

# ConnectMe: Uma abordagem de modelagem ER/EER com geração sintética de dados para testes

Daniel de Oliveira Trindade<sup>1</sup>

<sup>1</sup>Instituto de Computação – Universidade Federal do Amazonas (UFAM)  
69067-005 – Amazonas – AM – Brazil

daniel.trindade@icomp.ufam.edu.br

**Resumo.** *Este artigo apresenta o desenvolvimento de uma rede social rudimentar denominada ConnectMe, com foco na modelagem conceitual utilizando diagrama ER/EER. O trabalho detalha a evolução do modelo conceitual até a sua implementação física, incluindo a aplicação do processo de normalização de modelos até a 3FN. Em conjunto apresenta uma abordagem recente do uso de IA's generativas para a geração de scripts para gerar dados sintéticos que serão usados para testar a estrutura física da rede social.*

## 1. Introdução

Os sistemas de bancos de dados surgiram da necessidade de gerenciar eficientemente grandes volumes de dados nas organizações, evoluindo desde sistemas de arquivos tradicionais até sistemas mais sofisticados [Elmasri and Navathe 2011]. O modelo relacional, proposto por Codd[Codd 1970], trouxe uma nova abordagem matemática para a organização de dados, que posteriormente levou ao desenvolvimento dos Sistemas Gerenciadores de Bancos de Dados Relacionais (SGBDR). O modelo Entidade-Relacionamento (ER), introduzido por Chen[Chen 1976], revolucionou a modelagem conceitual, permitindo representar dados de forma mais em conformidade com a realidade.

A prática da modelagem de dados é uma habilidade fundamental para estudantes e profissionais de tecnologia, em especial em um contexto onde sistemas mais complexos como redes sociais que demandam uma estrutura robusta e escalável [Silberschatz et al. 2020]. E desenvolver projetos acadêmicos que simulam nas devidas proporções a experiência prática de como normalizar, manter a integridade referencial e otimizar consultas é fundamental.

De maneira tradicional, o teste desses sistemas requeria dados que normalmente eram gerados manualmente ou scripts básico de população. Com o advento das IAs generativas [Brown et al. 2023], surge um novo modo de automatizar a criação de scripts que sejam mais sofisticados, e que gerem conjuntos melhores de dados para representar cenários reais. Este trabalho explora essa abordagem através do desenvolvimento do ConnectMe, uma rede social simplificada que serve como caso de estudo para aplicação de conceitos de modelagem ER/EER e utilização de IA na geração de dados de teste.

## 2. Modelagem Conceitual do ConnectMe

### 2.1. Análise de Requisitos

A ConnectMe é uma plataforma online que permite aos usuários compartilhar informações, se conectar com amigos, participar de grupos de interesse e interagir por

meio de postagens, comentários e mensagens privadas. O objetivo da ConnectMe é fornecer uma experiência social envolvente e personalizada para seus usuários, promovendo conexões significativas e facilitando a comunicação e a interação online. A ConnectMe possui as seguintes características principais:

- **Perfil do Usuário:** Cada usuário possui um perfil que inclui informações pessoais como nome, foto, biografia, localização, data de nascimento, interesses, etc.
- **Postagens:** Os usuários podem fazer postagens em seu próprio perfil ou em grupos aos quais pertencem. As postagens podem incluir texto, imagens, vídeos e outros tipos de mídia.
- **Grupos:** Os usuários podem criar e participar de grupos com base em interesses comuns, hobbies, profissão, localização geográfica, etc. Os grupos servem como fóruns de discussão onde os membros podem compartilhar conteúdo relevante e interagir.
- **Conexões:** Os usuários podem se conectar uns aos outros, formando uma rede de amigos e contatos. As conexões permitem que os usuários vejam as postagens e atualizações de seus amigos em seus feeds de notícias.
- **Mensagens Privadas:** Os usuários podem enviar mensagens privadas uns aos outros para iniciar conversas individuais ou em grupo.

Em conjunto também é solicitado que seja possível fazer as seguintes consultas:

- **Consulta de perfil do usuário:** Recuperar informações do perfil de um usuário específico, incluindo nome, foto, biografia, etc.
- **Consulta de conexões de um usuário:** Obter a lista de amigos de um determinado usuário.
- **Consulta de postagens de um usuário:** Recuperar todas as postagens feitas por um usuário específico, ordenadas por data de publicação (as mais recentes primeiro).
- **Consulta de postagens em um grupo:** Listar 20 as postagens mais recentes feitas em um grupo específico.
- **Consulta de mensagens privadas:** Listar as 10 mensagens privadas mais recentes trocadas entre 2 usuários dados como entrada.
- **Consulta de busca de usuários:** Dado uma string como entrada, buscar os usuários cujos nomes contenham a string fornecida.
- **Consulta de tendências:** Listar o identificador dos 5 posts com mais interações nos últimos 7 dias.
- **Consulta de análise de engajamento:** Dado um post de um usuário, fazer a contagem de quantos usuários interagiram com ele nos últimos 7 dias.

## 2.2. Entidades e Relacionamentos

Depois de fazer a análise dos requisitos e ponderar sobre como seria a estrutura com base no que foi pedido, foi destacado as seguintes entidades:

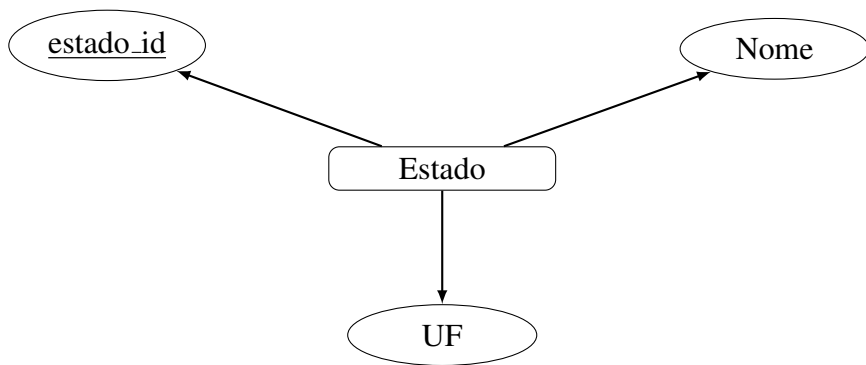


Figura 1. Diagrama da entidade Estado e seus atributos.

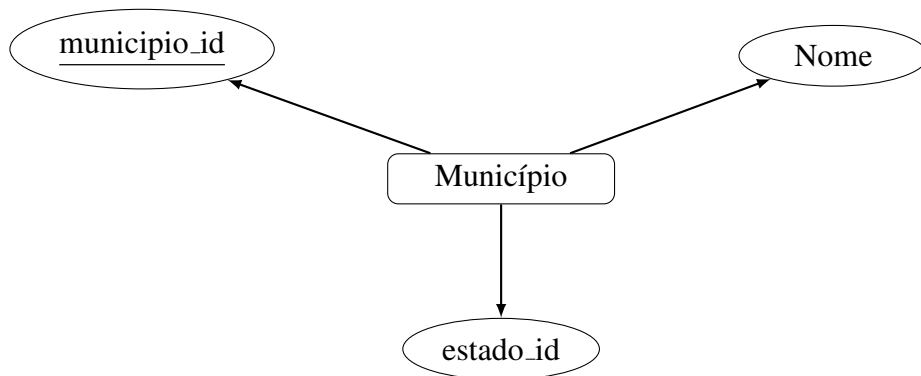


Figura 2. Diagrama da entidade Município e seus atributos.

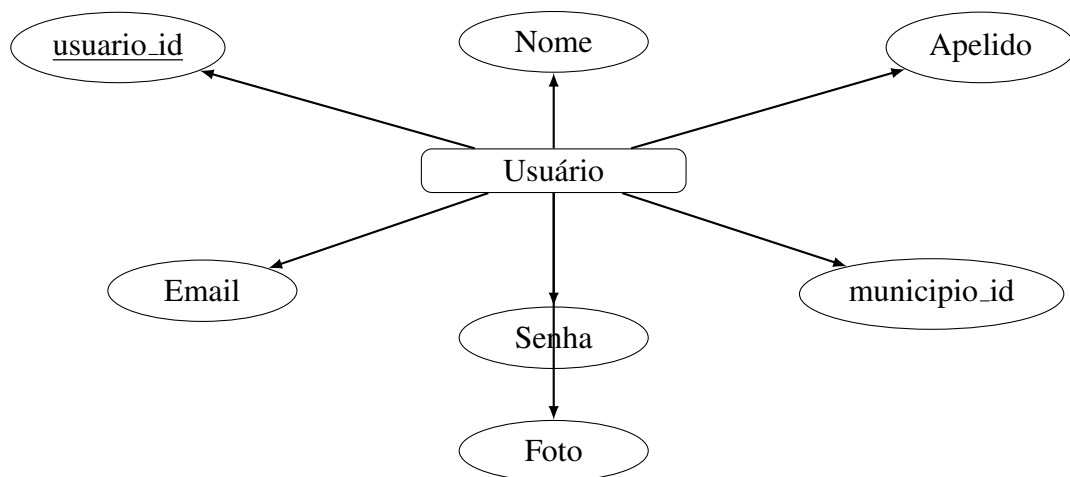


Figura 3. Diagrama da entidade Usuário e seus atributos.

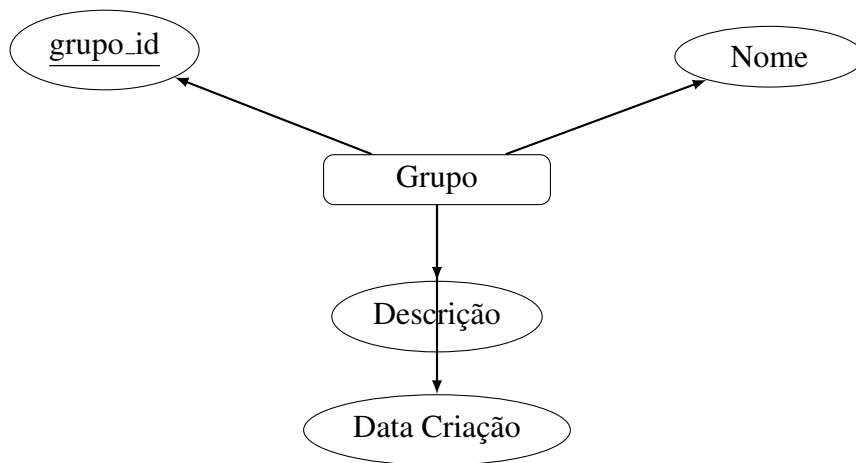


Figura 4. Diagrama da entidade Grupo e seus atributos.

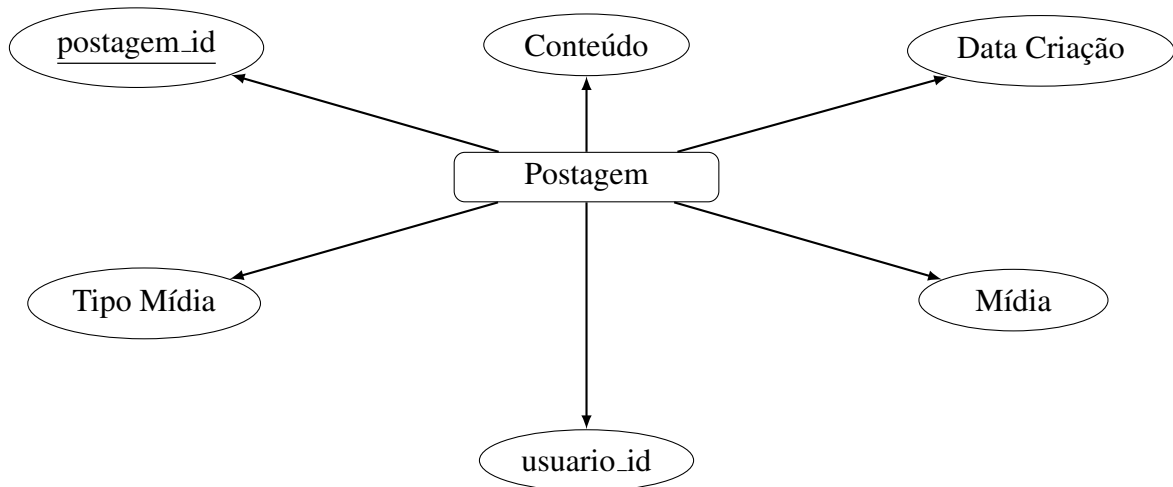
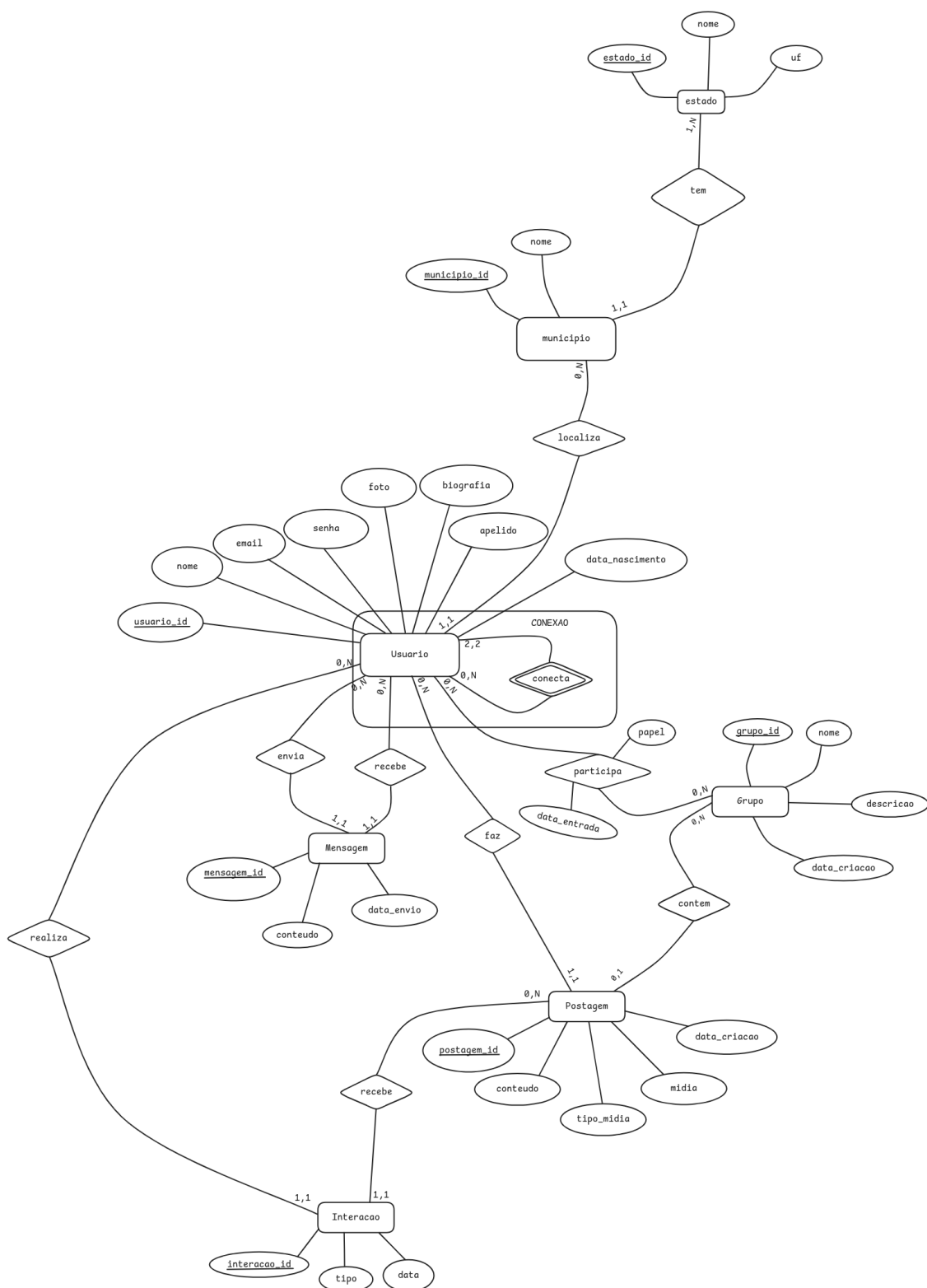


Figura 5. Diagrama da entidade Postagem e seus atributos.

### 2.3. Modelo ER/EER

Verificado as entidades com seus devidos atributos e feito a abstração do problema, o resultado foi o seguinte modelo ER/EER:



**Figura 6. Modelo Entidade-Relacionamento (ER) do sistema ConnectMe.**

No modelo acima (Figura 7), estão representadas as principais entidades, relacionamentos e atributos do sistema.

## 2.4. Normalizando o sistema

Depois de montado o modelo, foi realizado o processo de normalizar o banco até 3FN, que se mostrou necessário e o resultado se encontra abaixo:

- **ESTADO**(estado\_id, nome, uf)
- **MUNICIPIO**(municipio\_id, nome, #estado\_id)  
MUNICIPIO[estado\_id] = ESTADO[estado\_id]
- **USUARIO**(usuario\_id, nome, apelido, email, senha, foto, biografia, data\_nascimento, #municipio\_id)  
USUARIO[municipio\_id] = MUNICIPIO[municipio\_id]
- **GRUPO**(grupo\_id, nome, descricao, data\_criacao)
- **POSTAGEM**(postagem\_id, conteudo, data\_criacao, tipo\_midia, midia, #usuario\_id, #grupo\_id)  
POSTAGEM[usuario\_id] = USUARIO[usuario\_id]  
POSTAGEM[grupo\_id] = GRUPO[grupo\_id]
- **MENSAGEM**(mensagem\_id, conteudo, data\_envio, #remetente\_id, #destinatario\_id)  
MENSAGEM[remetente\_id] = USUARIO[usuario\_id]  
MENSAGEM[destinatario\_id] = USUARIO[usuario\_id]
- **CONEXAO**(conexao\_id, data\_conexao, #usuario\_id1, #usuario\_id2)  
CONEXAO[usuario\_id1] = USUARIO[usuario\_id]  
CONEXAO[usuario\_id2] = USUARIO[usuario\_id]
- **INTERCAO**(interacao\_id, tipo, data, #usuario\_id, #postagem\_id)  
INTERCAO[usuario\_id] = USUARIO[usuario\_id]  
INTERCAO[postagem\_id] = POSTAGEM[postagem\_id]
- **MEMBRO\_GRUPO**(membro\_id, papel, data\_entrada, #usuario\_id, #grupo\_id)  
MEMBRO\_GRUPO[usuario\_id] = USUARIO[usuario\_id]  
MEMBRO\_GRUPO[grupo\_id] = GRUPO[grupo\_id]

## 3. Implementação do ConnectMe

Depois do passo anterior, com base no diagrama e no modelo relacional. Foi implementado os scripts de criação das tabelas e relacionamentos no banco de dados.

```
1 CREATE DATABASE connectme;
2 USE connectme;
3
4 CREATE TABLE ESTADO (
5     estado_id INT AUTO_INCREMENT PRIMARY KEY,
6     nome VARCHAR(100) NOT NULL,
7     uf CHAR(2) NOT NULL
8 );
9
10 CREATE TABLE MUNICIPIO (
11     municipio_id INT AUTO_INCREMENT PRIMARY KEY,
12     nome VARCHAR(100) NOT NULL,
13     estado_id INT NOT NULL,
14     FOREIGN KEY (estado_id) REFERENCES ESTADO(estado_id)
15 );
16
17 CREATE TABLE USUARIO (
```

```

18  usuario_id INT AUTO_INCREMENT PRIMARY KEY,
19  nome VARCHAR(100) NOT NULL,
20  apelido VARCHAR(100) NOT NULL,
21  email VARCHAR(100) UNIQUE NOT NULL,
22  senha VARCHAR(255) NOT NULL,
23  foto VARCHAR(255),
24  biografia TEXT,
25  data_nascimento DATE NOT NULL,
26  municipio_id INT NOT NULL,
27  FOREIGN KEY (municipio_id) REFERENCES MUNICIPIO(municipio_id)
28 );
29
30 CREATE TABLE GRUPO (
31  grupo_id INT AUTO_INCREMENT PRIMARY KEY,
32  nome VARCHAR(100) NOT NULL,
33  descricao TEXT,
34  data_criacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP
35 );
36
37 CREATE TABLE POSTAGEM (
38  postagem_id INT AUTO_INCREMENT PRIMARY KEY,
39  conteudo TEXT NOT NULL,
40  data_criacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
41  tipo_midia VARCHAR(50),
42  midia VARCHAR(255),
43  usuario_id INT NOT NULL,
44  grupo_id INT,
45  FOREIGN KEY (usuario_id) REFERENCES USUARIO(usuario_id),
46  FOREIGN KEY (grupo_id) REFERENCES GRUPO(grupo_id)
47 );
48
49 CREATE TABLE MENSAGEM (
50  mensagem_id INT AUTO_INCREMENT PRIMARY KEY,
51  conteudo TEXT NOT NULL,
52  data_envio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
53  remetente_id INT NOT NULL,
54  destinatario_id INT NOT NULL,
55  FOREIGN KEY (remetente_id) REFERENCES USUARIO(usuario_id),
56  FOREIGN KEY (destinatario_id) REFERENCES USUARIO(usuario_id)
57 );
58
59 CREATE TABLE CONEXAO (
60  conexao_id INT AUTO_INCREMENT PRIMARY KEY,
61  data_conexao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
62  usuario_id1 INT NOT NULL,
63  usuario_id2 INT NOT NULL,
64  FOREIGN KEY (usuario_id1) REFERENCES USUARIO(usuario_id),
65  FOREIGN KEY (usuario_id2) REFERENCES USUARIO(usuario_id)
66 );
67
68 CREATE TABLE INTERACAO (
69  interacao_id INT AUTO_INCREMENT PRIMARY KEY,
70  tipo VARCHAR(50) NOT NULL,
71  data TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
72  usuario_id INT NOT NULL,
73  postagem_id INT NOT NULL,

```

```

74 FOREIGN KEY (usuario_id) REFERENCES USUARIO(usuario_id),
75 FOREIGN KEY (postagem_id) REFERENCES POSTAGEM(postagem_id)
76 );
77
78 CREATE TABLE MEMBRO_GRUPO (
79     membro_id INT AUTO_INCREMENT PRIMARY KEY,
80     papel VARCHAR(50) NOT NULL,
81     data_entrada TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
82     usuario_id INT NOT NULL,
83     grupo_id INT NOT NULL,
84     FOREIGN KEY (usuario_id) REFERENCES USUARIO(usuario_id),
85     FOREIGN KEY (grupo_id) REFERENCES GRUPO(grupo_id)
86 );
87
88 CREATE INDEX idx_usuario_nome ON USUARIO(nome);
89 CREATE INDEX idx_postagem_data ON POSTAGEM(data_criacao);
90 CREATE INDEX idx_mensagem_data ON MENSAGEM(data_envio);
91 CREATE INDEX idx_conexao_usuarios ON CONEXAO(usuario_id1, usuario_id2);

```

#### 4. Populando o banco

Com a criação das tabelas, foi utilizado a IA generativa chamada claude da Anthropic para gerar scripts que irão popular o nosso banco com dados para testar a modelagem feita.

##### Script 1. Script para popular usuários

```

1 # Funcao para popular a tabela de usuarios
2 def populate_usuarios(cursor, municipio_ids, num_usuarios=3000):
3     """
4     Funcao para popular a tabela de usuarios no banco de dados.
5
6     :param cursor: O cursor do banco de dados para execucao das queries.
7     :param municipio_ids: Lista de IDs de municipios para atribuicao aos
8         usuarios.
9     :param num_usuarios: Numero de usuarios a serem criados (default:
10         3000).
11     :return: Lista de IDs dos usuarios criados.
12     """
13
14     usuarios = [] # Lista para armazenar os IDs dos usuarios criados
15     used_emails = set() # Conjunto para garantir que os e-mails sejam
16         unicos
17
18     # Loop para criar os usuarios
19     for _ in tqdm(range(num_usuarios), desc="Populando usuarios", unit="
20         usuario"):
21         while True:
22             # Gerando um e-mail unico para o usuario
23             email = fake.email()
24             if email not in used_emails:
25                 used_emails.add(email)
26                 break
27
28         # Gerando os outros dados do usuario
29         data_nascimento = fake.date_of_birth(minimum_age=13, maximum_age
30             =80)

```



```

25     usuario = (
26         fake.name(), # Nome do usuario
27         fake.first_name(), # Apelido
28         email, # E-mail unico
29         fake.password(), # Senha
30         fake.image_url(), # URL de imagem (foto)
31         fake.text(max_nb_chars=200), # Biografia
32         random.choice(municipio_ids), # ID do municipio
33         data_nascimento # Data de nascimento
34     )
35
36     try:
37         # Inser o do usuario na tabela USUARIO
38         cursor.execute("""
39             INSERT INTO USUARIO (nome, apelido, email, senha, foto,
40             biografia,
41                                     municipio_id, data_nascimento)
42             VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
43             """, usuario)
44
45         # Adiciona o ID do usu rio recém-criado a lista
46         usuarios.append(cursor.lastrowid)
47
48     except mysql.connector.IntegrityError:
49         # Caso ocorra erro de integridade (por exemplo, e-mail duplicado)
50         , tenta criar o pr ximo usu rio
51         continue
52
53     return usuarios # Retorna a lista de IDs dos usuarios criados

```

The screenshot shows the MySQL Workbench interface. The 'Schemas' sidebar on the left lists various databases including 'connectme'. The central pane displays a SQL query: `SELECT * FROM connectme.USUARIO;`. The results pane on the right shows a table with 25 rows of data. The columns are: `#`, `usuario_id`, `nome`, `apelido`, `email`, `senha`, `foto`, `biografia`, and `data_nascimento`.

#	usuario_id	nome	apelido	email	senha	foto	biografia	data_nascimento
1		Luiz Gustavo F...	Bruno	mcamara@example.org	&@wS43Re_U	https://placekitten.com/609/721	Quia voluptas ut. Consectetur unde...	1969-02-05
2		Maria Sophia Vi...	Marcela	jade83@example.com	E8TtJnJ_L	https://placekitten.com/950/165	Dignissimos qui reprehenderit in en...	1948-09-12
3		Augusto Sá	Maria Flor	fernandes@example.net	5%WkOvPR...	https://placekitten.com/1019/87	Optio accusamus illum rem consect...	1961-04-12
4		Agatha Cardoso	Emily	qgoncalves@example.net	%kYKa&g03	https://dummyimage.com/44x654	Sed consequuntur saepe architecto...	1970-12-13
5		Alexia Cassiano	Luiz Fernando	santosmarcelo@example.net	NPK4lCnF	https://picsum.photos/597/704	Voluptatem consectetur harum qua...	1974-08-02
6		Sophie Costela	Bela	castromanuelly@example.net	comD7+kd*	https://picsum.photos/458/129	Voluptates vel sunt eveniet simil...	1994-07-12
7		Thiago Monten...	Maya	matheus74@example.org	*blitArr2	https://dummyimage.com/731x386	Eligendi voluptate qui officia at qua...	1995-05-30
8		Davi Lucca Castro	Mariah	rodriguesnicolas@example.org	(b3qwUU'i3	https://dummyimage.com/262x305	Quis quærat voluptas illo. Maiores...	1967-05-13
9		Isaque Cavalc...	Sabrina	xfarias@example.net	Bj2Hu7oyy	https://placekitten.com/745/304	Ratione ratione repellendus ex. Qui...	1945-07-16
10		St. Joaquim Me...	Ana Sophia	amacedo@example.org	L_!EQgMe17	https://placekitten.com/360/257	Libro suscipi id nobis reprehender...	1986-11-27
11		Luiz Otávio Gue...	Nicolas	psampaio@example.org	y2XINwCk4	https://dummyimage.com/814x592	Magni necessitatibus tempora offici...	1985-02-16
12		Leandro Pacheco	Raul	raparecota@example.net	0M40Xcm@6	https://picsum.photos/228/158	Ipsam vero quum est unde neque...	1976-06-19
13		Maria Clara Ma...	Vitor	kribeiro@example.com	Z@#4M5Aqtp	https://picsum.photos/724/330	Animi quidem reprehenderit molest...	1982-05-18
14		Juliana Duarte	Davi Lucas	ysales@example.com	#Mn@RH5B9l	https://dummyimage.com/966x183	Maxime nesciunt minima magnam...	1945-10-29
15		Ana Laura Mart...	Luna	nicole74@example.com	1)_5kGyJ	https://placekitten.com/155/345	Saepe vel neque ipsum expeditat veni...	2010-08-05
16		Bruna da Mota	Maria Liz	lararodrigues@example.net	@0CUwvqj	https://picsum.photos/339/307	Dicta modi laborum sunt. Voluptatu...	1976-11-30
17		Manuela Correa	Matheus	helenacorreia@example.org	+B2XL_Hho	https://placekitten.com/54/942	Ullam minus dolorum ipsam odit dol...	1964-02-18
18		Henrique Mend...	Dum	rodrigues@example.org	60C4widiK	https://picsum.photos/36/184	Enim commodi descripto ab ut. Illo...	2009-07-15
19		Maya Lopes	Paulo	jao-guilhermeoliveira@exa...	eXBWj08rD	https://placekitten.com/360/156	Architecto placeat saepe dolores p...	1994-11-04
20		Bianca Rios	Théo	rav49@example.net	H49dQoc*	https://dummyimage.com/101x72	Aliquid rem voluptatibus quis autem...	2009-12-09
21		Sita, Isabel Porto	Mariane	ana-beatriz75@example.org	SSCL@3nA@n	https://dummyimage.com/593x745	Assumenda dolorum perferenda as...	1951-01-30
22		Aurora Leão	Raquel	elisarova@example.net	IY_3RinDEP	https://placekitten.com/772/370	Dignissimos debitis maxime cumqu...	1965-10-23
23		Juliana Nogueira	Luana	fretasbenjamin@example.org	yn6gAEf&w	https://picsum.photos/360/185	Tenetur saepe provident harum per...	1981-10-02
24		Matheus Costela	Lorena	maria-luisa76@example.net	jwPpYwGq8	https://dummyimage.com/515x591	Magni fugiat atque officia quidem re...	1978-07-11
25		Maya Pacheco	Nathan	rcavalcanti@example.net	IGWdriQae1	https://dummyimage.com/369x966	Tenetur voluptas ad maiores magni...	1982-05-17

Figura 7. Exemplo da tabela usuário populada pelo script.

## 5. Implementação das consultas

Para verificar se o modelo está de acordo com o solicitado, iremos implementar 8 queries e que foram solicitadas.

### 5.1. Consulta de perfil do usuário

```
1 SELECT
2   u.usuario_id ,
3   u.nome ,
4   u.email ,
5   u.foto ,
6   u.biografia ,
7   u.data_nascimento ,
8   m.nome AS municipio ,
9   e.nome AS estado ,
10  e.uf
11 FROM USUARIO u
12 JOIN MUNICIPIO m ON u.municipio_id = m.municipio_id
13 JOIN ESTADO e ON m.estado_id = e.estado_id
14 WHERE usuario_id = ?;--substituir o ? pelo usuario_id que quer procurar
```

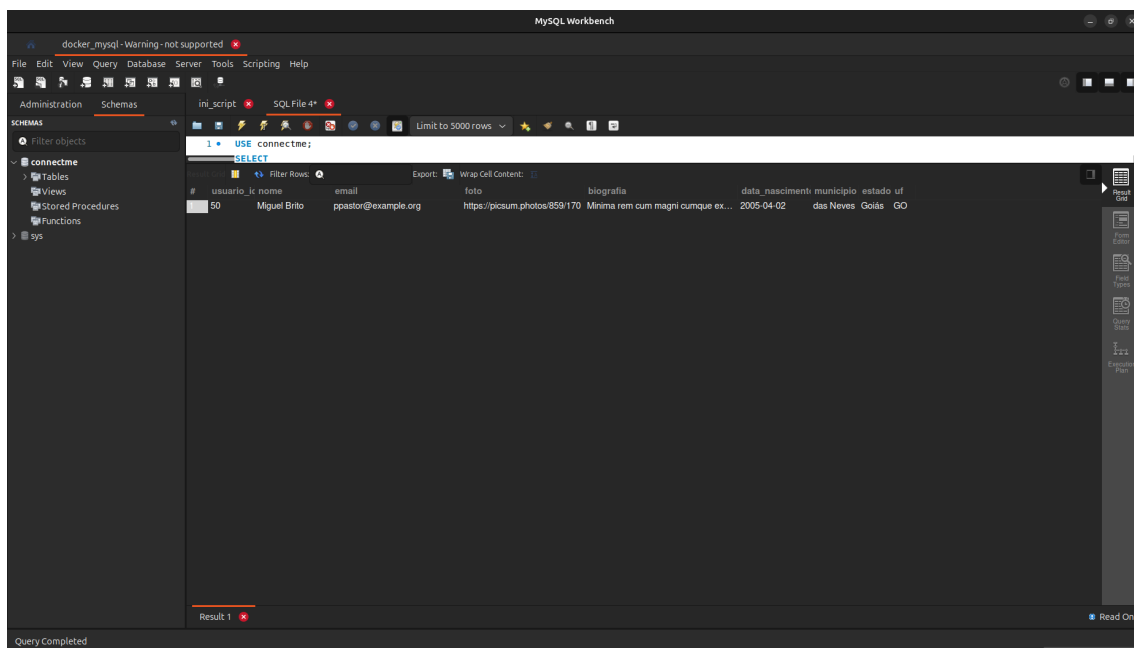


Figura 8. um exemplo de saída para a query 1

### 5.2. Consulta de conexões de um usuário

```
1 SELECT
2   u.usuario_id ,
3   u.nome ,
4   u.email
5 FROM
6   CONEXAO c
7 JOIN
```

```

8  USUARIO u ON (u.usuario_id = c.usuario_id1 AND c.usuario_id2 = ?) --
    substituir o ? por pelo usuario_id de que quer procurar
9      OR (u.usuario_id = c.usuario_id2 AND c.usuario_id1 = ?); --
    substituir o ? pelo mesmo usuario_id usado acima

```

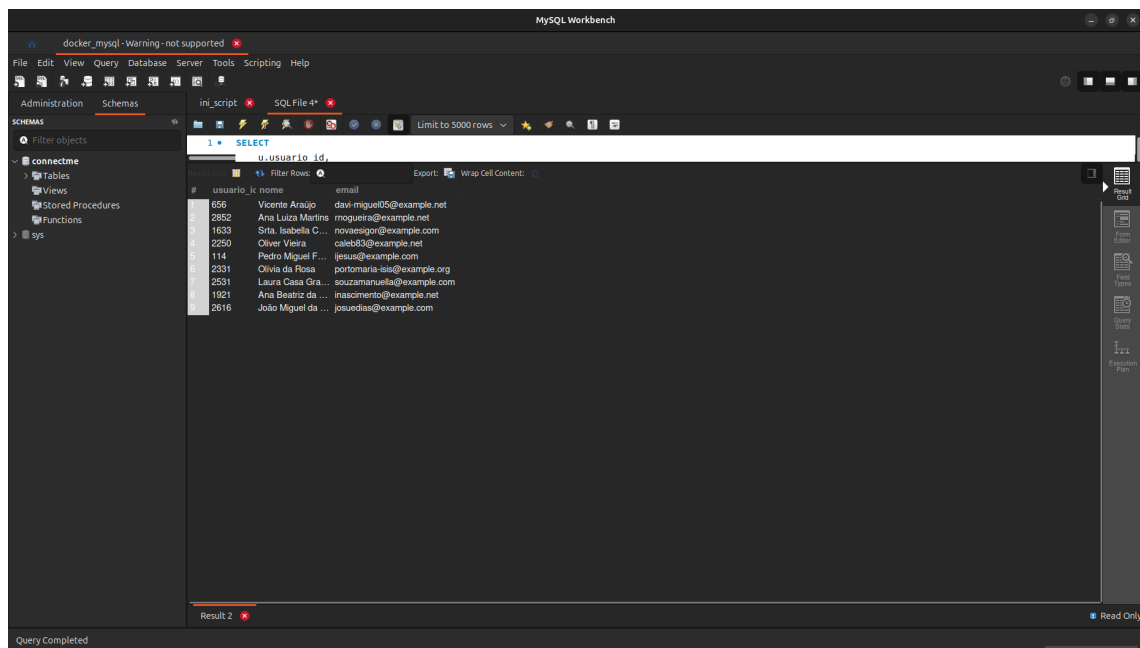


Figura 9. um exemplo de saída para a query 2

### 5.3. Consulta de postagens de um usuário

```

1  SELECT
2      *
3  FROM
4      POSTAGEM
5  WHERE
6      usuario_id = ? --substituir o ? por pelo usuario_id de que quer
    procurar
7  ORDER BY
8      data_criacao DESC;

```

#	postagem_id	conteudo	data_criacao	tipo_midia	midia	usuario_id	grupo_id
1	58526	Ea expedita repellat quae laborum l...	2024-12-02 18:47:11	texto	...	50	16
2	57853	Quod inodum ut ture sc...	2024-12-02 18:47:11	video	https://placekitten.com/14/139	50	562
3	57376	Eius vero sapiente ipsa perspiciatis ...	2024-12-02 18:47:11	video	https://dummyimage.com/1011x303	50	381
4	57333	Ullam dislindo optio beatae aliqu...	2024-12-02 18:47:11	texto	...	50	...
5	41806	Similique quidem magnan conno...	2024-12-02 18:47:08	video	https://dummyimage.com/302x520	50	267
6	39714	Nostrum quaerat nulla. Beatae sapi...	2024-12-02 18:47:07	video	https://picsum.photos/320/998	50	510
7	38446	Etisque ille eligendi delendi iste mnu...	2024-12-02 18:47:07	video	https://dummyimage.com/119x595	50	...
8	36943	Sit ullam ab explicabo. Nulla fugit s...	2024-12-02 18:47:07	video	https://dummyimage.com/119x595	50	...
9	27952	Assumenda sit debilis architecto a...	2024-12-02 18:47:07	texto	...	50	130
10	27378	Iste aperiam voluptas maxime. Nob...	2024-12-02 18:47:04	video	https://picsum.photos/539/994	50	387
11	20352	Esse facere doloribus ratione volup...	2024-12-02 18:47:04	video	https://dummyimage.com/303x536	50	289
12	16598	Voluptate ducimus dicta tempora d...	2024-12-02 18:47:03	video	https://dummyimage.com/32x240	50	162
13	16332	Modi quasi culpa dolore aspernatur...	2024-12-02 18:47:02	texto	...	50	213
14	16332	Aspernatur debilis natus reiciendis...	2024-12-02 18:47:02	imagem	https://dummyimage.com/591x767	50	...

Figura 10. um exemplo de saída para a query 3

## 5.4. Consulta de postagens em um grupo

```

1 SELECT
2   p.postagem_id,
3   p.conteudo,
4   p.data_criacao,
5   p.tipo_midia,
6   p.midia,
7   p.usuario_id,
8   u.nome AS nome_usuario
9 FROM
10  POSTAGEM p
11 JOIN
12  USUARIO u ON p.usuario_id = u.usuario_id
13 WHERE
14  p.grupo_id = ? -- substituir o ? pelo id da postagem que quer
15                  procurar
16 ORDER BY
17  p.data_criacao DESC
18 LIMIT 20;

```

#	mensagem_id	conteudo	data_envio	tipo	midia	usuario_id	nome_usuario
59316		Veritatis opto tempora minus accus...	2024-12-02 18:47:11	texto	...	1474	Kaikue Jesus
54436		Magnam consequuntur itaque nulla...	2024-12-02 18:47:10	texto	...	289	Sr. Ryan Montenegro
53697		Adipisci voluptate maxime eos natu...	2024-12-02 18:47:10	video	https://dummyimage.com/621x688	1017	Maria Vitoria Siqueira
52334		Non eos enim dolorum ex tempora...	2024-12-02 18:47:10	texto	...	491	Joaquim Souza
46593		Dignissimos voluptate at dignissimo...	2024-12-02 18:47:09	video	https://dummyimage.com/716x937	1835	Luz Miguel Casa Grande
46652		Quod explicabo illum quam praese...	2024-12-02 18:47:09	video	https://placekitten.com/337/277	1318	Sra. Emily Fogaca
47131		Dicta illum doloremque quod amet...	2024-12-02 18:47:09	texto	...	2272	Melina Pinto
49896		Ipse nobis doloribus. Harum volut...	2024-12-02 18:47:09	imagem	https://placekitten.com/411/294	1465	Luz Henrique Pinto
50219		Totam eius accusamus. Consequat...	2024-12-02 18:47:09	video	https://picsum.photos/743/908	2891	Davi Miguel da Rocha
50454		In occaecati sapiente itaque nam. F...	2024-12-02 18:47:09	imagem	https://picsum.photos/589/752	292	Leandro Costa
50732		Reprehenderit nemo reprehenderit d...	2024-12-02 18:47:09	texto	...	889	Fernanda Oliveira
45614		Atque perspiciatis odia aliquet dabo...	2024-12-02 18:47:08	texto	...	1615	Ana Carolina Brito
42767		Excepuntur iste mei adipisci ea cumq...	2024-12-02 18:47:08	video	https://placekitten.com/545/81	123	Lorenzo Correia

Figura 11. um exemplo de saída para a query 4

## 5.5. Consulta de mensagens privadas

```

1 SELECT
2   mensagem_id,
3   conteudo,
4   data_envio,
5   remetente_id,
6   destinatario_id
7 FROM
8   MENSAGEM
9 WHERE
10  (remetente_id = ? AND destinatario_id = ?) -- ex: substituir o 1   ?
    por 2 e o 2   por 3
11  OR (remetente_id = ? AND destinatario_id = ?) -- ex: substituir o 1
    ? por 3 e o 2   por 2
12 ORDER BY
13   data_envio DESC
14 LIMIT 10;

```

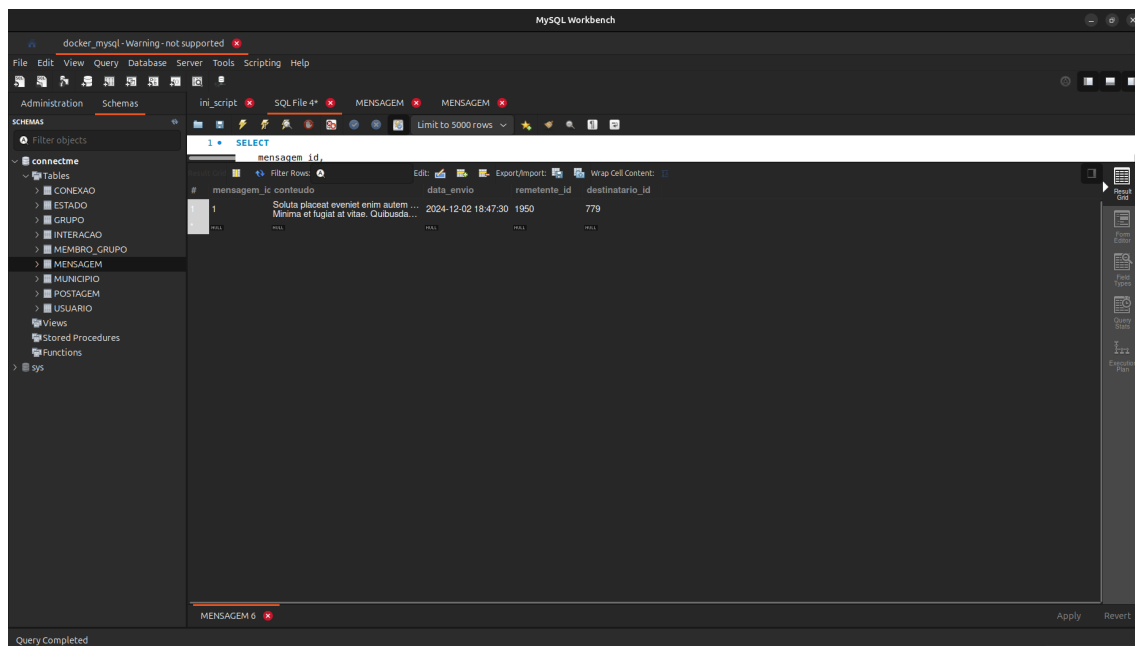


Figura 12. um exemplo de saída para a query 5

## 5.6. Consulta de busca de usuários

```

1 SELECT
2     usuario_id,
3     nome,
4     email,
5     municipio_id
6 FROM
7     USUARIO
8 WHERE
9     nome LIKE CONCAT('%', ?, '%'); -- substituir o ? pela string que quer
    procurar, ex: 'Daniel'

```

usuario_id	nome	email	municipio_id
59	Sr. Daniel Moraes	gabrielly91@example.org	449
60	Daniela Pacheco	garciaocelia@example.net	93
91	Daniela Borges	tviana@example.com	477
135	Srta. Daniela C.	barbararodrigues@example.net	466
380	Daniela Correia	gab-pedro79@example.net	184
399	Daniela Pacheco	hrezende@example.com	146
419	Daniela Lopes	fernanda14@example.com	355
425	Daniela Lima	das-nevesjoaquim@example...	872
460	Daniela Barros	bentz18@example.com	724
539	Daniel da Cunha	qda-cunha@example.com	810
1097	Daniela Correia	maria-luzacostela@example...	70
1196	Daniel da Cruz	marina51@example.com	572
1488	Daniela Rodrigues	caldeiramariah@example.com	159
1529	Dr. Daniel Melo	uiveira@example.org	965
1576	Daniel Peixoto	rafaela44@example.net	577
1720	Dr. Daniel Ribeiro	lucas41@example.com	872
1841	Daniel Pastor	guerraeduardo@example.net	673
1854	Daniel Cavalcanti	maria-fz46@example.com	748
1903	Sr. Daniel Perera	whenera@example.org	602
1943	Dr. Daniel Fogaga	ana-leurada-rocha@example...	218
2042	Daniela Vargas	rodriguesandres@example.org	645
2278	Daniela Ramos	leobrito@example.com	99
2317	Daniela Mendo...	abreu-luz-otavio@example.org	855
2326	Daniel Caldeira	garcaberganin@example.net	402
2425	Daniela da Mata	rebecamendes@example.net	455
2742	Daniel Teixeira	rebeca03@example.net	64
2783	Sr. Daniel Costela	maria-fermandanovaes@exa...	140
2817	Dr. Daniel Rocha	santosjosue@example.org	704
2898	Daniel Melo	siquieraramaria-fernanda@exa...	682

Figura 13. um exemplo de saída para a query 6

## 5.7. Consulta de tendências

```

1 SELECT
2   p.postagem_id,
3   p.conteudo AS conteudo_postagem,
4   p.data_criacao AS data_postagem,
5   p.tipo_midia,
6   p.midia,
7   u.nome AS nome_usuario,
8   COUNT(i.interacao_id) AS total_interacoes,
9   CASE
10     WHEN p.grupo_id IS NOT NULL THEN g.nome
11     ELSE 'N o      de grupo'
12   END AS nome_grupo
13 FROM
14   INTERACAO i
15 JOIN
16   POSTAGEM p ON i.postagem_id = p.postagem_id
17 JOIN
18   USUARIO u ON p.usuario_id = u.usuario_id
19 LEFT JOIN
20   GRUPO g ON p.grupo_id = g.grupo_id
21 WHERE
22   i.data >= CURDATE() - INTERVAL 7 DAY
23 GROUP BY
24   p.postagem_id
25 ORDER BY
26   total_interacoes DESC
27 LIMIT 5;

```

The screenshot shows the MySQL Workbench interface. The left sidebar displays a schema tree with the following structure:

- connectme
  - Tables
    - CONEXAO
    - ESTADO
    - GRUPO
    - INTERACAO
    - MEMBRO\_GRUPO
    - MENGAGEM
    - MUNICIPIO
    - POSTAGEM
    - USUARIO
  - Views
  - Stored Procedures
  - Functions
- sys

The central query editor contains the following SQL query:

```
SELECT
  COUNT(DISTINCT i.usuario_id) AS total_usuarios_interagiram
FROM
  INTERACAO i
JOIN
  POSTAGEM p ON i.postagem_id = p.postagem_id
WHERE
  p.postagem_id = ? -- Substitua '?' pelo postagem_id do post desejado
AND i.data >= CURDATE() - INTERVAL 7 DAY;
```

The results pane on the right displays the output of the query, showing a table with the following columns: #, postagem\_id, ic conteudo, postagem, data, postagem, tipo, midia, midia, nome\_usuario, total\_interacao, nome\_grupo. The table contains 5 rows of data.

#	postagem_id	ic conteudo	postagem	data	postagem	tipo	midia	midia	nome_usuario	total_interacao	nome_grupo
17534		Eius distinctio laudantium quidem d...		2024-12-02 18:47:02	video		https://placekitten.com/196/853		Ana Beatriz Viana	12	da Conceição
16630		Animi corrupti placeat labore...		2024-12-02 18:47:02	texto				Maria Sophia da Mata	11	Não é de grupo
16615		Aperiam repellat laudantium. Sit sin...		2024-12-02 18:47:02	texto				Joana Gonçalves	11	Não é de grupo
12744		Ea enim corrupti eaum corrupti que...		2024-12-02 18:47:01	texto				Brenda da Rocha	11	Farias S/A
8171		A aliquam eligendi praesentium vol...		2024-12-02 18:47:00	texto				Anthony Andrade	11	Não é de grupo

Figura 14. um exemplo de saída para a query 7

## 5.8. Consulta de tendências

```

1 SELECT
2   COUNT(DISTINCT i.usuario_id) AS total_usuarios_interagiram
3 FROM
4   INTERACAO i
5 JOIN
6   POSTAGEM p ON i.postagem_id = p.postagem_id
7 WHERE
8   p.postagem_id = ? -- Substitua '?' pelo postagem_id do post desejado
9   AND i.data >= CURDATE() - INTERVAL 7 DAY;
```



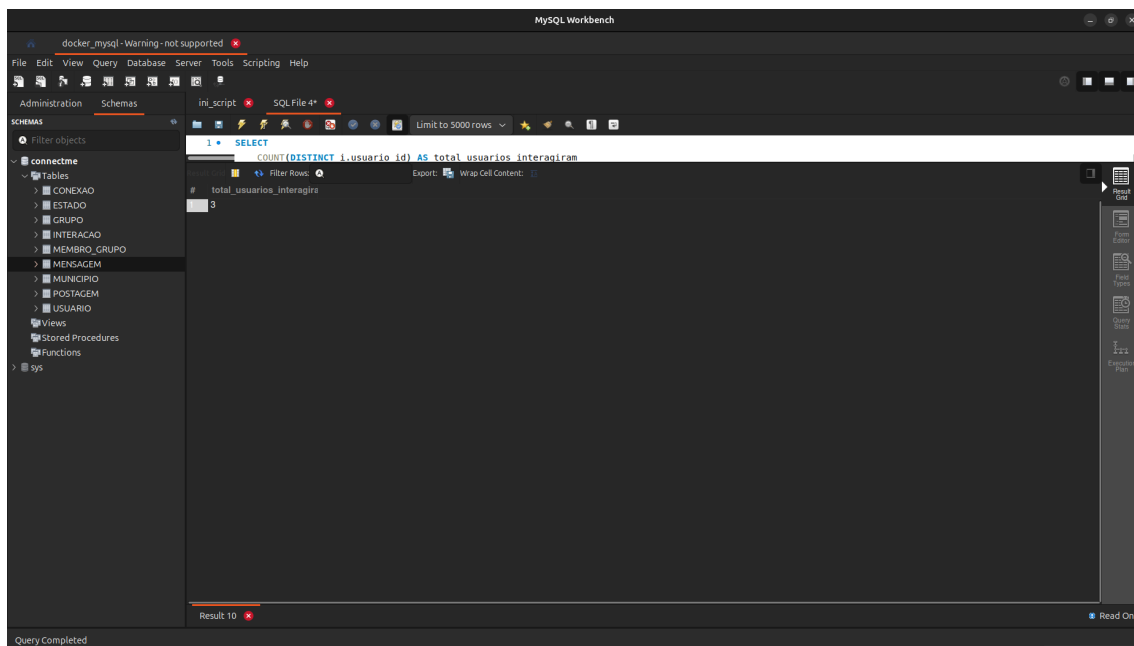


Figura 15. um exemplo de saída para a query 8

## 6. Conclusão

Neste trabalho, abordado o desenvolvimento de um sistema de banco de dados para o gerenciamento da rede social ConnectMe para o gerenciamento dos seus usuário, post, grupos e etc. A estrutura adotada foi possível devido ao uso da técnica de modelagem de dados ER, para garantir um sistema que atenda as necessidades de escalabilidade e robustez do sistema, essenciais para redes sociais de larga escala.

Além disso, a implementação de script usando IA generativa para auxiliar nos testes populando o banco de dados com dados falsos e realistas, mostrou-se uma estratégia eficaz no desenvolvimento de projetos reais. Não apenas no processo de teste, mas também contribuindo para a criação de um sistema mais alinhado com a realidade.

Este trabalho demonstrou a importância de um design bem construído e adoção de boas práticas garantem uma qualidade e eficiência no desenvolvimento de sistemas baseados em bancos de dados relacionais e como um projeto construído em passos bem definidos é importante para o sucesso. Por fim, a utilização de IA se consolidou como uma ferramenta de grande valia, para otimizar o ciclo de desenvolvimento de sistemas.

## Referências

- Brown, T. B. et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Chen, P. P.-S. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36.
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.
- Elmasri, R. and Navathe, S. B. (2011). *Fundamentals of Database Systems*. Addison-Wesley, 6 edition.

Silberschatz, A., Korth, H. F., and Sudarshan, S. (2020). *Database System Concepts*. McGraw-Hill, 7 edition.