17.18

Consider a disk with block size B = 512 bytes. A block pointer is P = 6 bytes long, and a record pointer is PR = 7 bytes long. A file has r = 30,000 EMPLOYEE records of fixed length. Each record has the following fields:Name (30 bytes),Ssn (9 bytes), Department_code (9 bytes), Address (40 bytes), Phone (10 bytes), Birth_date (8 bytes), Sex (1 byte), Job_code (4 bytes), and Salary (4 bytes, real number). An additional byte is used as a deletion marker.

a. Calculate the record size R in bytes.
   ➔ Record size (R) = (30+9+9+40+10+8+1+4+4)+1 = 116 bytes

b. Calculate the blocking factor bfr and the number of file blocks b, assuming an unspanned organization.
   ➔ Blocking factor(bfr) = floor (B/R) = floor (512/116) = 4 records per block
   ➔ Number of file blocks (b) = ceiling(r/bfr) = ceiling (30000/4) = 7500

c. Suppose that the file is ordered by the key field Ssn and we want to construct a primary index on Ssn. Calculate (i) the index blocking factor bfri (which is also the index fan-out fo); (ii) the number of first-level index entries and the number of first-level index blocks; (iii) the number of levels needed if we make it into a multilevel index; (iv) the total number of blocks required by the multilevel index; and (v) the number of block accesses needed to search for and retrieve a record from the file—given its Ssn value—using the primary index.

i. Index record size (R_i) = (Ssn + P)

= (9+6)

= 15 bytes

Index blocking factor (bfr_i)= floor(B/R_I)

= floor(512/15)

= 34

ii. Number of first-level index entries (r_1) = number of file blocks(b)

= 7500 entries

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

= ceiling(7500/34)

= 221 blocks

iii. Number of second-level index entries (r_2) = number of first-level blocks(b_1)

= 221 entries

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

= ceiling (221/34) = 7 blocks.

Number of third-level index entries (r_3) = number of second level index blocks(b_2)

= 7 entries

Number of third-level index blocks (b_3) = ceiling (r_3/bfr_i)

= ceiling (7/34)

= 1

We can see that the third level has only one block. So, it is the top index level. Therefore, there are x=3 index levels.

iv. Total number of blocks required by the multilevel index (b_i) = b_1+b_2+b_3

= 221+7+1

= 229 blocks

v. Number of block accesses needed to search = x+1 = 3+1 = 4

d. Suppose that the file is not ordered by the key field Ssn and we want to construct a secondary index on Ssn. Repeat the previous exercise (part c) for the secondary index and compare with the primary index

i Index record size (R_i) = (SSN + P)

= 9 + 6 = 15 bytes

Index blocking factor (bfr_i) = floor (B/R_I) = floor (512/15) = 34 index records per block

ii. Number of 1^{st} level index entries (r_1) = number of file records(r) = 30000

Number of 1^{st} level index blocks (b_1) = ceiling (r_1/bfr_i)

= ceiling (30000/34)

= 883 blocks

iii. Number of 2^{nd} level index entries (r_2) = number of 1^{st} level index blocks (b_1)

= 883 entries

Number of 2^{nd} level index blocks (b_2) = ceiling (r_2/bfr_i)

= ceiling (883/34)

= 26 blocks

Number of 3^{rd} level index entries (r_3) = number of 2^{nd} level index blocks(b_2)

= 26 entries

Number of 3^{rd} level index blocks (b_3) = ceiling (r_3/bfr_i) = ceiling(26/34) = 1

The third level has 1 block, so it the top index level. Therefore,the index has 3 levels.

iv. Total number of blocks required by multilevel index (b_i) = $b_1+b_2+b_3$ = 883 + 26 + 1 = 910

v. Number of block accesses = x+1 = 3+1 = 4

e. Suppose that the file is not ordered by the nonkey field Department_code and we want to construct a secondary index on Department_code, using option 3 of Section 17.1.3, with an extra level of indirection that stores record pointers. Assume there are 1,000 distinct values of Department_code and that the EMPLOYEE records are evenly distributed among these values. Calculate (i) the index blocking factor bfri (which is also the index fan-out fo); (ii) the number of blocks needed by the level of indirection that stores record pointers; (iii) the number of first-level index entries and the number of first-level index blocks; (iv) the number of levels needed if we make it into a multilevel index; (v) the total number of blocks required by the multilevel index and the blocks used in the extra level of indirection; and (vi) the approximate number of block accesses needed to search for and retrieve all records in the file that have a specific Department_code value, using the index.

i Index record size (R_i) = (Department_code + P)

= 9 + 6

= 15 bytes

Index blocking factor (bfr_i) = floor (B/R_I) = floor(512/15)

= 34 index records per block

ii. Given,

Record pointer size (PR) = 7 bytes

distinct values of department code = 1000

Therefore, the average number of records for each value is (r/1000) = (30000/1000) = 30

The number of bytes needed at the level ofindirection for each value of Department code is

7*30 = 210 bytes, which fits in one block. Therefore, we need 1000 blocks for the level of indirection.

iii. Number of 1^{st} level index entries (r_1) = distinct values of departmentCode = 1000 entries

Number of 1^{st} level index blocks (b_1) = ceiling (r_1/bfr_1)

= ceiling (1000/34)

= 30 blocks

iv. Number of 2^{nd} level index entries (r_2) = number of 1^{st} level index blocks (b_1) = 30 entries

Number of 2^{nd} level index blocks (b_2) = ceiling (r_2/bfr_i) = ceiling (30/34) = 1

Since there is only 1 block at the second level. The index has x= 2 levels

v Total number of blocks required by the multilevel index (b_i) = $b_1 + b_2$ = 30 + 1 = 31

vi. Number of block accesses = x+1 = 2+1 = 3