香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF
DATA SCIENCE
數據科學學院

CSC3170
Tutorial 3

School of Data Science

The Chinese University of Hong Kong, Shenzhen

Practice

- Given a heap file implemented as a Page Directory, what is the I/O cost to insert a record in the worst case?

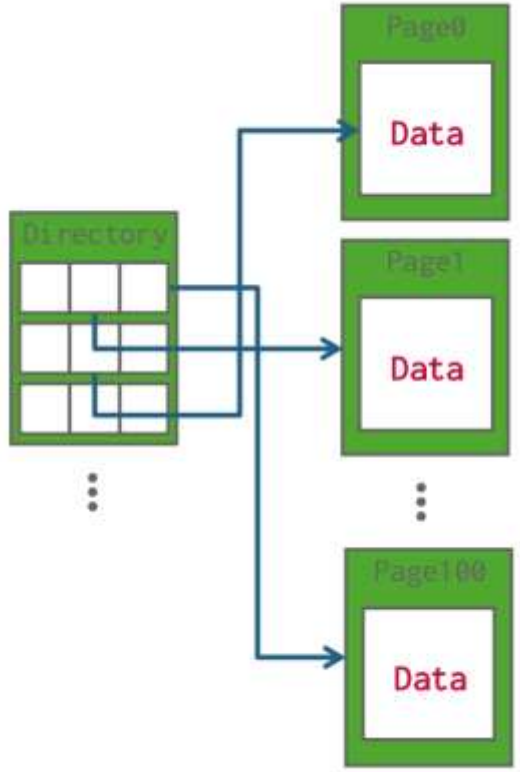
  The directory contains 4 header pages (directory pages) and 8 data pages for each header page.

  Assume that at least one data page has enough space to fit the record, and the cost of each read/write operation is 1 I/O.

Heap File: Page Directory

- The DBMS maintains special pages that tracks the location of data pages in the database files.

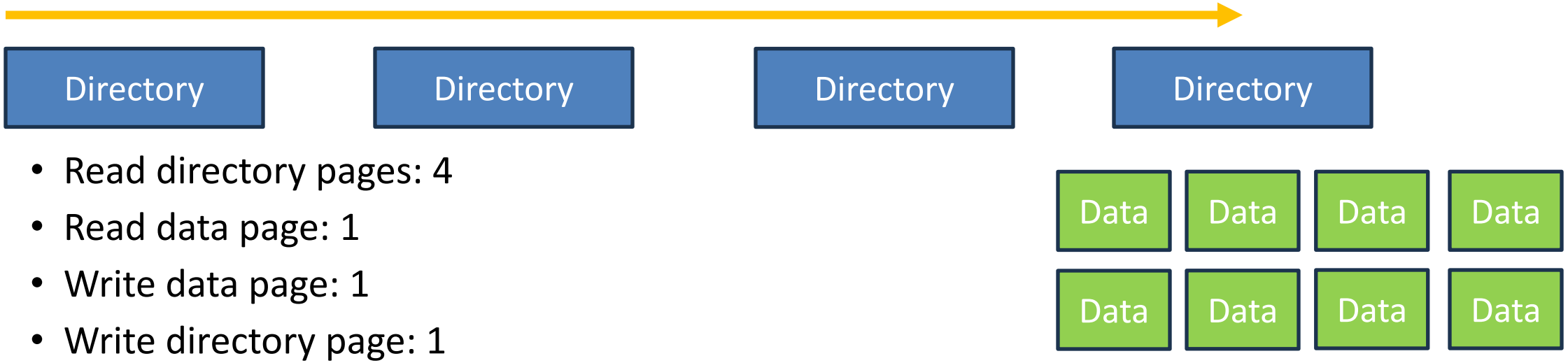
- The directory also records meta-data about available space:
  - The number of free slots per page.
  - List of free / empty pages.



Practice

- Given a heap file implemented as a Page Directory, what is the I/O cost to insert a record in the worst case?

➢ In the worst case, the only data page with enough free space is on the very last directory page.



- Read directory pages: 4
- Read data page: 1
- Write data page: 1
- Write directory page: 1
- In total: 4 + 1 + 1 + 1 = 7 I/O

Practice

- What is the smallest size, in bytes, of a record from the following schema in a slotted page? Assume that the record header is 5 bytes, boolean = 1 byte, date = 8 bytes, disregarding word-alignment.

```
name VARCHAR

student BOOLEAN

birthday DATE

state VARCHAR
```

- What is the maximum number of records that can be stored on a 1 KB page given the schema above? Assume that slot count, free space pointer, record pointer and record length costs 4B each.

Pages with Variable Length Records

- The main difference between variable length records and fixed length records is that we no longer have a guarantee on the size of each record. To work around this, each page uses a **page footer** that maintains a **slot directory** tracking **slot count**, **a free space pointer**, and **entries**. The footer starts from the bottom of the page rather than the top so that the slot directory has room to grow when records are inserted.

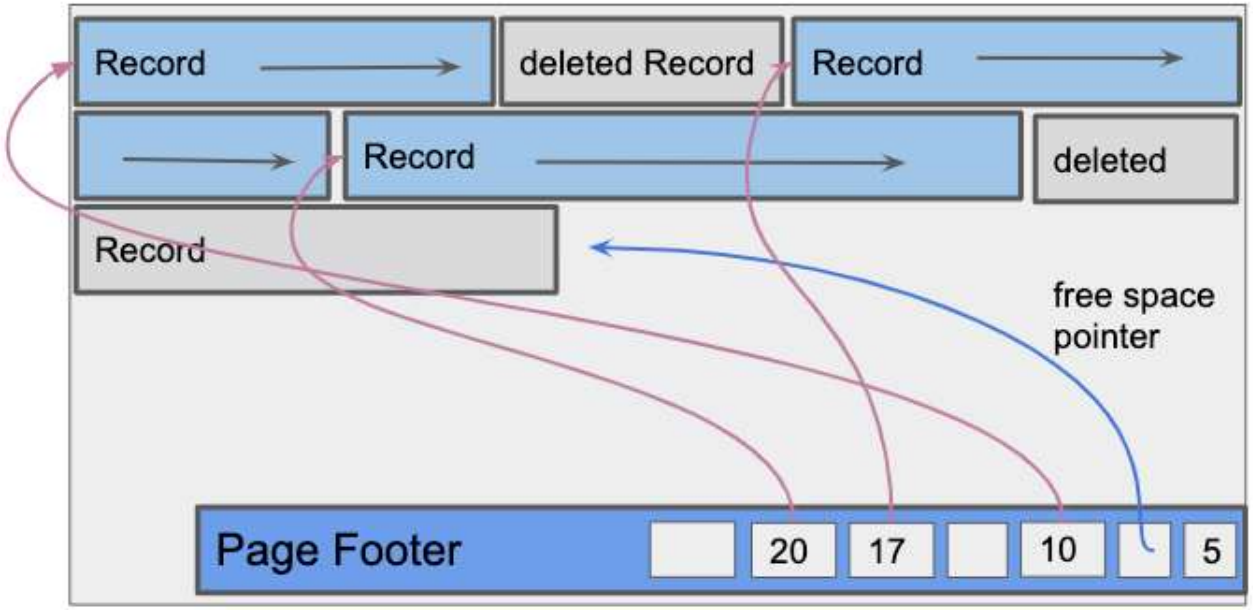
- The slot count tracks the total number of slots. This includes both filled and empty slots. The free space pointer points to the next free position within the page. Each entry in the slot directory consists of a **[record pointer, record length]** pair.

- If the page is unpacked, deletion involves finding the record's entry within the slot directory and setting both the record pointer and record length to null.

- Reference: ucb cs186

Pages with Variable Length Records

- A slot directory page
  = (record + length + pointer) * n
  + slot count
  + free space pointer



Reference: ucb cs186

Practice

- What is the smallest size, in bytes, of a record from the following schema in a slotted page? Assume that the record header is 5 bytes, boolean = 1 byte, date = 8 bytes.

```
name VARCHAR
student BOOLEAN
birthday DATE
state VARCHAR
```

- VARCHAR: minimum 0B
- A record: 0 + 1 + 8 + 0 + 5 = 14 Bytes

Practice

- What is the maximum number of records that can be stored on a 1 KB page given the schema above? Assume that slot count, free space pointer, record pointer and record length costs 4B each.

- A slot directory page
  = (record + length + pointer) * n
  + slot count
  + free space pointer

- 1 KB = 1024 B
  = (14 B + 4 B + 4 B) * n
  + 4 B
  + 4 B

n = floor [(1024 – 8) / (14 + 8)]

Practice

- What is the maximum size, in bytes, of a record from the following schema? Assume that the record header is 5 bytes, boolean = 1 byte, date = 8 bytes.

```
name VARCHAR(12)

student BOOLEAN

birthday DATE

state VARCHAR(2)
```

Practice

- What is the maximum size, in bytes, of a record from the following schema? Assume that the record header is 5 bytes, boolean = 1 byte, date = 8 bytes.

```
name VARCHAR(12)

student BOOLEAN

birthday DATE

state VARCHAR(2)
```

- VARCHAR: minimum 12B
- A record: 12 + 1 + 8 + 2 + 5 = 28 Bytes

Practice

- Given a database schema with an attribute student of type *INT32* and a total of 64 tuples, calculate the amount of storage space required to represent NULL values using the following three methods:
  - Null Column Bitmap Header: Store a bitmap in a centralized header that specifies which attributes are null.
  - Special Values: Designate a special value to represent NULL for a data type (e.g., INT32_MIN).
  - Per Attribute Null Flag: Store a flag that marks whether a value is null.
- For each method, determine the storage space required to handle NULL values across all 64 tuples.

NULL Data Types

- **Choice #1: Null Column Bitmap Header**

→ Store a bitmap in a centralized header that specifies what attributes are null.

- This is the most common approach.

- **Choice #2: Special Values**

→ Designate a value to represent `NULL` for a data type (e.g., `INT32_MIN`).

- **Choice #3: Per Attribute Null Flag**

→ Store a flag that marks that a value is null.

→ Must use more space than just a single bit because this messes up with word alignment.

Practice

- Given a database schema with an attribute student of type *INT32* and a total of 64 tuples, calculate the amount of storage space required to represent NULL values. For each method, determine the storage space required to handle NULL values across all 64 tuples.

  - Null Column Bitmap Header: 1 bit * 64 = 64 bits = 8 Bytes
  - Special Values: no extra space needed.
  - Per Attribute Null Flag: 1 Byte * 64 = 64 Bytes

Q&A

Thanks to the previous CSC3170 teaching team from which part of the content was sourced.