

11208012 Caio Rodrigues
11208030 Felipe Furquim
10843552 Stefany Ramos
10723836 Renan Nakazawa

ARTEFATO B (UMA RESTRIÇÃO DE SEGURANÇA):

- Caso um usuário queira que partes da sua watchlist não sejam públicas, teremos uma visão que não contará itens privados dentro dela. Além disso, itens na visão terão as suas notas médias disponibilizadas.

- Solução em código Postgres (mesma solução do SQL padrão):

```
CREATE VIEW watchlist_publica AS
SELECT u.nickname, o.nome AS nome_obra, o.nota_media FROM usuario
u
INNER JOIN watchlist w ON u.id = w.usuario_id
INNER JOIN obra o ON w.obra_id = o.id
WHERE w.publico IS TRUE;
```

- Transcrição do código:

Cria uma view que projeta uma relação entre usuários, obras e watchlists, mostrando somente os itens que estão especificados como públicos

- Relação custo/benefício desta visão + discussão sobre a possibilidade de permitir atualização de dados via esta visão

O atributo nota_media da tabela obra é um atributo calculado a partir de um **TRIGGER AFTER INSERT** na tabela avaliacao, e em nosso cenário hipotético é uma das tabelas que mais possui eventos de inserção. O PostgreSQL possui uma restrição para atualização de visões em consultas que envolvem mais de uma tabela, portanto não é possível realizar um UPDATE automaticamente na view:

DETAIL: Views that do not select from a single table or view are not automatically updatable.

Caso quiséssemos atualizar a nota média da view a cada inserção, teríamos alguns problemas de performance no banco, dado que o trigger que atualiza esse atributo possui uma função de agregação (**AVG**), que requer uma busca sequencial na tabela avaliacao, já que o atributo nota não está indexado, conforme evidenciado no plano de consulta:

```
Update on obra (cost=22.93..30.95 rows=1 width=102)
InitPlan 1 (returns $0)
```

```
-> Aggregate (cost=22.77..22.78 rows=1 width=32)
    -> Seq Scan on avaliacao a (cost=0.00..22.75 rows=5 width=4)
        Filter: (obra_id = 1)
    -> Index Scan using obra_pkey on obra (cost=0.15..8.17 rows=1
width=102)
        Index Cond: (id = 1)
```

No entanto, dado que a tabela de usuário possui dados sensíveis como email e senha, não deveríamos expor essas informações em uma busca tão comum quanto uma busca por itens de uma watchlist, por isso essa visão de restrição se faz necessária.

Como a atualização da visão iria alterar somente a nota_media, poderíamos atualizá-la periodicamente através de uma procedure periódica ao invés de atualizar a cada inserção ou remover o atributo da visão, e consultar a nota média a cada requisição da visão.