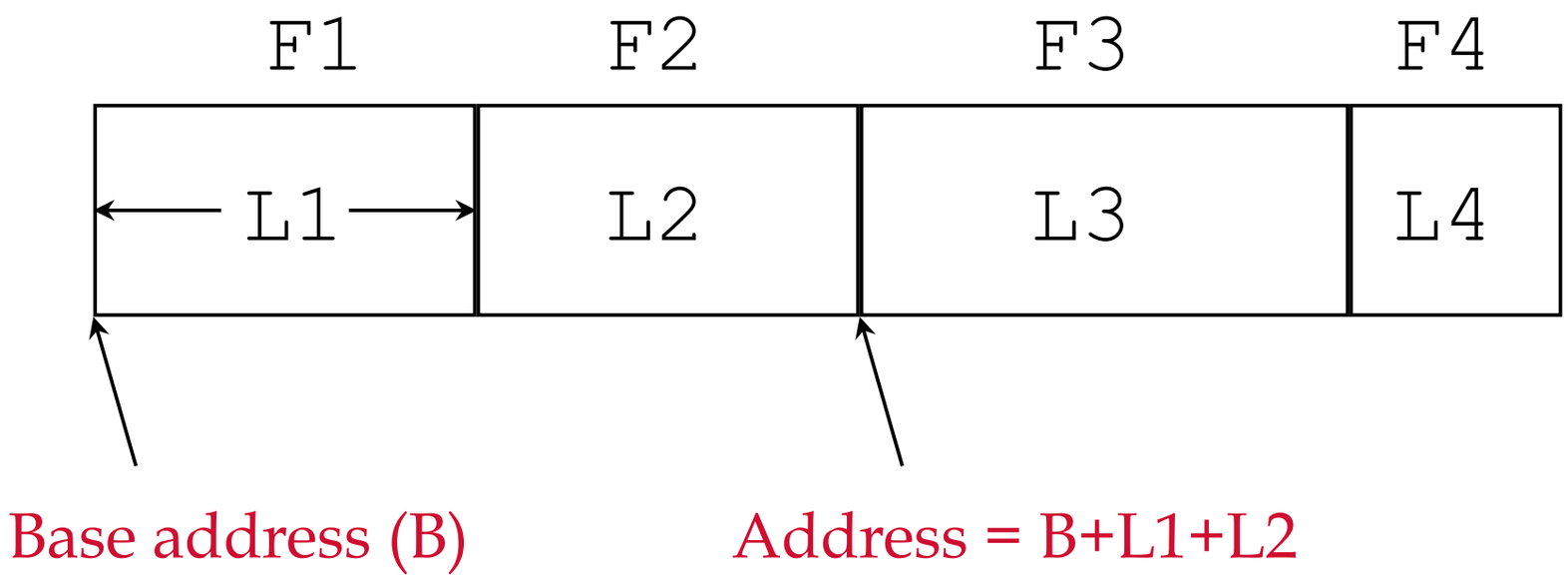
COSC 460 Lecture 3: Data Storage

Professor Michael Hay
Fall 2018

Credits: Slides adapted from Gehrke

Fixed-length record format

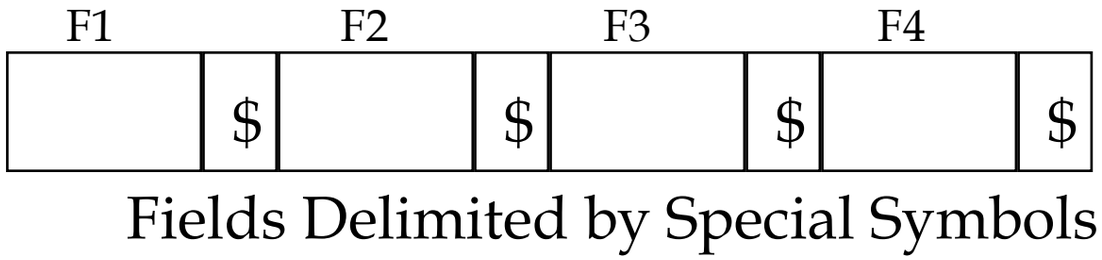


- Typically, all records in same file have same *schema*
- Information about schema stored in **System Catalog**
- To access i^{th} field, use arithmetic

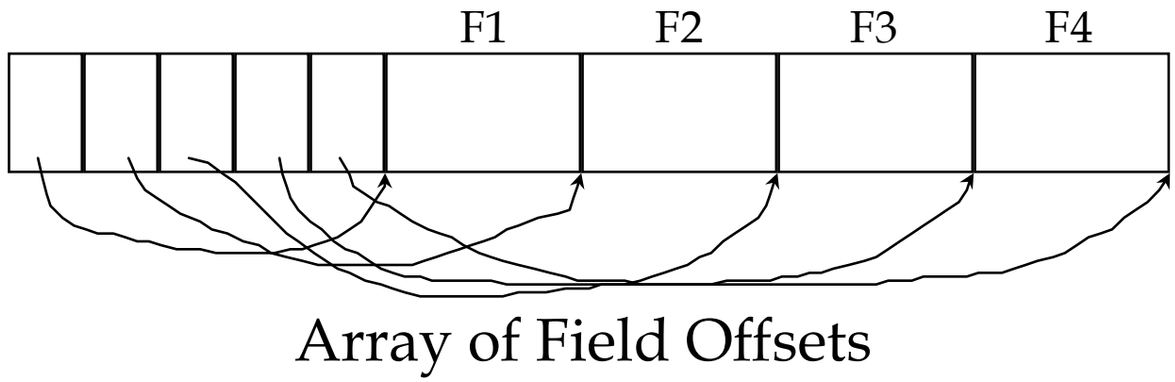
Variable-length record format

- Two alternative formats

Delimiter approach



Offsets approach



- Comparison?

pollev.com/cosc460

Correct answer: B.

Poll: free space

Suppose page is as drawn on board. The space marked "unused" is 30 bytes and the space marked "free" is 12 bytes. What is the largest variable-length record you can add…. (1) *without* compacting and (2) *with* compacting?

A. 12 w/o compacting; 30 w/ compacting

B. 12 w/o compacting; 42 w/ compacting

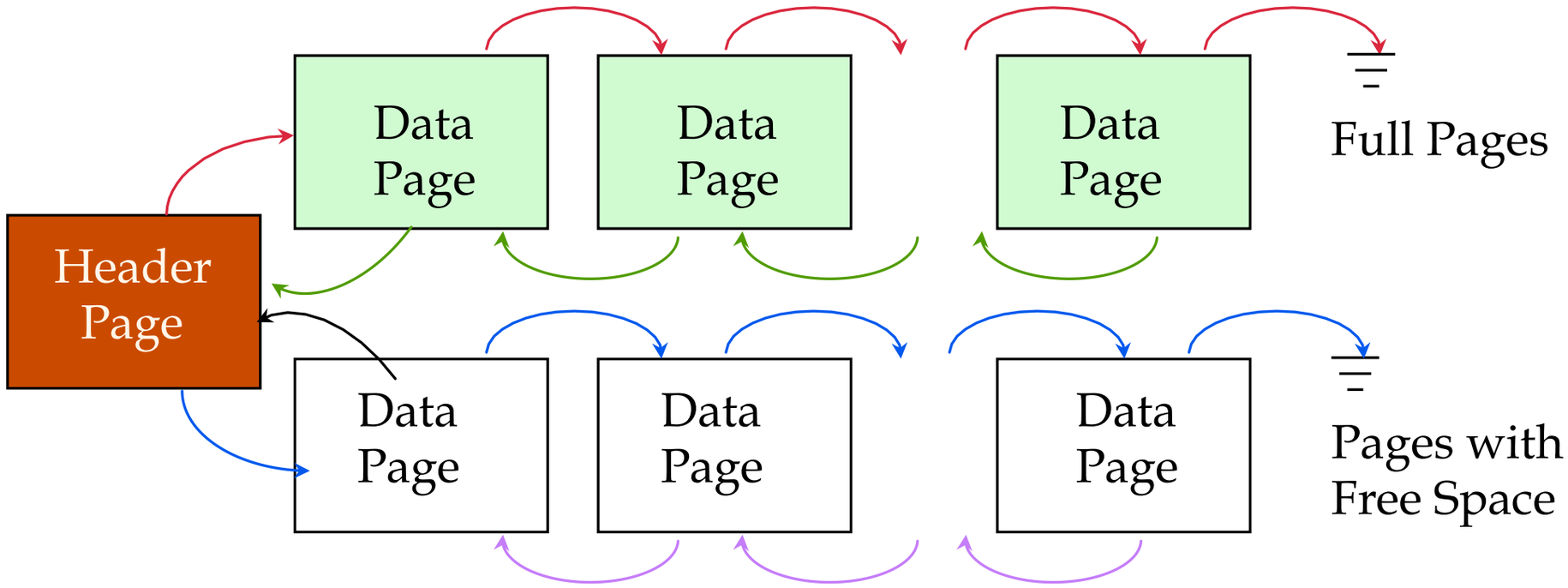
C. 30 w/o compacting; 30 w/ compacting

D. 30 w/o compacting; 42 w/ compacting

E. None of above

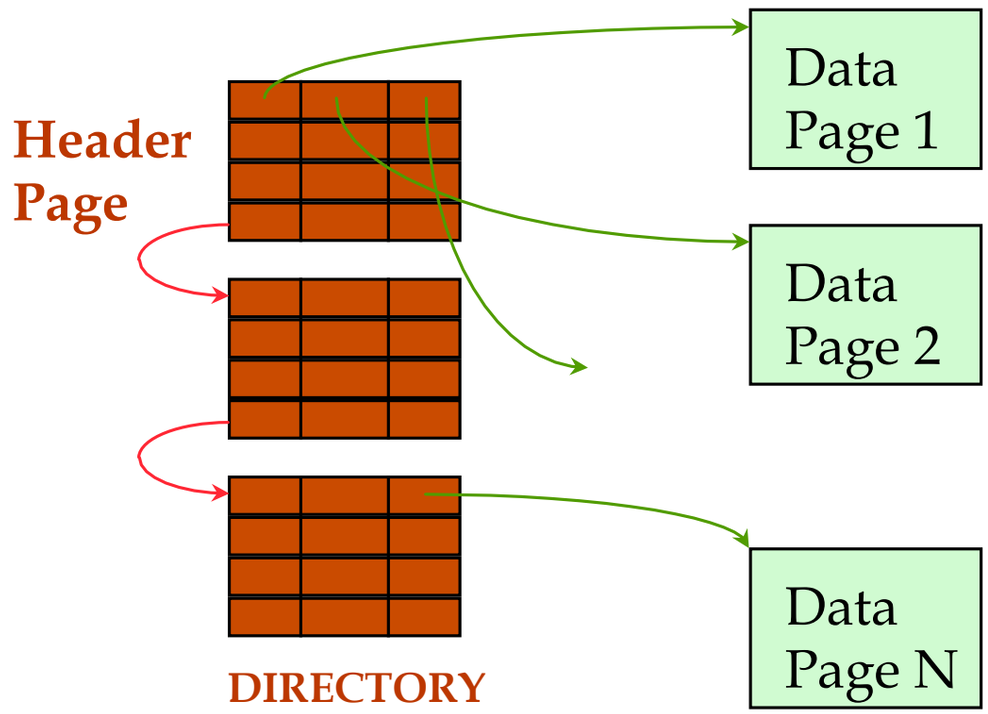
Instructions: *I will give you 1-2 minutes to think on your own.*
Vote 1.
Then you will discuss w/ neighbor (1 min).
Vote 2.
Then we'll discuss as class.

Heap File Implemented as List



- Page id of header page stored in System Catalog
- Page format: requires space for 2 “pointers” (page ids)

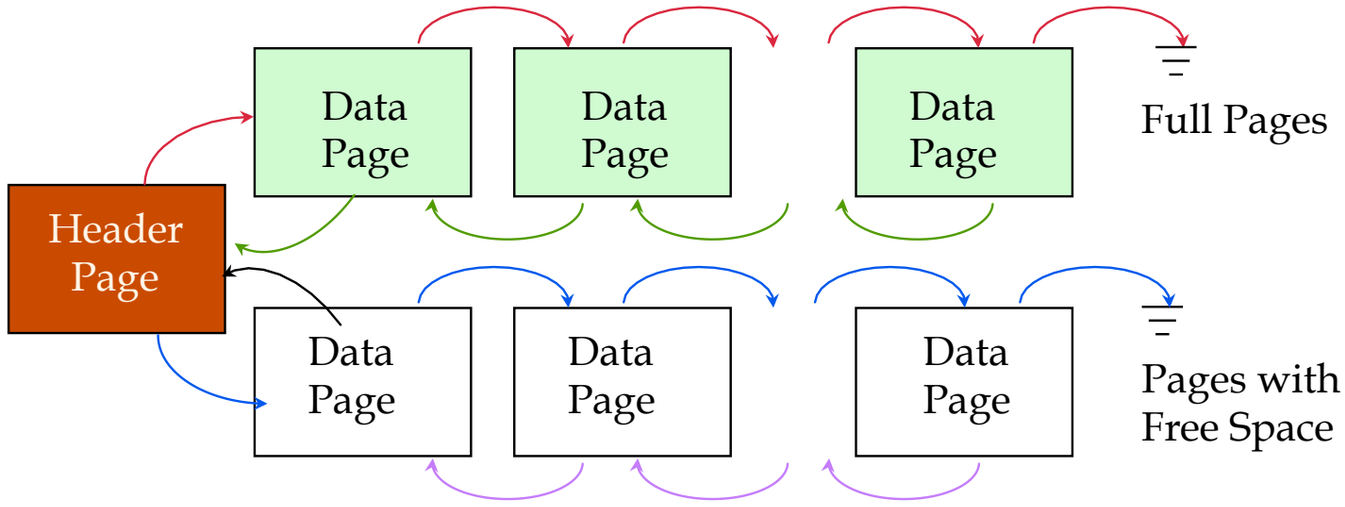
Heap File Using Page Directory



- Page id of *first* directory page stored in System Catalog
- Directory page format: directory entries <page id, # free bytes>, plus “pointer” (page id) for next directory page

Correct answer: D.

Poll: heap



Suppose you have *variable-length records* and you implement Heap File using the *linked list approach*. Assume buffer pool is empty and the Heap File has N pages. To insert a tuple, how many pages must be read from disk? Consider the *best*- and *worst*-case possibilities:

A. Best: 1 page; Worst: 2 pages

B. Best: 1 page; Worst: N pages

C. Best: 2 pages; Worst: N/2 pages

D. Best: 2 pages; Worst: N pages

E. Best: N/2 pages; Worst: N pages

Instructions: *I will give you 1-2 minutes to think on your own.*
Vote 1.
Then you will discuss w/ neighbor (1 min).
Vote 2.
Then we'll discuss as class.

Heap File: List vs. Directory

- Linked list approach
  - Simple to implement
  - Efficient for fixed-length records: header + first free page
- Directory
  - More complex to implement (linked list of header pages)
  - Better support for variable-length records: directory can report *available space* on each free page.
  - Must keep directory data up to date

Alternative format: store data by "column" rather than by "row"

Columnar storage especially good for ***data analytics***

C-Store: A Column-oriented DBMS

Mike Stonebraker*, Daniel J. Abadi*, Adam Batkin+, Xuedong Chen†, Mitch Cherniack+, Miguel Ferreira*, Edmond Lau*, Amerson Lin*, Sam Madden*, Elizabeth O'Neil†, Pat O'Neil†, Alex Rasin‡, Nga Tran+, Stan Zdonik‡

*MIT CSAIL Cambridge, MA
+Brandeis University Waltham, MA
†UMass Boston Boston, MA
‡Brown University Providence, RI

Abstract

This paper presents the design of a read-optimized relational DBMS that contrasts sharply with most current systems, which are write-optimized. In which periodically a bulk load of new data is performed, followed by a relatively long period of ad-hoc queries. Other read-mostly applications include customer relationship management (CRM) systems, electronic library card catalogs, and other ad-hoc inquiry systems. In

Column-Stores vs. Row-Stores: How Different Are They Really?

Daniel J. Abadi
Yale University
New Haven, CT, USA
dna@cs.yale.edu

Samuel R. Madden
MIT
Cambridge, MA, USA
madden@csail.mit.edu

Nabil Hachem
AvantGarde Consulting, LLC
Shrewsbury, MA, USA
nhachem@agdba.com

ABSTRACT

There has been a significant amount of excitement and recent work on column-oriented database systems ("column-stores"). These database systems have been shown to perform more than an order of magnitude better than traditional row-oriented database systems ("row-stores") on analytical workloads such as those found in data warehouses, decision support, and business intelligence applications. The elevator pitch behind this performance difference is straightforward: column-stores are more I/O efficient for read-only queries since they only have to read from disk (or from memory) those attributes accessed by a query.

This simplistic view leads to the assumption that one can obtain the performance benefits of a column-store using a row-store:

General Terms

Experimentation, Performance, Measurement

Keywords

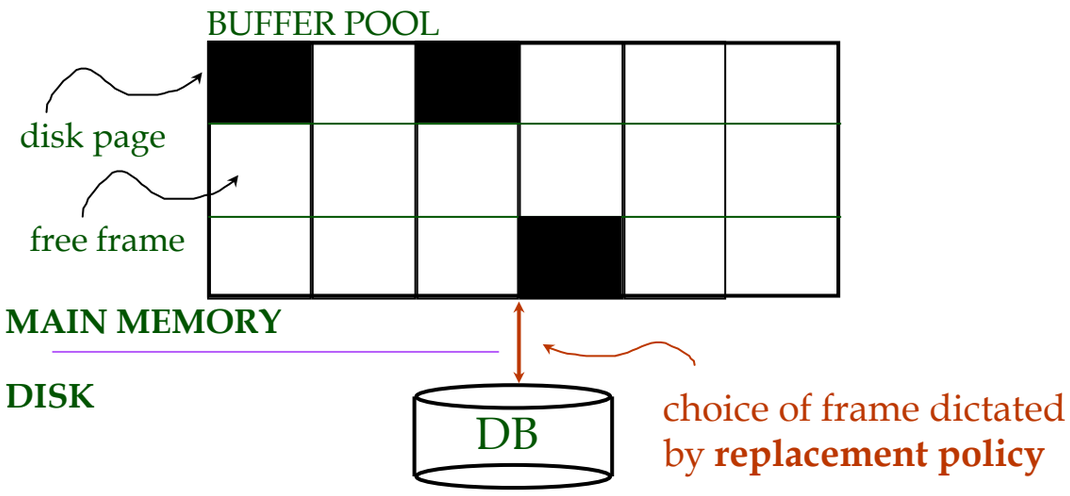
C-Store, column-store, column-oriented DBMS, invisible join, compression, tuple reconstruction, tuple materialization.

1. INTRODUCTION

Recent years have seen the introduction of a number of column-oriented database systems, including MonetDB [9, 10] and C-Store [22]. The authors of these systems claim that their approach offers order-

Correct answer: A.

Poll: clock



Which of the following is true about the clock replacement approach? In answers below, "recently used" means used at some point after the last page eviction.

A. Recently used pages have ref bit set to 1

B. It never chooses most recently used page

C. A and B

D. None of above

Instructions: *I will give you 1-2 minutes to think on your own.*
Vote 1.
Then you will discuss w/ neighbor (1 min).
Vote 2.
Then we'll discuss as class.

Architecture of DBMS