Database Implementation

Bommineni Vishwajith Reddy
EE21BTECH11012

Contents

1 ER-Diagram

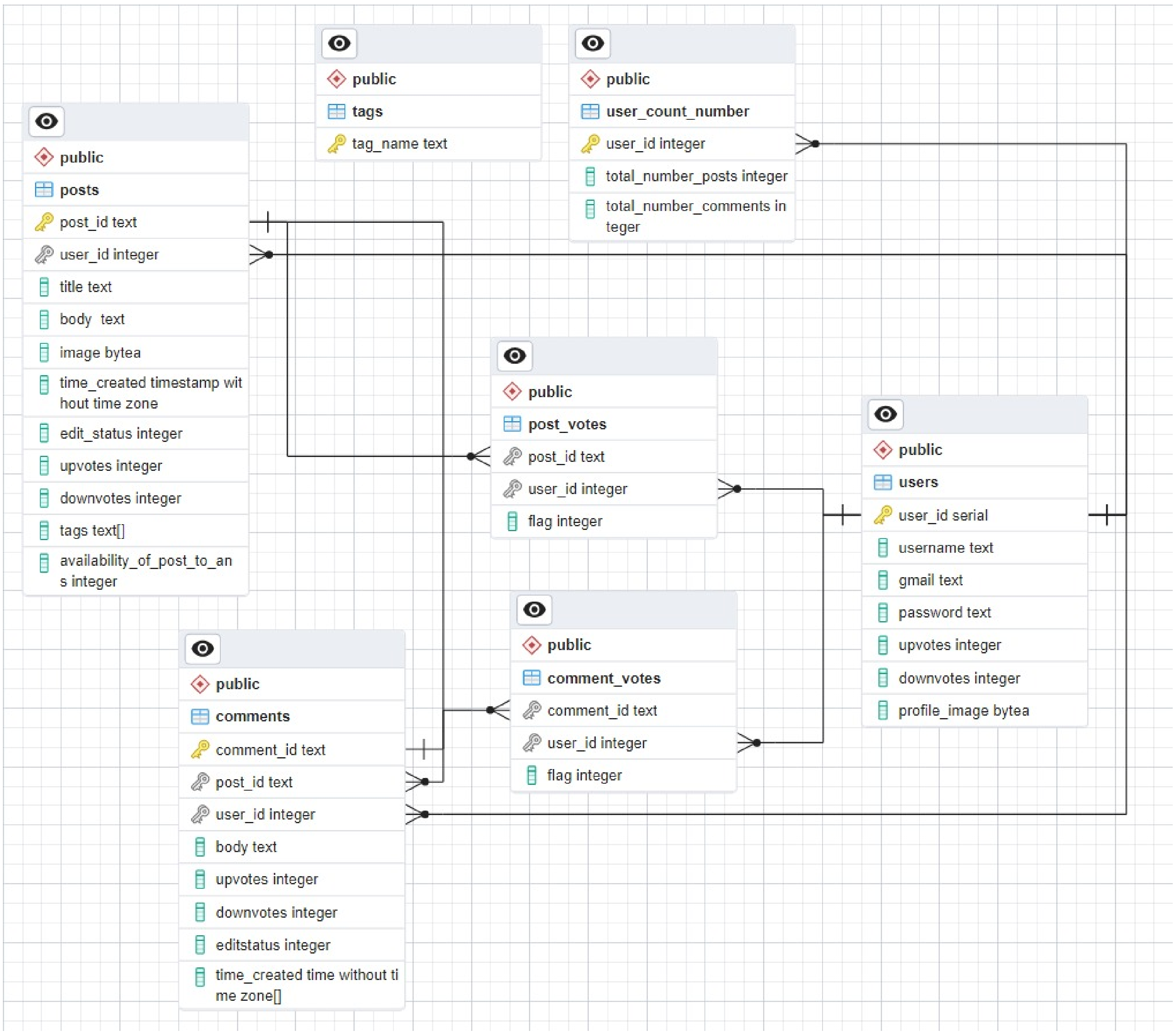

Figure 1: ERD figure

2 Relations

- The DDL of the database can be found in the file **ddl.sql**

2.1 users

2.1.1 Usage

- This table stores the information of users

2.1.2 Attributes

- **user_id:** This is auto generated user identification number.
- **username:** Attribute which stores the username of the user.
- **gmail:** Attribute which stores the e-mail of the user.
- **password:** Attribute which stores the password of the user with **MD5** hashing.
- **upvotes:** Attribute which stores the total upvotes count of the user.
- **downvotes:** Attribute which stores the total downvotes count of the user.
- **profile_image:** Attribute which stores the profile picture of the user.

2.2 posts

2.2.1 Usage

- This table stores the information regarding posts

2.2.2 Attributes

- **post_id:** This is auto generated post identification string.
- **user_id:** Stores the user_id of the user who made the post.
- **title:** Attribute which stores the title of the post.
- **body:** Attribute which stores the body of the post.
- **image:** Attribute which stores the image corresponding to post.
- **time_created:** Attribute which stores the information of time of creation of post.
- **edit_status:** Attribute which stores the info regarding if the post is edited or not.
- **upvotes:** Attribute which stores the upvotes count for the post.
- **downvotes:** Attribute which stores the downvotes count for the post.
- **tag:** Attribute which stores the information regarding tags assigned to the post.
- **availability_of_post_to_ans:** Attribute which stores the information regarding if the post is still open for answering.

2.3 comments

2.3.1 Usage

- This table stores the information regarding comments

2.3.2 Attributes

- **comment_id:** This is auto generated comment identification string.
- **post_id:** Stores the post_id for which post the current comment belongs to.
- **user_id:** Stores the user_id of the user who made the post.
- **body:** Attribute which stores the body of the comment.
- **upvotes:** Attribute which stores the upvotes count for the comment.
- **downvotes:** Attribute which stores the downvotes count for the comment.
- **edit_status:** Attribute which stores the info regarding if the comment is edited or not.
- **time_created:** Attribute which stores the information of time of creation of comment.

2.4 tags

2.4.1 Usage

- This table stores the information regarding tags available in the website to choose for a question.

2.4.2 Attributes

- **tag_name:** Attribute which stores tag name.

2.5 user_count_number

2.5.1 Usage

- This table stores the information regarding activity of the user.

2.5.2 Attributes

- **user_id:** This is the user_id of the user.
- **total_number_posts:** This is the count of total number of posts made by the user.
- **total_number_comments:** This is the count of total comments made by the user.

2.6 post_votes

2.6.1 Usage

- This table stores the information regarding which user upvoted/ downvoted a post.

2.6.2 Attributes

- **post_id:** This is the post_id of the post.
- **user_id:** This is the user_id of person who upvotes or downvoted the post.
- **flag:** Status of the post w.r.t the user with id post_id whether it is upvoted/ downvoted/ no action.

2.7 comment_votes

2.7.1 Usage

- This table stores the information regarding which user upvoted/ downvoted a comment.

2.7.2 Attributes

- **post_id:** This is the comment_id of the comment.
- **user_id:** This is the user_id of person who upvotes or downvoted the comment.
- **flag:** Status of the post w.r.t the user with id comment_id whether it is upvoted/ downvoted/ no action.

3 Functions

- For details regarding functions used in the database, look into the file **functions.sql**

```
   login(user_id_in TEXT, pass TEXT)
```

- The above function **login()**, takes in two arguments **user_id** and **password** for the account and returns:
  - 1 if given credentials are correct
  - 0 if given credentials are incorrect
- This function is called when a user tries to login.

```
 votes_track_post(user_idi int, post_idi TEXT, votess int)
```

- The above function **votes_track_post()**, takes in three arguments **user_id**, **post_id** and **votess** (+1 for upvote, -1 for downvote) for the account and returns:
  - 1 if task is successful
  - 0 if task is unsuccessful
- This function is called when a user upvotes/ downvotes a post.

```
 votes_track_comment(user_idi int, comment_idi TEXT, votess int)
```

- The above function **votes_track_comment**(), takes in three arguments **user_id**, **comment_id** and **votess** (+1 for upvote, -1 for downvote) for the account and returns:
  - 1 if task is successful
  - 0 if task is unsuccessful
- This function is called when a user upvotes/ downvotes a comment.

```
update_password(user_id_in int , pass text)
```

- The above function **update_password()**, takes in two arguments **user_id**, **password** and returns:
  - 1 if task is successful
  - 0 if task is unsuccessful
- This function is called when a user tries to change the account's password.

```
update_upvotes_downvotes_post(post_id_in text, up int, down int)
```

- The above function **update_upvotes_downvotes_post()**, takes in three arguments **post_id**, **up** (default fixed value = +1), **down** (default fixed value = -1) and returns:
  - 1 if task is successful
  - 0 if task is unsuccessful
- This function is called when a user's post is upvoted/ downvoted.

```
  update_upvotes_downvotes_comment(comment_id_in text, up int, down int)
```

- The above function **update_upvotes_downvotes_comment()**, takes in three arguments **comment_id**, **up** (default fixed value = +1), **down** (default fixed value = -1) and returns:
  - 1 if task is successful
  - 0 if task is unsuccessful
- This function is called when a user's comment is upvoted/ downvoted.

```
search_posts_by_user_id(user_id_in int)
```

- The above function **search_posts_by_user_id()**, takes in one argument **user_id** and returns:
  - All the posts made by the user.
- This function is called when a we need to know about the posts made by a user.

```
search_posts_by_tags(tags text[])
```

- The above function **search_posts_by_tags()**, takes in one argument **tags** (An array of tag names) and returns:
  - All the posts which have tags as the given tags.
- This function is called when a we need to know about the posts made by some specific tags.

4 Triggers

4.1 Insert

- For details regarding triggers used in the database, look into the file **insert_triggers.sql**

  ```
      insert_users_1
  ```

  Trigger function: insert_users_table_1()
  Task: Sets default password as username of the user before insertion of new user to **users** table.

  ```
    insert_users_2
  ```

  Trigger function: insert_users_table_2()
  Task: Adds new tuple to **user_count_number** table after insertion of new tuple to **users** table.

  ```
    insert_posts
  ```

  Trigger function: insert_posts_table()
  Task: Checks if given tags assigned to a post are valid or not and updates the user's count of post count by 1 if the post is valid to be posted.

  ```
    insert_comments
  ```

  Trigger function: insert_comments_table()
  Task: Assigns comment_id for the comment and updates the user's count of comment count by 1.

4.2 Update

- For details regarding triggers used in the database, look into the file **update_triggers.sql**

  ```
      update_to_post
  ```

  Trigger function: update_post()
  Task: This trigger is called when a post is edited, based on that flag value is set.

  ```
  update_to_comment
  ```

  Trigger function: update_comment()
  Task: This trigger is called when a comment is edited, based on that flag value is set.

4.3 Delete

- For details regarding triggers used in the database, look into the file **delete_triggers.sql**

  ```
    delete_comments_tab
  ```

  Trigger function: delete_comments_table()
  Task: This trigger is called when a comment is deleted, this resets the total upvotes and downvotes of the user by deleting the contribution of deleted comment.

  ```
  delete_posts_tab
  ```

Trigger function: delete_posts_table()
Task: This trigger is called when a post is deleted, this resets the total upvotes and downvotes of the user by deleting the contribution of deleted post.

- **Note:** Any deletions which are done on **users** table this is handled in other tables by using the **on delete cascade** foerign key constraint.

5 Data Filling

- For filling data in the database, look into the file **data_filling.sql**