CS222/CS122C: Principles of Data Management

UCI, Fall 2019
Lecture #02

Storing Data: Record/Page Formats

Instructor: Chen Li

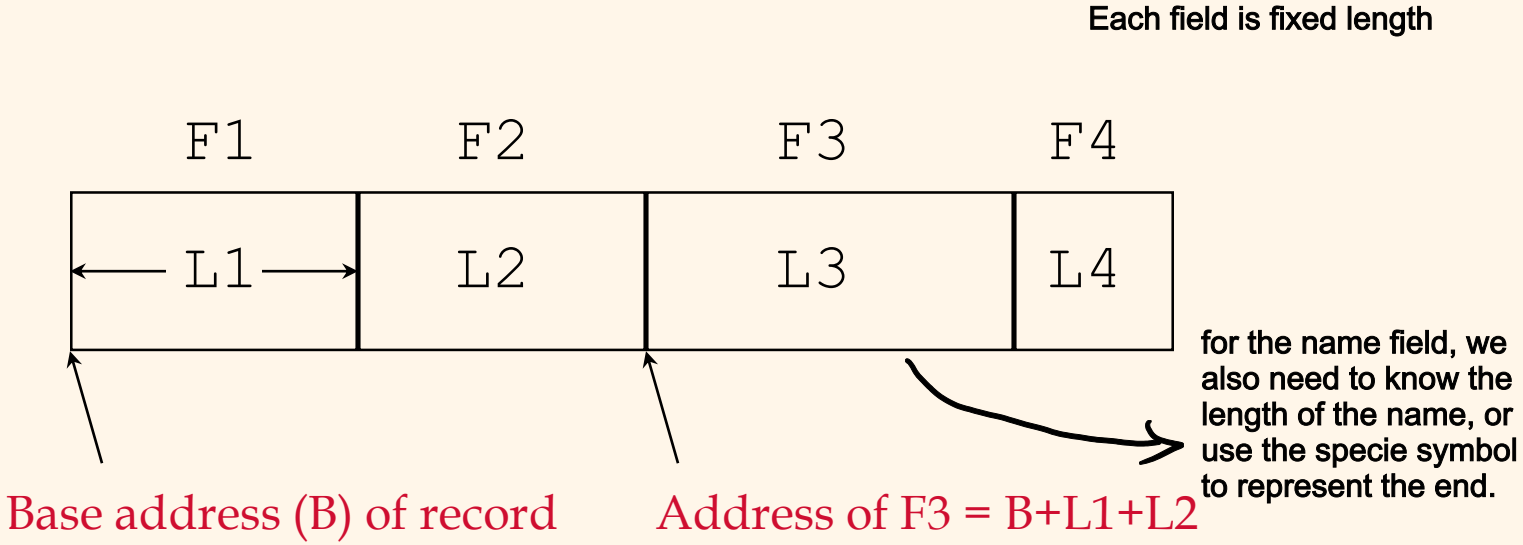
Files of Records

- Page or block is OK when doing I/O, but higher levels of DBMS operate on *records*, and thus want *files of records*.
- FILE: A collection of pages, each containing a collection of records. Must support:
  - Insert (append)/delete/modify record
  - Read a particular record (specified using *record id*)
  - Scan all records (possibly with some conditions on the records to be retrieved)

Example

```
CREATE TABLE Emp(id INT,
gender CHAR(1),
name VARCHAR(30),
Salary float
);
```

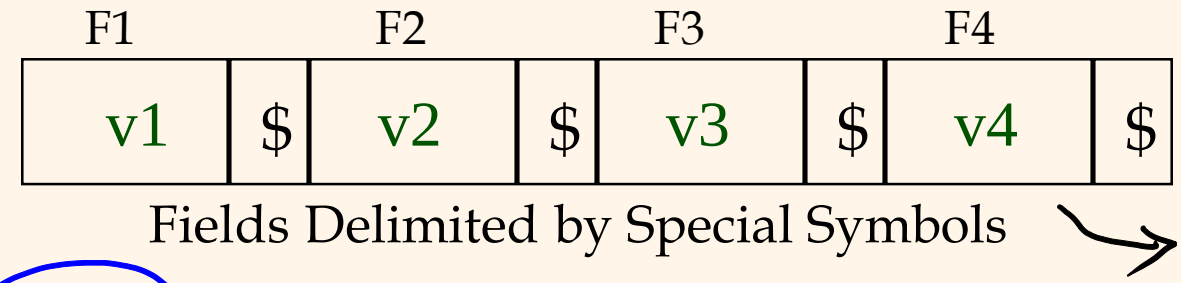
Record Formats: Fixed Length



- Information about field types is the same for all records in file; it is stored in the *system catalogs.*
  (Note: Record field info in Project 1 passed in "from above"…!)
- Finding the *i'th* field of a record does ***not*** require scanning the record.

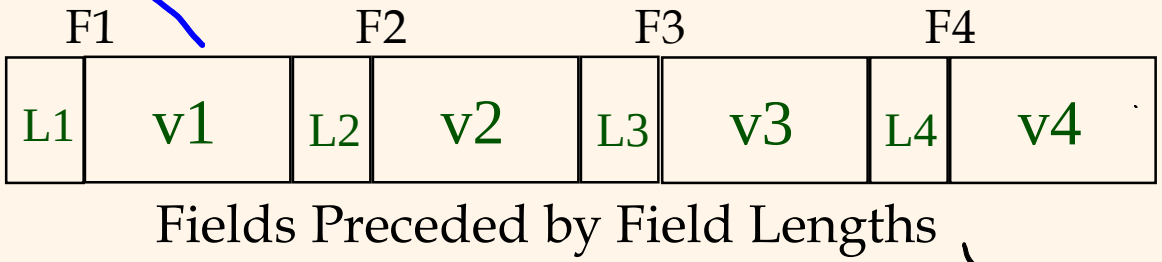
Record Formats: Variable Length

- Several alternative formats (# fields is fixed):



Consider is the data contains the special symbol, then it doesn't work.

If V1 is null, put some special to L1, like (-1) to represent V1 is null



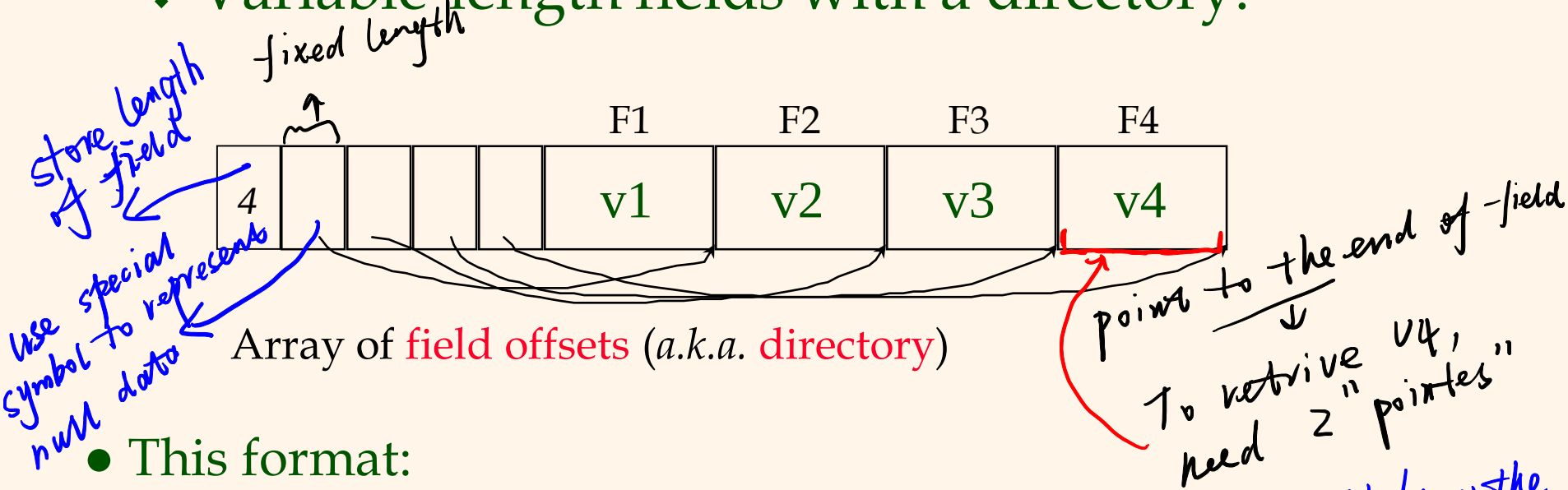
Use one byte -> L to shown the length of the field data.
If we know V1 is integer, V2 is float, we don't need to use L1 and L2.

- Some thought questions for you:
  - (1) What's true of the second format but not the first?
  - (2) What annoying disadvantage do both formats share?
  - (3) And, how do we know the field count in each case?

Because the length is variable, if you want query a special field data, you may need to scan the entire record, and this is expensive.

Record Formats: Variable Length *(continued)*

- Variable-length fields with a directory:



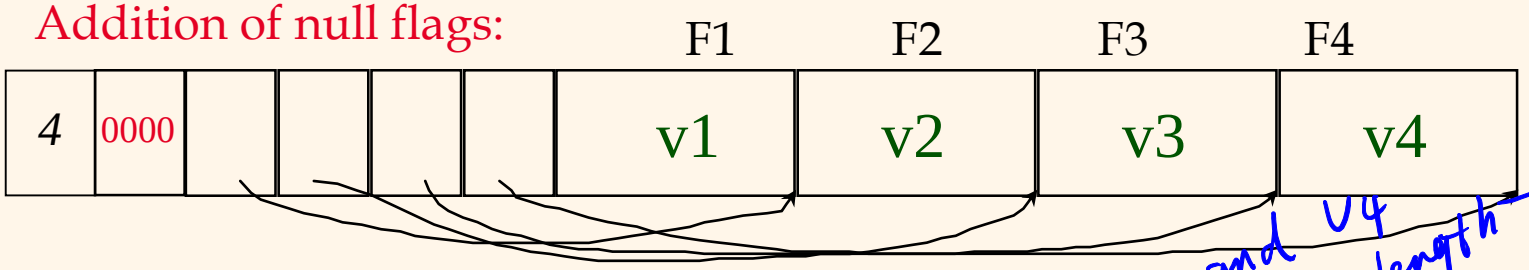
  - This format:
    - (1) Offers direct access to the *i'th* field.
    - (2) Helps support efficient storage of *null* values. (Q: How?)
    - (3) Just requires a small directory overhead.
    - (4) Can even help with ALTER TABLE ADD COLUMN! (*Q*: How?)

→ means I don't know the value

Record Formats: Variable Length

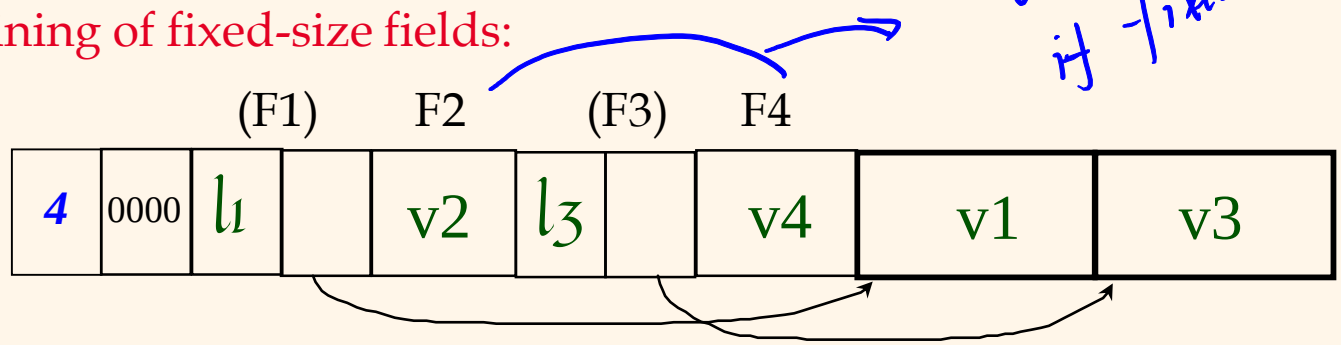
- More variations on a theme...

Addition of null flags:



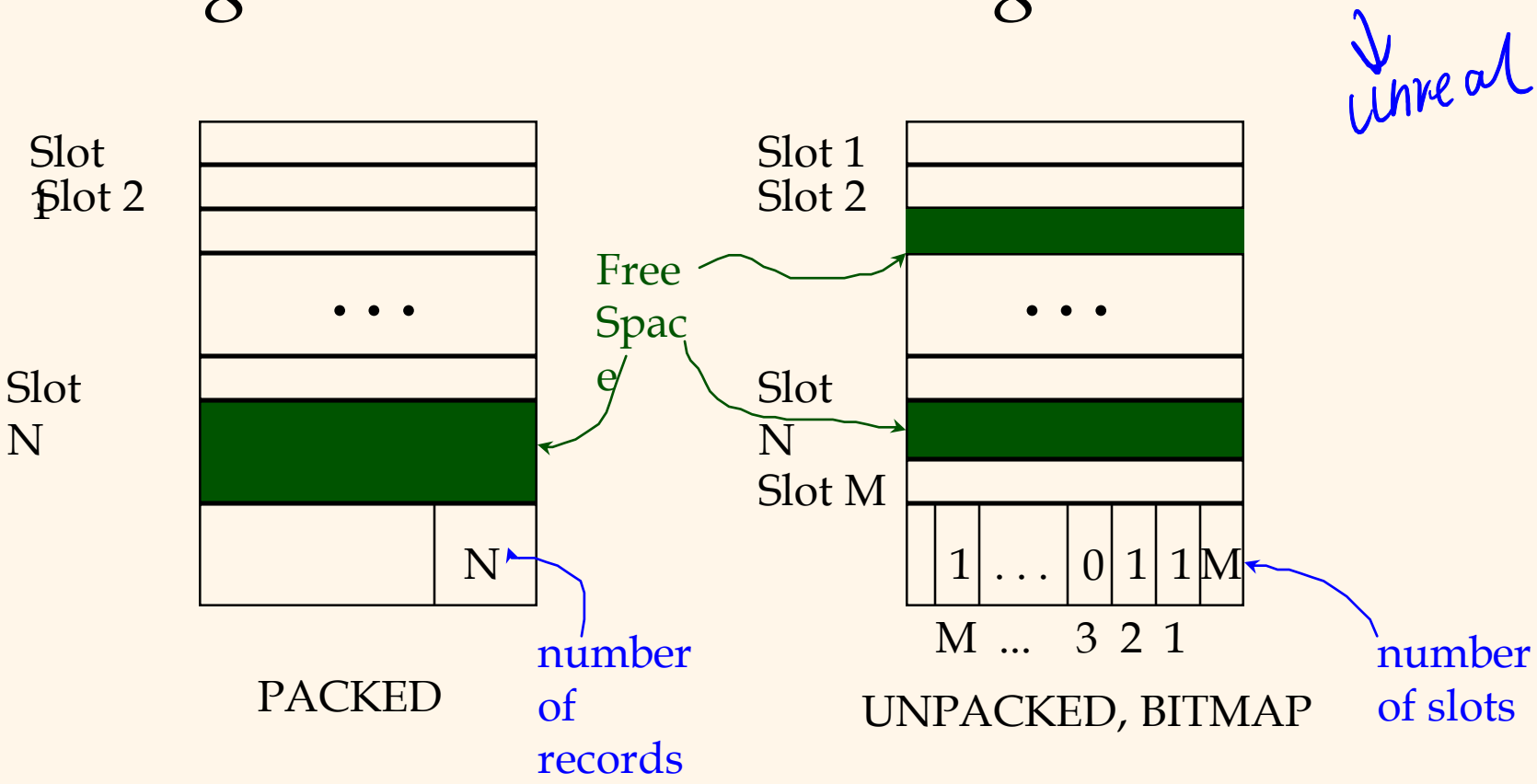
v2 and v4 if fixed length for int and float ...

Inlining of fixed-size fields:



Next: Page formats

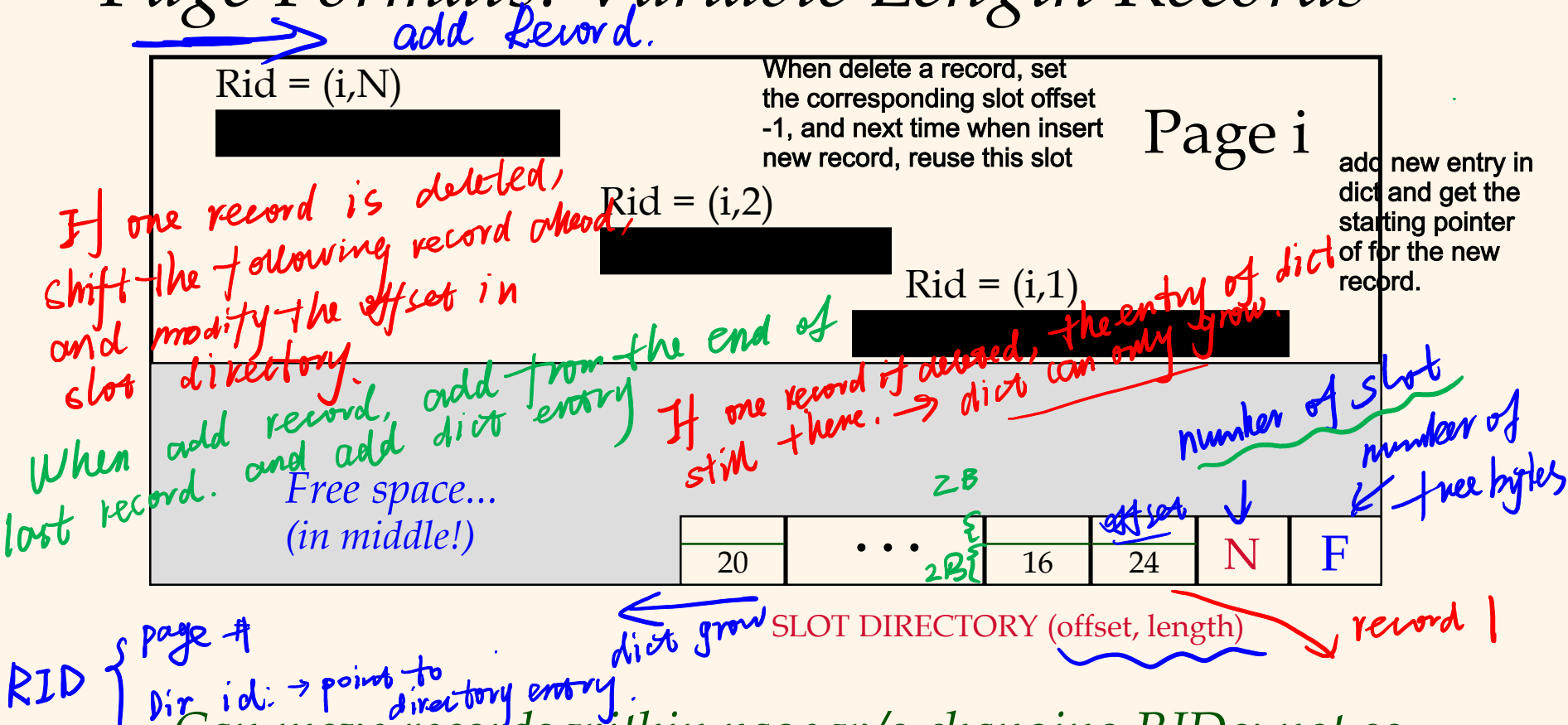
Page Formats: Fixed Length Records



- *Record id = <page id, slot #>. In the first (packed) alternative, records will move around for free space management: Rids change ☐ may be unacceptable!*

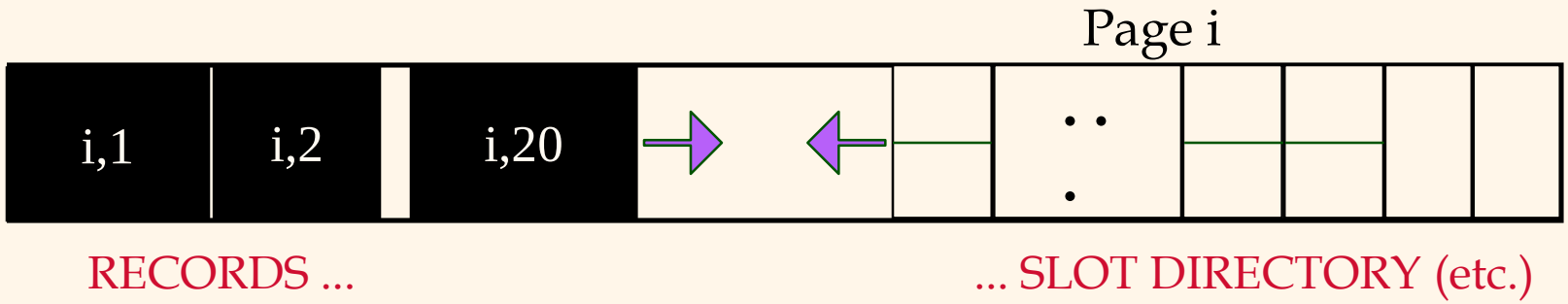
location needs 2 Byte → $\log_2^{4k}$ = 12 bit → 2 Bytes

Page Formats: Variable Length Records



- *Can move records within page w/o changing RIDs; not so unattractive for fixed-length records as a result.*
- *Record movement? (1) Tombstones, or (2) PKeys (vs. RIDs)*

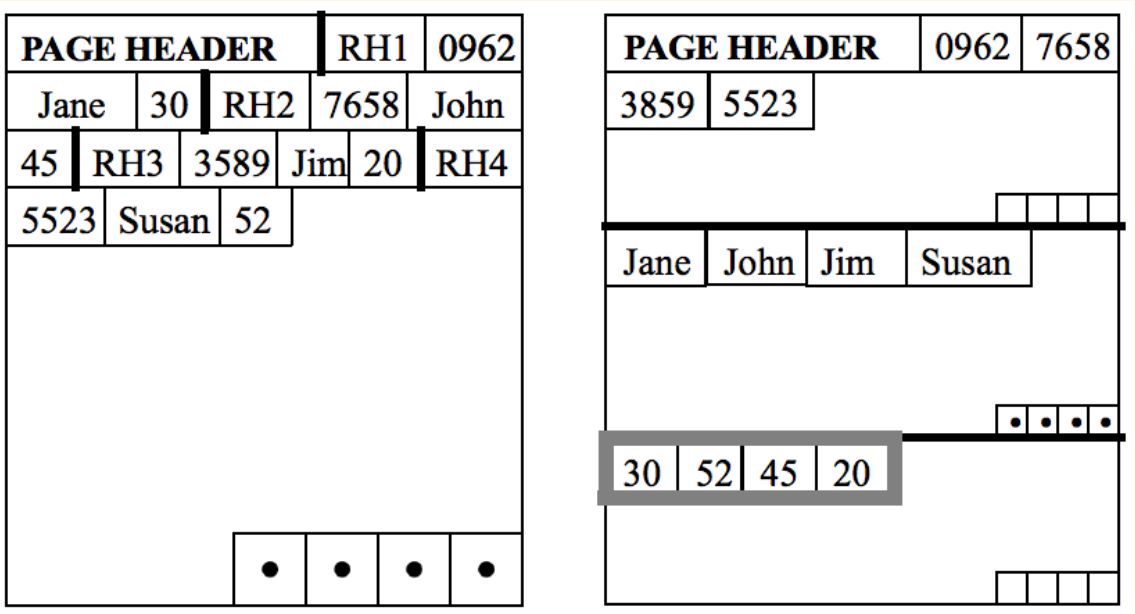
... Variable Length Records *(cont.)*



- *Two variable-sized areas growing towards to each other (living within a one-page space budget!)*
- Other variations on these formats are possible as well
  - Could track free space holes with an offset-based list structure
  - Could use a different *record* format (e.g., PAX, which clusters values by field in page rather than by record and then field)
  - ....

PAX format

Traditional Format



PAX Format

- PAX partitions each page into minipages based on fields
  - Good caching behaviors for "select fields from …";
  - Compression
  - www.pdl.cmu.edu/PDL-FTP/Database/pax.pdf
- Column store (e.g., Vertica)

Project 1: RecordBasedFileManager

freespace

start of free space: PAGE-SIZE - 2 - 2 - N*4 - F

PageDirectory slotDirectory

insertRecord

readRecord

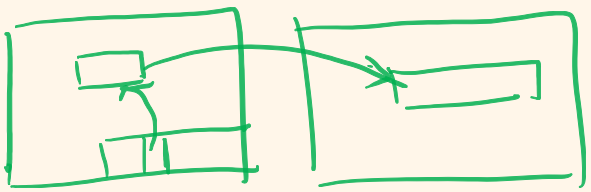
printRecord

deleteRecord: set the slot special symbol, shift the following records, change the offset of the following reset. When insert a new record, if there is a slot unused(previously deleted), first use this slot.

updateRecord:
case 1: update record needs less space: shift the following record from the start to end and change the offset and freespace.

case2: update record needs more space: If the page is available to contain this updated recrod, shift the folloeing record from the end to start. We can't change the RID, and rewrite the recrod corresponding to this record as a pointer, which points to the location of another page where the record is actually stored. Remember to change the free space of the new page.

Update ①



Update ②