

BOĞAZİÇİ UNIVERSITY

CMPE 321

PROJECT 1

Implementing A Store Manager System

SPRING 2020

Yunus Kardaş

March 27, 2020

Contents

1	Introduction	2
2	Changes From the Initial Design	2
3	Conclusions & Assessment	2

1 Introduction

In this project, I am expected to implement a storage manager system that supports DDL operations and DML. There should be a system catalogue which stores metadata and multiple data files that store the actual data. This document explains my implementation by showing how my design is different from the first project and my actual code. My implementation contains one StoreManager.java file and a SystemCatalogue.java file. In StoreManager, system catalogue is called and all of the procedures are done in the system catalogue. I have used RandomAccessFile to write bytes to the files and File and FileReader to take input from the user via text file. I have done almost no error checking because description says that user has to input valid data. I have 8 functions for each procedure and one helper function to determine whether that record type already exists or not.

2 Changes From the Initial Design

A record type name is at most 10 characters.

-Because my code will run in a 10-Mb disc drive, seemed memory consuming to have 12 character long type names.

I am not keeping isEmpty information in a record

-Because I already have an isDeleted info, seemed memory consuming to have them both

I am not keeping a pointer to the next page in the header.

-Because both pages are in the same file, I can use byte count to find the next page.

3 Conclusions & Assessment

In this project, I have implemented a simple storage manager which has a system catalogue file and data files. In my design each file can hold at most one record type and can have at most 10 pages. This makes accessing a record with its primary key faster but insertion is slower since we have to access a specific page to insert a record. Since we didn't do any error checking, if a user enters a wrong input, this storage manager cannot handle it. To sum up, this is a really simple storage manager design and it has its own pros

and cons. But mostly, it is very efficient while accessing a record but not so much while insertion. But we can modify this design and improve it.