Assignment

Subject:-
Data Base Systems

Topic:-

Submitted to:-
Mr. Samyan Qayyum Wahla

Submitted by:-
Muhammad Mudassir

Date of Submission:-
Feb 17, 2024

Problem 1:-

Question 1.1:-

To calculate the size of each row in bytes, we sum the size of each individual field.

Size of Review Date (Date): 3 bytes
Size of Academic Year (int32): 4 bytes
Size of Academic Quarter(char[10]): 10 bytes
Size of Course ID (char[5]): 5 bytes
Size of Rating(float32): 4 bytes
Size of Grade in the course(char[2]): 2 bytes
Size of Estimated Hours per week(int 32): 4 bytes
Size of Reveew (text) (char[224]): 224 bytes

By adding upper data-

$3+4+10+5+4+2+4+224 = 256$ bytes.

So, the size of each row in bytes is 256 bytes.

Question 1.2:-

Given:-

- Disk block size: 64 KB(64000 bytes)
- Size of each row: 256 bytes.

We divide the size of disk block by the size of each row to find out how many rows can be stored per disk block:

Number of rows per disk block = Disk block size / Size of each row.

= 64000/256

= 250

So, approximately 250 rows can be stored per disk block.

Question 1.3:-

Given

- DB block size = 64MB (64000KB) (64000000 bytes)
- Size of each row = 256 bytes.

We first convert DB block from MB to bytes.

DB block size = 64MB * 1000 KB/MB * 1000 bytes/KB

= 64000000 bytes.

Now, we divide the size of DB block by the size of each row to find out how many rows can be stored per DB block:

Number of rows per DB block = DB block size / Size of each row

= 64000000 bytes / 256 bytes

= 250,000

So, approximately 250,000 rows can be stored per DB block.

Question 1.4:-

Given:

- Number of class-taking students at Stanford: 15000
- Average number of classes per quarter per student: 3
- Percentage of students who write an evaluation: 50%

First, we calculate the number of reviews added per quarter.

Number of class-taking stu * Average number of classes per quar per stu = Total number of classes per-quarter.

15000 students * 3 classes = 45000 classes per quarter.

Next we calculate number of reviews per quarter:

Total number of classes per quarter * Percentage of students = Number of reviews added per quarter

45000 classes per quarter * 0.5 = 22500 reviews added per quarter.

Now, we calculate the total number of reviews added over 10 years (40 quarter):

Number of reviews added per quarter * Number of quarters = Total number of reviews added.

22500 reviews added per quarter * 40 quarters = 900,000 added over 10 years.

Finally, we calculate the size of table of course reviews in MB:

Total number of reviews added * Size of each review row / 1000000 = Size of table in MB.

900,000 reviews * 256 bytes /1000000 bytes per MB = 230.4

So after 10 years (40 quarters), the table of course reviews will be approximately 230.4MB in size.

Question 1.5:-

- Size of table of course reviews: 230.4MB
- Size of each DB block: 64MB (64000KB) (64000000 bytes)

First, we convert the size of table from MB to bytes
= 230.4MB * 1000 KB/MB * 1000 bytes/KB
= 230,400,000 bytes.

Now, we divide the total size of table by the size of each DB block to find out how many DB blocks are needed.

Number of DB blocks needed = Total size of table / Size of each DB block
= 230400000 bytes / 64000000 bytes
= 3.6 DB blocks.

We need to round up the nearest whole number which is 4 DB blocks.

Question 1.6:-

- Hard disk access (seek) time: 10ms
- Disk transfer speed: 100 MB/sec.
- Size of each row: 256 bytes.

We calculate seek and scan operations separately and then add them together.

1. Seek Time:

We assume the rows are randomly on disk, we calculate average seek time. Every seek operation has the same time cost, the average seek time is simply half of total seek time.

Average seek time = Total seek time / 2

Total seek time = Hard disk access (seek) time = 10ms

Avg(Seek Time) = 10/2 = 5ms

2. Scan Time:-

Scan time is time it takes to read data from the disk. We need to read entire row, so its time it takes to transfer from disk.

Size of each row = 256 bytes.

Disk transfer speed = 100MB/sec = 100000 KB/sec = 100000000 bytes/sec

Scan time = Size of each row / Disk transfer speed.

Scan time = 256 bytes/100,000,000 bytes/sec = 0.00000256 sec.

Now, we add the average seek and scan time.

Total time = 5ms + 0.00000256sec

Total time ≈ 0.005 sec.

Converting seconds to hours.

Total time ≈ 0.005 sec * (1 hour / 3600 sec)

Total time ≈ 0.0000013889 hours

Round to 1 decimal place

Total time ≈ 0.0 hours.

So, it would take 0.0 hours to retrieve row if table rows are stored randomly on disk.

Question 1.7:

- Hard disk access (seek) time = 10ms
- Disk transfer speed: 100MB/sec = 100000KB/sec = 100000000 bytes/sec
- Size of each DB block: 64MB = (64000KB) (64000000 bytes)

1. Seek time:-

Since disk blocks are randomly stored on disk, we need to calculate the average seek time.

Average Seek time = Total Seek time / 2

Total seek time = Hard disk access (seek) time = 10ms

Avg (Seek time) = 10/2 = 5ms

2. Scan time:-

Scan time is time it takes from the disk block.

Scan time = Size of each DB block / Disk transfer speed

Scan time = 64000,000 bytes / 100,000,000 bytes/sec

= 0.64 sec

Now add average seek and scan time

Total time = Average Seek time + Scan time.

Total time = 5ms + 0.64 sec

Converting ms to sec.

Total time = 0.005 sec + 0.64 sec = 0.645 sec

Rounded to 3 decimal places ≈ 0.645 sec.

So, it would take approximately 0.645 sec to retrieve row if rows are randomly stored on disk.

Problem 2:-

Question 2.1:-

- Number of rows : 5×10^7
- Row Size : 64KB
- Total data size : 3.2TB
- Hard disk space : 10TB

1. Calculate average seeking time:-

The average seeking time is half of the maximum time which goes from beginning to end of disk:

- Average (seeking time) = Maximum seek time/2

Maximum seek time = Hard disk access time = 10ms

= 10/2 = 5ms

2. Calculate scan timer

Scan time = Size of each row / Disk transfer speed.

Given:- • Size of each row = 64KB

• Disk transfer speed = 100MB/sec

Scan time = 64KB / 100MB/sec

Converting KB to bytes and MB to bytes/sec.

Scan time = $(64 * 1024 \text{ bytes}) / (100 * 10^6 \text{ bytes/sec}) \approx 0.00064\text{sec}$

3. Calculate total timer

Total time = Average seek time + Scan time

= 5ms + 0.00064sec

Converting milliseconds to seconds.
Total time ≈ 0.005 sec + 0.00064 sec ≈ 0.00564 sec
So, time to fetch a row is 0.00564 sec.

Question 2.2:-

Given information provided, it's evident that a small percentage of table rows are responsible for a significant portion of query traffic. To improve performance, I would suggest implementing a caching mechanism where the frequently queried rows from this 1% subset are stored in memory for faster access.

Here concise suggestion:
Implementing a caching mechanism to store frequently queried rows in RAM.
After implementing suggestion, the average response time significantly decreased because accessing data from RAM than accessing from disk.

Problem 3:

Question 3.1:

According to the fields given

1. UserID: Since there are 1 billion users, we need data type that can accommodate this range. An int data can store upto $2^{31}-1$ which is enough.
2. UserName: The longest string for user names containing 64 characters. We will use char[64] to store.
3. ItemID: Similar to UserID, we will use int datatype.
4. ItemName: Similar to User Name, we will use char[64] datatype.
5. TransactionID: We will use long data type to store trillion transactions.
6. Amount of Money (Rs): We will have decimal amount, then use double datatype.

- Size of User ID (int): 4 bytes
- Size of UserName (char[64]): 64 bytes
- Size of ItemID (int): 4 bytes
- Size of Item Name (char[64]): 64 bytes
- Size of Transaction ID (long): 8 bytes
- Size of Amount of Money (double): 8 bytes.

The size of each row is 152 bytes.

Question 3.2:-

The most appropriate data type for the User ID is int. This data type can efficiently accomodate the range of values expected for user IDs, which is approximately 2 billion with a size of 4 bytes.

Question 3.3:-

The most appropriate data type for the User Name column is char[64]. This data type allows for efficient storage of user names with an maximum length of 64 characters.

Question 3.4:-

The most appropriate data type for itemID column is int. Similar to the user ID column, an int data type can efficiently accomodate the range of values expected for item IDs which is upto 2 billion with a size of 4 bytes.

Question 3.5:-

The most appropriate datype for item Name column is char[64]. This type allows for efficient storage of item names with a maximum length of ~~length~~ 64 characters.

Question 3.6:-

The most appropriate data type of Transaction ID is long. A long data type can efficiently accomodate the range of values expected for transaction IDs, which is upto 9 quintillion, with a size of 8 bytes.

Question 3.7:-

The most appropriate data type for amount of money is double. A double data type can efficiently store decimal values representing money, allowing for flexiblity in handling various transaction, with a size of 8 bytes.

Question 3.8:-

Given: Size of each row: 4+64+4+64+8+8 = 152 bytes.

Number of transactions: 1 trillion.

Total data size = Size of each row * Number of transactions

= 152 bytes/transaction * 1000,000,000,000

= 152 000,000,000,000. bytes.

Converting bytes to TB:

1TB = 1099511627776 bytes.

Size of table in TB = Total data size / 1099511627776 bytes/T.

≈ 192.9 TB

Size of table in TB ≈ 192.9 TB.

Problem 4:-

Question 4.1:-

Given: • Size of the table: 200TB
• RAM Size: 64GB
• RAM Transfer speed: 100GB/sec
First, we need to convert size of table from TB to bytes:
200TB = 200×10^{12} bytes
Now, we calculate time it takes to transfer the entire table from RAM:
Time = Size of table / RAM transfer speed
Time = $(200 \times 10^{12}$ bytes$)/(100 \times 10^{9}$ bytes/sec$)$
Time = 2000 sec
The whole time is 2000 sec to read the entire table from RAM.

Question 4.2:-

Size of table = 200TB
Disk transfer speed = 100MB/sec
Size of each row = 152 bytes
Hard disk access (seek) time = 10ms
Total number of rows = Size of table / size of each row
= $(200 \times 10^{12}$ bytes$)/(152$ bytes/row$)$
T = 1.31579×10^{12} rows
Time to transfer each row from disk.

Time per row = Size of each row / Disk transfer speed
= 152 bytes / ($100 * 10^6$ bytes/sec)
= $1.52 * 10^{-6}$ sec

Next, we need to calculate the average seek time. Since each seek operation has same time, the average seek time is half of the total.

Avg(seek time) = Total seek time / 2

Maximum seek time = Hard disk access time = 10ms

Avg = 10/2 = 5ms = 5×10^{-3} sec

Now,

Total time = (time per row * Total no. of rows) + Avg(seek time)
= $(1.52 \times 10^{-6} * 1.31579 * 10^{12}) + 5 * 10^{-3}$ sec.
= $2 * 10^{6}$ seconds.

Finally, convert it to days
$\approx (2 * 10^{+6}) / (24 * 60 * 60) \approx 23$ days.
if each row stored randomly on disk.

Question 4.3:-

Given · Size of table: 200TB
· Size of each DB block: 64MB (64000KB) (64000000 bytes)
· Disk transfer speed: 100MB/sec.

Converting TB to bytes
$200TB = 200 * 10^{12}$ bytes.

Total number of DB blocks = Size of table / Size of each DB block
= $(200 * 10^{12}$ bytes) / 64000000 bytes/DB blocks

$= 3.125 * 10^6$ DB blocks.

Now

Time per DB block = Size of each DB block/Disk transfer speed

$= 64000000$ bytes/$(100 * 10^6$ bytes/sec$)$

$= 0.64$ sec.

Total time = Time per DB block * Number of DB blocks.

$\approx (0.64$ sec/DBblock$) * 3.125 * 10^6$ DB blocks.

$T \approx 2000000$ sec

Converting into days.

$\approx 2000000 / (24 \times 60 \times 60) \approx 23$ days.

It will approximately 23 days to read whole table is stored in DB blocks.

Question 4.4:-

- Size of table = 200TB
- Cost of RAM = \$6000/TB

First, find total cost of saving table in RAM.

Total cost = Size of table * Cost of RAM per TB.

$= 200 * 6000$

$\boxed{\text{Total cost} = \$1,200,000}$

Question 4.5:-

- Size of table = 200TB
- Cost of disk space = \$100/TB

Total cost = Size of table * Cost of disk space per TB

$= 200 * 100$

$= \$20{,}000$

Problem 5:

Question 5.1:-

There are many tables needed for e-commerce site hosting. Following two tables are necessary in database design-

1. User Table:-

This table will contain information of each us such as userID, username, email address, shipping address, payment information, etc

2. Product Table:-

This table will contain information such as product ID, product name, description, price, etc.

These tables are necessary for an e-commerce website.

Question 5.2:-

We first need to find the minimum number of bits required to represent 5 billion products.

For this, we need 5 billion unique IDs

The minimum number of bits can be

calculated as:

Num of bits = $\lceil \log_2(\text{Number of unique IDs}) \rceil$

$= \log_2[5 \times 10^9]$

$= \log_2(5) + 9 = \lceil 2.32 + 9 \rceil = \lceil 11.32 \rceil = 12$

So, to store each unique ID for each Product we need 12 bits.

Given this answer, datatype appropriate for product ID:

Smallint (2 bytes) as it can store values up to $2^{12} - 1 = 4095$ which is enough.

Question 5.3:-

To store 1 billion users, the unique IDs will be 1 billion.

Calculations.

Number of bits = $\log_2(\text{Number of unique IDs})$

$= \log_2(1 \times 10^9) = 9.97 \approx 10$

Given this answer, datatype appropriate is:

Smallint (2 bytes) as it can store values upto $2^{10} - 1 = 1023$ which is enough.

Question 5.4:-

Given

Order ID: int64 (8 bytes)

Product ID: smallint (2 bytes)

User ID: smallint (2 bytes)

- Quantity: int32 (4bytes)
- Timestamp: 4 bytes
- IP Address: 4 bytes
- Mailing Address: char[100]

By adding

$= 8+2+2+4+4+4+100 = 124$ bytes.

So one row of table is 124 bytes.

Question 5.5:-

- Number of orders received in a day = 100 million
- Number of days in a week = 7 day
- Size of one row = 124 bytes.

Total no. of orders in week = 100 * 7 days.

= 700 million orders.

Total size of table = 700 * 124

By converting in MB

Finally.

$= (700 * 124) / (1024 * 1024) \approx 85,067$ MB.

Question 5.6:-

- Size of table: 10GB.
- RAM access time = 20ns.

GB to bytes.

$= 10 * 1024 * 1024 * 1024$ bytes.

Number of records in RAM $= (10 \times 1024 \times 1024 \times 1024)/124$

≈ 86730769

Time = 20ns * 86730769.

$\approx$ 1.73461538 sec.

Seconds to milliseconds.

$\approx$ 1734.61538 ms

Question 5.7:-

- Size of table: 10GB.
- Hard disk access time: 10ms
- Disk transfer speed: 100MB/sec.

GB to bytes.

$= 10 \times 1024 \times 1024 \times 1024$

Number of records = $10 \times 1024 \times 1024 \times 1024 / 124$

Time to transfer one block = $124 \times$ Number of records / 1024 B

Finally calculate time.

Total time = Avg(seek time) + Time to transfer one block.

Total time in days = Total time in ms / $(1000 \times 60 \times 60 \times 24)$

$\approx$ 0.0127 days.

Question 5.8:-

- Size of each block: 64MB
- Average seek time: 10ms
- Disk transfer speed: 100MB/sec.

Time to transfer one block = $\frac{\text{Size of one block}}{\text{Disk transfer speed}}$

$= \frac{64 \times 1024 \times 1024 \times KB}{100} = \frac{60 \times 1024 \times 1024 \times 1024}{100} \approx 640$ ms

Total time = Time to transfer one block + Average seek time.

$\approx 640ms + 10ms$

$\approx 650ms.$

Question 5.9:-

· Original lookup time: 1098145 sec (as cal previous)

With 10 machines

$$\text{Lookup time per machine} = \frac{\text{Original lookup time}}{\text{Number of machines}}$$

$$= \frac{1098145}{10} \approx 109.8145 \text{ sec}$$

$$\text{Speedup factor} = \frac{\text{Original lookup time}}{\text{Lookup time with 10 machines}}$$

$$= \frac{1098145}{109.8145} \approx 10$$

Therefore with 10 machines, the lookup will be 10 times faster compared with single machine scenario.

Question 5.10:-

If the data was stored on another machine and that other machine had data readily available in RAM.

Given:-

Network transfer speed = 100GB/sec.

$$\text{Time} = \frac{\text{Data size}}{\text{Network transfer speed.}}$$

$$\text{Time} = \frac{10}{100}$$

$$\text{Time} = 0.1 \text{ sec.}$$

$$\text{Time} \approx 100 \text{ msec.}$$