Yi-Ping Tseng
Uni: yt2690

1.1 An OS attempts to make sure that consecutive disk blocks store consecutive blocks of a particular file. Why is this practice especially important for magnetic disks?

Because the access time of magnetic disk includes seek time and rotational time, it makes the access time faster to store files in consecutive blocks. By doing so, the magnetic disk doesn't have to spend time on seeking different part of the file in disk.

1.2 A power failure that occurs while a disk block is being written could result in the block being only partially written. Assume that partially written blocks can be detected. An atomic block write is one where either the disk block is fully written or nothing is written (i.e., there are no partial writes). Suggest schemes for getting the effect of atomic block writes with the following RAID schemes.

RAID-1: since everything stores in two disks, we can check whether the written data are the same in both disks. If not, we delete them to make sure the rule of integrity.

RAID-5: we can use the error correction part of the data to make sure every quarters of the data were successfully stored. Otherwise, we should delete them to make sure the rult od atomic block write.

1.3 Suppose you have data that should not be lost on disk failure, and the application is write-intensive. How would you store the data? (Note: Your answer should consider differences between primarily sequential writes and primarily random writes with regard to RAID).

If the case is also read-intensive, I would choose RAID-5 with sequential access pattern. Since sequential access is more efficient in accessing data and RAID-5 splits the data into 4 part which makes the access process faster.

If the case is **not** read-intensive, I would choose RAID-1 with random access pattern because random access pattern makes it faster to access small pieces of data and storing data in RAID-1 reduces the searching space (from 4 to 1 compared with RAID-5).

2.1 Give benefits and disadvantages of variable length record management versus fixed length record management.

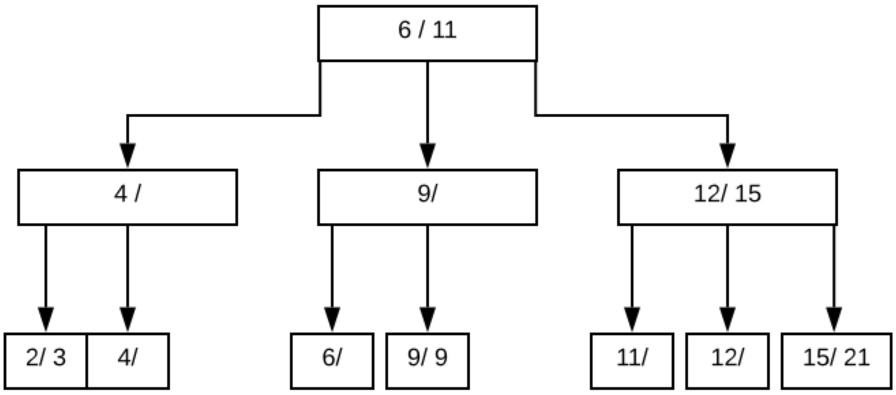
Advantages: no memory waste and allows different length of records. It is better when many small records with occasional long records.
Disadvantages: read and write is less efficient and needs extra time to manage the pointer.

2.2 In the sequential file organization, why is an overflow block used even if there is, at the moment, only one overflow record?

Overflow block is used to do quick insertion in sequential file organization.

2.3 Build and draw a B+ tree after inserting the following values. Assume the maximum degree of the B+ tree is 3. This cannot be hand-drawn
Values: 3, 11, 12, 9, 4, 6, 21, 9, 15, 2

The final B+ tree is as following.



2.4 What would the occupancy of each leaf node of a B+-tree be if index entries were inserted in sorted order? Explain why.

If it inserts in ascending order, the new entry will go to the last right leaf. When degree is 3, only the last right leaf will contain 2 elements because B+ tree need to maintain the maximum children restriction in each node. It is forced to split 3 elements into 1 element and 2 elements (the last right leaf).

If it inserts in descending order, the new entry will go to the first left leaf. When degree is 3 and the number of entries is odd, only the first left leaf contains 1 element. Otherwise, all leaf contains 2 elements because B+ tree will split 3 elements into 1 element node (the new entry) and 2 elements node to follow the restriction.

3.1 Briefly explain the following normalization concepts:
Functional dependency: Functional dependency (FD) is a set of constraints between two attributes in a relation. Functional dependency says that if two tuples have same values for attributes A1, A2,..., An, then those two tuples must have to have same values for attributes B1, B2, ..., Bn. (source: https://www.tutorialspoint.com/dbms/database_normalization.htm)

Data redundancy and update anomaly: it happens when few rows of data share the same information. If a column needs to update, the user needs to update at least 2 times or there will be inconsistent data in the database. If the user performing the update does not realize the data is stored redundantly the update will not be done properly. (source: https://databasemanagement.fandom.com/wiki/Category:Data_Anomalies)

Decomposition: Decomposition is the process of breaking down in parts or elements. It replaces a relation with a collection of smaller relations. It breaks the table into multiple tables in a database. It should always be lossless, because it confirms that the information in the original relation can be accurately reconstructed based on the decomposed relations. If there is no proper decomposition of the relation, then it may lead to problems like loss of information. (source: https://www.tutorialride.com/dbms/database-decomposition.htm)

BCNF:
This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:
R must be in 3rd Normal Form
and, for each functional dependency (X → Y), X should be a super Key.
(source: https://www.studytonight.com/dbms/database-normalization.php)

First Normal Form and atomic domain:
A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain.
First normal form enforces these criteria

1. Eliminate repeating groupsin individual tables
2. Create a separate table for each set of related data
3. Identify each set of related data with a primary key

(source: https://en.wikipedia.org/wiki/First_normal_form)

3.2 Assume (for simplicity in this exercise) that only one tuple fits in a block and memory holds at most three blocks. Show the runs created on each pass of the sort-merge algorithm when applied to sort the following tuples on the first attribute: (kangaroo, 17), (wallaby, 21), (emu, 1), (wombat, 13), (platypus, 3), (lion, 8), (warthog, 4), (zebra, 11), (meerkat, 6), (hyena, 9), (hornbill, 2), (baboon, 12).

4.1 Rewrite/transform the following query into an equivalent query that would be significantly more efficient.

4.2 Provide a two or three sentence definition of each of the following concepts, and provide an example:

Structured data: Structured data refers to any data that resides in a fixed field within a record or file. This includes data contained in relational databases and spreadsheets. Like ID in database table.

Semi-structured data: Semi-structured data is a cross between the structured and unstructured data. It is a type of structured data, but lacks the strict data model structure.

Unstructured data: Unstructured data is all those things that can't be so readily classified and fit into a neat box. Like photos. Like a email contains title, sender, recipient and date, but contains email message content and any attachments that are unstructured data. (Source for structured, unstructured and semi-structured data: https://www.webopedia.com/TERM/S/structured_data.html)

Metadata: it is "data that provides information about other data". In other words, it is "data about data." Many distinct types of metadata exist, including descriptive metadata, structural metadata, administrative metadata, reference metadata and statistical metadata. (source: https://en.wikipedia.org/wiki/Metadata)

BLOB: A Binary Large OBject (BLOB) is a collection of binary data stored as a single entity in a database management system. Blobs are typically images, audio or other multimedia objects, though sometimes binary executable code is stored as a blob. Database support for blobs is not universal. (source: https://en.wikipedia.org/wiki/Binary_large_object)