Consider a disk with block size B=512 bytes. A block pointer is P=6 bytes long,and a record pointer is P R =7 bytes long. A file has r=30,000 EMPLOYEE records of fixed-length. Each record has the following fields: NAME (30 bytes), SSN (9bytes), DEPARTMENTCODE (9 bytes), ADDRESS (40 bytes), PHONE (9 bytes),BIRTHDATE (8 bytes), SEX (1 byte), JOBCODE (4 bytes), SALARY (4 bytes, real number). An additional byte is used as a deletion marker.

(a) Calculate the record size R in bytes.

Record length R = (30 + 9 + 9 + 40 + 9 + 8 + 1 + 4 + 4) + 1 = 115 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(512/115) = 4 records per block
Number of blocks needed for file = ceiling(r/bfr) = ceiling(30000/4) = 7500

(c) Suppose 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 bfr_i (which is also the index fan-out fo);

Index record size R_i = (V SSN + P) = (9 + 6) = 15 bytes
Index blocking factor bfr_i = fo = floor(B/R_i) = floor(512/15) = 34

(ii) the number of first-level index entries and the number of first-level index blocks;

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) the number of levels needed if we make it into a multi-level index;

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
Since the third level has only one block, it is the top index level.
Hence, the index has x = 3 levels

(iv) the total number of blocks required by the multi-level index; and

Total number of blocks for the index $b_i = b_1 + b_2 + b_3 = 221 + 7 + 1$
$= 229$ blocks

(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.

Number of block accesses to search for a record $= x + 1 = 3 + 1 = 4$

(d) Suppose 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 $R_i = (V_{SSN} + P) = (9 + 6) = 15$ bytes
Index blocking factor bfr_i = (fan-out) $fo = floor(B/R_i) = floor(512/15)$
$= 34$ index records per block
(This has not changed from part (c) above)

ii. Number of first-level index entries r_1 = number of file records $r = 30000$
Number of first-level index blocks $b_1 = ceiling(r_1 /bfr_i) = ceiling(30000/34)$
$= 883$ 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 $b_2 = ceiling(r_2 /bfr_i) = ceiling(883/34)$
$= 26$ 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 $b_3 = ceiling(r_3 /bfr_i) = ceiling(26/34) = 1$
Since the third level has only one block, it is the top index level.

Hence, the index has $x = 3$ levels

iv. Total number of blocks for the index $b_i = b_1 + b_2 + b_3 = 883 + 26 + 1 = 910$

v. Number of block accesses to search for a record $= x + 1 = 3 + 1 = 4$

(e) Suppose the file is not ordered by the non-key field DEPARTMENTCODE and we want to construct a secondary index on SSN using the Option given in Slide 14 , with an extra level of indirection that stores record pointers. Assume there are 1000 distinct values of DEPARTMENTCODE, and that the EMPLOYEE records are evenly distributed among these values. Calculate

(i) the index blocking factor bfr_i (which is also the index fan-out fo)

Index record size R_i = (V DEPARTMENTCODE + P) = (9 + 6) = 15 bytes
Index blocking factor bfr_i = (fan-out) fo = floor(B/R_i) = floor(512/15)
= 34 index records per block

(ii) the number of blocks needed by the level of indirection that stores record pointers

There are 1000 distinct values of DEPARTMENTCODE, so the average number of records for each value is (r/1000) = (30000/1000) = 30
Since a record pointer size P R = 7 bytes, the number of bytes needed at the level of indirection for each value of DEPARTMENTCODE is 7 * 30 =210 bytes, which fits in one block. Hence, 1000 blocks are needed for the level of indirection.

(iii) the number of first-level index entries and the number of first-level index blocks

Number of first-level index entries r_1
= number of distinct values of DEPARTMENTCODE = 1000 entries
Number of first-level index blocks b_1 = ceiling(r_1 /bfr_i) = ceiling(1000/34)
= 30 blocks

(iv) the number of levels needed if we make it a multi-level index

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
= 30 entries
Number of second-level index blocks b_2 = ceiling(r_2 /bfr_i) = ceiling(30/34) = 1

Hence, the index has x = 2 levels