

# Fundamentos de Banco de Dados

## Estrutura de um SGBD

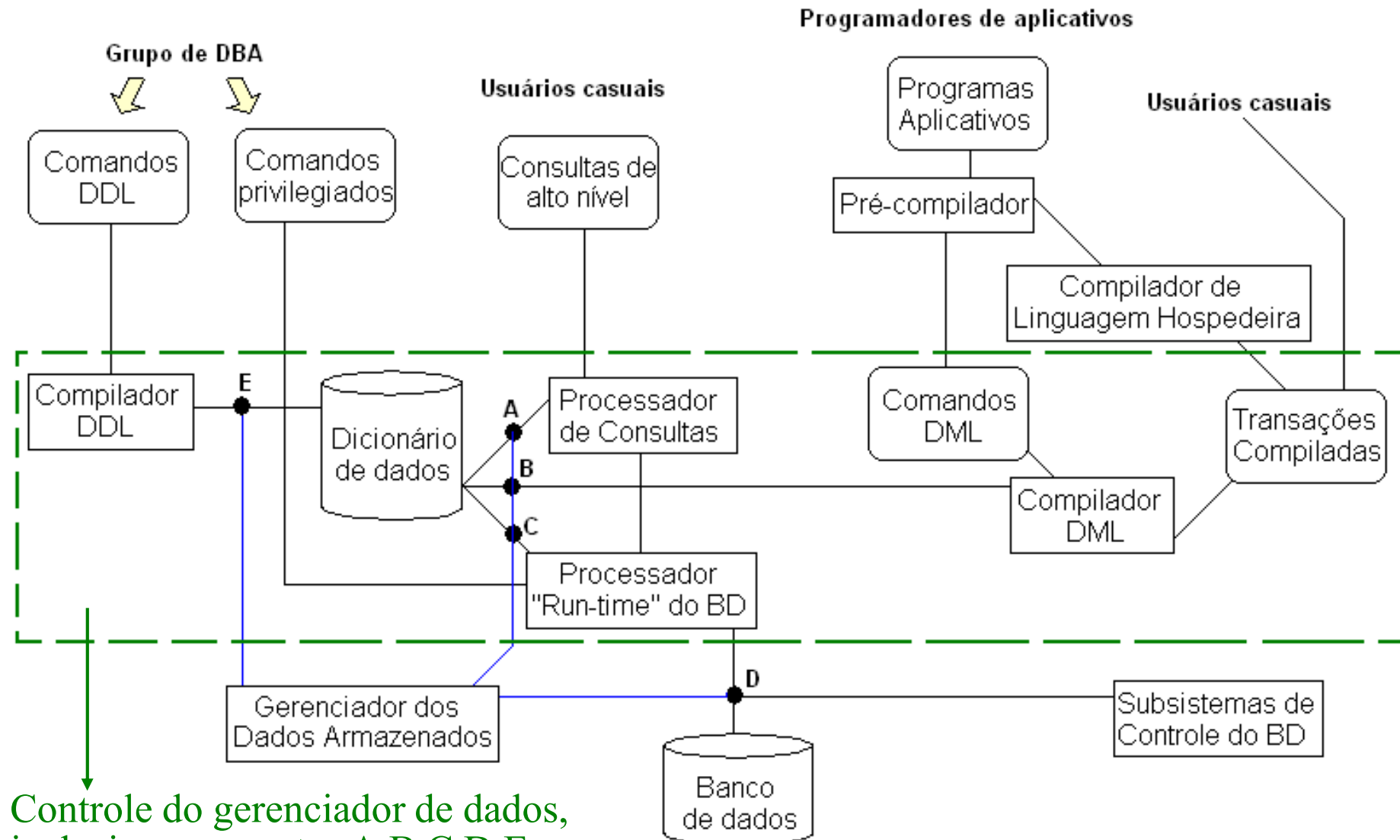
Cleyton Carvalho da Trindade

---

# Tópicos

- Componentes de um SGBD
  - Meio de Armazenamento Físico
  - Buffering de Blocos
  - Disposição de Registros de Arquivos em Disco
  - Organização de Arquivos
-

# Componentes Típicos do SGBD



---

# Características Gerais do Funcionamento

- A base de dados e seu catálogo são armazenados em disco;
  - O controle primário de acesso a disco é efetuado pelo SO por meio de funções de entrada/saída (E/S);
  - O Gerenciador de Acesso a Disco do SGBD usa serviços fornecidos pelo SO e controla o acesso aos dados do BD;
-

---

# Características Gerais do Funcionamento

- O **catálogo** contém informações como:
    - ❑ nome dos arquivos
    - ❑ itens de dados
    - ❑ detalhes de armazenamento de cada arquivo
    - ❑ informação de mapeamento entre os esquemas;
    - ❑ restrições;
  - O **compilador DDL** processa definições do esquema e armazena estas descrições no seu catálogo (metadados);
-

---

## Características Gerais do Funcionamento

- O **processador Runtime** do BD manipula requisições ao BD realizados em tempo de execução (recuperação e atualização);
  - O **processador de consultas** manipula as consultas fornecidas pelo usuário através de uma linguagem de alto nível (esses comandos serão analisados e será gerada chamadas a serem executadas pelo processador Runtime);
-

---

# Características Gerais do Funcionamento

- O **pré-compilador** extrai comandos DML a partir de um aplicativo escrito em uma linguagem de programação hospedeira, sendo estes comandos enviados ao compilador DML para a geração de código de acesso ao BD;
  - Os **códigos objetos** do DML e da **linguagem hospedeira** são link-editados, formando uma transação, cujo código executável acionará o **processador de Runtime**.
-

# Meio de Armazenamento Físico

---



---

# Hierarquia de armazenamento

- **Armazenamento primário** – Mídias que podem ser operadas diretamente pela CPU.
    - Acesso rápido, mas capacidade limitada
    - Memória principal, memória cache
  - **Armazenamento secundário** – Não podem ser operadas diretamente pela CPU. Eles devem ser copiados num armazenamento primário.
    - Possuem maior capacidade, menor custo, porém um acesso lento aos dados.
-

---

# Visão Geral dos Meios Físicos de Armazenamento

- **CACHE**: forma de armazenamento mais rápida e cara, sendo pequena e controlada pelo sistema operacional (SO)
  - **Memória Principal**: mídia de armazenamento para os dados que estão disponíveis para serem usados (ou trabalhados)
  - **Memória Flash** (EEPROM): velocidade próxima a memória principal, mas o seu armazenamento é mais “complicado” (apagar tudo de uma vez para regravação)
-

---

# Visão Geral dos Meios Físicos de Armazenamento

- **Disco Magnético:** mídia primária para o armazenamento de dados por longos períodos, além do acesso online (normalmente tem-se um BD inteiro armazenado nele)
  - **Disco Óptico:** forma mais popular de armazenamento óptico, onde os dados são armazenados opticamente e lidos por laser, podendo ser trocados por outros discos ópticos
-

---

## Visão Geral dos Meios Físicos de Armazenamento

- **Fita Magnética**: dispositivos mais baratos e lentos porque precisam ser acessados seqüencialmente a partir do início.
  - As fitas são comumente usadas para **backup e armazenamento de dados**. Elas possuem grande capacidade de armazenamento e podem ser removidas da unidade de fita (troçadas).
-

---

# Formas de Acesso

- **On-line** – Podem ser acessado diretamente, a qualquer momento
  - **Off-line** – É necessário a intervenção de um operador ou algum dispositivo automático de carga.
-

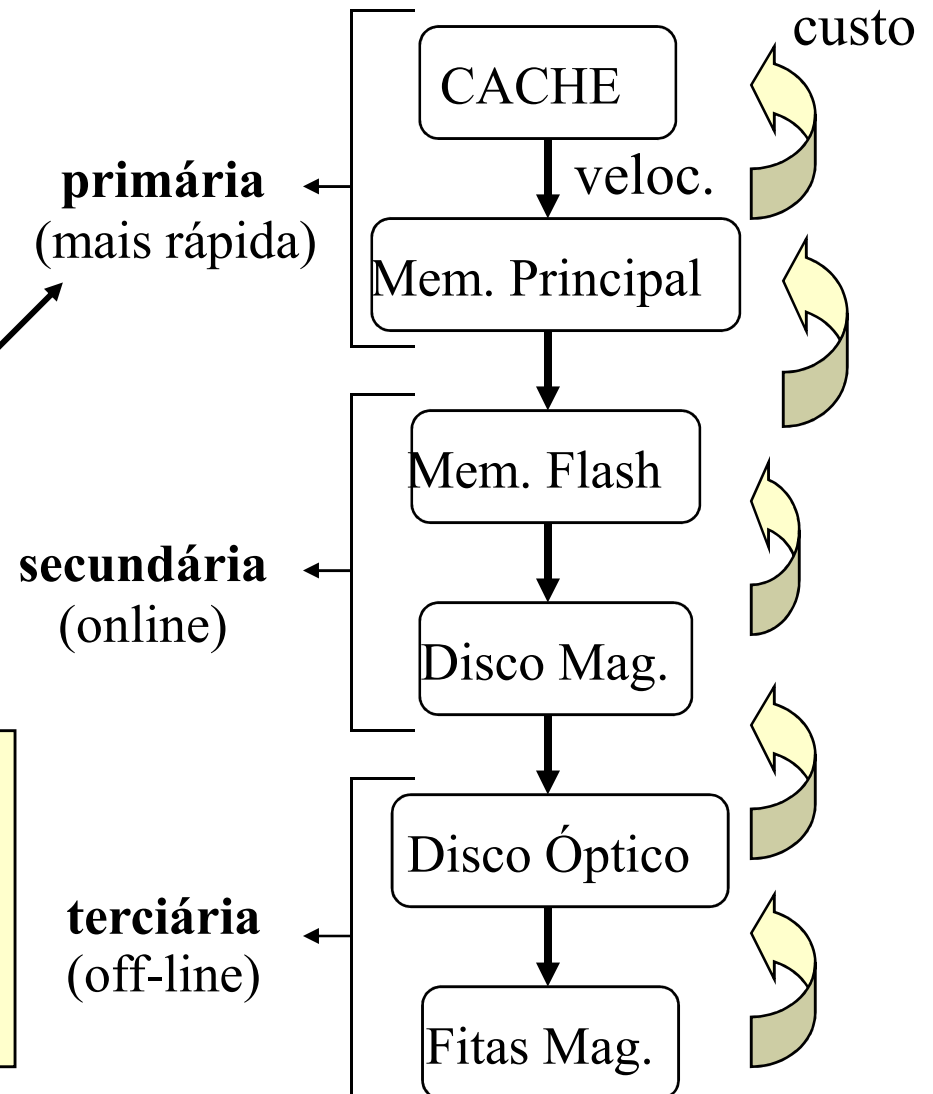
# Armazenamento de registro

Aspectos importantes na escolha:

- custo
- velocidade
- Capacidade de armazenamento

**Perda de dados com a falta de energia (volátil)**

Os discos magnéticos fornecem o maior volume de memória secundária dos sistemas computacionais modernos.

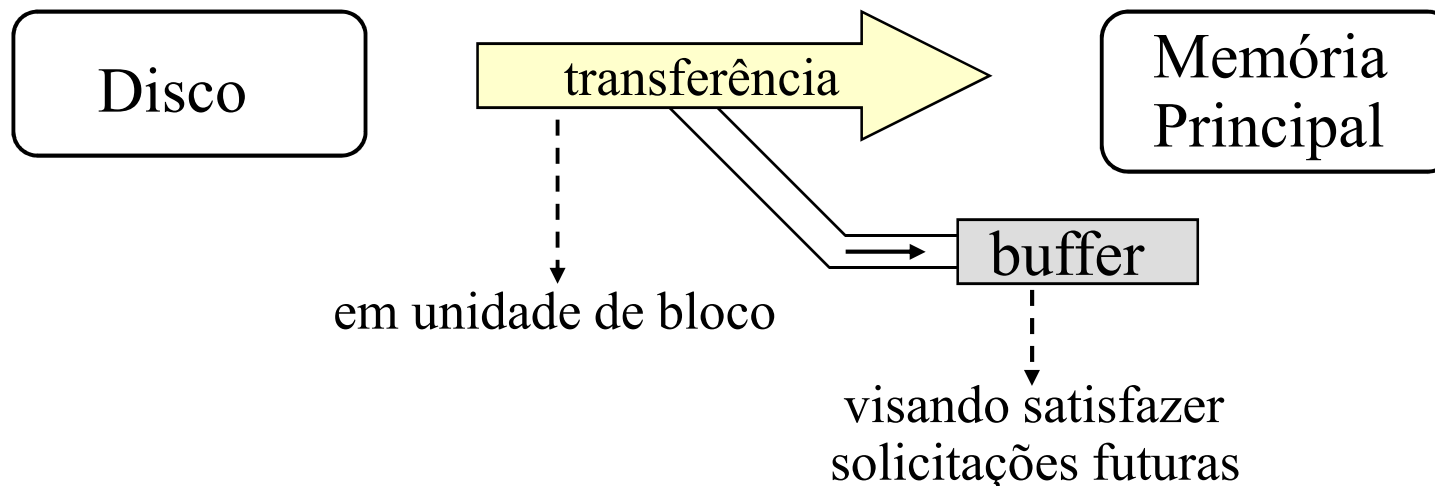


# Buffering de Blocos

---

# Acesso ao armazenamento

- um BD é mapeado em um número de arquivos diferentes que são mantidos pelo SO associado;
- esses arquivos residem permanentemente em discos magnéticos, com backups normalmente em fitas magnéticas;





---

## Acesso ao armazenamento

- cada arquivo é particionado em unidades de armazenamento de tamanho fixo chamadas de blocos (tanto para unidades novas de alocações como de transferência de dados = 4 KB)
-

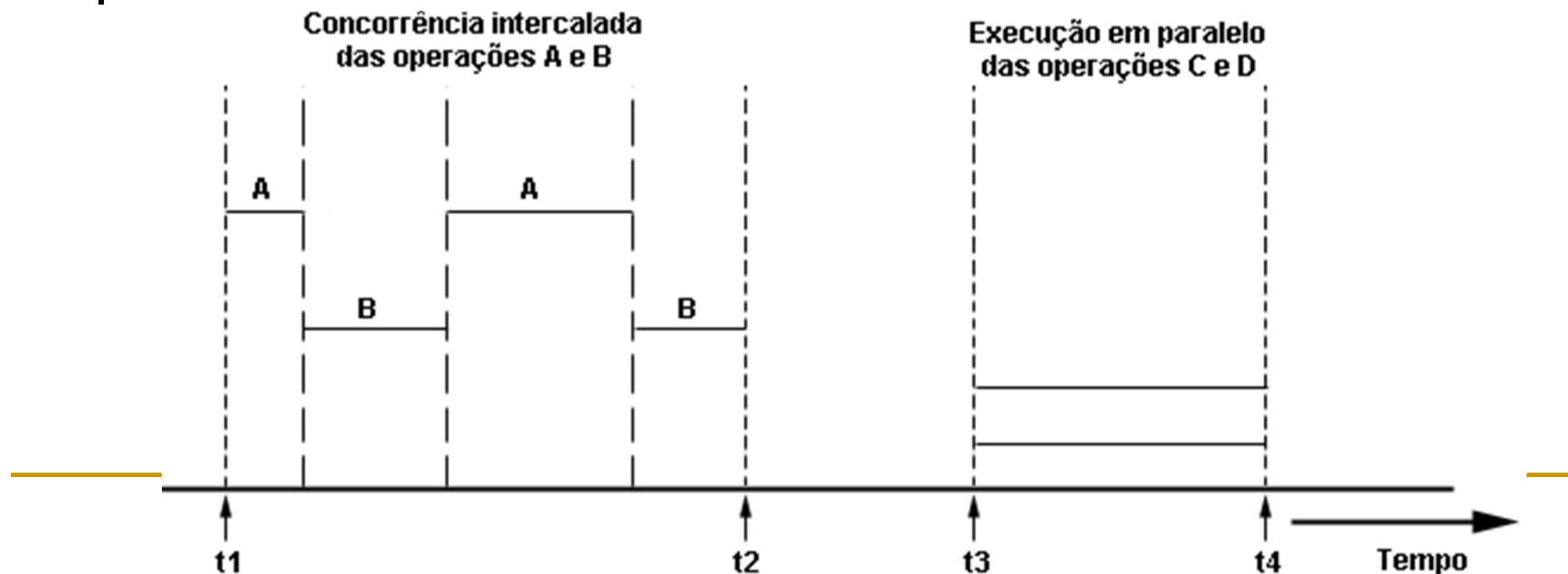
---

# Buffering de Blocos

- **Buffer** – é uma região de memória temporária utilizada para escrita e leitura de dados.
  - Parte da memória principal disponível para o armazenamento de cópias dos blocos de disco, apesar de sempre existir uma cópia em disco para cada bloco, podendo esta cópia, estar mais desatualizada que o buffer
  - Normalmente são utilizados quando existe uma diferença entre a taxa em que os dados são recebidos e a taxa em que eles podem ser processados, ou no caso em que essas taxas são variáveis.
-

# Buffering de Blocos

- Para transferência de diversos blocos do disco para a memória principal, com todos os endereços de blocos conhecidos
- Enquanto um buffer é lido/escrito, a CPU pode processar os dados de outro buffer



---

# Buffering de Blocos

- Minimizar o número de transferências de blocos entre o disco e a memória;
    - manter o máximo de blocos possíveis na memória principal, maximizando a chance do bloco desejado já estar na memória;
    - evitando ou reduzindo o número de acessos ao disco;
  - Gerenciar o espaço disponível na memória principal
    - não é possível manter todos os blocos na memória principal;
    - utilização de buffer (memória temporária);
-

# Disposição de Registros de Arquivos em Disco

---

---

# Registro e Tipos de Registro

- **Dados** são armazenados na forma de registros
  - Cada **registro** consiste de uma coleção de valores ou item relacionados
    - Cada valor corresponde a um **campo** de dado registro
  - Uma coleção de nomes de campo e seus tipos de dados constituem um **tipo de registro**
  - Os **tipos de dados** especifica os valores que um campo pode receber
-

---

# Registro e Tipos de Registro

■ Ex:

```
Struct empregado{  
    char nome[30];  
    char cpf[9];  
    int salario;           // Equivale a 4 bits  
    int codigo;  
    char departamento[20];  
}
```

---

---

# Arquivo, registro de tamanho fixo e variável

- Um **arquivo** é organizado logicamente como uma sequência de registros, sendo estes registros mapeados em blocos de disco.
  - Embora os blocos tenham um tamanho fixo determinado pelas propriedades físicas do disco e pelo SO, o tamanho dos registros pode variar.
    - **Registro de Tamanho Fixo** – Se todos os registros possuem o mesmo tamanho (bytes).
    - **Registro de Tamanho Variável** – Se registros diferentes num arquivo possui tamanho diferente.
-



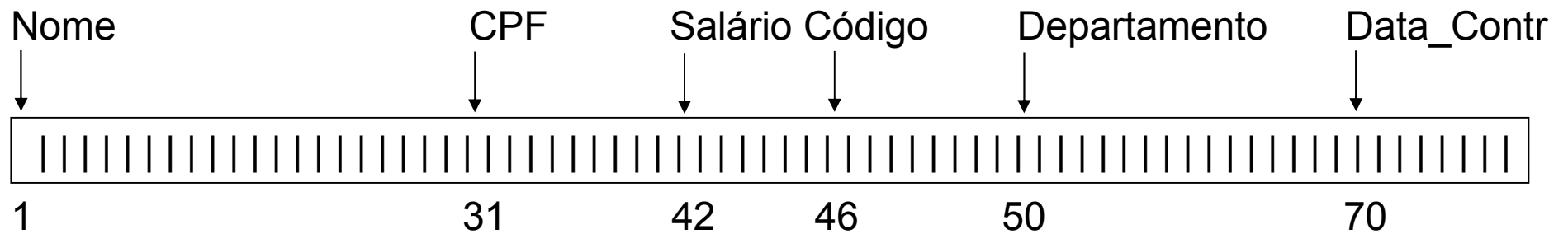
---

# Registro de Tamanho Variável

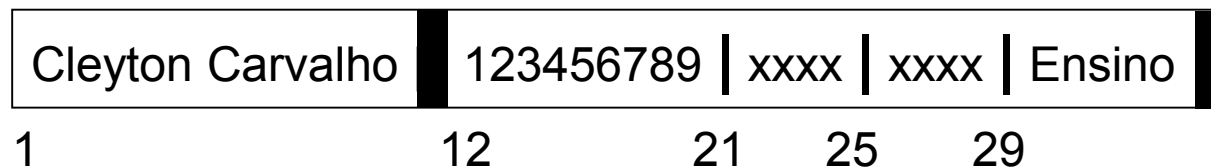
## ■ Razões

- ❑ Campos de tamanho variável (Nome do Empregado)
  - ❑ Campos multivalorados (Número de Telefone)
  - ❑ Campos opcionais
  - ❑ Arquivo com registros de diferentes tipos de registros (Registro de Aluno com o Boletim)
-

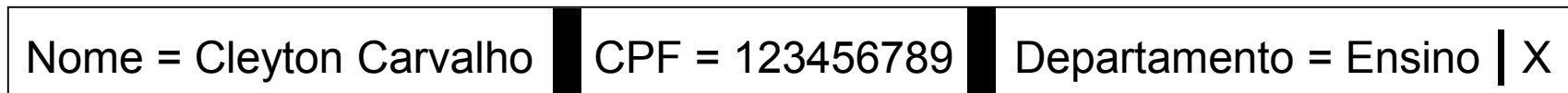
# Exemplos



Registro de tamanho fixo com 6 campos



Registro com dois campos de tamanho variável e três fixos



Registro de Tamanho variável com três tipos de caracteres separadores

---

# Divisão de Arquivos em Blocos

## Arquivos Spanned Versus Não-Spanned

- Os registros de um arquivo são alocados em blocos de disco
    - Unidade de transferência de dados entre o disco e a memória.
  - Quando o tamanho do bloco é maior que o tamanho do registro, cada bloco conterá vários registros.
    - Porém, sobram espaços sem uso.
-

# Divisão de Arquivos em Blocos

## Arquivos Spanned Versus Não-Spanned

- Armazena-se parte de um registro em um bloco e o restante em outro.
- **Spanned** – Permite fragmentar os registros em mais de um bloco.
  - Ao final do primeiro bloco um ponteiro aponta para o bloco que contém o restante do registro.



---

# Divisão de Arquivos em Blocos

## Arquivos Spanned Versus Não-Spanned

- **Não-Spanned** – Os registros não podem atravessar as fronteiras do bloco

Bloco i

Registro 1		Registro 2		Registro 3		\\\\\\\\\\\\\\\\
------------	--	------------	--	------------	--	------------------

Bloco i + 1

Registro 4		Registro 5		Registro 6		\\\\\\\\\\\\\\\\
------------	--	------------	--	------------	--	------------------

---

# Cabeçalhos de Arquivo

- Contém informações sobre o arquivo, necessárias aos programas de sistema que os acessa
    - ❑ Endereços do disco dos blocos de arquivo
    - ❑ Descrições de formato dos campos
    - ❑ Caracteres separadores
    - ❑ Etc...
-

---

# Pesquisa em Arquivos

- Os blocos são copiados nos buffers da memória principal.
  - Os programas procuram o(s) registro(s) dentro dos buffers usando a informação do cabeçalho do arquivo.
  - Se o endereço do bloco com o registro desejado não é conhecido, inicia-se uma pesquisa linear pelos blocos do arquivo.
-

---

# Pesquisa em Arquivos

- Em arquivos grandes isso consome muito tempo
  - Os arquivos devem ser organizados de forma a realizar um número mínimo de transferências de bloco.
-



# Organização de Arquivos

---

---

# Organização de Arquivos

- Arquivos de Registros Desordenados
  - Arquivos de Registros Ordenados
-

---

# Registros de Tamanho Fixo

- Observe a estrutura descrita a seguir, usando a linguagem de programação C.

```
struct Banco {  
    char Nome_agencia[25];  
    char Conta[11];  
    float Saldo;  
};
```

Lembrando que:  
char → caractere  
float → real  
1 char → 1 byte  
1 float → 4 bytes

---

---

# Exemplo usa a estrutura Banco

Arquivos contendo contas (Banco)

	<b>Agência</b>	<b>Conta</b>	<b>Saldo</b>
Registro 0	São Paulo	A-102	400
Registro 1	Brasília	A-501	500
Registro 2	Recife	A-723	700
Registro 3	Curitiba	A-439	900
Registro 4	São Paulo	A-101	750
Registro 5	Goiânia	A-607	250
Registro 6	Brasília	A-512	610
Registro 7	São Paulo	A-110	100

Banco → consome 40 bytes para armazenar cada registro

---

---

# Arquivos de Registros Desordenados

- Os registros são posicionados no arquivo segundo a ordem de inclusão.
  - Inclusão: Fácil
    - O ultimo bloco de disco do arquivo é copiado no buffer e o novo registro é acrescentado e o bloco é reescrito no disco.
  - Pesquisa: Linear
-

# Problemas

	Agência	Conta	Saldo
Registro 0	São Paulo	A-102	400
Registro 1	Brasília	A-501	500
Registro 2	Recife	A-723	700
Registro 3	Curitiba	A-439	900
Registro 4	São Paulo	A-101	750
Registro 5	Goiânia	A-607	250
Registro 6	Brasília	A-512	610
Registro 7	São Paulo	A-110	100

Registro 2 apagado e todos os  
outros registros deslocados



**Grande quantidade de  
deslocamento de registros**

	Agência	Conta	Saldo
Registro 0	São Paulo	A-102	400
Registro 1	Brasília	A-501	500
Registro 3	Curitiba	A-439	900
Registro 4	São Paulo	A-101	750
Registro 5	Goiânia	A-607	250
Registro 6	Brasília	A-512	610
Registro 7	São Paulo	A-110	100

# Problemas

Registro 2 apagado e o último registro deslocado em sua posição



**Acessos adicionais aos blocos**

	Agência	Conta	Saldo
Registro 0	São Paulo	A-102	400
Registro 1	Brasília	A-501	500
Registro 7	São Paulo	A-110	100
Registro 3	Curitiba	A-439	900
Registro 4	São Paulo	A-101	750
Registro 5	Goiânia	A-607	250
Registro 6	Brasília	A-512	610

Outra alternativa seria observar que as inserções tendem a ser mais frequentes que as remoções, sendo aceitável deixar o espaço livre e marcar o registro apagado, para no momento de uma nova inserção, ele ser preenchido.



Porém será difícil (onerosa) a localização do registro com uma marca simples no momento da inserção, por isso será introduzida uma estrutura adicional (cabeçalho e ponteiros).

# Problemas

Incluir no arquivo um cabeçalho que conterà as suas informações. Observe o exemplo que apaga os registros 2,4 e 6.

Cabeçalho					
		Agência	Conta	Saldo	
Registro 0		São Paulo	A-102	400	
Registro 1		Brasília	A-501	500	
Registro 2					←
Registro 3		Curitiba	A-439	900	
Registro 4					←
Registro 5		Goiânia	A-607	250	
Registro 6					←
Registro 7		São Paulo	A-110	100	≡

lista livre ←

O cabeçalho guarda o endereço do primeiro registro apagado, enquanto que neste primeiro registro será armazenado o endereço do segundo registro e assim por diante.



---

# Problemas

## O processo de Inserção

- troca o endereço apontado pelo cabeçalho;
- senão houver espaço removido disponível, inserir no final do arquivo (registro novo);

### **CUIDADO:**

Mover ou apagar um registro que é apontado fará com que o ponteiro não aponte mais para o registro desejado, gerando o conhecido ponteiro pendente.

Para estas situações é interessante que os registros sejam imobilizados, não permitindo a sua movimentação ou remoção.

---

---

# Arquivos de Registros Ordenados

- Os registros são armazenados em ordem seqüencial da chave primária de cada registro.
  - Características
    - Armazena os registros fisicamente na ordem da chave primária, ou o mais próximo possível desta ordem;
    - Projetado para o processamento eficiente de registros classificados na ordem de alguma chave primária;
    - Rápida recuperação de registro baseados na chave primária (encadeia-se os registros entre si usando ponteiros);
    - Minimiza o número de acesso aos blocos no processamento de arquivos sequenciais (Pesquisa Binária);
-

# Arquivos de Registros Ordenados

- Registros são lidos na ordem de classificação, o que é muito útil no processo de apresentação e para certos algoritmos de consulta;
- Difícil manter em ordem após possíveis remoções e inserções;
  - onerosa a movimentação de muitos registros;
  - usar cadeia de ponteiros respeitando as regras:

Registro	Agência	Conta	Saldo	
0	Brasília	A-110	500	→
1	Brasília	A-132	610	←
2	Curitiba	A-439	900	←
3	Goiânia	A-512	250	←
4	Recife	A-607	700	←
5	São Paulo	A-671	750	←
6	São Paulo	A-781	400	←
7	São Paulo	A-823	100	←

---

# Arquivos de Registros Ordenados

- Reorganização do Arquivo
    - Processo que realiza a organização física dos registros em ordem seqüencial, obedecendo o valor do campo chave.
  - Processo oneroso, devendo ser efetuado no período de pouco uso do sistema;
  - Sua freqüência depende da freqüência de inserção de novos registros;
-

# Fundamentos de Banco de Dados

## Estrutura de um SGBD

Cleyton Carvalho da Trindade