Names: Kevin Gutierrez, Kiran Kadariya, Sagar N, Abdul

CSCI 331-54

Professor: Andrew Anda

Date: 4/24/25

The program manages a blocked sequence set and B+ Tree using CSV-based data. It will do:

* CSV conversion into fixed-sized blocks
* B+ Tree construction for efficient record search
* Dynamic insertion, deletion, and search
* Debugging via dumps

**Objectives**

Input: CSV file containing structured records

Output: Blocked sequence set file blocked\_sequence\_set.txt and B+ Tree representation

**Functional Requirements**

CSV Handling: Read, parse, and store structured records Block Creation: Fixed-size storage with logical linking B+ Tree Construction: Leaf block initialization, hierarchical index creation Dynamic Operations: Insert, delete, search records Debugging: Dump blocks and tree structure

**Core Logic**

CSV Parser readCSV: Loads structured records from a CSV file Blocked Sequence Set createBlocks: Generates sequentially numbered data blocks Dump Functions: dumpPhysical outputs physical block structure, dumpLogical displays logical sequence B+ Tree BPlusTree: Constructs tree hierarchy, supports insertions and queries

**Order of Operations**

Step 1: Parse command-line arguments

Step 2: Read CSV file and process records

Step 3: Convert records into data blocks

Step 4: Build B+ Tree from blocks

Step 5: Perform dynamic operations

Step 6: Debugging

Step 7: Output blocked sequence file and tree

**Entities**

**Block**: Represents a storage unit with blockNumber sequential ID, nextBlock logical pointer, records data list

**BlockBuffer**: Manages file operations with writeBlocks and readBlocks Record: Holds structured data with index primary key, field1, field2, field3

**BPlusTree**: Handles buildTree for constructing hierarchy, insert for adding new records, dumpTree for displaying structure

**General Structure of the logic**

Block Creation: Parse records, allocate sequential blocks, assign logical pointers Tree Construction: Treat leaf blocks as base, extract largest keys for parent index, recursively aggregate into a root node Insertion: Locate correct block, insert record in sorted order, handle overflow by splitting

**General plan**

1. Develop CSV handling functions (recycle from last project)
2. Implement block management
3. Rebuild Block and BlockBuffer
4. Build B+ Tree structure
5. Support debugging output Validate tree operations with test cases