

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.

Hints:

You will only need three tables.

Two tables will have foreign key references.

One table will have two foreign key references.

Screenshots:

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

socialmediapp

Tables

comment

Columns

comment_id

username

post_id

comment_date

comment_content

Indexes

Foreign Keys

Triggers

post

Columns

post_id

username

post_date

post_content

Indexes

Foreign Keys

Triggers

user

Columns

username

email

password

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

Administration Schemas

Information

Week1Employees Week2Employees SQL File C

Limit to 1000 rows

```
1 DROP database if exists socialMediaApp;
2 create database if not exists socialMediaApp;
3 use socialMediaApp;
4
5 -- DROP TABLE IF EXISTS comment;
6 -- DROP TABLE IF EXISTS post;
7 -- DROP TABLE IF EXISTS user;
8
9 CREATE TABLE user (
10     username varchar(50) NOT NULL,
11     email varchar(50) NOT NULL,
12     password varchar(50) NOT NULL,
13     primary key (username)
14 );
15
16 CREATE TABLE post (
17     post_id int NOT NULL auto_increment,
18     username varchar(50) NOT NULL,
19     post_date date NOT NULL,
20     post_content varchar(255) NOT NULL,
21     primary key (post_id),
22     foreign key (username) references user(username)
23 );
24
25 CREATE TABLE comment (
26     comment_id int NOT NULL auto_increment,
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Table: post

Columns:

post_id int AI PK

username varchar(50)

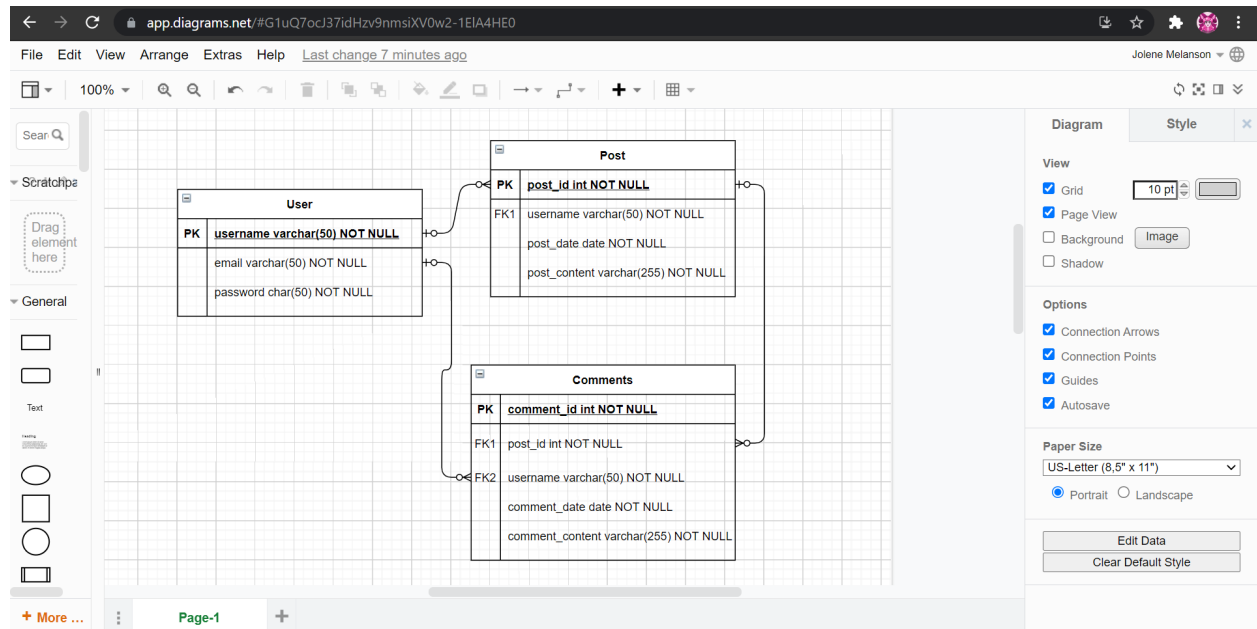
post_date date

post_content varchar(255)

Output

#	Time	Action	Message	Duration / Fetch
1	15:20:40	DROP database if exists socialMediaApp	3 row(s) affected	0.032 sec
2	15:20:40	create database if not exists socialMediaApp	1 row(s) affected	0.000 sec
3	15:20:40	use socialMediaApp	0 row(s) affected	0.000 sec
4	15:20:40	CREATE TABLE user (username varchar(50) NOT NULL, email varchar(50) NOT NULL, password varchar(50) NOT NULL, primary key (username))	0 row(s) affected	0.015 sec
5	15:20:40	CREATE TABLE post (post_id int NOT NULL auto_increment, username varchar(50) NOT NULL, post_date date NOT NULL, post_content varchar(255) NOT NULL, primary key (post_id), foreign key (username) references user(username))	0 row(s) affected	0.031 sec
6	15:20:40	CREATE TABLE comment (comment_id int NOT NULL auto_increment, username varchar(50) NOT NULL, post_id int NOT NULL, comment_date date NOT NULL, comment_content varchar(255) NOT NULL, primary key (comment_id), foreign key (post_id) references post(post_id), foreign key (username) references user(username))	0 row(s) affected	0.016 sec

```
9  ● ○ CREATE TABLE user (  
10      username varchar(50) NOT NULL,  
11      email varchar(50) NOT NULL,  
12      password varchar(50) NOT NULL,  
13      primary key (username)  
14  );  
15  
16  ● ○ CREATE TABLE post (  
17      post_id int NOT NULL auto_increment,  
18      username varchar(50) NOT NULL,  
19      post_date date NOT NULL,  
20      post_content varchar(255) NOT NULL,  
21      primary key (post_id),  
22      foreign key (username) references user(username)  
23  );  
24  
25  ● ○ CREATE TABLE comment (  
26      comment_id int NOT NULL auto_increment,  
27      username varchar(50) NOT NULL,  
28      post_id int NOT NULL,  
29      comment_date date NOT NULL,  
30      comment_content varchar(255) NOT NULL,  
31      primary key (comment_id),  
32      foreign key (username) references user(username),  
33      foreign key (post_id) references post(post_id)  
34  );
```



URL to GitHub Repository:

<https://github.com/JoleneMel/Week3SocialMediaApp>