GATE

CS & IT

Database Management System

File organization and Indexing

DPP: 1

Q1 Assume a relational database system that holds relation: C(colleges) with the following characteristics

- Records are stored as fixed length, fixed format records, length is 256 bytes.
- There are 16384 records.
- Records contains key attribute CollegeNumber (C.N), length 22 bytes and other fields.
- Unspanned organization is used to store the information or record.

Let's suppose we want to build a sparse primary index on C.N then how many numbers of 4096-byte blocks are needed to store the primary index when block pointer size is 10 bytes ______?

(A) 7 (B) 8

(C) 9 (D) 10

Q2 Assume a relational database system that holds relation: Product (P) with the following characteristics

- Records are stored as fixed length, fixed format records, with the length of 256 bytes.
- There are 262144 records.
- Records contain attribute P.I (The identifier of the product involved), with the length 24 bytes, and an attribute P.C (the cost of product), with the length 32 bytes and other fields.
- Unspanned organization is used to store the record.

Assume that we want to build a dense secondary index on P.C, then how many numbers of 4096-byte blocks needed to store the dense secondary index. When record pointer size is 32 bytes? _______.

Q3 Consider a SQL statement SELECT P_1, P_2, P_3 from Q WHERE P_2 = 'Pavan' is frequently executed, which column(s) should be considered for indexing based only on the statement itself ?

(A) P_1 only (B) P_2 only

(C) P_3 only (D) P_1, P_2 and P_3

Q4 Consider the following specification of system-

Disk block size = 2048 bytes

Block pointer size = 16 bytes

Record pointer size = 20 bytes long

file contains 30,000 records.

Each record of the file has the following fields:

Fields	Size (in Bytes)
Emp Name	5
Emp Num	10
Dept Num	9
Addr	20
PhNum	9
DOB	1
Sex	1
Job	3
Sal	5

An extra/additional byte is used per record to represent end of the record.
What is the block factor of the database file assuming unspanned file organization ?

(A) 16 (B) 32
(C) 48 (D) 64

Q5 Which one of the following statements is/are True___regarding indexing ?

(A) A database file can contain multiple clustered indexes.
(B) A database file can consist of only one clustered index with multiple secondary indexes
(C) A database file can consist of multiple primary indexes.
(D) A database file can consist of both primary and clustered index.

Q6 Consider a database of fixed-length records stored as an ordered file. The database has 25,000 records with each records being 100 bytes, of which the non-key attribute on which clustering index is formed occupies 10 bytes. The data file is completely block aligned.
Suppose, block size, of the file system is 512 bytes and a pointer to the block occupy 5 bytes.
You may assume that a binary search on an index file of b block may take $\lceil \log_2 b \rceil$ accesses in worst case.
Given that a cluster consumes 2 blocks, the number of block accesses required to identify the desired data in the worst case is ______.

Q7 Consider the following statements-
S_1: If the records of a relation X are physically ordered over a non-key field P and an index is build over the key-field of relation X, then the index is necessarily a secondary index over key attribute.
S_2: More than one secondary indexes are possible.
Which of the given statement(s) is/are CORRECT ?

(A) S_1 only
(B) S_2 only
(C) Both S_1 and S_2
(D) Neither S_1 nor S_2

Q8 The order of a leaf node in a B^+ tree is the maximum number of (value, data record pointer) pairs it can hold. Given that the block size is 1K bytes, data record pointer is 8 bytes long, the value field is 10 bytes long and a block pointer is 6 bytes, then what is the order of the leaf node?

(A) 53 (B) 54
(C) 55 (D) 56

Q9 Given a block can hold either 3 records or 10 key pointers. A database contains P records, then how many blocks do we need to hold the data file and the dense index?

(A) $\frac{P}{30}$ (B) $\frac{P}{3}$
(C) $\frac{13P}{30}$ (D) $\frac{P}{10}$

Q10 The order of an internal node in B^+ tree index is the maximum number of children it can have. Assume that a child pointer takes 6 bytes, the search field value takes 34 bytes and the blocks size is 2048 bytes. The order of the internal node is ________.

Q11 Assume a disk with block size B = 1024 Bytes, A block pointer is P_B = 12 bytes long and a record pointer is P_R = 18 bytes long. A file has 1,00,000 patients records of size 100 bytes. Suppose the file is ordered by the key field PID and we want to construct a secondary (dense) index on non-key field DeptID (14 bytes), then minimum of how many blocks are required to store index file assuming an unspanned organisation?
(A) 3000
(B) 3100
(C) 3125
(D) None of the above

Q12 The order of a node in B tree is the maximum number of block pointers it can hold. Given that the block size is 2K bytes, data record pointer is 8 bytes long, the search key is 9 bytes long and a block pointer is 5 bytes long. The best possible order of B tree node is____.

Q13 Consider the keys (1– 5000) are going to be interested into a B^+ tree. Assume, all the order are available before insertion. The orders P for B^+ tree node is defined as-
2 to P pointer for root
$\left\lceil \frac{P}{2} \right\rceil$ to P pointer for another node.
The maximum possible levels in a B^+ tree index for P = 9 is_________.
(Assume that level of the root node is 1)

Q14 Consider the following statements:
S_1: In a B^+ tree, data pointers are stored only at the leaf nodes of the tree.
S_2: The leaf node has an entry for every value of the search field, along with the data pointer to the record.
Choose the correct statements.
(A) Only S_1 is true
(B) Only S_2 is true
(C) Both S_1 and S_2 are true
(D) Neither S_1 nor S_2 is true

Q15 Which of the following is/are true reading B^+ tree?
(A) Records can be fetched in equal number of disk access.
(B) Height of the tree remains balanced and less as compared to B tree.
(C) Keys are used for indexing
(D) Faster search queries as the data is stored only on the leaf nodes.

Q16 Consider the following specification of system with disk block size 2048 bytes, block pointer size 14 bytes, record pointer size 18 bytes long and file size 60,000 records. Each record of file is 256 bytes long and record of the size is sorted on the key field. If the primary index (sparse) is built on the kye field (ESN) which is 18 bytes long. What is the Index blocking factors (That is number of indexes per block)
Assuming unspanned file organization ________.

Q17 **Data For This & Next Question :**
Consider a disk blocking size B = 1024 bytes. A block pointer is PB =12 bytes long and a record pointer is PR = 7 bytes long. A file has r = 60,000 patient records of fixed length. The size of record is 230 bytes. Suppose the file is not ordered by the key field PSN (18 bytes) and we want to construct a secondary index on PSN.
The number of first level index entries is_____.

Q18 The number of first level index block is_______?
(A) 1800 (B) 1825
(C) 1850 (D) 1857

Q19 Consider an unordered file of 10^6 records with records size of 200 bytes stored on blocks of 8KB with a spanned records organization. We will assume that no system related information is stored within a block, then how many blocks would it be need to store this file ?
(A) 24400 (B) 24405

(C) 24410 (D) 24415

Q20 Consider the following statements:

S_1: for any given data file, it is possible to create two different sparse first level indexes on various keys.

S_2: for any given data file, it is possible to create two different denes first level indexes on various keys.

Select the correct statements.

(A) Only S_1 correct

(B) Only S_2 correct

(C) Both S_1 and S_2 is correct

(D) Neither is S_1 nor S_2 is correct.

Answer Key

Q1	(B)
Q2	(4096)
Q3	(B)
Q4	(B)
Q5	(B)
Q6	(10)
Q7	(C)
Q8	(D)
Q9	(C)
Q10	52
Q11	(C)
Q12	93
Q13	6
Q14	(C)
Q15	(A, B, C, D)
Q16	(64)
Q17	(60000)
Q18	(D)
Q19	(D)
Q20	(B)

Hints & Solutions

Q1 Text Solution:

In the primary index, number of entries in the index block equals to number of blocks of relation.

Number of database records in a single block B = 4096/256 = 16

Number of blocks of relation C = 16384/16 = 1024

Size of indexes = size of key field

+ size of block pointer

= 22 + 10 = 32 bytes

Number of indexes records present in single block = 4096/32 = 128

Total number of blocks required to store primary index = 1024/128 = 8.

Q2 Text Solution:

In dense secondary index, number of entries in the index blocks equals to number of records of relation.

- Number of records in the relation P = 262144

Size of index = size of key field

+ size of record pointer

= 32 + 32 = 64 bytes

- Number of index entry in single block

= 4096/64 = 64

So, the total number of blocks required to store primary index = 262144/64 = 4096.

Q3 Text Solution:

The column on which condition gets applied should be considered for indexing.

$\therefore$ P_2 is the answer.

Q4 Text Solution:

Blocking factor (i.e number of records per block)

$= \frac{\text{Block size}}{\text{record size}}$

Record size of file = Sum of all field + additional bytes = 63 + 1 = 64

$\therefore$ Number of records per block = $\frac{2048}{64}$ = 32.

Q5 Text Solution:

(a) False: A database file can contain one clustered index because the database is sorted on one field only.

(b) True: A database file can consist of one clustered index and multiple secondary index.

(c) False: The index on a unique field on which database is sorted is primary index and there can be only one primary index.

(d) False: A database file can consist of either a primary or clustered index but not both.

Q6 Text Solution:

Block factor of database file = $\left\lfloor \frac{512}{100} \right\rfloor$

= 5 records/block

Number of blocks required to store 25,000 records

= $\left\lceil \frac{25000}{5} \right\rceil$

= 5000 blocks

Each cluster consumer 2 blocks

Number of entries in index file = $\frac{5000}{2}$ = 2500.

Block factor of index file = $\left\lfloor \frac{512}{15} \right\rfloor$ = 15 entries/block

Number of blocks in index file = $\left\lceil \frac{25000}{15} \right\rceil$ = 167

The number of block accesses in worst case

$$= \underbrace{\lceil \log_2 167 \rceil}_{\substack{\text{for index}\\ \text{file search}}} + \underbrace{1}_{\substack{\text{Accessing } 1^{st}\\ \text{block of cluster}}} + \underbrace{1}_{\substack{\text{Accessing } 2^{nd}\\ \text{block of cluster}}}$$

= 8 + 1 + 1 = 10

Q7 Text Solution:

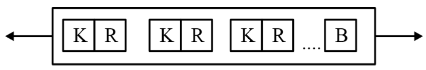
S_1: Records are ordered over non-key field. It is unordered over key field.

Hence, secondary index if formed over unordered key-field. Hence, its CORRECT.

S_2: CORRECT. More than one secondary index is possible.

Q8 Text Solution:

Disk block



Given data,

Disk block size = 1K byte = 2^{10} bytes = 1024 bytes

Block pointer (B) = 6 bytes

Key field (K) = 10 bytes

Record/ data pointer (R) = 8 bytes

Order of leaf node= P

B + (P)(K + R) $\leq$ D

6 + (P)(10 + 8) $\leq$ 1024

18 P $\leq$ 1024 – 6

$P = \left\lfloor \frac{1018}{18} \right\rfloor$

$\therefore$ P = 56

Maximum number of (value, data record pointer)

Pairs = 56

The order of the leaf node is 56.

Q9 Text Solution:

For storing the records, numbers of blocks required = $\frac{P}{3}$ and for storing the keys in dense index number of blocks required = $\frac{P}{10}$

So, total blocks required are $\frac{P}{3} + \frac{P}{10} = \frac{13P}{30}$

Q10 Text Solution:

Size of child pointer = 6 bytes

Size of search field value = 34 bytes

Block size = 2048.

Order of internal node = P

(Number of blocks pointer in any node)

(P –1)34 + P × 6 $\leq$ 2048

34P + 6P $\leq$ 2048 + 34

40P $\leq$ 2082

$P \leq \frac{2082}{40}$

$= \lfloor 52.05 \rfloor \simeq 52$

Q11 Text Solution:

Blocking factor, bfr = $\lfloor 1024/100 \rfloor$

= 10 records per block

Number of blocks needs for file = $\lceil \mathrm{r}/\mathrm{bfr} \rceil$

$= \lceil 100000/10 \rceil = 10000$

Index records size R_i= (Non – Key DeptID + P_R))

= 14 + 18 = 32 bytes

Index blocking factors bfri = $\lfloor \mathrm{B}/\mathrm{R_i} \rfloor = \lfloor 1024/32 \rfloor = 32$

Number of 1st level index entries r_1 = number of – records in the file = 100000 entries.

Number of first level index blocks b_1 = $\lceil \mathrm{r_1}/\mathrm{bfri} \rceil$

$= \lceil 100000/32 \rceil$

= 3125 blocks

Q12 Text Solution:

Order P: maximum blocks pointers per node.

Block size $\geq$ P × (Block size pointer) + (P –1) × (size of keys + size of record pointers)

Block size $\geq$ P ×5 + (P – 1) × (9 + 8)

2048 $\geq$ 5P + 17P – 17

22P$\geq$ 2065

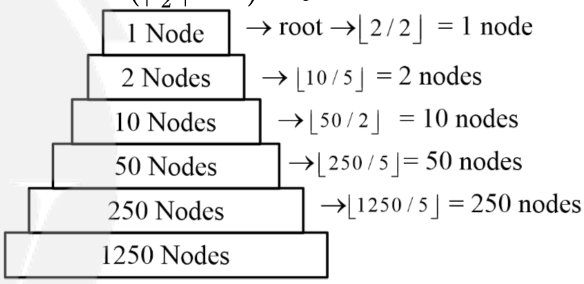
$\mathrm{P} = \left\lfloor \frac{2065}{22} \right\rfloor = 93$

Q13 Text Solution:

For maximum possible levels, minimum number of keys should be present in an index node.

Number of nodes in the last level = $\left\lfloor \frac{5000}{4} \right\rfloor$=1250

[Minimum $\left(\left\lceil \frac{9}{2} \right\rceil - 1\right)$ keys for other node]



Q14 Text Solution:

S_1(True): In a B^+ tree, data pointers are stored only at the leaf nodes of the tree.

S_2(True): the leaf nodes have an entry for every value of the search field, along with the data pointer to the record.

Q15 Text Solution:

True: Records can be fetched in equal number of accesses

True: Height of the tree remains balanced and less as compared to B tree.

True: We can access the data stored in a B^+ tree sequentially as well.

True: Faster search queries as the data is stored only on the leaf node.

Q16 Text Solution:

Number of indexes per blocks = $\frac{2048}{14+18}$ = 64.

Q17 Text Solution:

Number of first level index entries r_1 = Number of files records r = 60,000.

Q18 Text Solution:

Index records size R_i = PSN + PR = 18 + 14 = 32 bytes

Index blocking factor bfr_i = fan-out = $\lfloor B/R_i \rfloor$

= $\lfloor 1024/32 \rfloor$ = 32 index Records per block.

Number of 1st level index block b_1= $\lceil r_1/bfr_i \rceil$

= $\lceil 60{,}000/32 \rceil$

= 1875 blocks.

Q19 Text Solution:

Blocks size = 8 KB

Records size = 200 bytes

Number of records in a blocks = $\frac{8192}{200}$ = 40.96 records.

As it is spanned hence it takes whole as 40.96

1 block contains 40.96 records.

$\therefore 10^{12}$ in $\frac{10^6}{40.96}$ = 24415 blocks.

Q20 Text Solution:

S_1: (false): It is not possible because the requirement of sparse indexing is that the database must be stored and as we know that database is sorted only on one column.

S_2:(True): Any number of dense index is possible to construct because no requirement of sorting database in it.