Nhat Nguyen
4-10-2024

1.
    a. There are three main alternatives of data entry k*:
        i. A data entry k* is an actual record
        ii. A data entry is a <k, rid> pair, where rid is the record id of a data record with search key k
        iii. A data entry is a <k, rid-list> pair, where rid-list is a list of record ids of data records with search key k
    b.
        i. Clustered index - when a file is organized so that the ordering of the data records is the same or as close to the ordering of the data entries in some index. They are relatively expensive to maintain when the file is updated.
        ii. Unclustered index - simply the reverse of clustered index: an index that is not clustered. To give more details, we can have multiple unclustered indexes on a data file. For example, if a Student records are sorted by age, any sorted-by-age index would be clustered, and any additional index (gpa for example) would be unclustered.
    c.
        i. A data file can be clustered on at most one search key - we can have at most one clustered index on a data file. There is no need to always create at least one clustered index because it is expensive to maintain and generally not needed for adequate query performance.
2.
    a. Seek time - the time taken to move disk heads to track on which a desired block is located
    b. Rotational delay - the waiting time for a desired block to rotate under the disk head; it is the time required for half a rotation on average and is usually less than seek time
    c. Transfer time - the time to actually read or write the data in the block once the head is positioned, or, the time the disk rotate over the block

3. First thing first, LRU or least-recently used is a buffer replacement policy - used to choose an unpinned page for replacement. Basically, sequential flooding is an access pattern in which the performance of LRU suffers when a set of pages S with its size larger than the buffer pool size is accessed multiple times repeatedly. For example, suppose that there is a file to be scanned that has 11 pages, while the buffer pool size of LRU is 10, every scan of the file using LRU will result in reading every page of that file, leading to a significantly worse off performance.
4.
    a. Fixed-length records - each field has a fixed length (value in each field has the same length) and fixed number of fields.
    b. Variable-length records - a record is of variable length only because some of its fields are of variable length
    c. Trade-offs
        i. For fixed length records - the fields can be stored consecutively and the address of a field can be easily calculated. Overall easy to implement.
        ii. For variable length records - records can be stored with an array of offset in different ways, making them fairly flexible in exchange for some subtle issues raised by this flexibility (growing field and modified records may no longer fit into the space of its page, no longer fit on any one page, etc.)

5. In my opinion, it's because of the nature of pin_count itself. Basically, pin count is the number of times that the page currently has been requested in a given frame, or simply, the current number of users on the page. Based on this definition, a pin_count will be able to keep track of this "number of users" that requested the page through the incrementation of itself (pinning). A pin_flag wouldn't have been able to do so and would be very generalized in terms of depicting user's count. In addition, a pin_count can help achieve page replacement when the buffer pool is full or no requested page is available by signifying a page that is unpinned (when pin_count = 0), a functionality that pin_flag cannot accomplish as it can only show whether there are users requesting the page.