**Índices**

* Acessos a registros baseados em campos
* Qualquer campo pode ser índice
* Mais de um campo pode ser índice. (Exemplo: R(A,B,C))
* Modo de melhorar o desempenho do BD
* A recuperação é mais rápida
* Produz trabalho adicional

**Tipos**

* Denso:

Uma entrada no arquivo de índice para cada registro

* Esparso:

Apenas alguns registros de dados são representados no arquivo de índice. (Exemplo: índice primário)

**Índice Primário ->**

Arquivo ordenado cujos registros são de tamanho fixo e têm 2 campos.

O primeiro é do mesmo tipo de dados do campo chave de classificação

o Segundo é um inteiro para um bloco do disco

* Há uma entrada no arquivo de índice para cada bloco no disco.
* O numero de entradas no indice é igual ao numero de blocos
* O primeiro registro de cada bloco é o ancora
* Índice primário é esparso

(Exemplo:

**Custo da busca sem índice:**

V = 30.000 registros

B = 1024B (tamanho do bloco)

R = 100b (tamanho do registro)

Registro por Bloco(brf) = 1024/100 = 10

Blocos Necessários(b) = v/brf = 30.000/10 = 3.000

**Busca binária exige = log2 3000 = 12**

**Custo de busca com índice:**

Vi = 3000 (nº total de entrada de índices)

B = 1024b

Ri = 9 bytes para campo + 6 bytes ponteiro

= 15b

brfi = 1024/12 = 68

3000/68 = 45

**Busca binária com índice:**

**[log b] +1 = log2 45 +1 = 7**

)

**Índice Clustering**

-É utilizado para registros ordenados por um campo que não seja chave.

-Campo é agrupado por valor.

**Índice Secundário**

-Meio secundário de acesso para arquivo que já tenha índice primário

-Geralmente é feito na chave candidata

-É denso

**Indice de Multiplos Niveis**

-O objeto é reduzir o arquivo de indice, subdivindo-o

**Hashing**

-Obtemos diretamente o bloco do disco através de função

-Seja K o conjunto de valor de chave e B os endereços dos buckets. Uma função hash h é uma função de K para B