship Diagram (ERD) using draw.io to model the database you will create. Insert a screenshot of the ERD in the screenshots section below.

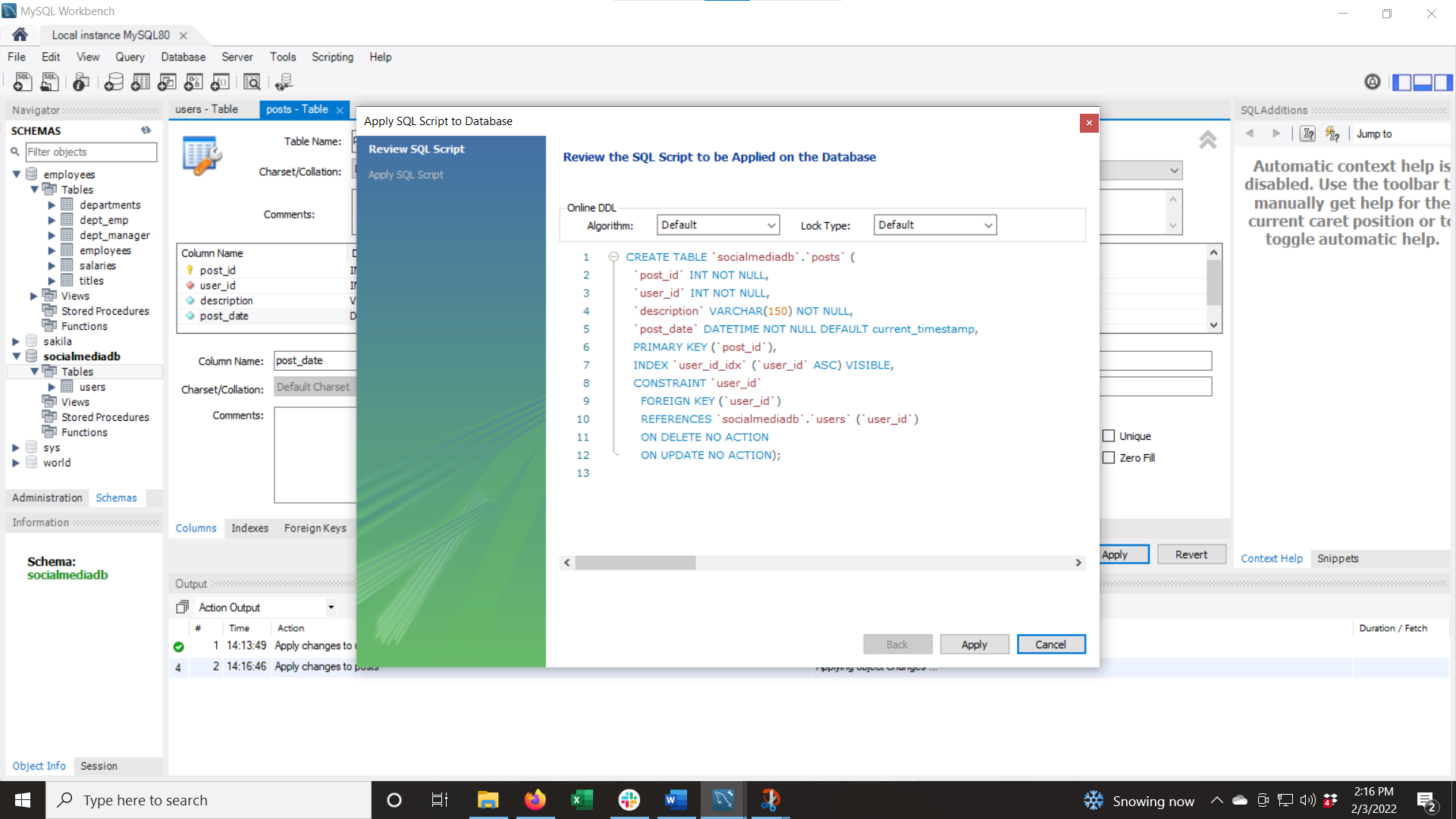
Write a SQL script to create the database. Insert a screenshot of the SQL in your script.

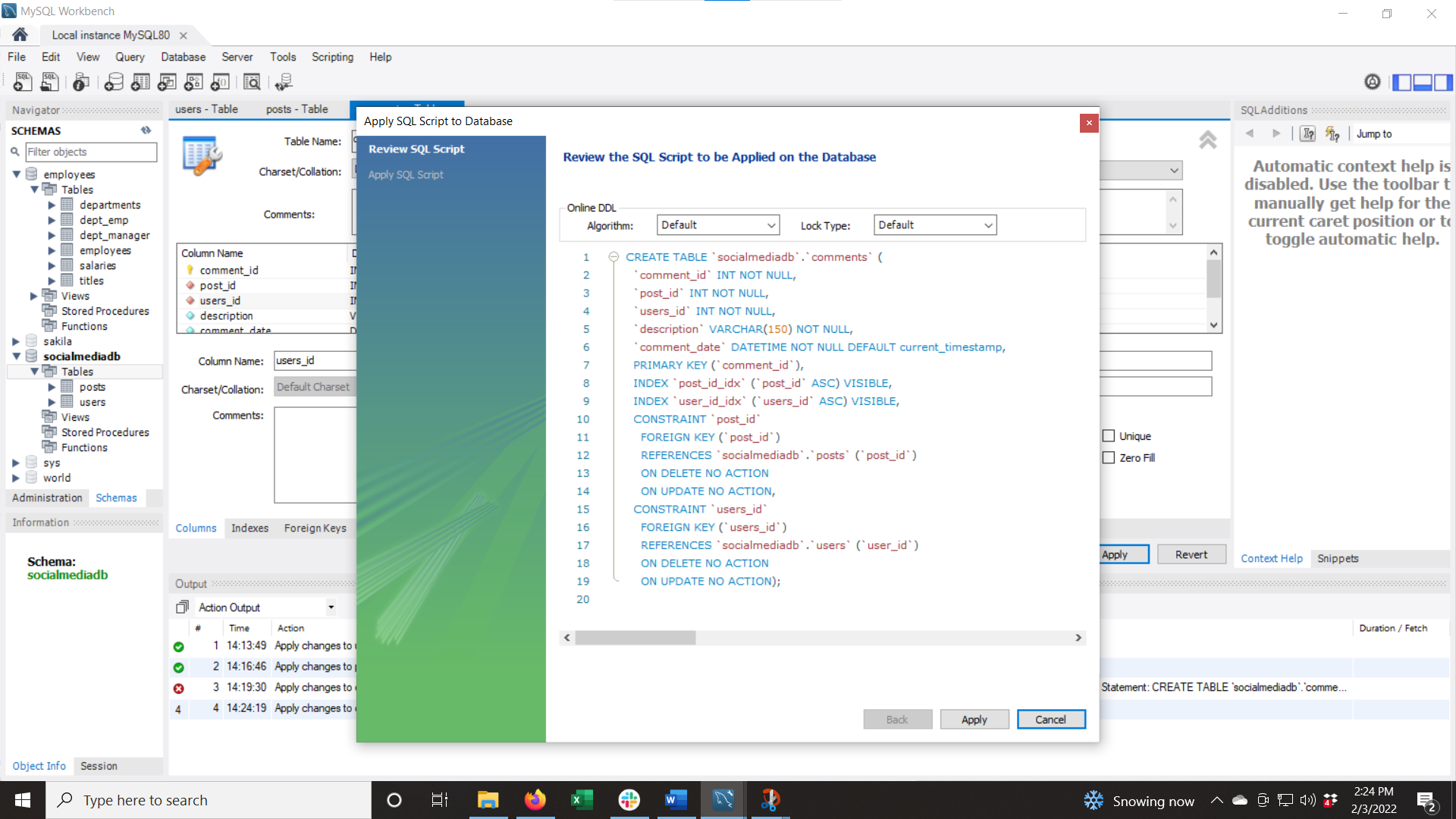
**Screenshots:**

I initially thought to create a composite key for the users table containing the post\_id and user\_id as foreign keys. But with more thinking, a user who creates a post may comment on their post multiple times. For example, user Bob may initially post “What’s up, world?” but then Bob may comment multiple times on that same post (example: “No one cares to respond?” and then comment again with “Whatever…”) So in those instances there would have to be a unique primary key, comment\_id, to differentiate the comments instead of creating and using a composite key of post\_id and user\_id in the comments table.

****

****

****

****