Script started on Sun 04 Feb 2018 07:56:49 PM MST

]0;spoorthi@spoorthi-VirtualBox: ~/Downloads/minjava/javaminibase/src[01;32mspoorthi@spoorthi-VirtualBox[00m:[01;34m~/Downloads/minjava/javaminibase/src[00m$ make test

cd tests; make bmtest dbtest; whoami; make hftest bttest indextest jointest sorttest sortmerge

make[1]: Entering directory '/home/spoorthi/Downloads/minjava/javaminibase/src/tests'

/usr/lib/java/jdk-9.0.4/bin/javac -classpath .:.. TestDriver.java BMTest.java

/usr/lib/java/jdk-9.0.4/bin/java -classpath .:.. tests.BMTest

Running Buffer Management tests....

Replacer: Clock

Test 1 does a simple test of normal buffer manager operations:

- Allocate a bunch of new pages

- Write something on each one

- Read that something back from each one

(because we're buffering, this is where most of the writes happen)

- Free the pages again

Test 1 completed successfully.

Test 2 exercises some illegal buffer manager operations:

- Try to pin more pages than there are frames

\*\*\* Pinning too many pages

--> Failed as expected

- Try to free a doubly-pinned page

\*\*\* Freeing a pinned page

--> Failed as expected

- Try to unpin a page not in the buffer pool

\*\*\* Unpinning a page not in the buffer pool

--> Failed as expected

Test 2 completed successfully.

Test 3 exercises some of the internals of the buffer manager

- Allocate and dirty some new pages, one at a time, and leave some pinned

- Read the pages

Test 3 completed successfully.

...Buffer Management tests completely successfully.

/usr/lib/java/jdk-9.0.4/bin/javac -classpath .:.. TestDriver.java DBTest.java

/usr/lib/java/jdk-9.0.4/bin/java -classpath .:.. tests.DBTest

Running Disk Space Management tests....

Replacer: Clock

Test 1 creates a new database and does some tests of normal operations:

- Add some file entries

- Allocate a run of pages

- Write something on some of them

- Deallocate the rest of them

Test 1 completed successfully.

Test 2 opens the database created in test 1 and does some further tests:

- Delete some of the file entries

- Look up file entries that should still be there

- Read stuff back from pages we wrote in test 1

Test 2 completed successfully.

Test 3 tests for some error conditions:

- Look up a deleted file entry

\*\*\*\* Looking up a deleted file entry

--> Failed as expected

- Try to delete a deleted entry again

\*\*\*\* Delete a deleted file entry again

--> Failed as expected

- Try to delete a nonexistent file entry

\*\*\*\* Deleting a nonexistent file entry

--> Failed as expected

- Look up a nonexistent file entry

\*\*\*\* Looking up a nonexistent file entry

--> Failed as expected

- Try to add a file entry that's already there

\*\*\*\* Adding a duplicate file entry

--> Failed as expected

- Try to add a file entry whose name is too long

\*\*\*\* Adding a file entry with too long a name

--> Failed as expected

- Try to allocate a run of pages that's too long

\*\*\*\* Allocating a run that's too long

--> Failed as expected

- Try to allocate a negative run of pages

\*\*\*\* Allocating a negative run

--> Failed as expected

- Try to deallocate a negative run of pages

\*\*\*\* Deallocating a negative run

--> Failed as expected

Test 3 completed successfully.

Test 4 tests some boundary conditions.

(These tests are very implementation-specific.)

- Make sure no pages are pinned

- Allocate all pages remaining after DB overhead is accounted for

- Attempt to allocate one more page

\*\*\*\* Allocating one additional page

--> Failed as expected

- Free some of the allocated pages

- Allocate some of the just-freed pages

- Free two continued run of the allocated pages

- Allocate back number of pages equal to the just freed pages

- Add enough file entries that the directory must surpass a page

- Make sure that the directory has taken up an extra page: try to

allocate more pages than should be available

\*\*\*\* Allocating more pages than are now available

--> Failed as expected

- At this point, all pages should be claimed. Try to allocateone more.

\*\*\*\* Allocating one more page than there is

--> Failed as expected

- Free the last two pages: this tests a boundary condition in the space map.

Test 4 completed successfully.

...Disk Space Management tests completely successfully.

make[1]: Leaving directory '/home/spoorthi/Downloads/minjava/javaminibase/src/tests'

spoorthi

make[1]: Entering directory '/home/spoorthi/Downloads/minjava/javaminibase/src/tests'

/usr/lib/java/jdk-9.0.4/bin/javac -classpath .:.. TestDriver.java HFTest.java

/usr/lib/java/jdk-9.0.4/bin/java -classpath .:.. tests.HFTest

Running Heap File tests....

Replacer: Clock

Test 1: Insert and scan fixed-size records

- Create a heap file

- Add 100 records to the file

- Scan the records just inserted

Test 1 completed successfully.

Test 2: Delete fixed-size records

- Open the same heap file as test 1

- Delete half the records

- Scan the remaining records

Test 2 completed successfully.

Test 3: Update fixed-size records

- Open the same heap file as tests 1 and 2

- Change the records

- Check that the updates are really there

Test 3 completed successfully.

Test 4: Test some error conditions

- Try to change the size of a record

\*\*\*\* Shortening a record

--> Failed as expected

\*\*\*\* Lengthening a record

--> Failed as expected

- Try to insert a record that's too long

\*\*\*\* Inserting a too-long record

--> Failed as expected

Test 4 completed successfully.

...Heap File tests completely successfully.

/usr/lib/java/jdk-9.0.4/bin/javac -classpath .:.. TestDriver.java BTTest.java

/usr/lib/java/jdk-9.0.4/bin/java -classpath .:.. tests.BTTest

Replacer: Clock

Running tests....

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The file name is: AAA0 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The file name is: AAA1 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The file name is: AAA2 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :5

Please input the integer key to insert:

1

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :5

Please input the integer key to insert:

12

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :5

Please input the integer key to insert:

3

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :5

Please input the integer key to insert:

14

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :5

Please input the integer key to insert:

5

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :5

Please input the integer key to insert:

16

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :3

---------------The B+ Tree Leaf Pages---------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 5

Left Link : -1

Right Link : -1

0 (key, [pageNo, slotNo]): (1, [ 1 1 ] )

1 (key, [pageNo, slotNo]): (3, [ 3 3 ] )

2 (key, [pageNo, slotNo]): (5, [ 5 5 ] )

3 (key, [pageNo, slotNo]): (12, [ 12 12 ] )

4 (key, [pageNo, slotNo]): (14, [ 14 14 ] )

5 (key, [pageNo, slotNo]): (16, [ 16 16 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

------------- All Leaf Pages Have Been Printed --------

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :2

---------------The B+ Tree Structure---------------

1 5

--------------- End ---------------

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :3 6

Please input the integer key to delete:

3

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :7

Please input the number of keys to insert:

5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The file name is: AAA3 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :3

---------------The B+ Tree Leaf Pages---------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 7

Left Link : -1

Right Link : -1

0 (key, [pageNo, slotNo]): (0, [ 0 0 ] )

1 (key, [pageNo, slotNo]): (1, [ 1 1 ] )

2 (key, [pageNo, slotNo]): (2, [ 2 2 ] )

3 (key, [pageNo, slotNo]): (3, [ 3 3 ] )

4 (key, [pageNo, slotNo]): (4, [ 4 4 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

------------- All Leaf Pages Have Been Printed --------

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :8

Please input the number of keys to insert:

5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The file name is: AAA4 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :3

---------------The B+ Tree Leaf Pages---------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 9

Left Link : -1

Right Link : -1

0 (key, [pageNo, slotNo]): (1, [ 1 1 ] )

1 (key, [pageNo, slotNo]): (2, [ 2 2 ] )

2 (key, [pageNo, slotNo]): (3, [ 3 3 ] )

3 (key, [pageNo, slotNo]): (4, [ 4 4 ] )

4 (key, [pageNo, slotNo]): (5, [ 5 5 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

------------- All Leaf Pages Have Been Printed --------

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :9

Please input the number of keys to insert:

5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The file name is: AAA5 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :3

---------------The B+ Tree Leaf Pages---------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 11

Left Link : -1

Right Link : -1

0 (key, [pageNo, slotNo]): (0, [ 0 0 ] )

1 (key, [pageNo, slotNo]): (1, [ 1 1 ] )

2 (key, [pageNo, slotNo]): (2, [ 2 2 ] )

3 (key, [pageNo, slotNo]): (3, [ 3 3 ] )

4 (key, [pageNo, slotNo]): (4, [ 4 4 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

------------- All Leaf Pages Have Been Printed --------

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :10

Please input the number of keys to insert:

5

Please input the number of keys to delete:

2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The file name is: AAA6 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :3

---------------The B+ Tree Leaf Pages---------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 13

Left Link : -1

Right Link : -1

0 (key, [pageNo, slotNo]): (0, [ 0 0 ] )

1 (key, [pageNo, slotNo]): (1, [ 1 1 ] )

2 (key, [pageNo, slotNo]): (2, [ 2 2 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

------------- All Leaf Pages Have Been Printed --------

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :4

Please input the page number:

11

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 11

Left Link : -1

Right Link : -1

0 (key, [pageNo, slotNo]): (0, [ 0 0 ] )

1 (key, [pageNo, slotNo]): (1, [ 1 1 ] )

2 (key, [pageNo, slotNo]): (2, [ 2 2 ] )

3 (key, [pageNo, slotNo]): (3, [ 3 3 ] )

4 (key, [pageNo, slotNo]): (4, [ 4 4 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :4

Please input the page number:

9

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 9

Left Link : -1

Right Link : -1

0 (key, [pageNo, slotNo]): (1, [ 1 1 ] )

1 (key, [pageNo, slotNo]): (2, [ 2 2 ] )

2 (key, [pageNo, slotNo]): (3, [ 3 3 ] )

3 (key, [pageNo, slotNo]): (4, [ 4 4 ] )

4 (key, [pageNo, slotNo]): (5, [ 5 5 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :11

Please input the LOWER integer key(>=0):

2

Please input the HIGHER integer key(>=0)

10

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :3

---------------The B+ Tree Leaf Pages---------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 13

Left Link : -1

Right Link : -1

0 (key, [pageNo, slotNo]): (0, [ 0 0 ] )

1 (key, [pageNo, slotNo]): (1, [ 1 1 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

------------- All Leaf Pages Have Been Printed --------

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :5

Please input the integer key to insert:

2

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :5

Please input the integer key to insert:

3

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :5

Please input the integer key to insert:

4

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :5

Please input the integer key to insert:

5

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :12

Please input the LOWER integer key (null if -3):

2

Please input the HIGHER integer key (null if -2):

4

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :13

SCAN RESULT: 2 [ 2 2 ]

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :14

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :15

Please input the number of keys to insert:

2

Please input the number of keys to delete:

1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The file name is: AAA7 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :16

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* You close the file: AAA7 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :17

4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* You open the file: AAA4 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :18

4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* You destroy the file: AAA4 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Naive delete (new file)

[1] Full delete(Default) (new file)

[2] Print the B+ Tree Structure

[3] Print All Leaf Pages

[4] Choose a Page to Print

---Integer Key (for choices [6]-[14]) ---

[5] Insert a Record

[6] Delete a Record

[7] Test1 (new file): insert n records in order

[8] Test2 (new file): insert n records in reverse order

[9] Test3 (new file): insert n records in random order

[10] Test4 (new file): insert n records in random order

and delete m records randomly

[11] Delete some records

[12] Initialize a Scan

[13] Scan the next Record

[14] Delete the just-scanned record

---String Key (for choice [15]) ---

[15] Test5 (new file): insert n records in random order

and delete m records randomly.

[16] Close the file

[17] Open which file (input an integer for the file name):

[18] Destroy which file (input an integer for the file name):

[19] Quit!

Hi, make your choice :19

... Finished .

/usr/lib/java/jdk-9.0.4/bin/javac -classpath .:.. TestDriver.java IndexTest.java

/usr/lib/java/jdk-9.0.4/bin/java -classpath .:.. tests.IndexTest

Running Index tests....

Replacer: Clock

------------------------ TEST 1 --------------------------

BTreeIndex created successfully.

BTreeIndex file created successfully.

Test1 -- Index Scan OK

------------------- TEST 1 completed ---------------------

------------------------ TEST 2 --------------------------

BTreeIndex opened successfully.

Test2 -- Index Scan OK

------------------- TEST 2 completed ---------------------

------------------------ TEST 3 --------------------------

BTreeIndex created successfully.

BTreeIndex file created successfully.

Test3 -- Index scan on int key OK

------------------- TEST 3 completed ---------------------

...Index tests

completely successfully

.

Index tests completed successfully

/usr/lib/java/jdk-9.0.4/bin/javac -classpath .:.. TestDriver.java JoinTest.java

Note: JoinTest.java uses unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.

/usr/lib/java/jdk-9.0.4/bin/java -classpath .:.. tests.JoinTest

Replacer: Clock

Any resemblance of persons in this database to people living or dead

is purely coincidental. The contents of this database do not reflect

the views of the University, the Computer Sciences Department or the

developers...

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query1 strating \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query: Find the names of sailors who have reserved boat number 1.

and print out the date of reservation.

SELECT S.sname, R.date

FROM Sailors S, Reserves R

WHERE S.sid = R.sid AND R.bid = 1

(Tests FileScan, Projection, and Sort-Merge Join)

[Mike Carey, 05/10/95]

[David Dewitt, 05/11/95]

[Jeff Naughton, 05/12/95]

Query1 completed successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query1 finished!!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query2 strating \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query: Find the names of sailors who have reserved a red boat

and return them in alphabetical order.

SELECT S.sname

FROM Sailors S, Boats B, Reserves R

WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = 'red'

ORDER BY S.sname

Plan used:

Sort (Pi(sname) (Sigma(B.color='red') |><| Pi(sname, bid) (S |><| R)))

(Tests File scan, Index scan ,Projection, index selection,

sort and simple nested-loop join.)

After Building btree index on sailors.sid.

[David Dewitt]

[Mike Carey]

[Raghu Ramakrishnan]

[Yannis Ioannidis]

Query2 completed successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query2 finished!!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query3 strating \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query: Find the names of sailors who have reserved a boat.

SELECT S.sname

FROM Sailors S, Reserves R

WHERE S.sid = R.sid

(Tests FileScan, Projection, and SortMerge Join.)

[Mike Carey]

[Mike Carey]

[Mike Carey]

[David Dewitt]

[David Dewitt]

[Jeff Naughton]

[Miron Livny]

[Yannis Ioannidis]

[Raghu Ramakrishnan]

[Raghu Ramakrishnan]

Query3 completed successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query3 finished!!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query4 strating \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query: Find the names of sailors who have reserved a boat

and print each name once.

SELECT DISTINCT S.sname

FROM Sailors S, Reserves R

WHERE S.sid = R.sid

(Tests FileScan, Projection, Sort-Merge Join and Duplication elimination.)

[David Dewitt]

[Jeff Naughton]

[Mike Carey]

[Miron Livny]

[Raghu Ramakrishnan]

[Yannis Ioannidis]

Query4 completed successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query4 finished!!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query5 strating \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query: Find the names of old sailors or sailors with a rating less

than 7, who have reserved a boat, (perhaps to increase the

amount they have to pay to make a reservation).

SELECT S.sname, S.rating, S.age

FROM Sailors S, Reserves R

WHERE S.sid = R.sid and (S.age > 40 || S.rating < 7)

(Tests FileScan, Multiple Selection, Projection, and Sort-Merge Join.)

[Mike Carey, 9, 40.3]

[Mike Carey, 9, 40.3]

[Mike Carey, 9, 40.3]

[David Dewitt, 10, 47.2]

[David Dewitt, 10, 47.2]

[Jeff Naughton, 5, 35.0]

[Yannis Ioannidis, 8, 40.2]

Query5 completed successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query5 finished!!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query6 strating \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query: Find the names of sailors with a rating greater than 7

who have reserved a red boat, and print them out in sorted order.

SELECT S.sname

FROM Sailors S, Boats B, Reserves R

WHERE S.sid = R.sid AND S.rating > 7 AND R.bid = B.bid

AND B.color = 'red'

ORDER BY S.name

Plan used:

Sort(Pi(sname) (Sigma(B.color='red') |><| Pi(sname, bid) (Sigma(S.rating > 7) |><| R)))

(Tests FileScan, Multiple Selection, Projection,sort and nested-loop join.)

After nested loop join S.sid|><|R.sid.

After nested loop join R.bid|><|B.bid AND B.color=red.

After sorting the output tuples.

[David Dewitt]

[Mike Carey]

[Raghu Ramakrishnan]

[Yannis Ioannidis]

Query6 completed successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query6 finished!!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Finished joins testing

join tests completed successfully

/usr/lib/java/jdk-9.0.4/bin/javac -classpath .:.. TestDriver.java SortTest.java

/usr/lib/java/jdk-9.0.4/bin/java -classpath .:.. tests.SortTest

Running Sort tests....

Replacer: Clock

------------------------ TEST 1 --------------------------

Test1 -- Sorting OK

------------------- TEST 1 completed ---------------------

------------------------ TEST 2 --------------------------

Test2 -- Sorting OK

------------------- TEST 2 completed ---------------------

------------------------ TEST 3 --------------------------

-- Sorting in ascending order on the int field --

Test3 -- Sorting of int field OK

-- Sorting in descending order on the float field --

Test3 -- Sorting of float field OK

------------------- TEST 3 completed ---------------------

------------------------ TEST 4 --------------------------

Test4 -- Sorting OK

------------------- TEST 4 completed ---------------------

...Sort tests

completely successfully

.

Sorting tests completed successfully

/usr/lib/java/jdk-9.0.4/bin/javac -classpath .:.. SM\_JoinTest.java TestDriver.java

Note: SM\_JoinTest.java uses unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.

/usr/lib/java/jdk-9.0.4/bin/java -classpath .:.. tests.SM\_JoinTest

Replacer: Clock

Any resemblance of persons in this database to people living or dead

is purely coincidental. The contents of this database do not reflect

the views of the University, the Computer Sciences Department or the

developers...

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query1 strating \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query: Find the names of sailors who have reserved boat number 1.

and print out the date of reservation.

SELECT S.sname, R.date

FROM Sailors S, Reserves R

WHERE S.sid = R.sid AND R.bid = 1

(Tests FileScan, Projection, and Sort-Merge Join)

[Mike Carey, 05/10/95]

[David Dewitt, 05/11/95]

[Jeff Naughton, 05/12/95]

Query1 completed successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query1 finished!!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query3 strating \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query: Find the names of sailors who have reserved a boat.

SELECT S.sname

FROM Sailors S, Reserves R

WHERE S.sid = R.sid

(Tests FileScan, Projection, and SortMerge Join.)

[Mike Carey]

[Mike Carey]

[Mike Carey]

[David Dewitt]

[David Dewitt]

[Jeff Naughton]

[Miron Livny]

[Yannis Ioannidis]

[Raghu Ramakrishnan]

[Raghu Ramakrishnan]

Query3 completed successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query3 finished!!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query4 strating \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query: Find the names of sailors who have reserved a boat

and print each name once.

SELECT DISTINCT S.sname

FROM Sailors S, Reserves R

WHERE S.sid = R.sid

(Tests FileScan, Projection, Sort-Merge Join and Duplication elimination.)

[David Dewitt]

[Jeff Naughton]

[Mike Carey]

[Miron Livny]

[Raghu Ramakrishnan]

[Yannis Ioannidis]

Query4 completed successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query4 finished!!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query5 strating \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Query: Find the names of old sailors or sailors with a rating less

than 7, who have reserved a boat, (perhaps to increase the

amount they have to pay to make a reservation).

SELECT S.sname, S.rating, S.age

FROM Sailors S, Reserves R

WHERE S.sid = R.sid and (S.age > 40 || S.rating < 7)

(Tests FileScan, Multiple Selection, Projection, and Sort-Merge Join.)

[Mike Carey, 9, 40.3]

[Mike Carey, 9, 40.3]

[Mike Carey, 9, 40.3]

[David Dewitt, 10, 47.2]

[David Dewitt, 10, 47.2]

[Jeff Naughton, 5, 35.0]

[Yannis Ioannidis, 8, 40.2]

Query5 completed successfully!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Query5 finished!!!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Finished joins testing

join tests completed successfully

make[1]: Leaving directory '/home/spoorthi/Downloads/minjava/javaminibase/src/tests'

]0;spoorthi@spoorthi-VirtualBox: ~/Downloads/minjava/javaminibase/src[01;32mspoorthi@spoorthi-VirtualBox[00m:[01;34m~/Downloads/minjava/javaminibase/src[00m$ exit

Script done on Sun 04 Feb 2018 08:01:58 PM MST