

FTC – Faculdade de Tecnologia e Ciências  
Sistemas de Informação  
V Semestre

## II Unidade Indexação

Rafael Neiva  
[rafael@prodados.inf.br](mailto:rafael@prodados.inf.br)



# Indexação



- Muitas consultas de BD fazem referência a apenas uma pequena proporção de registro de um arquivo ou tabela.
- Exemplo: “Encontre todas as contas na agência Downtown”
- O que SGBD faz quando uma consulta destas é realizada?



# Conceitos Básicos



Perryridge	102	Hayes	400
Round Hill	305	Turner	350
Mianus	215	Smith	700
Downtown	101	Johnson	500
Redwood	222	Lindsay	700
Perryridge	201	Williams	900
Brighton	217	Green	750
Downtown	110	Peterson	600
Perryridge	218	Lyle	700
...			

# Conceitos Básicos



- O sistema deve ser capaz de localizar os registros dessa consulta, sem ter que ler cada registro e verificar se o campo nome\_agência contém “Downtown”
- Para isso são usadas estruturas adicionais:
  - Índices – oferecem um caminho “mais curto” para chegar até um determinado registro no BD



# Conceitos Básicos

- Há dois tipos básicos de índices
  - Índices Ordenados – Baseiam-se na ordenação dos valores
  - Índices Hash – Baseiam-se na distribuição uniforme dos valores por meio de uma faixa de **buckets**.
    - O **bucket** ao qual um valor é atribuído é determinado por uma função, chamada de **função hash**
- Existem diversas técnicas que implementam os dois tipos básicos.
- Cada uma possui vantagens e desvantagens, dependendo do tipo de aplicação.



# Conceitos Básicos

- Cada técnica precisa ser avaliada tendo por base:
  - Tipos de Acesso – Quais tipos de acesso são aceitos de forma eficiente. Os tipos de acesso podem incluir a localização de registros com um valor de atributo especificado e a localização de registros cujos valores de atributos se encontrem em um intervalo específico.
    - Encontrar um registro pontual
    - Encontrar um conjunto de registros em uma faixa





# Conceitos Básicos



- Tempo de acesso : tempo gasto para encontrar um item ou conjunto de itens usando a técnica em questão
- Tempo de inserção : Tempo para encontrar o local atual para inserir o novo item e atualizar a estrutura de índice.
- Tempo de remoção : Tempo para encontrar o item a ser excluído e atualizar a estrutura de índice.
- Espaço adicional : espaço extra ocupado pela estrutura de índice

# Conceitos Básicos



- É interessante ter mais de um índice para um arquivo ou tabela.
- Por ex: uma biblioteca mantém diversos catálogos: por autor, por título, por assunto, etc.
- O atributo ou conjunto de atributos utilizado para procurar registros em uma tabela é chamado de **chave de procura**
- **Atenção:** chave de procura não é *chave primária, candidata e superchave*.





# Índices Ordenados



- Para permitir um rápido acesso aleatório aos registro num arquivo, uma estrutura de índice é usada.
- Em um arquivo pode ser incluído diversos índices em diferentes chaves de busca.
- O índice cuja chave de busca especifica a ordem seqüencial do arquivo é o *índice primário*. Os outros índices são chamados de *índices secundários*.

# Indexação

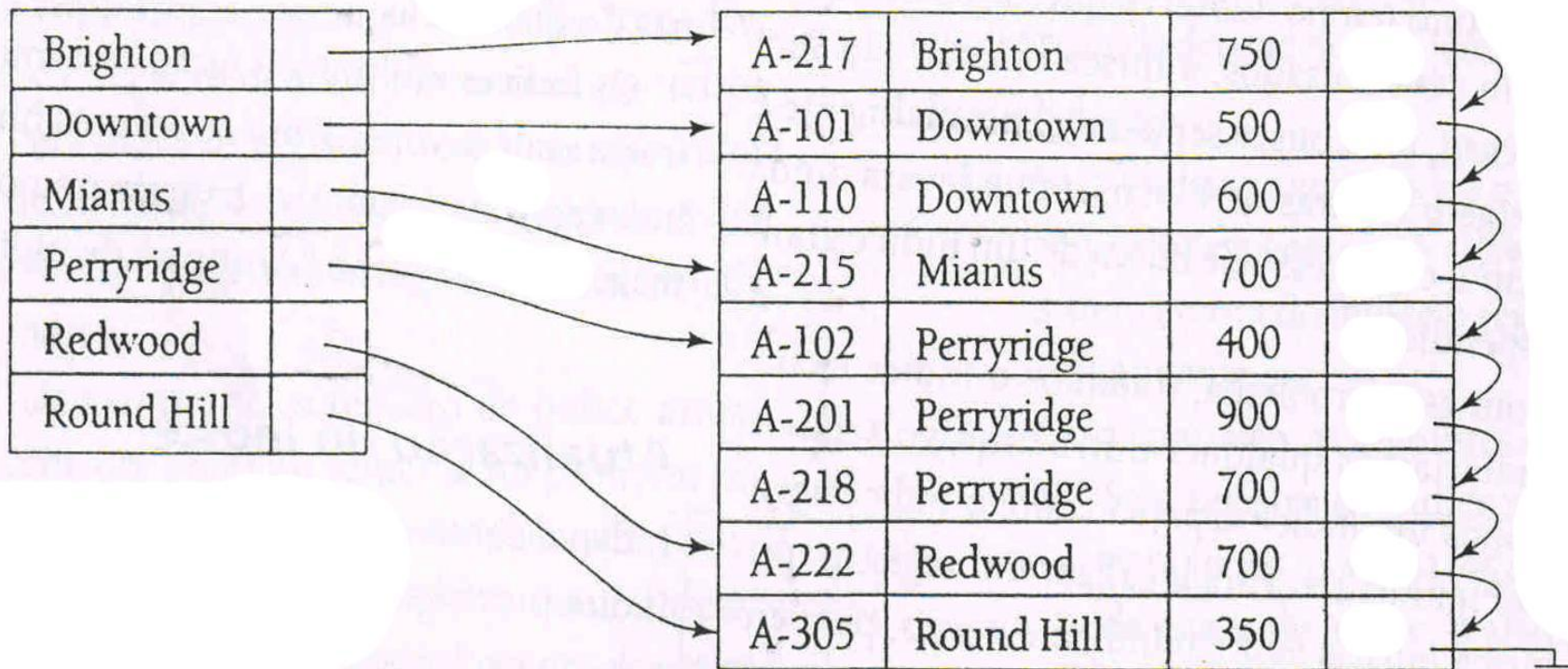


- Dois tipos de índices podem ser usados:
  - **Índice denso** - um registro de índice aparece para cada valor da chave de busca no arquivo.
  - **Índice esparsos** - registros de índices são criados somente para alguns registros.

# Índice denso



Cada registro do índice contém o valor da chave de busca e um ponteiro para o registro.

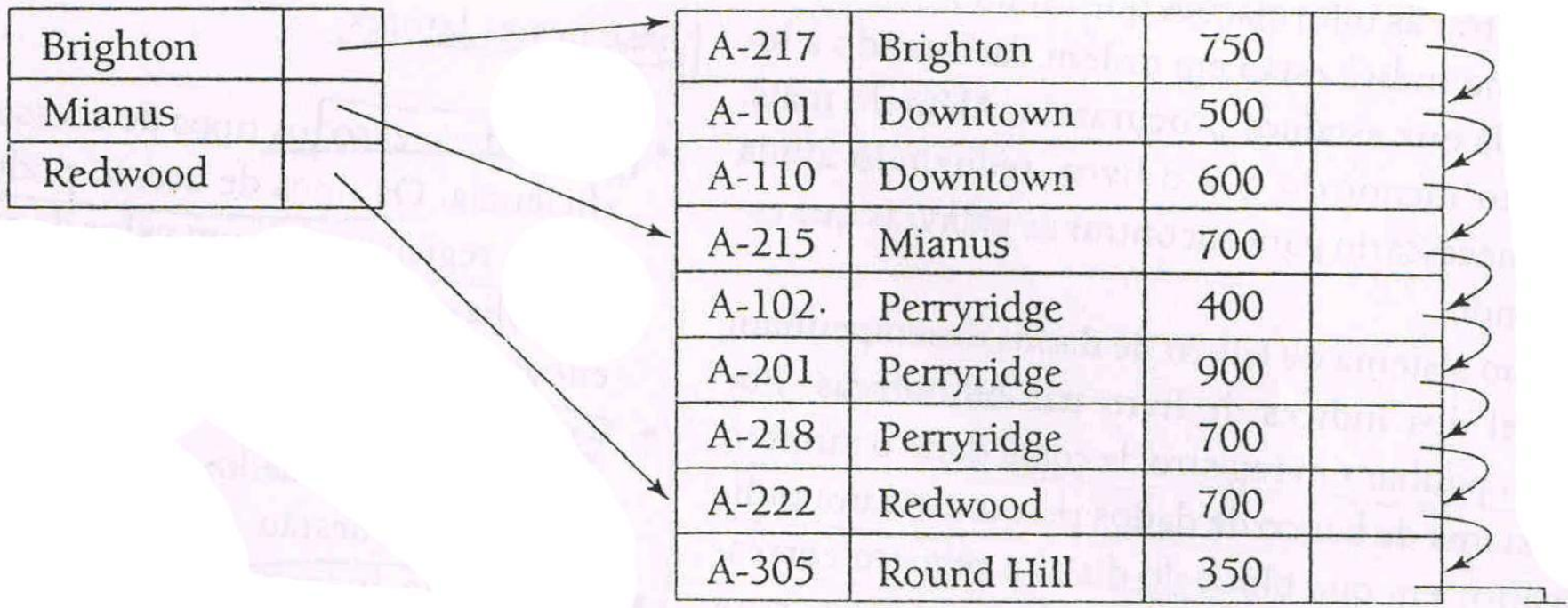




# Índice esparsos



registros de índices são criados somente para alguns registros.



# Índices

- Suponha que estejamos procurando os registros da agência **Perryridge**.
- Busca em índice denso:
  - 1 - seguimos o ponteiro diretamente até o primeiro registro da agência **Perryridge**.
  - 2 - Processamos este registro e seguimos o ponteiro naquele registro para localizar o próximo registro na ordem da chave de busca.
  - 3 - repetimos o passo 2 até encontrar um registro com nome de agência diferente de Perryridge.



# Índices



- Busca em índice esparsos:

- Não encontramos uma entrada com Perryridge, então seguimos a última entrada em ordem alfabética antes de Perryridge que é Mianus.
- Lemos o arquivo de depósito na ordem seqüencial até encontrar o primeiro registro Perryridge, e iniciamos o processamento naquele pontos.





# Índices



- Vantagem do índice denso:
  - mais rápido localizar um registro com índice denso do que com índice esparsos.
- Vantagem do índice esparsos:
  - índices esparsos requerem menos espaço e impõem menos sobrecarga de manutenção para inserção e remoção.

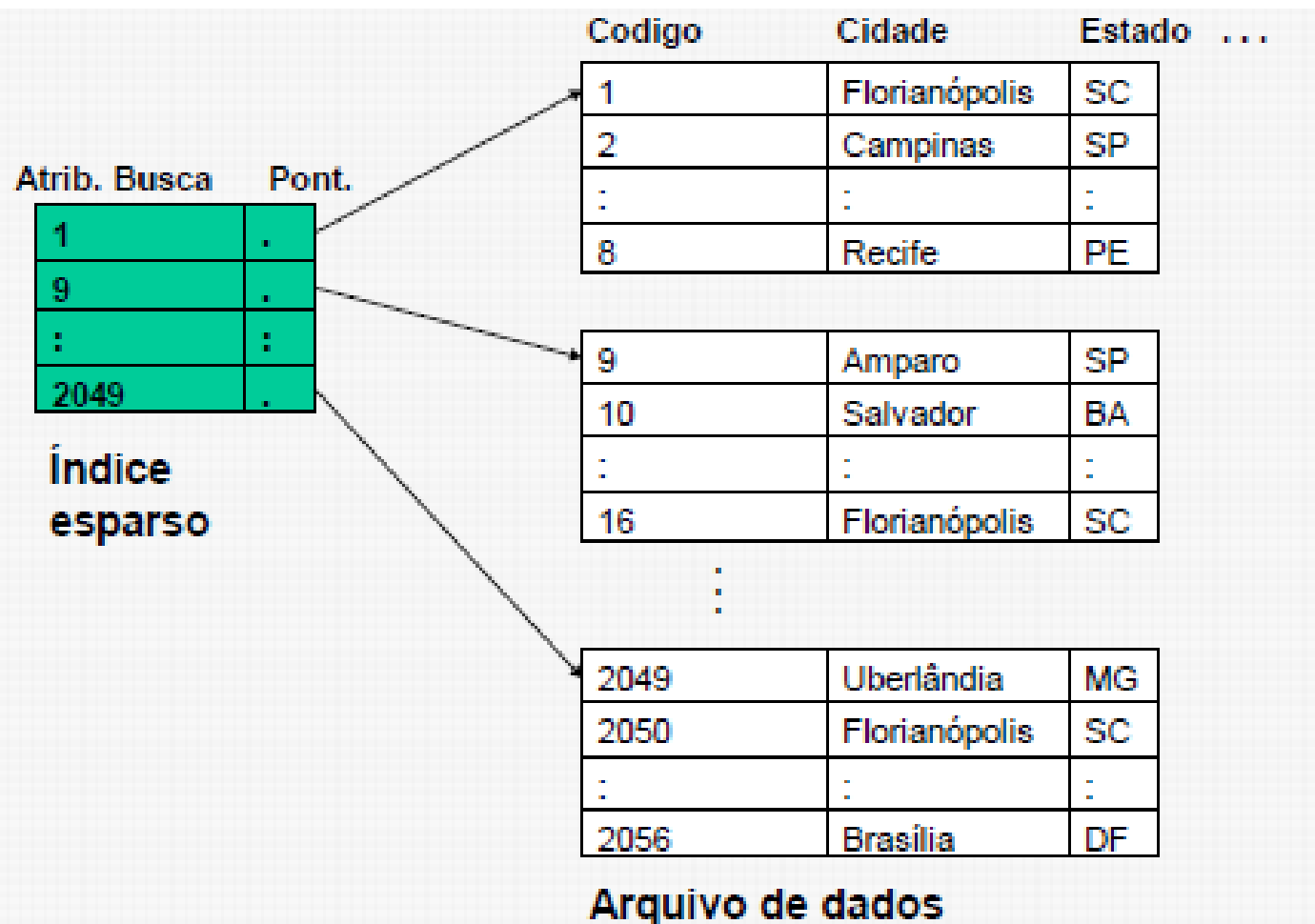


# Índice Primário

- **ordenação, inclusive dos dados, pela chave primária**
- Esparso
- Máxima eficiência



# Índice Primário



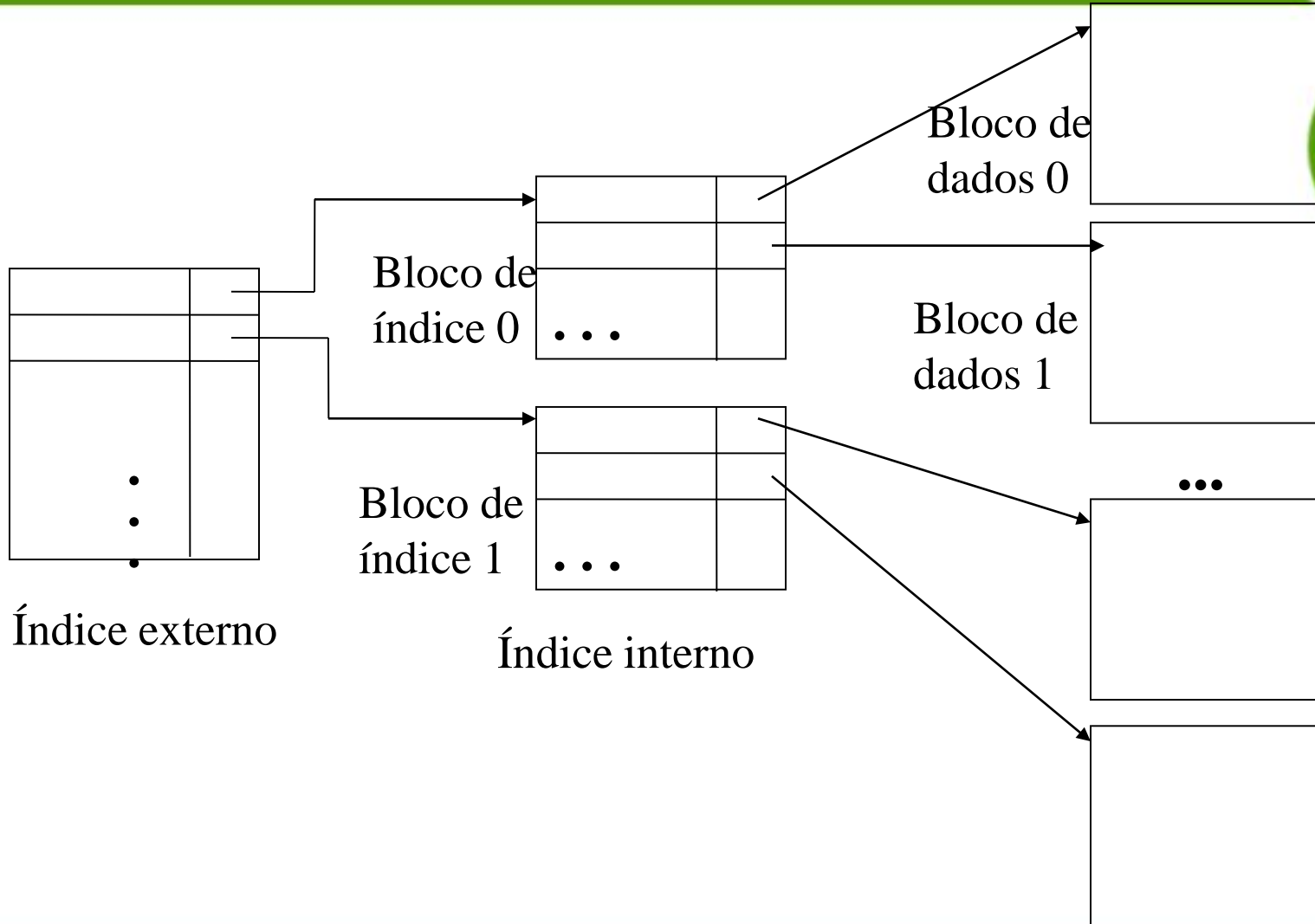
# Índices de Níveis Múltiplos



- Temos um problema quando registros para um dado valor da chave de busca ocupam vários blocos e o índice fique muito grande para o processamento eficiente.
- Para resolver, tratamos o índice como trataríamos qualquer arquivo seqüencial e construimos um índice esparsos sobre o índice primário.



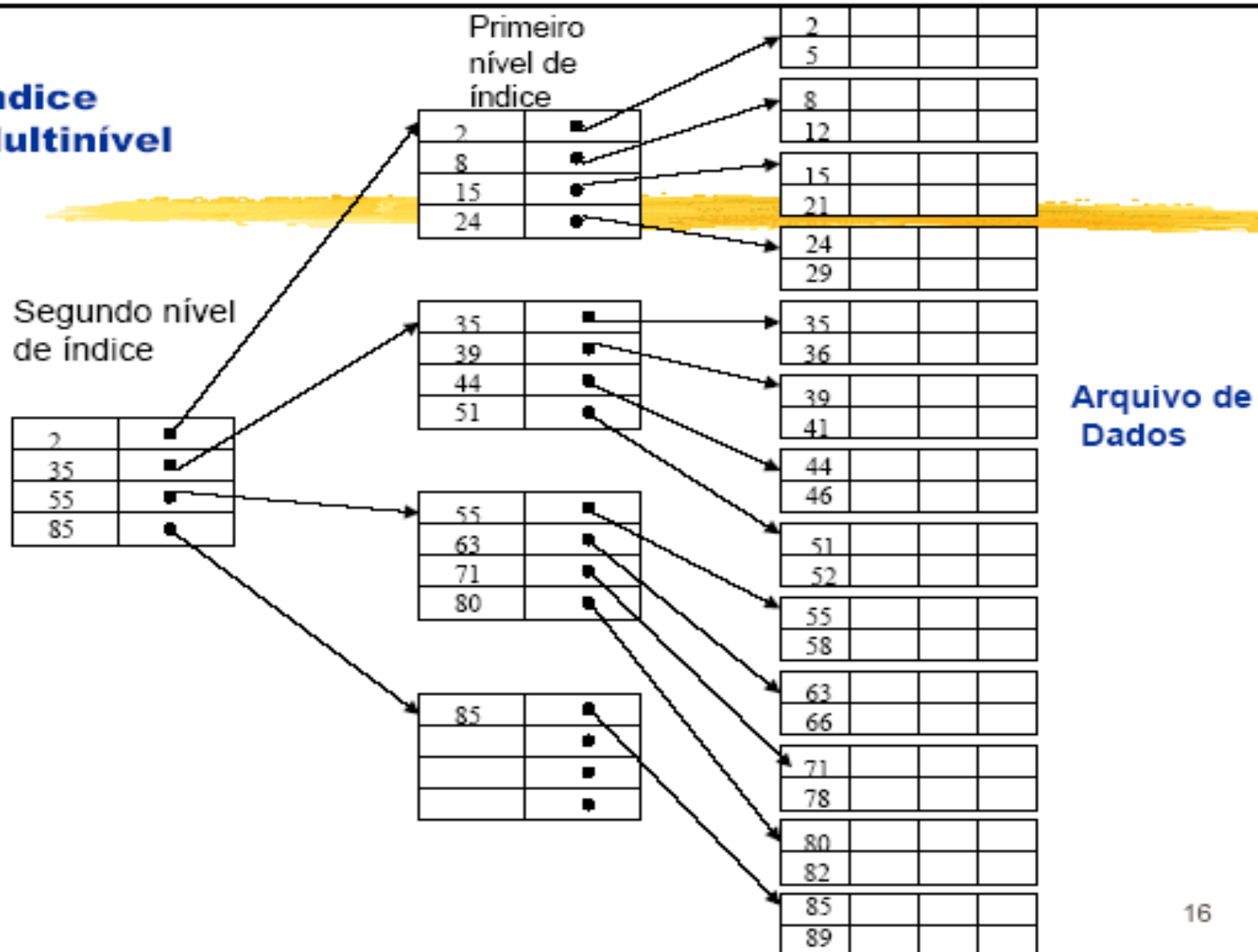
# Índice de Níveis Múltiplos



# Índice de Níveis Múltiplos



## Índice Multinível





# Índice de Múltiplos Níveis



## ■ Para localizar um registro:

- ☐ Primeiro usamos uma busca binária no índice externo para encontrar o registro para o maior valor da chave de busca menor ou igual ao que desejamos
- ☐ O ponteiro aponta para um bloco de índice interno. Varremos esse bloco até encontrar o registro que tem a maior chave de busca menor ou igual à que desejamos.
- ☐ O ponteiro neste registro indica o bloco do arquivo que contém o registro que estamos procurando.



# Índice Secundário

- Se o índice for construído sobre quaisquer outros campos que não os de ordenação do arquivo, então o índice é chamado de **Índice Secundário**.



# Índice Secundário



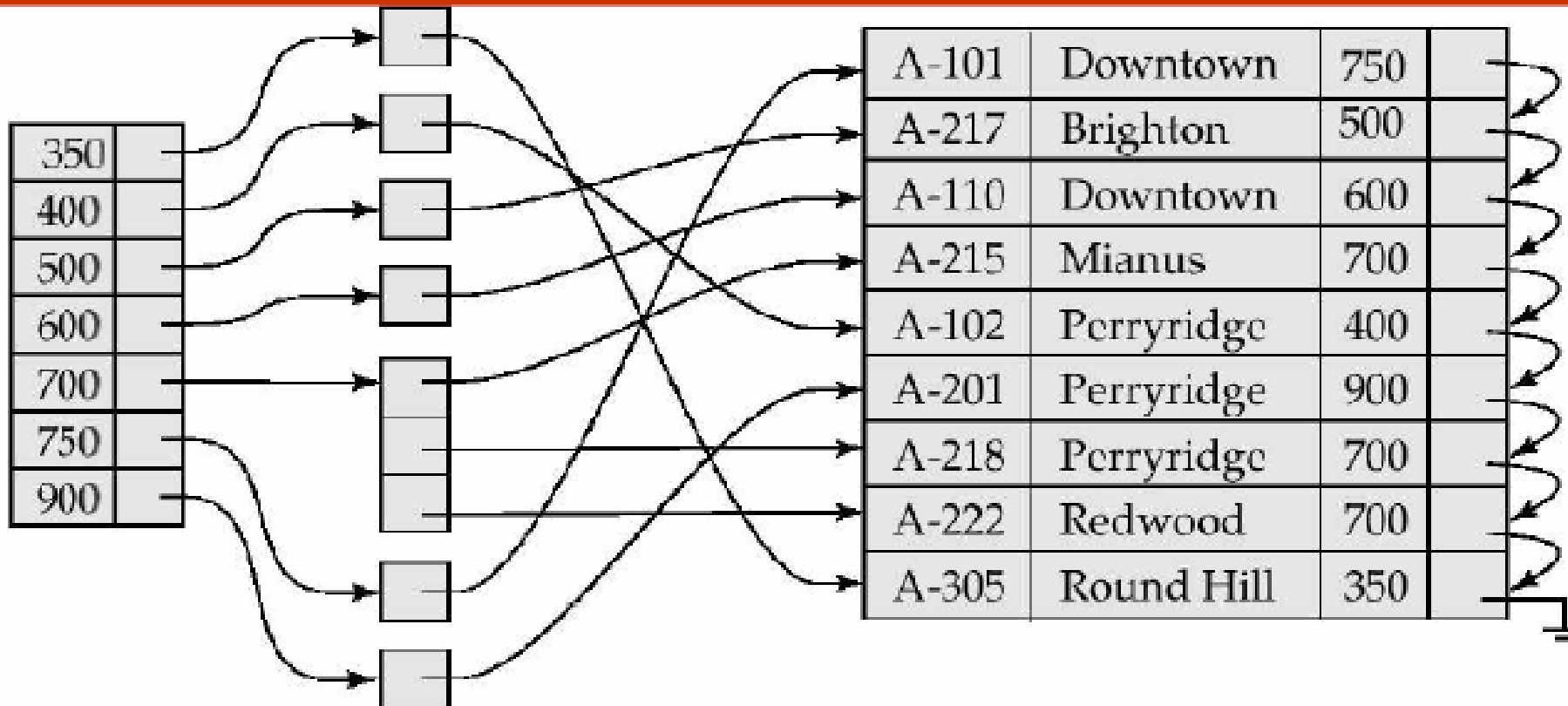
- Esse tipo de índice deve ser estruturado de forma diferente dos índices primários.
- Eles devem usar um nível extra de indireção.
- Os ponteiros de um índice secundário não apontam diretamente para o arquivo, eles apontam para um *bucket* que contém ponteiros para o arquivo.



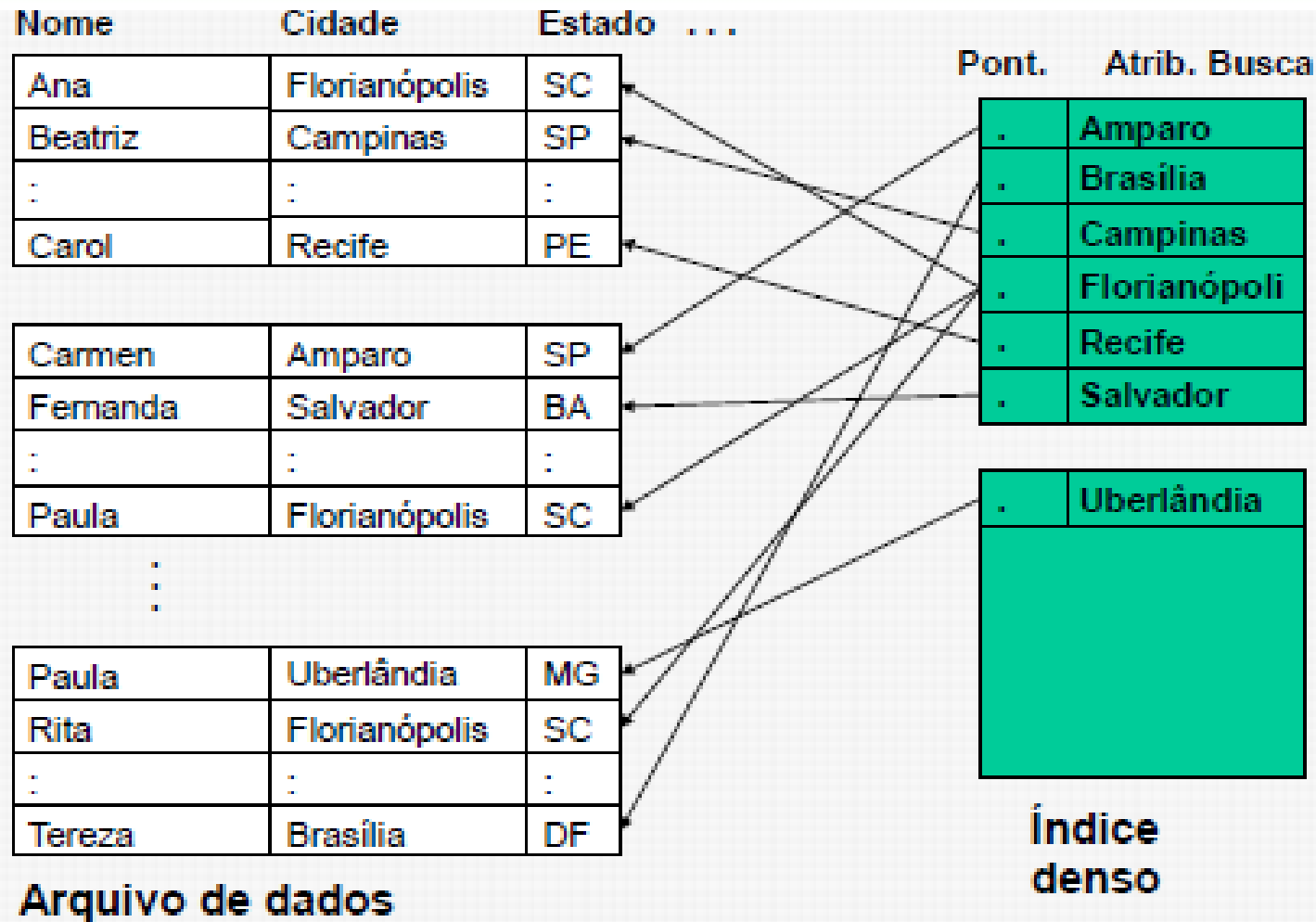
# Índice Secundário



Buckets



# Índice Secundário



# Índice Secundário



- Melhoram o desempenho das consultas que usam chave de busca diferente da chave de busca do índice agrupado
- Impõem sobrecarga significativa na modificação do banco de dados



# Inserção em índices densos



1. Se o valor da chave de busca não aparece no índice, o sistema insere um registro de índice com o valor de chave de busca no índice, na posição apropriada.



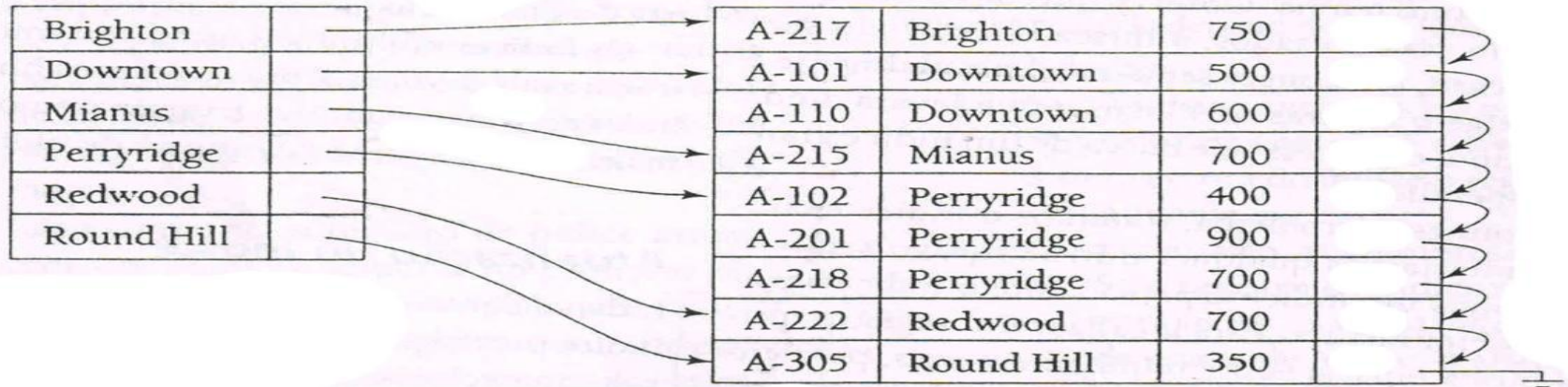
2. Caso contrário:

a) Se o registro de índice armazena ponteiros para todos os registros com o mesmo valor de chave de busca, o sistema acrescenta um ponteiro para o novo registro no registro de índice.



b) Caso contrário, o registro de índice armazena um ponteiro somente no primeiro registro com o valor da chave de busca. O sistema, então, coloca o registro sendo inserido após os outros registros com os mesmos valores de chave de busca.

# Inserção em índices densos



Inserir a agência de nome Village.



# Inserção em índices esparsos

1. Se o sistema cria um novo bloco, ele insere no índice o primeiro valor de chave de busca (na ordem de chave de busca) que aparece no novo bloco.
2. Se o novo registro tiver o menor valor de chave de busca em seu bloco, o sistema atualiza a entrada de índice apontando para o bloco.
3. Caso contrário, o sistema não faz qualquer mudança no índice.





# Inserção em índices esparsos



Brighton		A-217	Brighton	750	
Mianus		A-101	Downtown	500	
Redwood		A-110	Downtown	600	
		A-215	Mianus	700	
		A-102	Perryridge	400	
		A-201	Perryridge	900	
		A-218	Perryridge	700	
		A-222	Redwood	700	
		A-305	Round Hill	350	

Inserir no índice a agência de nome Boston

<b>Boston</b>		<b>A-107</b>	<b>Boston</b>	<b>550</b>	
Mianus		A-217	Brighton	750	
Redwood		A-101	Downtown	500	
		A-110	Downtown	600	
		A-215	Mianus	700	
		A-102	Perryridge	400	
		A-201	Perryridge	900	
		A-218	Perryridge	700	
		A-222	Redwood	700	
		A-305	Round Hill	350	

# Exclusão em índices densos

- 1. Se o registro excluído foi o único registro com seu valor específico de chave de busca, então o sistema retira do índice o registro de índice correspondente.
- 2. Senão:
  - a) Se o registro de índice armazena ponteiros para todos os registros com o mesmo valor de chave de índice, o sistema exclui do registro de índice o ponteiro para o registro excluído.
  - b) Caso contrário, o registro de índice armazena um ponteiro somente para o primeiro registro com o valor da chave de busca. Se o registro excluído foi o primeiro registro com o valor da chave de busca, o sistema atualiza o registro de índice para que aponte para o próximo registro.



# Exclusão em índices densos



Brighton			A-217	Brighton	750	
Downtown			A-101	Downtown	500	
Mianus			A-110	Downtown	600	
Perryridge			A-215	Mianus	700	
Redwood			A-102	Perryridge	400	
Round Hill			A-201	Perryridge	900	
			A-218	Perryridge	700	
			A-222	Redwood	700	
			A-305	Round Hill	350	

Registro referente a conta A-102 foi excluído

Brighton			A-217	Brighton	750	
Downtown			A-101	Downtown	500	
Mianus			A-110	Downtown	600	
Perryridge			A-215	Mianus	700	
Redwood			A-201	Perryridge	900	
Round Hill			A-218	Perryridge	700	
			A-222	Redwood	700	
			A-305	Round Hill	350	



# Exclusão em índices esparsos



1. Se o índice não tiver um registro de índice com o valor de chave de busca do registro excluído, nada precisa ser feito com o índice.

2. Se não:

a) Se o registro excluído foi o único com sua chave de busca, o sistema substitui o registro de índice correspondente por um registro de índice para o próximo valor da chave de busca. Se o próximo valor de chave de busca já tiver uma entrada de índice, a entrada é excluída, em vez de ser substituída.

b) Caso contrário, se o registro de índice para o valor da chave de busca apontar para o registro sendo excluído, o sistema atualiza o registro de índice para que aponte para o próximo registro com o mesmo valor da chave de busca.

# Exclusão em índices esparsos



Brighton		A-217	Brighton	750	
Mianus		A-101	Downtown	500	
Redwood		A-110	Downtown	600	
		A-215	Mianus	700	
		A-102	Perryridge	400	
		A-201	Perryridge	900	
		A-218	Perryridge	700	
		A-222	Redwood	700	
		A-305	Round Hill	350	

Registro referente a conta A-217 foi excluído

<b>Downtown</b>		A-101	Downtown	500	
Mianus		A-110	Downtown	600	
Redwood		A-215	Mianus	700	
		A-102	Perryridge	400	
		A-201	Perryridge	900	
		A-218	Perryridge	700	
		A-222	Redwood	700	
		A-305	Round Hill	350	

# Índices em SQL



- Sintaxe:
- `create [unique] index <nome_indice> on`
- `<nome_relação> (<lista atributos>)`
- Exemplo:
- `create index meu_indice on`  
`agencia(nome_agencia)`
- O atributo unique exige que a chave de procura seja uma chave candidata (não duplicada).
- Para remover um índice :
- `drop index <nome_índice>`