1. Created a database in mysql workbench:
2. For this first create a new connection for this click on plus option and give the database name and host as localhost and user is root only, then create a new connection.
3. Go to the connection and then on left side right click, then u will get an option to create a new schema so create it, i.e social in this project.
4. Again right click and create a table with users as table names and in that panel only add new columns and their constraints, columns are : id, username, email, password, name, coverPic, profilePic, city, website.
5. Another table for posts: id,desc,img,userid,createdAt.
6. Table for comments: CREATE TABLE `social`.`comments` (

`id` INT NOT NULL AUTO\_INCREMENT,

`desc` VARCHAR(200) NOT NULL,

`createdAt` DATETIME NULL,

`userid` INT NOT NULL,

`postid` INT NOT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE,

INDEX `postid\_idx` (`postid` ASC) VISIBLE,

INDEX `commentUserid\_idx` (`userid` ASC) VISIBLE,

CONSTRAINT `commentUserid`

FOREIGN KEY (`userid`)

REFERENCES `social`.`users` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `postid`

FOREIGN KEY (`postid`)

REFERENCES `social`.`posts` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE);

1. Table for stories:

CREATE TABLE `social`.`stories` (

`id` INT NOT NULL AUTO\_INCREMENT,

`img` VARCHAR(200) NULL,

`userid` INT NOT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE,

INDEX `storyUserid\_idx` (`userid` ASC) VISIBLE,

CONSTRAINT `storyUserid`

FOREIGN KEY (`userid`)

REFERENCES `social`.`users` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

1. Table for relationships:

CREATE TABLE `social`.`relationships` (

`id` INT NOT NULL AUTO\_INCREMENT,

`followerUserid` INT NOT NULL,

`followedUserid` INT NOT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE,

INDEX `followerUser\_idx` (`followerUserid` ASC) VISIBLE,

INDEX `followedUser\_idx` (`followedUserid` ASC) VISIBLE,

CONSTRAINT `followerUser`

FOREIGN KEY (`followerUserid`)

REFERENCES `social`.`users` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `followedUser`

FOREIGN KEY (`followedUserid`)

REFERENCES `social`.`users` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE);

1. Table for likes:

CREATE TABLE `social`.`likes` (

`id` INT NOT NULL AUTO\_INCREMENT,

`userid` INT NOT NULL,

`postid` INT NOT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `id\_UNIQUE` (`id` ASC) VISIBLE,

INDEX `likesUserid\_idx` (`userid` ASC) VISIBLE,

INDEX `likesPostid\_idx` (`postid` ASC) VISIBLE,

CONSTRAINT `likesUserid`

FOREIGN KEY (`userid`)

REFERENCES `social`.`users` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `likesPostid`

FOREIGN KEY (`postid`)

REFERENCES `social`.`posts` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE);