

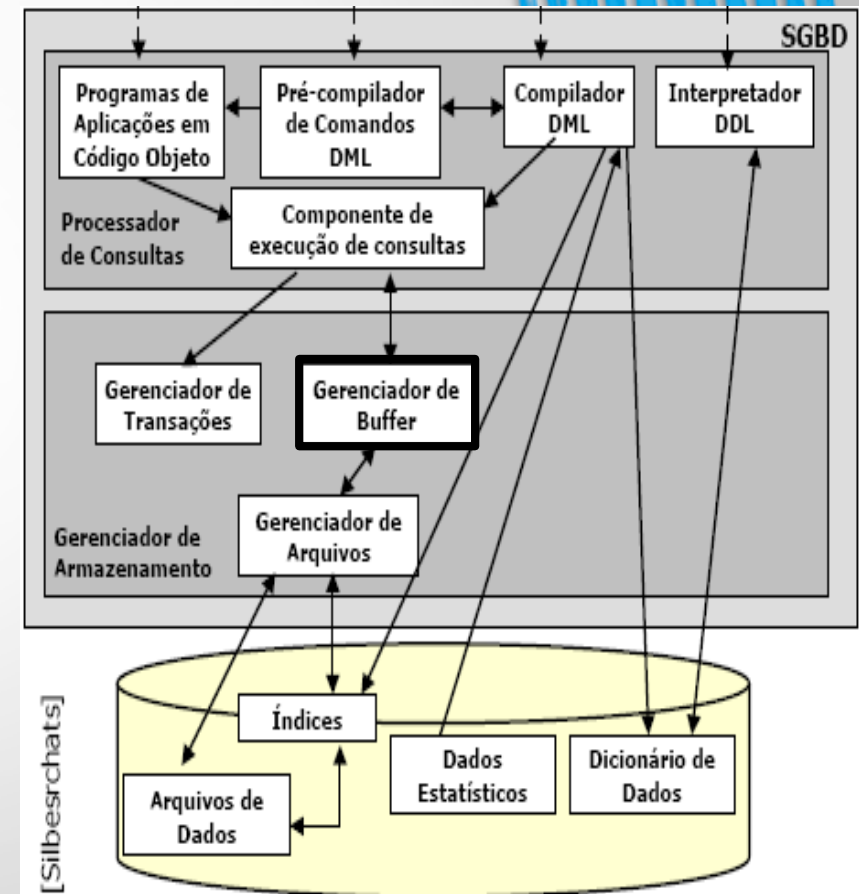
# Acesso e controle do armazenamento

# Gerenciamento do Espaço de Armazenamento em Disco

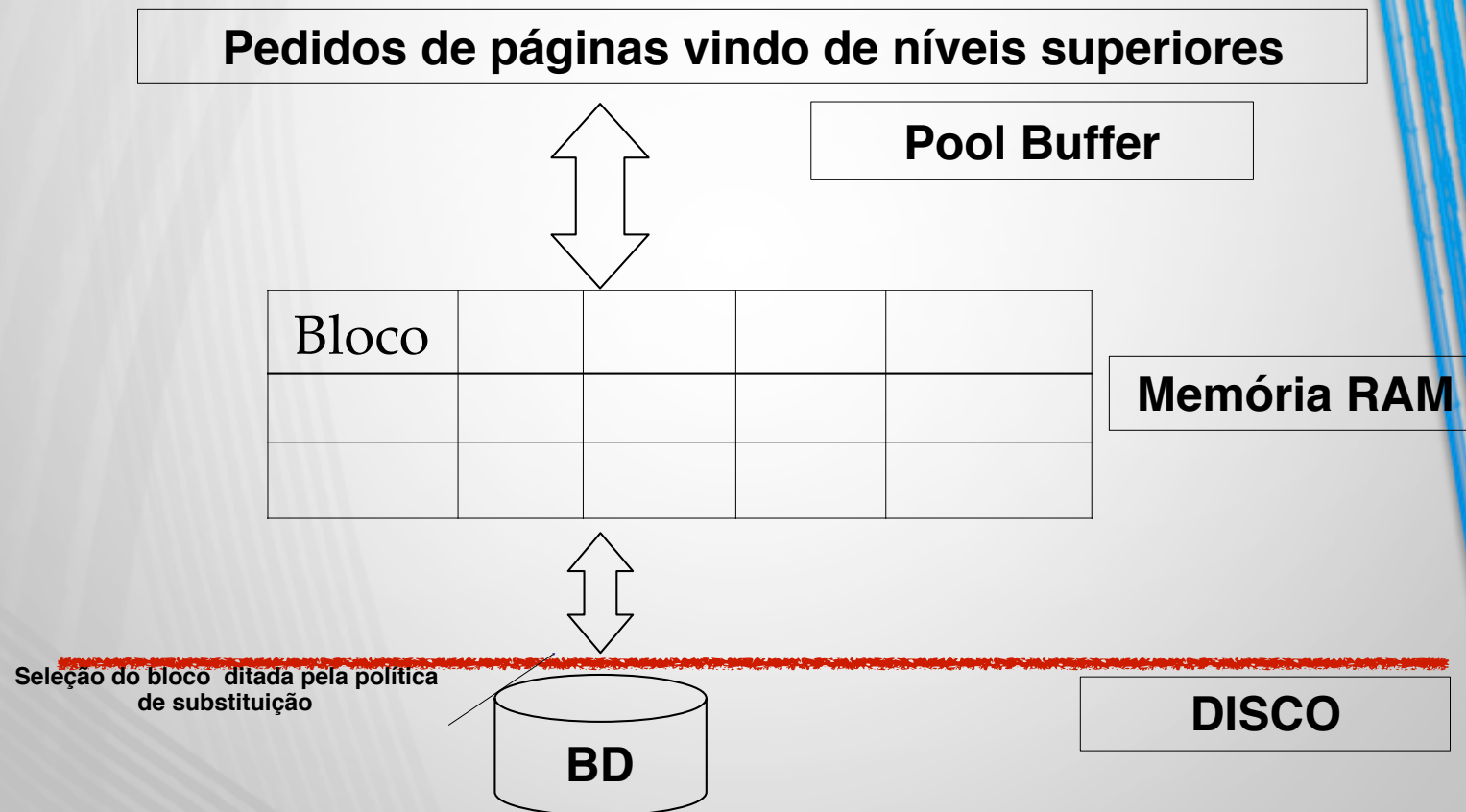
- A camada mais baixa de um SGBD faz o gerenciamento de espaço em disco junto com o SO
- As camadas ou níveis mais elevados acessam esta camada para:
- alocar/desalocar um bloco e ler/escrever um bloco
- Melhor seria se os pedidos por uma sequência de blocos fossem satisfeitos pelos blocos armazenados 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 dentro 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

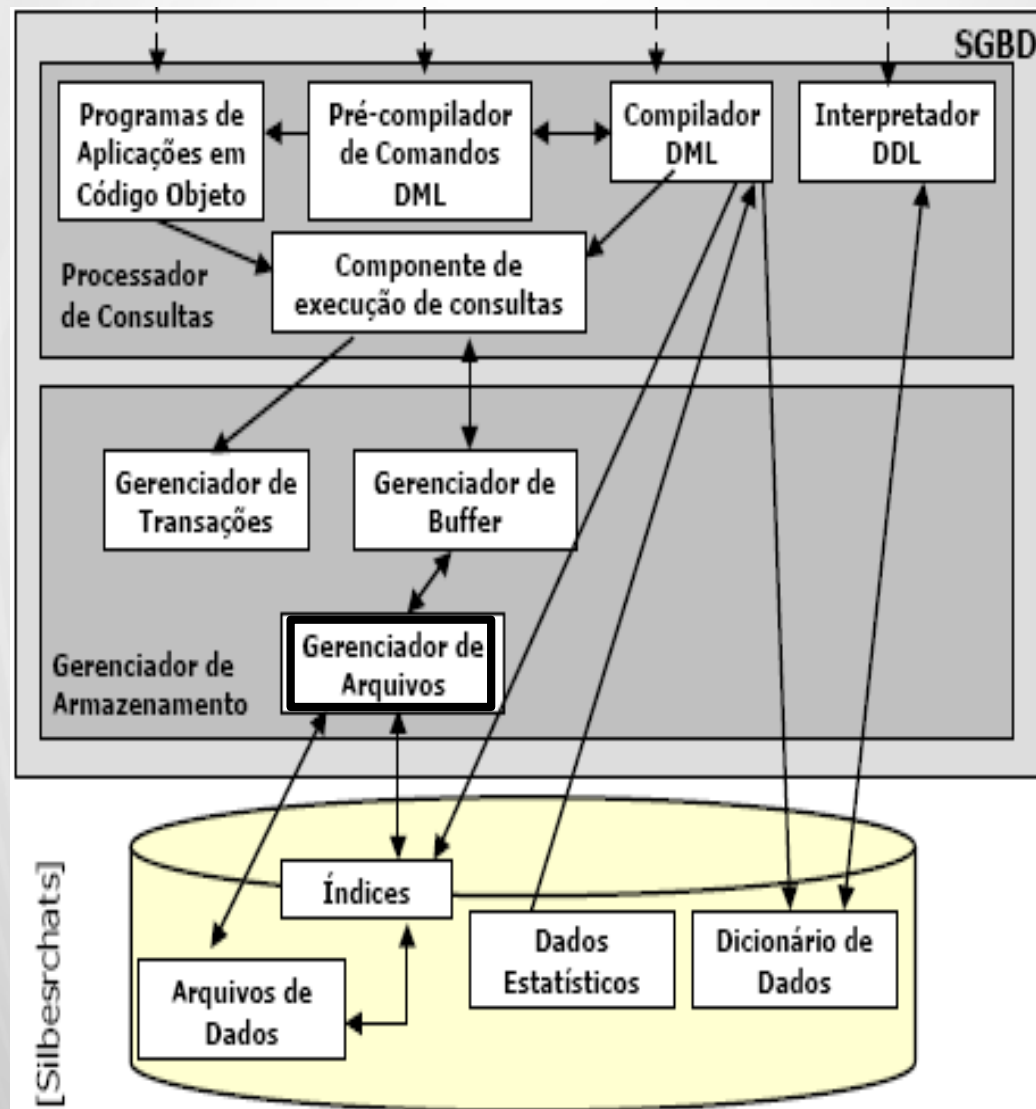
- Most Recently Used (Mais Recentemente Utilizado)
  - oposto da estratégia LRU
  - para cada bloco no buffer, registrar o tempo da última substituição
  - substituir o bloco com o tempo mais recente



# 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



[Silbesrchats]

# 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**

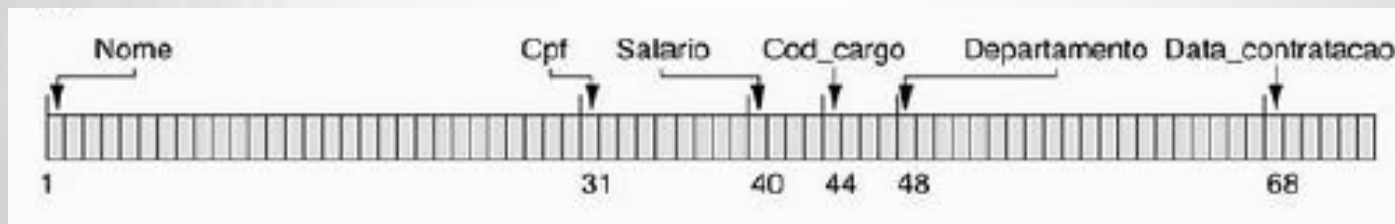
**C3**

— T1 —	— T2 —	— T3 —
--------	--------	--------

**C<sub>i</sub> = campo i**

**T<sub>i</sub> = tamanho do campo i**

# Registro de tamanho fixo





# Registro de formato variável

- Campos opcionais

Nome = Silva, João	Cpf = 12345678966	DEPARTAMENTO = Computação	<input checked="" type="checkbox"/>
--------------------	-------------------	---------------------------	-------------------------------------

Caracteres separadores

- = Separa nome de campo do valor do campo
- █ Separa campos
- ☒ Termina registro

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

Nome	Cpf	Salario	Cod_cargo	Departamento	
Silva, João	12345678966	XXXX	XXXX	Computação	█
1	12	21	25	29	

Caracteres separadores

█

# 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)



	Nome	Cpf	Data_nascimento	Cargo	Salario	Sexo
Bloco 1	Aaron, Eduardo					
	Abilio, Diana					
	⋮					
	Acosta, Marcos					
Bloco 2	Adams, João					
	Adams, Roberto					
	⋮					
	Akers, Janele					
Bloco 3	Alexandre, Eduardo					
	Alfredo, Roberto					
	⋮					
	Allen, Samuel					
Bloco 4	Allen, Tiago					
	Anderson, Kely					
	⋮					
	Anderson, Joel					
Bloco 5	Anderson, Isaac					
	Angeli, José					
	⋮					
	Anita, Suelli					
Bloco 6	Arnoldo, Marcelo					
	Arnoldo, Estevan					
	⋮					
	Atilio, Timóteo					
⋮						
Bloco $n-1$	Wanderley, Jaime					
	Wesley, Ronaldo					
	⋮					
	Wong, Manuel					
Bloco $n$	Wong, Pâmela					
	Wuang, Charles					
	⋮					
	Zimmer, André					



# 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

**$h(\text{nome\_agencia})$  = 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



Bucket 0

--	--	--

Bucket 1

--	--	--

Bucket 2

--	--	--

Bucket 3

Brighton	A-217	750
Round Hill	A-305	350

Bucket 4

Redwood	A-222	700

Bucket 5

Perryridge	A-102	400
Perryridge	A-201	900
Perryridge	A-218	700

Bucket 6

--	--	--

Bucket 7

Mianus	A-215	700

Bucket 8

Downtown	A-101	500
Downtown	A-110	600

Bucket 9

--	--	--

# 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.

<i>nome_cliente</i>	<i>número_conta</i>
Hayes	A-102
Hayes	A-220
Hayes	A-503
Turner	A-305

**Relação  
depositante**

<i>nome_cliente</i>	<i>rua_cliente</i>	<i>cidade_cliente</i>
Hayes	Main	Brooklyn
Turner	Putnam	Stamford

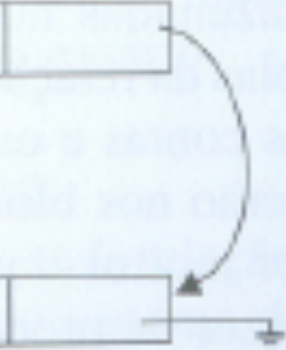
**Relação cliente**



# Arquivo Clustering Multitabela

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

**Clustering de  
arquivo**

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

**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\_relção, 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