LAB WEEK 06

We need a blog web application that allow people publishes their post. The infoemation of each object that related to our web are descript as below

User Table

In this section, we will design the **User Table** to store user information of all the post authors. The same table can be used to relate the post authors so that all the authors can manage their own posts. Below mentioned is the description of all the columns of the User Table.

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 passwords.
Registered At	This column can be used to calculate the life of the user with the blog.
Last Login	It can be used to identify the last login of the user.
Intro	The brief introduction of the Author to be displayed on each post.
Profile	The author details to be displayed on the Author Page.

Post Table

In this section, we will design the **Post Table** to store the post data. Below mentioned is the description of all the columns of the Post Table.

Id	The unique id to identify the post.
Author Id	The author id to identify the post author.
Parent Id	The parent id to identify the parent post. It can be used to form the table of content of the parent post of series.
Title	The post title to be displayed on the Post Page and the lists.
Meta Title	The meta title to be used for browser title and SEO.
Summary	The summary of the post to mention the key highlights.
Published	It can be used to identify whether the post is publicly available.
Created At	It stores the date and time at which the post is created.
Updated At	It stores the date and time at which the post is updated.
Published At	It stores the date and time at which the post is published.
Content	The column used to store the post data.

Post Comment Table

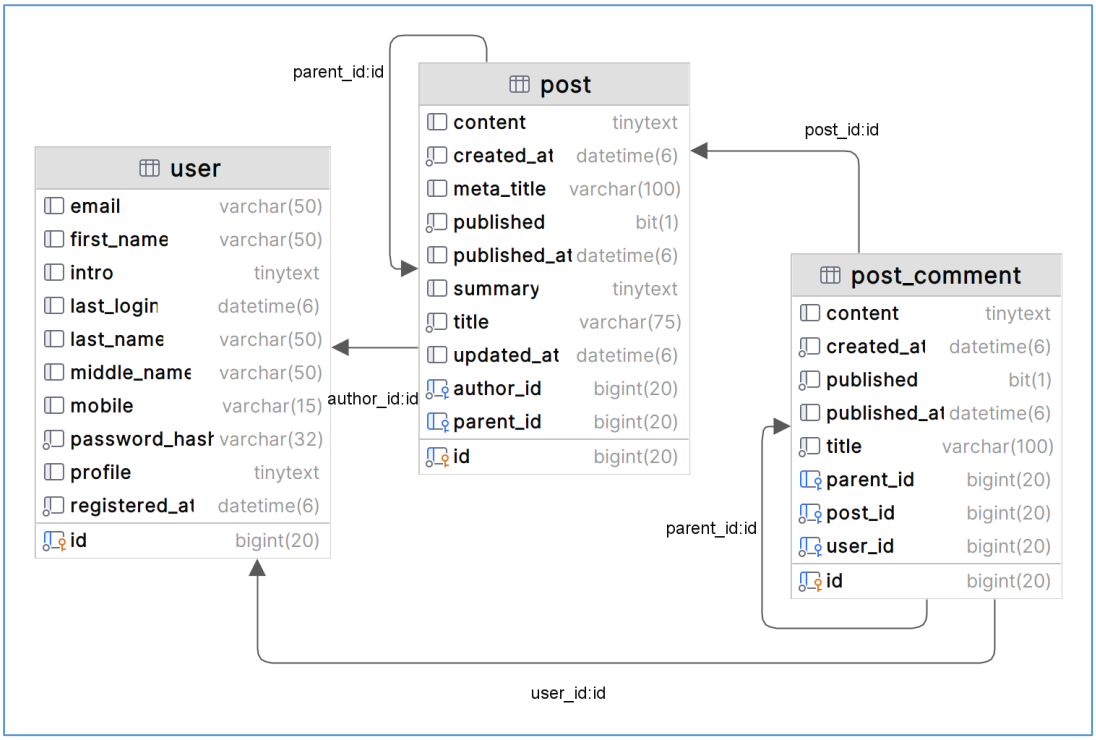
In this section, we will design the **Post Comment Table** to store the post comments. Below mentioned is the description of all the columns of the Post Comment Table.

Id	The unique id to identify the post comment.
Post Id	The post id to identify the parent post.
Parent Id	The parent id to identify the parent comment.
Title	The comment title.

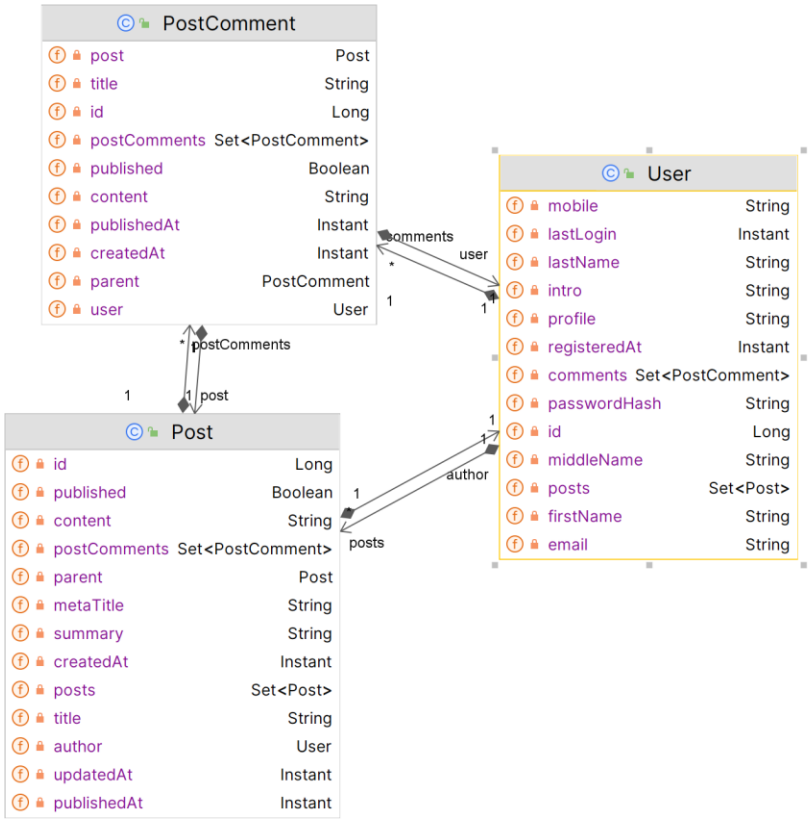
Published	It can be used to identify whether the comment is publicly available.
Created At	It stores the date and time at which the comment is submitted.
Published At	It stores the date and time at which the comment is published.
Content	The column used to store the comment data.

Create a Spring web application with name lab- 06, select uploading to Git hub.

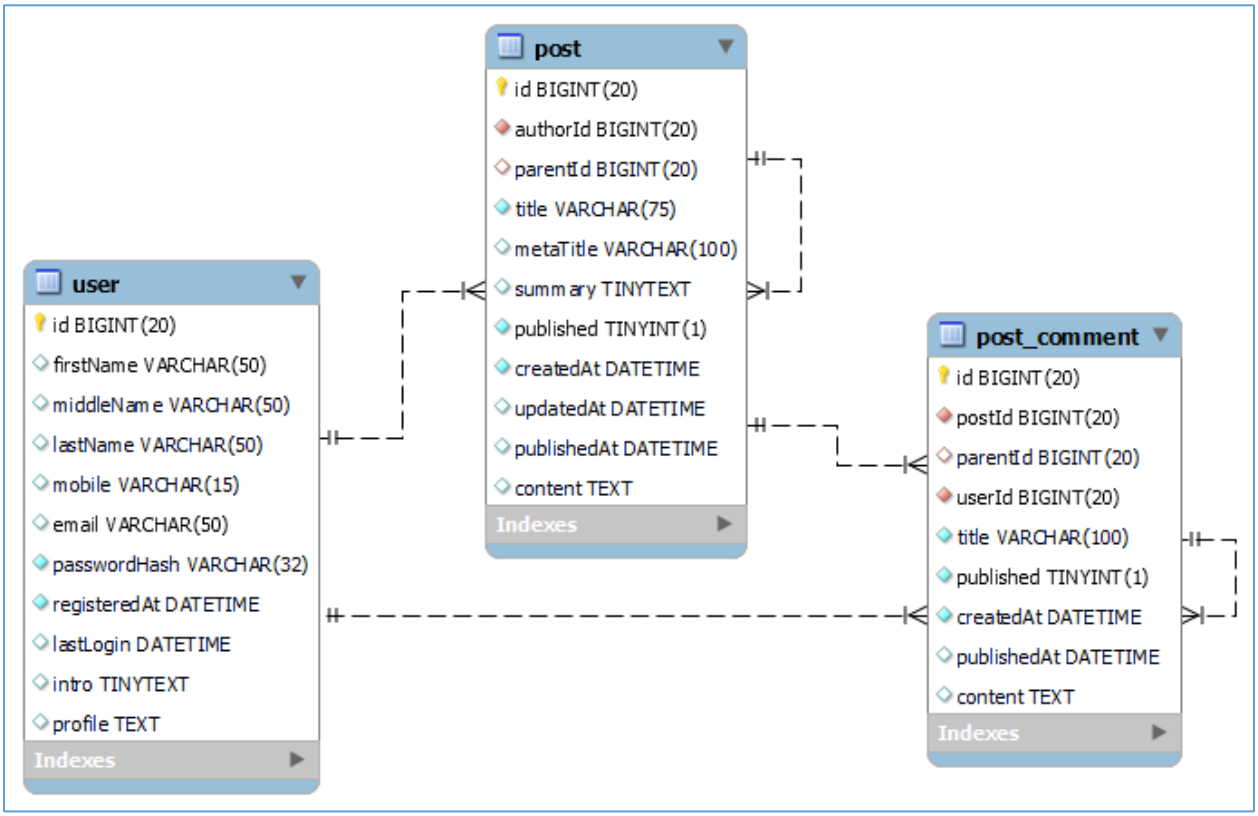
Use JPA mapping these tables. You can compare with the reference database as follow.



ER diagram



Class diagram



Database relationship diagram

Database script

```
DROP DATABASE IF EXISTS `blog`;
CREATE DATABASE IF NOT EXISTS `blog` /*!40100 DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci */;
USE `blog`;

-- Dumping structure for table blog.post
DROP TABLE IF EXISTS `post`;
CREATE TABLE IF NOT EXISTS `post` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `authorId` bigint(20) NOT NULL,
  `parentId` bigint(20) DEFAULT NULL,
  `title` varchar(75) NOT NULL,
  `metaTitle` varchar(100) DEFAULT NULL,
  `summary` tinytext DEFAULT NULL,
  `published` tinyint(1) NOT NULL DEFAULT 0,
  `createdAt` datetime NOT NULL,
  `updatedAt` datetime DEFAULT NULL,
  `publishedAt` datetime DEFAULT NULL,
  `content` text DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_post_user` (`authorId`),
  KEY `idx_post_parent` (`parentId`),
  CONSTRAINT `fk_post_parent` FOREIGN KEY (`parentId`) REFERENCES `post` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_post_user` FOREIGN KEY (`authorId`) REFERENCES `user` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

-- Data exporting was unselected.

-- Dumping structure for table blog.post_comment
DROP TABLE IF EXISTS `post_comment`;
CREATE TABLE IF NOT EXISTS `post_comment` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `postId` bigint(20) NOT NULL,
  `parentId` bigint(20) DEFAULT NULL,
  `title` varchar(100) NOT NULL,
  `published` tinyint(1) NOT NULL DEFAULT 0,
  `createdAt` datetime NOT NULL,
  `publishedAt` datetime DEFAULT NULL,
  `content` text DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_comment_post` (`postId`),
  KEY `idx_comment_parent` (`parentId`),
  CONSTRAINT `fk_comment_parent` FOREIGN KEY (`parentId`) REFERENCES `post_comment` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_comment_post` FOREIGN KEY (`postId`) REFERENCES `post` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

-- Data exporting was unselected.

-- Dumping structure for table blog.user
DROP TABLE IF EXISTS `user`;
CREATE TABLE IF NOT EXISTS `user` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `firstName` varchar(50) DEFAULT NULL,
  `middleName` varchar(50) DEFAULT NULL,
  `lastName` varchar(50) DEFAULT NULL,
  `mobile` varchar(15) DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  `passwordHash` varchar(32) NOT NULL,
```

```
  `registeredAt` datetime NOT NULL,
  `lastLogin` datetime DEFAULT NULL,
  `intro` tinytext DEFAULT NULL,
  `profile` text DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `uq_mobile` (`mobile`),
  UNIQUE KEY `uq_email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Create web pages that act like a blog web site.

** Add entities,… to complete the task