

1 True and False

- (a) When querying for a 16 byte record, exactly 16 bytes of data is read from disk.
- (b) Writing to an SSD drive is more costly than reading from an SSD drive.
- (c) In a heap file, all pages must be filled to capacity except the last page.
- (d) Assuming integers take 4 bytes and pointers take 4 bytes, a slot directory that is 512 bytes can address 64 records in a page.
- (e) In a page containing fixed-length records with no nullable fields, the size of the bitmap never changes.

Which of the following are true about the benefits of using a record header for variable length records?

- (a) Does not need a delimiter character to separate fields in the records
- (b) Always matches or beats space cost when compared to fixed-length record format
- (c) Can access any field without scanning the entire record
- (d) Has compact representation of null values

2 Fragmentation And Record Formats

- (a) Is fragmentation an issue with packed fixed length record page format?
- (b) Is fragmentation an issue with variable length records on a slotted page?
- (c) We usually use bitmaps for pages with fixed-length records. Why not just use a slotted page for pages with fixed-length records?

3 Record Formats

Assume we have a table that looks like this:

```
CREATE TABLE Questions (  
    qid integer PRIMARY KEY,  
    answer integer,  
    qtext text,  
);
```

Recall that integers and pointers are 4 bytes long. Assume for this question that the record header stores pointers to all of the variable length fields (but that is all that is in the record header).

- (a) How many bytes will the smallest possible record be?
- (b) Now assume each field is nullable so we add a bitmap to the beginning of our record header indicating whether or not each field is null. Assume this bitmap is padded so that it takes up a whole number of bytes (i.e. if the bitmap is 10 bits it will take up 2 full bytes). How big is the largest possible record assuming that the qtext is null?

4 Calculate the IOs with Linked List Implementation

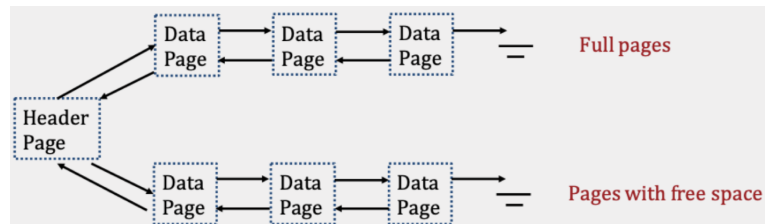


Figure 1: Linked List Implementation

Assume we have a heap file A implemented with a linked list and heap file A has 5 full pages and 2 pages with free space, at least one of which has enough space to fit a record.

- (a) In the worst case, how many IOs are required to find a page with enough free space to insert a record?
- (b) In the worst case, how many IOs are required to write a record to the 2nd page with free space? Consider what happens when after writing, the page becomes full and assume that the header page can insert at the beginning of the full pages linked list.

5 Calculate the IOs with Page Directory Implementation

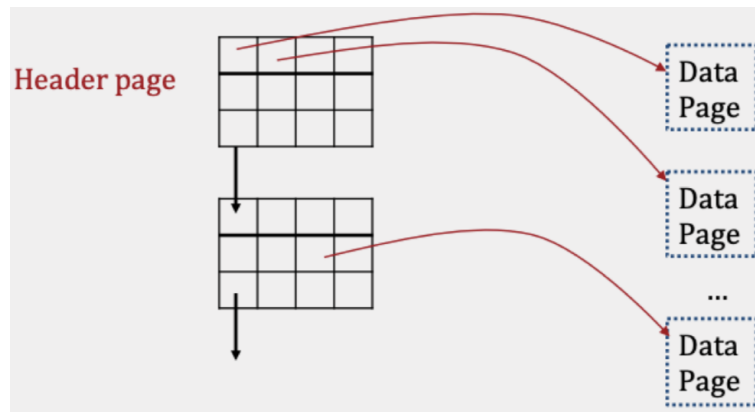


Figure 2: Page Directory Implementation

Assume we have a heap file B implemented with a page directory. One page in the directory can hold 16 page entries. There are 54 pages in file A in total.

- (a) In the worst case, how many IOs are required to find a page with free space?
- (b) In the worst case, how many IOs are required to write a record to a page with free space (assuming at least one free page with enough space to insert a record exists)?