

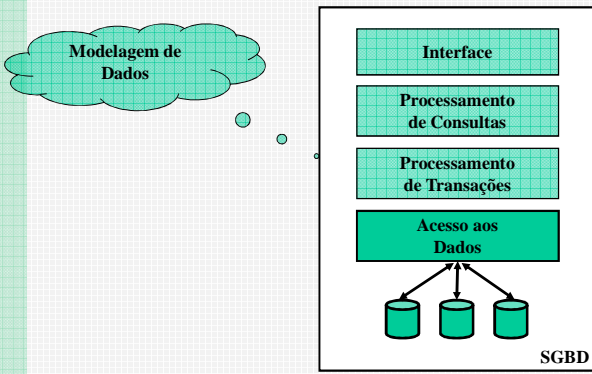
Organização e Indexação de Dados em Memória Secundária

Disciplina Bancos de Dados II
Prof. Renato Fileto
 INE/CTC/UFSC

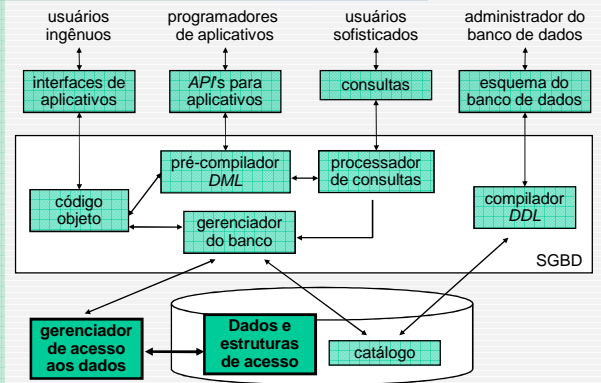
Tópicos

- Bancos de dados e acesso a memória secundária
- Noções de organização de dados em memória secundária
- Indexação
 - Índices primários
 - Índices de agrupamento (*clustering*)
 - Índices secundários
- Índices em múltiplos níveis
 - B-Trees e B+_Trees
- Bibliografia e leituras recomendadas
- Lista de exercícios

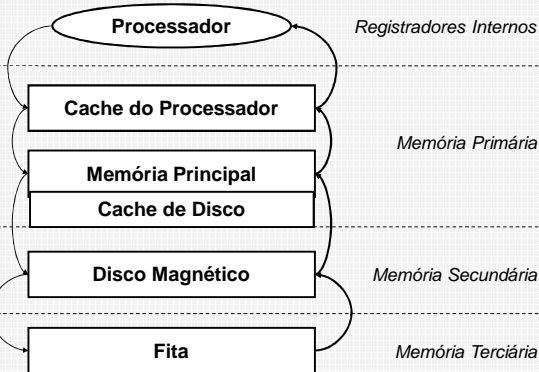
Acesso a dados em um SGBD



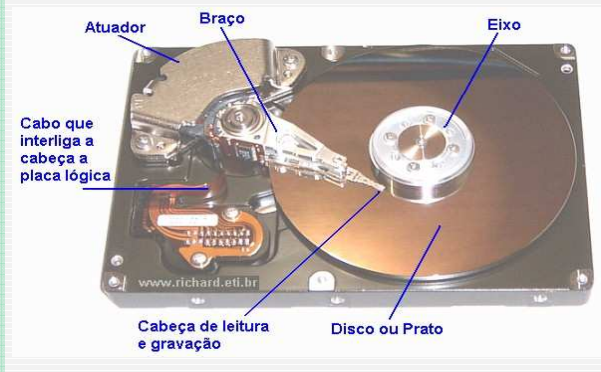
A arquitetura interna de um SGBD (figura adaptada de Korth & Silberchatz 1994)



Níveis de armazenamento de dados



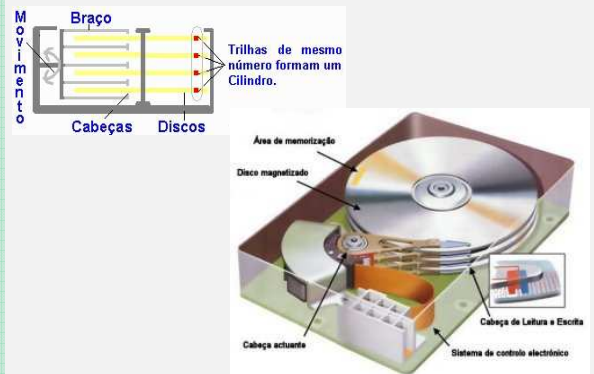
Um disco simples



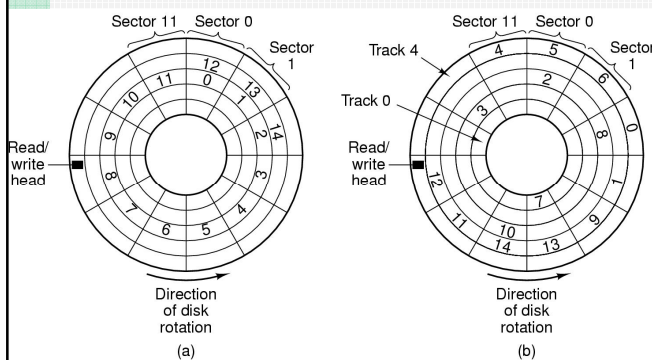
Trilhas e setores



Pilha de discos



Alocação de disco



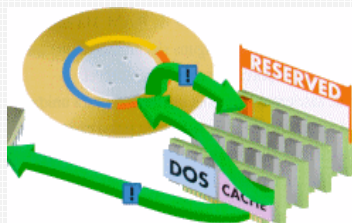
Regs. de tabela em páginas de disco

record_id	campno	maketxt	...	cor_action
42009	02E009000	NXT	...	Nxt will remove the...
13621	02E010000	CATERPILLAR	...	The Dealer will inspect...
42010	02E010000	NEXL	...	Nxt will notify its custom...
35966	99E039000	APC	...	APS will replace these...
42011	02E010000	NEXL	...	Nxt will notify its custom...
12927	01T008000	HERCULES	...	Tires will be replaced...
42012	02E010000	NEXL	...	Nxt will notify its custom...
35974	99E039000	APC	...	APS will replace these...

record_id	campno	maketxt	...	cor_action
42013	02E010000	NEXT	...	Nxt will notify its custom...
13624	02T014000	GOODYEAR	...	The Dealer will replace all...
42014	02E010000	NXT	...	Nxt will notify its custom...
35153	99E019000	D	...	Dealers will inspect their...
42002	02E009000	NEXL	...	Nxt will remove the...
12924	01T008000	NANKANG	...	Defective tires will be...
41467	01X003000	BRITAX	...	Customers will receive a...
35131	99E012000	MERITOR	...	Monitor will inspect and...

record_id	campno	maketxt	...	cor_action
42054	02V074000	PONTIAC	...	Dealers will properly tight...
41863	02V044000	HYUNDAI	...	Dealers will inspect the...
42139	02V095000	INTERNATIONAL	...	Dealers will inspect the...
41926	02V065000	COUNTRY COACH	...	Dealers will replace the...
42138	02V095000	INTERNATIONAL	...	Dealers will inspect the...
41827	02V095000	COUNTRY COACH	...	Dealers will replace the...
42140	02V095000	INTERNATIONAL	...	Dealers will inspect the...
41630	02V065000	COUNTRY COACH	...	Dealers will replace the...

Cache de disco



- Páginas disponíveis em cache não precisam ser acessadas em disco
- Mas seus valores são perdidos se não gravados em disco antes de ocorrer uma falha

Análises de tempo de acesso

- Consideram somente o número de páginas recuperadas para atender uma requisição
- Ignoram ganho no acesso a blocos previamente capturados e mantidos na cache de disco
- Fatores relevantes:
 - Número e tamanho dos registros de dados
 - Número de registros armazenados por página de disco
 - Número de páginas de disco ocupadas pelos dados e estruturas de acesso aos mesmos
 - Tempo médio para ler ou escrever uma página

Exemplo de análise

Sejam:

- NroRegs = 30.000
- TamReg = 100 bytes
- TamPag = 1 Kbyte = 1024 bytes

Pode-se calcular:

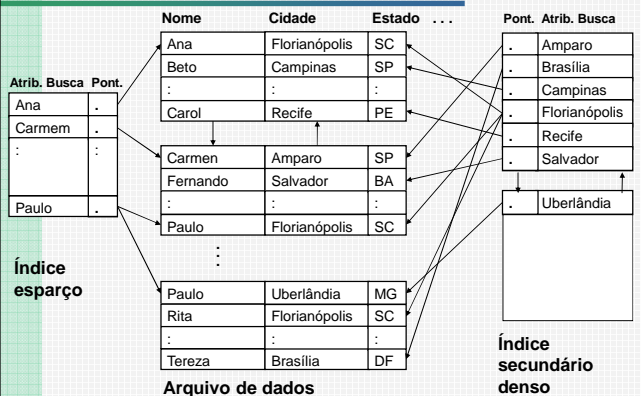
- NroRegsPorPag = $1024 / 100 = 10$ registros
(24 bytes inutilizados por página)
- NroPagsOcupadas = $30.000 / 10 = 3.000$ páginas

O tempo de acesso depende do tipo de requisição e da estrutura utilizada para a organização e o acesso às páginas de dados

Alternativas para organização dos dados em memória secundária

- **Arquivos Heap:** Melhor opção quando o acesso típico é um busca sequencial, sem ordem, recuperando todos registros.
 - Exemplo: um simples arquivo de registros de dados.
- **Arquivos Ordenados:** Melhor opção se os registros devem ser recuperados em alguma ordem, ou quando somente uma porção contígua dos registros ordenados é necessária.
 - Ex.: Arquivo de dados, possivelmente ordenado, além de vários índices.
- **Arquivos Hashed:** Melhor para seleções de igualdade.
 - O arquivo de dados é uma coleção de *buckets*.
 - *Bucket* = uma página primária e possíveis páginas de *overflow*.
 - Função de *Hashing*: $h(K)$ = endereço do *bucket* contendo o(s) registro(s) com o valor de busca *K*.
 - Não é adequado para acesso sequencial ordenado.

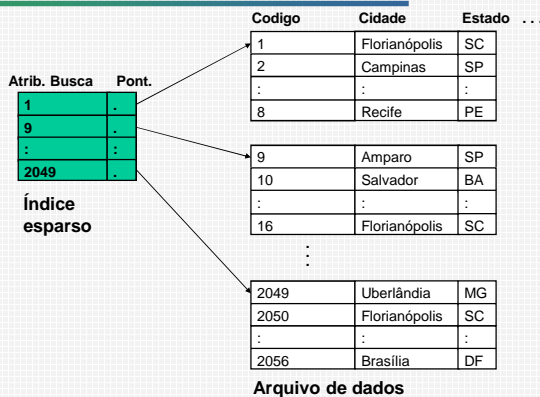
Indexação



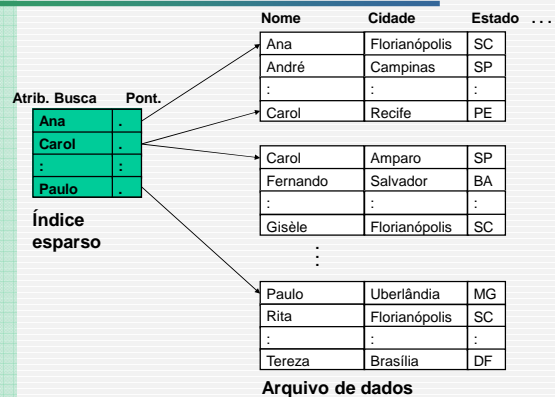
Tipos de índices

- **Índice primário:** ordenação, inclusive dos dados, pela chave primária
 - Esparso
 - Máxima eficiência
- **Índice de agrupamento (*clustering index*):** ordenação, inclusive dos dados, por atributo(s) não chave (que pode(m) ter valores repetidos)
 - Esparso
 - Eficiência próxima à do índice primário
- **Índices secundários:** ordenações distintas (auxiliares)
 - Densos
 - Menos eficientes que os anteriores, porém muito mais eficientes que busca sequencial

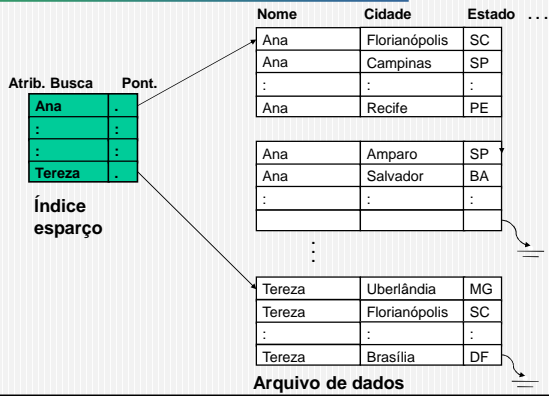
Índice primário



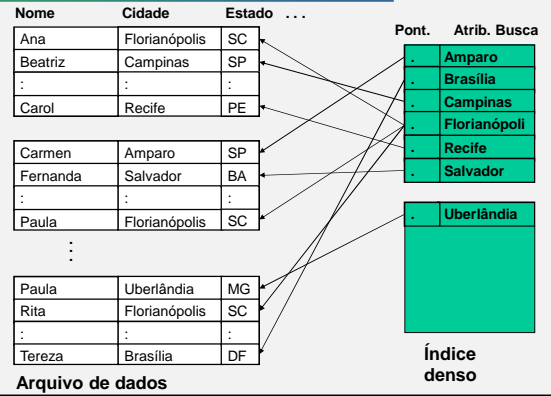
Índice de agrupamento (*clustering index*)



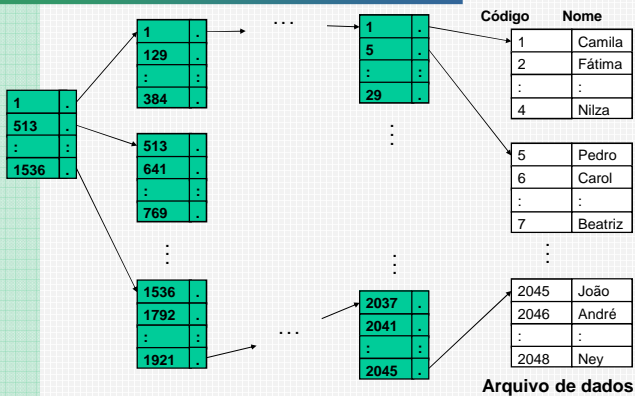
Índice de agrupamento com blocos para cada valor de busca



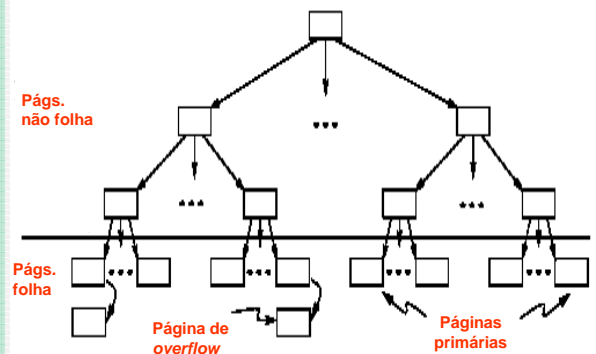
Índice secundário



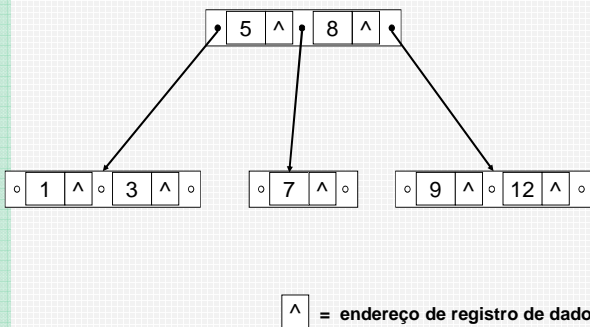
Índices com múltiplos níveis



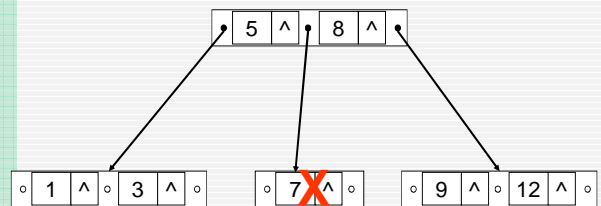
Estrutura de índice em árvore (figura extraída de Ramakrishnan 1998)



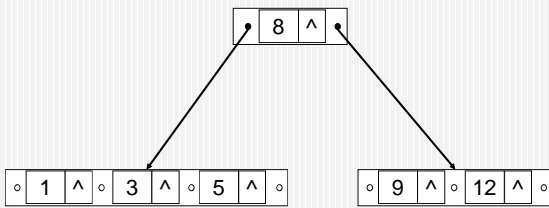
B-Tree de ordem 3



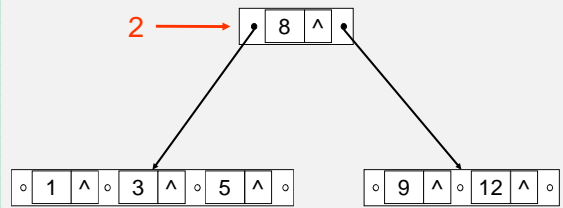
Remoção na B-Tree



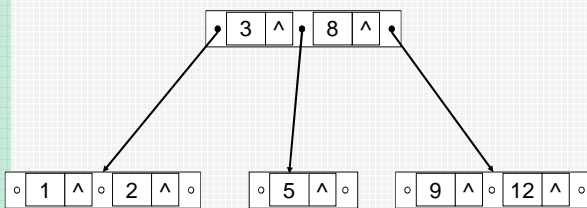
Possível resultado da remoção



Inserção na B-Tree

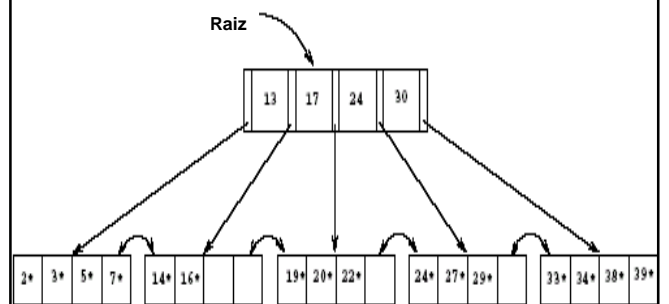


Possível resultado da inserção



B+_Tree

(figura extraída de Ramakrishnan 1998)



Especificação de índices em SQL

```
CREATE INDEX IndicePorNome ON Pessoa
WITH STRUCTURE = BTREE,
KEY = (nome)
```

```
CREATE INDEX IndicePorCidadeEstado ON Pessoa
WITH STRUCTURE = BTREE,
KEY = (cidade, estado)
```

```
CREATE INDEX IndicePorEstadoCidade ON Pessoa
WITH STRUCTURE = BTREE,
KEY = (estado, cidade)
```

Leituras recomendadas

- Elmasri, R.; Navathe, S.B. *Fundamentals of database Systems. 4 ed.*, Pearson, 2003.
 - Chapter 13 – Disk Storage, Basic File Structures, and Hashing
 - Chapter 14 – Indexing Structures for Files
- Ramakrishnan, R. *Database management systems*. McGraw-Hill, 1998.
 - Part 3 – Data Storage and Indexing
 - Chapter 7 – Data Storage: Disks and Files
 - Chapter 8 – Files and indexes
 - Chapter 9 – Tree-based Indexing
 - Chapter 10 – Hashing
- Abiteboul, S.; Buneman, P.; Suciu, D. *Data on the Web – from Relations to Semistructured Data and XML*. Morgan Kaufmann, 2000.
 - Section 8.2.2 – Indexing