# CS & DA

# Database Management System

## File Organisation & Indexing

**DPP 01 (Discussion Notes)**


By- Mili Dhara Ma'am

# [MCQ]

#Q. 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 _____?

# blocks of relation C = $\frac{16384}{16} = 1024$

# records in a block = $\frac{4096}{256} = 16$

Size of index = 22+10 = 32

# index record in single block = $\frac{4096}{32} = 128$

# blocks to store PI = $\frac{1024}{128} = 8$

A 7

B 8

C 9

D 10

# [NAT]

#Q. 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? 4096.

Size of index = 32+32 = 64

# index entry in a block = 4096/64 = 64

# entries in index block = # records = 262144

# blocks = 262144/64

# [MCQ]

#Q. 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 $P_3$

# [MCQ]

#Q. 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:

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

Block factor = Block size / Record size

= 2048 / 64 = 32

Σ size = 63 + 1 = 64

| Fields | Size (in Bytes) |
|---|---|
| EmpName | 5 |
| EmpNum | 10 |
| DeptNum | 9 |
| Addr | 20 |
| PhNum | 9 |
| DOB | 1 |
| Sex | 1 |
| Job | 3 |
| Sal | 5 |

# [MSQ]

#Q. 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 at most 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.

**[NAT]**

# block accesses = search in index file + 1st block + 2nd block
$= \lceil \log_2 74 \rceil + 1 + 1 = 7 + 1 + 1 = 9$

#Q. 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 _____.

Block factor of DB file $= \left\lfloor \frac{512}{100} \right\rfloor = 5$ records/block

# blocks to store 25000 records $= \frac{25000}{5} = 5000$ blocks

# entries in index file $= \frac{5000}{2} = 2500$

Block factor of index file $= \left\lfloor \frac{512}{15} \right\rfloor = 34$

# blocks in index file $= \lceil \frac{2500}{34} \rceil = 74$

# [MCQ]

#Q. 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 $S_2$

[MCQ]

#Q. 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 ?

P

$B + P(K+R) \leq 1024$

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

$\therefore$

$P = \left\lfloor \frac{1018}{18} \right\rfloor = 56$

A 53

B 54

C 55

D 56

**[NAT]**

#Q. 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 __52__.

$(p-1)34 + 6p \leq 2048$

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

$p = \lfloor 52.05 \rfloor = 52$

[MCQ]

#Q. 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 $\frac{P}{10}$

# blocks storing records = $\frac{P}{3}$

# blocks storing keys = $\frac{P}{10}$

Total no of blocks = $\frac{P}{3} + \frac{P}{10}$

$= \frac{13P}{30}$

[MCQ]

#Q. Assume a disk with block size B = 1024 Bytes, A block pointer is PB = 12 bytes long and a record pointer is PR = 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 Dept ID (14 bytes), then minimum of how many blocks are required to store index file assuming an unspanned organization?

BF : $\frac{1024}{100}$ = 10

A 3000

Index record size = 14 + 18 = 32 B

B 3100

# blocks : $\frac{100000}{10}$ = 10,000

C 3125

D None of the above

$BF_{ind} = \frac{1024}{32} = 32$

# 1st level index blocks = $\frac{10000}{32}$ = 3125

**[NAT]**

#Q. 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 .

$$\text{Block size} \geq 5p + (p-1)(8+9)$$

$$\Rightarrow 2048 \geq 5p + (p-1)\,17$$

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

[NAT]

#Q. 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).

$\lceil \frac{9}{2} \rceil = 5$

$10/5 = 2$

$50/5 = 10$

$\frac{250}{5} = 50$

$\lceil \frac{1250}{5} \rceil = 250$

last level $= \lceil \frac{5000}{4} \rceil = 1250$

[MCQ]

#Q. 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

[MSQ]

#Q. 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.

**[NAT]**

#Q. 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 __64__.

# indexes per block = $\frac{2048}{14+18} = 64$

[NAT]

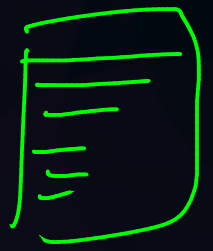
#Q. Data For This & Next Question:

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 ____.

60,000

1st level index file



[MCQ]

#Q. The number of first level index block is _____?

A 1800

B 1825

C 1850

D ~~1857~~ 1875

Index Record

Size = 18 + 14 = 32 B

$BF_{ind} = \left\lfloor \frac{1024}{32} \right\rfloor = 32$

# 1st level index block = $\frac{60000}{32}$

= 1875

[MCQ]

#Q. 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

# records in a block = $\frac{8KB}{200} = 40.96$

# blocks = $\frac{10^6}{40.96} = 24415$

[MCQ]

#Q. 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 $S_2$ are correct

D. Neither is $S_1$ and $S_2$ is correct

# THANK - YOU