

7

Modifying records

- Modifications:
 - (1) insert a new record
 - (2) Delete an existing record
 - (3) update ($\approx$ delete + insert)

• Insertion:

unordered records: (easy).

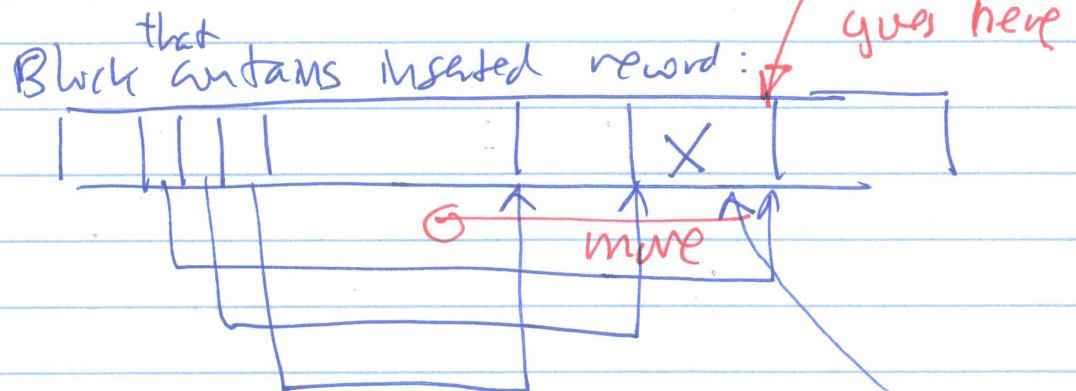
- Find a block with an empty space $\rightarrow$ insert record there
- Get a new block, insert record into the new block.

No record need to be moved !!!

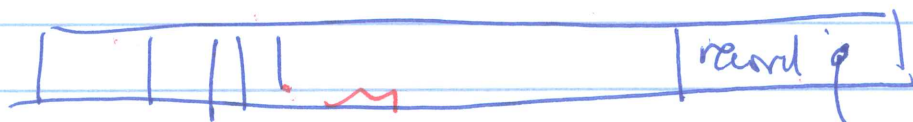
Ordered records: + insertion

- Insertion requires "moving" of some records.

- Result of moving:



Another Block with record that points to X



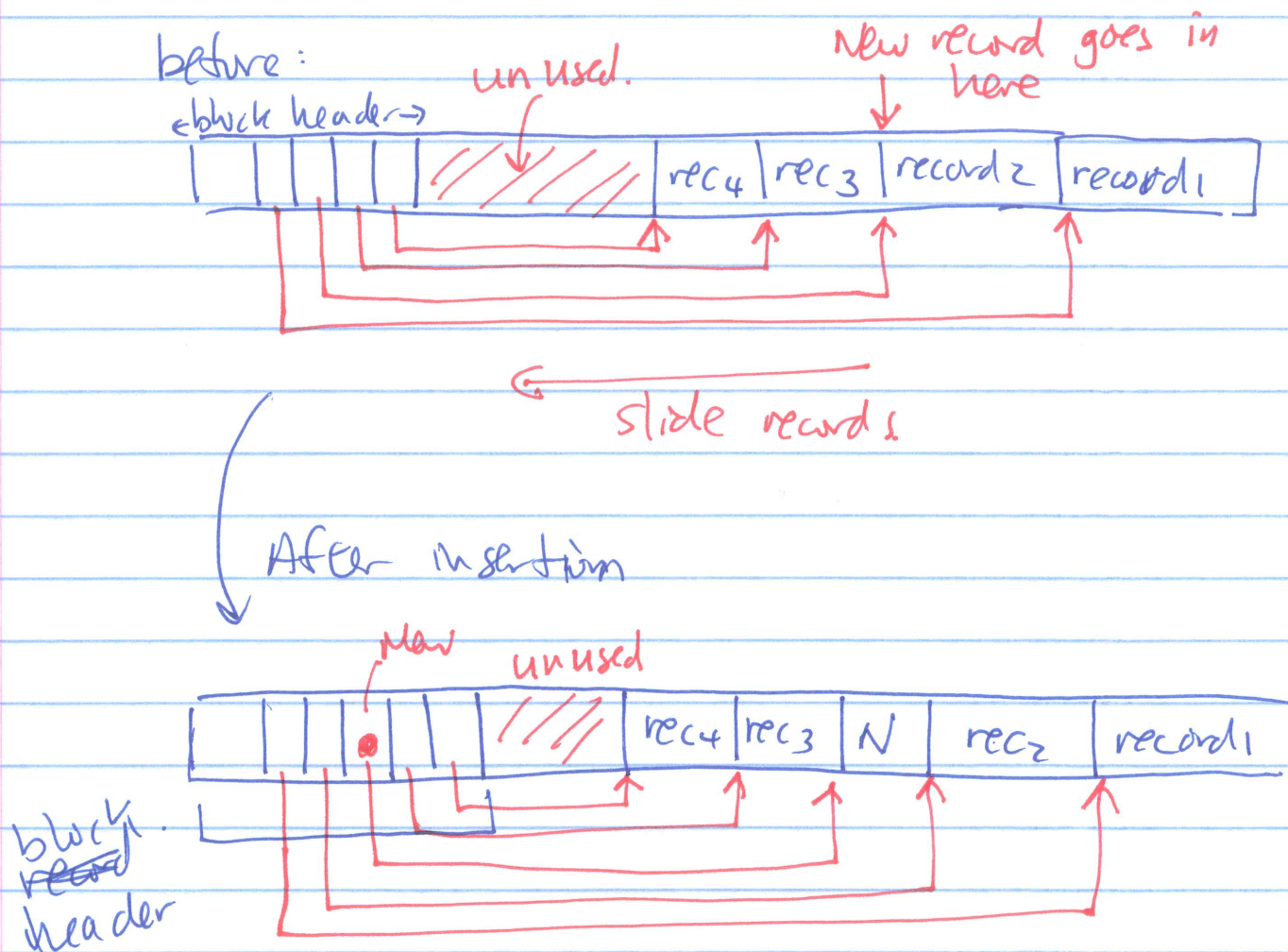
Recall there are many ways to find record in block:

- (1) Block phys. addr + offset
- (2) Block phys. addr + key

~~Answer~~ For sorted records: always use key (many search!!!)

Ordered records.

- Insert into an existing block:



- If no space in the block:

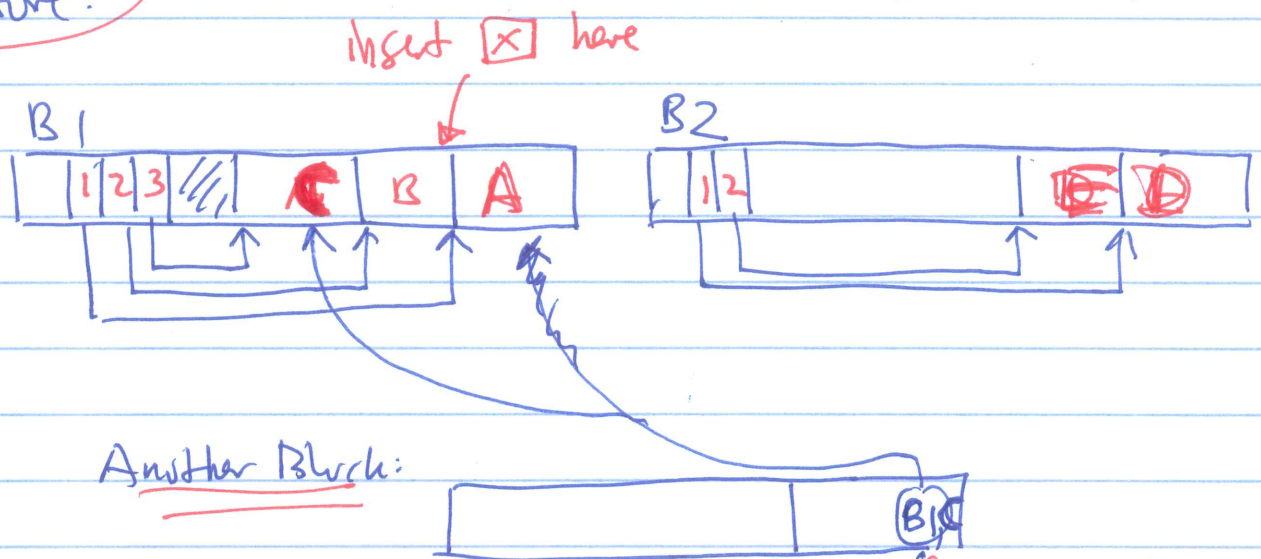
(1) Find space in a "nearby" block.
- slide

(2) Use "overflow" blocks.

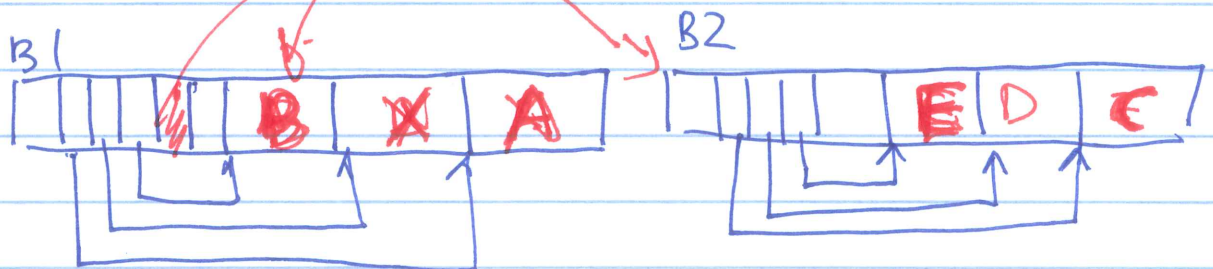
Insertion causes record to move to neighbor block

→ leave forwarding pointer!!

Before:



After insertion:



When C gets pushed out,
we must leave a
forwarding pointer in original block.

Using overflow blocks:

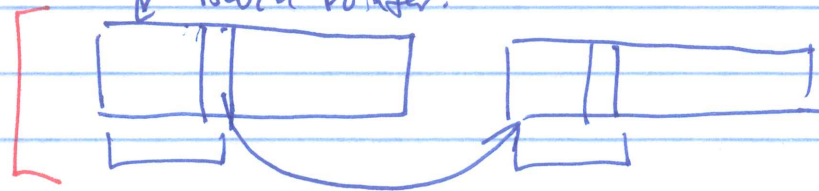
Primary Data blocks:

Block Header → Block header contains an "overflow block pointer".



overflow
Block pointer.

overflow
Blocks



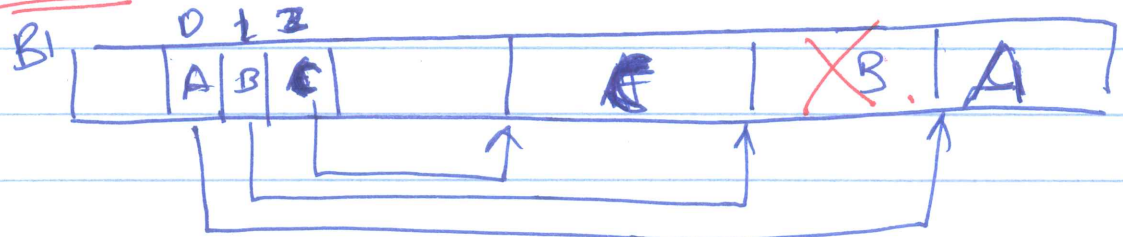
Chain of overflow Data blocks

This organization is "cleaner".

(Note: these 2 approaches are similar to
the overflow approaches used
in Hashing: $\left\{ \begin{array}{l} \text{Open addressing !!!} \\ \text{separate Chaining !!!} \end{array} \right.$

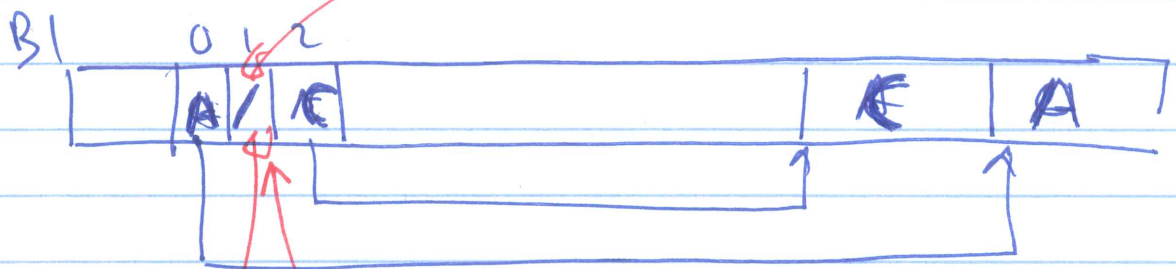
Deleting words:

- Unsorted: (+ variable length!)
(use Block phys. Addr + ~~index~~ ~~id~~ get)



reclaim space.

Tombstone ($\emptyset$).



leave a tombstone for B.

Because:

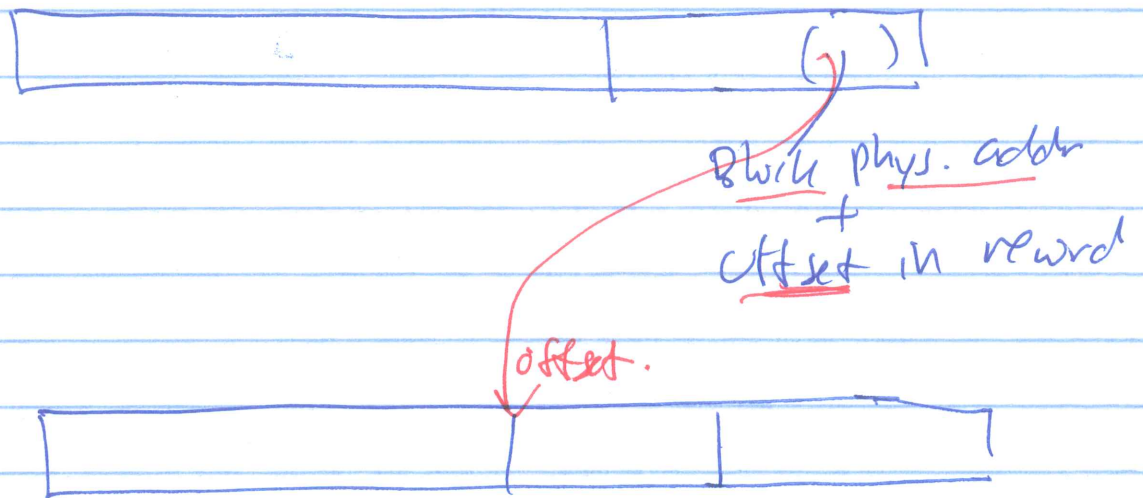
another block B2



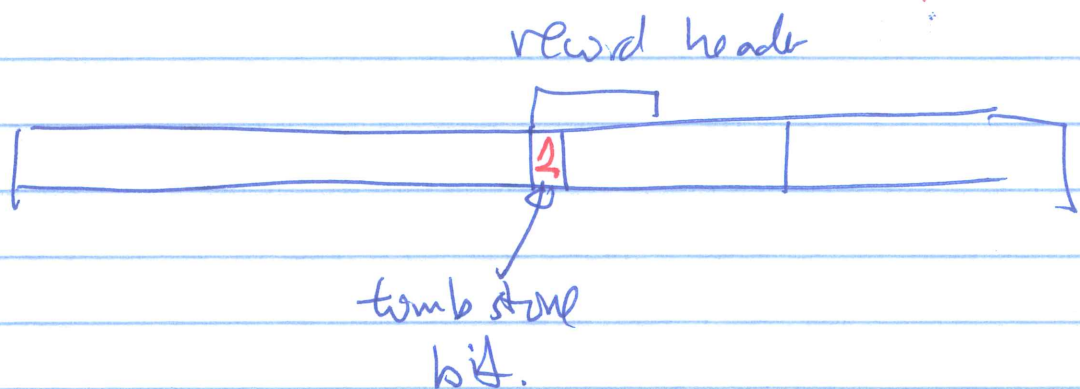
Can cause reference to deleted word to refer to

Another word !!!

- You cannot compact or deleted record if you use offset (as record pointer).

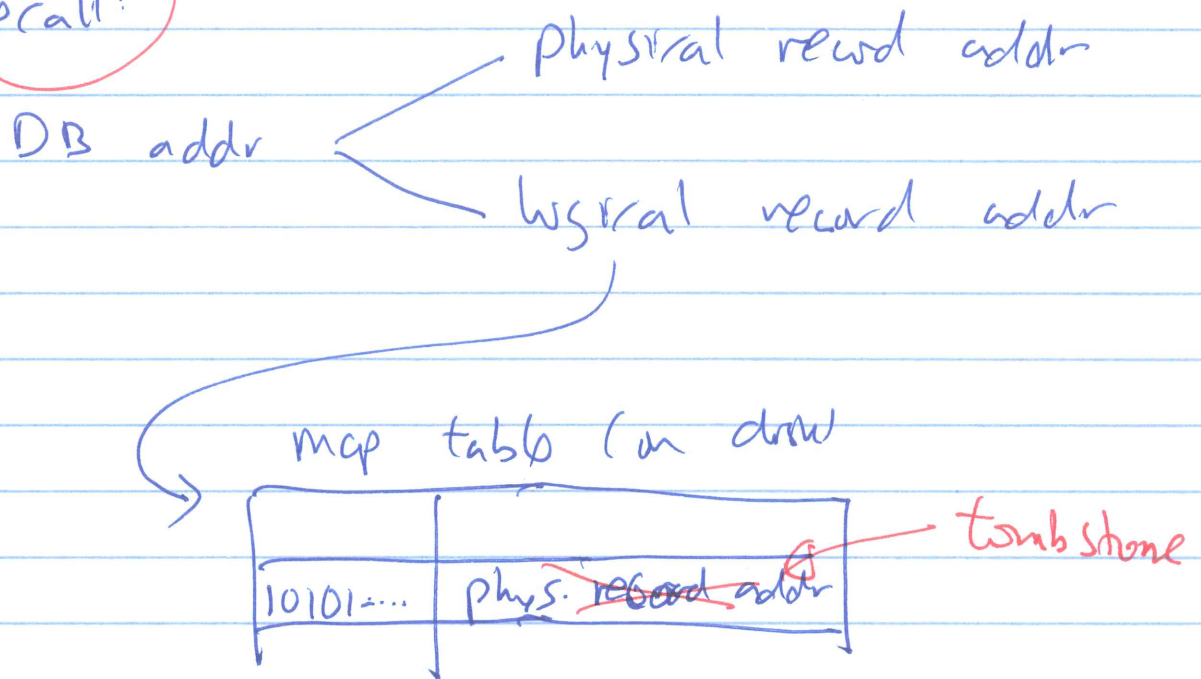


- In that case, deleting a record must set a bit in record header to mark it as a tomb stone



- Database Address is a logical record addr.

Recall:



Delete a record → replace phys. record addr. in map table with a tombstone entry.

(special bit on special phys. addr. value)

Variable length record:

If ~~you~~ use

(Block phys. addr, key)

We ~~can~~ delete the record with
the key.

⇒ key not found in block

(or overflow block)

means: record does not exist.

Update:

Fixed length records: easy

— in place update.

(no dangling record references)

Variable length records:

Handle it as an

delete
+
insert.

may result in
using forwarding reference

~~But with~~

No need to create the
tombstone!!