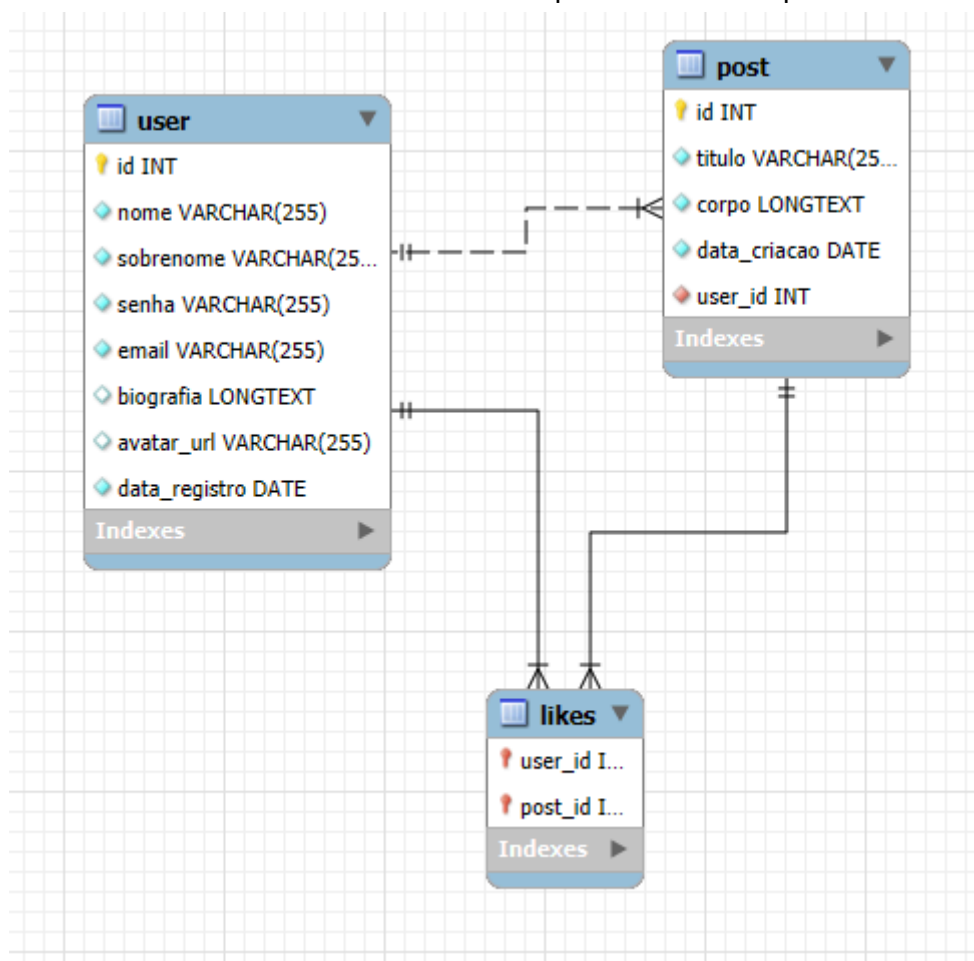


Mapeamento Inicial da Aula do dia 11/08/2025 que será utilizado para atividade prática



SQL do Modelo (MySQL)

Note que terá que adequar conforme o RDBMS usado.
Arquivo .sql para MariaDB(opensource mysql) está no sigaa.

```
-- -----  
-- Schema blogweb  
-- -----  
CREATE SCHEMA IF NOT EXISTS `blogweb` DEFAULT CHARACTER SET utf8 ;  
USE `blogweb` ;  
  
-- -----  
-- Table `blogweb`.`user`  
-- -----  
CREATE TABLE IF NOT EXISTS `blogweb`.`user` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(255) NOT NULL,  
  `sobrenome` VARCHAR(255) NOT NULL,  
  `senha` VARCHAR(255) NOT NULL,  
  `email` VARCHAR(255) NOT NULL,  
  `biografia` LONGTEXT NULL,  
  `avatar_url` VARCHAR(255) NULL,
```

```

    `data_registro` DATE NOT NULL,
    PRIMARY KEY (`id`),
    UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE)
ENGINE = InnoDB;

```

```

-----
-- Table `blogweb`.`post`
-----

```

```

CREATE TABLE IF NOT EXISTS `blogweb`.`post` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `titulo` VARCHAR(255) NOT NULL,
  `corpo` LONGTEXT NOT NULL,
  `data_criacao` DATE NOT NULL,
  `user_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_post_user_idx` (`user_id` ASC) VISIBLE,
  CONSTRAINT `fk_post_user`
    FOREIGN KEY (`user_id`)
      REFERENCES `blogweb`.`user` (`id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `blogweb`.`likes`
-----

```

```

CREATE TABLE IF NOT EXISTS `blogweb`.`likes` (
  `user_id` INT NOT NULL,
  `post_id` INT NOT NULL,
  PRIMARY KEY (`user_id`, `post_id`),
  INDEX `fk_likes_post1_idx` (`post_id` ASC) VISIBLE,
  CONSTRAINT `fk_likes_user1`
    FOREIGN KEY (`user_id`)
      REFERENCES `blogweb`.`user` (`id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
  CONSTRAINT `fk_likes_post1`
    FOREIGN KEY (`post_id`)
      REFERENCES `blogweb`.`post` (`id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Funções identificadas em Sala (e adicionais):

Funções GET (busca e retorna algo):

```
getUserByEmail(email){  
  // realizar a procura de usuário por email e retornar o id de usuário ou  
  // os dados de usuário  
}  
  
getUserByID(userid){  
  // realizar a procura de usuário por ID e retornar os dados de usuário  
}  
  
getAllPosts(limit){  
  // buscar todas as postagens e retornar dentro do limite  
}  
  
getPostByID(postid){  
  // realizar a procura por uma postagem específica (postid) e retornar os  
  // dados desta postagem  
}  
  
getAllPostByUser(userid){  
  // buscar todas as postagens realizadas por um usuário e retornar os  
  // dados da postagem  
}  
  
getPostBySearchContent(string de busca){  
  // realizar uma busca por similaridade da string com o 'corpo' das  
  // postagens  
  // retornar as postagens que se enquadrem  
}  
  
getPostBySearchTitle(string da busca){  
  
  // realizar uma busca por similaridade da string com o 'título' das  
  // postagens  
  // retornar as postagens que se enquadrem  
}  
  
getPostLikedByUser(userid){
```

```
// procurar as postagens curtidas pelo usuário
}

getPostLikes(postid){
// buscar e contar os Likes de uma determinada postagem, retornar a
contagem
}
```

Funções Create/Update (atualiza ou cria registro no banco de dados):

```
createUser(dados do usuario){
// Insere os dados de usuário no banco criando um novo usuário
}

updateUser(dados do usuario){
// atualiza informações do usuário
}

deleteUser(user_id){
// apagar um usuário do banco
}

createPost(dados post, dados usuario){
// Insere um novo post com vínculo ao usuário
}

updatePost(dados post, dados usuario){
// atualiza os dados de um post
}

deletePost(post_id, user_id){
// apagar um post do banco
}

updateLikes(post_id, user_id){
// função dois em 1, se não houver um like adiciona, se houver o like
remove (toggle)
// retorna true ou false (conforme a execução foi bem sucedida)
// ou cria um função para cada e lida com a existência de like no
programa.
}
```