SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

50.043 Database Lab 1 Report

Name	Student ID
Matthaeus Choo Zhi Jie (Zhu Zhijie)	1008103
Lee Jing Kang	1008240
Chong Juo Ru Faustina	1007967

50.043 Database System and Big Data ~ Lab 1

Summary and Design decisions

Exercise 1 ~ Tuple and TupleDesc

The Tuple class maintains information about the content of a tuple, and has a schema specified by a TupleDesc object using a field array to contain data for each field. Both objects contain getter and setter functions for each field value, handling changes to the schema and managing RecordId for tuple location. TupleDesc has an inner class TDitem to help organize tuple information, retrieve field details, calculate tuple sizes and merger descriptors. Altogether these choices in design prioritize simplicity, validation, and fixed-size tuple handling for database operations.

Exercise 2 ~ Catalog

The Catalog class stores a list of the tables and schemas of the tables that are currently in the database. This is achieved by storing the table file, name and pkeyfield in a Table object that is stored in a ConcurrentHashMap to map table IDs to Table objects. Table objects contain DbFile, name and pkeyField information. Catalog objects contain methods to manage tables and their associated schemas such as adding tables, retrieving table information, and clearing the catalog. Catalog objects are also able to load schemas from a file to create the appropriate tables in the database.

Exercise 3 ~ Bufferpool

The Bufferpool class manages the reading and writing of pages from the disk. The Bufferpool constructor creates a ConcurrentHashmap of the size of the number of pages on the input that maps the pageId to each page. The concurrentHashmap is used as there might be multiple requests for the bufferpool at the same time, which this hashmap helps to manage. When a page is requested from the bufferpool object, the getPage function checks the pagesMap hashmap if there exists a associated page with the inputted pageId to return to the caller; If there isn’t, the, the bufferpool then searches for the page on the database.

Exercise 4 ~ HeapPageId, RecordId, HeapPage

The HeapPageId class identifies the pages within the tables by storing the table ID and page numbers. The RecordID class uses the combined PageId and tuple number to pinpoint the exact location of the tuple in the page. The HeapPage class stores the data for a single page of tuples in HeapFile. The HeapPage class manages the creation, storage and retrieval of tuples in Heapfile. The class also tracks which slots are used or empty and provides methods for inserting and deletion of tuples (Currently not implemented).

Exercise 5 ~ HeapFile

The HeapFile class is a collection of pages which stores tuples in a file on the disk. There are methods for reading and writing to pages as well as to iterator to iterate over all the tuples in the file. We also used the RandomAccessFile API to access specific pages after calculating the offset using pageSize and pageNumber. In order to adhere to the Single Responsibility Principle, we have switched from implementing HeapFileIterator class in the same file to a different file. This ensures that HeapFile class and HeapFile Iterator class each have their own clear and distinct purpose. The purpose of HeapFileIterator is to read the tuples across the different pages given a transaction.

Exercise 6 ~ SeqScan

The SeqScan class scans over the specified table as part of a specific transaction. It does through an iterator interface to sequentially scan over all the tuples that are in the table.

Changes Made to API

Exercise 2 ~ Table class

The Table class is used in exercise 2 to consolidate the table details such as file, name and primary key into a single object. This design improves the readability and maintainability of the code as it reduces the number of data structures needed to manage the table.

Missing/Incomplete elements of our Code

These are the areas of our code that are currently missing/incomplete as they are currently out of the scope of Lab 1.

1. Locking mechanism for the bufferpool
2. Dirty or clean pages
3. Eviction Policy
4. Transaction Support