Projeto de BD – Parte 3 Grupo 12

Inara Parbato n° 91110

Jenisha Lalgi n°89467

Rodrigo Gomes n° 92548

Turno de 3a Feira às 15h:00 e 6a Feira às 17h:30 – BD2L07 **Docente:** Rodrigo Borges Pessoa de Sousa

Horas de trabalho por elemento: 9h Percentagem de contribuição por elemento: 33.3%

2-Restrições de integridade

(RI-5): Para o desenvolvimento desta restrição, assim como para o resto do projeto, assumimos que a tabela tem_categoria tem todas as combinações de um dado produto e as suas possíveis categorias diversas, sendo cada combinação uma diferente linha na tabela. Para além disso, assumimos que o atributo cat da tabela produto indica a categoria principal deste e é uma exceção à suposição anterior (esta relação não se encontra como entrada na tabela tem_categoria).

3-Queries:

Para o desenvolvimento da primeira e segunda query, assim como para o resto do projeto, assumimos que a tabela responsavel_por tem uma entrada diferente para cada categoria em cada IVM que um determinado retalhista seja responsável por.

Índices

7.1 – Podemos otimizar esta query através do uso de índices do tipo hash tanto para o atributo tin da tabela retalhista como para o atributo nome_cat da tabela responsável_por. Com a criação do segundo índice podemos obter um tempo de procura de O(1) para a segunda parte da condição, resultante de um index only scan utilizando o hash do nome_cat. Ao mesmo tempo, com a criação do primeiro índice conseguimos reduzir bastante o número de linhas a serem acedidas para a primeira parte da condição, através do uso de um inner index scan. Poderíamos ainda ter usado índices do tipo Btree para ambas as partes da condição, contudo, tratando-se ambas especificamente de igualdades, índices do tipo hash tornam-se ligeiramente vantajosos porque resultam num tempo de procura menor.

```
CREATE INDEX tin_index ON retalhista using hash(tin);
CREATE INDEX cat_index ON responsavel_por using hash(nome_cat);
```

7.2 – Podemos otimizar esta query através da criação de um índice do tipo Btree para o atributo nome da tabela tem_categoria pois este se traduziria numa otimização na verificação da primeira condição e, mais importantemente, no group by, sendo esta última a razão principal para a nossa decisão de usar esse tipo de índice. Adicionalmente, podemos também criar outro índice para o atributo cat da tabela produto para otimizar ainda mais a verificação da primeira condição, no entanto, já que agrupar este não consiste em qualquer vantagem, um índice do tipo hash é preferível tratanto a condição

de uma igualdade. Por último, a criação de um índice do tipo Btree para o atributo descr da tabela do produto também seria vantajoso para a verificação da segunda condição, e seria preferível a um índice do tipo hash pois trata-se de uma procura não de um valor em concreto, mas sim de um conjunto de valores (range search) na qual os índices Btree são mais úteis e eficientes.

```
CREATE INDEX nome_index ON tem_categoria(nome);
CREATE INDEX cat_index ON produto USING hash(cat);
CREATE INDEX desc_index ON produto(descr);
```

Aplicação

(Não temos a aplicação complete e apesar de termos colocado o link, o mesmo aparece como INTERNAL SERVER ERROR).

https://web2.ist.utl.pt/ist189467/app.cgi/– página inicial da aplicação onde podemos verificar as varias operações que podem ser desenvolvidas na aplicação da IVM, como verificar as categorias.

Este encaminha para uma página onde é apresentada/listada todas as categorias presentes nas IVMs. Ainda apresenta a opção de podermos inserir uma categoria ou remover.


{% if cursor %} {% for record in cursor %} {% endfor %}
Nome
{{ record[0] }}
{% else %}
{% endif %}
Erro: ao obter dados da base de dados!
Inserir Categoria
Remover Categoria