UNIVERSIDADE da MADEIRA

ESCOLA SUPERIOR DE TECNOLOGIAS E GESTÃO

Ano Letivo 2019/2020

Curso Técnico Superior Profissional em: Tecnologias e Programação de Sistemas de Informação

1 º **Ano**/2 º **Semestre**

Unidade Curricular: Sistemas Gestores de Bases de Dados I

Docente: Magno Andrade **Época:** Normal

INDEXAÇÃO

ÍNDICES

Uma das maiores confusões que existe entre os utilizadores de bases de dados reside na confusão entre os termos ordenação e indexação.

Pensemos por momentos no serviço telefónico de informação, que permite obter números de telefone por nome ou morada, nomes e moradas por número de telefone, e assim por adiante. Imaginemos que para este serviço existe uma aplicação com uma base de dados que faz esta gestão. Esta base de dados tem 30 milhões de registos.

Esta lista de registos pode ou não estar ordenada. Em princípio a lista deverá estar ordenada, pois só assim será possível evitar a pesquisa de um registo sem ter que percorrer toda a lista. Se está ordenada, está ordenada por que campo? Nome? Morada? Telefone?

A lista se está ordenada, só poderá sê-lo por um único critério, isto é, se estiver ordenada pelo nome, não poderá estar ordenada pelo telefone e vice-versa. Portanto, é aqui que é encontrado um problema, pois se a lista estivesse ordenada, cada novo registo introduzido obrigaria a uma remodelação quase total da lista, o que seria impraticável.

Para ter múltiplas perspectivas de ordenação sobre esta tabela da base de dados, os registos quando são adicionados são sempre colocados no final dos mesmos. Evita-se assim, que sejam realizadas pesadas reorganizações sistemáticas da tabela sempre que um novo registo é introduzido.

Podemos imaginar a reorganização de uma base de dados com 30 milhões de registos após a inserção de um novo registo de Nome “Ana”.

Sendo que os registos colocados estão sempre no final da tabela terá que existir outra forma para simular uma determinada ordenação sobre os dados.

A **ordenação** obriga que os registos estejam fisicamente armazenados pela ordem definida para serem guardados.

Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

A **indexação** consiste numa estrutura adicional que permite simular a ordenação dos dados. Na realidade, baseia-se normalmente numa "árvore" de termos ou valores que apontam para uma determinada posição na tabela.

Uma **árvore binária** não é mais do que uma estrutura em que, para cada um dos nós da árvore, todos os nós à esquerda são menores ou iguais à chave de indexação e todos os nós à direita são superiores à chave de indexação.

Em cada um dos nós vamos guardar a chave de indexação e o número de registo.

ID	Nome	Apelido
2456	Célia	Morais
4561	José	Lopes
6452	Florinda	Simões
1289	António	Dias
4978	Beatriz	Costa
3254	Ana	Rita
5698	Paulo	Viegas

Tabela 1 - Conjunto de dados desordenados.

Para sabermos se a pessoa com o nome "Glória" existe na tabela sem usar a indexação, seriam realizadas um conjunto de leituras sequenciais de dados para chegar ao resultado:

1. Ler registo. Nome (Célia) é igual a Glória? Não.
2. Ler registo. Nome (José) é igual a Glória? Não.
3. Ler registo. Nome (Florinda) é igual a Glória? Não.
4. Ler registo. Nome (António) é igual a Glória? Não.
5. Ler registo. Nome (Beatriz) é igual a Glória? Não.
6. Ler registo. Nome (Ana) é igual a Glória? Não.
7. Ler registo. Nome (Paulo) é igual a Glória? Não.
8. Ler registo. Fim de tabela, logo, o registo não existe.

Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

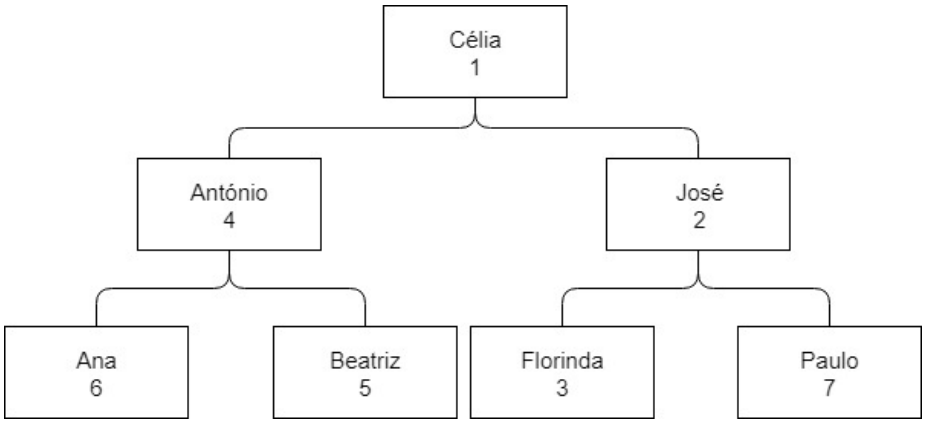

Figura 1 – Árvore binária dos nomes presentes na Tabela 1.

Se optarmos por usar o índice, o conjunto de operações era o seguinte:

1. Ler registo. Nome (Célia) é igual a Glória? Não.
    a. Como Glória é alfabeticamente maior que Célia (pois "G" vem depois da letra "C" no alfabeto), vamos passar para o nó-filho direito de Célia.
2. Ler registo. Nome (José) é igual a Glória? Não.
    a. Como Glória é alfabeticamente menor do que José, vamos passar para o filho esquerdo de José.
3. Ler registo. Nome (Florinda) é igual a Glória? Não.
    a. Como Glória é alfabeticamente maior do que Florinda, vamos passar para o filho direito de Florinda.
4. Ler registo. Fim de árvore, logo, o registo não existe.

Como se pode confirmar, o conjunto de operações é muito menor do que o que seria executado caso se optasse por percorrer um ficheiro de dados de forma sequencial.

Existem três tipos de índices:

- Simples (apenas numa coluna).
- Composto (mais do que uma coluna).
- Único (não permitir valores duplicados)

COMANDOS PARA ÍNDICES

CREATE INDEX - para criar um índice numa tabela, valores duplicados são permitidos –

Sintaxe:

```
CREATE INDEX nome_do_index
ON nome_da_tabela (coluna1, coluna2, …);
```

Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

Ex: CREATE INDEX idx_nome

ON pessoa (nome);

CREATE UNIQUE INDEX - para criar um índice numa tabela, valores duplicados não são permitidos –

Sintaxe:

CREATE UNIQUE INDEX nome_do_index

ON nome_da_tabela (coluna1, coluna2, …);

DROP INDEX – para apagar um índice numa tabela –

Sintaxe:

ALTER TABLE nome_da_tabela

DROP INDEX nome_do_index;

Ex: ALTER TABLE pessoa

DROP INDEX idx_nome;

SHOW INDEXES – para apresentar os índices existentes numa tabela –

Sintaxe:

SHOW INDEXES FROM nome_da_tabela;

Ex: SHOW INDEXES FROM pessoa;

Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu