

# AULA 2- ÍNDICES

PROFA. DRA. LEILA BERGAMASCO

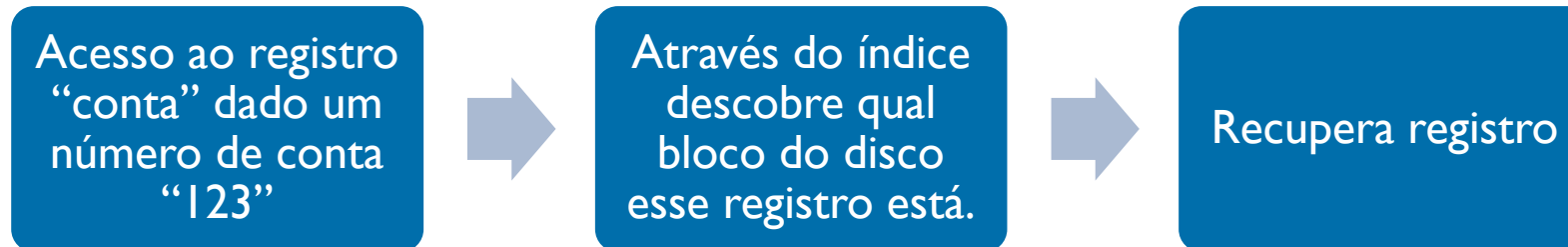
CC6240 – Tópicos Avançados em Banco de Dados

# AGENDA

- Índices

# CONCEITOS PRELIMINARES

- Consulta = recuperação de uma pequena fração de registros
  - Ler um por um não é uma opção eficiente.
  - Como acessar diretamente esses dados?
    - Criação de estruturas especiais para associar tais registros
    - Índices!



- Índices ordenados (sequenciais): baseados em uma ordem de classificação de valores
- Índices hash: distribuem os índices uniformemente em buckets. Os buckets na qual um índice será associado é determinado pela função hash.

# CONCEITOS PRELIMINARES

- Várias técnicas de índice. Como escolher?
  - Tipos de acesso
    - Como é feito o acesso ao dado. A localização de cada atributo pode possuir um valor específico ou um intervalo.
  - Tempo de acesso
    - Tempo gasto para encontrar o registo ou um conjunto de registos
  - Tempo de inserção
    - Tempo gasto para inserir um novo item.
  - Tempo de exclusão
    - Tempo gasto para excluir um item.
  - Espaço adicional
    - Espaço gasto pela estrutura do índice.
- Chaves de busca
  - Um ou mais índices que serão utilizados durante a busca.

Diferente de  
chave primária!


# ÍNDICES ORDENADOS

# ÍNDICES ORDENADOS

- Estrutura de índice: chave de busca → registro de índice
- Índices primários ou índices de agrupamento
  - Quando os registros estão ordenados sequencialmente assim como a chave de busca.
  - Muitas vezes é a chave primária do registro, porém NÃO é regra
  - Técnicas:
    - Índices densos
    - Índices esparsos
- Índice secundário ou índice de não agrupamento
  - Ordem da chave de busca é diferente da ordem dos registros

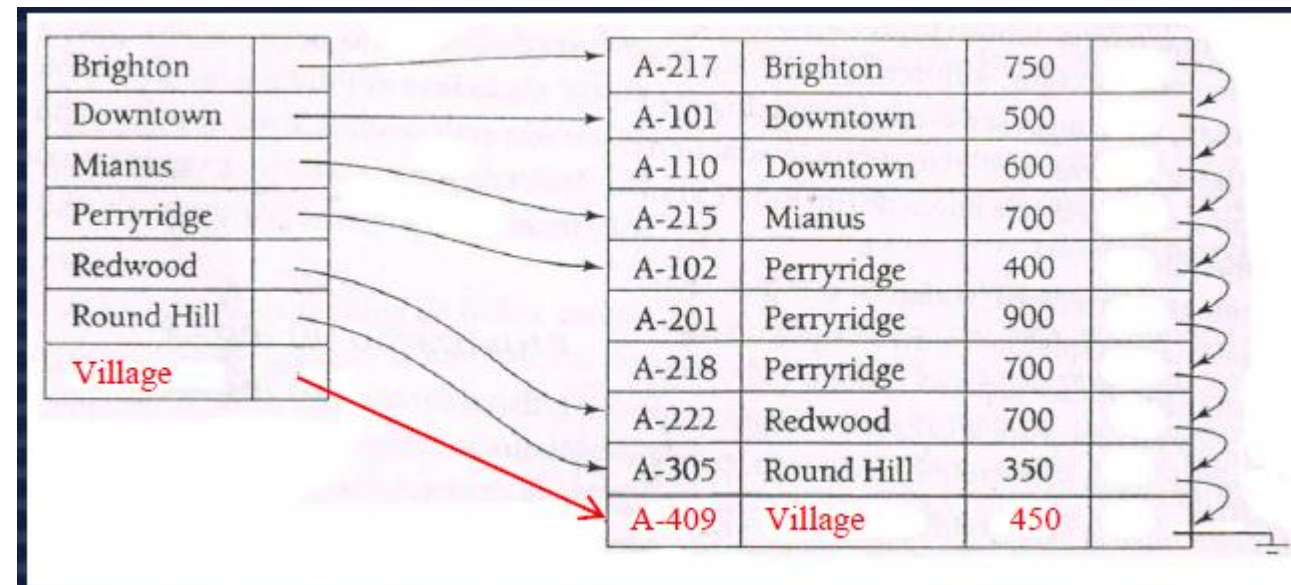
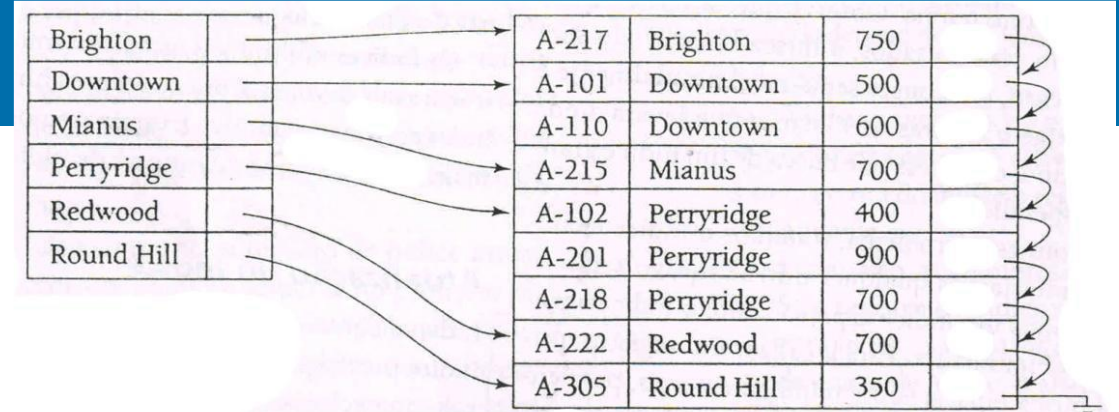
# ÍNDICES DENSOS

- Temos um registro de índice para cada valor de chave de busca

Chave de busca = nome da agência	Reg_ind	numero_conta	nome_agência	saldo	
Bairro Assunção		A-217	Bairro Assunção	750	
Bairro dos Casa		A-101	Bairro dos Casa	500	
Centro		A-110	Bairro dos Casa	600	
Nova Petrópolis		A-215	Centro	700	
Rudge Ramos		A-102	Nova Petrópolis	400	
Vila Duzzi		A-201	Nova Petrópolis	900	
		A-218	Nova Petrópolis	700	
		A-222	Rudge Ramos	700	
		A-305	Vila Duzzi	350	

# INSERÇÃO DENSO

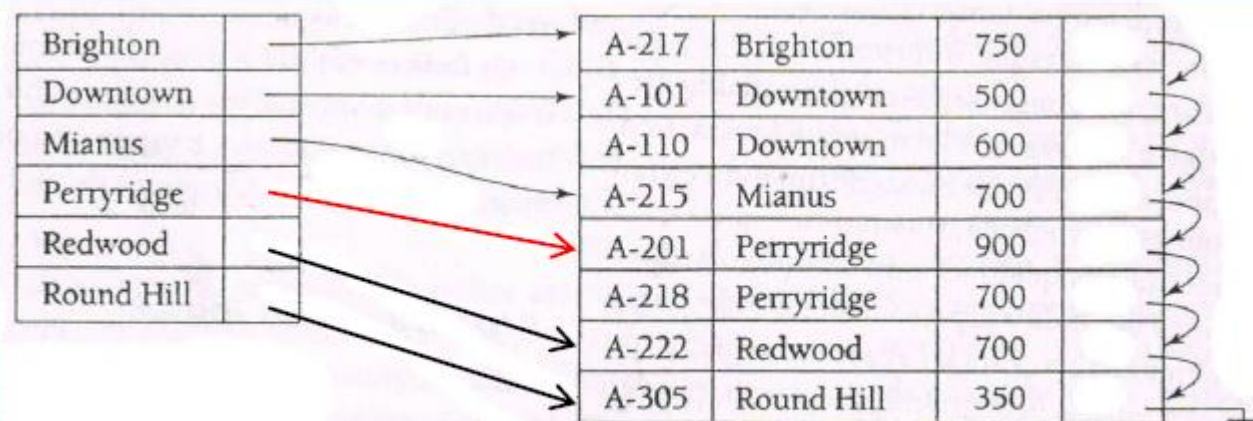
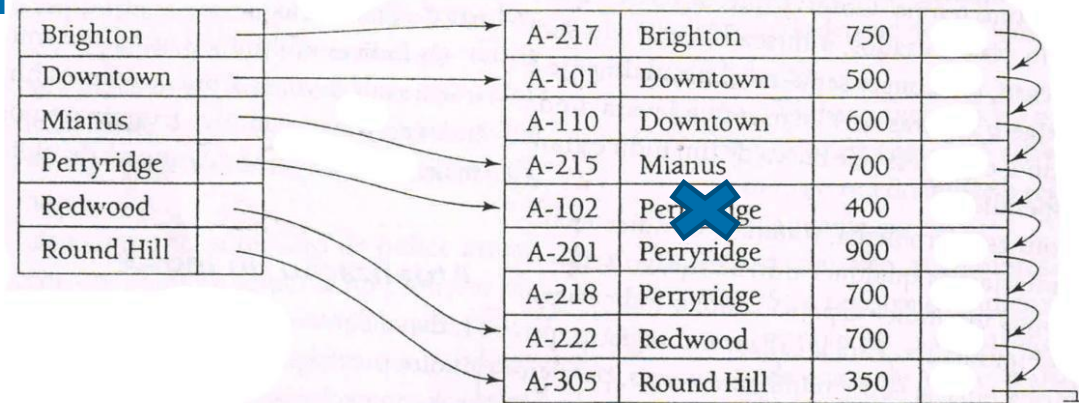
- Se o valor da chave de busca não aparece no índice, o sistema insere um registro de índice com o valor de chave de busca no índice, na posição apropriada.
- Caso contrário:
  - Se o registro de índice armazena ponteiros para todos os registros com o mesmo valor de chave de busca, o sistema acrescenta um ponteiro para o novo registro no registro de índice.
  - 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.






# EXCLUSÃO DENSO

- 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.
- Se não:
  - 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.
  - 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.



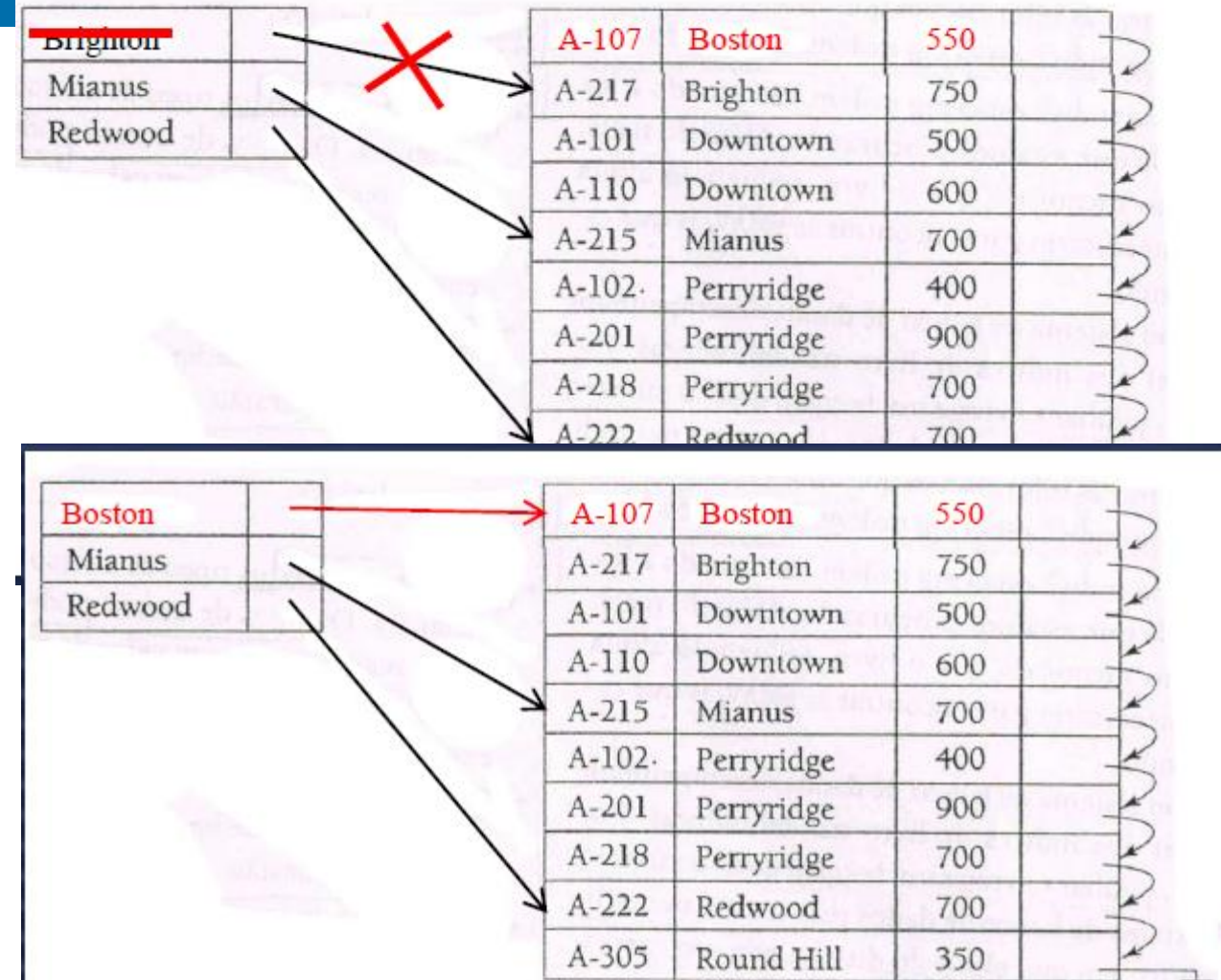
# ÍNDICES ESPARSOS

- Temos um registro de índice somente para alguns valores de chaves de busca

Chave de busca = nome da agência	Reg_ind	numero_conta	nome_agência	saldo	
Bairro Assunção		A-217	Bairro Assunção	750	
		A-101	Bairro dos Casa	500	
		A-110	Bairro dos Casa	600	
		A-215	Centro	700	
Centro		A-102	Nova Petrópolis	400	
		A-201	Nova Petrópolis	900	
		A-218	Nova Petrópolis	700	
Rudge Ramos		A-222	Rudge Ramos	700	
		A-305	Vila Duzzi	350	

# INSERÇÃO ESPARSO

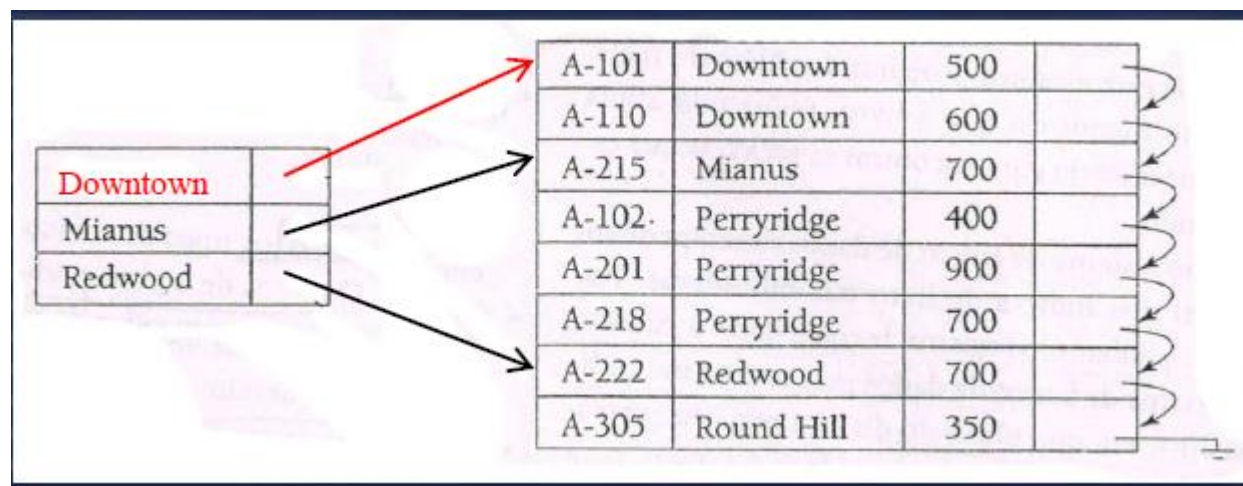
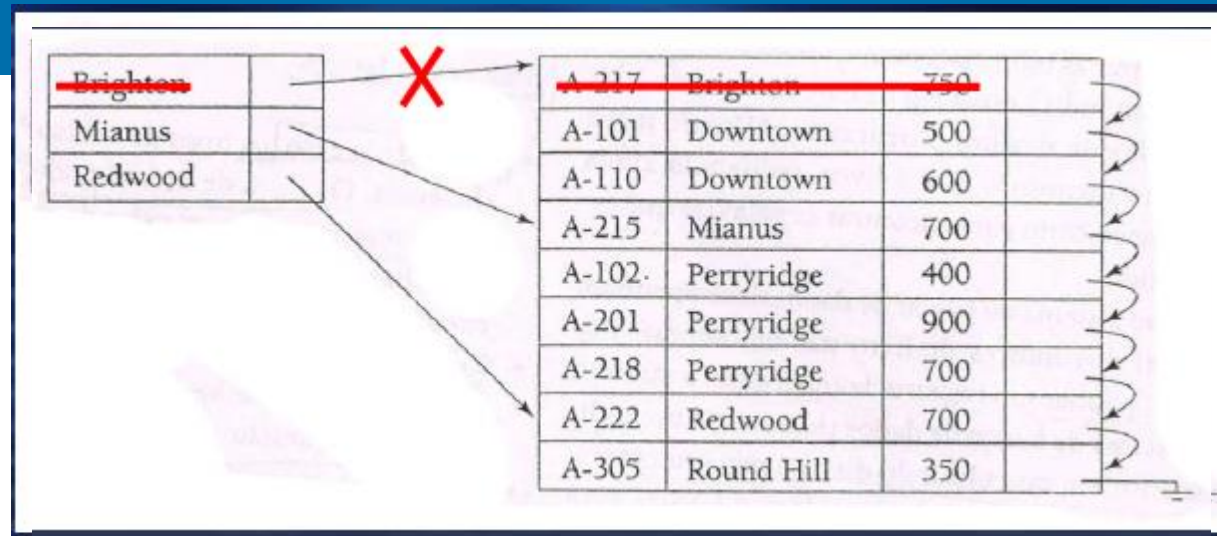
- Se o sistema cria um novo bloco, ele insere no índice o primeiro valor de chave de busca (na ordem de chave de busca) que aparece no novo bloco.
- Se não, se o novo registro tiver o menor valor de chave de busca em seu bloco, o sistema atualiza a entrada de índice apontando para o bloco.
- Caso contrário,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





# EXCLUSÃO ESPARSO

- 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.
- Se não:
  - 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.
  - 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.

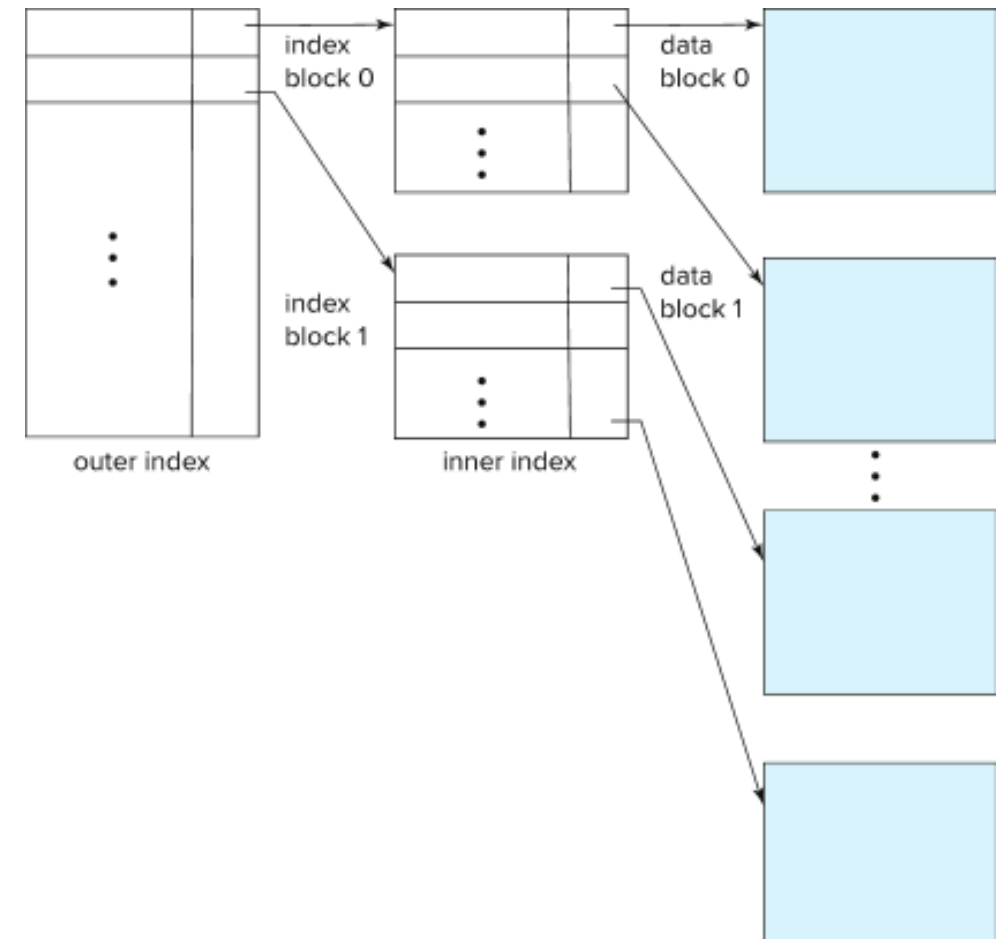


# ÍNDICES DENSO X ESPARSOS

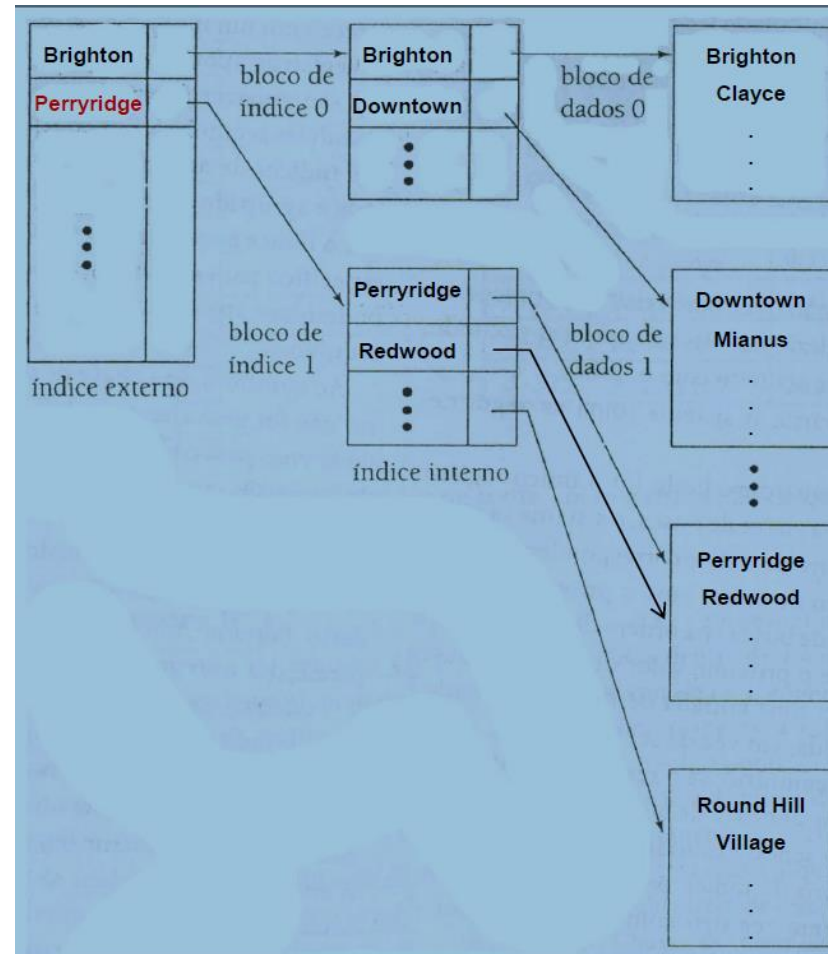
- Teoricamente é mais fácil localizar um registro via índice denso, porém é muito custoso manter a estrutura de índice atualizada e fazer a sua leitura.
- Solução atual mais utilizada:
  - Índice esperso com uma entrada por índice por bloco
    - Custo de acesso é alto para acessar o bloco → Uma vez que temos o bloco correto sua leitura é extremamente rápida.
    - Logo: Melhora tempo de acesso e espaço adicional reduzido

# ÍNDICES ESPARSOS MULTINÍVEIS

- A solução de índices esparsos funciona bem enquanto o índice couber na memória principal. Porém ele se torna custoso para grandes quantidades de dados.
- Supondo que temos um Banco de dados com 100 blocos e utilizamos busca binária para buscar no índice esparsos. Isso exigiria 7 leituras ao bloco, se cada leitura demora 30 milissegundos, temos que, para encontrar 1 registro a busca demoraria 210 milissegundos. O que é muito!
- Solução índices esparsos multiníveis



# ÍNDICES ESPARSOS MULTINIVEIS



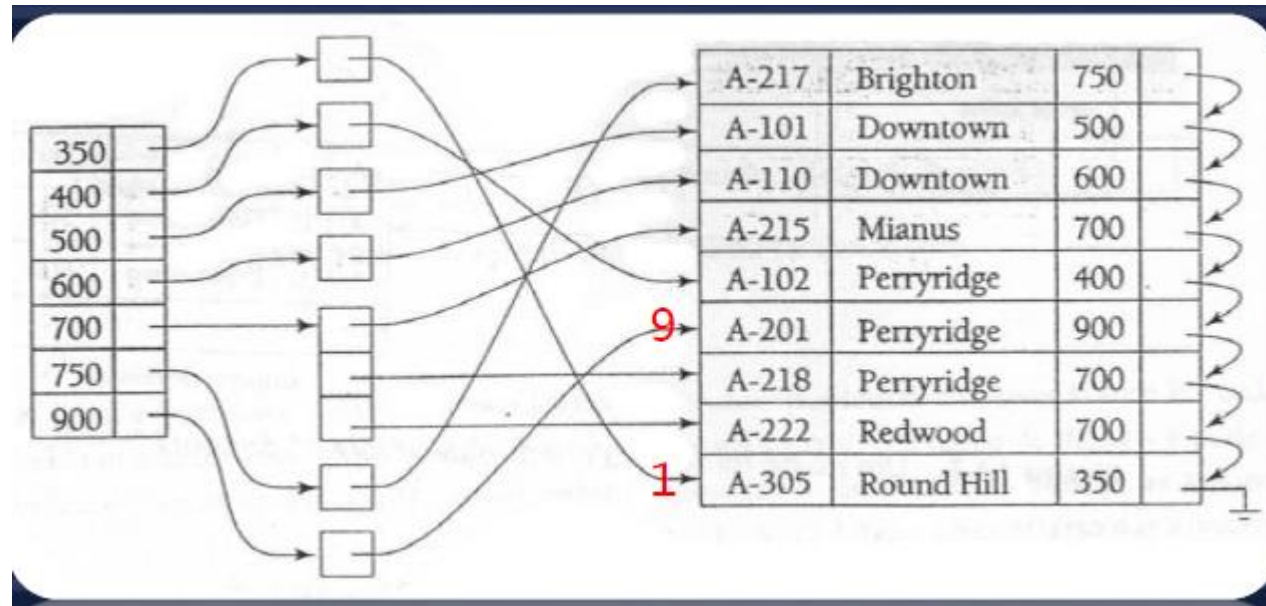
# ÍNDICES SECUNDÁRIOS



# ÍNDICES SECUNDÁRIOS

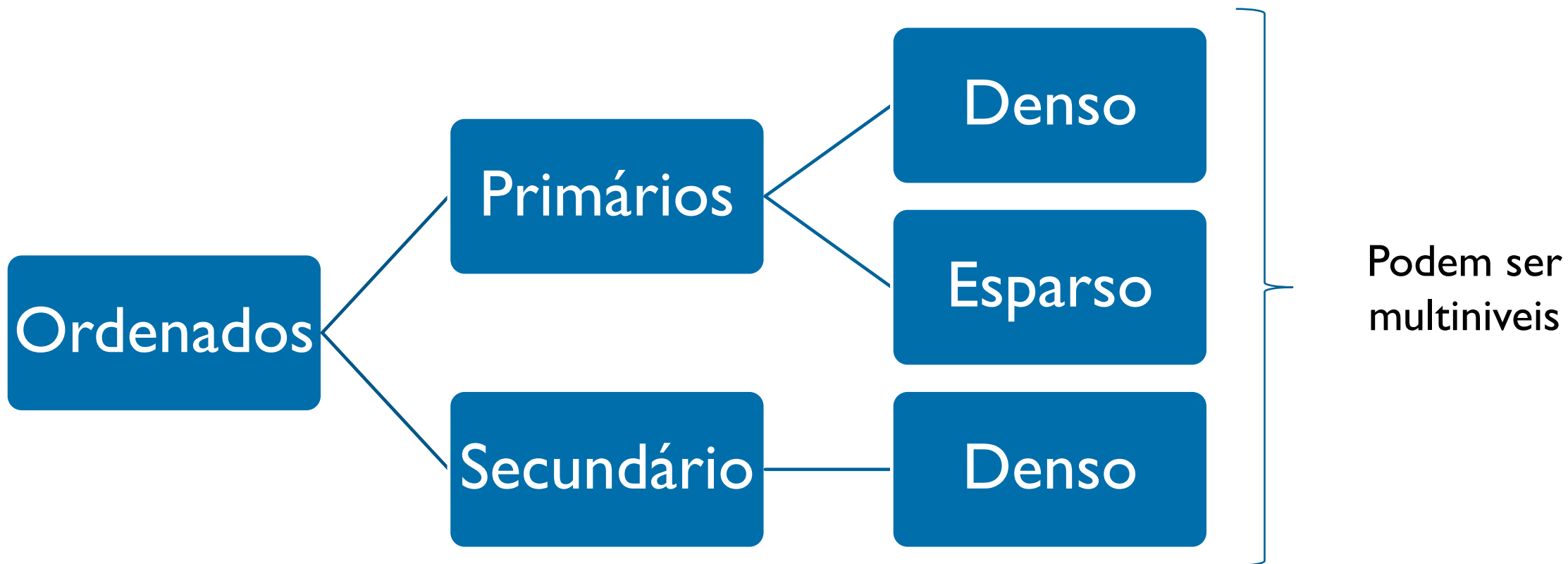
- Devem ser densos
- Ter ponteiro para cada registro
  - Bloco de registros são ordenados pela chave de busca do índice agrupado
  - Registros de índice secundário não são sequenciais
- Nível extra de indexação
  - Ponteiros não apontam diretamente para o arquivo
  - Ponteiros apontam para buckets

# ÍNDICES SECUNDÁRIOS



Melhora o desempenho já que para índices secundários não é necessário percorrer todo o arquivo. Entretanto, adiciona uma camada extra de indexação.

## RECAPITULANDO



## PROBLEMAS DE ÍNDICES ORDENADOS

- Performance degrada muito na medida que a estrutura de índice cresce.
- Solução
  - Outras estruturas de organização, que não sejam ordenadas.

OBRIGADO E ATÉ A PRÓXIMA AULA!

## BIBLIOGRAFIA

- ABRAHAM SILBERSCHATZ, HENRY F. KORTH, S. SUDARSHAN. Sistema de Banco de Dados. 6. Campus. 0. ISBN 9788535245356.
- ELMASRI, RAMEZ, SHAMKANT B. NAVATHE. Sistemas de banco de dados. Vol. 6. São Paulo: Pearson Addison Wesley, 2011.
- DATE, CHRISTHOPER J. Introdução a Sistemas de Bancos de Dados, 5ª. Edição. Campus, Rio de Janeiro (2004).

OBRIGADO E ATÉ A PRÓXIMA AULA!