Database Management Systems

STUDENT FORUM

An online platform that connects aspiring students, helpful peers and teachers willing to share their knowledge, learn and grow together

Second Year B.Tech Information Technology
Batch A, Group 3

Ganadhish Acharekar

Akshat Shah

Arnav Shah

Saharsh Jain

Acknowledgement

We take this opportunity to thank the Department of Computer Engg & IT, VJTI, for giving us a chance to pursue this project work. Special thanks to Dr. V. B. Nikam and our course instructor, Mayuri More for capacity building in this domain, which helped us to take up this project work to a deliverable level.

Table Of Contents

DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

1. Problem Statement

Overview

Numerous students don't have access to necessary educational resources and a nourishing community to help them learn, grow and develop their skills, knowledge and personality.

Goals

We aim to provide an online platform that:

1. Connects students aspiring to learn, irrespective of their location or background
2. Brings together peers willing to share their knowledge and contribute to building a community
3. Provides a nurturing atmosphere where students can ask queries and answer others' queries

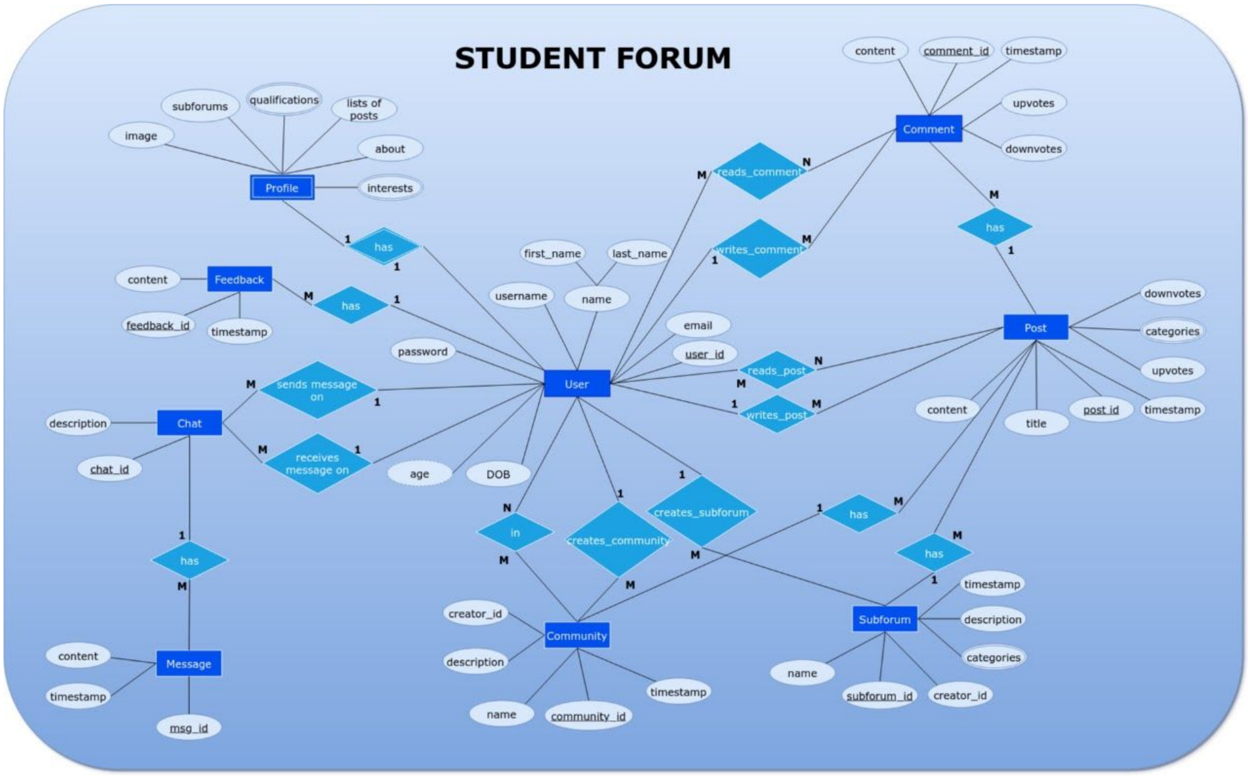
Specifications

1. We propose to create an online forum allowing users to ask their questions, share good resources and answer others' queries in focused subforums in the form of posts. Users can create public or private communities for discussions. Private communities can be made for use cases such as classrooms, schools and institutes.
2. Users can upload files and images of multiple formats related to their posts, which can be accessed by readers. Based on the posts and upvotes, every user will have field-wise ratings to add gamification factor and so that experts in the field can be found.
3. Posts can be upvoted by users based on relevance and authenticity, so users can be presented with the best results. To discourage irrelevant or unruly content, posts can also be downvoted. Hence this will be a community-driven system.
4. To encourage interaction, users will have the option of chatting with other users. Users can ask for items in the subforum and members of the community can help if possible.

DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

2. ER Model


STUDENT FORUM
content
comment_id
timestamp
subforums
qualifications
lists of posts
upvotes
image
Comment
about
downvotes
N
M
reads_comment
Profile
interests
M
has
1
M
has
writes_comment
1
1
first_name
last_name
content
Feedback
M
has
1
username
name
downvotes
feedback_id
timestamp
email
Post
categories
password
user_id
N
reads_post
upvotes
M
M
sends message on
User
M
1
writes_post
description
Chat
1
content
post id
timestamp
title
M
receives message on
1
chat_id
age
DOB
1
1
1
has
M
N
in
creates_community
creates_subforum
M
has
1
has
M
M
M
has
M
1
timestamp
creator_id
Subforum
description
Community
content
Message
description
categories
name
timestamp
timestamp
name
community_id
subforum_id
creator_id
msg_id

DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

3. Revised Problem Statement

Overview

Numerous students don't have access to necessary educational resources and a nourishing community to help them learn, grow and develop their skills, knowledge and personality.

Goals

We aim to provide an online platform that:

1. Connects students aspiring to learn, irrespective of their location or background
2. Brings together peers willing to share their knowledge and contribute to building a community
3. Provides a nurturing atmosphere where students can ask queries and answer others' queries

Specifications

1. We propose to create an online forum allowing users to ask their questions, share good resources and answer others' queries in focused subforums in the form of posts. Users can create public or private communities for discussions. Private communities can be made for use cases such as classrooms, schools and institutes.
2. Users can upload files and images of multiple formats related to their posts, which can be accessed by readers. Based on the posts and upvotes, every user will have field-wise ratings to add gamification factor and so that experts in the field can be found.
3. Posts can be upvoted by users based on relevance and authenticity, so users can be presented with the best results. To discourage irrelevant or unruly content, posts can also be downvoted. Hence this will be a community-driven system.
4. To encourage interaction, users will have the option of chatting with other users. Users can ask for items in the subforum and members of the community can help if possible.

Data to be processed:

User:

- Username
- Email
- Age
- Fields of interest
- Qualifications
- List of chat rooms
- Followed communities
- Followed subforums
- Posts
- Total field-wise upvotes and downvotes

Subforum:

- Name
- Creator
- List of members
- List of categories
- List of Posts

Community:

- Name
- Creator
- List of members
- List of posts

Post:

- Title
- Category
- Subforum
- Author
- Date and time of post
- Content
- Upvotes
- Downvotes
- List of Comments

SOFTWARE TOOLS USED :

For front end design :

- **HTML**
  - Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

- **Bootstrap CSS**
  - Bootstrap is a front-end framework for faster and easier web development. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins. Bootstrap also gives you the ability to easily create responsive designs

- **JavaScript**
  - Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior, and all major web browsers have a dedicated JavaScript engine to execute it.

For back end design :

- **Node.js**
  - Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

- **Express**
  - Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. With a myriad of HTTP utility methods and middleware, creating a robust API is quick and easy.Express provides a thin layer of fundamental web application features, without obscuring any Node.js features.

- **PostgreSQL**
    - Postgres is a free and open-source relational database management system (RDBMS) emphasizing extensibility and SQL compliance. PostgreSQL features transactions with Atomicity, Consistency, Isolation, Durability (ACID) properties, automatically updatable views, materialized views, triggers, foreign keys, and stored procedures. It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users.
- **pgAdmin**
    - PgAdmin is a commonly used database management tool in the PostgresDB community. It simplifies the creation, maintenance, and use of database objects by offering a clean and intuitive user interface
- **SocketIO**
    - Socket.IO is a JavaScript library for realtime web applications. It enables real time, bi-directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for Node.js.

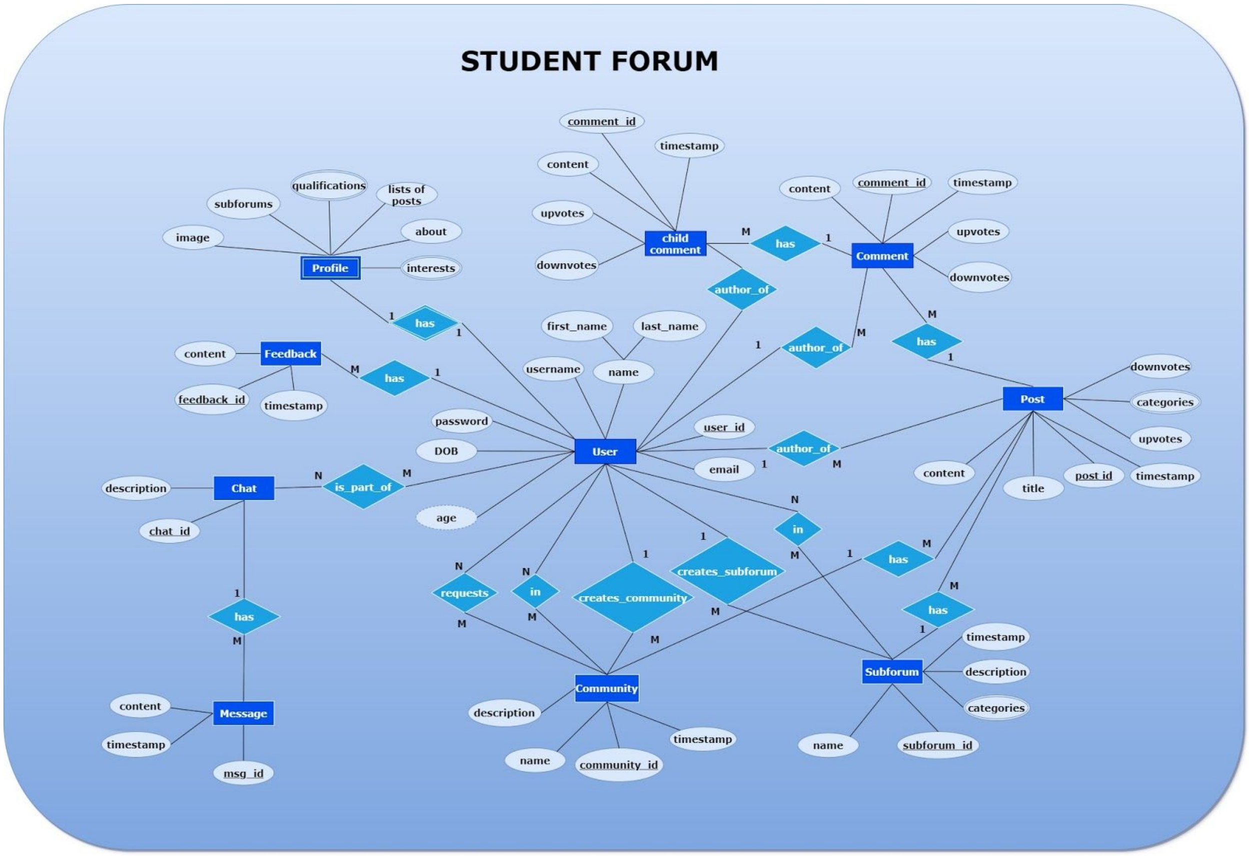
Deployment:

- **Heroku**
    - Heroku is a cloud platform as a service (PaaS) supporting several programming languages. The Heroku network runs the customer's apps in virtual containers which execute on a reliable runtime environment. Heroku calls these containers "Dynos." These Dynos can run code written in Node, Ruby, PHP, Go, Scala, Python, Java, or Clojure.
- **Heroku Postgres**
    - Heroku Postgres is the Cloud database (DBaaS) service for Heroku based on PostgreSQL. Heroku Postgres provides features like continuous protection, rollback, and high availability; also forks, followers, and dataclips.

DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

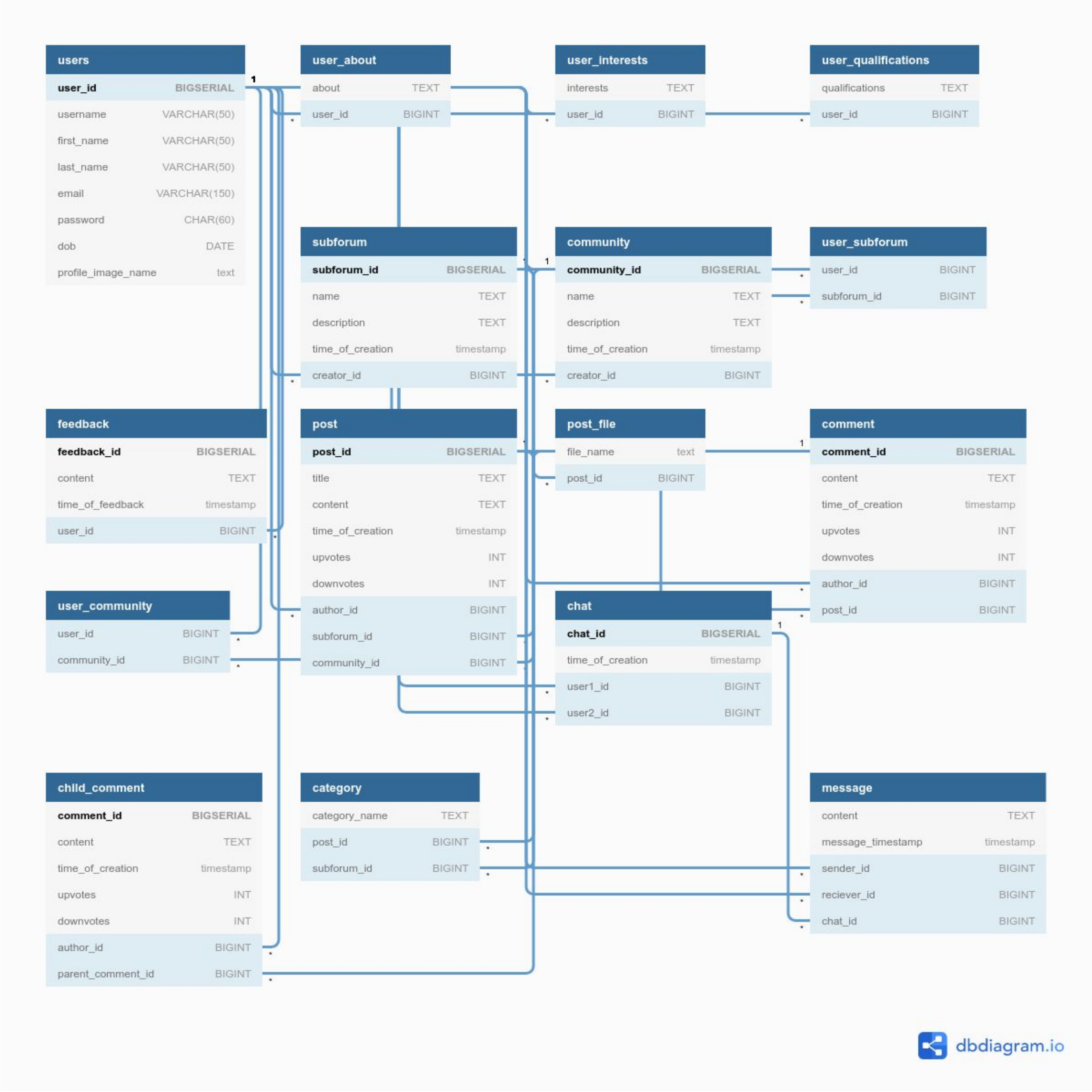
4. Revised ER Model


STUDENT FORUM
comment_id
timestamp
content
upvotes
downvotes
child comment
M
has
1
Comment
content
comment_id
timestamp
upvotes
downvotes
qualifications
lists of posts
subforums
about
image
Profile
interests
author_of
M
has
1
1
has
1
first_name
last_name
author_of
M
content
Feedback
M
has
1
username
name
Post
downvotes
categories
feedback_id
timestamp
password
user_id
upvotes
DOB
User
author_of
email
1
M
content
post_id
timestamp
title
description
Chat
N
is_part_of
M
age
chat_id
N
in
M
1
creates_subforum
1
has
M
1
N
requests
N
in
creates_community
M
has
M
M
M
1
has
M
M
Subforum
timestamp
description
categories
name
subforum_id
Community
description
timestamp
name
community_id
content
Message
timestamp
msg_id

DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

5. UML Diagram (Schema Design)



users	
user_id	BIGSERIAL
username	VARCHAR(50)
first_name	VARCHAR(50)
last_name	VARCHAR(50)
email	VARCHAR(150)
password	CHAR(60)
dob	DATE
profile_image_name	text

user_about	
about	TEXT
user_id	BIGINT

user_interests	
interests	TEXT
user_id	BIGINT

user_qualifications	
qualifications	TEXT
user_id	BIGINT

subforum	
subforum_id	BIGSERIAL
name	TEXT
description	TEXT
time_of_creation	timestamp
creator_id	BIGINT

community	
community_id	BIGSERIAL
name	TEXT
description	TEXT
time_of_creation	timestamp
creator_id	BIGINT

user_subforum	
user_id	BIGINT
subforum_id	BIGINT

feedback	
feedback_id	BIGSERIAL
content	TEXT
time_of_feedback	timestamp
user_id	BIGINT

post	
post_id	BIGSERIAL
title	TEXT
content	TEXT
time_of_creation	timestamp
upvotes	INT
downvotes	INT
author_id	BIGINT
subforum_id	BIGINT
community_id	BIGINT

post_file	
file_name	text
post_id	BIGINT

comment	
comment_id	BIGSERIAL
content	TEXT
time_of_creation	timestamp
upvotes	INT
downvotes	INT
author_id	BIGINT
post_id	BIGINT

user_community	
user_id	BIGINT
community_id	BIGINT

chat	
chat_id	BIGSERIAL
time_of_creation	timestamp
user1_id	BIGINT
user2_id	BIGINT

child_comment	
comment_id	BIGSERIAL
content	TEXT
time_of_creation	timestamp
upvotes	INT
downvotes	INT
author_id	BIGINT
parent_comment_id	BIGINT

category	
category_name	TEXT
post_id	BIGINT
subforum_id	BIGINT

message	
content	TEXT
message_timestamp	timestamp
sender_id	BIGINT
reciever_id	BIGINT
chat_id	BIGINT

DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

6. Data Dictionary for the Schema

Table users

Column	Data Type	Reference	Not Null	Auto-Increment
user_id	bigint		YES	YES
username	Character varying(50)		YES	NO
first_name	Character varying(50)		YES	NO
last_name	Character varying(50)		YES	NO
email	Character varying(150)		YES	NO
password	character(50)		YES	NO
dob	date		YES	NO

Unique keys

Type	Table	Key	Column
primary	users	pk_users	user_id
unique	users	uk_email	email
unique	users	uk_username	username

Table - subforum

Column	Data Type	Reference	Not Null	Auto-Increment
subforum_id	bigint		YES	YES
name	text		YES	NO
description	text		NO	NO
time_of_creation	timestamp without timezone		NO	NO
creator_id	bigint	users	NO	NO

Unique keys

Type	Table	Key	Column
primary	subforum	pk_subforum_id	subforum_id
unique	subforum	uk_name	name

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_creator_id	subforum	creator_id	users	user_id

Table - community

Column	Data Type	Reference	Not Null	Auto-Increment
community_id	bigint		YES	YES
name	text		YES	NO
description	text		NO	NO
time_of_creation	timestamp without timezone		NO	NO
creator_id	bigint	users	NO	NO

Unique keys

Type	Table	Key	Column
primary	community	pk_community_id	community_id
unique	community	uk_name	name

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_creator_id	community	creator_id	users	user_id

Table - user_subforum

Column	Data Type	Reference	Not Null	Auto-Increment
user_id	bigint	users	NO	NO
subforum_id	bigint	subforum	NO	NO

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_user_id	user_subforum	user_id	users	user_id
fk_subforum_id	user_subforum	subforum_id	subforum	subforum_id

Table - user_qualification

Column	Data Type	Reference	Not Null	Auto-Increment
user_id	bigint	users	NO	NO
qualification	text		NO	NO

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_user_id	user_qualification	user_id	users	user_id

Table - user_interest

Column	Data Type	Reference	Not Null	Auto-Increment
user_id	bigint	users	NO	NO
interest	text		NO	NO

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_user_id	user_interest	user_id	users	user_id

Table - user_community

Column	Data Type	Reference	Not Null	Auto-Increment
user_id	bigint	users	NO	NO
community_id	bigint	community	NO	NO

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_user_id	user_community	user_id	users	user_id
fk_community_id	user_community	community_id	community	community_id

Table - user_about

Column	Data Type	Reference	Not Null	Auto-Increment
user_id	bigint	users	NO	NO
about	text		NO	NO

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_user_id	user_about	user_id	users	user_id

Table - post

Column	Data Type	Reference	Not Null	Auto-Increment
post_id	bigint		YES	YES
title	text		YES	NO
content	text		YES	NO
time_of_creation	timestamp without timezone		NO	NO
upvotes	integer		NO	NO
downvotes	integer		NO	NO
author_id	bigint	users	NO	NO
subforum_id	bigint	subforum	NO	NO
community_id	bigint	community	NO	NO

Unique keys

Type	Table	Key	Column
primary	post	pk_post_id	post_id

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_author_id	post	author_id	users	user_id
fk_subforum_id	post	subforum_id	subforum	subforum_id
fk_community_id	post	community_id	community	community_id

Table - post_file

Column	Data Type	Reference	Not Null	Auto-Increment
post_id	bigint	post	NO	NO
file_name	text		YES	NO

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_post_id	post_file	post_id	post	post_id

Table - category

Column	Data Type	Reference	Not Null	Auto-Increment
category_name	text		YES	NO
post_id	bigint	post	NO	NO
subforum_id	bigint	subforum	NO	NO

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_post_id	category	post_id	post	post_id
fk_subforum_id	category	subforum_id	subforum	subforum_id

Table - chat

Column	Data Type	Reference	Not Null	Auto-Increment
chat_id	bigint		YES	YES
time_of_creation	timestamp without timezone		NO	NO
user1	Character varying(50)	users	NO	NO
user2	Character varying(50)	users	NO	NO

Unique keys

Type	Table	Key	Column
primary	chat	pk_chat_id	chat_id

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_user1	chat	user1	users	username
fk_user2	chat	user2	users	username

Table - comment

Column	Data Type	Reference	Not Null	Auto-Increment
comment_id	bigint		YES	YES
content	text		YES	NO
time_of_creation	timestamp without timezone		NO	NO
upvotes	integer		NO	NO
downvotes	integer		NO	NO
author_id	bigint	users	NO	NO
post_id	bigint	post	NO	NO

Unique keys

Type	Table	Key	Column
primary	comment	pk_comment_id	comment_id

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_author_id	comment	author_id	users	user_id
fk_post_id	comment	post_id	post	post_id

Table - child_comment

Column	Data Type	Reference	Not Null	Auto-Increment
comment_id	bigint		YES	YES
content	text		YES	NO
time_of_creation	timestamp without timezone		NO	NO
upvotes	integer		NO	NO
downvotes	integer		NO	NO
author_id	bigint	users	NO	NO
parent_comment_id	bigint	comment	NO	NO

Unique keys

Type	Table	Key	Column
primary	child_comment	pk_comment_id	comment_id

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_author_id	child_comment	author_id	users	user_id
fk_parent_comment_id	child_comment	parent_comment_id	comment	comment_id

Table - message

Column	Data Type	Reference	Not Null	Auto-Increment
message_id	bigint		YES	YES
content	text		YES	NO
message_timestamp	timestamp without timezone		NO	NO
sender	Character varying(50)	users	NO	NO
receiver	Character varying(50)	users	NO	NO
chat_id	bigint	chat	NO	NO

Unique keys

Type	Table	Key	Column
primary	message	pk_message_id	message_id

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_sender	message	sender	users	username
fk_reciever	message	receiver	users	username
fk_chat_id	message	chat_id	chat	chat_id

Table - feedback

Column	Data Type	Reference	Not Null	Auto-Increment
feedback_id	bigint		YES	YES
content	text		YES	NO
time_of_feedback	timestamp without timezone		NO	NO
user_id	bigint	users	NO	NO

Unique keys

Type	Table	Key	Column
primary	feedback	pk_feedback_id	feedback_id

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_user_id	feedback	user_id	users	user_id

Table - pending_requests

Column	Data Type	Reference	Not Null	Auto-Increment
user_id	bigint	users	NO	NO
community_id	bigint	community	NO	NO

Foreign keys

Key	Table	Column	Ref table	Ref column
fk_user_id	pending_requests	user_id	users	user_id
fk_community_id	pending_requests	community_id	community	community_id

DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

7. Data Definition Language (DDL) Queries

TABLE DEFINITIONS:

1. **users**

```
CREATE TABLE users (
      user_id BIGSERIAL NOT NULL PRIMARY KEY,
      username VARCHAR(50) NOT NULL,
      first_name VARCHAR(50) NOT NULL,
      last_name VARCHAR(50) NOT NULL,
      email VARCHAR(150) NOT NULL,
      password CHAR(60) NOT NULL,
      dob DATE NOT NULL,
      profile_image_name text,
      UNIQUE(email),
      UNIQUE (username)
);
```

2. **user_about**

```
CREATE TABLE user_about(
      about TEXT,
      user_id BIGINT REFERENCES users(user_id)
);
```

3. **user_interest**

```
CREATE TABLE user_interest (
      interest TEXT,
      user_id BIGINT REFERENCES users(user_id)
);
```

4. **user_qualification**

```
CREATE TABLE user_interest (
      interest TEXT,
      user_id BIGINT REFERENCES users(user_id)
);
```

5. feedback

```
CREATE TABLE feedback (
      feedback_id BIGSERIAL NOT NULL PRIMARY KEY,
      content TEXT NOT NULL,
      time_of_feedback TIMESTAMP,
      user_id BIGINT REFERENCES users(user_id)
);
```

6. subforum

```
CREATE TABLE subforum (
      subforum_id BIGSERIAL NOT NULL PRIMARY KEY,
      name TEXT NOT NULL,
      description TEXT,
      time_of_creation TIMESTAMP,
      creator_id BIGINT REFERENCES users(user_id),
      UNIQUE(name)
);
```

7. user_subforum

```
CREATE TABLE user_subforum (
      user_id BIGINT REFERENCES users(user_id),
      subforum_id BIGINT REFERENCES subforum(subforum_id)
);
```

8. community

```
CREATE TABLE community (
      community_id BIGSERIAL NOT NULL PRIMARY KEY,
      name TEXT NOT NULL,
      description TEXT,
      time_of_creation TIMESTAMP,
      creator_id BIGINT REFERENCES users(user_id),
      UNIQUE(name)
);
```

9. user_community

```
CREATE TABLE user_community (
      user_id BIGINT REFERENCES users(user_id),
      community_id BIGINT REFERENCES community(community_id)
);
```

10. pending_requests

```
CREATE TABLE pending_requests (
      user_id BIGINT REFERENCES users(user_id),
      community_id BIGINT REFERENCES community(community_id)
);
```

11. post

```
CREATE TABLE post (
      post_id BIGSERIAL NOT NULL PRIMARY KEY,
      title TEXT NOT NULL,
      content TEXT NOT NULL,
      time_of_creation TIMESTAMP,
      upvotes INT DEFAULT 0,
      downvotes INT DEFAULT 0,
      author_id BIGINT REFERENCES users(user_id),
      subforum_id BIGINT REFERENCES subforum(subforum_id),
      community_id BIGINT REFERENCES community(community_id)
);
```

12. post_file

```
CREATE TABLE post_file (
      file_name text NOT NULL,
      post_id BIGINT REFERENCES post(post_id)
);
```

13. comment

```
CREATE TABLE comment (
      comment_id BIGSERIAL NOT NULL PRIMARY KEY,
      content TEXT NOT NULL,
      time_of_creation TIMESTAMP,
      upvotes INT DEFAULT 0,
      downvotes INT DEFAULT 0,
      author_id BIGINT REFERENCES users(user_id),
      post_id BIGINT REFERENCES post(post_id)
);
```

14. child_comment

```
CREATE TABLE child_comment (
      comment_id BIGSERIAL NOT NULL PRIMARY KEY,
      content TEXT NOT NULL,
      time_of_creation TIMESTAMP,
      upvotes INT DEFAULT 0,
      downvotes INT DEFAULT 0,
      author_id BIGINT REFERENCES users(user_id),
      parent_comment_id BIGINT REFERENCES comment(comment_id) DEFAULT
NULL
);
```

15. category

```
CREATE TABLE category (
      category_name TEXT NOT NULL,
      post_id BIGINT REFERENCES post(post_id),
      subforum_id BIGINT REFERENCES subforum(subforum_id)
);
```

16. chat

```
CREATE TABLE chat (
      chat_id BIGSERIAL NOT NULL PRIMARY KEY,
      time_of_creation TIMESTAMP,
      user1 VARCHAR(50) NOT NULL REFERENCES users(username),
      user2 VARCHAR(50) NOT NULL REFERENCES users(username)
);
```

17. message

```
CREATE TABLE message (
      message_id BIGSERIAL NOT NULL PRIMARY KEY,
      content TEXT NOT NULL,
      message_timestamp TIMESTAMP,
      sender VARCHAR(50) REFERENCES users(username),
      receiver VARCHAR(50) REFERENCES users(username),
      chat_id BIGINT REFERENCES chat(chat_id));
```

DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

8. Data Manipulation Language (DML) Queries

1. Related to users

a. Adding users (registration)

```
INSERT INTO users
      (username,first_name,last_name,email,password,dob)
VALUES
      ('ram123', 'ram', 'shah', 'ram@ayod.com',
'qwerty123', '2000-01-01'),
      ('rahul123', 'rahul', 'shah', 'rahul@ddlj.com',
'qwerty123', '2000-02-01'),
      ('karan123', 'karan', 'shah', 'karan@arj.com',
'qwerty123', '2000-03-01');
```

b. Adding 'about' of users

```
INSERT INTO user_about
      (about,user_id)
VALUES
      ('Eager to learn, willing to help peers', 1);
```

c. Adding interests of a user

```
INSERT INTO user_interest
      (interest,user_id)
VALUES
      ('reading', 1),
      ('writing', 1),
      ('geography', 1);
```

d. Adding qualifications of a user

```
INSERT INTO user_qualification
      (qualification,user_id)
VALUES
      ('BCOM', 1),
      ('BAF', 1);
```

e. Retrieve all the data of a user

```
SELECT user_id, username, first_name, last_name, email,
dob, profile_image_name FROM users
WHERE username = 'rahul123';
```

f. Retrieve the ‘about’ of a user

```
SELECT about FROM user_about
WHERE user_id =1;
```

g. Retrieve the interests of a user

```
SELECT interest from user_interest
WHERE user_id = 1;
```

h. Retrieve the qualifications of a user

```
SELECT interest from user_interest
WHERE user_id = 1;
```

i. Retrieve all the posts made by a user in reverse chronological order

```
SELECT * FROM post
WHERE author_id = 1
ORDER BY time_of_creation DESC;
```

j. Retrieve all the posts made by a user except the ones part of a community

```
SELECT * FROM post
WHERE author_id = 1 AND community_id IS NULL
ORDER BY time_of_creation DESC;
```

k. Retrieve the users a user has a chat with (retrieve list of chats)

```
SELECT user1, user2 FROM chat
WHERE user1='rahul123' OR user2='rahul123';
```

l. Search query for users related to ‘data analyst’

```
SELECT username, first_name, last_name, email, dob,
profile_image_name FROM users
WHERE to_tsvector(username) @@ to_tsquery('data analyst')
OR to_tsvector(first_name) @@ to_tsquery('data analyst')
OR to_tsvector(last_name) @@ to_tsquery('data analyst')
OR to_tsvector(email) @@ to_tsquery('data analyst');
```

2. Related to subforums

a. Creating a subforum

```
INSERT INTO subforum
(name,description,time_of_creation,creator_id)
VALUES
     ('DBMS', 'A subforum for database professionals,
students and enthusiasts', CURRENT_TIMESTAMP, 1);
```

b. Adding categories of a subforum

```
INSERT INTO category
     (category_name, subforum_id)
     (SELECT 'DBMS', subforum_id
     FROM subforum
     WHERE name = 'Computer Science' OR name='database'
     );
```

c. Retrieve data about a subforum

```
SELECT * FROM subforum
WHERE subforum_id = 1;
```

d. Retrieve the username of the creator of a subforum

```
SELECT username FROM users
WHERE user_id =
(SELECT creator_id FROM subforum
WHERE subforum_id=1);
```

e. Retrieve the categories of a subforum

```
SELECT category_name FROM category
WHERE subforum_id = 1;
```

f. Retrieve the subforum id using subforum name

```
SELECT subforum_id FROM subforum
WHERE name = 'DBMS';
```

g. Retrieve the posts in a subforum in reverse chronological order

```
SELECT * FROM post
WHERE subforum_id = 1
ORDER BY subforum_id DESC;
```

h. Search query for subforums related to database, in reverse chronological order of creation

```
SELECT * FROM subforum
WHERE to_tsvector(name) @@ to_tsquery('database')
OR to_tsvector(description) @@ to_tsquery('database')
OR subforum_id IN
(SELECT subforum_id FROM category
WHERE to_tsvector(category_name) @@ to_tsquery('database')
AND subforum_id IS NOT NULL)
ORDER BY time_of_creation DESC;
```

3. Related to communities

a. Creating a community

```
INSERT INTO community
(name,description,time_of_creation,creator_id)
VALUES
     ('VJTI', 'A community for students of Veermata
Jijabai Technological Institute', CURRENT_TIMESTAMP, 2);
```

b. Retrieve data about a community

```
SELECT * FROM community
WHERE community_id = 1;
```

c. Retrieve the username of the creator of a community

```
SELECT username FROM users
WHERE user_id =
(SELECT creator_id FROM community
WHERE community_id=1);
```

d. Retrieve the posts in a subforum in reverse chronological order

```
SELECT * FROM post
WHERE community_id = 1
ORDER BY time_of_creation DESC;
```

e. Search query for communities related to ‘python’, in reverse chronological order of creation

```
SELECT * FROM community
WHERE to_tsvector(name) @@ to_tsquery('python')
OR to_tsvector(description) @@ to_tsquery('python')
ORDER BY time_of_creation DESC;
```

4. Related to posts

a. Creating a post

```
INSERT INTO post
(title,content,time_of_creation,author_id,subforum_id)
VALUES
     (Why managing a database efficiently is essential',
'<content>', CURRENT_TIMESTAMP, 1, 1)
RETURNING post_id;
```

b. Adding categories of the post

```
INSERT INTO category
     (category_name,post_id)
     'new category 2', 1;
```

c. Retrieve the data of a post

```
SELECT * FROM post
WHERE post_id = 1;
```

d. Retrieve the username of the author of a post

```
SELECT username FROM users
WHERE user_id =
(SELECT author_id FROM post
WHERE post_id=1);
```

e. Retrieve the name of the subforum a post belongs to

```
SELECT name FROM subforum
WHERE subforum_id =
(SELECT subforum_id FROM post
WHERE post_id=1);
```

f. Retrieve the categories of a post

```
SELECT category_name FROM category
WHERE post_id = 1;
```

g. Retrieve the comments on a post

```
SELECT * FROM comment
WHERE post_id = 1
ORDER BY time_of_creation DESC;
```

h. Retrieve the files associated with a post

```
SELECT file_name FROM post_file
WHERE post_id = 1;
```

i. Search query for posts related to ‘PostgreSQL’, with results ordered by upvotes

```
SELECT * FROM post
WHERE (to_tsvector(title) @@ to_tsquery('PostgreSQL')
OR to_tsvector(content) @@ to_tsquery('PostgreSQL'))
OR post_id IN
(SELECT post_id FROM category
WHERE to_tsvector(category_name) @@
to_tsquery('PostgreSQL')
AND post_id IS NOT NULL)
AND community_id IS NULL
ORDER BY upvotes DESC;
```

5. Related to comments

a. Adding a comment to a post

```
INSERT INTO comment
     (content,time_of_creation,author_id,post_id)
VALUES
     ('Great post! But I have a doubt...',
CURRENT_TIMESTAMP, 1, 1);
```

b. Adding a comment to a comment (child comment)

```
INSERT INTO child_comment
(content,time_of_creation,author_id,post_id,
parent_comment_id)
VALUES
     ('I'll send you a message on chat',
CURRENT_TIMESTAMP, 1, 1, 1);
```

c. Retrieving the comments on a post

```
SELECT * FROM comment
WHERE post_id = 1
ORDER BY time_of_creation DESC;
```

d. Retrieving the child comments of a comment

```
SELECT * FROM child_comment
WHERE parent_comment_id = 1;
```

6. Related to chats

a. Create a chat between two users

```
INSERT INTO chat (user1, user2, time_of_creation)
VALUES(1, 2, CURRENT_TIMESTAMP)
RETURNING chat_id;
```

b. Insert a message in a chat

```
INSERT INTO message
(content, sender, receiver, message_timestamp, chat_id)
VALUES ('Hi!', 'ram123', 'rahul123', CURRENT_TIMESTAMP, 1)
RETURNING message_timestamp;
```

c. Retrieve chats of a user

```
SELECT user1, user2 FROM chat
WHERE user1='ram123' OR user2='ram123';
```

d. Retrieve messages in a chat in reverse chronological order

```
SELECT * FROM message
WHERE chat_id=2
ORDER BY message_id DESC;
```

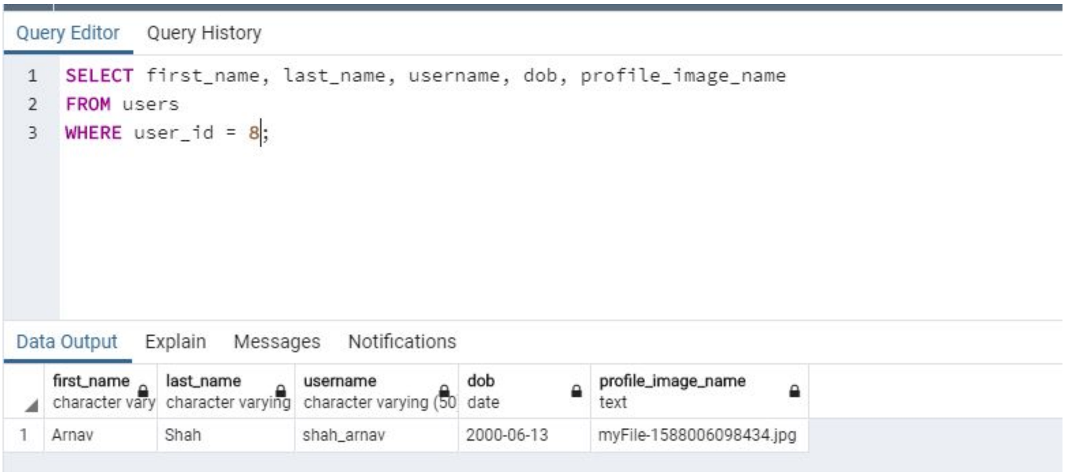
DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

9. SELECT Queries

Queries related to users:

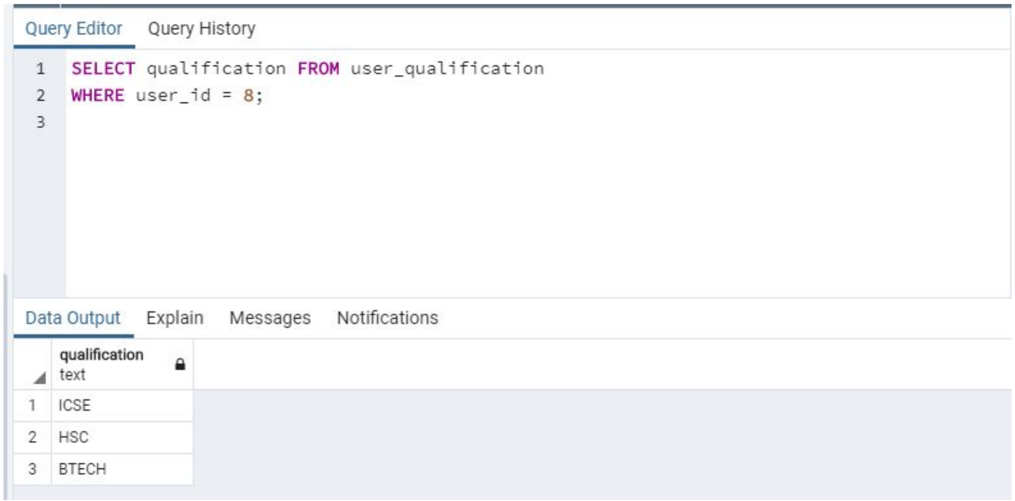
1. View profile of a user which includes name, username, age and profile image.



```
SELECT first_name, last_name, username, dob, profile_image_name
FROM users
WHERE user_id = 8;
```

	first_name character vary	last_name character varying	username character varying (50	dob date	profile_image_name text
1	Arnav	Shah	shah_arnav	2000-06-13	myFile-1588006098434.jpg

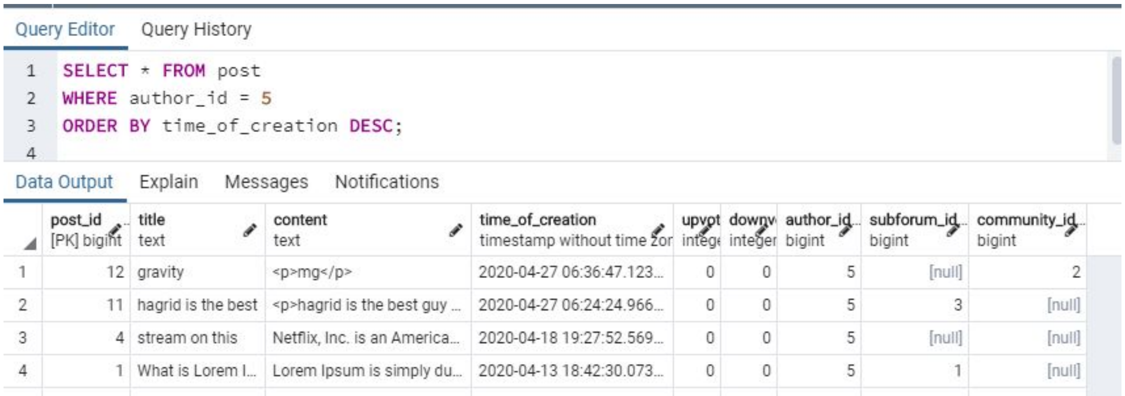
2. View the about section of the user.
3. View feedback given by various users.
4. View interests of a user.
5. View qualifications of a user.



```
SELECT qualification FROM user_qualification
WHERE user_id = 8;
```

	qualification text
1	ICSE
2	HSC
3	BTECH

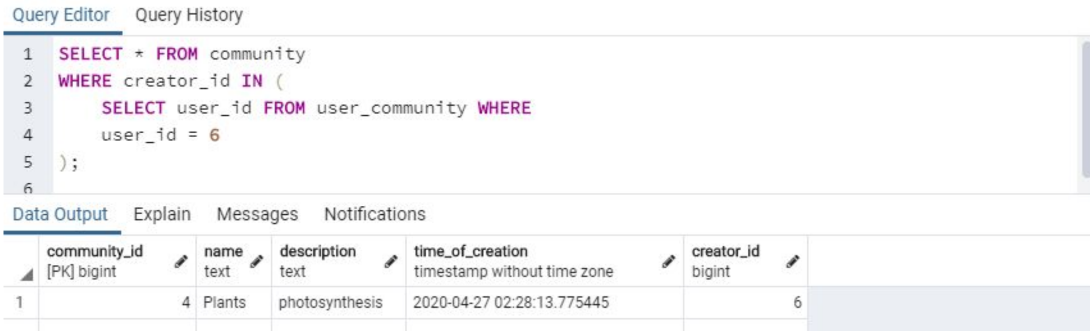
6. View posts made by a user ordered by timestamp.



```
SELECT * FROM post
WHERE author_id = 5
ORDER BY time_of_creation DESC;
```

	post_id [PK] bigint	title text	content text	time_of_creation timestamp without time zor	upvot intege	downv integer	author_id... bigint	subforum_id... bigint	community_id... bigint
1	12	gravity	<p>mg</p>	2020-04-27 06:36:47.123...	0	0	5	[null]	2
2	11	hagrid is the best	<p>hagrid is the best guy ...	2020-04-27 06:24:24.966...	0	0	5	3	[null]
3	4	stream on this	Netflix, Inc. is an America...	2020-04-18 19:27:52.569...	0	0	5	[null]	[null]
4	1	What is Lorem I...	Lorem Ipsum is simply du...	2020-04-13 18:42:30.073...	0	0	5	1	[null]

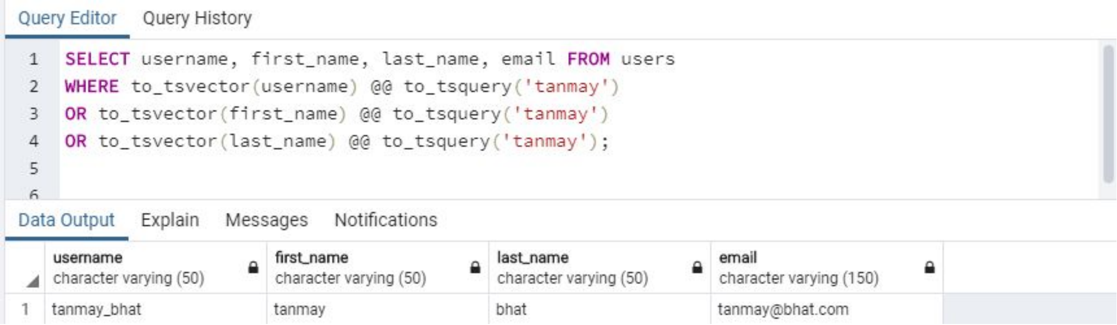
7. View posts made by a user ordered by upvotes.
8. View posts made by a user ordered by downvotes.
9. View subforums created by a user.
10. View subforums followed by a user.
11. View communities created by a user.
12. View communities followed by a user.



```
SELECT * FROM community
WHERE creator_id IN (
    SELECT user_id FROM user_community WHERE
    user_id = 6
);
```

	community_id [PK] bigint	name text	description text	time_of_creation timestamp without time zone	creator_id bigint
1	4	Plants	photosynthesis	2020-04-27 02:28:13.775445	6

13. View pending requests of other users to follow a community.
14. View profile of other users that you have searched for.
15. Search on the basis of username, first name and last name of a user.



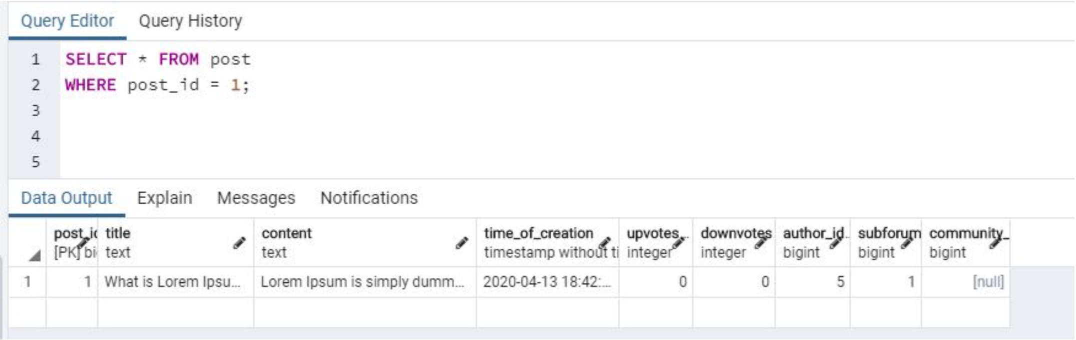
```sql
SELECT username, first_name, last_name, email FROM users
WHERE to_tsvector(username) @@ to_tsquery('tanmay')
OR to_tsvector(first_name) @@ to_tsquery('tanmay')
OR to_tsvector(last_name) @@ to_tsquery('tanmay');
```

	username character varying (50)	first_name character varying (50)	last_name character varying (50)	email character varying (150)
1	tanmay_bhat	tanmay	bhat	tanmay@bhat.com

16. Search on the basis of email of a user.

Queries related to posts:

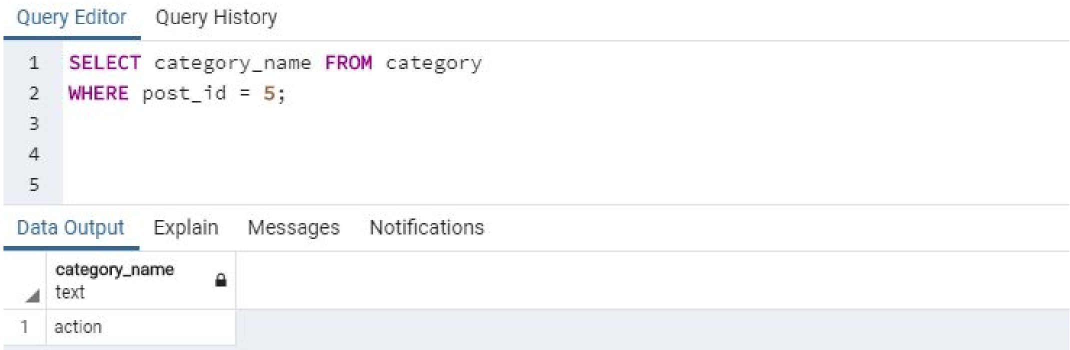
1. View a post, which consists of post title, username of author, time of creation.



```sql
SELECT * FROM post
WHERE post_id = 1;
```

	post_ic [PK] bi	title text	content text	time_of_creation timestamp without ti	upvotes_ integer	downvotes integer	author_id bigint	subforum bigint	community_ bigint
1	1	What is Lorem Ipsu...	Lorem Ipsum is simply dumm...	2020-04-13 18:42:...	0	0	5	1	[null]

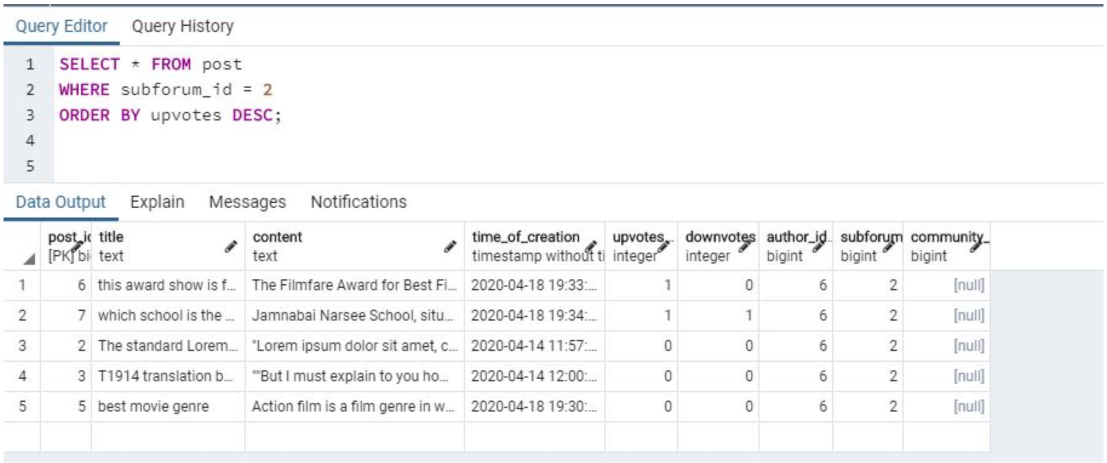
2. View profile image of the author of each post.
3. View HTML rich content of each post which may include images, extra markup etc.
4. Access files uploaded by a user on a post(.pdf, .txt, .png etc.).
5. View categories of each post.



```sql
SELECT category_name FROM category
WHERE post_id = 5;
```

	category_name text
1	action

6. View upvotes of each post.
7. View downvotes of each post.
8. View all posts in a particular subforum ordered by timestamp.
9. View all posts in a particular subforum ordered by upvotes.

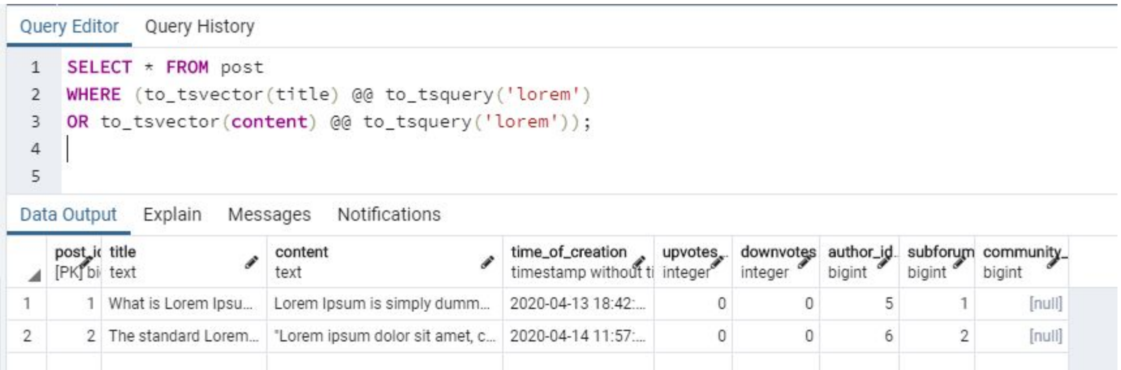


```sql
SELECT * FROM post
WHERE subforum_id = 2
ORDER BY upvotes DESC;
```

	post_ic [PK] bi	title text	content text	time_of_creation timestamp without ti	upvotes_ integer	downvotes integer	author_id bigint	subforum bigint	community_ bigint
1	6	this award show is f...	The Filmfare Award for Best Fi...	2020-04-18 19:33:...	1	0	6	2	[null]
2	7	which school is the ...	Jamnabai Narsee School, situ...	2020-04-18 19:34:...	1	1	6	2	[null]
3	2	The standard Lorem...	"Lorem ipsum dolor sit amet, c...	2020-04-14 11:57:...	0	0	6	2	[null]
4	3	T1914 translation b...	""But I must explain to you ho...	2020-04-14 12:00:...	0	0	6	2	[null]
5	5	best movie genre	Action film is a film genre in w...	2020-04-18 19:30:...	0	0	6	2	[null]

10. View all posts in a particular subforum ordered by downvotes.
11. View all posts in a particular community ordered by timestamp.
12. View all posts in a particular community ordered by upvotes.
13. View all posts in a particular community ordered by downvotes.
14. View posts on the home page ordered by upvotes.
15. View posts on the homepage ordered by downvotes.
16. View posts on the home page based on the user's interests.
17. View posts on the home page based on the user's qualifications.
18. View posts made by a user.
19. View posts that you have searched for.

20. Search on the basis of title and content of a post.



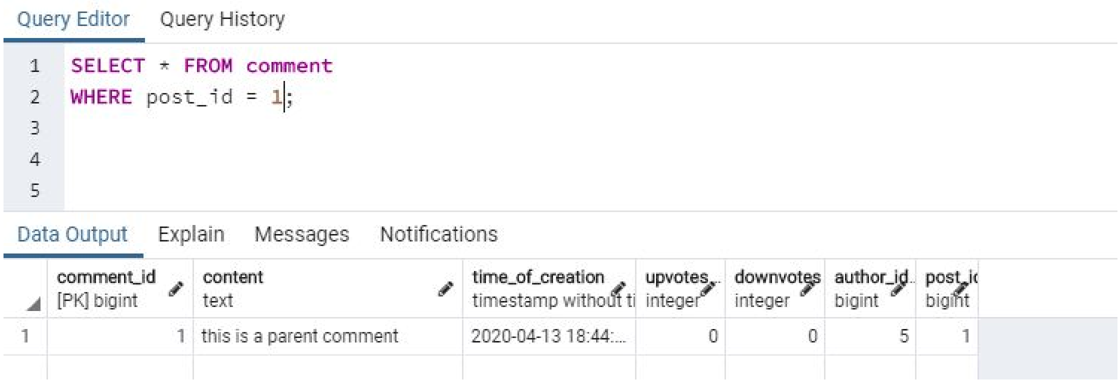
```sql
SELECT * FROM post
WHERE (to_tsvector(title) @@ to_tsquery('lorem')
OR to_tsvector(content) @@ to_tsquery('lorem'));
```

	post_ic [PK] bi	title text	content text	time_of_creation timestamp without ti	upvotes_ integer	downvotes integer	author_id_ bigint	subforum bigint	community_ bigint
1	1	What is Lorem Ipsu...	Lorem Ipsum is simply dumm...	2020-04-13 18:42:...	0	0	5	1	[null]
2	2	The standard Lorem...	"Lorem ipsum dolor sit amet, c...	2020-04-14 11:57:...	0	0	6	2	[null]

21. Search on the basis of categories of a post.

Queries related to comments:

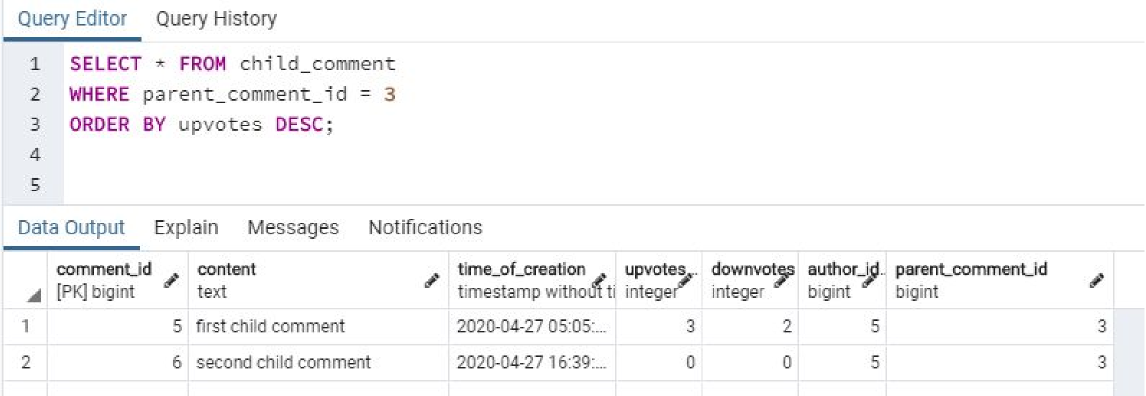
1. View comments on posts where different comments are from different users.



```sql
SELECT * FROM comment
WHERE post_id = 1;
```

	comment_id [PK] bigint	content text	time_of_creation timestamp without ti	upvotes_ integer	downvotes integer	author_id_ bigint	post_ic bigint
1	1	this is a parent comment	2020-04-13 18:44:...	0	0	5	1

2. View child comments (i.e. comments on primary comment).
3. View upvotes of each comment/child comment.
4. View downvotes of each comment/child comment.
5. View comments/child comments on a post ordered by timestamp.
6. View comments/child comments on a post ordered by upvotes.



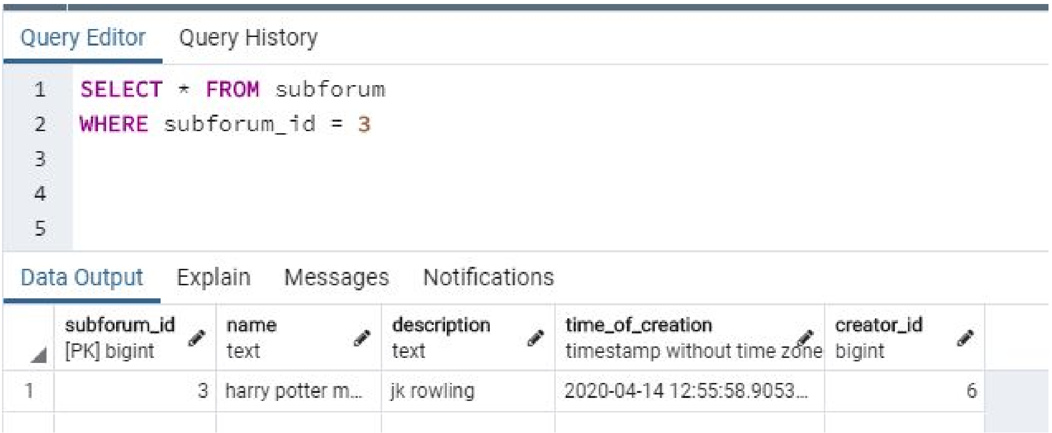
```sql
SELECT * FROM child_comment
WHERE parent_comment_id = 3
ORDER BY upvotes DESC;
```

	comment_id [PK] bigint	content text	time_of_creation timestamp without ti	upvotes integer	downvotes integer	author_id bigint	parent_comment_id bigint
1	5	first child comment	2020-04-27 05:05:...	3	2	5	3
2	6	second child comment	2020-04-27 16:39:...	0	0	5	3

7. View comments/child comments on a post ordered by downvotes.
8. View username and profile image of creator of each comment/child comment.

Queries related to subforums:

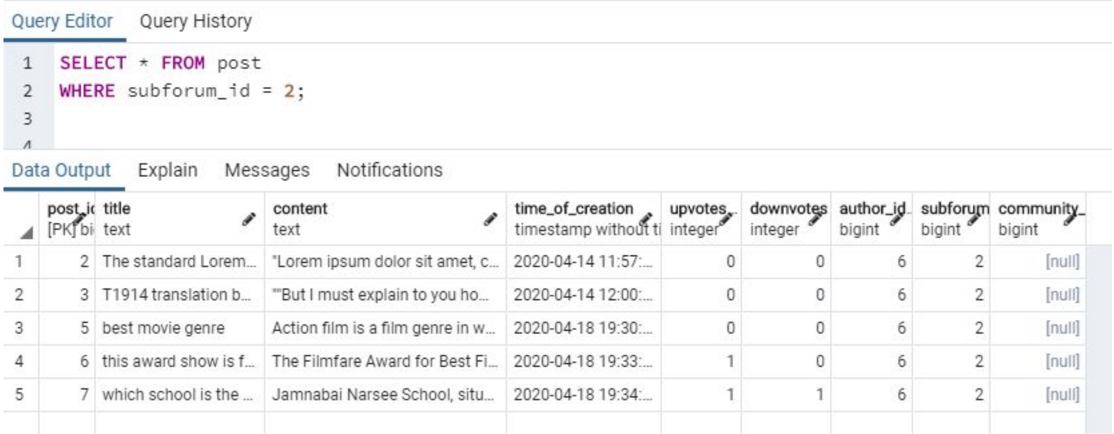
1. View a subforum, which consists of the name of the subforum, username of its creator and time of creation.



```sql
SELECT * FROM subforum
WHERE subforum_id = 3
```

	subforum_id [PK] bigint	name text	description text	time_of_creation timestamp without time zone	creator_id bigint
1	3	harry potter m...	jk rowling	2020-04-14 12:55:58.9053...	6

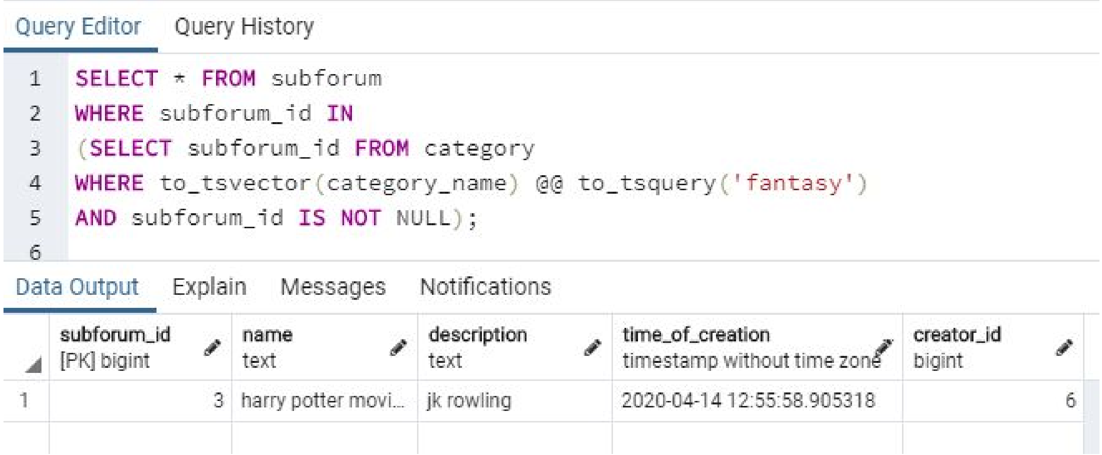
2. View subforums created by a particular user.
3. View subforums followed by a particular user
4. View all posts in a particular subforum.



```
SELECT * FROM post
WHERE subforum_id = 2;
```

	post_ic [PK] bi	title text	content text	time_of_creation timestamp without ti	upvotes_ integer	downvotes integer	author_id bigint	subforum bigint	community_ bigint
1	2	The standard Lorem...	"Lorem ipsum dolor sit amet, c...	2020-04-14 11:57:...	0	0	6	2	[null]
2	3	T1914 translation b...	""But I must explain to you ho...	2020-04-14 12:00:...	0	0	6	2	[null]
3	5	best movie genre	Action film is a film genre in w...	2020-04-18 19:30:...	0	0	6	2	[null]
4	6	this award show is f...	The Filmfare Award for Best Fi...	2020-04-18 19:33:...	1	0	6	2	[null]
5	7	which school is the ...	Jamnabai Narsee School, situ...	2020-04-18 19:34:...	1	1	6	2	[null]

5. View categories of each subforum.
6. View subforums on the home page based on the user's interests.
7. View subforums on the home page based on the user's qualifications.
8. View subforums that you have searched for.
9. Search on the basis of name and description of a subforum.
10. Search on the basis of categories of a subforum.



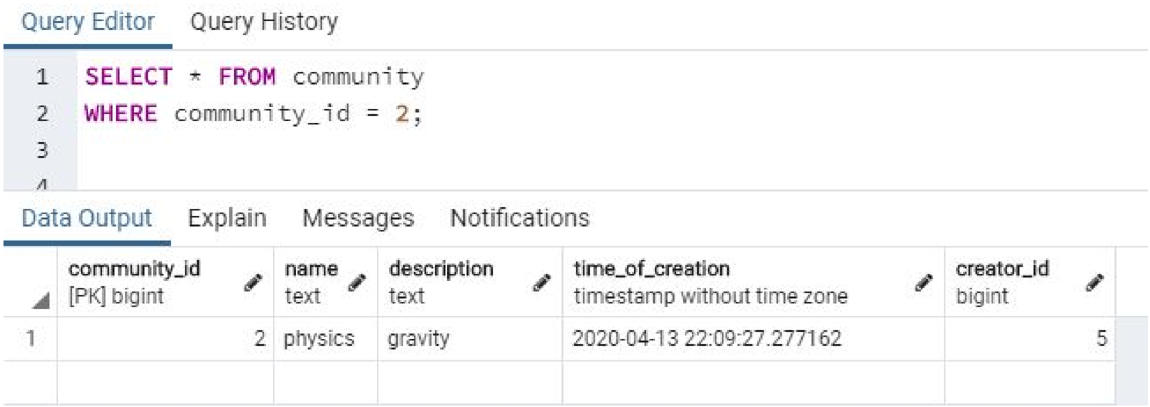
```
SELECT * FROM subforum
WHERE subforum_id IN
(SELECT subforum_id FROM category
WHERE to_tsvector(category_name) @@ to_tsquery('fantasy')
AND subforum_id IS NOT NULL);
```

	subforum_id [PK] bigint	name text	description text	time_of_creation timestamp without time zone	creator_id bigint
1	3	harry potter movi...	jk rowling	2020-04-14 12:55:58.905318	6

11. See if you are following a particular subforum or not.
12. View which users are following a subforum that you have created.

Queries related to communities:

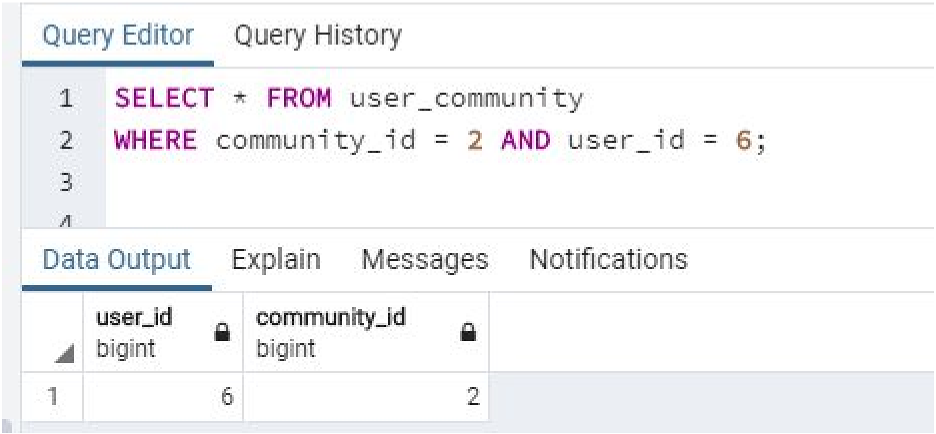
1. View a community created by a user, which consists of the name of the community, username of its creator and time of creation only if you are a member of the community.



```
SELECT * FROM community
WHERE community_id = 2;
```

	community_id [PK] bigint	name text	description text	time_of_creation timestamp without time zone	creator_id bigint
1	2	physics	gravity	2020-04-13 22:09:27.277162	5

2. View all posts in a particular community only if you are a member of the community.
3. View communities that you have searched for.
4. Search on the basis of name and description of a community.
5. See If you are following a particular community or not.



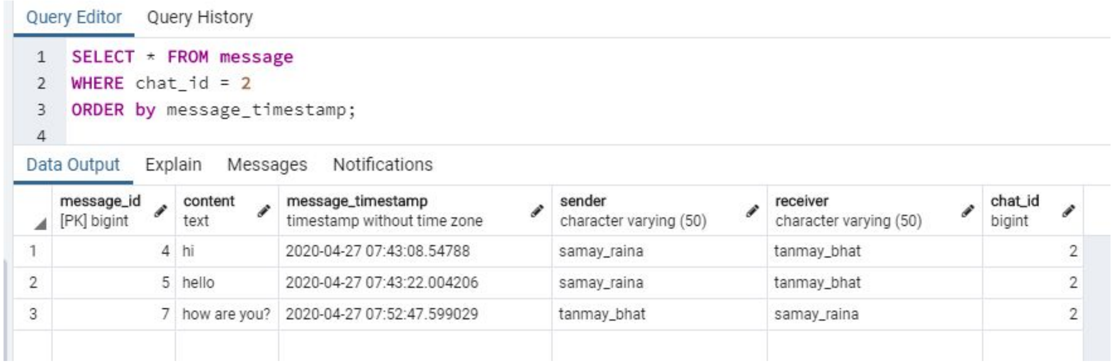
```
SELECT * FROM user_community
WHERE community_id = 2 AND user_id = 6;
```

	user_id bigint	community_id bigint
1	6	2

6. View which users are following a community that you have created.

Queries related to chatting:

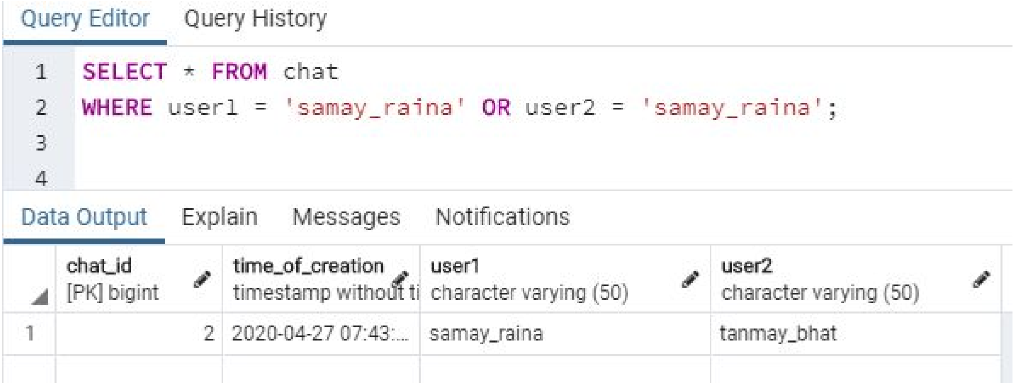
1. Load all the messages in a chat ordered by time.



```
SELECT * FROM message
WHERE chat_id = 2
ORDER by message_timestamp;
```

	message_id [PK] bigint	content text	message_timestamp timestamp without time zone	sender character varying (50)	receiver character varying (50)	chat_id bigint
1	4	hi	2020-04-27 07:43:08.54788	samay_raina	tanmay_bhat	2
2	5	hello	2020-04-27 07:43:22.004206	samay_raina	tanmay_bhat	2
3	7	how are you?	2020-04-27 07:52:47.599029	tanmay_bhat	samay_raina	2

2. Display sent messages.
3. Display received messages.
4. View the users you are chatting with which includes the username and profile image of that user on the sidebar.



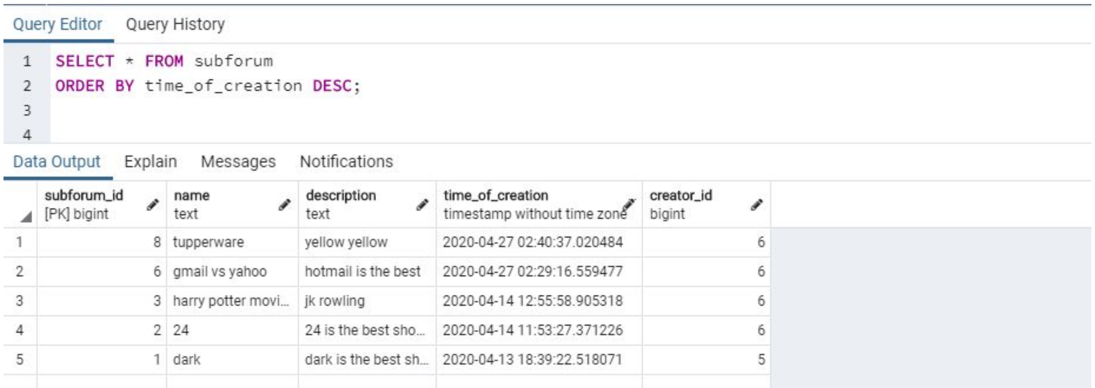
```
SELECT * FROM chat
WHERE user1 = 'samay_raina' OR user2 = 'samay_raina';
```

	chat_id [PK] bigint	time_of_creation timestamp without ti	user1 character varying (50)	user2 character varying (50)
1	2	2020-04-27 07:43:...	samay_raina	tanmay_bhat

5. Select any user and chat with them in the chat window.

Miscellaneous:

1. If no user is logged in, view posts on the home page based on time of creation of posts(i.e. latest displayed first).
2. If no user is logged in, view subforums on the home page based on time of creation of subforums(i.e. latest displayed first).



```
SELECT * FROM subforum
ORDER BY time_of_creation DESC;
```

	subforum_id [PK] bigint	name text	description text	time_of_creation timestamp without time zone	creator_id bigint
1	8	tupperware	yellow yellow	2020-04-27 02:40:37.020484	6
2	6	gmail vs yahoo	hotmail is the best	2020-04-27 02:29:16.559477	6
3	3	harry potter movi...	jk rowling	2020-04-14 12:55:58.905318	6
4	2	24	24 is the best sho...	2020-04-14 11:53:27.371226	6
5	1	dark	dark is the best sh...	2020-04-13 18:39:22.518071	5

DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

10. Triggers and Procedures

Triggers:

1. When a post is inserted into the post table, its corresponding category is inserted in the category table.

```
CREATE TRIGGER category_for_post ON post
    AFTER INSERT ON post
    FOR EACH ROW
    BEGIN
        INSERT INTO category (category_name, post_id)
        SELECT 'category_name', post_id FROM INSERTED
    END;
```

2. When a subforum is inserted into the subforum table, its corresponding category is inserted in the category table.

```
CREATE TRIGGER category_for_subforum ON subforum
    AFTER INSERT ON subforum
    FOR EACH ROW
    BEGIN
        INSERT INTO category (category_name, subforum_id )
        SELECT 'category_name', subforum_id FROM INSERTED
    END;
```

Procedures:

1. Delete post requires multiple delete statements to be executed as post_id acts as a foreign key for other tables. Thus, to satisfy the foreign key constraint, we use stored procedures to execute multiple queries.

```
CREATE PROCEDURE delete_post (IN p_id INT)
   BEGIN
    AS
        DELETE FROM comment WHERE post_id = p_id;
        DELETE FROM category WHERE post_id = p_id;
        DELETE FROM post_file WHERE post_id = p_id;
        DELETE FROM post WHERE post_id = p_id;
    END;
```

2. Delete subforum requires multiple delete statements to be executed as subforum_id acts as a foreign key for other tables. Thus, to satisfy the foreign key constraint, we use stored procedures to execute multiple queries.

```
CREATE PROCEDURE delete_subforum (IN s_id INT)
    BEGIN
    AS
        DELETE FROM category WHERE subforum_id = s_id;
        DELETE FROM user_subforum WHERE subforum_id = s_id;
        DELETE FROM subforum WHERE subforum_id = p_id;
    END;
```

3. Delete community requires multiple delete statements to be executed as community_id acts as a foreign key for other tables. Thus, to satisfy the foreign key constraint, we use stored procedures to execute multiple queries.

```
CREATE PROCEDURE delete_community (IN c_id INT)
   BEGIN
    AS
        DELETE FROM user_community  WHERE community_id = c_id ;
        DELETE FROM community WHERE community_id = c_id ;
    END;
```

DATABASE MANAGEMENT SYSTEMS

STUDENT FORUM

11. Front End Design

- **Register:**
    - Users can register by providing their first and last name, a unique username, email and password. The user also needs to input his/her interests, qualifications and 'about'.
- **Login:**
    - Users once registered, can login using their respective email and password. Only a logged in user can create posts, subforums and communities.
- **Home page:**
    - The home page displays all the posts and subforums ordered by the logged in user's interests, qualifications and description. This ensures that all posts and subforums relevant to the current user are displayed first for a better user experience. If user is not logged in, then home page displays all the posts and subforums in decreasing order of their time of creation(i.e. Latest posts and subforums are displayed first) .
- **Profile:**
    - Users can view their or anyone else's profile page. It consists of the respective user's first and last name, username, email, profile image, interests, qualifications and 'about'. The profile also consists of that user's recent posts, created and followed subforums and communities.
- **Create post:**
    - Each post should consist of a title, content, categories and any additional files that may be attached by the user. An editor is provided which provides extra functionality to write your post.

This includes fonts, colors, text formatting options etc. Custom HTML can also be posted.

- **View post**
  - Users can view any post by simply clicking on it. The display page consists of the title, content, categories, author details and associated comments. Users can also comment on any post. Upvote and downvote functionality is provided so that relevant posts(or posts with more upvotes) can be shown to users first.
- **Create subforum**
  - Users can create subforums which represent a collection of related posts. They are open to all users. Each subforum consists of a name, description and categories. Posts can then be added to the subforum.
- **Create community**
  - Users can create communities which represent a collection of related posts. However, unlike subforums, they are only open to select users, which are called members of that community. Each community consists of a name and description. Posts can then be added to the community by any member.
- **View subforum**
  - All subforums are open and can thus be viewed by any user. The display page consists of the name, description and associated posts.
- **View community**
  - A community can only be viewed by its members. However, users can request the creator of a particular community to be added to that community.

- **Chat page**
  - The chat page consists of a chat window with an input box and an output screen to display received messages. It consists of a sidebar which lists all the users that are chatting with the current user. The current user can select any user from the sidebar and chat with him/her.
- **Search**
  - Users can search for a particular post, subforum, community or user by typing the necessary keyword/s in the search bar. For the search functionality, each word that is being searched against the inputted keyword is given a score or weightage that is used to determine the result. Common words such as ‘a’, ‘the’, ‘and’ etc. are given a lower score so that they don't interfere in the search results as any result can have these words.

Project Link

A live demonstration is at https://student-forum-saga.herokuapp.com/

Login credentials (sample user):

Email:	rameshsoni@mail.in
Password:	password

Github link: https://github.com/ganadhish1999/SAGA

THANK YOU