

**1a QUESTÃO)** Considere-se um disco com o tamanho do bloco  $B = 512$  bytes. Um ponteiro de bloco é  $P = 6$  bytes de comprimento, e um ponteiro de registro é  $PR = 7$  bytes. Um arquivo tem  $r = 30000$  registros de funcionários de comprimento fixo. Cada registro tem os seguintes campos: **Nome** (30 bytes), **Cpf** (9 bytes), **Código\_dept** (9 bytes), **Endereço** (40 bytes) **Telefone** (9 bytes), **Data\_nasc** (8 bytes), **Sexo** (1 byte), **Código\_cargo** (4 bytes) e **Salário** (4 bytes, número com decimais). Um byte adicional é utilizado como um marcador de exclusão.

(a) Calcule o tamanho de registro **R** (incluindo o marcador de exclusão) em bytes.

$$R = (30 + 9 + 9 + 40 + 9 + 8 + 1 + 4 + 4) + 1 = 115 \text{ bytes}$$

(b) Calcule o fator de bloqueio **bfr** e o número de blocos de arquivo **b** assumindo uma organização não espalhada.

$$bfr = \lfloor B / R \rfloor = \lfloor 512 / 115 \rfloor = 4 \text{ registros por bloco.}$$

$$\text{Número de blocos necessários para o arquivo } b = \lceil r / bfr \rceil = \lceil 30000 / 4 \rceil = 7500$$

(c) Suponha que o arquivo é ordenado pelo campo-chave **Cpf** e queremos construir um índice primário em **Cpf**. Calcular (i) o fator de bloco de índice **bfri**, (ii) o número de entradas do índice de primeiro nível e o número de blocos de índice de primeiro nível, (iii) o número de níveis necessário para torná-lo um índice multi-nível, (iv) o número total de blocos exigidos pelo índice multi-nível, e (v) o número de blocos acessados para buscar e recuperar um registro do arquivo - dado o seu valor de **Cpf** - utilizando o índice principal.

(i)

Tamanho de registro de índice  $R_i = (Cpf + P) = (9 + 6) = 15$  bytes.

$$\text{Fator de bloco de índice } bfri = \lfloor B / R_i \rfloor = \lfloor 512 / 15 \rfloor = 34 \text{ registros por bloco.}$$

(ii)

Número de registros do índice em 1o nível =  $ri1$  = número de blocos do arquivo = 7500 entradas.

$$\text{Número de blocos do índice em 1o nível } = bi1 = \lceil ri1 / bfri \rceil = \lceil 7500 / 34 \rceil = 221 \text{ blocos.}$$

(iii)

Número de registros do índice em 2o nível =  $ri2$  = número de blocos em 1o nível = 221 entradas.

$$\text{Número de blocos do índice em 2o nível } = bi2 = \lceil ri2 / bfri \rceil = \lceil 221 / 34 \rceil = 7 \text{ blocos.}$$

Número de registros do índice em 3o nível =  $ri3$  = número de blocos em 2o nível = 7 entradas.

$$\text{Número de blocos do índice em 3o nível } = bi3 = \lceil ri3 / bfri \rceil = \lceil 7 / 34 \rceil = 1 \text{ bloco.}$$

Uma vez que o terceiro nível tem apenas um bloco, é o nível de índice superior.

Assim, o índice tem  $x = 3$  níveis.

(iv)

O número de blocos do índice é  $bi1 + bi2 + bi3 = 221 + 7 + 1 = 229$  blocos.

(v)

Número de blocos acessados em uma busca via índice primário =  $x + 1 = 3 + 1 = 4$  blocos.

(d) Suponha que o arquivo não está ordenado pelo campo-chave **Cpf** e queremos construir um índice secundário para **Cpf**. Repita o exercício anterior (parte c) para o índice secundário e compare com o índice primário.

(i)

Tamanho de registro de índice  $R_i = (Cpf + P) = (9 + 6) = 15$  bytes.

Fator de bloco de índice  $bfri = \lfloor B / R_i \rfloor = \lfloor 512 / 15 \rfloor = 34$  registros por bloco.

(ii)

Número de registros do índice em 1o nível =  $ri1$  = número de registro do arquivo = 30000 entradas.

Número de blocos do índice em 1o nível =  $bi1 = \lceil ri1 / bfri \rceil = \lceil 30000 / 34 \rceil = 883$  blocos.

(iii)

Número de registros do índice em 2o nível =  $ri2$  = número de blocos em 1o nível = 883 entradas.

Número de blocos do índice em 2o nível =  $bi2 = \lceil ri2 / bfri \rceil = \lceil 883 / 34 \rceil = 26$  blocos.

Número de registros do índice em 3o nível =  $ri3$  = número de blocos em 2o nível = 26 entradas.

Número de blocos do índice em 3o nível =  $bi3 = \lceil ri3 / bfri \rceil = \lceil 26 / 34 \rceil = 1$  bloco.

Uma vez que o terceiro nível tem apenas um bloco, é o nível de índice superior.

Assim, o índice tem  $x = 3$  níveis.

(iv)

O número de blocos do índice é  $bi1 + bi2 + bi3 = 883 + 26 + 1 = 910$  blocos.

(v)

Número de blocos acessados em uma busca via índice primário =  $x + 1 = 3 + 1 = 4$  blocos.

(e) Suponha que o arquivo não é ordenado pelo campo **Codigo\_dept**, o qual também não é um campo chave, e queremos construir um índice secundário para **Codigo\_dept** usando um nível extra de indireção que armazena ponteiros de registro. Suponha que existem 1000 valores distintos de **Codigo\_dept**, e que os registros de funcionários estão uniformemente distribuídos entre esses valores. Calcular (i) o fator de bloco  $bfri$  do índice secundário, (ii) o número de blocos necessários pelo nível de indireção, o qual armazena ponteiros de registros, (iii) o número de entradas e de blocos do índice de primeiro nível, (iv) o número de níveis necessários se tornar um índice multi-nível, (v) o número total de blocos do índice de níveis múltiplos e nível extra de indireção, e (vi) o número aproximado de blocos acessados para buscar e recuperar todos os registros com um certo valor de **Codigo\_dept** usando o índice secundário.

(i)

Tamanho do registro de índice  $R_i = (Codigo\_dept + P) = (9 + 6) = 15$  bytes.

Fator de bloco  $bfri = \lfloor B / R_i \rfloor = \lfloor 512 / 15 \rfloor = 34$  registros de índices por bloco.

(ii)

Existem 1000 valores distintos de **Codigo\_dept**, de modo que o número médio de registro para cada valor é  $(r/1000) = (30000/1000) = 30$ .

Visto que um ponteiro registro possui  $PR = 7$  bytes, o número de bytes necessários ao nível de indireção para cada valor de DEPARTMENTCODE é de  $7 * 30 = 210$  bytes (encaixa em um único bloco). Assim,  $bind = 1.000$  blocos são necessários para o nível de indireção.

(iii)

Número de entradas de índice de primeiro nível  $ri1$  = número de valores distintos de **Codigo\_dept** = 1000 entradas.

Número de blocos de índice de primeiro nível  $bi1 = \lceil ri1 / bfri \rceil = \lceil 1000 / 34 \rceil = 30$  blocos.

(iv)

Número de entradas de índice de segundo nível  $ri2$  = número de blocos de índice de primeiro nível  $bi1 = 30$  entradas

Número de blocos de índice de segundo nível  $bi2 = \lceil ri2 / bfri \rceil = \lceil 30 / 34 \rceil = 1$ .

Assim, o índice  $x = 2$  níveis.

(v)

$bi1 + bi2 + bind = 30 + 1 + 1000 =$  blocos.

(vi)

Número de blocos acessados para procurar e recuperar o bloco contendo os ponteiros de registro no nível de indireção =  $1 + 1 + 1 = 3$ .

Se assumirmos que os 30 registros estão distribuídos por 30 blocos distintos, precisamos de um 30 blocos adicionais acessados para recuperar todos os 30 registros:  $3 + 30 = 33$ .

(f) Suponha que o arquivo é ordenado pelo **Codigo\_dept**, que é um campo não-chave, e queremos construir um índice de agrupamento em **Codigo\_dept** que usa o conceito de âncora para bloco (cada novo valor de **Codigo\_dept** inicia em um novo bloco). Suponha que existem 1000 valores distintos de **Codigo\_dept**, e que os registros de funcionários estão uniformemente distribuídas entre esses valores. Calcular (i) o fator de bloqueio do índice, (ii) o número de entradas do índice de primeiro nível e o número de blocos de índice de primeiro nível, (iii) o número de níveis necessários para se tornar um índice multi-nível, (iv) o número total de blocos exigidos pelo índice multi-nível, e (v) o número de bloco de acessados para buscar e recuperar todos os registros no arquivo de ter um específico valor DEPARTMENTCODE utilizando o índice de agrupamento (supor que vários blocos em um *cluster* são contíguas ou ligados por ponteiros).

(i)

Tamanho de registro de índice  $R_i = (\text{Codigo\_dept} + P) = (9 + 6) = 15$  bytes.

Fator de bloco do índice  $bfr_i = \lfloor B / R_i \rfloor = \lfloor 512 / 15 \rfloor = 34$  registros de índices por bloco.

(ii)

Número de entradas de índice de primeiro nível  $ri_1 = \text{número de valores distintos Codigo\_dept} = 1000$  entradas.

Número de blocos de índice de primeiro nível  $bi_1 = \lceil ri_1 / bfr_i \rceil = \lceil 1000 / 34 \rceil = 30$  blocos.

(iii)

Número de entradas de índice de segundo nível  $ri_2 = \text{número de blocos de índice de primeiro nível } bi_1 = 30$  entradas.

Número de blocos de índice de segundo nível  $bi_2 = \lceil ri_2 / bfr_i \rceil = \lceil 30 / 34 \rceil = 1$ .

O índice tem  $x = 2$  níveis.

(iv)

Número total de blocos para o índice de  $bi = bi_1 + bi_2 = 30 + 1 = 31$  blocos.

(v)

Número de blocos acessados para procurar o primeiro bloco do *cluster* de blocos =  $x + 1 = 2 + 1 = 3$ .

Os 30 registros (com mesmo valor de **Codigo\_dept**) são agrupados em  $\lceil 30 / bfr_i \rceil = \lceil 30 / 34 \rceil = 1$  bloco. Assim, o número total de blocos acessados para um determinado **Codigo\_dept** é  $x + 1 = 2 + 1 = 3$ .

(g) Suponha que o arquivo não é ordenado por **Cpf**, que é um campo de chave, e queremos construir um índice usando uma estrutura em árvore B+ em **Cpf**. Calcular (i) as ordens **p** e **pfolha** da árvore B+, (ii) o número de blocos de nível folha necessários, se os blocos estão aproximadamente 69% cheios (arredondado para cima), (iii) o número de níveis necessários, se nós internos também são 69% completos (arredondado para cima), (iv) o número total de blocos exigidos pela árvore B+, e (v) o número de blocos acessados para buscar e recuperar um registro do arquivo de dados, conhecendo seu **Cpf** e utilizando o índice em árvore B+.

(i)

Para uma árvore B+ de ordem  $p$ , a seguinte desigualdade deve ser satisfeita para cada nó interno da árvore:  $(p * P) + ((p - 1) * Cpf) < B$ , ou  $(P * 6) + ((p - 1) * 9) < 512$ , o que dá  $15p < 521$ , então  $p = 34$ .

Para os nós folha, assumindo que os ponteiros de registros estão incluídos nos nós folha, a seguinte

desigualdade deve ser satisfeita:  $(p_{\text{folha}} * (C_{\text{pf}} + P_{\text{R}})) + P < B$ , ou  $(p_{\text{folha}} + (9 + 7)) + 6 < 512$ , o que dá  $16p_{\text{folha}} < 506$ , de modo  $p_{\text{folha}} = 31$ .

(ii)

Assumindo que os nós estão 69% completos, em média, o número médio de valores de chave num nó de folha é  $0,69 * p_{\text{folha}} = 0,69 * 31 = 21,39$ . Se arredondar isso por conveniência, temos 22 valores-chave (e 22 ponteiros de registros) por nó folha.

Uma vez que o arquivo tem 30.000 registros e, portanto, 30.000 valores de **Cpf**, o número de nós (blocos) em nível folha  $b_1 = \lceil 30000 / 22 \rceil = 1364$  blocos.

(iii)

Considere o nível folha como nível 1, que possui  $b_1 = 1364$  blocos.

Os nós internos possuem  $\lceil 0,69 * p \rceil = \lceil 0,69 * 34 \rceil = \lceil 23,46 \rceil = 24$  valores por nó.

O número de bloco da árvore de segundo nível  $b_2 = \lceil b_1 / 24 \rceil = \lceil 1364/24 \rceil = 57$ .

O número de blocos de árvores de terceiro nível  $b_3 = \lceil b_2 / 24 \rceil = \lceil 57/24 \rceil = 3$ .

O número de blocos de árvores de quarto nível  $b_4 = \lceil b_3 / 24 \rceil = \lceil 3/24 \rceil = 1$  (nó raiz).

Desde o quarto nível tem um único bloco, a árvore tem  $x = 4$  níveis (contando o nível folha).

(iv)

O número total de blocos para a árvore  $b_i = b_1 + b_2 + b_3 + b_4 = 1364 + 57 + 3 + 1 = 1425$  blocos.

(v)

O número de blocos acessados para procurar um registro  $= x + 1 = 4 + 1 = 5$  blocos.