# 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, with your Java project code, to the 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.

CREATE SCHEMA FakeBook;

USE FakeBook;

CREATE TABLE USER (

user\_ID Int NOT NULL AUTO\_INCREMENT,

firstName VARCHAR(50) NULL,

lastName VARCHAR(50) NULL,

userName VARCHAR(50)NOT NULL,

email VARCHAR(128) NOT NULL,

password VARCHAR(12) NOT NULL,

createDtTm DATETIME NOT NULL,

PRIMARY KEY(user\_ID)

);

CREATE TABLE POST (

post\_ID int NOT NULL AUTO\_INCREMENT,

user\_ID int NOT NULL,

title Varchar(100) NOT NULL,

body text NOT NULL,

createDtTm DATETIME NOT NULL,

Primary key (post\_ID),

Constraint FK\_post\_user

Foreign Key(user\_ID)

References User(user\_ID)

);

Create table COMMENT (

comment\_ID int NOT NULL AUTO\_INCREMENT,

post\_ID int NOT NULL,

User\_ID int NOT NULL,

Body Text NOT NULL,

CreateDtTm DATETIME NOT NULL,

Primary key (Comment\_ID),

Constraint FK\_comment\_post

Foreign key (Post\_ID)

References POST(post\_ID),

Constraint FK\_comment\_user

Foreign key (user\_ID)

References USER(user\_ID)

);

INSERT into USER(firstName, lastName, userName, email, password, createDtTm)

VALUES ('Elizabeth', 'Rossi', 'LizRossi', 'elizabethbrossi@yahoo.com', 'password', now());

INSERT into POST(user\_ID, title, body, createDtTm)

VALUES('1', 'Hello', 'Hello, local database, it is lovely to meet you!', now());

INSERT into COMMENT(post\_ID, user\_ID, BODY, createDtTm)

VALUES ('1', '1', 'I feel I should add that this is the realm of my existence, impossible as it is.', now());

SELECT\*FROM USER u JOIN POST p on u.user\_ID = p.user\_ID;

SELECT\*FROM USER u JOIN COMMENT c on u.user\_ID = c.user\_ID;

SELECT\*FROM USER u JOIN POST p on u.user\_ID = p.user\_ID JOIN COMMENT c on p.post\_ID = c.post\_ID;

Hints:

You will only need three tables.

Two tables will have foreign key references.

One table will have two foreign key references.

**Screenshots:**

A picture containing text, computer

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

**URL to GitHub Repository:**