## PROJETO BASE DE DADOS 2018/19

NOME NÚMERO ESFORÇO

Francisco Santos 86421 10 horas

João Torres 87540 10 horas

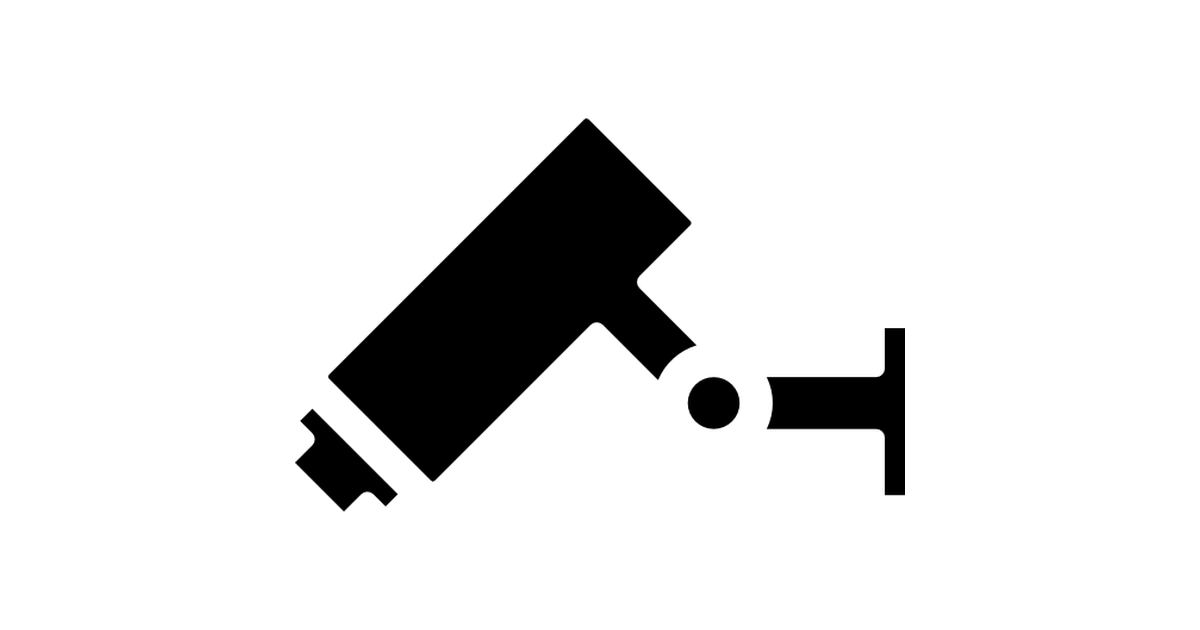
Yuri Dagot 82318 10 horas

Grupo 81 - Turno – Quarta Feira 09:30

LEIC-A

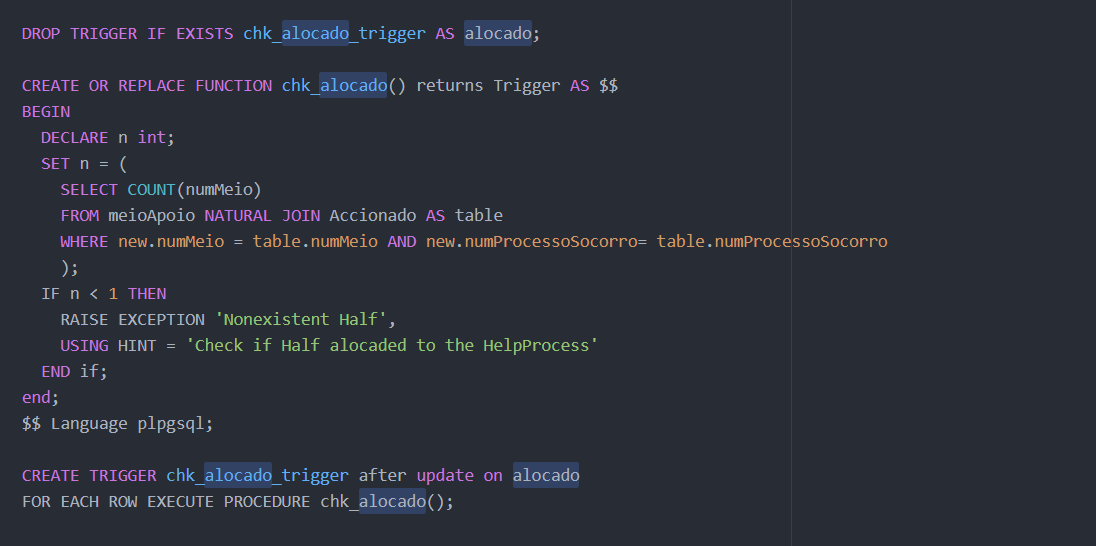
INSTITUTO SUPERIOR TÉCNICO

Relatório Parte 4

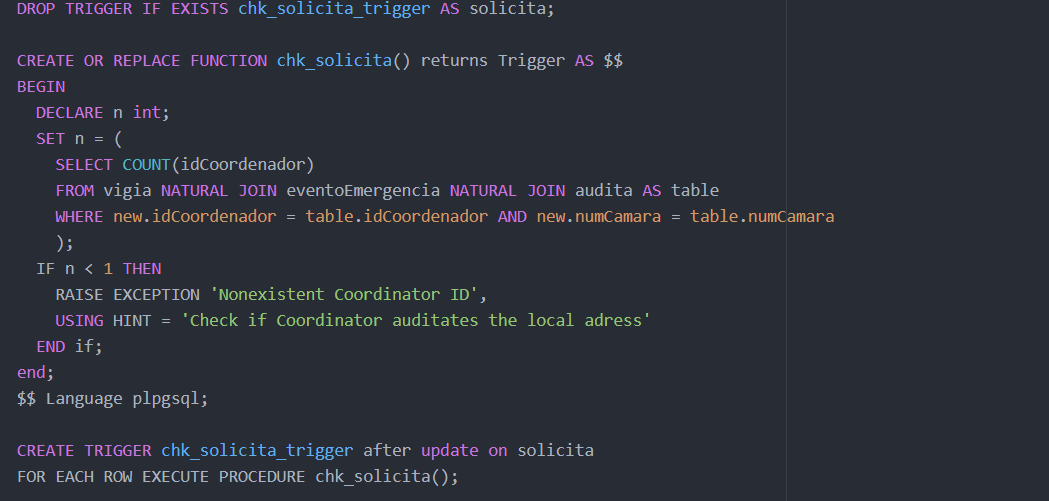


## RESTRIÇÕES DE INTEGRIDADE

a)



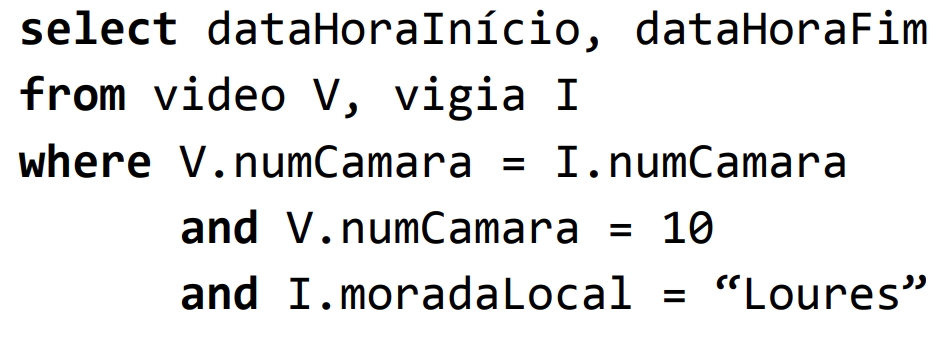
b)



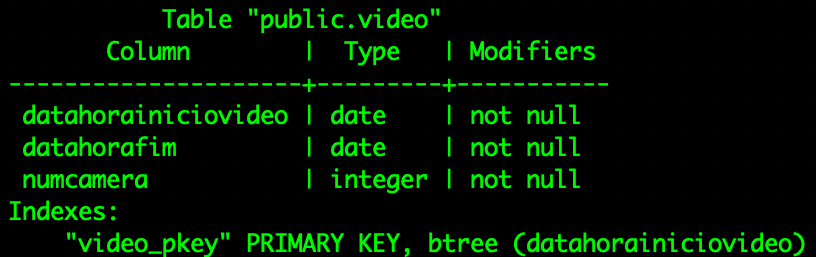
## INDICES

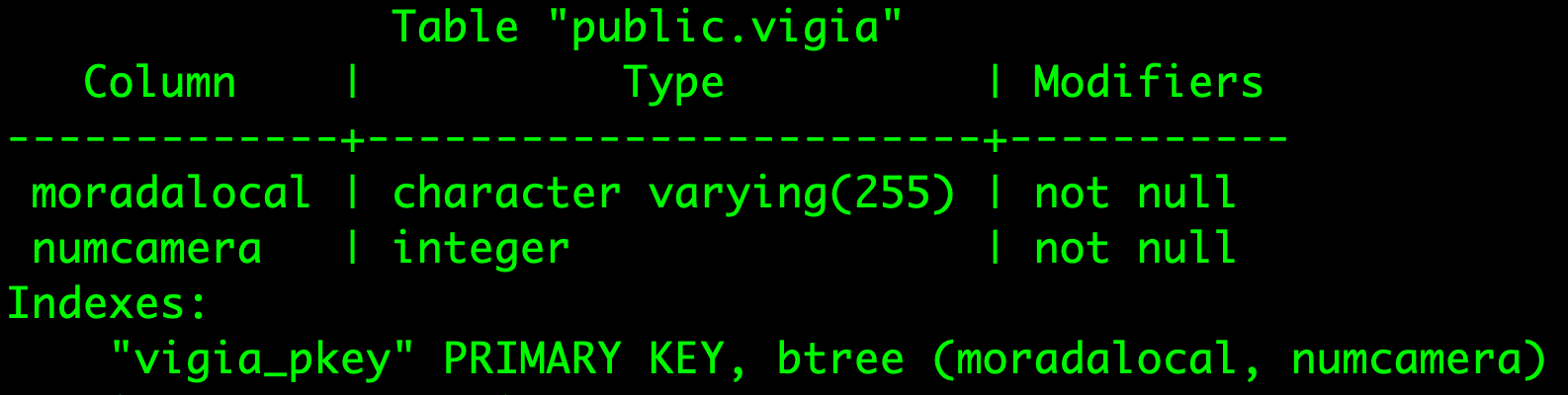
Os índices são uma maneira comum de melhorar o desempenho de uma base de dados. Um índice permite que o servidor de uma base de dados localize linhas específicas muito mais rápido do que seria possível sem um índice. Mas os índices também adicionam sobrecarga ao sistema de banco de dados como um todo, portanto devem ser usados ​​de maneira sensata e eficiente.





* 1. Para acelerar esta interrogações deve podem ser utilizados 2 tipos de índice: um HASH ou um BITMAP pois existe uma comparação de atributos e também uma condição AND que pode ser utilizada por ambos. Para esta alínea vamos utilizar o HASH. Este índice optimiza queries que tenham testes de igualdade. Os atributos numCamera e moradaLocal fazem parte da chave primária de Camera e Vigia, assim pode-se acelerar o processo de pesquisa, considerando um número grande de entradas com a mesma moradaLocal de um dado evento!





Quando é criada a base de dados no postgreSQL, automaticamente é fornecido um índice BTree a tabela. Neste caso iremos criar outro índice HASH (Já que o BITMAP não é suportado pelo postgreSQL) como foi o escolhido na alínea anterior para poder comparar com o antigo índice e ver se houve melhoria.

Para a criação do índice HASH em video:

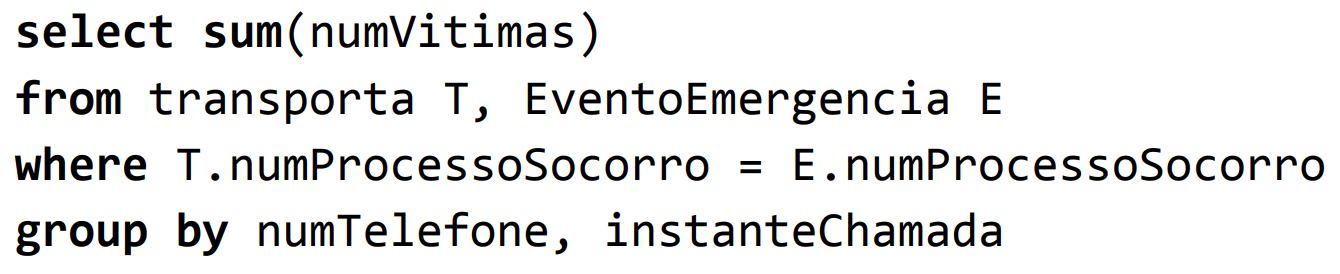
CREATE INDEX num\_camera\_indx ON video USING HASH(numCamera);

INDICE BTree no Video -> Time: 1.581ms

INDICE HASH no Video -> Time: 0.756ms

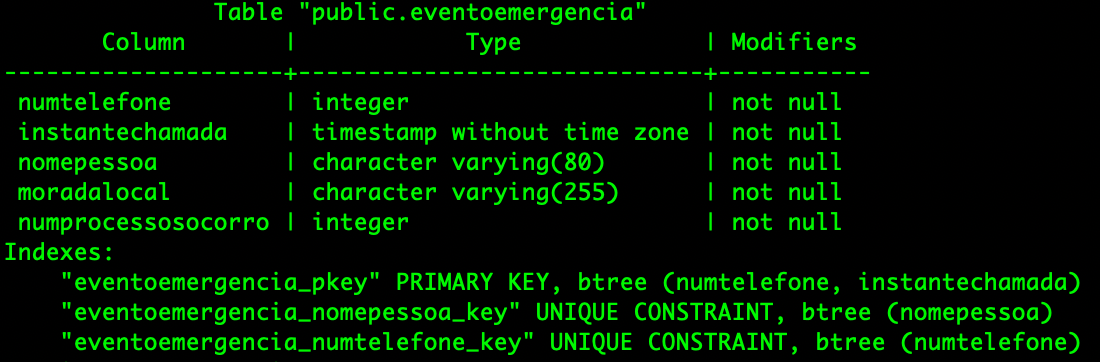
Apesar de a nossa base de dados ser pequena, podemos verificar uma certa melhoria com os testes antes e depois de criado o índice HASH.

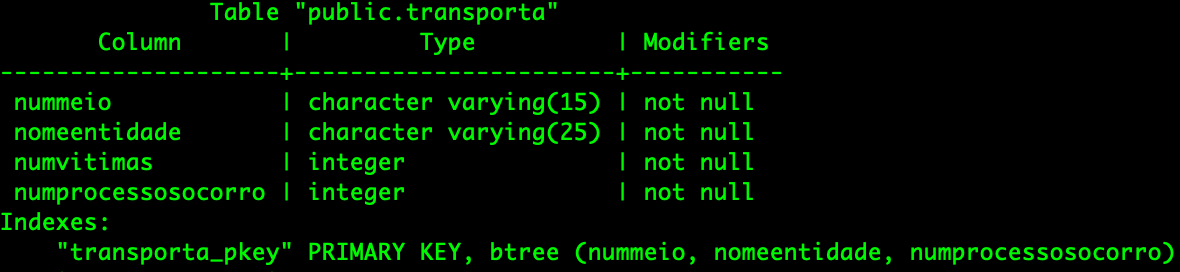
Para a tabela Vigia, o método de criação é igual ao do Video e também pode se verificar uma descida/melhoria de tempo de execução.



* 1. Para acelerar esta interrogação deve ser utilizado o índice HASH de chave de pesquisa <numTelefone, instanteChamada> nas tabelas Transporta e EventoEmergencia. Como o WHERE é feito com uma relação de igualdade, onde T.numProcessoSocorro = E.numProcessoSocorro torna a procura mais eficiente do que o índice BTree já criado pela tabela inicialmente.

Como os atributos que são utilizados são chaves primárias da sua respetiva tabela, então não é necessário percorrer toda a tabela para obter os resultados necessários.





Tal como em 1.b), quando são criadas as tabelas, por definição os postgreSQL, fornece automaticamente um índice BTree. Para uma melhor eficácia iremos tratar com um índice HASH.

Para a criação do índice HASH em transporta:

CREATE INDEX numProcSoc\_indx ON transporta USING HASH(numProcessoSocorro);

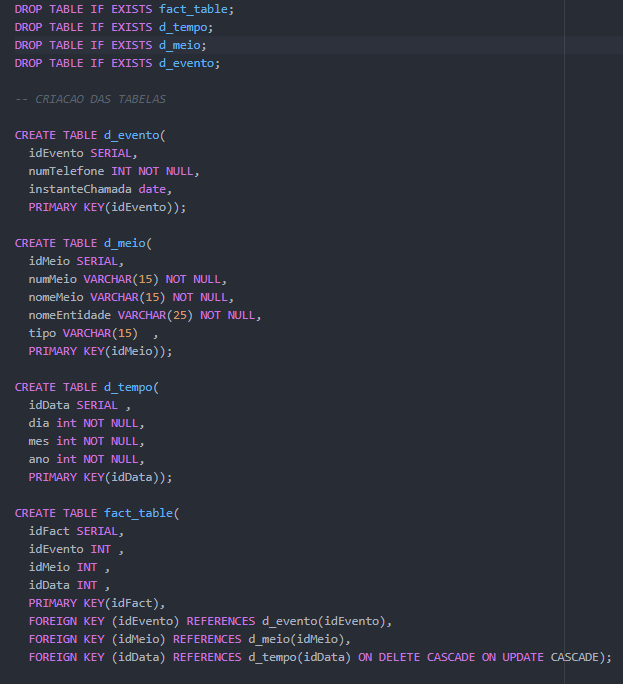
INDICE BTree no transporta -> Time: 3.609 ms

INDICE HASH no transporta -> Time: 1.7544ms

Apesar de a nossa base de dados ser pequena, podemos verificar uma certa melhoria com os testes antes e depois de criado o índice HASH.

# DATA WAREHOUSE

Criação das Tabelas



Populate

