

Megadados

Índices, particionamento

Engenharia

Maciel C. Vidal

Indices

Índices

Índices são estruturas de dados que facilitam a localização de informação no banco de dados.

Cria-se um índice a partir de uma coluna de uma tabela.

Quando criar índice?

Sempre que:

- Chave primária
- Chave estrangeira
- Colunas
 - Com muita demanda de busca,
 - sem muita repetição de valores,
 - sem muitos NULL, e que
 - não mudam com frequência (preferencialmente nunca)

Quando não criar indice

- Colunas que mudam com frequencia
- Colunas com muita repetição de valores
- Colunas com muitos NULL
- Colunas que não são usadas com frequencia em buscas

Alem disso:

- Não crie mais que 6 indices por tabela

CREATE INDEX

```
CREATE INDEX index_name [index_type]  
    ON tbl_name (index_col_name,...)
```

index_type:

USING {BTREE | HASH}

- Como funcionam os índices? Implementações mais comuns:
 - Hash tables
 - B-trees

CREATE INDEX

```
USE megadados;  
CREATE INDEX idxTelefone  
ON Funcionario (telefone);
```

Quando criar?

- Chave primária
- Chave estrangeira
- Colunas usadas frequentemente para busca

Quando não criar?

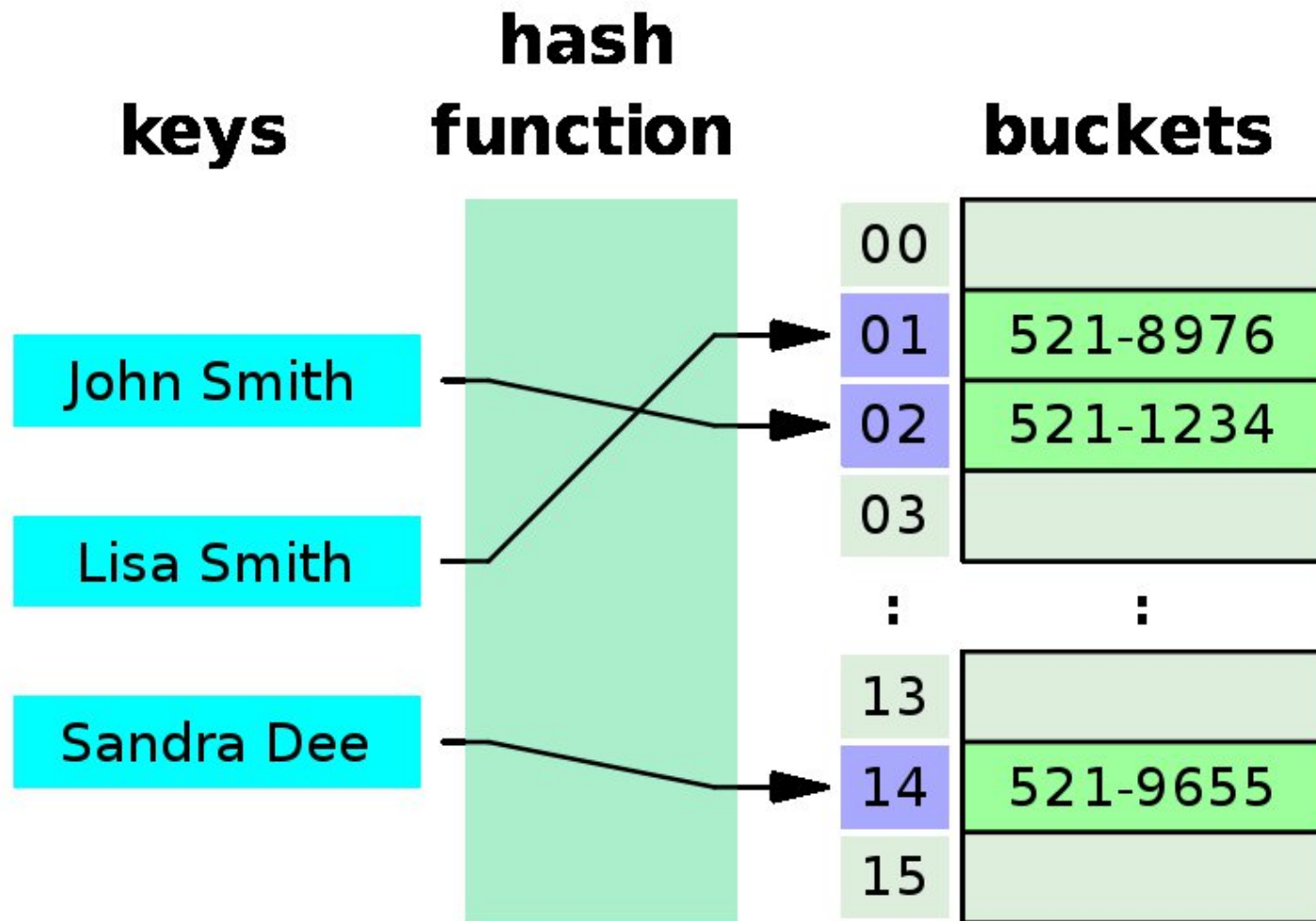
- Colunas com muitos NULL
- Colunas com muitos valores iguais
- Tabelas muito pequenas

Não crie mais do que 6 índices por tabela

Hash tables

- Mapa chave-valor organizados pelo valor hash da chave
- Bom para buscas por valor exato
- Ruim para buscas por faixas de valor

Example



Complexidade

Lembrando de “Desafios de Programação”:

- Busca? ?
- Inserção? ?
- Remoção? ?

Complexidade

Lembrando de “Desafios de Programação”:

- Busca? $O(1)$
- Inserção? $O(1)$
- Remoção? $O(1)$

B-trees

Um tipo de árvore balanceada de busca

- Razoável para buscas por valor exato
- Boa para buscas por faixa de valor

O professor demonstrará o processo de construção da árvore.

Complexidade das B-trees

- Busca? ?
- Inserção? ?
- Remoção? ?

Complexidade das B-trees

- Busca? $O(\log(n))$
- Inserção? $O(\log(n))$
- Remoção? $O(\log(n))$

Particionamento

Particionamento

Particionar é dividir as tabelas de um banco de dados em partes menores.

Permite distribuir o banco de dados em vários nós, aumentando o desempenho em situações de acesso concorrente intenso.

Tipos de particionamento

- **Vertical**: as tabelas são **separadas por colunas** em várias tabelas com relacionamento 1:1
 - A tabela original é o **INNER JOIN** das várias tabelas derivadas
- **Horizontal**: as tabelas são **separadas por linhas** em fragmentos (*shards*). Esse procedimento também é chamado de fragmentação (*sharding*)
 - A tabela original é o **UNION** das várias tabelas derivadas

Particionamento vertical

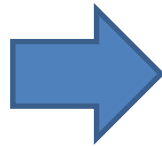
Id	Coluna_1	Coluna_2	Coluna_3
1	a	b	c
2	d	e	f
3	g	h	i
4	J	k	l



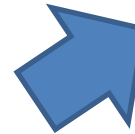
Id	Coluna_1	Id	Coluna_2	Coluna_3
1	a	1	b	c
2	d	2	e	f
3	g	3	h	i
4	J	4	k	l

Particionamento horizontal

id	colA	colB
1	a	b
2	c	d
3	e	f
4	g	h
5	i	j
6	k	l
7	m	n



Função de
partição



id	colA	colB
2	c	d
4	g	h
6	k	l

id	colA	colB
1	a	b
3	e	f
5	i	j
7	m	n



www.insper.edu.br