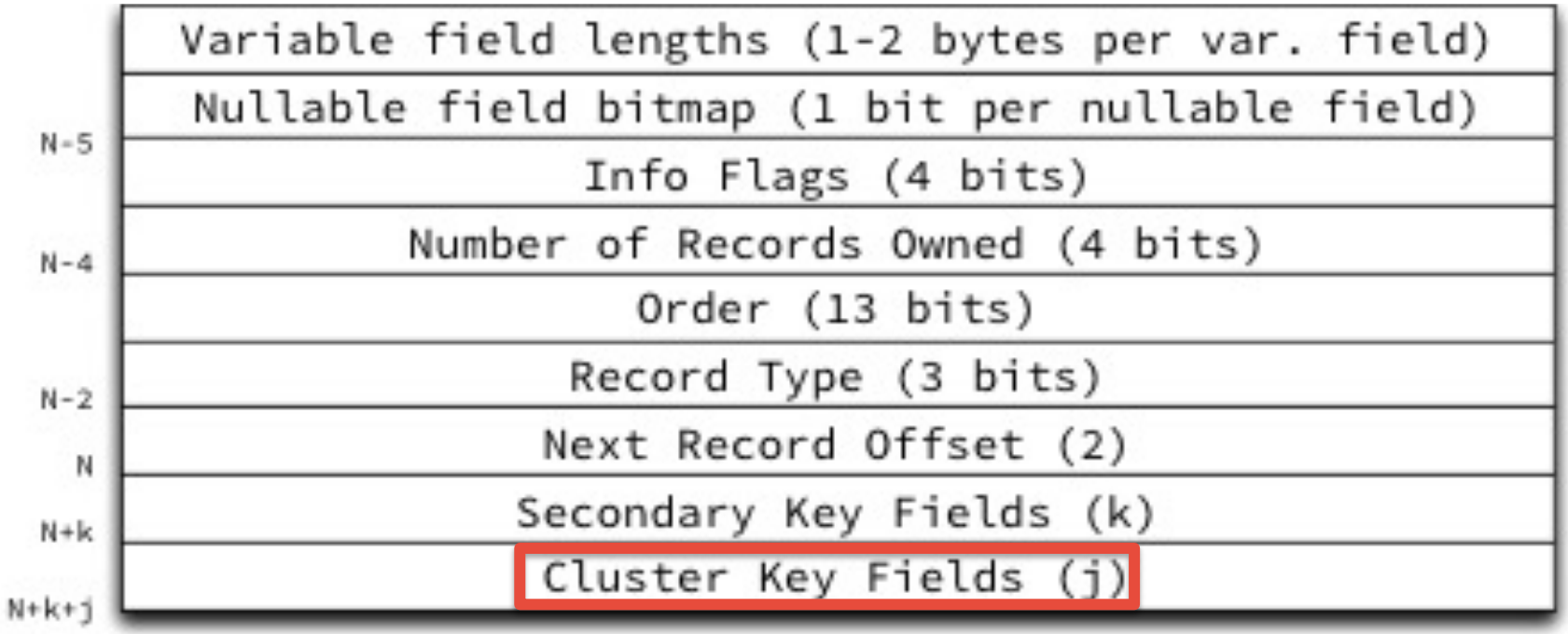
Secondary Index Search in MySQL

Mijin An
meeeeejin@gmail.com

Contents

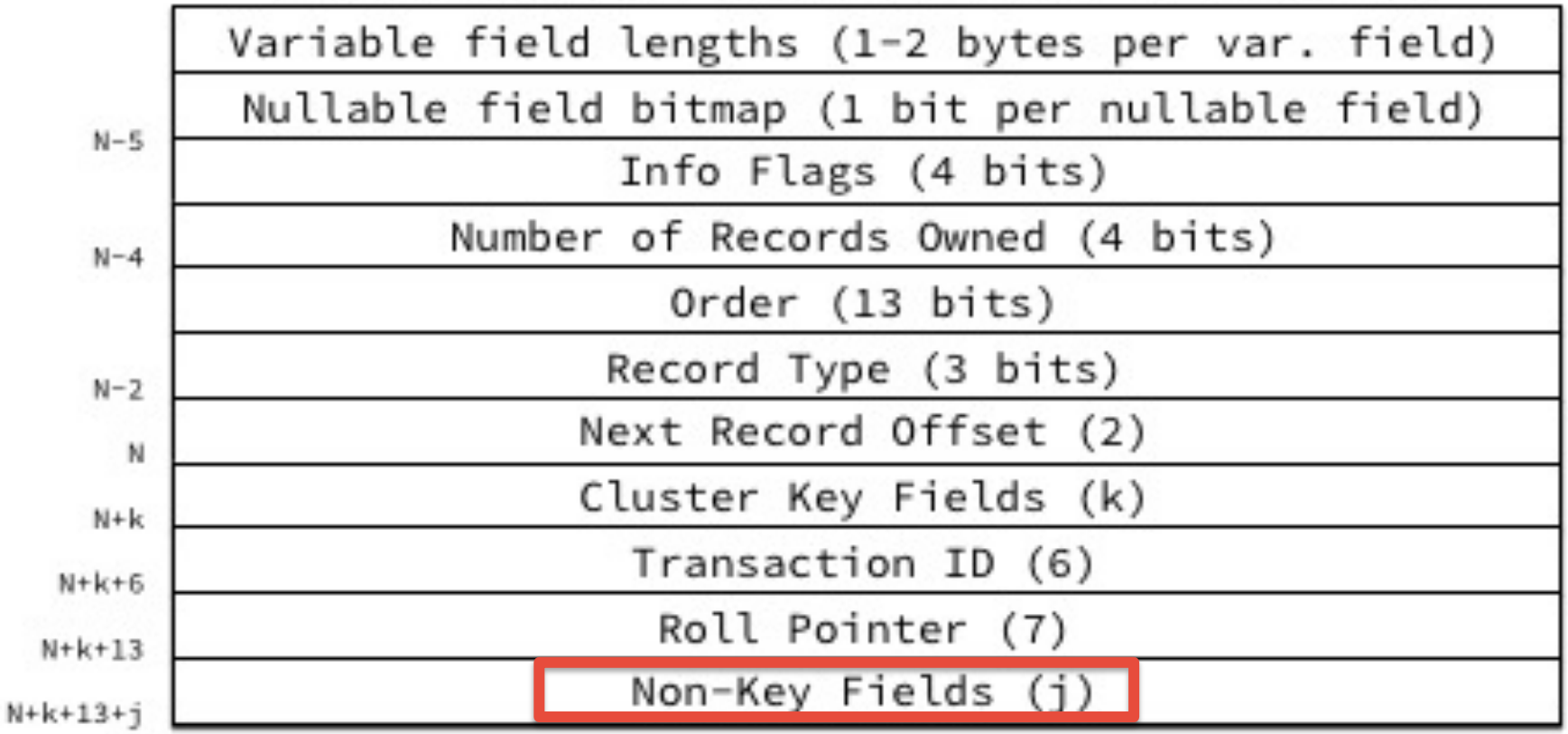
- Secondary/Primary Key Record Format
- Secondary Index Search
- Proposal for Improvement
- Improved Secondary Index Search

Secondary Key Record Format



= Primary Key Value

Primary Key Record Format



Secondary Index Search

row_search_for_mysql()

① Start a mini-transaction

② Searches an index tree, then positions a tree cursor(*pcur*) on a record corresponding to the index
(btr_pcur_open_with_no_init())

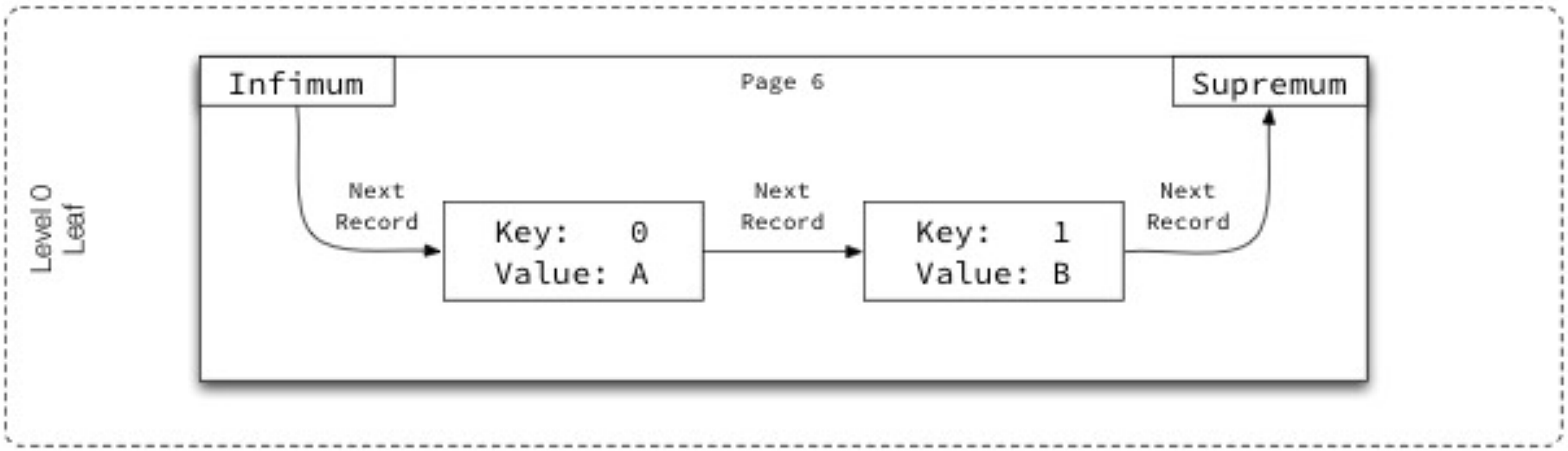
③ Get the record that *pcur* indicates(*rec*)

④ Go to ***rec_loop:***

rec_loop:

Look for matching records in a loop

a. Get the record that *pcur* indicates(*rec*)
b. If record is infimum or supremum, go to ***next_rec:***



c. If record is infimum ➔ move *pcur* to the next record
   Else if record is supremum ➔ move *pcur* to the first record of next page
   Else ➔ move *pcur* according to the search direction

d. If we may need to process the record the cursor is now on, go to ***rec_loop:***

rec_loop:

Look for matching records in a loop

a. Get the record that *pcur* indicates(*rec*)

b. If record is infimum or supremum, go to ***next_rec:***

c. Caculates the offsets to each field in the record (*offsets*)

d. If *select_lock_type* == *LOCK_S* or *LOCK_X,*

- try to place a lock on the index record

e. Go to ***locks_ok:***

If *index* != *clust_index* && *need_to_access_clustered,*

- go to ***requires_clust_rec:***

requires_clust_rec:

Get clustered record using secondary index

a. Retrieves the clustered record(*clust_rec*) corresponding to a record in a secondary index (row_sel_get_clust_rec_for_mysql())

row_sel_get_clust_rec_for_mysql()

Get clustered record using secondary index

a. Get the *clust_index* from *sec_index*
b. Searches an index tree, then positions a tree cursor (*clust_pcur*) on a record corresponding to the index
   (btr_pcur_open_with_no_init())
c. Get the record that *clust_pcur* indicates (*clust_rec*)
d. Caculate the offsets to each field in the record (*offsets*)
e. If *select_lock_type* == *LOCK_S* or *LOCK_X,*
   - try to place a lock on the index record

row_sel_get_clust_rec_for_mysql()

Get clustered record using secondary index

f. Go to ***func_exit:***

g. In ***func_exit:***, store the current *clust_pcur* position if *select_lock_type* is *LOCK_S* or *LOCK_X*

requires_clust_rec:

Get clustered record using secondary index

a. Retrieves the clustered record(*clust_rec*) corresponding to a record in a secondary index (row_sel_get_clust_rec_for_mysql())
b. If *clust_rec* is delete marked, skip it ➔ ***next_rec:***
c. Store *clust_rec* into the *result_rec*
d. Go to ***normal_return:***

Secondary Index Search

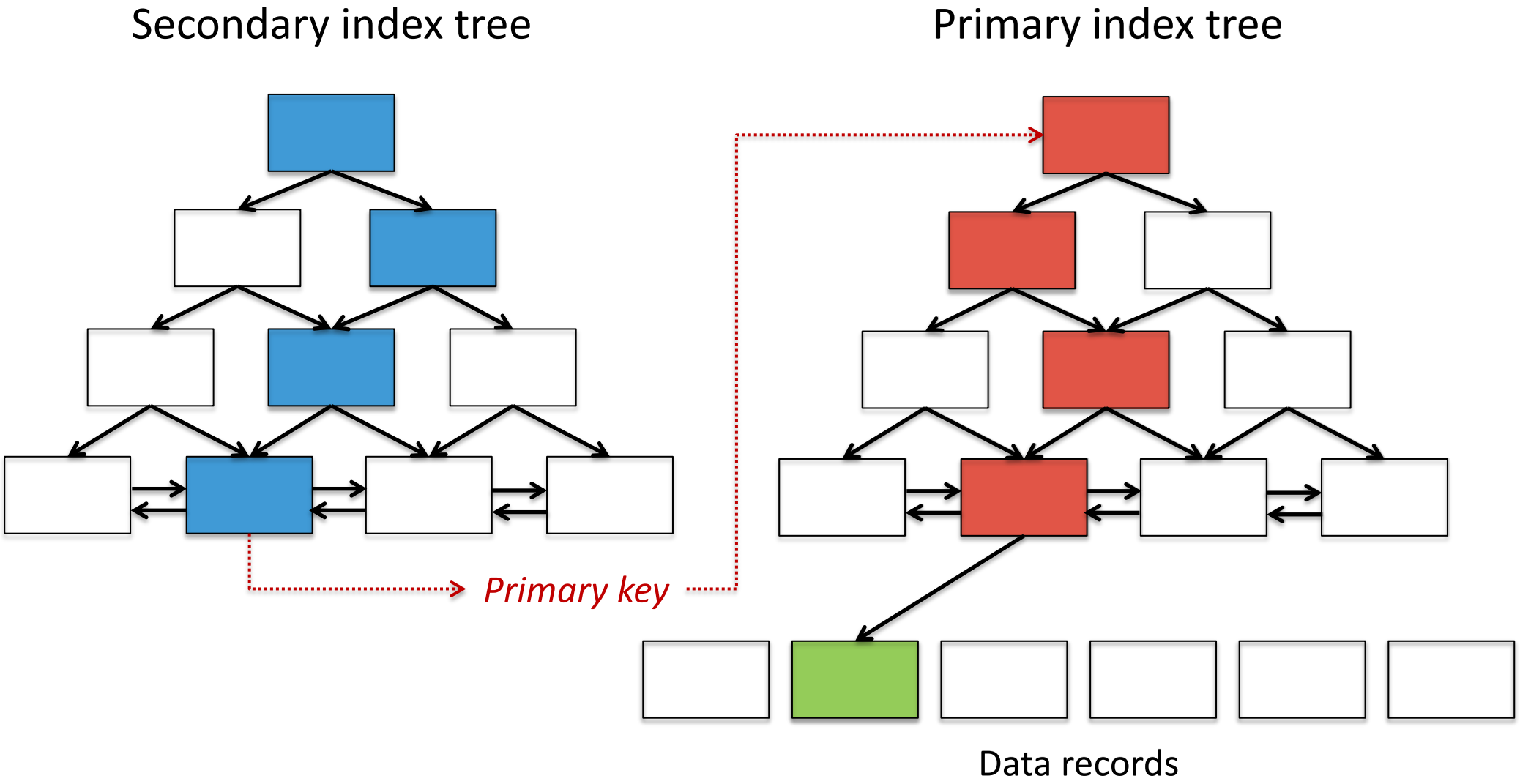
row_search_for_mysql() ➔ normal_return: ➔ func_exit:

⑤ Commit mini-transaction

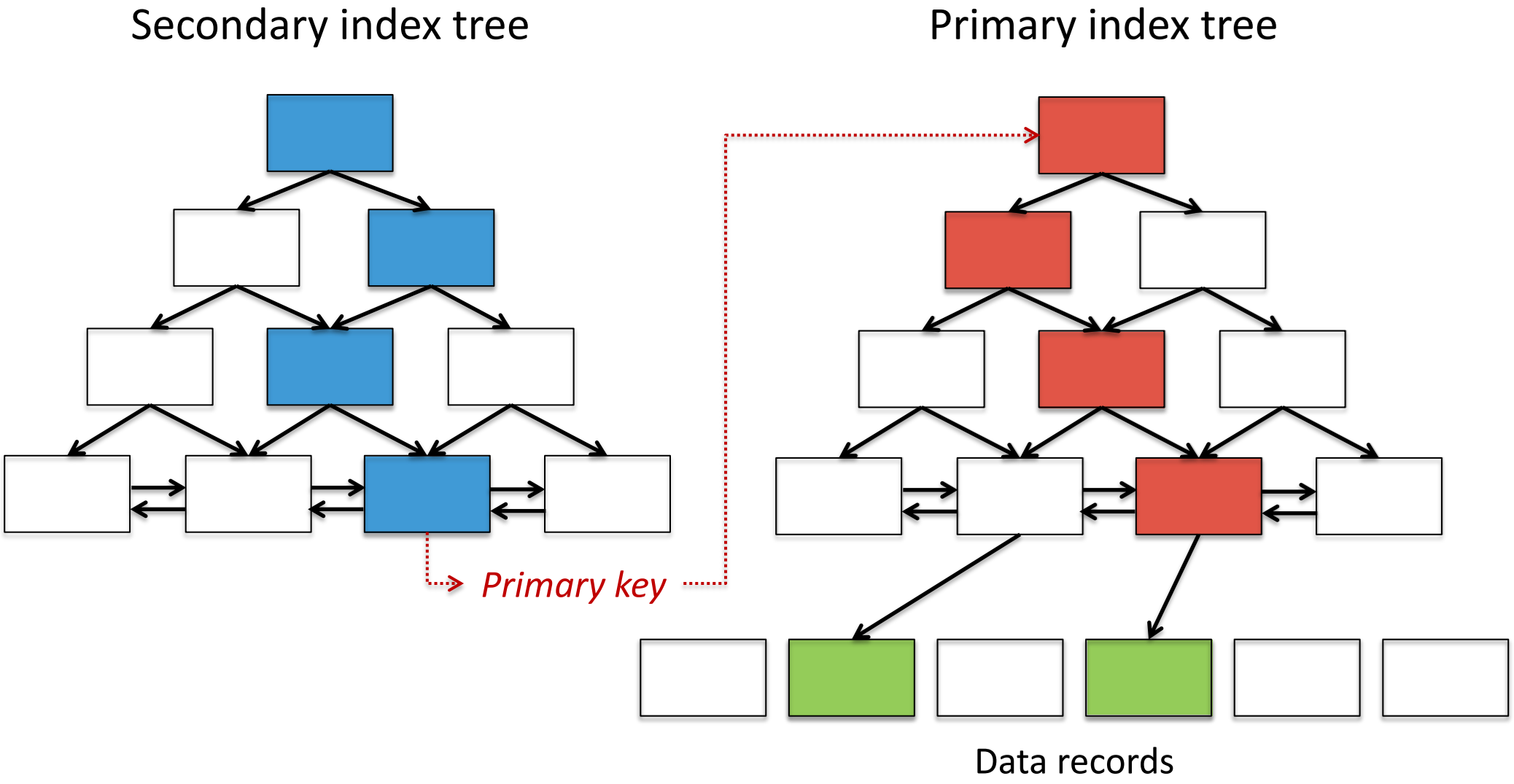
⑥ Go to ***func_exit:***

⑦ In ***func_exit:***, frees the space occupied by a memory heap then exit

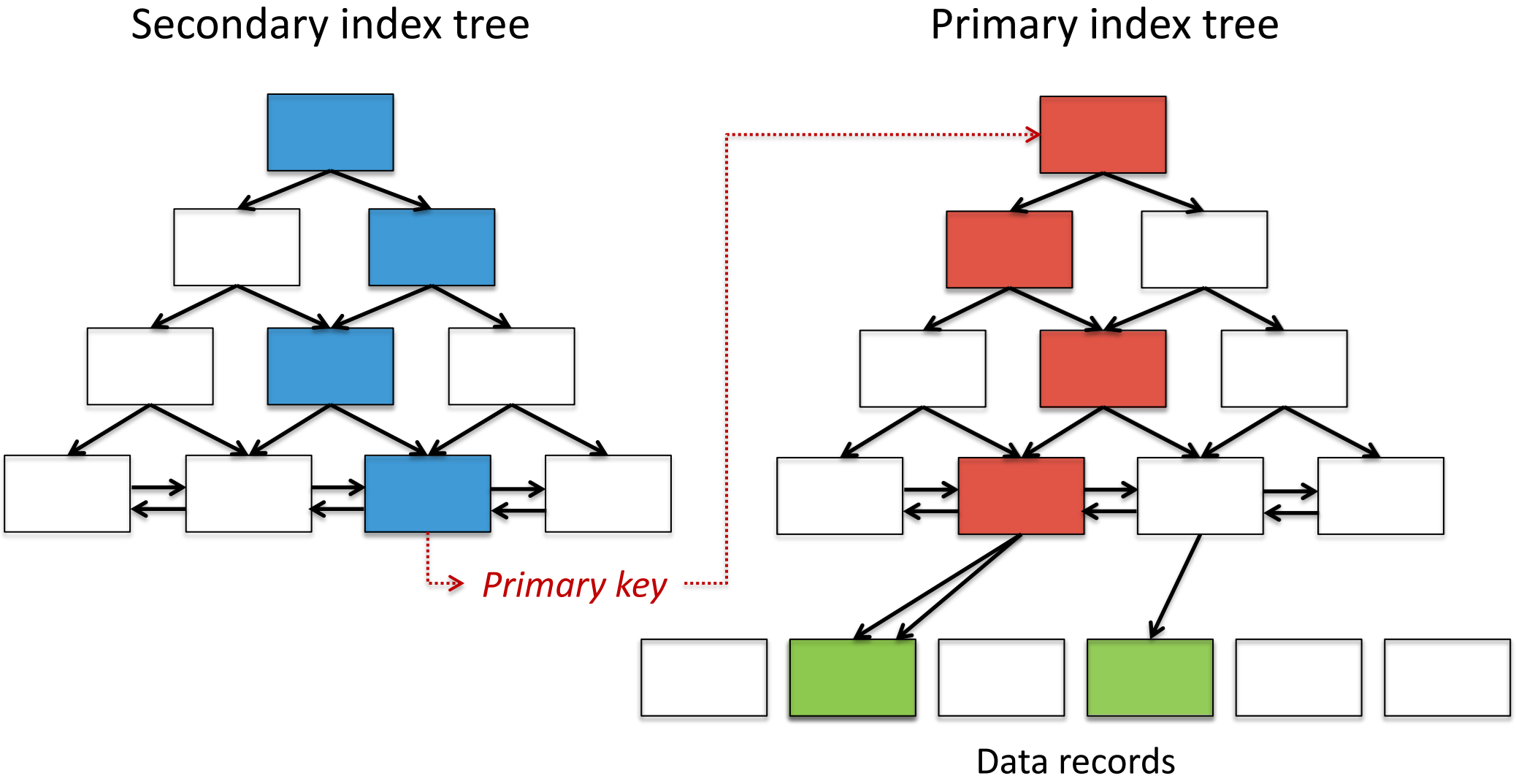
Secondary Index Search



Secondary Index Search



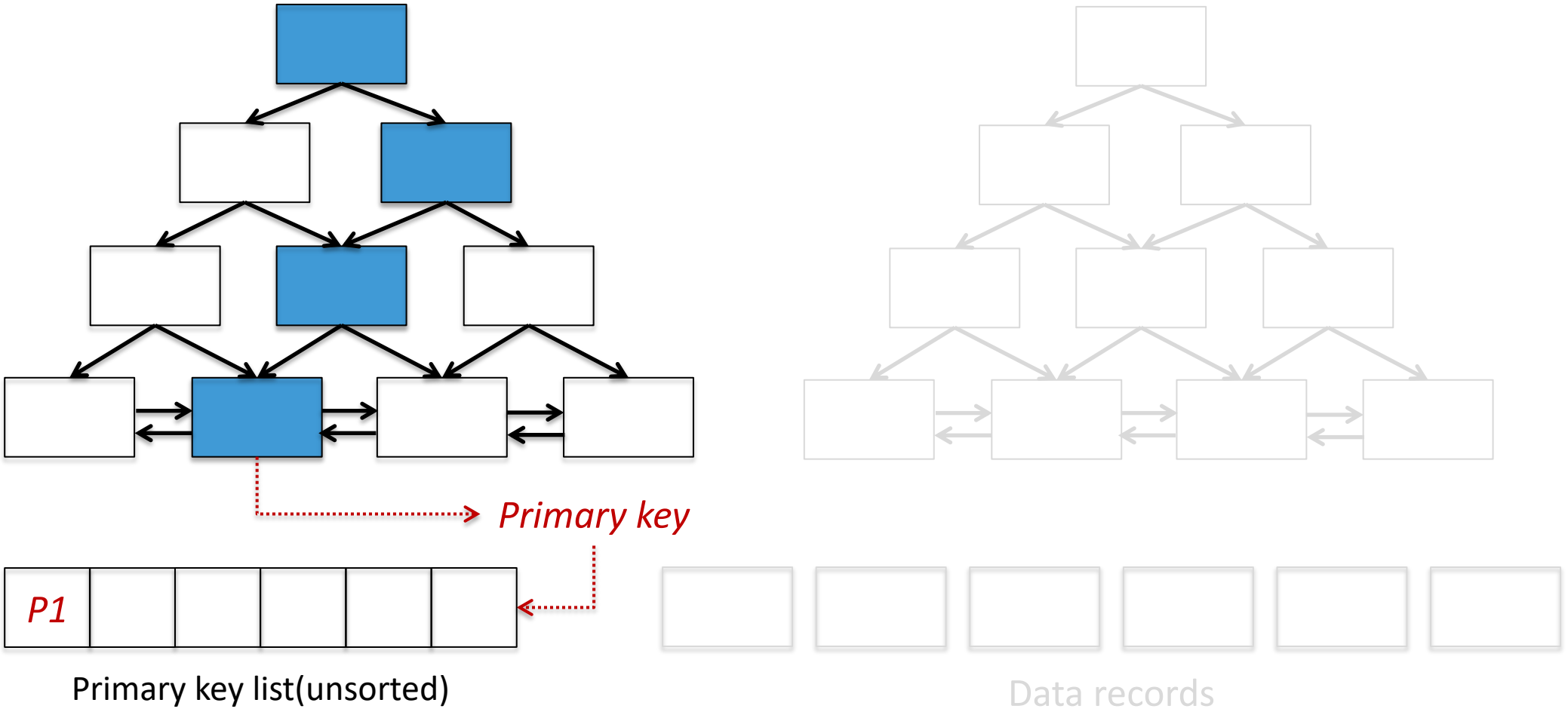
Secondary Index Search



Proposal: Prefetching in MySQL

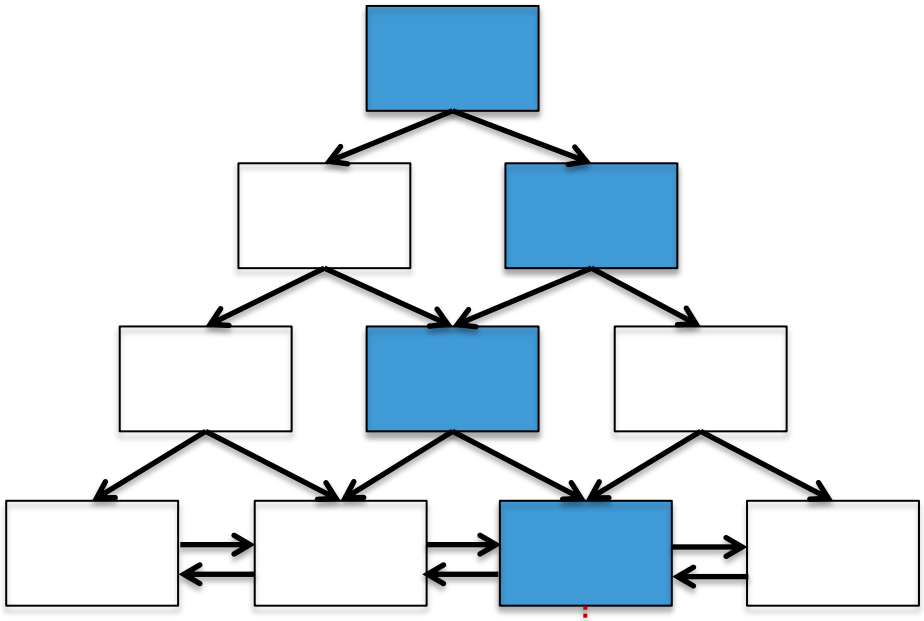
Secondary index tree

Primary index tree

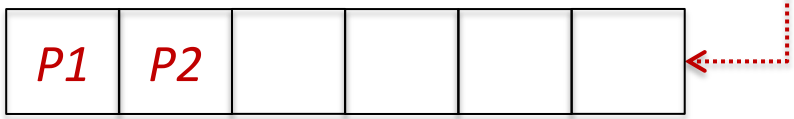


Proposal: Prefetching in MySQL

Secondary index tree



Primary key



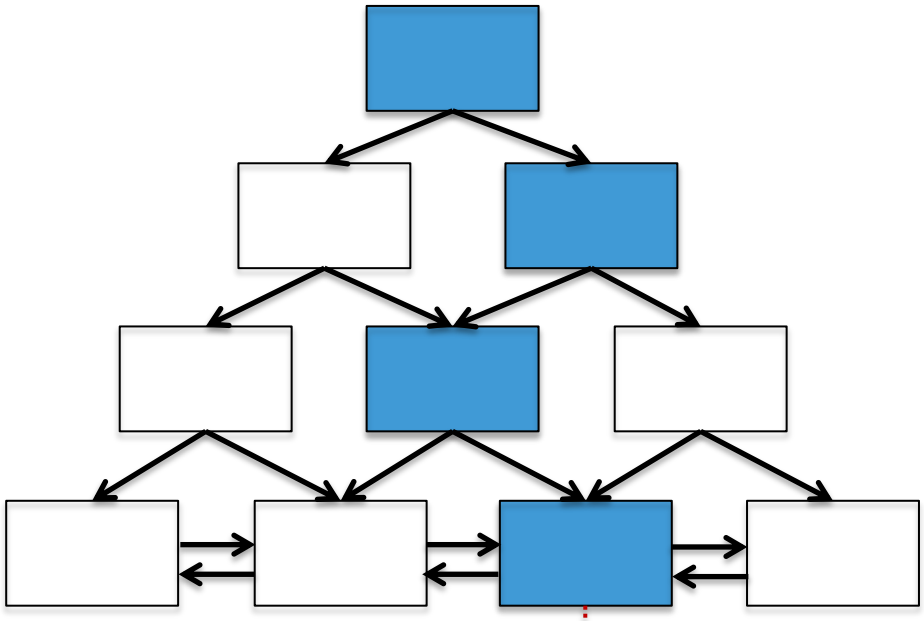
Primary key list(unsorted)

Primary index tree

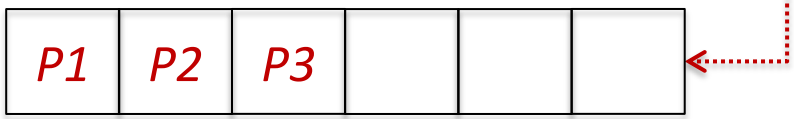
Data records

Proposal: Prefetching in MySQL

Secondary index tree



Primary key



Primary key list(unsorted)

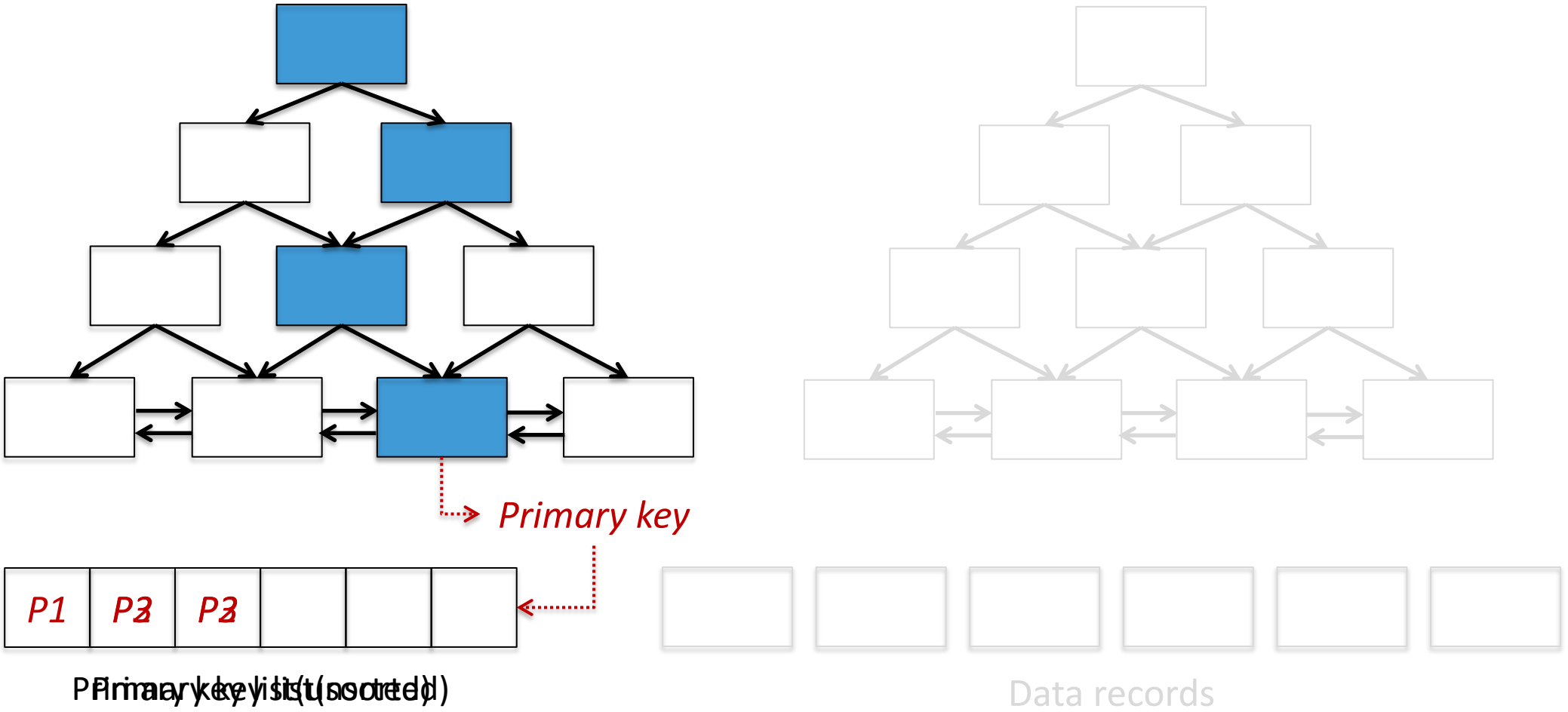
Primary index tree

Data records

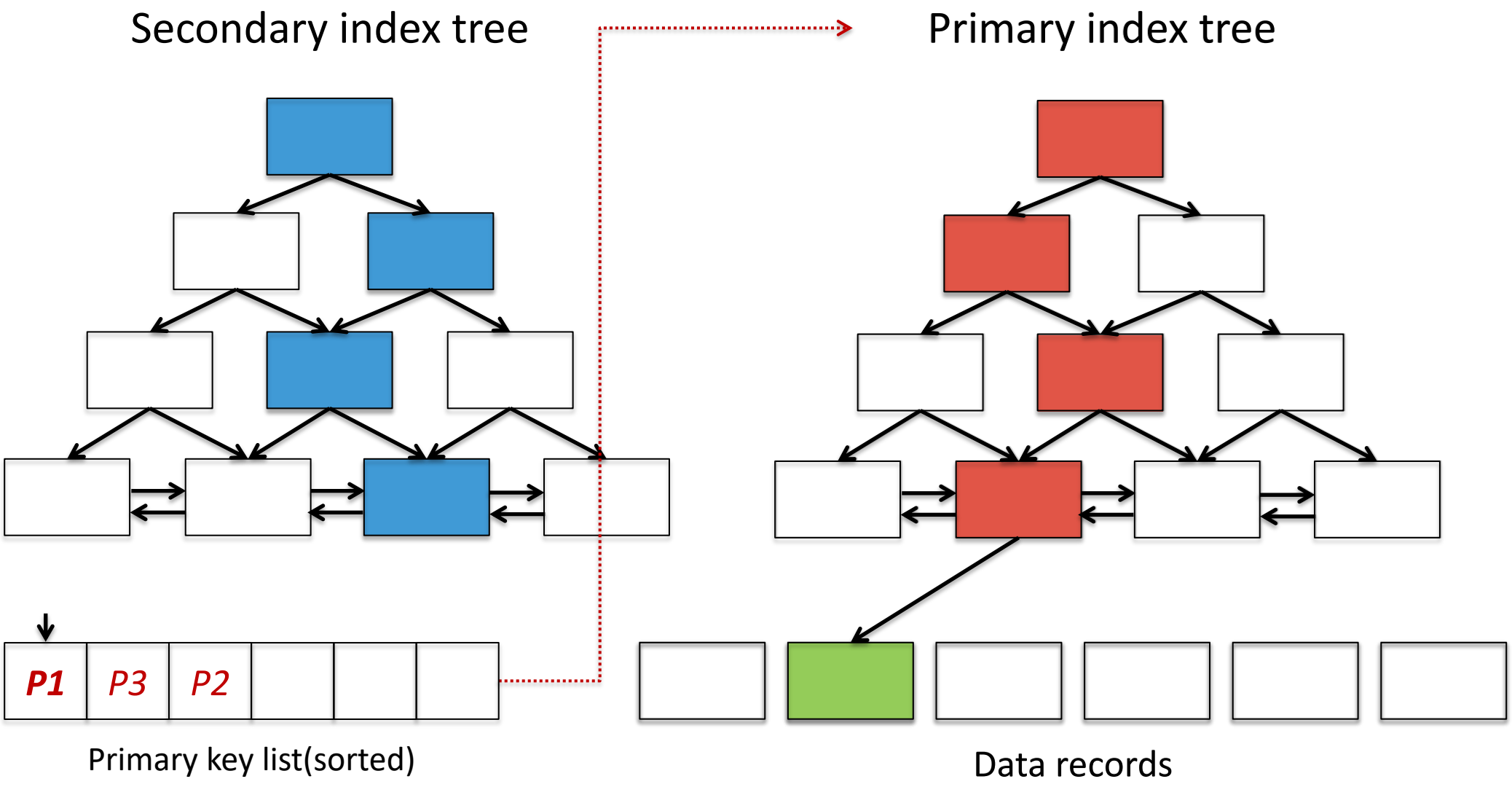
Proposal: Prefetching in MySQL

Secondary index tree

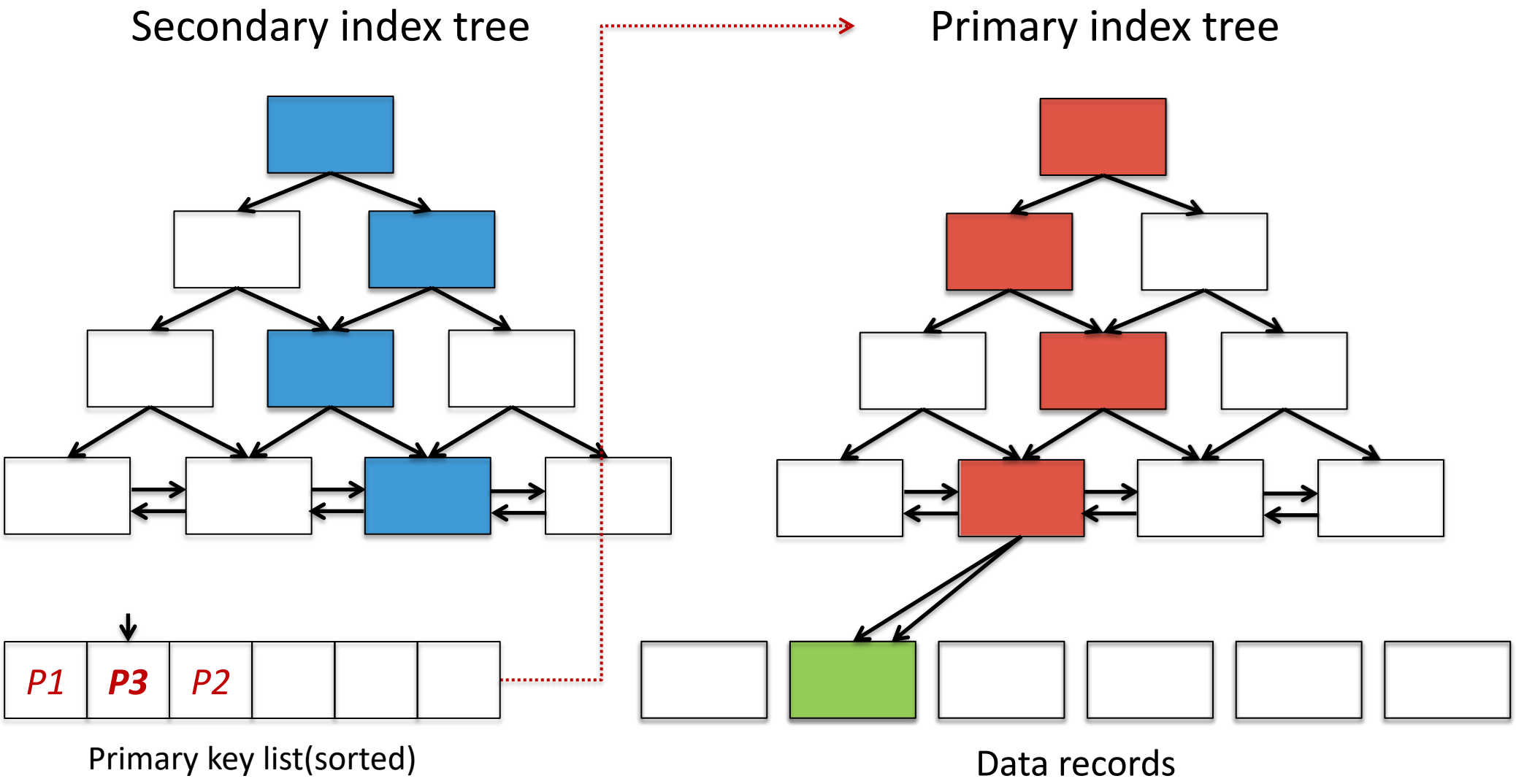
Primary index tree



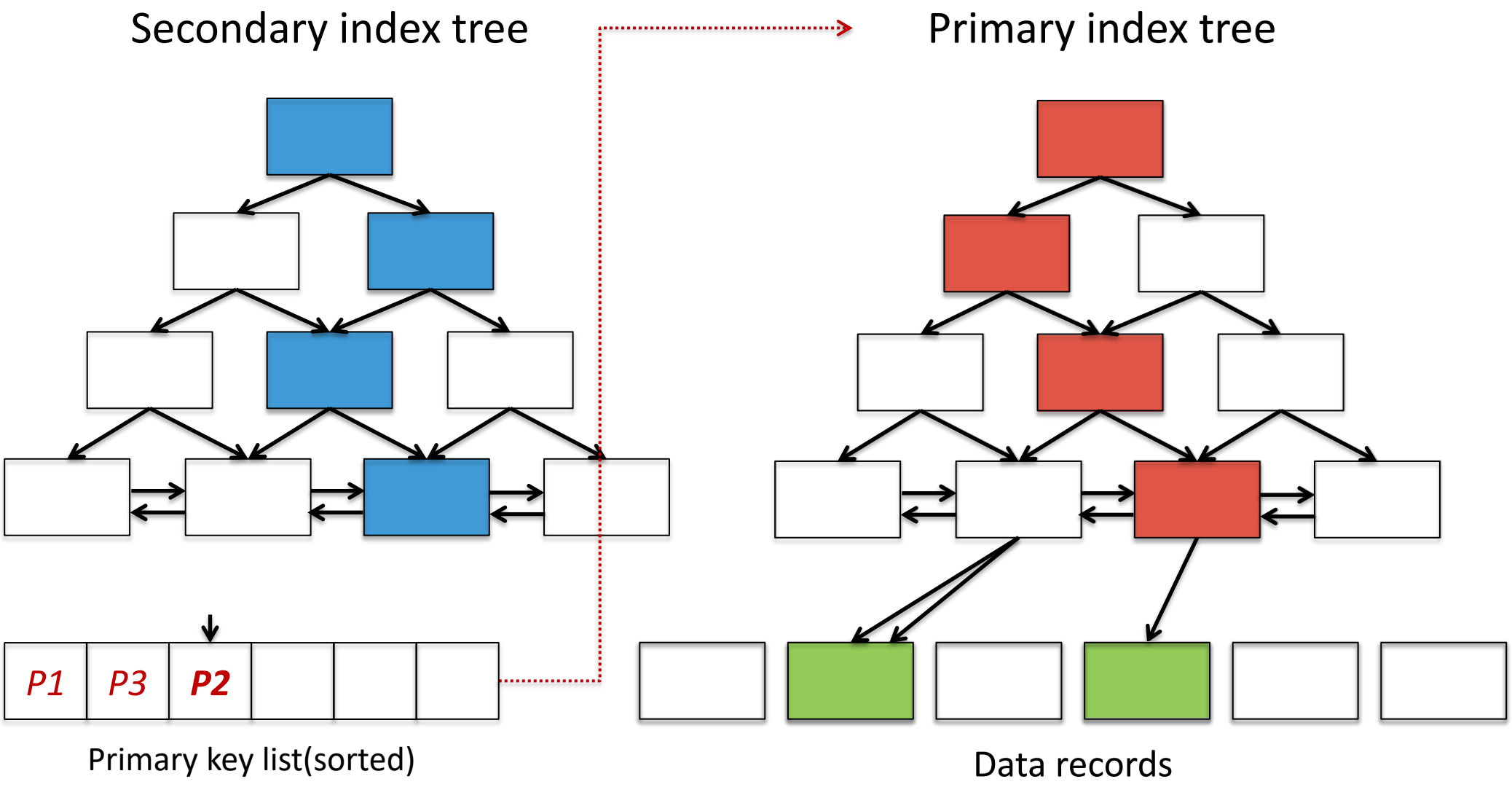
Proposal: Prefetching in MySQL



Proposal: Prefetching in MySQL



Proposal: Prefetching in MySQL



Improved Secondary Index Search

① Search a data record using secondary index

② Get primary key and insert it into primary key list, then repeat ①~② process until there is no record to search using secondary index

③ Sort the primary key list

④ Search primary index tree sequentially using sorted primary key list

⑤ Access data record sequentially

Reference

[1] “MySQL 5.6 Reference Manual”, MySQL, https://dev.mysql.com/doc/refman/5.6/en/

[2] Jeremy Cole, “InnoDB”, https://blog.jcole.us/innodb/