Projeto de BD - Parte 3

Grupo 002 — Turno L13 — LEIC-A

Prof. Flávio Martins

Gonçalo Bárias (103124) - 33.3$\overline{3}$% - 36h

Raquel Braunschweig (102624) - 33.3$\overline{3}$% - 36h

Vasco Paisana (102533) - 33.3$\overline{3}$% - 36h

Desenvolvimento da Aplicação

Índices

Foi pedido, ainda, que se indicasse (justificando) os índices que faria sentido criar, por forma a *agilizar* a execução de cada uma das *queries* apresentadas de seguida.

Primeira *query*:

```
SELECT order_no
FROM orders
    JOIN contains USING (order_no)
    JOIN product USING (SKU)
WHERE price > 50 AND
    EXTRACT(YEAR FROM date) = 2023;
```

Para a primeira *query*, optou-se por criar dois índices: um no atributo `price` da relação `product`, e outro no ano do atributo `date` da relação `orders`.

- Não foi criado nenhum índice para os atributos `order_no` e `SKU` de `contains`, pois o `PostgreSQL` já cria um índice **BTree** para (`order_no`, `SKU`) dado que se trata de uma chave primária. O `planner` é capaz de usar este índice e assim não se justifica criar dois índices separados para estes atributos apenas devido às clásulas `JOIN`.
- Optámos pela criação de um índice **BTree** no atributo `price` de `product`, pois a comparação pretendida engloba um intervalo de preços e assim um índice **Hash** não seria particularmente inteligente, já que não se pretende um único preço em concreto.
- Por fim, criámos um índice para o ano do atributo `date` de `orders`, pois estando a fazer uma comparação de igualdade no ano das datas das encomendas, faz todo o sentido usar um índice **Hash** já que a comparação em $O(1)$ é ideal.

O trecho de código correspondente à indexação pretendida encontra-se abaixo:

```
DROP INDEX IF EXISTS product_price_index;
DROP INDEX IF EXISTS order_date_index;

CREATE INDEX product_price_index
    ON product USING BTREE(price);
CREATE INDEX order_date_index
    ON orders USING HASH(EXTRACT(YEAR FROM date));
```

Segunda *query*:

```
SELECT order_no, SUM(qty * price)
FROM contains
    JOIN product USING (SKU)
WHERE name LIKE 'A%'
GROUP BY order_no;
```

Para a segunda *query*, optou-se por criar um índice no atributo `name` da relação `product`.

- Aqui a clásula `GROUP BY` beneficia de um índice **BTree** em `order_no` de `contains`, pois o `GROUP BY` procura agrupar os dados sobre o atributo `order_no` e os índices **BTree** já vêm, por natureza, ordenados. Porém, tal como no ponto anterior, o `planner` é capaz de usar o índice criado na chave primária de `contains` para agilizar a computação do `GROUP BY`, sendo esse já um índice **BTree**.
- Assim, apenas é necessário criar um índice **BTree** para o atributo `name` de `product`, pois a comparação pretendida engloba todo um intervalo de nomes de produtos que começam pela letra `A`, logo um índice **Hash** não ajudaria.

O trecho de código correspondente à indexação pretendida encontra-se abaixo:

```
DROP INDEX IF EXISTS product_name_index;

CREATE INDEX product_name_index
    ON product USING BTREE(name);
```