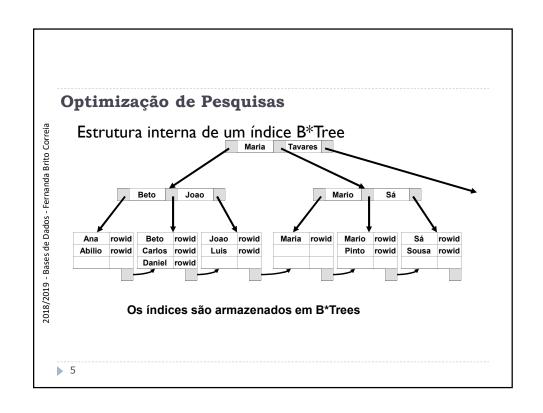
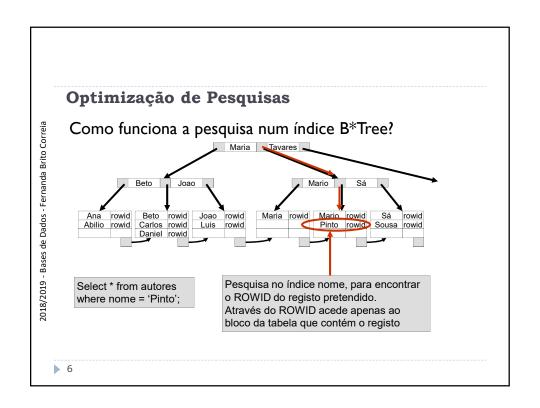


Optimização de Pesquisas • Índices • B+Tree • BITMAP

Optimização de Pesquisas Índices • Estruturas facultativas associadas a colunas de tabelas ou agrupamento de tabelas destinadas a acelerar o acesso aos dados • São o equivalente a um índice remissivo de um livro pois permitem localizar rapidamente a informação pretendida





Índices

- > podem ser construídas sobre uma ou mais coluna
- > Só pode existir um índice por coluna
- Podem ser realizados índices sobre todas as combinações das colunas de uma tabela
- Mas não podem haver combinações de colunas repetidas

7

2018/2019 - Bases de Dados - Fernanda Brito Correia

Optimização de Pesquisas

Índices

- estruturas lógica e fisicamente independentes dos dados da tabela a que estão associados
 - ▶ a criação / apagamento de índices não interferem com os dados da tabela nem com a estrutura da base de dados
- riados pelo utilizador através de comandos SQL
 - create index ...
 - drop index ...
 - ▶ alter index ...

8

Índices são geridos automaticamente pelo sistema

- em cada consulta (select) a tabela com índices associados, o sistema decide se usa ou não o índice para aceder aos dados
- a inserção, actualização e/ou apagamento de dados na tabela leva a que o sistema actualize, de maneira transparente para o utilizador, os índices existente sobre essa tabela
- O facto de uma tabela ter índices, não altera em nada os comandos DML que se efectuem sobre os dados da tabela
- ▶ A única diferença para o utilizador é a obtenção de resultados ás pesquisas mais rapidamente

9

2018/2019 - Bases de Dados - Fernanda Brito Correia

Optimização de Pesquisas Criação de um Índice CREATE INDEX titulos ON livros(titulo); Cria o índice títulos sobre a coluna titulo da tabela livros

Optimização de Pesquisas Vantagens da utilização de índices

- Tornam acessos aos registos muito mais rápidos
- Ex. na query sobre a tabela livros com 20 000 livros

Select titulo from livros where id livro = 9928;

- Sem índices é uma pesquisa sequencial, sendo necessário ler do disco todos os registos até encontrar o desejado
 - média de blocos lidos por query (Total de blocos / 2)
- Com índice na coluna idlivro é uma pesquisa em árvore, sendo necessário ler do disco apenas os blocos necessários do índice e o bloco da tabela que contém o registo
 - média de blocos lidos por query (N níveis + 1)

▶ 11

2018/2019 - Bases de Dados - Fernanda Brito Correia

Optimização de Pesquisas

Desvantagens da utilização de índices

- Ocupam espaço em disco
- Os acessos apenas se tornam mais rápidos se forem efectuados sobre as colunas que tem índices
- A construção de um índice (simples, não composto) para cada uma das colunas da tabela, esses índices ocupam mais espaço que a própria tabela
- Sempre que existe alteração dos dados (insert, update, delete) é necessário reestruturar todos os índices que existem sobre essa tabela
 - Processamento de transacções mais lento

Que tabelas se devem indexar?

- ▶ Tabelas grandes (com muitos registos)
- ► Tabelas em que as pesquisas sejam muito especificas (devolvam uma percentagem pequena de registos < 10%)
- ► Tabelas relativamente estáticas em que a principal operação é a pesquisa e as operações de escrita são pouco frequentes
 - operações de escrita ficam mais lentas, pois é necessário reestruturar os índices

13

2018/2019 - Bases de Dados - Fernanda Brito Correia

Optimização de Pesquisas

Que colunas se devem indexar?

- ▶ Colunas sobre as quais se fazem muitas pesquisas
- Chaves forasteiras para acelerar as junções (as chaves primárias são implicitamente indexadas, mas as chaves estrangeiras não)
- Colunas em que a maioria dos valores são distintos (não faz sentido indexar uma coluna como sexo na tabela de autores, pois apenas tem 2 estados possíveis)
- Colunas com muitos valores nulos, mas que a maioria das pesquisas são efectuadas sobre os valores não-nulos
- Não se podem indexar colunas do tipo LONG ou LONG RAW

Quando é que se deve indexar uma tabela?

- Pode-se indexar uma tabela em qualquer altura
- mas se a tabela vai receber uma grande quantidade de dados (por exemplo aquando de uma migração de dados), a indexação das colunas apenas deve ser realizada no fim da introdução dos dados
 - Se não por cada registo migrado terá que existir uma restruturação de cada um dos índices que existir sobre a tabela

Quantos índices pode haver por cada tabela?

 Pode haver qualquer número de índices por tabela, mas há que não esquecer de que quantos mais índices mais lentas se tornam as operações DML

15

2018/2019 - Bases de Dados - Fernanda Brito Correia

Optimização de Pesquisas

Ordem das colunas na criação dos índices?

- Exemplo : prod_lojas(id_prod, id_loja, preço, quantidade)
- Supondo que :
 - a tabela pode ter algumas dezenas de lojas, mas que cada loja (id_loja) pode ter milhares de produtos (ld_prod)
 - Há consultas frequentes do tipo

```
select * from prod_lojas where id_prod = 1243 and id_loja =8;
```

Neste caso o índice composto a criar deveria indicar primeiro a coluna com mais valores distintos (idprod)

CREATE INDEX Prod_loj_id ONprod_lojas(idprod, idloja);

16

2018/2019 - Bases de Dados - Fernanda Brito Correia

Eliminação de índices

DROP INDEX prod_loja_id;

- A eliminação de índices não altera a base de dados, só afecta a velocidade a que as operações passam a ser realizadas
- Quando se eliminam uma tabela todos os índices a ela associados também são eliminados

▶ 17

2018/2019 - Bases de Dados - Fernanda Brito Correia

Optimização de Pesquisas

Porque se eliminam índices?

- O índice não levou à melhoria de velocidade que se esperava
- As pesquisas efectuadas não permitem ao sistema a utilização dos índices
- A tabela vai passar por uma fase de inserção / actualização intensiva, pelo que é melhor eliminar o índice para o voltar a reconstruir mais tarde
- O índice está muito fragmentado, pelo que é melhor eliminá-lo e depois reconstruí-lo

2018/2019 - Bases de Dados - Fernanda Brito Correia

Tipos de índices

- ▶ **B***Tree
 - índices mais habituais, informação guardada numa estrutura em árvore

BITMAP

• índices muito utilizados para atributos com pouca cardinalidade (poucos valores distintos para esse atributo)

19

Optimização de Pesquisas

Características da B* Tree

- Dados encontram-se apenas nas folhas, sempre á mesma profundidade
- Pesquisa de qualquer registo demora sempre aproximadamente o mesmo tempo
- Os índices estão sempre ordenados para permitir acessos mais rápidos
- A B*Tree está sempre equilibrada pelo que após a inserção, alteração ou apagamentos de dados o sistema pode reestruturar a B*Tree
- Para minimizar o tempo na reestruturação da B* Tree, os blocos estão em média ocupados até 3/4 do seu espaço

20

2018/2019 - Bases de Dados - Fernanda Brito Correia

Criação implícita de índices B*Tree

- ao definir-se uma chave primária a coluna (ou colunas) que constituem a chave são automaticamente indexadas
 - (Joins entre tabelas normalmente são efectuadas através das chaves)
- o mesmo se aplica para as constraints com restrição UNIQUE (chaves candidatas)
- também é possível construir índices compostos de várias colunas

CREATE INDEX aut_liv ON livros(idautor, idlivro);

21

2018/2019 - Bases de Dados - Fernanda Brito Correia

Optimização de Pesquisas

Índices BITMAP

- são uma alternativa muito poderosa aos índices BTree para determinadas situações muito específicas
- realizado sobre colunas com muitos valores repetidos (baixa cardinalidade mas com poucos valores distintos.
 - Por exemplo:
 - > sexo: Masculino, Feminino
 - Estado civil: Solteiro, Casado, Separado, Divorciado, Viúvo
- Nestes casos torna-se impraticável a construção dum B-Tree sobre estas colunas pois ficariam com muitos ROWID para cada um dos valores distintos

22

Como funcionam os índices BITMAP

- Para cada ROWID distinto é adicionado Nbits que representam cada um dos estados possíveis.
 - Por exemplo:
 - ▶ Na tabela de autores com 10 000 registos
 - é criado um índice na coluna sexo contendo os ROWID e
 2 bits
 - é criado um índice na coluna estado_civil contendo os ROWID e 5 bits

> 23

2018/2019 - Bases de Dados - Fernanda Brito Correia

Optimização de Pesquisas

Vantagens dos índices BITMAP?

- Para colunas com pouca cardinalidade, permite pesquisas mais rápidas e ocupam menos espaço que os índices BTree
- Mecanismo que além de permitir uma maior rapidez na procura de registos de um determinado valor,
- permite uma maior rapidez na combinação de valores pois a sua estrutura facilita o OU e o E lógico
- Impraticável em colunas com muita cardinalidade

2018/2019 - Bases de Dados - Fernanda Brito Correia

Clusters

2018/2019 - Bases de Dados - Fernanda Brito Correia ► Clusters (indexados)

- Introdução
- Gerir clusters
- ▶ Criar e apagar clusters

▶ Hash clusters

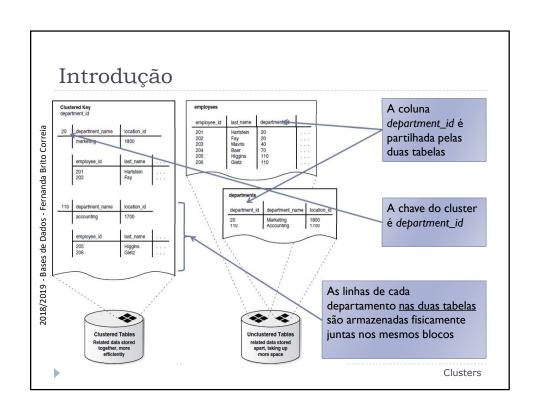
- Introdução
- Quando usar
- ▶ Criar e apagar hash clusters

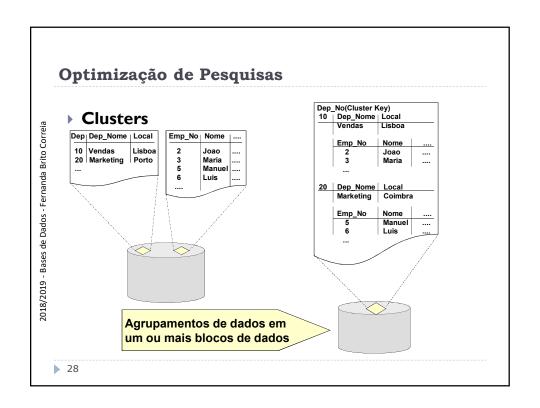
Clusters

Introdução

2018/2019 - Bases de Dados - Fernanda Brito Correia Clusters

- Forma alternativa de armazenar os dados das tabelas de uma BD
- Constituído por grupos de tabelas que partilham os mesmos blocos de dados
 - As tabelas juntam-se porque
 - □ Têm colunas em comum
 - $\hfill\Box$ São frequentemente usadas em conjunto





Introdução – Vantagens

Porque as linhas de tabelas diferentes relacionadas entre si são armazenadas juntas:

- lunções de tabelas em clusters são mais eficientes
 - Menos operações de I/O no disco
 - Melhores tempos de acesso
- Espaço necessário é menor
 - Cada valor da chave do cluster é armazenada apenas uma vez
 - ▶ Independentemente do n° de linhas que contenham esse valor

Clusters

Introdução - Funcionamento

Depois de criado um cluster

- ▶ Podem ser criadas tabelas no cluster
- Só podem ser inseridas linhas nas tabelas
 - Depois de ser criado um índice no cluster (num cluster indexado)
- Podem ser criados índices adicionais sobre as tabelas
 - Da forma habitual
- Não devem ser criados clusters para tabelas que sejam acedidas individualmente com frequência

Clusters

2018/2019 - Bases de Dados - Fernanda Brito Correia

Gerir clusters

 Algumas linhas orientadoras para a gestão dos clusters

- Escolher as tabelas a integrar no cluster
- Escolher as colunas para a chave do cluster
- Espaço necessário para uma chave e linhas associadas
- Localização do cluster

2018/2019 - Bases de Dados - Fernanda Brito Correia

2018/2019 - Bases de Dados - Fernanda Brito Correia

Clusters

Escolher as tabelas

▶ Faz sentido incluir tabelas que:

- Sejam principalmente consultadas
 - Não sejam principalmente objecto de inserções ou alterações
- ▶ Em que os registos
 - > São frequentemente consultados em conjunto
 - Du são objecto de junções

Escolher as colunas para a chave

- > Se as pesquisas sobre as tabelas usam várias colunas na junção
 - Criar uma chave de cluster composta
- Uma boa chave tem valores únicos suficientes para:
 - Um grupo de linhas correspondente a um valor da chave ocupa aproximadamente um bloco de dados
 - Poucas linhas por cada valor da chave desperdiçam espaço
 - Muitas linhas por cada chave implicam mais tempo de pesquisa
- O índice do cluster não pode ser UNIQUE nem ter colunas do tipo LONG

Clusters

Espaço necessário para uma chave

▶ O comando CREATE CLUSTER tem uma cláusula SIZE

- N° estimado de bytes necessários para um valor da chave e respectivas linhas
- Usado para
 - Estimar o nº de valores da chave (e respectivas linhas) que cabem num bloco de dados
 - Limitar o número de valores da chave colocados num bloco
 - □ Maximiza a eficiência de armazenamento
 - Por defeito, é colocado apenas um valor da chave em cada bloco

Clusters

2018/2019 - Bases de Dados - Fernanda Brito Correi

Apagar clusters

2018/2019 - Bases de Dados - Fernanda Brito Correia Quando um cluster é apagado

- ▶ São apagadas as tabelas que contenha
- ▶ É apagado o índice correspondente
- Apagar um cluster sem tabelas e respectivo índice

DROP CLUSTER emp dept;

Clusters

Apagar clusters (cont.)

▶ Se o cluster tiver tabelas

DROP CLUSTER emp dept INCLUDING TABLES;

- Se o cluster tiver tabelas e não for incluída a cláusula **INCLUDING TABLES**
 - ▶ Retorna um erro
- ▶ Se alguma tabela do cluster tiver colunas referenciadas por restrições FOREIGN KEY de tabelas fora do cluster

DROP CLUSTER emp_dept INCLUDING TABLES CASCADE CONSTRAINTS;

Hash Clusters

2018/2019 - Bases de Dados - Fernanda Brito Correia

 Forma alternativa de melhorar o desempenho das pesquisas

- Alternativa à criação de índices sobre tabelas
- Alternativa à criação de clusters indexados
- Nestes casos as linhas são localizadas usando valores chave guardados num índice armazenado separadamente
- Nos hash clusters o servidor localiza as linhas usando o resultado de uma função de hash

Clusters

Hash clusters 2018/2019 - Bases de Dados - Fernanda Brito Correia ▶ Função de hash Gera uma distribuição de valores numéricos com base em valores concretos da chave do cluster Segmento do cluste (...) Bloco 30 125 Função de hash 31 Registo Bloco 31 ID cliente = 125 Pretendido Bloco 32 Bloco 33 (...) Clusters

Hash clusters

2018/2019 - Bases de Dados - Fernanda Brito Correia Pesquisar ou gravar um registo

- Numa tabela ou cluster indexado
 - Um ou mais acessos ao disco para procurar ou inserir o valor chave no índice
 - Outro acesso ao disco para escrever o registo na tabela ou cluster
- Num hash cluster
 - Uso da função de hash para localizar o registo (sem acesso ao disco)
 - Apenas um acesso para ler ou gravar o registo

Clusters

Quando usar hash clusters?

▶ Situações em que é vantajoso

2018/2019 - Bases de Dados - Fernanda Brito Correia

A maioria das pesquisas têm restrições de igualdade na chave do cluster

SELECT ... WHERE cluster key = ...;

- As tabelas são eminentemente estáticas em tamanho
 - O número de registos e espaço necessário pode ser determinado à partida

Quando usar hash clusters? (cont.)

2018/2019 - Bases de Dados - Fernanda Brito Correia ▶ Situações em que <u>não</u> é vantajoso

A maioria das pesquisas devolve registos que correspondem a uma gama de valores da chave de cluster

SELECT ... WHERE cluster_key < ...;</pre>

- A tabela não é estática, mas cresce continuamente
 - Não é possível predeterminar o espaço necessário para o cluster
- Pesquisas frequentes que façam scan completo à tabela e a tabela está preenchida de forma esparsa
- Não é possível pré-alocar o espaço necessário ao cluster

Clusters

Apagar hash clusters

 Apagar hash clusters é semelhante a apagar clusters indexados

DROP CLUSTER emp dept;

DROP CLUSTER emp dept INCLUDING TABLES;

DROP CLUSTER emp dept INCLUDING TABLES CASCADE CONSTRAINTS;