ASSIGNMENT 3 DATABASE MANAGEMENT SYSTEMS. 01-Apr-2011.

MOHAN MUPIGERE PUTTARAJU
G39789082.

Indexing.

Disk with block size = $B = 512$ bytes.

Pointer $P = 6$ bytes long.

record pointer is $P_R = 7$ bytes.

A file has $r = 30,000$ EMPLOYEE records of fixed length.

Name (30 bytes)

SSN (9 bytes)

Department code (9 bytes)

Address (40 bytes)

Phone (9 bytes)

Birthdate (8 bytes)

Sex (1 byte)

Jobcode (4 bytes)

Salary (4 bytes, real numbers)

a) Record length = sum of all bytes used to save data.

$= (30 + 9 + 9 + 40 + 9 + 8 + 1 + 4 + 4) + 1$ (additional byte)

$= 115$ bytes

b) Blocking factor bfr.

$$\text{Blocking factor} = \text{floor}\left(\frac{\text{Block size}}{\text{record size}}\right)$$

$$= \text{floor}\left(\frac{512}{115}\right)$$

$$= 4.45 \approx 4 \text{ } \cancel{\text{blocks}} \text{ records / block.}$$

c)

i) index record size $R_i = [SSN + \text{pointer}(P)]$

$$= 9 + 6$$

$$= 15 \text{ bytes.}$$

index blocking factor $bfr_i = fo = \text{floor}\left(\frac{\text{Block size}}{\text{record size}}\right)$

$$= \text{floor}\left(\frac{512}{15}\right) = 34$$

ii) Number of first level index entries (r_1)

= number of file blocks b = 7500 entries.

i.e. blocks needed for file $= \text{ceiling}\left(\frac{\text{Record}}{\text{blocking factor}}\right)$

$$= \text{ceiling}\left(\frac{30000}{4}\right)$$

$$= 7500$$

Number of first level index blocks

$$= \text{ceiling}\left(\frac{r_1}{bfr_i}\right) = \text{Ceiling}\left(\frac{7500}{34}\right) = 221 \text{ blocks}$$

iii) For multi level.

Number of second-level index ~~blocks~~ entries r_2 = number of first-level blocks b_1.

= 221 entries.

Number of second-level index blocks b_2 = ceiling $\left(\frac{r_2}{bfr}\right)$

= ceiling $\left(\frac{221}{34}\right)$

= 7 entries.

Number of third-level index ~~blocks~~ entries r_3 = number of ~~first~~ second level index blocks

= 7 entries.

Number of third level index blocks b_3 = ceiling $\left({r_3}/{bfr} \right)$ = ceiling

= 1

Hence index has $X = 3$ levels.

iv) Total number of blocks for the index $b = b_1 + b_2 + b_3$

= 221 + 7 + 1

= 229 blocks

v) Number of blocks accesses to search for ~~for the index~~ record = $X+1$ = 3 + 1 =

d)

i) Index record size R_i = SSN + pointer = 9+6 = 15 bytes

Index blocking factor bfr = (fan-out) fo = floor $\left(\frac{\text{Block size}}{\text{Record size}}\right)$

= floor (512/15)

= 34 index records per block.

ii) Number of first-level index entries r_1 = number of records

r = 30000

number of first-level index blocks b_1 = ceiling $\left(r_1/bfr\right)$

= ceiling $\left(\frac{30000}{34}\right)$

= 883 blocks

iii) We can calculate the number of levels as follows.

Number of second-level index entries r_2 = number of first-level index blocks b_1 = 883 entries.

Number of second-level index blocks b_2 = ceiling $\left(r_2/bfr\right)$ = ceiling$\left(883/34\right)$

= 26 blocks.

Number of third level index entries r_3 = number of second-level index blocks b_2 = 26 entries.

Number of third-level index blocks b_3 = ceiling $\left(r_2/bfr\right)$ = ceiling$\left(\cancel{883}\ 26/34\right)$

= 1.

Since third-level index block has only one block, it is the top index level.

Hence, the index has $\underline{\underline{3 \text{ levels}}}$ (x).

iv) Total number of blocks for index = $b_1 + b_2 + b_3$

= 883 + 26 + 1

= $\underline{\underline{910}}$

v) Number of blocks accesses to search for a record

= $x + 1 = 3 + 1 = \underline{\underline{4}}$

e) i) Index record R = (Department code + pointer)

= 9+6 = 15 bytes.

Index blocking factor bfr i = (fan-out) fo = floor$\left(\frac{\text{Block size}}{\text{Record size}}\right)$

= floor $(512/15)$

= 34 index records per block.

ii) There are 1000 distinct values of Department code.

So average number of record for each value is $r/1000$

$= \frac{30000}{1000} = 30$

Since ~~a~~ record pointer size = 7 bytes.

Number of bytes needed at the level of indirection for each value of department code is $7 \times 30 = 210$ bytes.

which fits in one block.

∴ 1000 blocks are needed for level of indirection.

iii) Number of first-level index entries r_1 = number of ~~first-level block~~ distinct values of Department code = 1000 entries.

Number of first-level index blocks b_1 = ceiling(r_1/bfr) = ceiling$(1000/36)$

= 30 blocks.

iv) We can calculate the number of level as follows

Number of first-level index blocks b_1 = ceiling(r_1/bfr)

= ceiling$(1000/34)$

= 30 blocks.

Number of second-level index entries r_2 = number of first level index block, b_1

= 30 entries..

Number of second-level index-blocks b_2 = ceiling (r_2/bfr)

= $30/34$ = 1

Hence index has x = 2 levels.

v) Total number of blocks for index = $b_1 + b_2 + b$ indirection.

= $30 + 1 + \cancel{1030}\ 1031$

= 1031 blocks

vi) Number of block accesses to search for and retrieve the blocks containing the record pointers at the level of indirection = x + 1
= 2 + 1 = 3 blocks accesses

for 30 records are distributed over 30 district blocks.

We need 30 additional blocks accesses to retrive ~~30~~ all 30 records.

Hence total blocks accesses needed on avg. to retrive all records with given value of dept code.

$= x + 1 + 30$

$= 33$

f).

i) Index record size R = Department code + pointer

$= 9 + 6 = 15$ bytes.

Index blocking factor bfr = fan-out (fo) = floor $\left(\frac{\text{block size}}{\text{records}}\right)$

$= 512/15 = 34$ index records per block.

ii) Number of first-level index entries $r1$

= number of first-level index blocks $b1$.

$= \text{ceiling}\left(r1/bfr\ i\right)$

$= \text{ceiling}\left(\frac{1000}{34}\right) = 30$ blocks.

Number of Second-level index blocks b_2 = ceiling (r_2/bfr) = ceiling $(30/34)$

= 1

Since Second level has one block, it is the top index level.

Hence index has $x = 2$ levels.

iv) Total number of blocks for the index = $b_1 + b_2 = 30 + 1$

= 31 blocks

v) Number of block accesses to search for the first block in the cluster of blocks = $x + 1$

= $2 + 1$ = 3

The 30 records are clustered in ceiling $(30/bfr)$ = ceiling $(30/4)$

= 8 blocks

Hence, total block accesses needed on average to retrieve all the records with a given department code

$x + \cancel{8}$

= $2 + 8$

= 10 block accesses

9) B+ tree of order p.

following inequality must be satisfied for each internal tree node $(P * P) + ((p-1) * SSN) \leq$ Block size.

P = pointer length.

$$(P * 6) + ((P-1) * 9) \leq 512$$

which gives $15p \leq 521$

$\therefore$ $p = 34$

For leaf nodes, assuming that record pointers are included in the leaf nodes.

the following inequality must be satisfied.

(pointer leaf $\times$ (SSN + Pointer record)) + P $\leq$ B.

$\therefore$ $[P \text{ leaf} \times (9+7) + 6] \leq 512.$

$16p$ leaf $\leq 506.$

$\therefore$ P leaf = 31

ii) Assuming the nodes are 69% of leaf.

$\therefore$ 0.69 P leaf = $0.69 * 31 = 21.39$

Since the file has 30000 records and hence 30,000 values of SSN. the number of leaf-level nodes (blocks) needed is $bl = \text{ceiling}\left(\frac{30,000}{22}\right) = 1364$ blocks

iii) we can calculate the number of levels as follows.

The avg. fan-out for the internal nodes

$fo = \text{ceiling}(0.69 * p) = \text{ceiling}(0.69 * 34)$

$= \text{ceiling}(23.46) = \underline{24}$

Second-level tree blocks $b2 = \text{ceiling}(b1/fo) = \text{ceiling}(1364/24)$

$= \underline{57 \text{ blocks}}$

Third-level tree blocks $b3 = \text{ceiling}\ b_2 = \text{ceiling}(b1/fo)$

$= \text{ceiling}(\cancel{1364}\ 57/24)$

$= \cancel{57}\ 3$ blocks.

Fourth-level tree blocks $b4 = \text{ceiling}(b3/fo) = \text{ceiling}(3/24)$

$= 1$

Since fourth level has only one block. the tree has $x=4$ levels

$\therefore X = \text{ceiling}\left(\log_{(bo)}(b1) + 1\right)$.

$= \text{ceiling}\left(\log_{(24)} 1324\right) + 1$

$= 3+1 = 4$ levels.

iv) Total number of blocks for tree $= b1 + b2 + b3 + b4$

$= 1364 + 57 + 3 + 1$

$= 1425$ blocks

v) number of blocks accesses to search for a record

$= X + 1 = 4 + 1 = 5$

20.

a) P ← address of top level block of Index.

for (j = t; j < step-1 to 1 do.

begin.

read the index block (at j^{th} Index level)

whose address is P.

compare index block (j^{th} Index level) to next index block. (j+1 index level. block).

P ← $P_j(i)$ (picks appropriate pointer at j^{th} index level).

end;

read the data file block whose address is P.

b) $p \leftarrow$ address of top level block of Index.

for $i = 0$; $i < step-1$; $i++$ do

begin.

read the Index block. whose address is p.

; i[p]

search block p for entry i such that.

$K[i] \leq K < K[i+1]$

if (K[i] is last entry of the block), it is sufficient to satisfy $K[i] \leq K$);

$p \leftarrow p[i]$

end;

read the data file block whose address is p.

search block p for record with key = K;

e) Clustering _ Index.

index file : Clustering field value & Block pointer.

$p \leftarrow$ address of top level block of Index.

for $j \leftarrow t$ step -1 to 1 do.

begin.

read the Index block: whose address is p:

$k_j(i) \leq k < k_j(i+1)$

If ($k_j(i)$ is last entry in the block.

it is sufficient to satisfy $k_j(i) \leq k$:

$p \leftarrow P_j(i)$

end:

read the data file block whose address is

search block p for record with key $= k_j$.