Group No: 15
Ameya Hujare (A20545367)
Deep Pawar (A20545137)
Canyu Chen (A20479758)
Professor: Gerald Balekaki
Institute: Illinois Institute of Technology

CS 525: Advanced Database Organization

Spring 2025 - Assignment 4 - B+ Tree Implementation

1. INTRODUCTION

This assignment focuses on the implementation of a B+-Tree index. The B+-Tree is structured as a paged index, where each node of the tree occupies a full page. The implementation integrates with a buffer manager, ensuring efficient disk access. The index supports integer keys (DT_INT) and follows specific conventions for handling insertions, deletions, and rebalancing operations.

2. B+-TREE OVERVIEW

The B+-Tree is an ordered, balanced tree structure optimized for search, insert, and delete operations. Key functionalities include:

- **Index Management:** Creating, opening, and deleting a B+-Tree index.
- **Node Structure:** Each node follows a fixed structure, supporting a defined fan-out.
- **Insertion & Deletion Handling:** Proper redistribution and merging mechanisms are implemented.
- **Buffer Manager Integration:** Ensures pages are efficiently loaded and written back to disk.

3. FUNCTIONALITIES AND CONCEPTS

3.1 B+-Tree Node Representation

- Leaf nodes store key-pointer pairs, where the pointer represents a record identifier (RID).
- Internal nodes store key-page number pairs, directing searches down the tree.
- The root node may function as either a leaf or an internal node, depending on the number of keys.

3.2 Page Structure

- Each page in the index file corresponds to a B+-Tree node.
- Nodes contain metadata such as the number of keys, type (leaf/internal), and child pointers.

3.3 Insert Operation

- Inserts are performed by traversing the tree to locate the correct leaf node.
- If a node overflows, it is split, with the middle key promoted to the parent.
- The left node retains extra keys in the case of an even split.

3.4 Delete Operation

- If a node underflow, values are first redistributed from siblings.
- If redistribution is not possible, the node is merged with a sibling.
- Merges prioritize left siblings for consistency.

3.5 Search and Scan Operations

- Searches are performed by traversing the tree using key comparisons.
- The scan function enables in-order traversal of entries.

4. INTERFACE AND IMPLEMENTATION

4.1 Data Structures

- **BTreeHandle:**
  Represents the index structure.

- **BT_ScanHandle:**
  Manages active tree scans.

4.2 Index Manager Functions

- **initIndexManager():**
  Initializes the index manager.

- **shutdownIndexManager():**
  Frees allocated resources.

4.3 B+-Tree Operations

- **createBtree():**
  Creates a B+-Tree with a specified order.

- **openBtree() / closeBtree():**
  Manages access to index structures.

- **insertKey():**
  Inserts a key into the tree.

- **deleteKey():**
  Deletes a key from the tree.

- **findKey():**
  Searches for a key and retrieves the corresponding RID.

- **openTreeScan() / nextEntry() / closeTreeScan():**
  Manages in-order scans.

4.4 Debugging Functions

- **printTree():**
  Generates a human-readable representation of the tree.

5. ERROR HANDLING AND DEBUGGING

- **Error Codes:**
  Defined in dberror.h for consistent error reporting.

- **Debugging Methods:**
  Includes print functions for tree structure and node contents.

6. SOURCE CODE STRUCTURE

The project directory follows this structure:

```
assign4/
├── README.md
├── Makefile
├── buffer_mgr.h
├── buffer_mgr.c
├── buffer_mgr_stat.c
├── buffer_mgr_stat.h
├── btree_mgr.h
├── btree_mgr.c
├── dberror.h
├── dberror.c
├── dt.h
├── expr.h
├── expr.c
├── record_mgr.h
├── record_mgr.c
├── replacement_strat.c
├── replacement_strat.h
├── storage_mgr.h
├── storage_mgr.c
├── tables.h
├── test_assign4_1.c
├── test_expr.c
└── test_helper.h
```

7. TESTING AND VALIDATION

Test cases ensure the correctness of:

- **Basic B+-Tree operations** (Insert, Delete, Search)
- **Index structure management** (Splitting, Merging, Redistribution)
- **Tree Scans** (Sequential traversal of entries)

8. OPTIONAL EXTENSIONS ADDED

- **Pointer Swizzling:**
  Optimizes memory access by replacing disk page references with direct memory pointers.

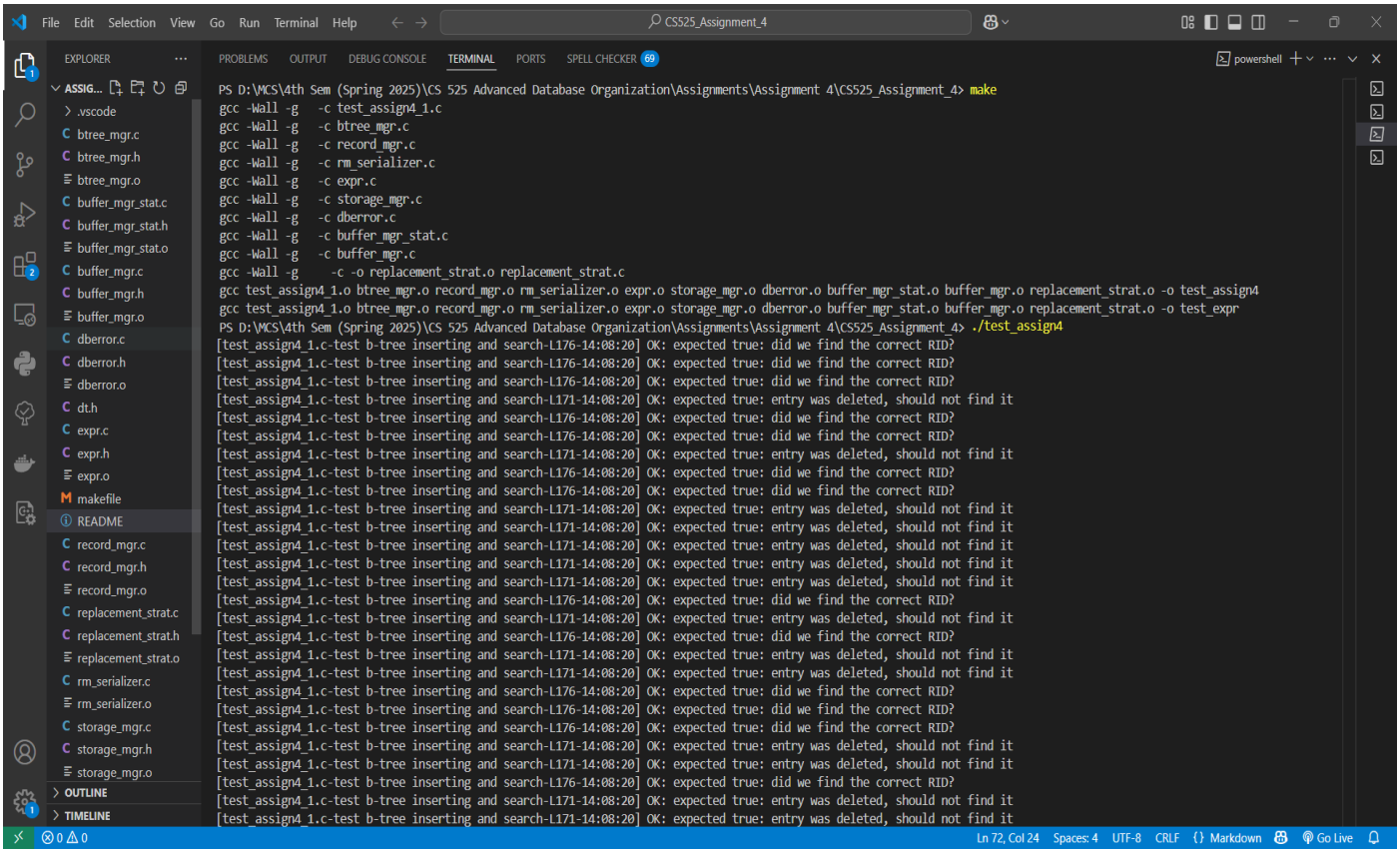
- **Integration with Record Manager:**
  Enables direct indexing support for table attributes.

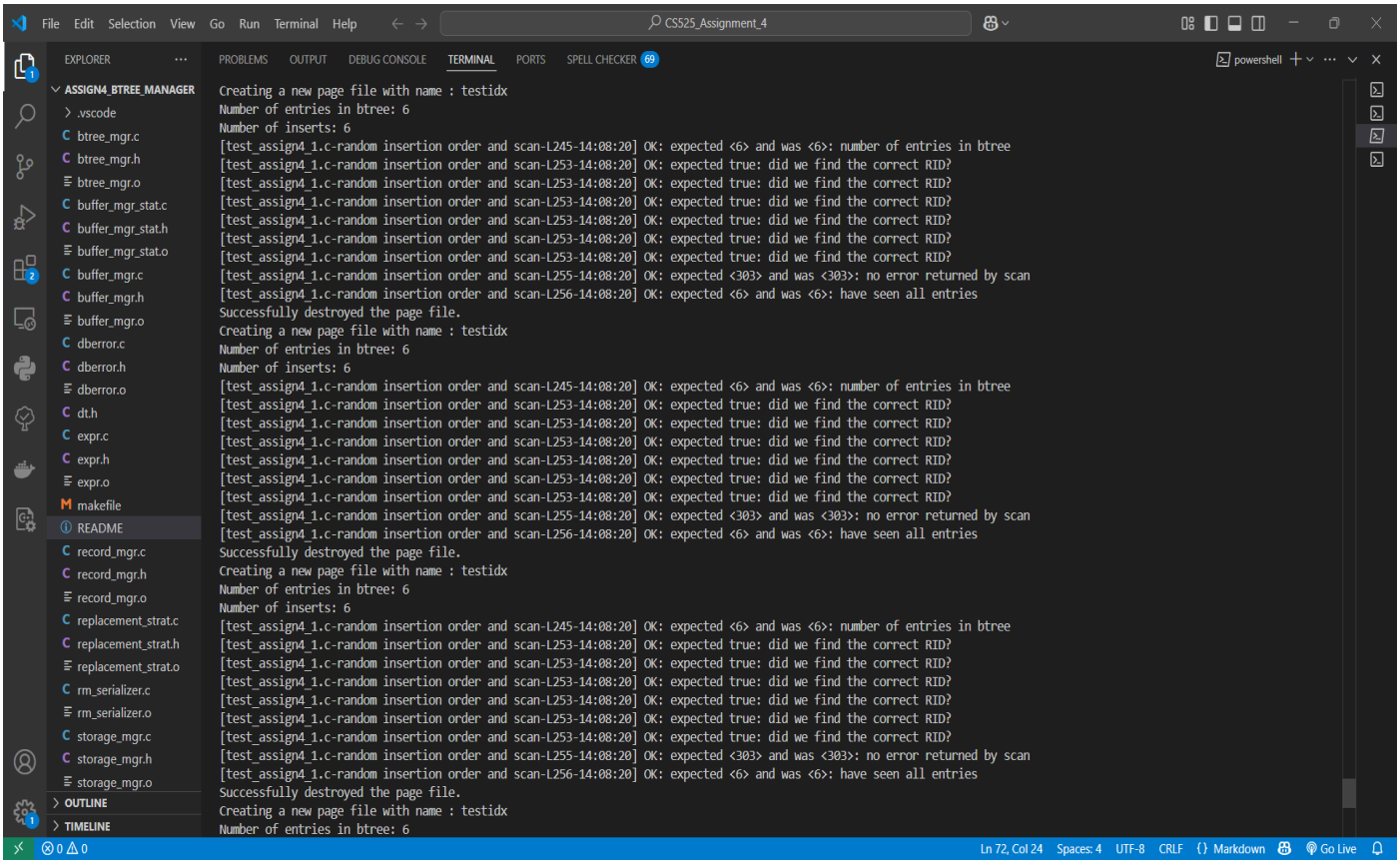
9. CONCLUSION

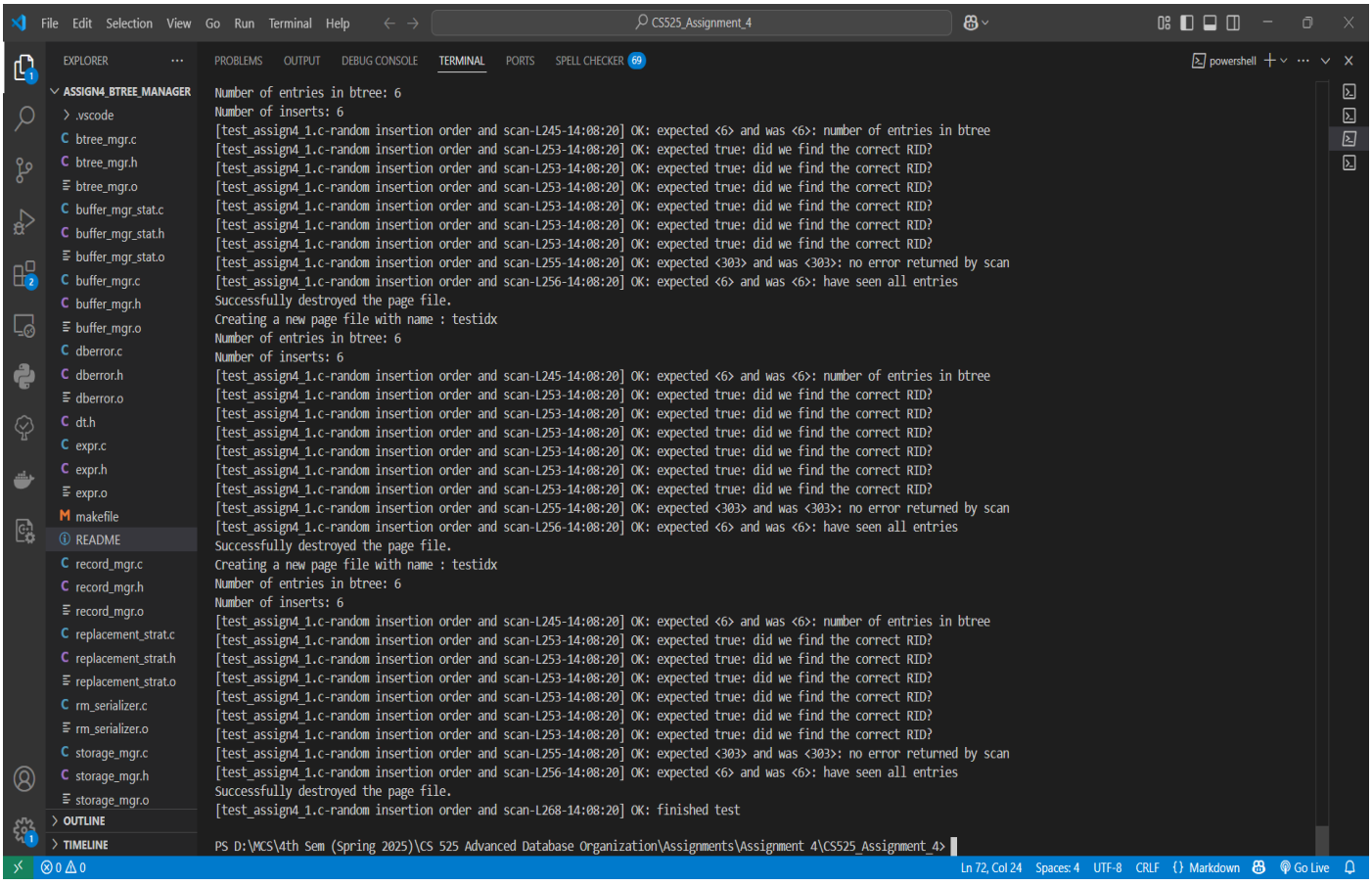
This assignment successfully implements a B+-Tree index structure, providing efficient search, insert, and delete functionalities. The integration with a buffer manager enhances performance by optimizing disk accesses. Additional features such as pointer swizzling and multiple entry support further improve efficiency and scalability.

10. OUTPUT

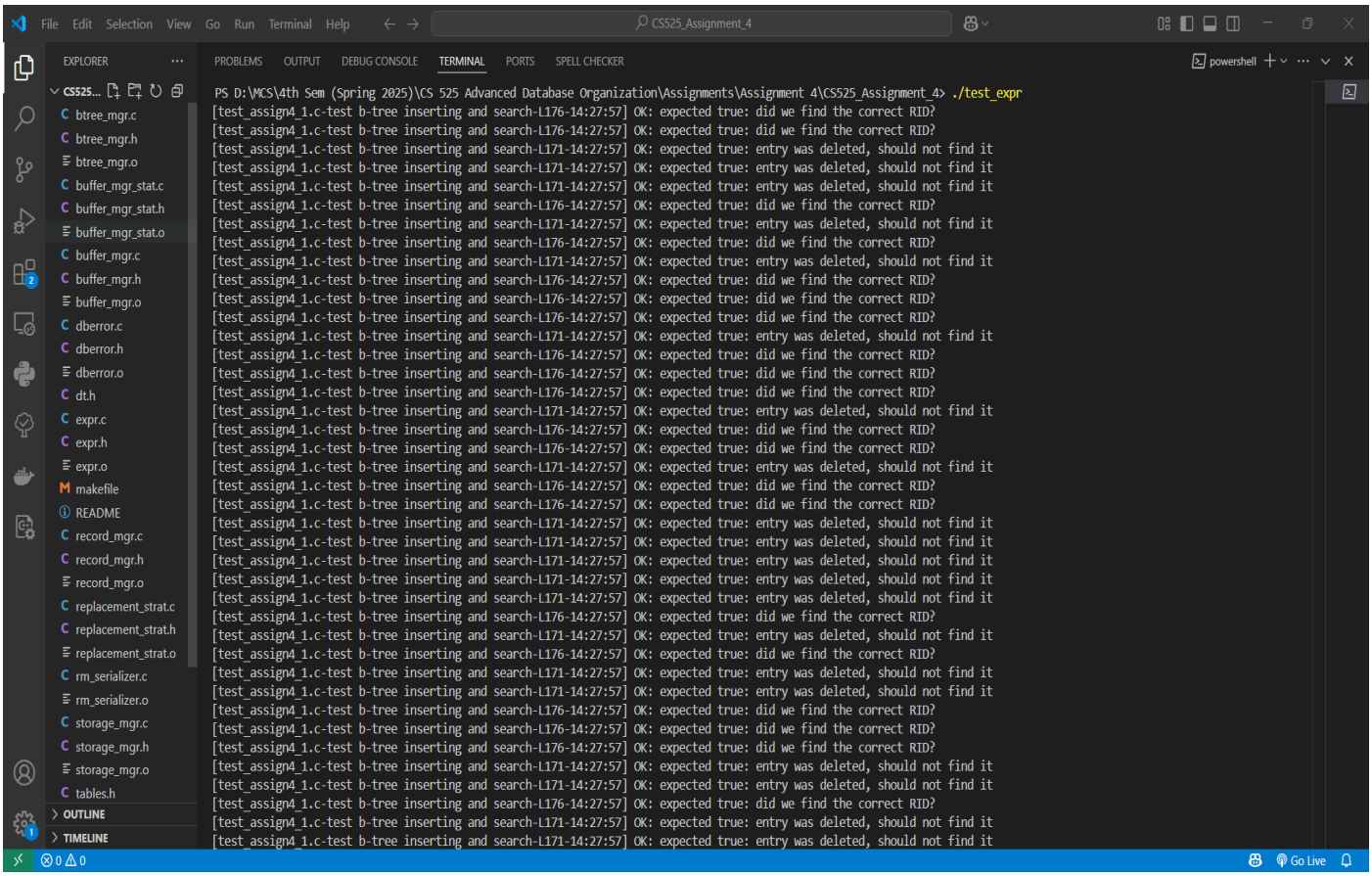
a. Executing test cases for the B+ tree functions

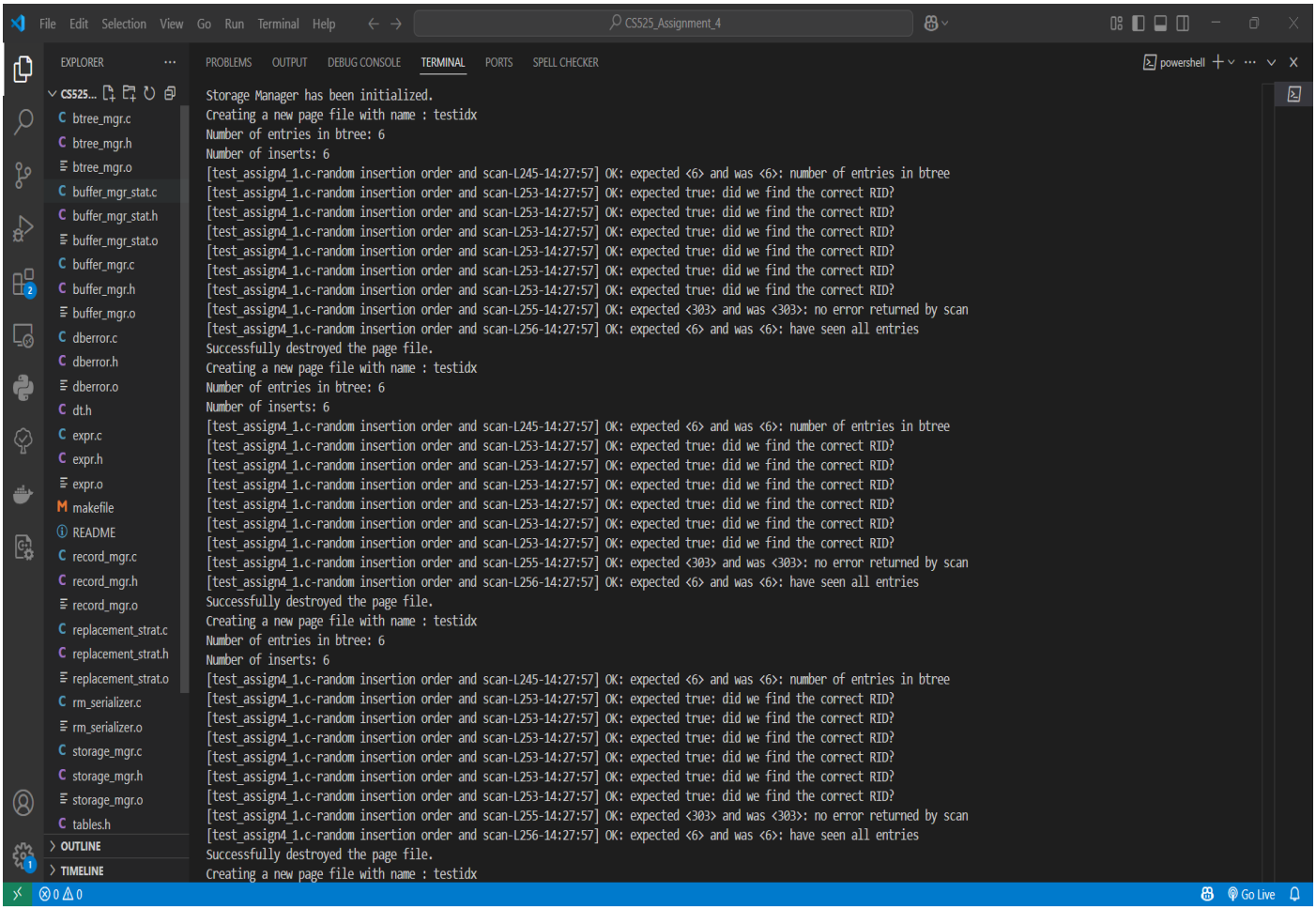


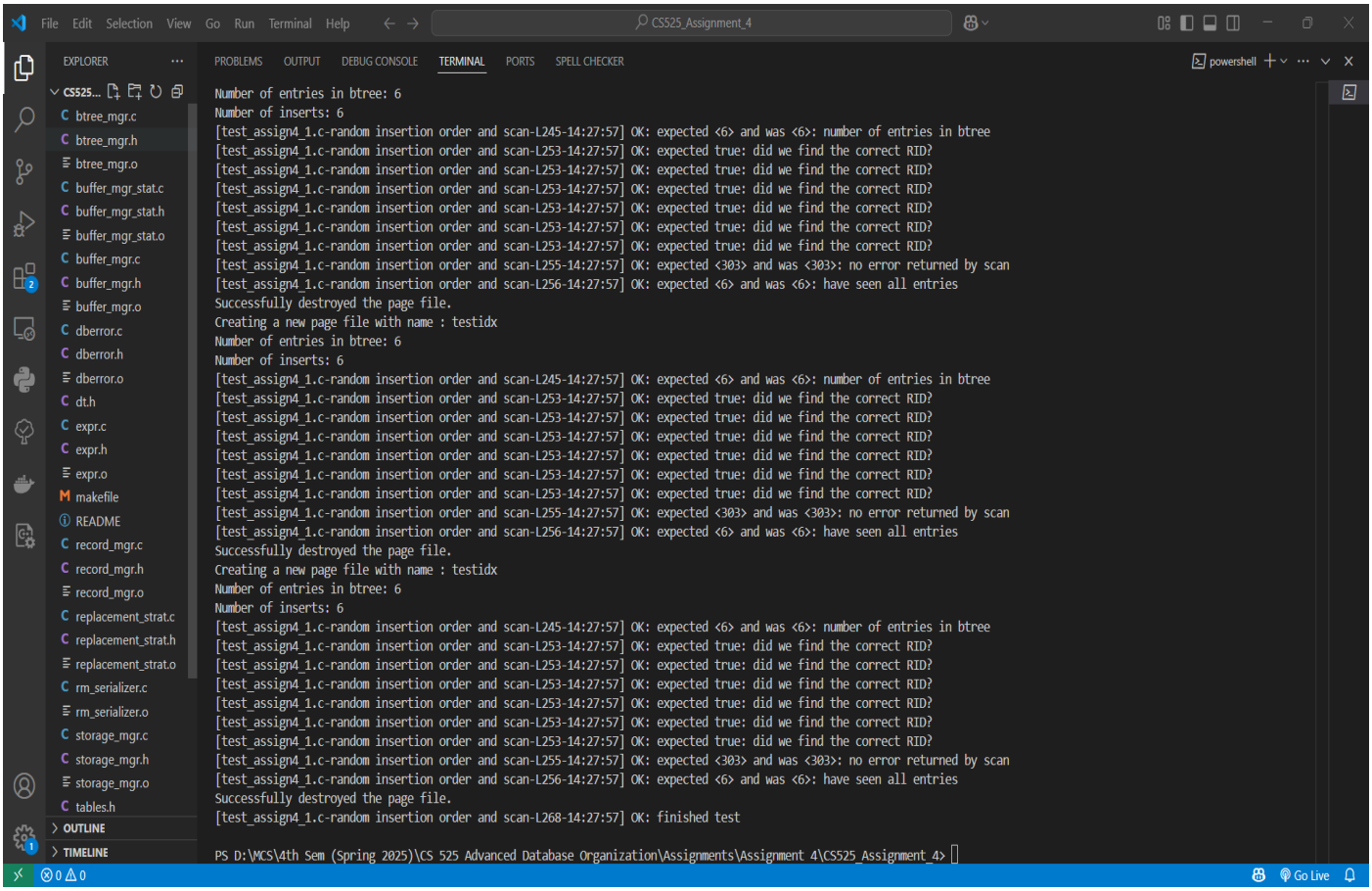

File Edit Selection View Go Run Terminal Help
CS525_Assignment_4
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 69
powershell
ASSIGN4_BTREE_MANAGER
.vscode
btree_mgr.c
btree_mgr.h
btree_mgr.o
buffer_mgr_stat.c
buffer_mgr_stat.h
buffer_mgr_stat.o
buffer_mgr.c
buffer_mgr.h
buffer_mgr.o
dberror.c
dberror.h
dberror.o
dt.h
expr.c
expr.h
expr.o
makefile
README
record_mgr.c
record_mgr.h
record_mgr.o
replacement_strat.c
replacement_strat.h
replacement_strat.o
rm_serializer.c
rm_serializer.o
storage_mgr.c
storage_mgr.h
storage_mgr.o
OUTLINE
TIMELINE
Creating a new page file with name : testidx
Number of entries in btree: 6
Number of inserts: 6
[test_assign4_1.c-random insertion order and scan-L245-14:08:20] OK: expected <6> and was <6>: number of entries in btree
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L255-14:08:20] OK: expected <303> and was <303>: no error returned by scan
[test_assign4_1.c-random insertion order and scan-L256-14:08:20] OK: expected <6> and was <6>: have seen all entries
Successfully destroyed the page file.
Creating a new page file with name : testidx
Number of entries in btree: 6
Number of inserts: 6
[test_assign4_1.c-random insertion order and scan-L245-14:08:20] OK: expected <6> and was <6>: number of entries in btree
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L255-14:08:20] OK: expected <303> and was <303>: no error returned by scan
[test_assign4_1.c-random insertion order and scan-L256-14:08:20] OK: expected <6> and was <6>: have seen all entries
Successfully destroyed the page file.
Creating a new page file with name : testidx
Number of entries in btree: 6
Number of inserts: 6
[test_assign4_1.c-random insertion order and scan-L245-14:08:20] OK: expected <6> and was <6>: number of entries in btree
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L255-14:08:20] OK: expected <303> and was <303>: no error returned by scan
[test_assign4_1.c-random insertion order and scan-L256-14:08:20] OK: expected <6> and was <6>: have seen all entries
Successfully destroyed the page file.
Creating a new page file with name : testidx
Number of entries in btree: 6
Ln 72, Col 24 Spaces: 4 UTF-8 CRLF {} Markdown Go Live


File Edit Selection View Go Run Terminal Help
CS525_Assignment_4
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 69
powershell
ASSIGN4_BTREE_MANAGER
.vscode
btree_mgr.c
btree_mgr.h
btree_mgr.o
buffer_mgr_stat.c
buffer_mgr_stat.h
buffer_mgr_stat.o
buffer_mgr.c
buffer_mgr.h
buffer_mgr.o
dberror.c
dberror.h
dberror.o
dt.h
expr.c
expr.h
expr.o
makefile
README
record_mgr.c
record_mgr.h
record_mgr.o
replacement_strat.c
replacement_strat.h
replacement_strat.o
rm_serializer.c
rm_serializer.o
storage_mgr.c
storage_mgr.h
storage_mgr.o
OUTLINE
TIMELINE
Number of entries in btree: 6
Number of inserts: 6
[test_assign4_1.c-random insertion order and scan-L245-14:08:20] OK: expected <6> and was <6>: number of entries in btree
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L255-14:08:20] OK: expected <303> and was <303>: no error returned by scan
[test_assign4_1.c-random insertion order and scan-L256-14:08:20] OK: expected <6> and was <6>: have seen all entries
Successfully destroyed the page file.
Creating a new page file with name : testidx
Number of entries in btree: 6
Number of inserts: 6
[test_assign4_1.c-random insertion order and scan-L245-14:08:20] OK: expected <6> and was <6>: number of entries in btree
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L255-14:08:20] OK: expected <303> and was <303>: no error returned by scan
[test_assign4_1.c-random insertion order and scan-L256-14:08:20] OK: expected <6> and was <6>: have seen all entries
Successfully destroyed the page file.
Creating a new page file with name : testidx
Number of entries in btree: 6
Number of inserts: 6
[test_assign4_1.c-random insertion order and scan-L245-14:08:20] OK: expected <6> and was <6>: number of entries in btree
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:08:20] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L255-14:08:20] OK: expected <303> and was <303>: no error returned by scan
[test_assign4_1.c-random insertion order and scan-L256-14:08:20] OK: expected <6> and was <6>: have seen all entries
Successfully destroyed the page file.
[test_assign4_1.c-random insertion order and scan-L268-14:08:20] OK: finished test
PS D:\MCS\4th Sem (Spring 2025)\CS 525 Advanced Database Organization\Assignments\Assignment 4\CS525_Assignment_4>
Ln 72, Col 24 Spaces: 4 UTF-8 CRLF {} Markdown Go Live

b. Executing test cases for the testing evaluation part of the B+ tree






File Edit Selection View Go Run Terminal Help
CS525_Assignment_4
EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER
powershell
CS525...
btree_mgr.c
btree_mgr.h
btree_mgr.o
buffer_mgr_stat.c
buffer_mgr_stat.h
buffer_mgr_stat.o
buffer_mgr.c
buffer_mgr.h
buffer_mgr.o
dberror.c
dberror.h
dberror.o
dt.h
expr.c
expr.h
expr.o
makefile
README
record_mgr.c
record_mgr.h
record_mgr.o
replacement_strat.c
replacement_strat.h
replacement_strat.o
rm_serializer.c
rm_serializer.o
storage_mgr.c
storage_mgr.h
storage_mgr.o
tables.h
OUTLINE
TIMELINE
Number of entries in btree: 6
Number of inserts: 6
[test_assign4_1.c-random insertion order and scan-L245-14:27:57] OK: expected <6> and was <6>: number of entries in btree
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L255-14:27:57] OK: expected <303> and was <303>: no error returned by scan
[test_assign4_1.c-random insertion order and scan-L256-14:27:57] OK: expected <6> and was <6>: have seen all entries
Successfully destroyed the page file.
Creating a new page file with name : testidx
Number of entries in btree: 6
Number of inserts: 6
[test_assign4_1.c-random insertion order and scan-L245-14:27:57] OK: expected <6> and was <6>: number of entries in btree
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L255-14:27:57] OK: expected <303> and was <303>: no error returned by scan
[test_assign4_1.c-random insertion order and scan-L256-14:27:57] OK: expected <6> and was <6>: have seen all entries
Successfully destroyed the page file.
Creating a new page file with name : testidx
Number of entries in btree: 6
Number of inserts: 6
[test_assign4_1.c-random insertion order and scan-L245-14:27:57] OK: expected <6> and was <6>: number of entries in btree
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L253-14:27:57] OK: expected true: did we find the correct RID?
[test_assign4_1.c-random insertion order and scan-L255-14:27:57] OK: expected <303> and was <303>: no error returned by scan
[test_assign4_1.c-random insertion order and scan-L256-14:27:57] OK: expected <6> and was <6>: have seen all entries
Successfully destroyed the page file.
[test_assign4_1.c-random insertion order and scan-L268-14:27:57] OK: finished test
PS D:\MCS\4th Sem (Spring 2025)\CS 525 Advanced Database Organization\Assignments\Assignment 4\CS525_Assignment_4>
0 0
Go Live