

# W4111 – Introduction to Databases

## Sections 002, V002; spring 2022

### Homework 3a – Written Assignment

#### Instructions

- The homework submission date/time is 2022-APR-17 at 11:59 PM.
- Submission format is a PDF version of this document with your answers. Place your answers in the document after the questions.
- The name of your PDF must be <UNI>\_S22\_W4111\_HW3a\_Written.pdf. For example, mine would be dff9\_S22\_W4111\_HW3a\_Written.pdf
- You must use the Gradescope functions to mark the location of your questions/answers in the submitted PDF. Failure to mark pages will cause point deductions. **Please, please read the countless Ed posts, TA produced instructions and videos, etc. to prepare your submission.**
- You can use online sources but you must cite your sources. You may not cut and paste text.
- Questions typically require less than five sentences for an answer. You will lose points if your answer runs on and wanders.

“Verbosity wastes a portion of the reader’s or listener’s life.”

#### Questions

Question 1: Relative to disk drives, briefly define the following terms:

- Seek time
- Rotational latency time
- Transfer time/data transfer rate

Answer:

Seek time, is time it takes to reposition the arm over the correct track.

Rotational latency time is the time it takes for the sector to be accessed to appear under the head.

Transfer time is the time required to transfer the required amount of data.

Data transfer Rate is the rate at which data can be retrieved from or stored to the disk.

Question 2: Briefly explain logical block addressing and cylinder-sector-head addressing.

Answer:

Logical block addressing (LBA) is a scheme used for specifying the location of blocks of data stored on computer storage devices. It is a linear addressing scheme, only one number is used to address data, and each linear base address describes a single block.

cylinder-sector-head (CHS) addressing is the process of identifying individual sectors on a disk by their position in a track, where the track is determined by the head and cylinder numbers. Head is used to read and write data. Cylinder and sector numbers are indicators for where the data is located.

Question 3: Some databases only map data to/use outer sectors/cylinders of hard disk drive. Why?

Answer:

By doing so, seek time and rotation latency time could be minimized due to the properties of LBA mapping.

Question 4: Briefly explain the elevator algorithm for disk I/O scheduling and how it may improve performance?

Answer:

Elevator algorithm is a disk-scheduling algorithm to determine the motion of the disk's head in incoming read and write requests.

When a new request arrives while the drive is idle, the initial head movement will be in the direction of the cylinder where the data is stored.

When other requests arrive, requests are serviced only in the current direction of arm movement until the arm reaches the edge of the disk.

When this happens, the direction of the arm reverses, and the requests that were remaining in the opposite direction are serviced, and so on.

The advantage is that the drive could maintain a buffer of pending read and write requests.

Question 5: Relative to database buffer management, briefly explain the following concepts:

- Cache hit/cache miss
- Pinned page/block
- Least recently used replacement algorithm

Answer:

Cash hit is when a cache successfully finds the requested data, satisfying the search query.

Cache miss is when a cache doesn't have the requested data in its memory.

Pinned Blocks are the Blocks which are being accessed right at that moment. They are placed at most recently used (end of LRU list).

Least Recently Used (LRU) algorithm is an algorithm where the page to be replaced is least recently used. The idea is based on the concept that the least recently used page is not likely to be the page used in the next moment.

Question 6: Briefly explain the concepts of and differences between row-oriented storage and column-oriented storage. For each model, give an example of row access/query patterns that would benefit from the model.

Answer:

Column oriented database stores and retrieves data in columns and hence it can only be able to read the relevant data if required.

Row oriented database stores and retrieves data one row at a time and hence could read unnecessary data if some of the data in a row are required.

Records in row-oriented storage are easier to read and write than column-oriented storage.

`select sum(balance_amount) from table` is better for column-oriented storage.

`Select * from table` is better for row-oriented storage.

Question 7: Provide one benefit and one disadvantage of variable length record management/organization relative to fixed length record management/organization.

Answer:

Variable length record management is more memory efficient than fixed length record management.

Fixed length record management accesses records easier and faster than variable length record management.

Question 8: The sample database associated with the book has a *takes* relation/table – *takes*(*ID*, *course\_id*, *section\_id*, *semester*, *year*, *grade*) that associated students and courses/sections.

- How might table partitioning (storage) be beneficial for storing the rows in the table?
- What columns would the database use to partition the data?
- Why might the database partition the *takes* table but not partition the *student* and *section* tables?

Answer:

Table partitioning reduces costs of some operations such as free space management

ID column would be used to partition the data.

Because student and section cannot be further partitioned to make a reasonable relation/table

Question 10: Briefly explain and give 2 examples of how the algorithm for mapping records/rows to file/disks blocks can significantly affect performance.

Answer:

For spanned mapping, record is stored inside the block even if it can only be stored partially, thus the record is spanned over two blocks.

In spanned mapping, there will be no wastage of memory.

For unspanned mapping, record is stored inside the block only if it can be stored completely inside it.

So in unspanned mapping, access time of a record is less because for a single record only a single block needs to be accessed every time and hence, it is faster.