Assignment- 4
Indexing and File Organization

Name : Nabin Kumar Bhakat
Roll Name: 2022PGCSCA070

1) Why can we have at most one primary or clustering index on a file, but several secondary indexes?

Ans: Nature of these indexes and their impact on the file organization is the differences here. A primary or clustering index is unique because it defines the physical order of the records in the file, when a new record is inserted into the file, it must be placed in its correct position according to the primary key value. If there were multiple primary index on different fields, there would be conflicting orders for the same field, making it impossible to maintain consistency.

On the otherhand, secondary indexes do not affect the physical order of the records. They only provide additional paths access the data based on different fields. Therefore it is possible to have several secondary indexes on a file without conflicting with each other or with primary index.

In short, the reason for this is that primary or clustering index determines the physical order of the record in a file, whereas a secondary index only provides a additional paths to access the data.

Therefore, having the multiple primary or clustering indexes would result in conflicting physical orders for the same file, which impossible to maintain.

2) a) Record Length = (30 + 9 + 9 + 40 + 9 + 8 + 1 + 4 + 4) = 115 bytes

b) Blocking factor = $\left\lfloor \frac{B}{R} \right\rfloor = \left\lfloor \frac{512}{115} \right\rfloor$ = 4 record / block

Number of blocks ~~records~~ needed = $\left\lceil \frac{r}{\text{blocking factor}} \right\rceil = \left\lceil \frac{30000}{4} \right\rceil$ = 7500

c) i) Index record size R_i = (USSN + P) = (9 + 6) = 15 bytes
index blocking factor block factor i = $\left\lfloor \frac{512}{15} \right\rfloor$ = 34

ii) Number of second-level index entries r_2 = no. of first level blocks b_1 = 221 entries.

~~No of~~ No. of second level index blocks $b_2 = \left\lceil \frac{221}{34} \right\rceil$ = 7 blocks

No. of third-level index block $b_3 = \left\lceil \frac{r_3}{\text{block factor}} \right\rceil = \left\lceil \frac{7}{34} \right\rceil = 1$

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

Total no. of blocks required by the multiple level index
Total no. of blocks for the index $b_i = b_1 + b_2 + b_3$
= 221 + 7 + 1 = 229

No. of blocks access to search for a record = x + 1 = 3 + 1 = 4

d) i) Index record size R_i = 9 + 6 = 15 bytes.
Index blocking factor $b r_i = \left\lfloor \frac{B}{R_i} \right\rfloor = \left\lfloor \frac{512}{15} \right\rfloor$ = 34 index record per block.

ii) No. of first-level index entries r_1 = no. of file record r = 30000

No of first-level index blocks $b_1 = \left\lceil \frac{30000}{34} \right\rceil$ = 883 blocks.

iii) No. of second-level entries r_2 = no. of first level index block b_1 = 883 entries.

no. of second-level index blocks $b_2 = \left\lceil \frac{r_2}{\text{block factor}} \right\rceil = \left\lceil \frac{883}{34} \right\rceil$ = 24 blocks

no. of third-level index entries r_3 = no. of second level index block b_2
= 26 entries

no. of third-level index block $b_3 = \left\lceil \frac{r_3}{\text{block factor}} \right\rceil = \left\lceil \frac{26}{34} \right\rceil = 1$

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

iv> Total no. of blocks for the index $b_i = b_1 + b_2 + b_3 = 863 + 26 + 1$
$= 890$

v> No. of blocks accesses of search for a record $= x + 1 = 3 + 1 = 4$

e> i> Index record size $= 9 + 6 = 15$ bytes

Index blocking factors $= \left\lfloor \frac{512}{15} \right\rfloor = 34$ index records per block.

ii> There are four distinct values of DEPARTMENT CODE, so the avg no. of records for each value is $\left(\frac{r}{1000}\right) = \left(\frac{30000}{1000}\right) = 30$

Since record pointer size $= 7$ bytes

No. of bytes needed at the level of indirection for each value of DEPARTMENT is $7 \times 30 = 210$ bytes.

iii> No. of first level index entries = no. of distinct values of DEPARTMENT CODE $= 1000$ entries

no. of first-level index-block $b_1 = \left\lceil \frac{r_1}{\text{block factor}} \right\rceil = \left\lceil \frac{1000}{34} \right\rceil$
$= 30$ blocks.

iv> No. of second level index block $b_2 = \left\lceil \frac{30}{34} \right\rceil = 1$

Hence index has $x = 2$ levels.

v> Total no. of blocks for the index $b_i = b_1 + b_2 + b_3$
$= 30 + 1 + 1000$
$= 1031$ blocks

vi> No. of block accesses to search for and retreive the block containing the record pointers at the level of indirection
$= x + 1 = 2 + 1 = 3$ blocks.

We need an additional 30 block accesses need on avg. to retreive all the records with a given value of DEPARTMENT CODE
$= x + 1 + 30 = 33$

8) i) Index record size = 9+6 = 15 bytes

Index blocking factor = $\lfloor \frac{512}{15} \rfloor$ = 34 index records per blocks

ii) No. of first-level index entries = no. of distinct DEPARTMENT CODE value = 1000 entries.

no. of first-level index block $b_1 = \lceil \frac{r_1}{\text{block factor}} \rceil = \lceil \frac{1000}{34} \rceil = 30$ blocks

iii) No. of second-level index entries r_2 = no. of first-level index block b_1 = 30 entries

No. of second-level index block = $\lceil \frac{30}{34} \rceil = 1$

Since the second level has one block it is due to p index level. Hence the index has $x = 2$ levels.

iv) Total no. of blocks for the index = $b_1 + b_2 = 30+1 = 31$ blocks

v) No. 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 $\lceil \frac{30}{4} \rceil = 8$ blocks

Hence total block accesses needed on avg to retrieve all the records with a given DEPARTMENT CODE = $x + 8$
$= 2+8 = 10$ blocks

9) i) Brand nodes will contain the SSN search key and block pointers to outer nodes so $p = \lfloor (B-P)/(V+P) \rfloor$
$= \lfloor (512-6)/(9+6) \rfloor$
$= 34$ nodes.

will have a record pointer for each key value (because the ~~field~~ file is not ordered on that field) plus one block pointer to the next leaf node, so $P_{leaf} = \lfloor (512-6)/(9+7) \rfloor = 31$

ii) There will be $\lceil 0.69 \cdot P_{leaf} \rceil = \lceil 0.69 \cdot 31 \rceil = 22$ search key per leaf node.

No. of leaf blocks is $\lceil 30000/22 \rceil = 1364$ blocks.

The celing because the last block will be partially ful.

iii) ~~There are 1364~~ By the same reasoning there will be

$\lceil 0.69 \cdot P \rceil = \lceil 0.69 \cdot 34 \rceil = 24$ block pointers per branch node

If there are 1364 blocks in the first level of the tree, there must be 1364 blocks pointers in the second level.

So there are $\lceil 1364/24 \rceil = 57$ blocks in the second level

Similarly $\lceil 57/24 \rceil = 3$ blocks in the third level and clearly 1st block in the fourth level. so $x = 4$

iv) 1364 at first level, 57 at the second, 3 at the third and 1 at the fourth. The sum is 1425 blocks.

v) For to traverse the height, plus one to access the full record in the primary file 505.

h) i) The order p and Pleaf of the B-tree

key field = 9 bytes

Block pointer = 6 bytes

Record pointer = 7 bytes

Let the order of the tree be p

Then we have $(p-1) * 16 + p * 6 < 512$

$$22p <= 528$$

$$p <= 24$$

$$p = 24$$

Assuming that each node of the B-tree is 69% full node on the avg will have $p * 0.69 = 24 \times 0.69 = 16$ nodes

Therefore the order of the tree is 16

ii) The number of leaf-level blocks needed if blocks approximately 69% full (round up for convenience)

Root	1 node	15 entries	16 pointers
level 1	16 nodes	16 * 15 = 240 entries	16 * 16 = 256 pointers
level 2	256 nodes	256 * 15 = 3840 entries	256 * 16 = 4096 pointers
level 3	4096 nodes	4096 * 15 = 81440 entries	

Hence the no. of leaf-level blocks needed = 4096

iii) Using the logarithmic height formula

$$x = [\log_{10}(4096)] + 1 = 4$$

iv) Total no. of blocks required by the B-tree

$$= 1 + 16 + 256 + 4096$$

$$= 4369$$

v) The no. of block access needed = no. of levels the tree + 1

$$= 4 + 1 = 5$$