Gerenciamento de disco e buffer

Banco de dados relacional

Introdução

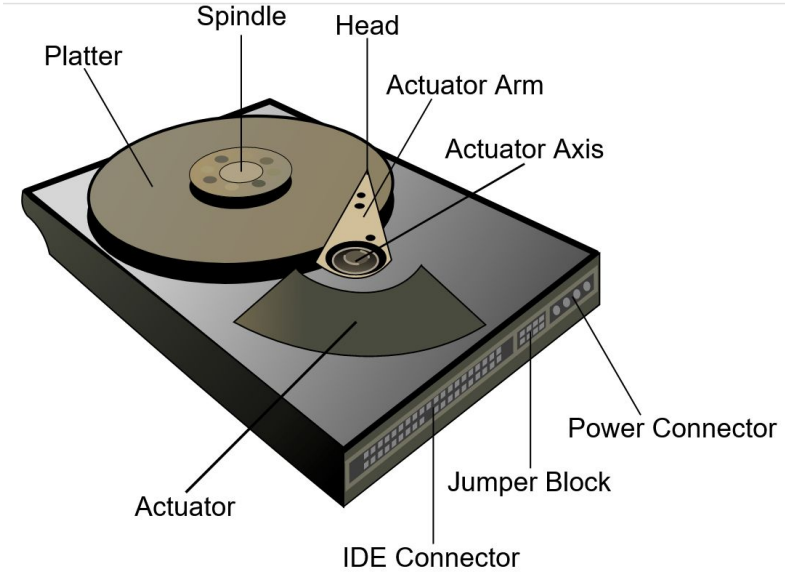
O que mais implica no projeto de um SGBD?

- Operações de escrita (write);
- Operações de leitura (read)

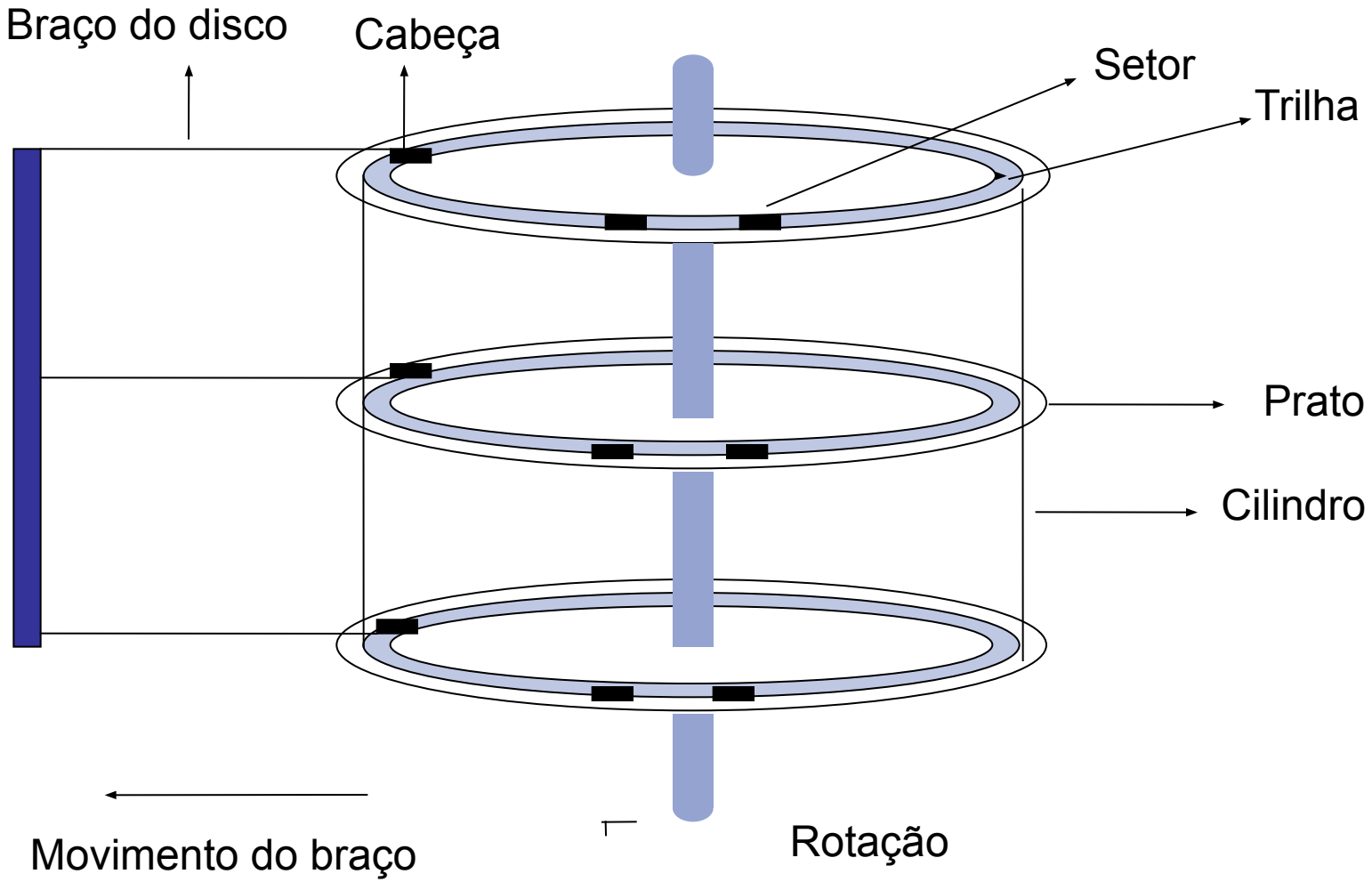
Discos Rígidos (HDs)

Dados são armazenados e recuperados em unidades chamadas *blocos de disco*.

O tempo para recuperar um bloco no disco rígido depende da posição onde se encontra.

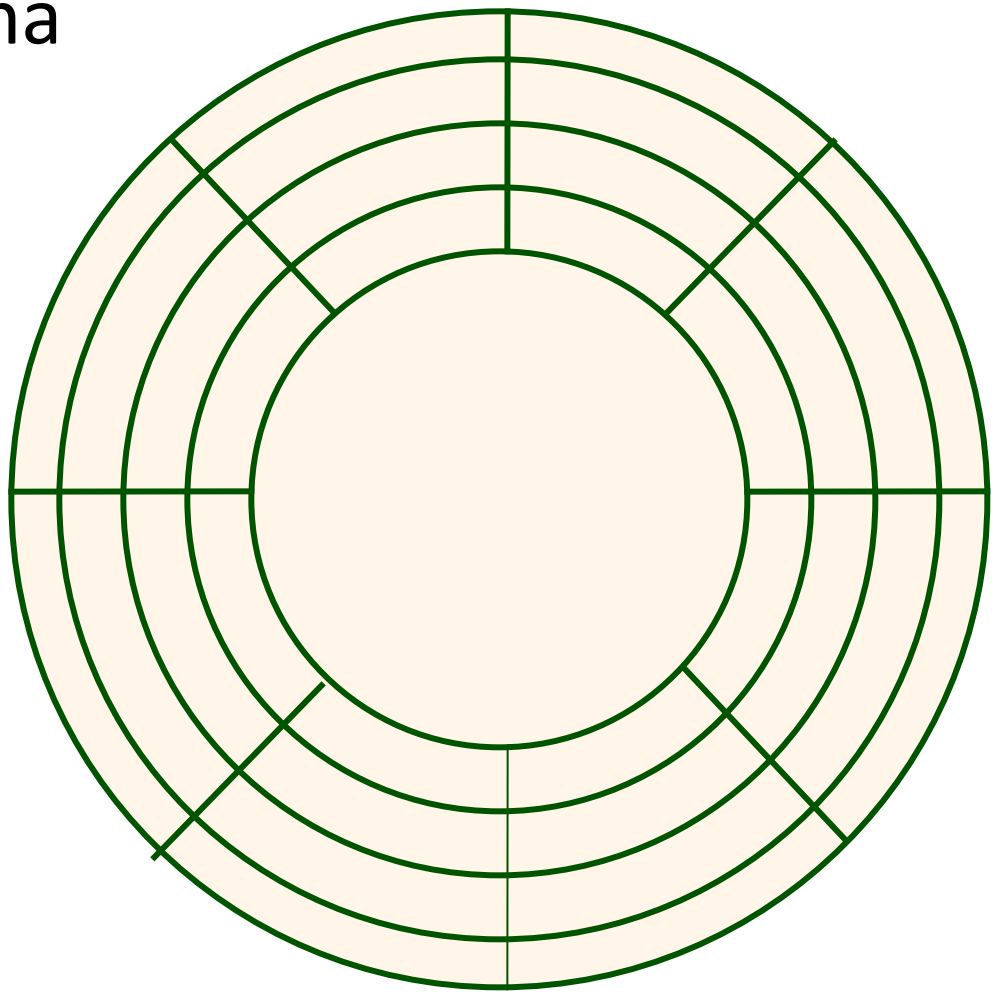


Estrutura do Disco Rígido

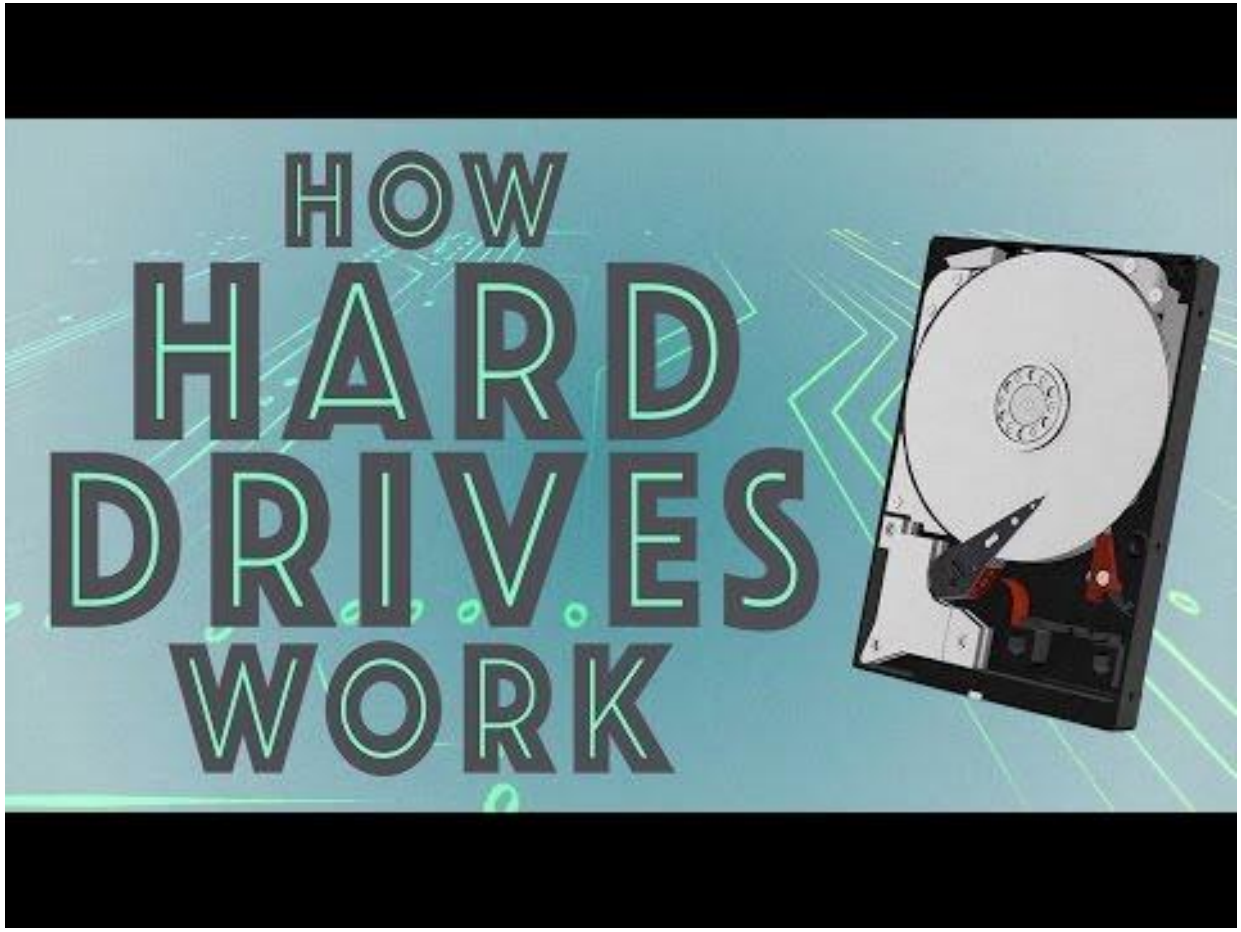


Componentes Disco Rígido

- Visto de cima



Componentes Disco



Operações no Disco Rígido

- Cabeça de Leitura e Gravação
  - Posicionada próxima a superfície do disco
  - Lê ou escreve dados codificados magneticamente
- Exemplo de 1 disco:

  -100 mil trilhas por polegada

  -1 milhão de bits por polegada nas trilhas
- Os setores são indivisíveis
- Gaps podem representar 10% do total da trilha

Tamanhos

- Trilha:
  - Tamanho é uma característica do disco que não pode ser alterado
- Bloco:
  - Tamanho pode ser configurado quando o disco é inicializado
  - Deve ser um múltiplo do tamanho de 1 setor

Tempos de Acesso

- Procura
  - Tempo para mover as cabeças dos discos para a trilha na qual um bloco desejado está localizado.
- Rotação
  - Tempo para que o bloco desejado se posicione sob a cabeça do disco = meia rotação em média;
- Transferência
  - Tempo para ler ou escrever no bloco = tempo de rotação do disco sobre o bloco.

Tempos de Acesso

- Rotação 7.200 rpm (1 rotação em 8,33 ms)
- 65.536 trilhas
- A disco “gasta” 1 ms para atravessar 4.000 trilhas. Assim, a cabeça “atravessa” o disco em 16,38 ms (65.536/4.000)
- Gaps ocupam 10% do espaço de uma trilha
- 1 Bloco == 4 setores
- 1 trilha tem 256 setores

Solid State Disks (SSD)

Sem tempo de busca (seek time)

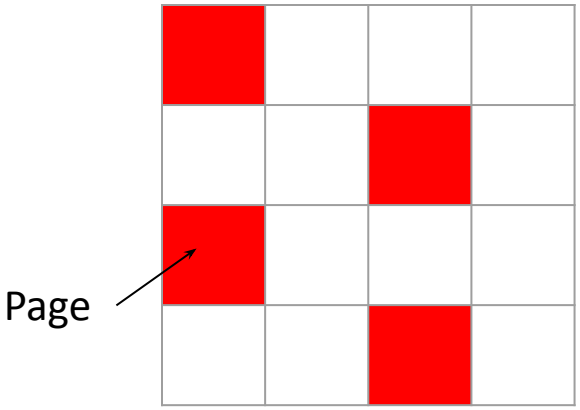
Sem latência de rotação (rotational latency)

Desafio

- Custo de escrita e leitura são diferentes

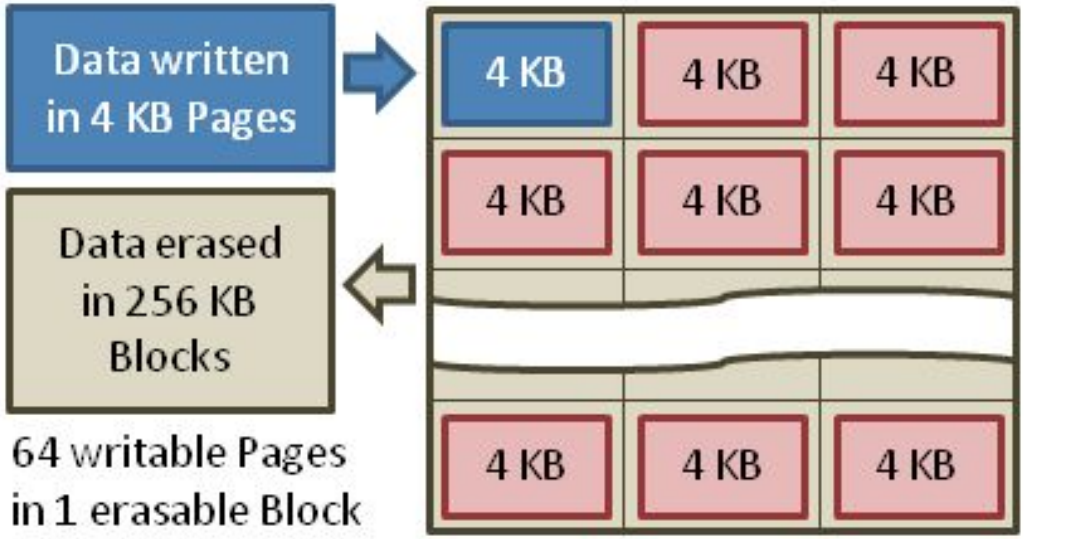
Escritas SSD

Block (256kb)



Operação	Área
Read	Page
Write	Page
Erase	Block

Escritas SSD

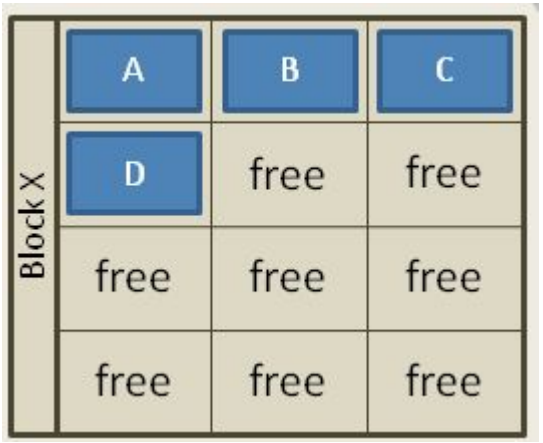


Typical NAND Flash Pages and Blocks

[Source]

Escritas no SSD

Write A, B, C, D

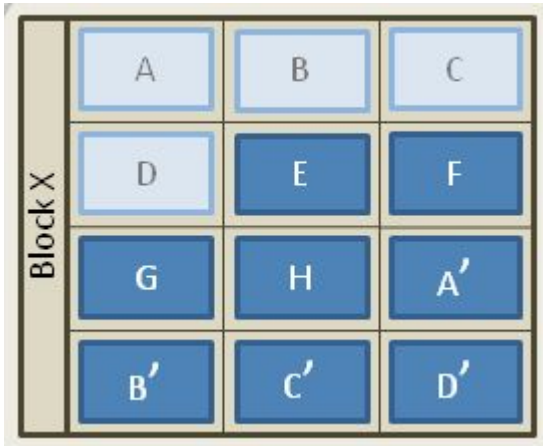


[Source]

Escritas no SSD

Write A, B, C, D

Write E, F, G, H



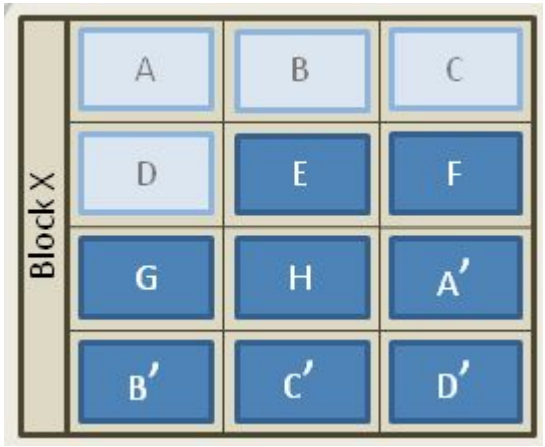
[Source]

Escritas no SSD

Write A, B, C, D

Write E, F, G, H

Write A’, B’, C’, D’



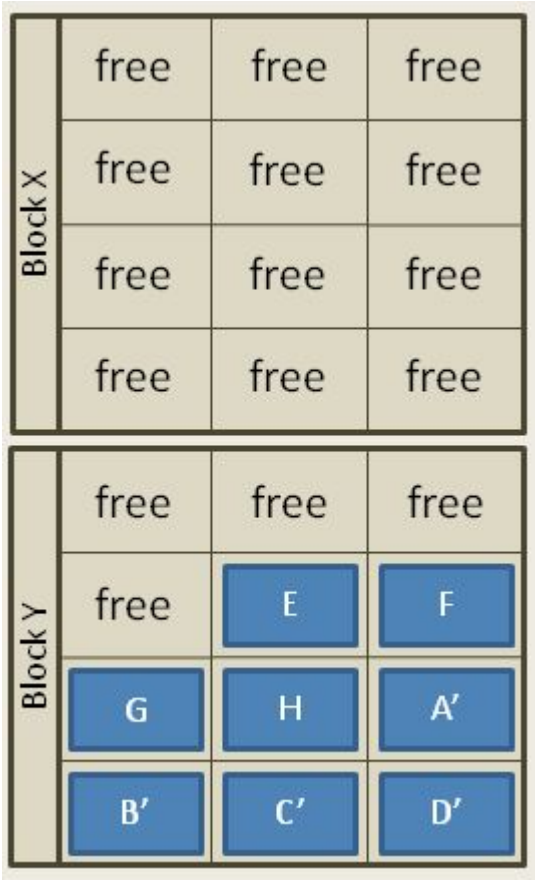
[Source]

Escritas no SSD

Write A, B, C, D

Write E, F, G, H, A’, B’, C’, D’

Páginas obsoletas são apagadas copiando o dado para um novo bloco



Quiz!

1 - True or False: Um bloco é a menor unidade endereçável no disco rígido.

2 - True or False: Um SSD tem custo de busca (seek time) igual a zero

3 - True or False: Para um HDD, as latências de leitura e escrita são similares

4 - True or False: Para um SSD, as latências de leitura e escrita são similares

slido

Para um SSD, as latências de leitura e escrita são similares

ⓘ Start presenting to display the poll results on this slide.

slido

A respeito da desfragmentação do ssd, é correto afirmar que

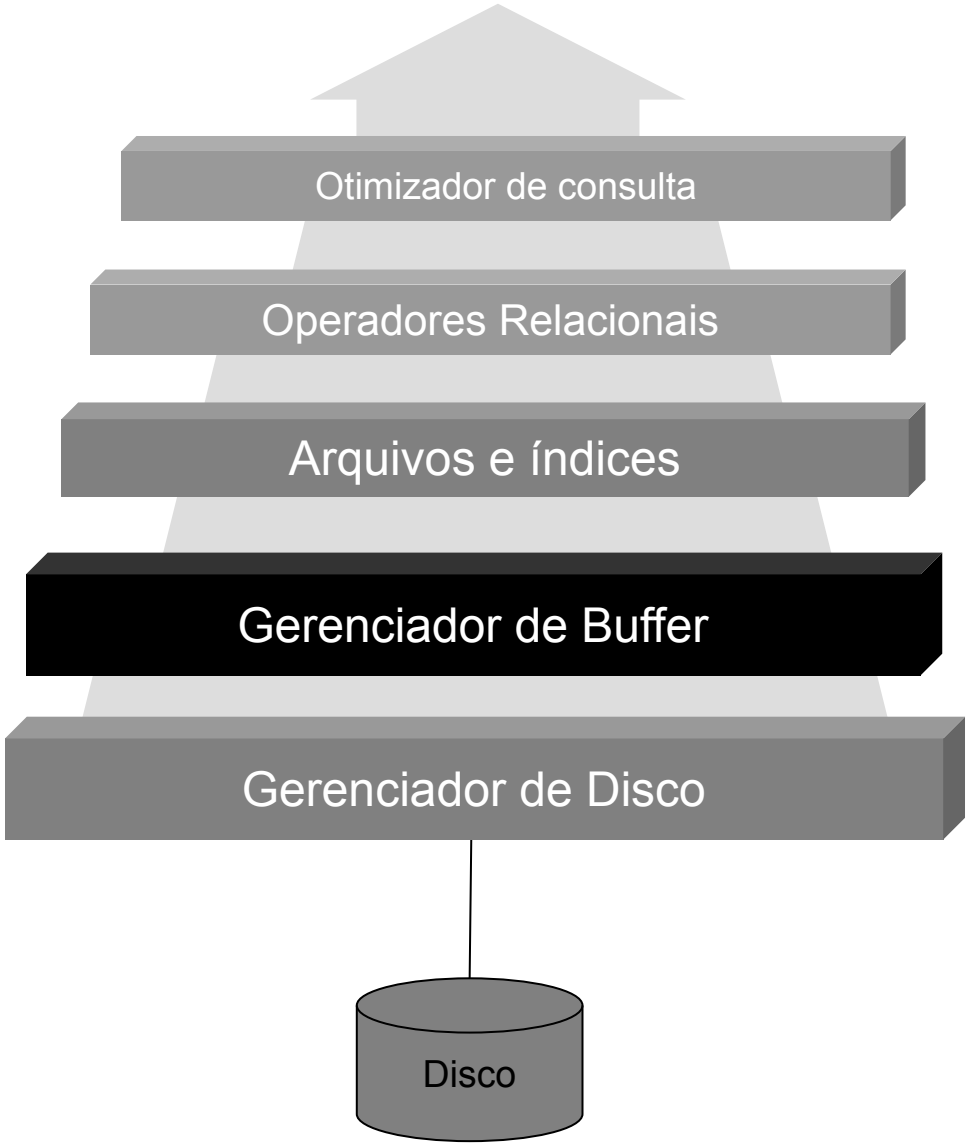
ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

slido

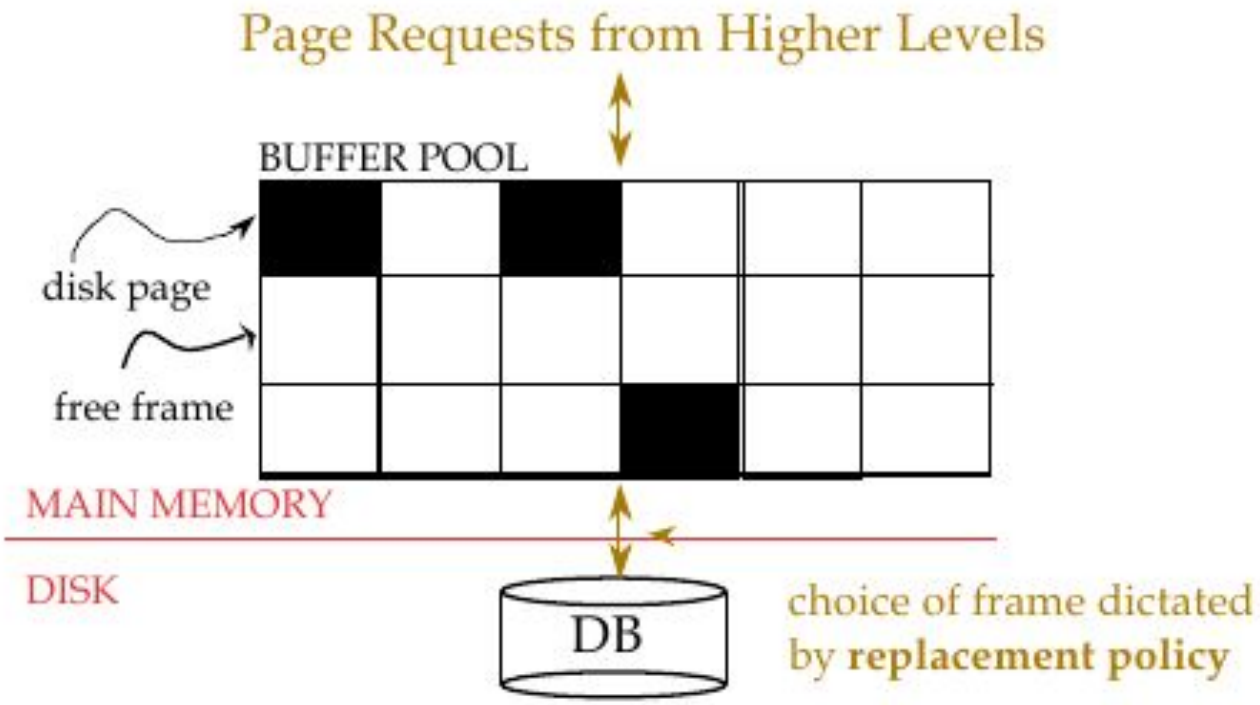
Assinale a opção que indica uma desvantagens do SSD.

ⓘ Click **Present with Slido** or install our Chrome extension to activate this poll while presenting.

Banco de dados relacional



Gerenciador de buffers



[Ramakrishna et. al. 3 edição, pag 264]

Otimização do Acesso

Buffer: porção da memória principal que armazena cópias dos blocos do disco.

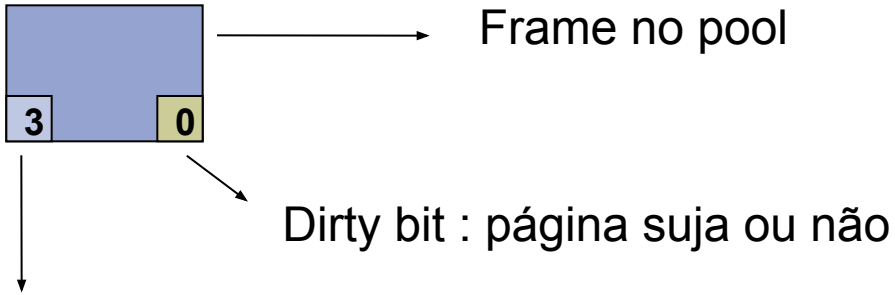
- O buffer é organizado em páginas que, geralmente, se relacionam 1:1 com os blocos

Gerenciador de buffer: subsistema de um SGBD responsável para gerenciar espaços no buffer

Gerenciador de Buffer

- Funções:
  - Testa se o dado procurado está no buffer
  - Traz a página do disco para a memória
  - Procura frames livres (espaço memória) para alocar a página
  - Aciona o algoritmo para liberar a página
  - Aloca a página
  - Caso o frame tenha que ser reutilizado, propaga a modificação no disco.

Algoritmo de Gerenciamento



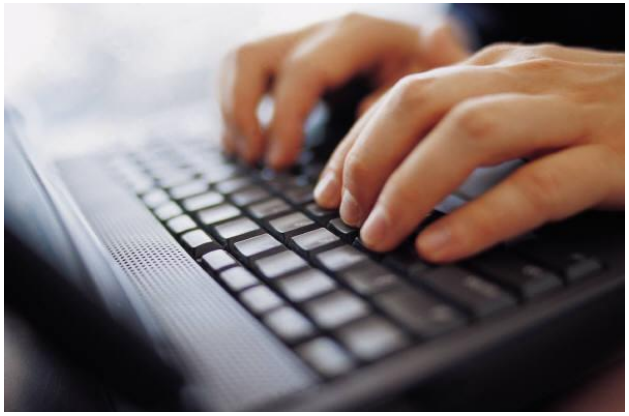
Pin-count = número de vezes que a página foi solicitada para consultas ou modificações mas não foi liberada ainda.

Inicialmente :

Dirty bit := 0

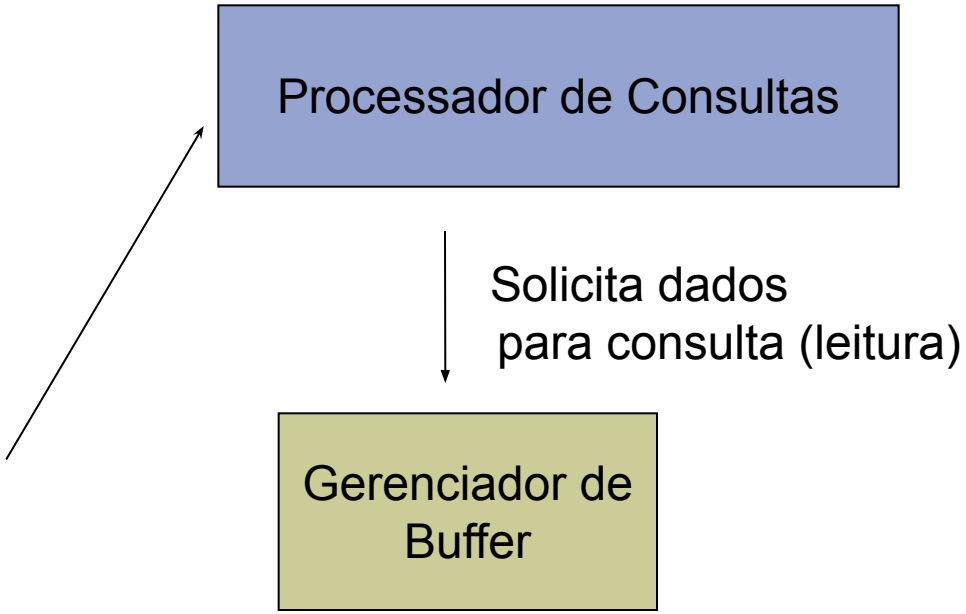
Pin-count := 0

Considere o seguinte cenário



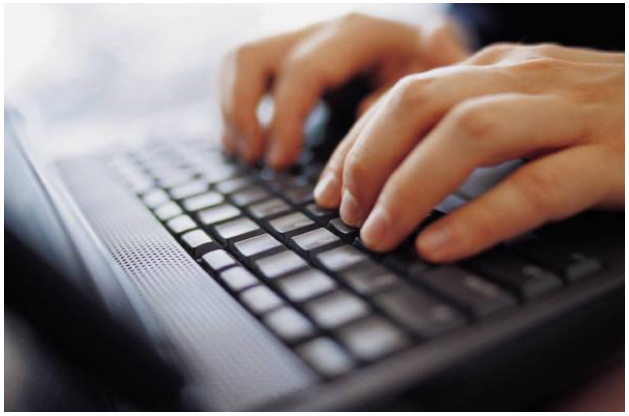
Usuário consulta banco de dados

```
SELECT *
      FROM emp
      WHERE cpf = 123
```



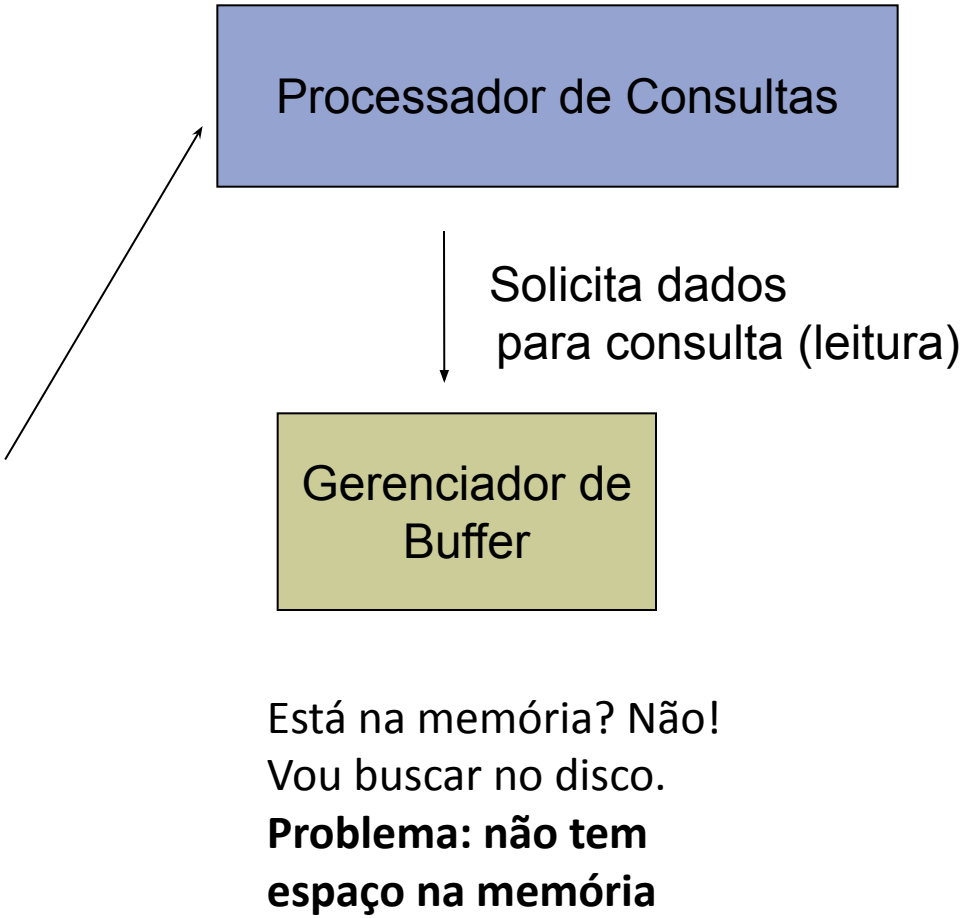
28

Considere o seguinte cenário



Usuário consulta banco de dados

```
SELECT *
    FROM emp
    WHERE cpf = 123
```



Está na memória? Não!
Vou buscar no disco.
Problema: não tem espaço na memória

Considere o seguinte cenário

- Verifica se existem frames com *pin-count* = 0
- **Caso positivo**: aciona gerenciador de disco
  - Gerenciador de **disco** providencia a transferência da página.
  - Gerenciador de **buffer** vai alocar a página em um frame com *pin-count* = 0
  - Qual frame será escolhido ?
    - Usa sua política de substituição (LRU, MRU, random)
  - Verifica o dirty bit deste frame

Considere o seguinte cenário

- **Dirty bit = 1**:
  - grava a página atual do frame no disco
  - Aloca a nova página no frame
  - Incrementa *pinout* do frame
  - Retorna o endereço do frame para o processador de consultas
- **Dirty bit = 0**:
  - Aloca a nova página no frame
  - Incrementa *pinout* do frame
  - Retorna o endereço do frame para o processador de consultas

Banco de dados relacional

Arquivos

Blocos são transmitidos entre disco e memória, mas...

o SGBD trabalha no nível de registro e arquivos

Arquivo: uma coleção de páginas, que contém um conjunto de registros. Estes devem suportar:

- Inserção/remoção/atualização
- Busca de registro em particular
- Busca de todos os registros

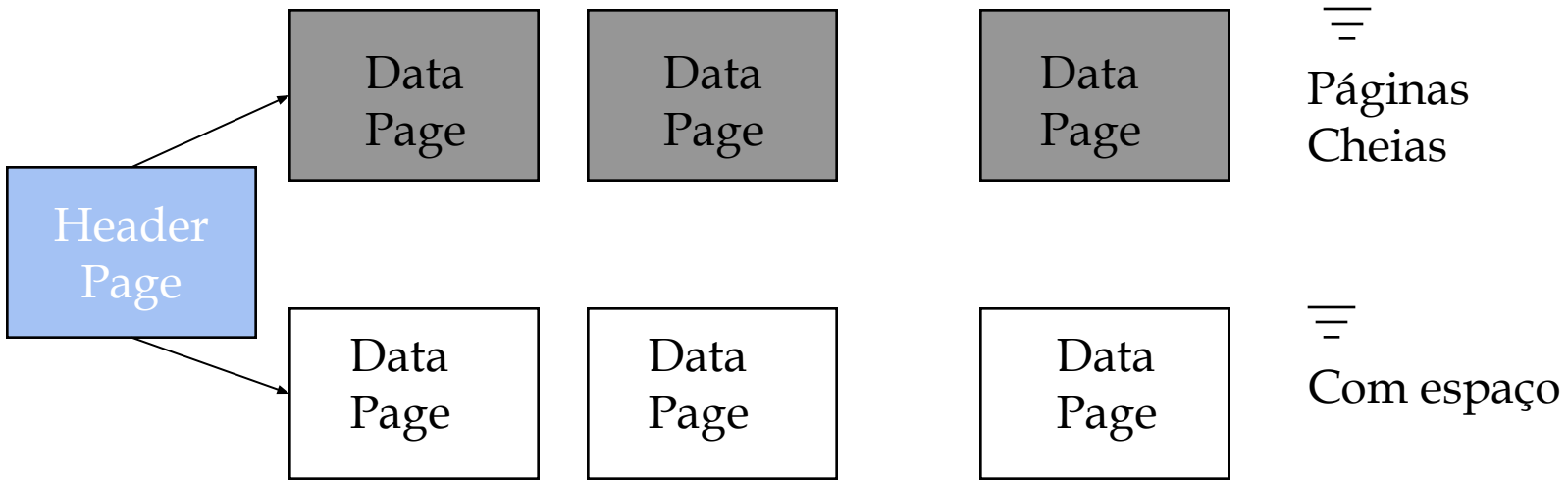
Arquivos não ordenados - Heap

Estrutura mais simples de armazenar registros sem ordenamento

Para suportar as operações, é importante:

- Manter o rastro das páginas no arquivos
- Manter o rastro das páginas vazias
- Manter o rastro dos registros em cada página

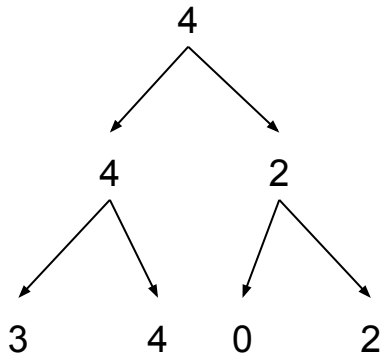
Heap - exemplo



O ponteiro da página inicial (header page) deve ser armazenado;

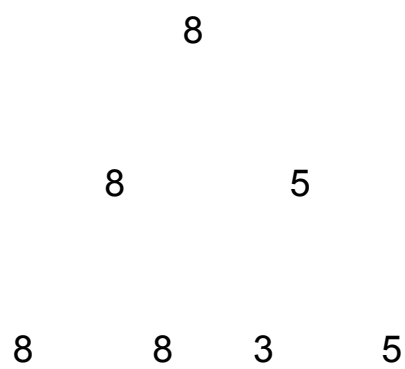
Max Heap no Postgres

Como encontrar um espaço nas páginas para armazenar um registro?



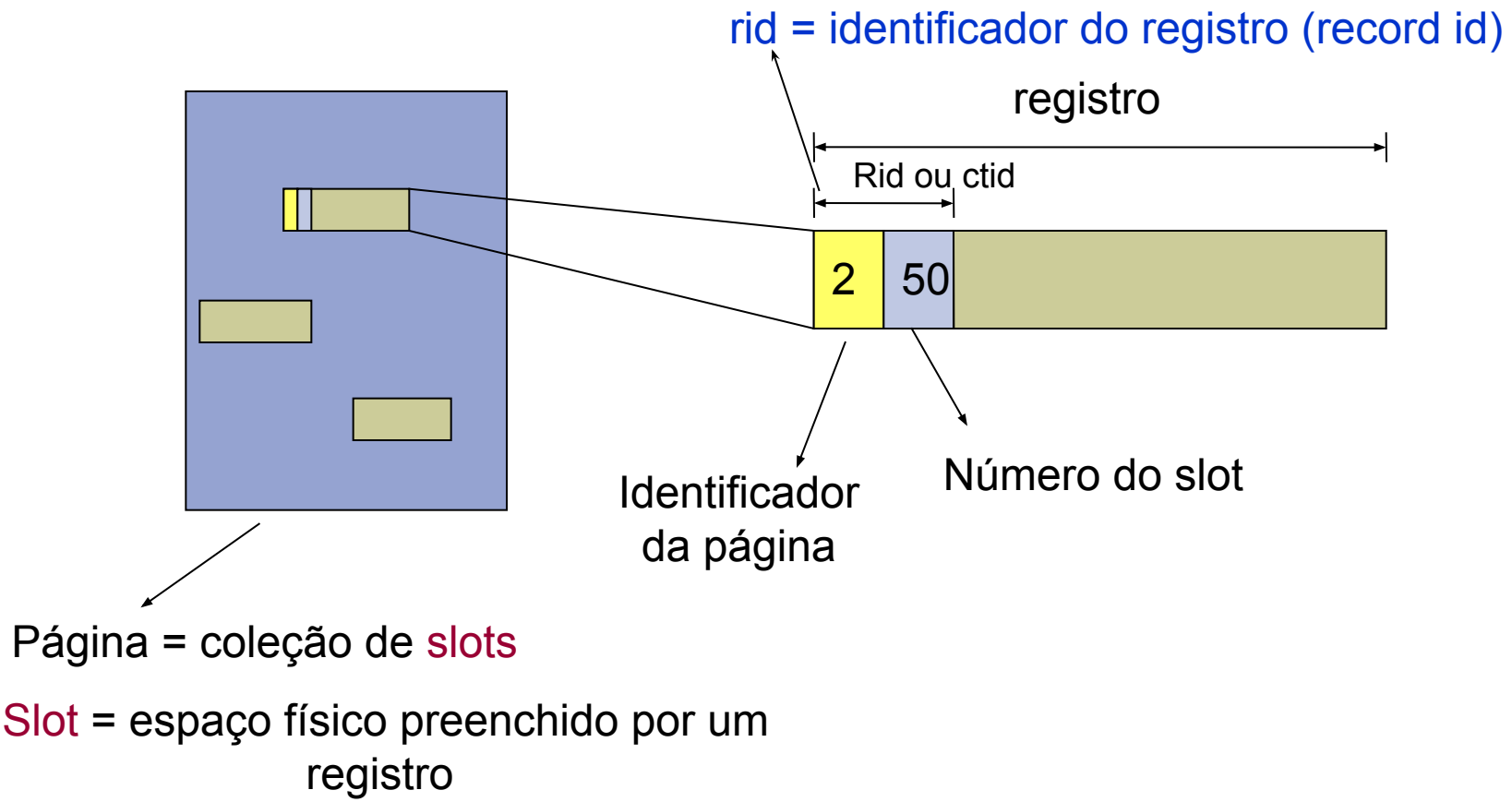
Atividade

1-Na árvore abaixo, é possível encontrar 10 unidades de espaço disponíveis? Explique



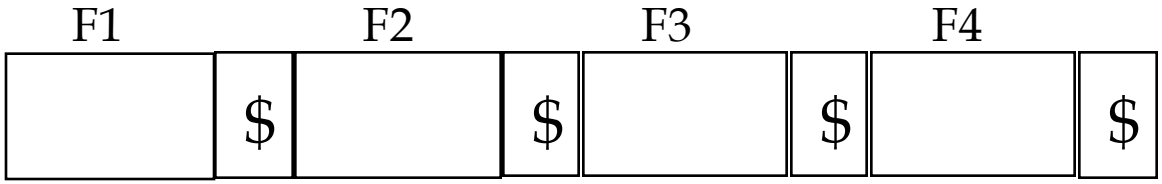
2) Atualize a árvore acima após a utilização de 4 unidades de espaço.

Organização dos registros no disco

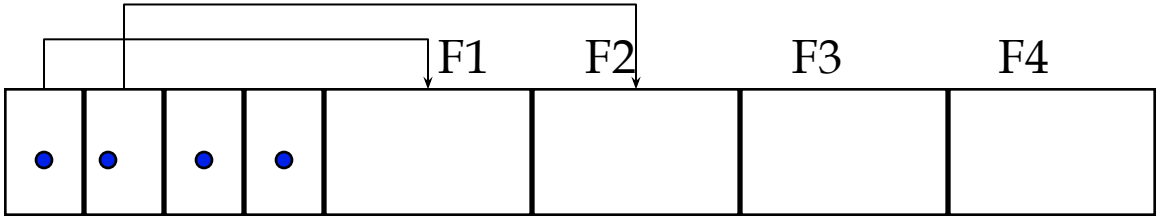


Formato dos registros no disco

Duas alternativas para tamanhos variados

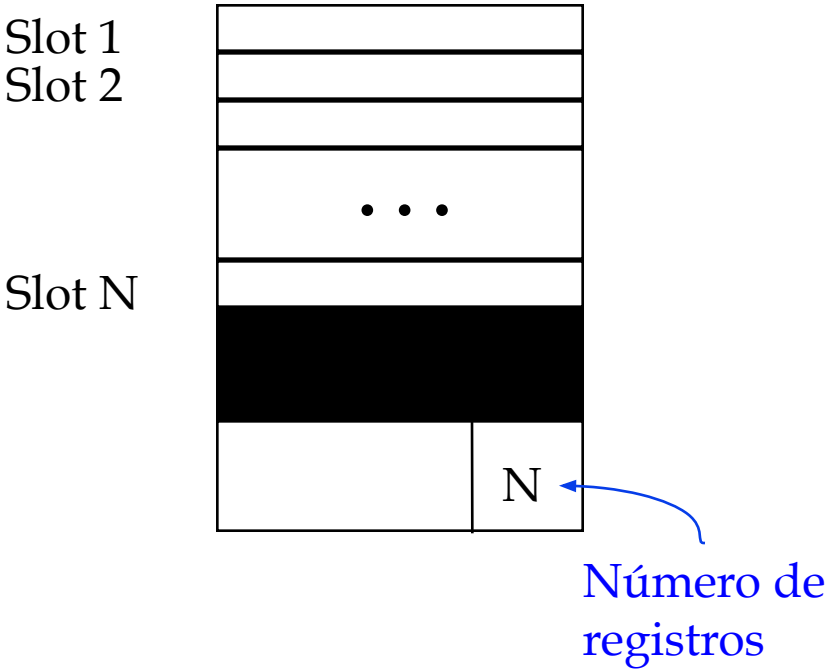


Campos determinados por caractere especial



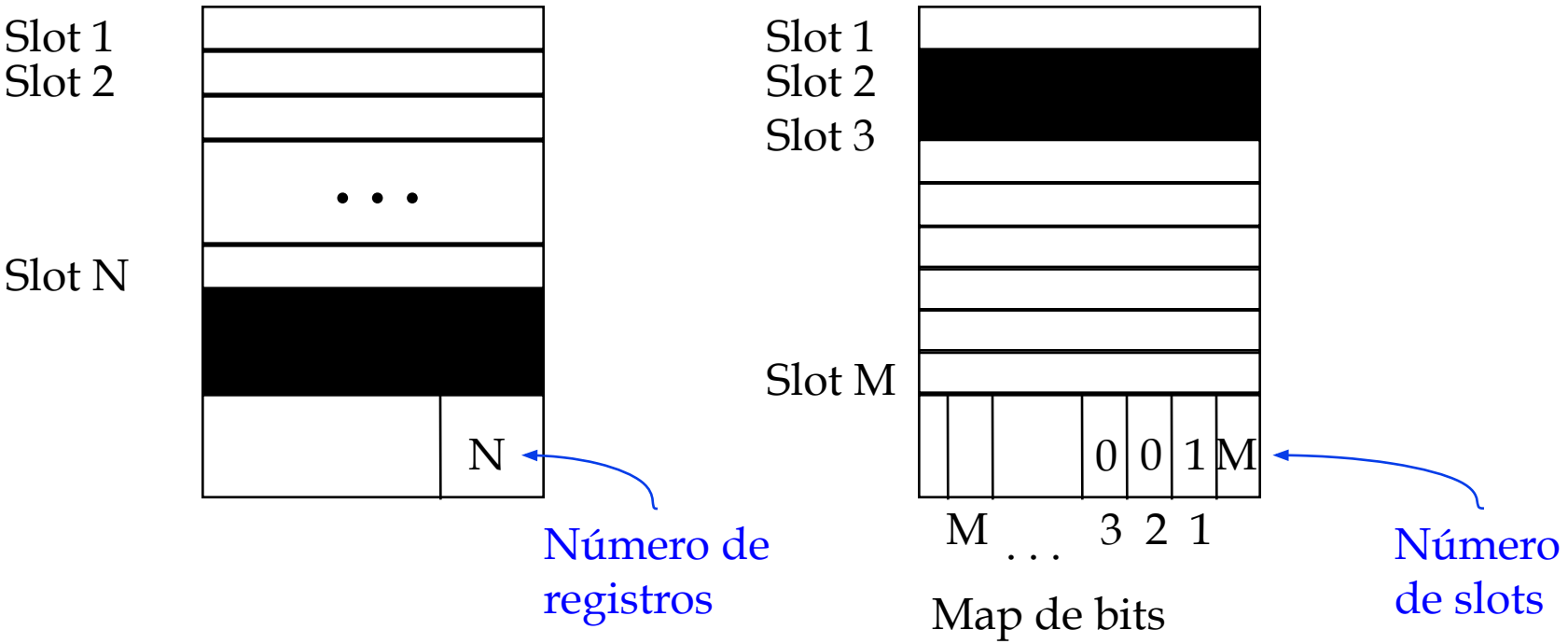
Array de offsets

Páginas com formato fixo



Record id = <page id, slot #>.

Páginas com formato fixo



Record id ou ctid = <page id, slot #>.

Atividade A

create table Movie(name char(30), address char (255), data date*)

*date ocupa 10 bytes

Exemplo: Suponha que os registros da tabela Movie serão armazenados em páginas de 4kb. O cabeçalho do registro ocupa 12 bytes (ponteiro para o esquema, tamanho registro). Quantos registros cabem na página, sabendo que o mapa de bits ocupa 20 bytes?

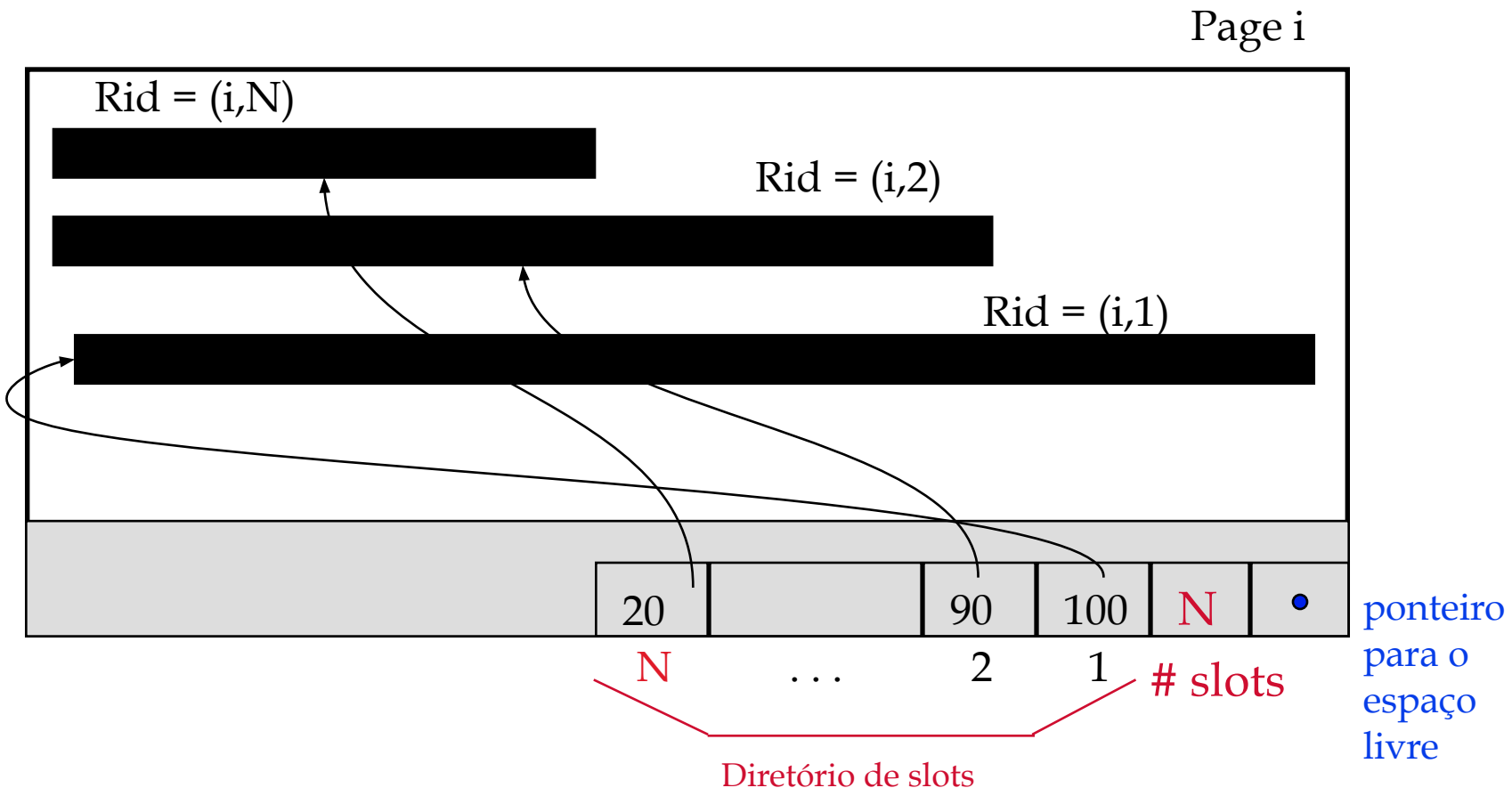
Atividade B

1- Construa um diagrama de página de tamanho fixo usando um mapa de bits. A tabela armazena 20 registros do tipo char[12].

A) Insira 3 registros com qualquer informação.

B) Apague o registro 0 e 2 previamente inseridos

Registros com tamanhos variados



- *É possível mover registros sem alterar o RID*

Exemplo

O que acontece na Atividade abaixo, em relação ao espaço ocupado pelos registros, se o atributo “name” fosse alterado para varchar(255)?

create table Movie(name char(30), address char (255), data date*) *date ocupa 10 bytes

Construa um diagrama de página de tamanho variado usando um mapa de bits. A tabela armazena 20 registros de 12 bytes cada.

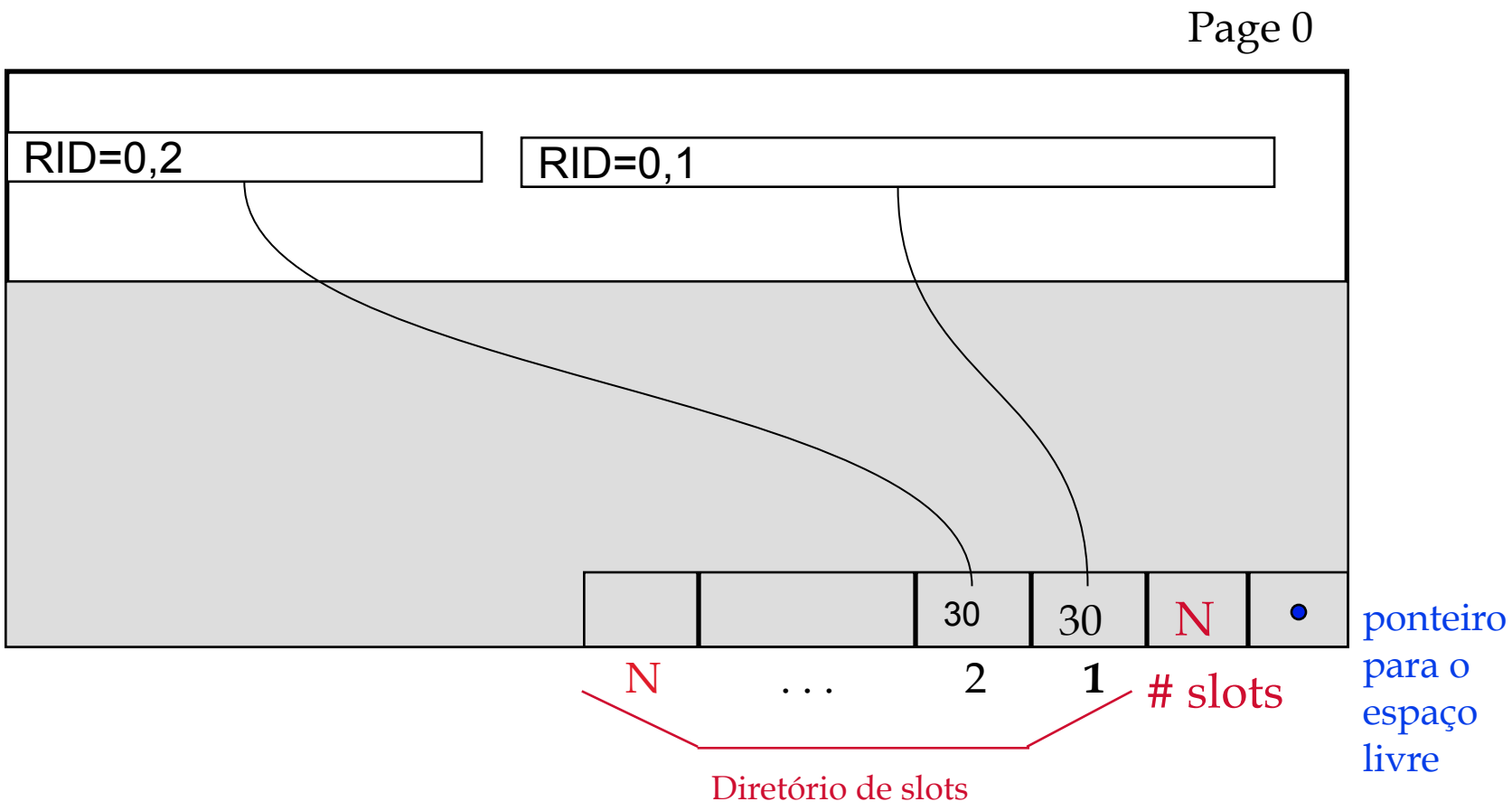
A) Insira 3 registros com qq informação

B) Apague o registro 0 e 2 previamente inseridos

Disco- exemplo

```
create table dados (id serial, nome varchar(50));
 #seleciona o número da transação atual
select ctid from dados;
 #acha o local dos dados
select pg_relation_filepath('dados');
 #mostra o dataset
sudo hexdump -C /var/lib/postgresql/14/main/*
 #insere dados
insert into dados(nome) values (“fsfsdsdfsd”)
```

Registros com tamanhos variados



Atividade C

Insira os registros abaixo em uma página com tamanho variado e monte o diagrama de página. Os registros têm 200 bytes no máximo cada. Tamanho total da página é 4kb.

Create table teste (varchar[200] x)

1) Inserir : “A”, “BBBBBBBB” , “DDDDDDDDDDDDDDD"
2) Demonstre o que acontece caso o primeiro registro seja alterado para “ABXXXXX”
3) Demonstre o que acontece caso os registros terem seu tamanho alterado para varchar[300].

Atividade C’

Insira os registros abaixo em uma página com tamanho variado no Postgres e monte o diagrama de página. Os registros tem 200 bytes no máximo cada. Tamanho total da página 4kb.

CREATE TABLE teste (x varchar(200));

1) Inserir : “A”, “BBBBBBBB” , “DDDDDDDDDDDDDDD"
2) Demonstre o que acontece caso o primeiro registro seja alterado para “ABXXXXX”
3) Demonstre o que acontece caso os registros terem seu tamanho alterado para varchar[300].

Atividade D

Create table teste (varchar[30] x, int y)

Monte o diagrama do registro conforme abaixo:

A) Insira 3 registros na tabela acima.

<’a’,1> ; <’b’,2>; <’c’,3>

B) Remova o primeiro registro

C) Insira 1 novo registro.

<’x’,4>

Atividade E- Entregar

Create table teste (text nome, int idade, text end)

Monte o diagrama do registro conforme abaixo:

A) Insira 3 registros na tabela acima.

‘joao’,40, ‘rua getulio vargas 21’ ;

‘maria’,45, ‘rua porto alegre N 35E’ ;

‘pedro’,30,’rua getulio vargas 210’ ;

B) Remova o segundo registro

C) Insira 1 novo registro.

“joao pedro”,40,’rua italia 1’;

The Oversized-Attribute Storage Technique- TOAST

- Postgres não permite fragmentar registros
- Página possuem tamanho de 8kb
- Registro possuem tamanho máximo de 2kb

Possibilidade de fragmentar colunas (toast-The Oversized Attribute Storage Technique)

Índices (preview)

Um heap file permite recuperar registros:

- A partir de um RID ou Busca sequencial por todos os registros

Algumas vezes precisamos recuperar registros em alguma ordem:

- ÍNDICES!!!