Estrutura de

Armazenamento e de

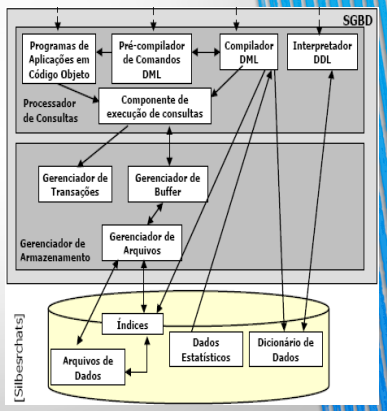
Arquivos

Objetivos

- Que tipos de memória existem num computador?
- Quais são as características físicas dos discos rígidos e das fitas e como que afetam o design de sistemas de bancos de dados?
- O que são os sistemas RAID de memória e quais são as suas vantagens?
- Como é que um SGBD registra o espaço em disco?
- Como é que um SGBD lê e modifica os dados em disco? Qual é o significado de um bloco enquanto unidade de armazenamento e transferência de dados?
- Como é que um SGBD cria e mantém arquivos de registros? Como é que os registros estão organizados em blocos, e como estão os blocos organizados dentro de um arquivo?

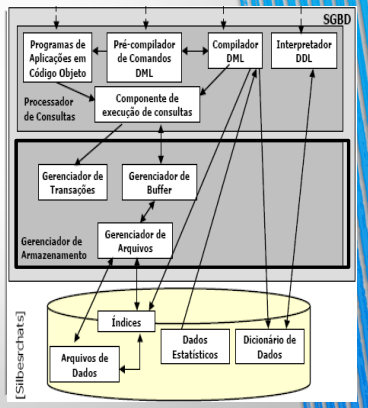
Estrutura Simplificada de um SGBD

- Bancos de Dados armazenam grandes quantidades de dados por períodos longos de tempo em meios de armazenamento secundário
- Um SGBD provê geralmente várias opções para organização física dos dados.
- Projeto Físico de Banco de Dados:
- Busca determinar o melhor tipo de organização dos dados, dentre todas as possíveis, para uma determinada aplicação;
- Especificar o Modelo Físico de Banco de Dados, levando em consideração o Modelo de Dados lógico e informações sobre volumes, acessos e necessidade de disponibilidade
- Visando garantir uma implementação com ótima performance
- Assegurando aspectos como padronização, portabilidade, disponibilidade e capacidade de recuperação tempestiva dos dados.
- Cada sistema tem as suas próprias particularidades.



Discos e Arquivos

- Um SGBD guarda informação em discos.
- Este fato tem grandes implicações no projeto de um SGBD
- **READ**: transferência de dados do disco para a memória principal (RAM).
- **WRITE**: transferência de dados da RAM para o disco.
- Ambas são operações de custo elevado em termos de tempo e espaço em memória, de modo que devem ser planejadas cuidadosamente!



Meios Físicos de Armazenamento

Aspectos a Serem Considerados

- Velocidade com a qual um dado pode ser acessado
- Custo para ler e armazenar cada unidade de dado
- Disponibilidade
- Perda de dados em caso de falha no sistema
- Falhas físicas nos dispositivos de armazenamento

Visão Geral dos Meios Físicos de Armazenamento

1. **Voláteis**
   - Memória RAM (Principal)
   - Memória Cache
2. **Não-voláteis**
   1. Memória secundária
      - Disco magnético
      - Disco ótico
      - Memória flash
   2. Memória terceária
      - Fita

Por que não armazenar tudo em memória principal?

- Custo muito alto
- Memória volátil

Hierarquia de Armazenamento

1. Armazenamento **Primário**:

- Memória Principal + Caches de Memória:
- acessado diretamente pela CPU, acesso rápido, custo alto

2. Armazenamento **Secundário** ou On-Line:

- Discos magnéticos, óticos e memória flash
- armazena a base de dados em si, acesso lento
- dados são copiados nos meios de armazenamento primário para serem processados e depois reescritos novamente

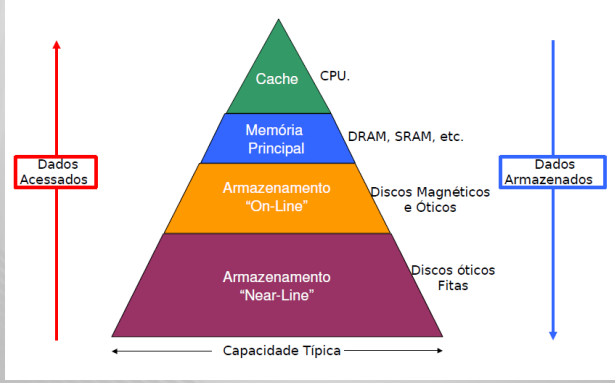
3. Armazenamento **Terceário** ou Off-Line:

- Fitas = para versões antigas da base de dados (ou backups)

Questão

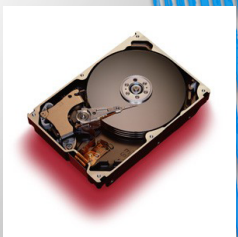
Qual a diferença entre armazenamento primário e secundário?

Hierarquia de Armazenamento



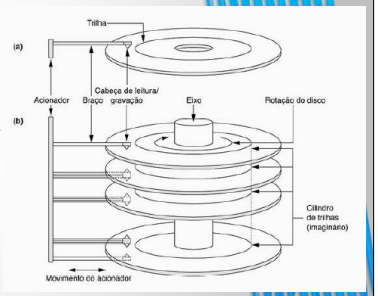
Discos Magnéticos

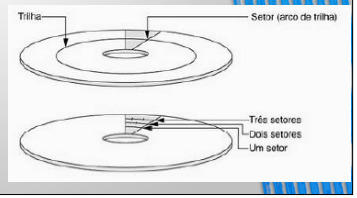
- Principal dispositivo de memória secundária utilizado.
- Principal vantagem com relação às fitas:
- **acesso aleatório (randômico) vs. sequencial.**
- Dados são armazenados e devolvidos em unidades chamadas blocos ou páginas.
- Ao contrário da RAM, o tempo para devolver um bloco de um disco varia com a sua localização em disco, o que tem grande impacto no desempenho de um SGBD.



Componentes de um disco

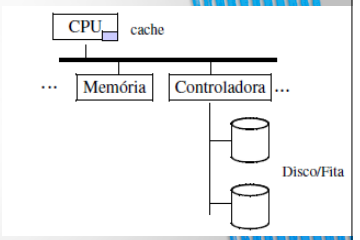
- Os blocos são armazenados em trilhas **trilhas** que formam cada prato (disco)
- 500 a 200 trilhas por superficie
- Cada trilha é dividida em vários **setores** (menor unidade de informação lida/escrita)
- 32 a 128 setores por trilha
- 1 setor = 32 até 4096 bytes
- Os **pratos** giram. Por exemplo,5400 rpm.
- A agulha move-se para dentro ou para fora de modo a posicionar a cabeça sobre a trilha desejada.
- As trilhas de mesmo diâmetro formam **<u>cilindro</u>** (imaginário!).
- *<u>Só uma cabeça lê/escreve em cada instante.</u>*





Subsistema de discos

- Um controlador de discos, comumente embutido na unidade de disco, controla o disco e o interliga aos sistema de computação, fazendo interface entre o disco e a memória RAM
- Recebe comandos de leitura/escrita de dados
- Remapeamento de setores ruins
- ATA, SATA, SCSI (mais usada para conectar discos a computadores pessoais e estações de trabalho)



Medidas de Desempenho de Disco

1. **Capacidade**
2. **Tempo de Acesso** – acesso de leitura ou escrita requer três passos:

- Tempo de procura (seek): posicionamento do braço na trilha correta. De 4 a 10 ms.
- Tempo de atraso (latência rotacional): espera até o setor desejado seja rotacionado até a cabeça de leitura/escrita. 5400 to 15000 r.p.m.
- Tempo de transferência: transferência dos bits armazenados no setor que está ao alcance da cabeça. 25 a 100 Mb por segundo
- Bloco: unidade de transferência

3. **Confiabilidade**

- Tempo médio para a ocorrência de falhas: média de tempo que se pode esperar que o disco trabalhe sem que ocorra falhas. De 3 a 5 anos.

Exemplo - Seagate

Especificações	4 TB¹	3 TB¹	2 TB¹	1 TB¹
Número do modelo	ST4000DM000	ST3000DM001	ST2000DM001	ST1000DM003
Nome do modelo	Desktop HDD	anteriormente Barracuda®	anteriormente Barracuda	anteriormente Barracuda
Opções de interface	SATA de 6 Gb/s com NCQ	SATA de 6 Gb/s com NCQ	SATA de 6 Gb/s com NCQ	SATA de 6 Gb/s com NCQ
Desempenho				
Cache, multisegmentado (MB)	64	64	64	64
Taxas de transferência aceitas por SATA (Gb/s)	6,0/3,0/1,5	6,0/3,0/1,5	6,0/3,0/1,5	6,0/3,0/1,5
Média de busca, leitura (ms)	<8,5	<8,5	<8,5	<8,5
Média de busca, gravação (ms)	<9,5	<9,5	<9,5	<9,5
Taxa média de dados, leitura/gravação (MB/s)	160	156	156	156
Taxa de dados sustentada máx., leitura DE (MB/s)	180	210	210	210
Configuração/organização				
Cabeças/discos	8/4	6/3	6/3 4/2	2/1
Bytes por setor	4.096	4.096	4.096	4.096

Exercício

1. Considere um disco com tamanho de setor igual a 512 bytes, 2000 trilhas por superfície, 50 setores por trilha, cinco pratos e tempo de busca médio de 10 ms.

a. Qual a capacidade de uma trilha em bytes? Qual a capacidade de cada superfície? Qual a capacidade do disco?

b. Dê exemplos de tamanhos válidos de blocos. 256 bytes? 2048 bytes? 51200 bytes?

c. Se os pratos do disco girarem a 5.400 rpm, qual a latência rotacional máxima?

d. Se uma trilha de dados puder ser transferida por rotação, qual a taxa de transferência?

e. Quantos cilindros o disco tem?

Otimização de Acesso de Blocos de Disco

- **Bloco** = é uma sequência contígua de bytes de uma única trilha de um prato
- Dados são transferidos do disco para a memória principal em blocos
- Os tamanhos dos blocos variam de 512 bytes a vários kb
- Blocos menores – mais transferências do disco
- Blocos maiores – mais espaço desperdiçado
- O tamanho mais comum varia de 4 a 16 kbytes

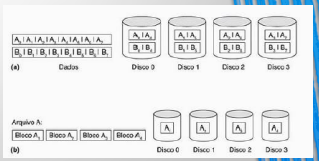
Melhoria na Confiabilidade e Desempenho

RAID (Arrays Redundantes de Discos Independentes)

- Array de pequenos discos independentes que atuam como único disco lógico de maior desempenho
- Conjunto de discos de dados + um conjunto de discos de verificação
- Duas técnicas principais:
  1. **Data striping**: particionamento de dados
  2. **Redundância**: informação redundante permite reconstrução de dados caso disco falhe

Striping de dados

- Emprega o paralelismo para melhorar o desempenho do disco
- O tamanho da partição é chamado unidade striping
- Partições de mesmo tamanho são distribuídas em vários discos.
- Para D discos a partição i é escrita no disco (i mod D)
- Permite leitura em paralelo
- Partição pode ser por **bit** ou **bloco**



Níveis de RAID - 0

- **Sem redundância**
- Este nível também é conhecido como "Striping" ou "Fracionamento".
- Os dados são divididos em pequenos segmentos e distribuídos entre os discos.
- Não oferece tolerância a falhas, pois não existe redundância.
- Isso significa que uma falha em qualquer um dos HDs pode ocasionar perda de informações.

Níveis de RAID - 1

- **Espelhamento**
- Consiste em espelhar os discos.
- A informação gravada num disco será gravada em dois discos componentes do array.
- Caso um deles falhe, o array continua funcionando.

Níveis de RAID - 5

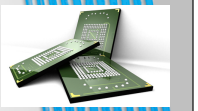
- **Paridade Distribuída de Blocos Interlaçados**
- Para cada bloco, um dos discos armazena a paridade e os outros armazenam os dados.
- O nível 5 aumenta a velocidade em gravações pequenas, uma vez que não há um disco separado de paridade como gargalo.
- Porém como o dado de paridade tem que ser distribuído entre todos os discos disponíveis, durante a leitura, a performance possui tendência de ser um pouco mais lenta que a do nível 4.
- É o tipo mais comum de RAID

Para Pensar

Considere o seguinte arranjo de quatro discos de blocos de dados e de paridade em que Bis representam blocos de dados e Pis representam os blocos de paridade. O bloco de paridade Pi é o bloco de paridade para os blocos de dados de B 4i-3 até B4i. Qual o problema (se houver) que esse arranjo pode representar?

Disco 1	Disco 2	Disco 3	Disco 4
B_1	B_2	B_3	B_4
P_1	B_5	B_6	B_7
B_8	P_2	B_9	B_{10}
.	.	.	.
.	.	.	.
.	.	.	.

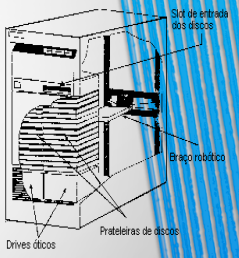
Memória Flash



- Opção de armazenamento secundário
  1. não volátil
  2. acesso rápido à memória RAM
  3. usada em dispositivos USB, câmeras, celulares, laptops
- Está "substituindo" os discos HD nos armazenamentos de dados, porém o custo ainda é bem maior (3 a 4 vezes)

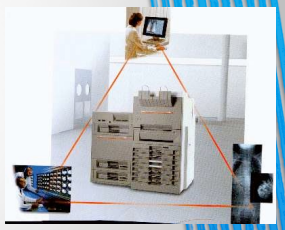
Discos Óticos

- Os dados são armazenados oticamente nos discos e lidos a laser (CDs e DVDs)
- Os discos podem ser carregados e removidos facilmente do acionador
- Grande capacidade de armazenamento, custo baixo
- Sistemas junkebox = contém várias unidades de discos que podem ser trocadas automaticamente por meios de braços mecânicos.



Fitas Magnéticas

- Usadas primordialmente para para backups
- Acesso muito lento (sequencial)
- Barato e de fácil armazenamento
- Junkeboxes de fitas = mantém um grande números de fitas (na casa das centenas) com troca automática entre elas.



Questão

Por que os discos e não as fitas são usados para armazenar arquivos de banco de dados on-line?

Novos Sistemas de Armazenamento

1. SAN - Storage Area Network (Área de Armazenamento em Rede)
   - Periféricos de armazenamento online são configurados como nós em uma rede de alta velocidade e podem ser conectados/desconectados dos servidores com flexibilidade
2. NAS - Network-Attached Storage (Armazenamento Conectado à rede)
   - São servidores que permitem o acréscimo de armazenamento para o compartilhamento de arquivos
3. iSCSI - Internet SCSI
   - Novo protocolo de rede que permite que os clientes enviem comandos para dispositivos de armazenamento SCSI em canais remotos

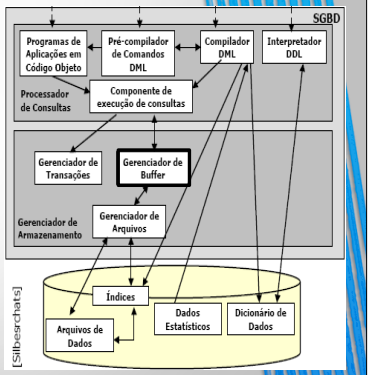
Acesso e controle do armazenamento

Gerenciamento do Espaço de Armazenamento em Disco

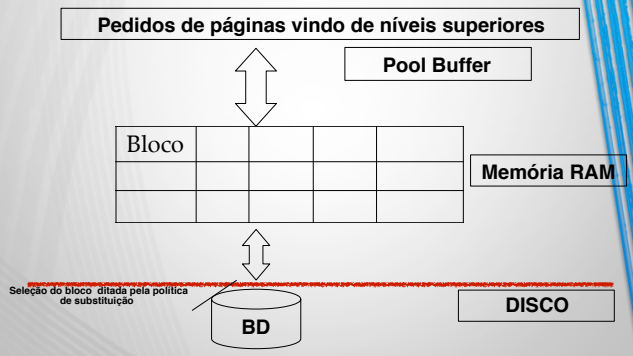
- A camada mais baixa de um SGBD faz o gerenciamento de espaço em disco junto com o SO
- As camadas ou níveis mais elevados acessam esta camada para:
- alocar/desalocar um bloco e ler/escrever um bloco
- Melhor seria se os pedidos por uma sequência de blocos fossem satisfeitos pelos blocos armazenados sequencialmente no disco!
- Níveis superiores não sabem como isto é feito, ou como o espaço livre é gerido.
- Embora eles possam assumir acesso sequencial a arquivos!
- Daí que o gestor de espaço em disco deve fazer um trabalho bem feito.

Gerenciamento de Buffer

- **Buffer** – parte da memória principal disponível para armazenamento de cópias dos blocos de discos
- O subsistema responsável pela alocação do espaço disponível em buffer é chamado Gerenciador de Buffer



Gerenciamento de Buffer



Quando um bloco de dados é requisitado

- Se o bloco requisitado não está no pool
- Escolhe-se um bloco no buffer para substituição
- Se este bloco estiver ocupado, escreve seus dados no disco
- Lê o bloco de dados requisitado e coloca-o dentre deste bloco que acabou de ser desocupado.

Políticas de Substituição em Buffer

- Um bloco é selecionado para substituição com base numa política de substituição:
- Least recently used (LRU)
- Most recently used (MRU)

Política de Substituição LRU

- Least Recently Used (Menos Recentemente Utilizado)
  - para cada bloco no pool do buffer, registrar o tempo da última substituição
  - substituir o bloco com o tempo mais antigo
  - política muito comum: intuitiva e simples
  - funciona bem para acessos repetidos a páginas populares

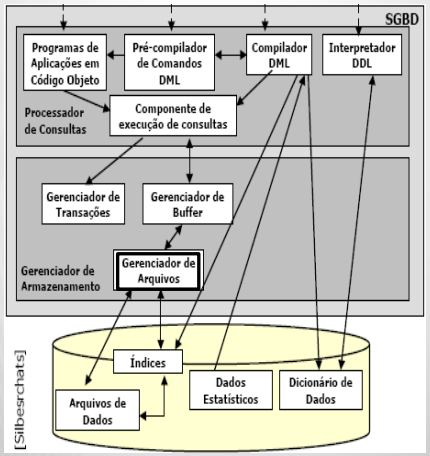
Política de Substituição MRU

- Most Recently Used (Mais Recentemente Utilizado)
  - oposto da estratégia LRU
    - para cada bloco no buffer, registrar o tempo da última substituição
  - substituir o bloco com o tempo mais recente

Estratégia Ideal

- Requer conhecimento das operações de banco de dados em cada aplicação específica
- Não uma estratégia que seja boa para todos os cenários…
- Outros fatores que influenciam
- Acesso concorrente ao dado
- Recuperação de falhas, etc
- A política pode ter um enorme impacto na quantidade de operações de E/S

Organização de Arquivos



Arquivos

- Blocos constituem a interface para E/S, mas…
- As camadas superiores do SGBD operam sobre registros e arquivos de registros.
- **ARQUIVO** = uma coleção de blocos, cada um contendo uma coleção de registros. Deve suportar operações de:
- inserir/apagar/modificar registros
- pesquisar um registro particular
- ler todos os registros (possivelmente com algumas condições sobre os registros a ser devolvidos)

Registros

- Os dados são armazenados na forma de **registros**
- Cada registro possui um conjunto de valores de dados onde cada valor é formado por um ou mais bytes e corresponde a um **campo** do registro

```
struct funcionario{
char nome[30];
char cpf[9];
int salario;
int cod_cargo;
char departamento[20];
}
```

Registros

1. **Tamanho fixo:**

- Todos os registros possuem o mesmo tamanho exatamente; mesma quantidade de bytes

2. **Tamanho variável** (formato ou tamanho):

- Um ou mais campos tem tamanho variável
- Campos com múltiplos valores (campos repetidos)
- Campos opcionais

Registros de tamanho fixo

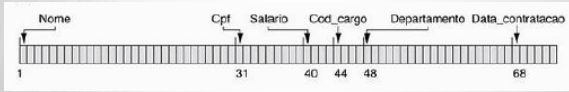
```
type deposito = record
  nome_agencia : char(22);
  numero_conta : char(10);
  saldo : real;
End
```

C1 **C2** **C3**

— T1 —	— T2 —	— T3 —

Ci = campo i
Ti = tamanho do campo i

Registro de tamanho fixo



Registro de formato variável

- Campos <u>opcionais</u>

- Campos <u>tamanho variável</u>, onde não se sabe ao certo o tamanho do campo, usa-se separadores especiais

Exercício

- Quais os motivos para a existência de registros de tamanho variáveis?

Alocação de registros

- **Não espalhada** - quando o registro cabe num bloco
- **Espalhada** - quando o registro NÃO cabe num bloco

Alocação de blocos no disco

- **Contígua** - blocos de arquivos são alojados em blocos consecutivos do disco
- **Ligada** - cada bloco de arquivo contém um ponteiro para o próximo bloco de arquivo

Cabeçalhos de Arquivo

- Contém informações sobre um arquivo para determinar os endereços de disco dos blocos, registra descrição de formato de registro, como tamanho e a ordem dos campos nos registros, entre outras.

Questão

- Quais são e como funcionam as técnicas para alocar blocos de arquivo nos discos?

Organização de registros em ***arquivos***

O objetivo de uma boa organização de arquivos é localizar o bloco desejado com um número mínimo de transferências de bloco entre o *disco* e a *memória principal*

Organização de Arquivos

- Arquivo **Heap**
- Arquivo **Sequencial**
- Arquivo **Hashing**
- Arquivo **Clustering**

Arquivos Heap

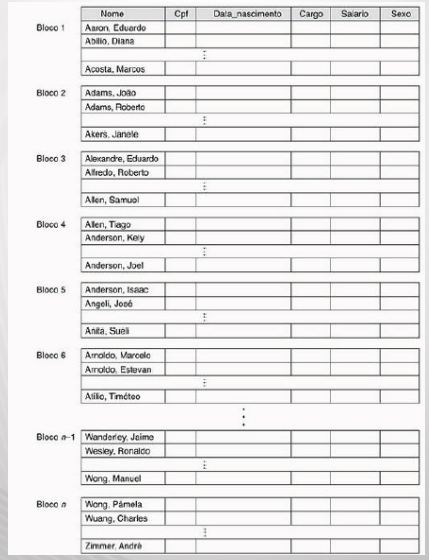
- A estrutura mais simples de um arquivo é aquela que contém os registros sem qualquer ordem em particular. Estes arquivos são conhecidos por heap files.
- Quando o arquivo aumenta ou diminui de tamanho, blocos em disco são alocados e desalocados.
- Normalmente, há um único arquivo para cada relação.

Arquivos Sequenciais

- Registros fisicamente ordenados por uma chave primária ou chave de ordenação
- Indicação de uso
  - Memória de acesso sequencial
  - Indicado para arquivos que sofrem recuperações/atualizações por lotes (em batch)
- Contra-indicação
  - Quando há mais do que uma chave
  - Quando exige-se respostas em tempo real
  - Aplicações com inserções/exclusões arbitrárias

Operações Sequenciais

- **Acesso**
  - Registros fisicamente armazenados de acordo com a sequência na qual são solicitados
  - Na maioria dos acessos o registro solicitado estará em memória por pertencer ao mesmo bloco do seu antecessor
- **Inserção**
  - Localizar registro anterior ao que será incluído pela ordem da chave primária
  - Se há espaço dentro do mesmo bloco desse registro, insere o novo registro. Senão, inserir o novo registro em um bloco de overflow.
- **Deleção**
  - Cadeias de ponteiros (marcação para remoção física)



Exercício

- Em uma organização de arquivo sequencial, por que um bloco de *overflow* é utilizado mesmo se houver apenas um registro de *overflow*?

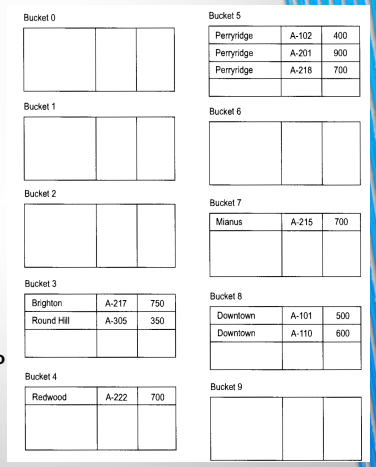
Arquivo Hashing

- Uma função hash é calculada sobre algum atributo de cada registro
  - Função hash h(k) = é uma função que transforma uma chave k num endereço. Este endereço é usado como a base para o armazenamento e recuperação de registros
- O resultado da função especifica em qual bloco do arquivo o registro deve ser colocado.

Exemplo

h(nome_agencia) = soma das representações binárias dos caracteres de uma chave e então retorna o módulo (MOD) da soma pelo número de *blocos*

Organização de *hash* do arquivo conta



Arquivo Clustering/Multitabela

- Registros de diferentes relações podem estar armazenados em um mesmo arquivo.
- Registros relacionados de diferentes relações são armazenados no mesmo bloco para que operações de E/S busquem registros relacionados de todas as relações.

nome_cliente	número_conta
Hayes	A-102
Hayes	A-220
Hayes	A-503
Turner	A-305

Relação depositante

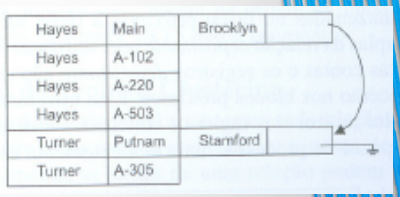
nome_cliente	rua_cliente	cidade_cliente
Hayes	Main	Brooklyn
Turner	Putnam	Stamford

Relação cliente

Arquivo Clustering Multitabela

Hayes	Main	Brooklyn
Hayes	A-102	
Hayes	A-220	
Hayes	A-503	
Turner	Putnam	Stamford
Turner	A-305	

Clustering de arquivo



Clustering de arquivo com cadeias de ponteiros

CATALOGO DO SISTEMA ou dicionário de dados

- Para cada relação:
  - nome, localização do arquivo, estrutura do arquivo(p.ex. heap file)
  - nome e tipo de cada atributo
  - nome de cada índice
  - restrições de integridade
- Para cada índice:
  - estrutura (p.ex. B+ tree) e campos-chave de pesquisa
- Para cada visão:
  - nome e definição
- + estatística, autorização, tamanho da buffer pool, etc.
- **Catálogos são eles próprios armazenados como relações!**

CATALOGO DO SISTEMA ou dicionário de dados

- Catálogos são eles próprios armazenados como relações!
  - Esquema_catalogo_sistema = (nome_relação, nome_atributos)
  - Esquema_atributo = (nome_atributo, nome_relacao, tipo_dominio, posição, tamanho)
  - Esquema_usuario = (nome_usuario, senha, grupo)
  - Esquema_indice = (nome_indice,nome_relacao, tipo_indice,atributos_indice)

Exercício

- Considere um banco de dados relacional com duas relações:

1. Curso (nome_curso,sala, instrutor)
2. Matrícula (nome_curso, nome_estudante, período)

Defina instâncias para essas relações para três cursos, cada qual com dois estudantes matriculados. Dê uma estrutura de arquivos para essas relações utilizando:
a) Arquivo Sequencial
b) Arquivo Clustering

Dúvidas???

- Capítulo 17 do livro do Navathe