

# Matheus Peviani Vellone

Caso alguma Query esteja difícil de ser lida, acesse <http://poorsql.com/> (<http://poorsql.com/>) e copie/cole a query para obter a versão indentada dela

## Usuários

- `private static final String createQuery = "INSERT INTO usuario(login, senha, nome, nascimento, descricao, foto, created, modified, servidor) VALUES(?, md5(?), ?, ?, ?, ?, ?, ?, ?) RETURNING id;";`
- `private static final String readQuery = "SELECT * FROM usuario WHERE id = ?;";`
- `private static final String updateQuery = "UPDATE usuario SET login = ?, nome = ?, nascimento = ?, descricao = ?, foto = ?, modified = 'now' WHERE id = ?;";`
- `private static final String updateWithPasswordQuery = "UPDATE usuario SET login = ?, nome = ?, nascimento = ?, descricao = ?, foto = ?, senha = md5(?), modified='now' WHERE id = ?;";`
- `private static final String deleteQuery = "DELETE FROM usuario WHERE id = ?;";`
- `private static final String allQuery = "SELECT * FROM usuario;";`
- `private static final String todosUsuariosQuery = "SELECT * FROM usuario WHERE servidor = 23;";`
- `private static final String authenticateQuery = "SELECT id, senha, nome, nascimento, descricao, foto FROM usuario WHERE login = ?;";`

## Seguidores

- `private static final String seguirQuery = "INSERT INTO seguidores(id_seguidor, id_segue, created) VALUES(?, ?, 'now')";`
- `private static final String quemEuSigoQuery = "SELECT * FROM seguidores s INNER JOIN usuario u ON s.id_segue = u.id WHERE id_seguidor = ?";`
- `private static final String quemMeSegue = "SELECT * FROM seguidores s INNER JOIN usuario u ON s.id_seguidor = u.id WHERE id_segue = ?";`
- `private static final String paraDeSeguirQuery = "DELETE FROM seguidores WHERE id_seguidor = ? AND id_segue = ?";`
- `private static final String euSigoCountQuery = "SELECT COUNT(*) AS contador FROM seguidores WHERE id_seguidor = ?";`
- `private static final String meSegueCountQuery = "SELECT COUNT(*) AS contador FROM seguidores WHERE id_segue = ?";`
- `private static final String postFollowQuery = "SELECT * FROM post p INNER JOIN seguidores s ON p.id_autor = s.id_segue WHERE s.id_seguidor = ? ORDER BY p.id DESC";`

## Pesquisa

- `private static final String searchQuery = "SELECT * FROM usuario WHERE LOWER(login) LIKE LOWER(?) OR LOWER(nome) LIKE LOWER(?) OR LOWER(descricao) LIKE LOWER(?)";`
- `private static final String searchQuery = "SELECT * FROM grupo WHERE LOWER(nome) LIKE LOWER(?)";`
- `private static final String searchQuery = "SELECT * FROM post WHERE LOWER(titulo) LIKE LOWER(?) OR LOWER(conteudo) LIKE LOWER(?);";`

Limpa dados antes de exportar dados

```
private static final String limpa_dados = "DELETE FROM comentarios WHERE servidor = ?; DELETE FROM grupo WHERE servidor = ?; DELETE FROM integrantes_grupo WHERE servidor = ?; DELETE FROM likes WHERE servidor = ?; DELETE FROM post WHERE servidor = ?; DELETE FROM seguidores WHERE servidor = ?; DELETE FROM usuario WHERE servidor = ?;";
```

## Grupo

- `private static final String createQuery = "INSERT INTO grupo(id_admin, nome, created, modified, servidor) VALUES(?, ?, ?, ?, ?);";`
- `private static final String updateQuery = "UPDATE grupo SET nome = ?, modified = 'now' WHERE id = ?;";`
- `private static final String readQuery = "SELECT * FROM grupo WHERE id = ?;";`
- `private static final String deleteQuery = "DELETE FROM grupo WHERE id = ?;";`
- `private static final String allQuery = "SELECT * FROM grupo;";`
- `private static final String allQueryAdmin = "SELECT * FROM grupo WHERE id_admin = ?;";`
- `private static final String SairGrupoQuery = "DELETE FROM integrantes_grupo WHERE id_usuario = ? AND id_grupo = ?;";`
- `private static final String allQueryIntegrante = "SELECT * FROM integrantes_grupo WHERE id_usuario = ?;";`
- `private static final String novoIntegranteQuery = "INSERT INTO integrantes_grupo(id_grupo, id_usuario, created) VALUES(?, ?, 'now');";`

## Post

- `private static final String createQuery = "INSERT INTO post(id_autor, titulo, conteudo, created, modified, republicacao, servidor) VALUES(?, ?, ?, ?, ?, false, ?) RETURNING id;";`
- `private static final String readQuery = "SELECT * FROM post WHERE id = ?;";`
- `private static final String autorQuery = "SELECT * FROM post WHERE id_autor = ?;";`
- `private static final String updateQuery = "UPDATE post SET titulo = ?, conteudo = ?, modified = 'now' WHERE id = ?;";`
- `private static final String deleteQuery = "DELETE FROM post WHERE id = ?;";`
- `private static final String allQuery = "SELECT * FROM post;";`
- `private static final String todosPostsQuery = "SELECT * FROM post WHERE republicacao = false AND servidor = 23;";`
- `private static final String todosRepostsQuery = "SELECT * FROM post WHERE republicacao = true AND servidor = 23;";`
- `private static final String allTemaQuery = "SELECT * FROM post WHERE LOWER(conteudo) LIKE LOWER(?);";`

## Post

- `private static final String createQuery = "INSERT INTO post(id_autor, titulo, conteudo, created, modified, republicacao, servidor) VALUES(?, ?, ?, ?, ?, false, ?) RETURNING id;"`;
- `private static final String readQuery = "SELECT * FROM post WHERE id = ?;"`;
- `private static final String autorQuery = "SELECT * FROM post WHERE id_autor = ?;"`;
- `private static final String updateQuery = "UPDATE post SET titulo = ?, conteudo = ?, modified = 'now' WHERE id = ?;"`;
- `private static final String deleteQuery = "DELETE FROM post WHERE id = ?;"`;
- `private static final String allQuery = "SELECT * FROM post;"`;
- `private static final String todosPostsQuery = "SELECT * FROM post WHERE republicacao = false AND servidor = 23;"`;
- `private static final String todosRepostsQuery = "SELECT * FROM post WHERE republicacao = true AND servidor = 23;"`;
- `private static final String allTemaQuery = "SELECT * FROM post WHERE LOWER(conteudo) LIKE LOWER(?)"`;
- `private static final String repostQuery = "INSERT INTO post(id_autor, republicacao, id_repub, created, modified, servidor) VALUES(?, true, ?, ?, ?, ?) RETURNING id;"`;

## Likes/Dislikes

- `private static final String todosLikes = "SELECT * FROM likes WHERE like_sit = 1 AND servidor = 23;";`
- `private static final String todosDislikes = "SELECT * FROM likes WHERE like_sit = -1 AND servidor = 23;";`
- `private static final String likeCount = "SELECT COUNT(*) AS contador FROM likes WHERE post_id = ? AND like_sit = 1;";`
- `private static final String dislikeCount = "SELECT COUNT(*) AS contador FROM likes WHERE post_id = ? AND like_sit = -1;";`
- `private static final String situacaoLike = "SELECT like_sit AS contador FROM likes WHERE usuario_id = ? AND post_id = ?;";`
- `private static final String colocaLike = "INSERT INTO likes(usuario_id, post_id, like_sit, servidor, created) VALUES(?, ?, 1, ?, ?);";`
- `private static final String colocaDisLike = "INSERT INTO likes(usuario_id, post_id, like_sit, servidor, created) VALUES(?, ?, -1, ?, ?);";`
- `private static final String tiraLike = "DELETE FROM likes WHERE usuario_id = ? AND post_id = ?;";`

## Comentários

- `private static final String comentar = "INSERT INTO comentarios(id_post, comentario, dono, created, modified, servidor) VALUES(?, ?, ?, ?, ?, ?);";`
- `private static final String comentarios = "SELECT * FROM comentarios WHERE id_post = ?";`
- `private static final String todosComentarios = "SELECT * FROM comentarios WHERE servidor = 23";`

# Top 20 Usuários

- ```
SELECT a.id_usuario as id, a.nome, a.foto, CASE WHEN num_comentario IS NULL THEN 0 ELSE num_comentario*2 END + CASE WHEN num_repost IS NULL THEN 0 ELSE num_repost*2 END + CASE WHEN num_like IS NULL THEN 0 ELSE num_like END - CASE WHEN num_dislike IS NULL THEN 0 ELSE num_dislike END as influencia FROM (SELECT u.id AS id_usuario, u.nome, u.foto FROM usuario u, post p WHERE p.created <= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.created >= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.republicacao = false AND p.id_autor = u.id GROUP BY id_usuario, u.nome) AS a NATURAL LEFT JOIN (SELECT b.id_autor AS id_usuario, count(c.id) as num_comentario FROM comentarios c, (SELECT p.id_autor, p.id FROM post p WHERE p.created <= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.created >= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.republicacao = false ORDER BY p.id_autor) AS b WHERE b.id = c.id_post GROUP BY id_usuario ORDER BY id_usuario) AS c NATURAL LEFT JOIN (SELECT l.usuario_id AS id_usuario, count(l.like_sit) AS num_like FROM likes l, (SELECT p.id FROM post p WHERE p.created <= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.created >= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.republicacao = false) AS d WHERE l.like_sit = 1 AND d.id = l.post_id GROUP BY id_usuario) as e NATURAL LEFT JOIN (SELECT l.usuario_id AS id_usuario, count(l.like_sit) AS num_dislike FROM likes l, (SELECT p.id FROM post p WHERE p.created <= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.created >= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.republicacao = false) AS f WHERE l.like_sit = -1 AND f.id = l.post_id GROUP BY id_usuario) as g NATURAL LEFT JOIN (SELECT p.id_autor as id_usuario, count(p.id) as num_repost FROM post p, (SELECT p.id FROM post p WHERE p.created <= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.created >= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.republicacao = false) AS h WHERE h.id = p.id_repub AND p.republicacao = true GROUP BY id_usuario) as i order by influencia desc limit 20
```

# Top 20 Publicações

- ```

SELECT a.id_usuario AS id ,a.nome ,a.foto ,a.titulo ,a.conteudo ,CASE WHEN num_comentario IS NULL THEN 0 ELSE num_comentario * 3 END + CASE WHEN num_repost IS NULL THEN 0 ELSE num_repost * 3 END + CASE WHEN num_like IS NULL THEN 0 ELSE num_like * 2 END + CASE WHEN num_dislike IS NULL THEN 0 ELSE num_dislike END AS impacto FROM (
SELECT u.id AS id_usuario ,u.nome ,u.foto ,p.titulo ,p.conteudo ,p.id AS id_post FROM
usuario u ,post p WHERE p.created <= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.created
>= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.republicacao = false AND p.id_autor = u.id ) AS a NATURAL LEFT JOIN (
SELECT c.id_post ,count(c.id) AS num_comentario FROM comentarios c ,(
SELECT p.id FROM post p WHERE p.created <= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.created
>= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.republicacao = false ) AS b WHERE b.id = c.id_post GROUP BY id_post ORDER BY id_post ) AS c NATURAL LEFT JOIN (
SELECT l.post_id AS id_post ,count(l.like_sit) AS num_like FROM likes l ,(
SELECT p.id FROM post p WHERE p.created <= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.created
>= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.republicacao = false ) AS d WHERE l.like_sit = 1 AND d.id = l.post_id GROUP BY id_post ) AS e NATURAL LEFT JOIN (
SELECT l.post_id AS id_post ,count(l.like_sit) AS num_dislike FROM likes l ,(
SELECT p.id FROM post p WHERE p.created <= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.created
>= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.republicacao = false ) AS f WHERE l.like_sit = - 1 AND f.id = l.post_id GROUP BY id_post ) AS g NATURAL LEFT JOIN (
SELECT p.id AS id_post ,count(p.id) AS num_repost FROM post p ,(
SELECT p.id FROM post p WHERE p.created <= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.created
>= TO_DATE(?, 'YYYY-MM-DD HH24:MI') AND p.republicacao = false ) AS h WHERE h.id = p.id_repub AND p.republicacao = true GROUP BY id_post ) AS i ORDER BY impacto DESC limit 20

```

## Influência de um usuário

- ```

SELECT COUNT(DISTINCT(id_seguidor)) AS influencia FROM (
(SELECT * FROM seguidores s WHERE s.id_segue = ?) UNION (SELECT * FROM seguidores s WHERE s.id_segue IN (
SELECT s2.id_seguidor FROM seguidores s2 WHERE s2.id_segue = ?) ) ) as a

```

## Similaridade



- `select *, similaridade(u.id,?) as similaridade from usuario u order by similaridade desc limit 10;`
- `CREATE OR REPLACE FUNCTION similaridade(id1 integer, id2 integer) RETURNS double precision AS $BODY$ DECLARE simi FLOAT; num_republicacoes_u1 INT; num_republicacoes_u1_e_u2 INT; num_comentarios_u1 INT; num_comentarios_u1_e_u2 INT; BEGIN if id1 = id2 then return 0; end if; select count(*) into num_republicacoes_u1 from post where id_autor = id1 and republicacao = true; select count(*) into num_republicacoes_u1_e_u2 from post where id_autor = id1 and id_repub in( select id_repub from post where id_autor = id2 and republicacao = true) and republicacao = true; select count(*) into num_comentarios_u1 from comentarios where dono = id1; select count(*) into num_comentarios_u1_e_u2 from comentarios c join post p on c.id_post = p.id where dono = id1 and id_post in ( select distinct(id_post) from comentarios c join post p on c.id_post = p.id where dono = id2); if num_republicacoes_u1 = 0 or num_comentarios_u1 = 0 then return 0; end if; simi := ( (num_republicacoes_u1_e_u2)/(num_republicacoes_u1) + (num_comentarios_u1_e_u2)/(num_comentarios_u1) ) / 2.0; RETURN simi; END; $BODY$ LANGUAGE plpgsql;`