Database Assignment

2022-CS-50

February 17, 2024

Problem 1

Question 1.1

What is the size of each row in bytes?

Question 1.2

Assume that this data is stored on a hard disk in disk blocks and the disk blocks are grouped in DB blocks. How many rows can be stored per disk block?

Question 1.3

How many rows can be stored per DB block?

Question 1.4

After 10 years (40 quarters), how large in MB will the table of course reviews be? Round your answer to 1 decimal place.

Question 1.5

How many DB blocks would be needed to store the table of course reviews?

Question 1.6

How long would it take in hours to retrieve an evaluation (row) if the table rows are stored randomly on disk?

Question 1.7

How long would it take in seconds if the rows are grouped in disk blocks (which are randomly stored on disk)?

Problem 2

Question 2.1

What is the average response time in secs for a query, i.e., time to fetch a row? Assume that all rows are equally likely to be queried.

Question 2.2

Would you suggest any change to the current architecture given this information? State your suggestion concisely. What's the average response time in secs after your suggestion?

Problem 3

Question 3.1

What is the size of each row in bytes?

Question 3.2

What is the most appropriate data type for the following column: User ID?

Question 3.3

What is the most appropriate data type for the following column: User Name?

Question 3.4

What is the most appropriate data type for the following column: Item ID?

Question 3.5

What is the most appropriate data type for the following column: Item Name?

Question 3.6

What is the most appropriate data type for the following column: Transaction ID?

Question 3.7

What is the most appropriate data type for the following column: Amount of money?

Question 3.8

What is the size of the table in TB?

Problem 4

Question 4.1

How long in seconds will it take to read the whole table from RAM?

Question 4.2

How long in days (round to nearest integer) will it take to read the whole table from disk if each row of the table is stored randomly in the disk?

Question 4.3

How long in days (round to nearest integer) will it take to read the whole table from disk if the table is stored in DB blocks?

Question 4.4

What is the cost in dollars for saving the table in RAM?

Question 4.5

What is the cost in dollars for saving the table in disk?

Problem 5

Question 5.1

What tables might you need for this?

Question 5.2

How big is one row of our table (one record) in bytes?

Question 5.3

How big is the entire table in MB if we assume that we store the data for a week, and we receive 100 million orders in a day?

Question 5.4

How much time does it take in milliseconds to look up a record if our table is in RAM?

Question 5.5

How much time does it take in days (round to nearest day) to look up a record on disk if all records are in random locations?

Question 5.6

How long would it take in seconds (round to nearest second) to look up a record then?

Question 5.7

If we had 10 machines, how would this impact the speed of looking up one record? How many times faster would look up be?

Question 5.8

What if the data was stored on another machine? How long would it in milliseconds take to get that data if that other machine had the data readily available in RAM?

Problem 1

Question 1.1

The size of each row in bytes is calculated by adding the sizes of each field:

Review Date:	3 bytes
Academic Year:	4 bytes (int32)
Academic Quarter:	10 bytes (char[10])
Course ID:	5 bytes (char[5])
Rating:	4 bytes (float32)
Grade in the course:	2 bytes (char[2])
Estimated Hours Per Week:	4 bytes (int32)
Review (text):	224 bytes (char[224])

Adding these up, we get a total of $3 + 4 + 10 + 5 + 4 + 2 + 4 + 224 = 256$ bytes.

Question 1.2

Given that disk blocks have a size of 64KB and the row size is 256 bytes, the number of rows that can be stored per disk block is:

$$\frac{64 \text{ KB}}{256 \text{ bytes}} = 256 \text{ rows}$$

Question 1.3

Since DB blocks have a size of 64MB and each block contains disk blocks, and each disk block can contain 256 rows, the number of rows per DB block is:

$$\frac{64 \text{ MB}}{64 \text{ KB} \times 256 \text{ rows}} = 256 \text{ rows}$$

Question 1.4

After 10 years (40 quarters), the number of evaluations submitted can be calculated as:

$$\begin{aligned}
&\text{Total number of students} = 15,000 \\
&\text{Number of evaluations per student per quarter} = 3 \\
&\text{Total evaluations per quarter} = 15,000 \times 3 = 45,000 \\
&\text{Total evaluations over 10 years} = 45,000 \times 40 = 1,800,000
\end{aligned}$$

Given that the size of each row is 256 bytes, the size of the table after 10 years will be:

$$1,800,000 \text{ evaluations} \times 256 \text{ bytes} = 460.8 \text{ MB}$$

Question 1.5

The number of DB blocks needed to store the table of course reviews is:

$$\frac{460.8 \text{ MB}}{64 \text{ MB}} = 7.2$$

Rounded up to the nearest integer, we need 8 DB blocks.

Question 1.6

To retrieve an evaluation (row) if the table rows are stored randomly on disk, we need to seek and scan every row. Given the seek time of 10ms and the time to transfer one row from disk, the total time to retrieve one row is approximately:

$$10 \text{ ms} + \frac{256 \text{ bytes}}{100 \text{ MB/sec}} = 10 \text{ ms} + 2.56\mu\text{s}$$

Question 1.7

If the rows are grouped in disk blocks (which are randomly stored on disk), the time to retrieve one row would be approximately equal to the seek time, which is 10ms.

Problem 2

Question 2.1

The average response time for a query, i.e., time to fetch a row, when all rows are equally likely to be queried, is approximately equal to the seek time for the hard disk, which is 10ms.

Question 2.2

Given that a fixed 1

Problem 3

Question 3.1

The size of each row in bytes can be calculated by determining the size of each column:

User ID:	4 bytes (int32)
User Name:	64 characters $\times$ 2 bytes per character
Item ID:	4 bytes (int32)
Item Name:	64 characters $\times$ 2 bytes per character
Transaction ID:	4 bytes (int32)
Amount of money:	8 bytes (double)

Adding these up, we get a total of $4 + (64 \times 2) + 4 + (64 \times 2) + 4 + 8 = 216$ bytes.

Question 3.2 to 3.7

The most appropriate data types for each column are:

User ID:	int32
User Name:	char[64]
Item ID:	int32
Item Name:	char[64]
Transaction ID:	int32
Amount of money:	double

Question 3.8

The size of the table in TB is:

$$\frac{1 \text{ trillion transactions} \times 216 \text{ bytes}}{1000^4 \text{ bytes per TB}} = 216 \text{ terabytes}$$

Problem 4

Question 4.1

The time it takes to read the whole table from RAM depends on the read speed of RAM.

Question 4.2

To read the whole table from disk if each row of the table is stored randomly, we need to consider the seek time, the time to transfer each row, and the total number of rows.

Question 4.3

To read the whole table from disk if the table is stored in DB blocks, it depends on the organization of the DB blocks and the seek time to access them.

Question 4.4

To calculate the cost for saving the table in RAM, we multiply the size of the table by the cost of RAM per TB.

Question 4.5

To calculate the cost for saving the table in disk, we multiply the size of the table by the cost of disk space per TB.

Problem 5

Question 5.1

Potential tables might include:

1. User table: To store user information such as username, email, password, etc.
2. Product table: To store product information such as product ID, name, price, description, etc.

Question 5.2

The size of one row of our table (one record) in bytes is calculated based on the specified fields.

Question 5.3

The size of the entire table in MB is calculated based on the data received over a week.

Question 5.4

To determine the size of one row (one record) in bytes, we need to consider the size of each field in the table.

From Question 5.2, we know the size of each field: - Order ID: int64 (8 bytes) - Product ID: variable (determined from Question 5.2) - User ID: variable (determined from Question 5.3) - Quantity: int32 (4 bytes) - Timestamp: 4 bytes - IP address: 4 bytes - Mailing address: char[100] (assuming 100 characters * 2 bytes per character for UTF-16 encoding)

Given the data types for Product ID and User ID are determined based on the number of products and users, respectively, let's assume: - Product ID: int32 (4 bytes) - User ID: int32 (4 bytes)

Now, we can calculate the size of one row (one record) in bytes by summing up the sizes of all the fields:

$$8 \text{ bytes (Order ID)} + 4 \text{ bytes (Product ID)} + 4 \text{ bytes (User ID)} + 4 \text{ bytes (Quantity)} + 4 \text{ bytes (Timestamp)} +$$

$$= 8 + 4 + 4 + 4 + 4 + 4 + 200 = 228 \text{ bytes}$$

So, one row of our table (one record) is 228 bytes in size.

Question 5.5

To calculate the size of the entire table in MB, we first need to find out how many records (rows) we'll have in total for a week.

Given that we receive 100 million orders in a day and we're storing the data for a week, the total number of orders for the week would be:

$$100 \text{ million orders/day} \times 7 \text{ days} = 700 \text{ million orders}$$

Since each order corresponds to one row in the table, and we know from Question 5.2 that the size of one row (one record) is 228 bytes, we can calculate the total size of the table in bytes:

$$700 \text{ million orders} \times 228 \text{ bytes/order}$$

$$= 159,600,000,000 \text{ bytes}$$

Now, to convert this to MB, we divide by 1024^2 (since 1 MB = 1024^2 bytes):

$$= \frac{159,600,000,000 \text{ bytes}}{1024^2} \text{ MB}$$

$$= 152.25 \text{ MB}$$

So, the entire table is approximately 152.25 MB in size.

Question 5.7

To determine the time it takes to look up a record on disk when all records are in random locations, we need to consider the seek time, the time to transfer the data, and the total number of records.

Given the information provided, we know: - Seek time: 10 milliseconds - Disk transfer speed: 100 MB/sec - Table size: 10 GB

First, we need to calculate the time it takes to transfer the entire table from disk to memory:

$$\text{Time to transfer} = \frac{\text{Table size}}{\text{Disk transfer speed}}$$

$$= \frac{10 \times 1024^3 \text{ bytes}}{100 \text{ MB/sec}}$$

$$= \frac{10 \times 1024^3}{100} \text{ seconds}$$

Next, we need to convert the time from seconds to days:

$$\text{Time in days} = \frac{\text{Time in seconds}}{86400}$$

$$= \frac{\frac{10 \times 1024^3}{100}}{86400} \text{ days}$$

$$\approx \frac{10 \times 1024^3}{100 \times 86400} \text{ days}$$

$$\approx 11.93 \text{ days}$$

Rounded to the nearest day, it takes approximately **12 days** to look up a record on disk if all records are in random locations.