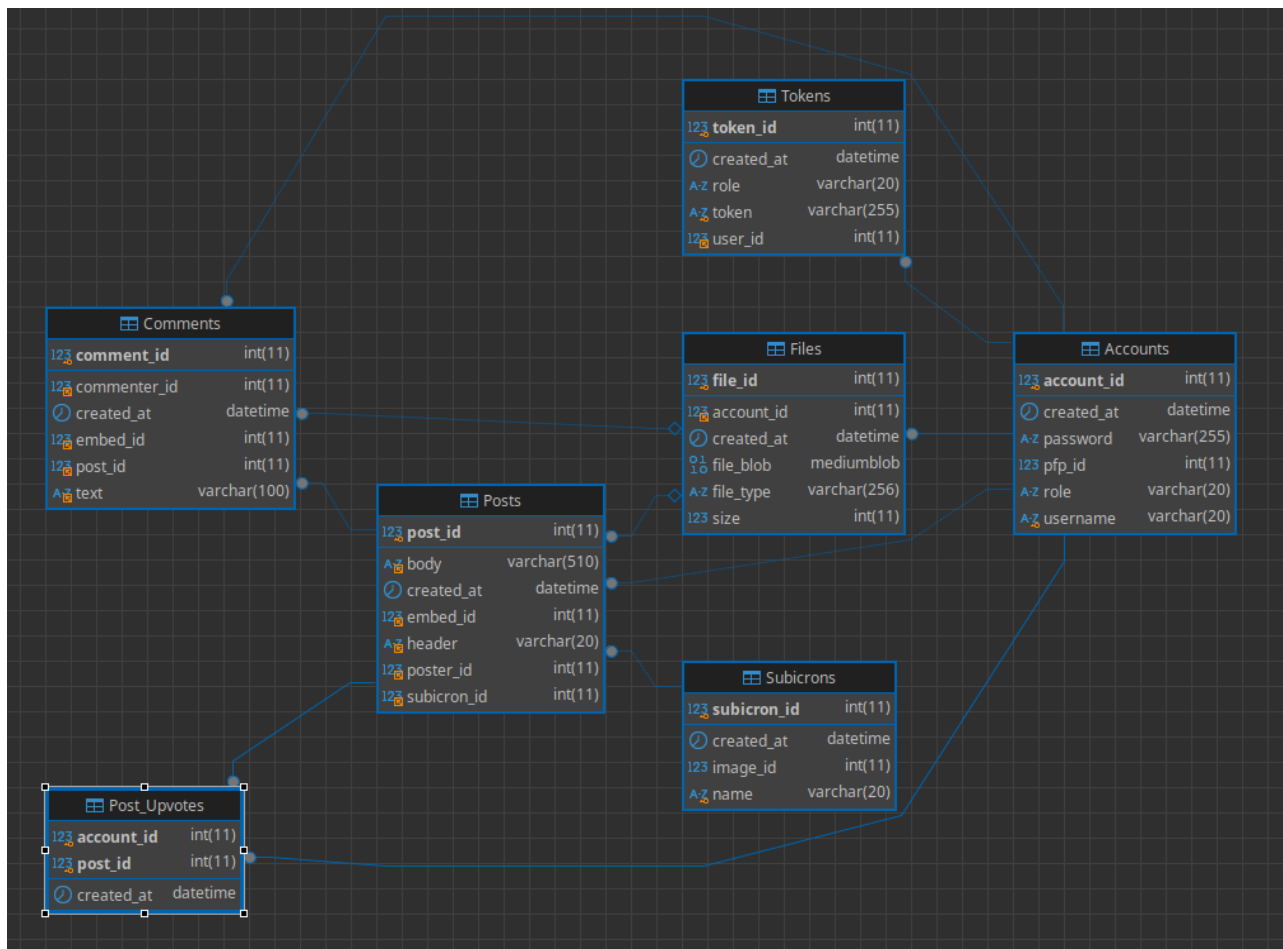


1. Въведение

Този проект представлява Full Stack проект, реализиран с MySQL, Rust, Actix и Yew. Проекта съдържа frontend, backend и база данни с 7 таблици с 8 релации. Базата данни е разработена в съответствие с изискванията на Национална програма "Обучение за ИТ умения и кариера".

2. ER диаграма



3. Описание на таблиците

Accounts

Таблица, която съхранява информация за потребителите.

- **account_id** (INT, PRIMARY KEY, AUTO_INCREMENT) – Уникален идентификатор
- **pfp_id** (INT) – Идентификатор на профилна снимка
- **username** (VARCHAR(20), UNIQUE) – Уникално потребителско име
- **password** (VARCHAR(255)) – Парола (хеширана)
- **role** (VARCHAR(20)) – Роля на потребителя
- **created_at** (DATETIME) – Дата на създаване

Files

Съхранява файлове, качени от потребителите.

- **file_id** (INT, PRIMARY KEY, AUTO_INCREMENT)
- **file_blob** (MEDIUMBLOB) – Съдържание на файла
- **size** (INT) – Размер на файла
- **file_type** (VARCHAR(256)) – Тип на файла
- **account_id** (INT, FOREIGN KEY) – Свързан потребител
- **created_at** (DATETIME) – Дата на качване

Tokens

Съдържа токени за удостоверяване на потребители (Many to one).

- **token_id** (INT, PRIMARY KEY, AUTO_INCREMENT)
- **user_id** (INT, FOREIGN KEY) – Свързан потребител
- **token** (VARCHAR(255), UNIQUE) – Токен за достъп
- **role** (VARCHAR(20)) – Роля на потребителя
- **created_at** (DATETIME) – Дата на създаване

Subicrons

Категории или групи за публикации.

- **subicron_id** (INT, PRIMARY KEY, AUTO_INCREMENT)
- **image_id** (INT) – Идентификатор на изображение
- **name** (VARCHAR(20), UNIQUE) – Име на групата
- **created_at** (DATETIME) – Дата на създаване

Posts

Съхранява публикации, създадени от потребители (Many to one).

- **post_id** (INT, PRIMARY KEY, AUTO_INCREMENT)
- **header** (VARCHAR(20)) – Заглавие на публикацията
- **body** (VARCHAR(510)) – Съдържание на публикацията
- **embed_id** (INT, FOREIGN KEY) – Прикачен файл
- **poster_id** (INT, FOREIGN KEY) – Автор на публикацията
- **subicron_id** (INT, FOREIGN KEY) – Категория
- **created_at** (DATETIME) – Дата на създаване

Comments

Съдържа коментари към публикации (Many to one).

- **comment_id** (INT, PRIMARY KEY, AUTO_INCREMENT)
- **text** (VARCHAR(100)) – Съдържание на коментара
- **embed_id** (INT, FOREIGN KEY) – Прикачен файл
- **commenter_id** (INT, FOREIGN KEY) – Автор на коментара
- **post_id** (INT, FOREIGN KEY) – Свързана публикация
- **created_at** (DATETIME) – Дата на създаване

Post_Upvotes

Регистрира гласове за публикации (Many to many).

- **post_id** (INT, FOREIGN KEY)
- **account_id** (INT, FOREIGN KEY)
- **created_at** (DATETIME)
- **PRIMARY KEY** (account_id, post_id)

4. SQL заявки

Някои примерни SQL заявки:

1. Създаване на нова публикация:

```
INSERT INTO Posts (header, body, embed_id, poster_id, subicron_id)
VALUES(?, ?, ?, ?, ?);
```

2. Добавяне на коментар:

```
INSERT INTO Comments (text, embed_id, commenter_id, post_id) VALUES(?, ?, ?, ?);
```

3. Изтриване на глас за публикация:

```
DELETE FROM Post_Upvotes WHERE account_id = ? AND post_id = ?;
```

4. Извличане на публикации в дадена група:

```
SELECT Posts.*, Accounts.username AS poster_username FROM Posts INNER JOIN
Accounts ON Posts.poster_id = Accounts.account_id WHERE Posts.subicron_id = ?
ORDER BY created_at DESC LIMIT 10;
```

5. Извличане на коментари за публикация:

```
SELECT Comments.comment_id, Comments.text, Comments.embed_id,
Comments.commenter_id, Comments.created_at, Accounts.username AS
commenter_username FROM Comments INNER JOIN Accounts ON Comments.commenter_id =
Accounts.account_id WHERE post_id = ? ORDER BY created_at DESC LIMIT ?;
```

5. Заключение

Проектът реализира функционална база данни с връзки между потребители, публикации и коментари. Създадени са примерни записи и SQL заявки за управление на данните.

<https://github.com/simeonnv/Omicron>

<https://epsilonsv.duckdns.org:8080>