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*

Operações em arquivos

- Recuperação
  - Apenas leitura

- Atualização
  - Inserção
  - Deleção
  - Atualização

Organização de Arquivos

- Heap
- Sequencial
- Hashing
- 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.

Operações em Heap

- **Acesso**
  - pesquisa linear bloco por bloco = um procedimento dispendioso
  - se apenas um registro satisfizer a condição de pesquisa, então, na média, um programa lerá a memória e pesquisará metade dos blocos de arquivo antes de encontrar o registro.
  - para um arquivo de **b** blocos, isso exige pesquisar (b/2) blocos, em média
  - se nenhum registro ou vários registros satisfizerem a condição de pesquisa, o programa deve ler e pesquisar todos os b blocos no arquivo
- **Inserção**
  - a inserção de um novo registro é muito eficiente
  - o último bloco de disco do arquivo é copiado para um buffer, o novo registro é acrescentado e o bloco é então regravado de volta no disco.
  - O endereço do último bloco de arquivo é mantido no cabeçalho do arquivo.
- **Deleção**
  - deixa espaço livre no bloco = desperdício de espaço de armazenamento
  - bit como marcador de exclusão = exige reorganização periódica do arquivo

Arquivos Sequenciais

- Registros fisicamente ordenados por 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
  - Pesquisa binária para localizar o primeiro registro
- **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					
	Abílio, Diana					
	⋮					
	Acosta, Marcos					
Bloco 2	Adams, João					
	Adams, Roberto					
	⋮					
	Akers, Janete					
Bloco 3	Alexandre, Eduardo					
	Alfredo, Roberto					
	⋮					
	Allen, Samuel					
Bloco 4	Allen, Tiago					
	Anderson, Kely					
	⋮					
	Anderson, Joel					
Bloco 5	Anderson, Isaac					
	Angeli, José					
	⋮					
	Anita, Sueli					
Bloco 6	Arnold, Marcelo					
	Arnold, Estêvão					
	⋮					
	Atkins, Timóteo					
	⋮					
Bloco *n*–1	Wong, Jaime					
	Wood, Ronaldo					
	⋮					
	Woods, Manuel					
Bloco *n*	Wright, Pâmela					
	Wyatt, Charles					
	⋮					
	Zimmer, André					

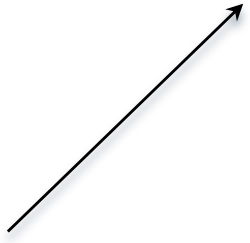
Arquivo Hashing

- 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

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

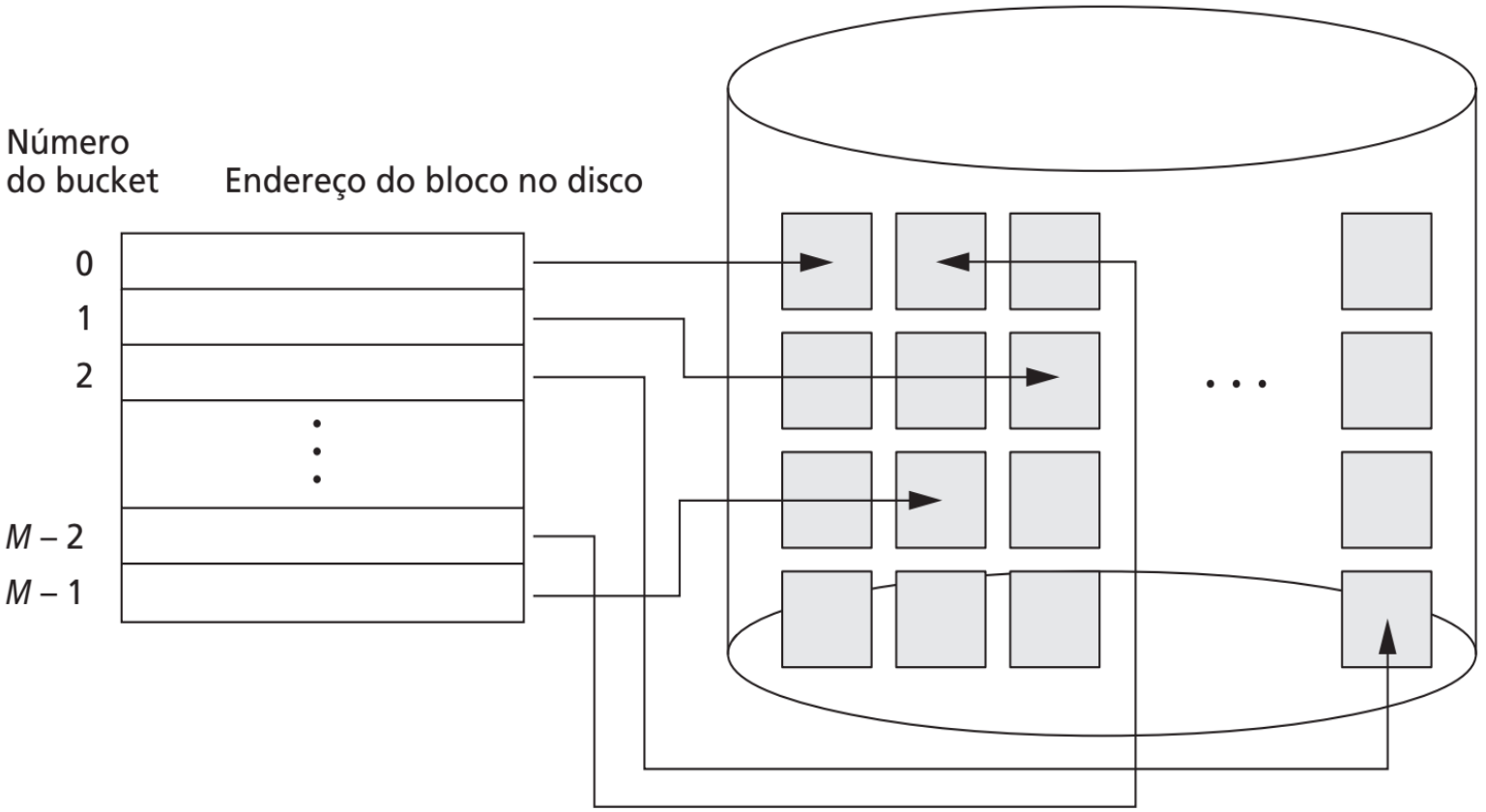
Bucket 7

Mianus	A-215	700

Bucket 8

Downtown	A-101	500
Downtown	A-110	600

Bucket 9


Número
do bucket
Endereço do bloco no disco
0
1
2
M – 2
M – 1

Arquivo Clustering/Multitabela/Misto

- 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.

select *nome_dept, prédio, orçamento, ID, nome, salário*

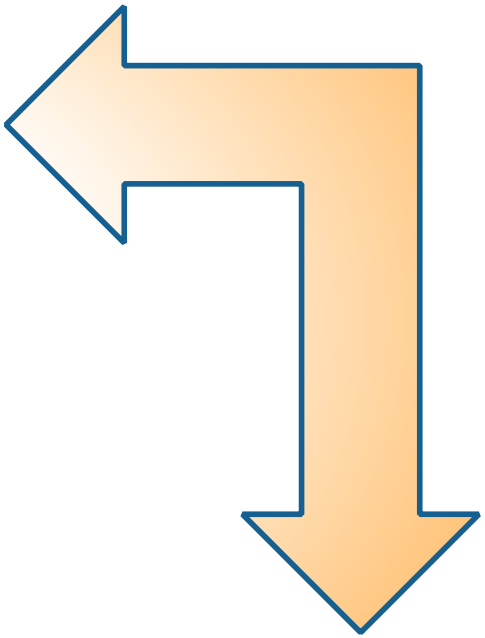
from *departamento* **natural join** *instrutor*;

nome_dept	*prédio*	*orçamento*
Comp. Sci.	Taylor	100000
Physics	Watson	70000

Figura 13.9 A relação *departamento*.

ID	*nome*	*nome_dept*	*salário*
10101	Srinivasan	Comp. Sci.	65000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000

Figura 13.10 A relação *instrutor*.



Comp. Sci.	Taylor	100000	
10101	Srinivasan	Comp. Sci.	65000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000
Physics	Watson	70000	
33456	Gold	Physics	87000

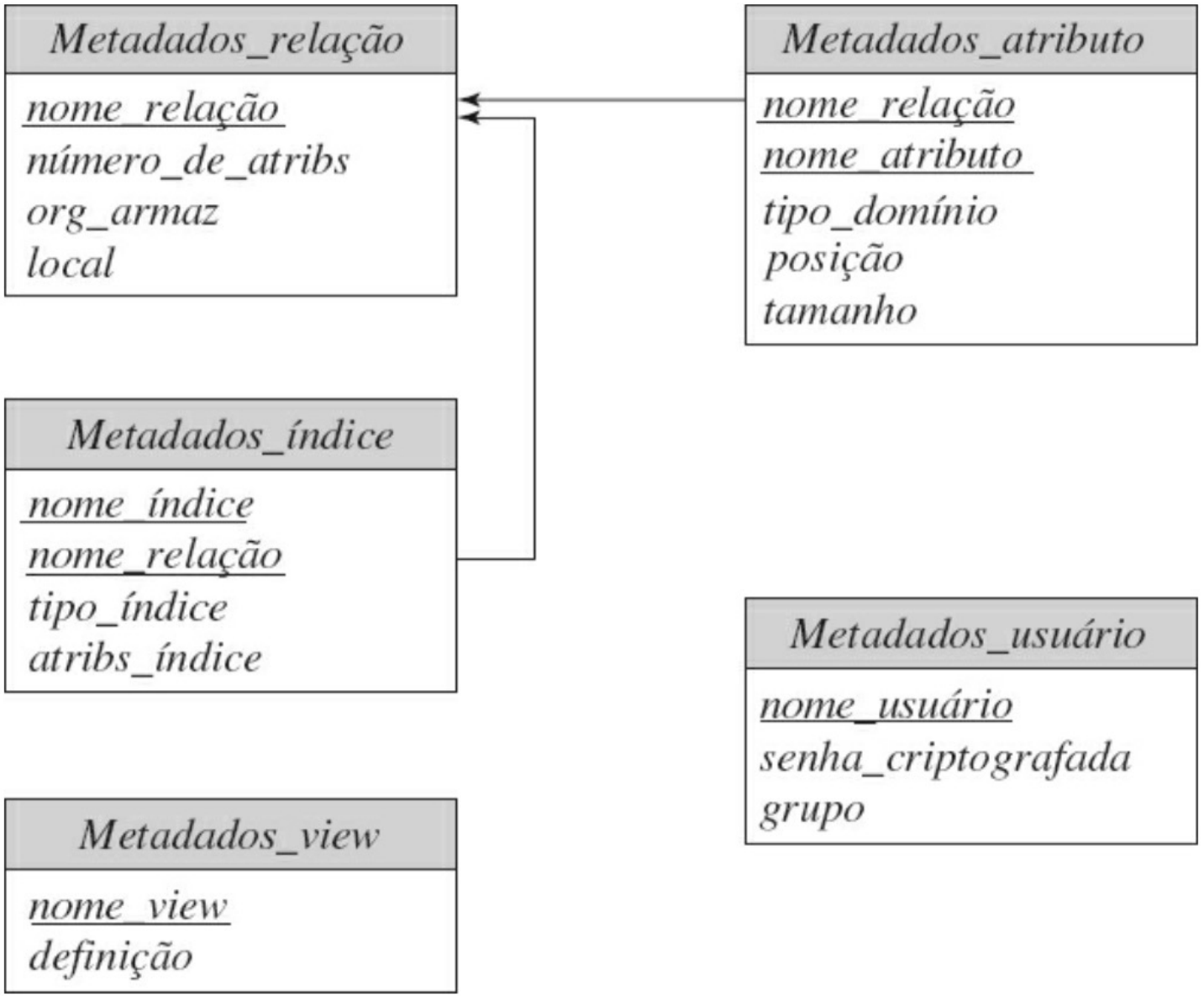
Figura 13.11 Estrutura de arquivos com agrupamento de múltiplas tabelas.

CATÁLOGO 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)


Metadados_relação
nome_relação
número_de_atribs
org_armaz
local
Metadados_atributo
nome_relação
nome_atributo
tipo_domínio
posição
tamanho
Metadados_índice
nome_índice
nome_relação
tipo_índice
atribs_índice
Metadados_usuário
nome_usuário
senha_criptografada
grupo
Metadados_view
nome_view
definição