

CIC0097 - Banco de Dados: Projeto
Tema: Rede Social

Guilherme da Rocha Cunha - 221030007

Arthur Diehl Barroso - 221029991

Eduardo Quirino de Oliveira - 211010305

Breno Costa Avelino Lima - 211010280

2024.1



UnB

Sumário

1	Introdução	3
2	Diagrama de Entidade Relacionamento	3
2.1	Entidades	3
2.1.1	User	3
2.1.2	Post	4
2.1.3	Group	4
2.1.4	Chat	4
2.2	Relacionamentos	4
3	Modelo Relacional	5
4	Consultas em Álgebra Relacional	5
4.1	Grupos nos quais um determinado usuário faz parte	5
4.2	Quantidade de likes ou comentários que um usuário deu em determinado grupo	6
4.3	Recomendação de usuários ou grupos	6
5	Avaliação das formas normais	6
5.1	Primeira Forma Normal	6
5.2	Segunda Forma Normal	7
5.3	Terceira Forma Normal	7
6	Diagrama da Camada de Mapeamento	7
7	Referências	7

1 Introdução

O projeto consiste na aplicação dos conhecimentos adquiridos na disciplina "CIC0097 - Banco de Dados". O banco de dados construído, com o tema Rede Social, utilizou como mecanismo de armazenamento o SQLite e o programa com as funções CRUD foi-se implementado na linguagem Python utilizando o framework web Flask.

2 Diagrama de Entidade Relacionamento

O diagrama abaixo representa o modelo entidade relacionamento do projeto.

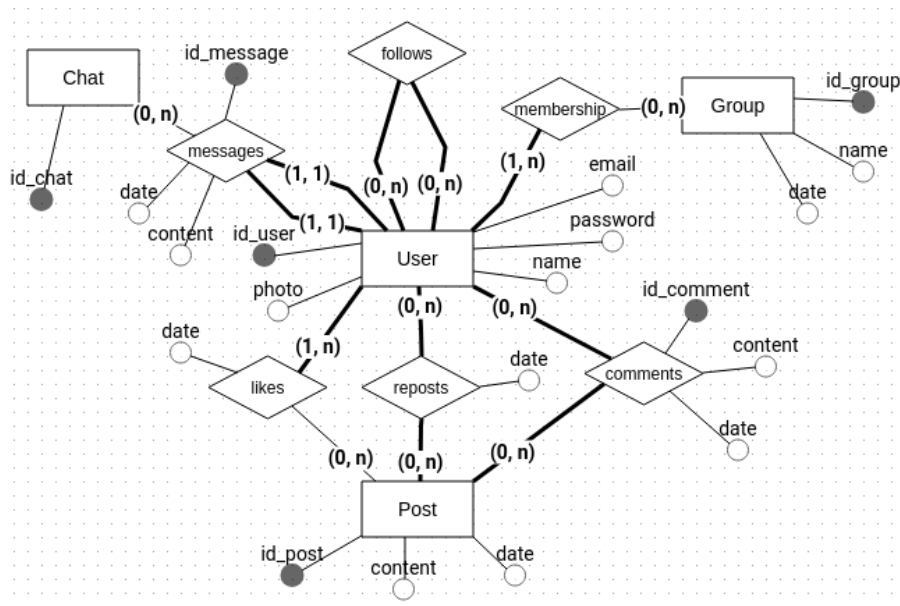


Figure 1: Modelo Entidade Relacional

2.1 Entidades

2.1.1 User

Definição: Representa um usuário no sistema.

- Atributo chave: id_user.
- Atributos simples: email, password, name e photo.

2.1.2 Post

Definição: Uma publicação feita por um usuário.

- Atributo chave: `id_post`.
- Atributos simples: `content` e `date`.

2.1.3 Group

Definição: Um grupo composto de usuários, similar à grupos do Facebook, comunidades do Orkut e dentre outros.

- Atributo chave: `id_group`.
- Atributos simples: `name` e `content`.

2.1.4 Chat

Definição: Local de conversa entre dois usuários.

- Atributo chave: `id_chat`.

2.2 Relacionamentos

As entidades se relacionam de 6 maneiras distintas:

1. Likes: um usuário pode curtir n publicações e uma publicação pode ser curtida por n usuários. Post tem participação total na relação, isto é, toda publicação é feita por algum usuário, mas nem todo usuário faz uma publicação (participação parcial).
2. Reposts: um usuário pode repostar n publicações e uma publicação pode ser repostada por n usuários. Ambas as entidades têm participação parcial no diagrama.
3. Comments: um usuário pode fazer n comentários em uma publicação e uma publicação pode ter n usuários comentando. Cada comentário possui sua própria identificação.
4. Follows: relacionamento recursivo onde um usuário pode seguir n usuários.
5. Membership: um usuário pode fazer parte de n grupos e um grupo pode ter n membros, onde cada grupo tem que ter pelo menos um usuário que faz parte dele.
6. Messages: relacionamento ternário entre dois usuários e um chat, onde um usuário pode ter n conversas com outros usuários e um chat pertence a um par de usuários.

3 Modelo Relacional

Abaixo encontra-se o modelo relacional gerado a partir do modelo entidade relacionamento, utilizando os tipos de dados do SQLite.

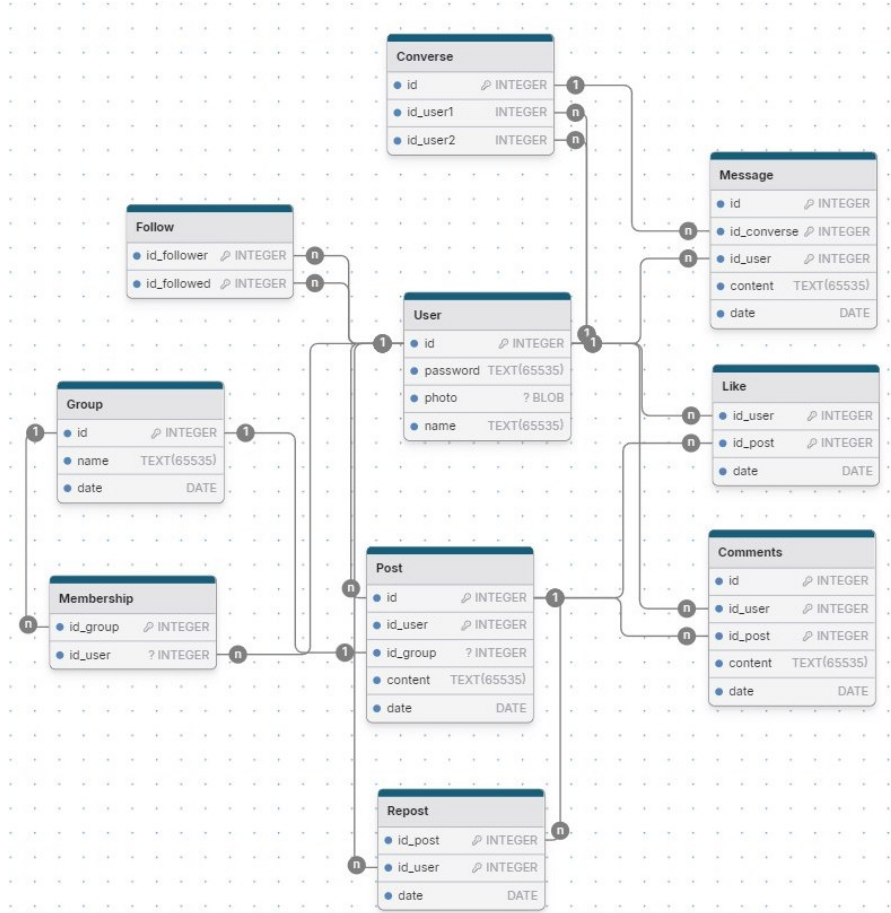


Figure 2: Modelo Relacional

4 Consultas em Álgebra Relacional

4.1 Grupos nos quais um determinado usuário faz parte

Seja um usuário u , queremos saber quais grupos ele faz parte. Para isso, definimos a função f da seguinte forma:

$$f : User \rightarrow Group$$

$$u \rightarrow Group \bowtie_{id=id_group} (\sigma_{id_user==u.id}(Membership)),$$

onde é feito uma seleção na tabela **Membership** nos registros nos quais possuem o mesmo **id** que o usuário u e em seguida uma junção com a tabela **Group**, sendo o atributo de ligação o **group_id**.

4.2 Quantidade de likes ou comentários que um usuário deu em determinado grupo

Seja um usuário u , queremos saber quantas publicações ele curtiu em determinado grupo no qual participa. Definimos a função g tal que:

$$g : User \rightarrow Post$$

$$u \rightarrow Post \bowtie_{id_group=id} (\pi_{id}(f(u))).$$

Similarmente, podemos definir a função h para encontrarmos a quantidade de comentários:

$$h : User \rightarrow Comments$$

$$u \rightarrow Comments \bowtie_{id_group=id} (\pi_{id}(f(u))).$$

4.3 Recomendação de usuários ou grupos

Seja um usuário u , ele pode ver os usuários nos quais ele ainda não segue. Para isso, definimos a seguinte função α :

$$\alpha : User \rightarrow User$$

$$u \rightarrow \sigma_{id_follower==u.id}((User \times User) - Follow),$$

e para grupos a mesma ideia segue para a função β :

$$\beta : User \rightarrow Membership$$

$$u \rightarrow \sigma_{id_user==u.id}((User \times Group) - Membership).$$

5 Avaliação das formas normais

Analisando as formas normais das tabelas, obtemos os seguintes resultados:

5.1 Primeira Forma Normal

As tabelas **Post**, **Comments** e **Message** estão na Primeira Forma Normal visto que os atributos do complemento das chaves candidatas não são totalmente funcionalmente dependentes de suas respectivas chaves, isto é, há dependências parciais.

Em todos os casos, como cada tabela possui um **id_<tabela>** como chave primária, os demais componentes das chaves compostas são desnecessários.

5.2 Segunda Forma Normal

Nenhuma tabela está na Segunda Forma Normal.

5.3 Terceira Forma Normal

A demais tabelas estão na Terceira Forma Normal visto que todas estão na Segunda Forma Normal e não possuem dependências transitivas.

6 Diagrama da Camada de Mapeamento

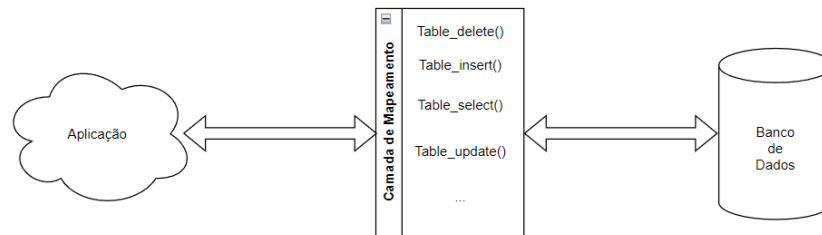


Figure 3: Camada de Mapeamento

7 Referências

- ELMASRI, R., NAVATHE, S. B., Sistemas de Banco de Dados, Sétima Edição, 2019, Editora Addison Wesley.
- HEUSER, C. A. , Projeto de banco de Dados, Editora Sagra Luzzatto
- Ferramenta de Modelagem MER: app.brmodeloweb.com
- Ferramenta de Modelagem MR: drawdb.verce.app

Repositório do projeto: [link](#).