

CMPE 321 Assignment I

Data on disk is stored in data structures called “pages”. Each page begins with a page header which is followed by records. Each record, in turn, begins with a record header followed by the fields of that record.

Page header			
Record #1 header	Record #1 Field #1	Record #1 Field #2	Record #1 Field #3
Record #2 header	Record #2 Field #1	Record #2 Field #2	Record #2 Field #3
Record #3 header	Record #3 Field #1	Record #3 Field #2	Record #3 Field #3
⋮	⋮	⋮	⋮

Table 1: Page structure

You are supposed to design a storage manager system. Most specifications of this design will be decided by you. The design should have a system catalog which stores metadata, and multiple data files that store actual data. You can choose to keep all types in the same page (mixed style) or to allocate a different page for each type (separate style). By the way, the number of types in the system may vary over time (See DDL Operations below).

All fields should be of type integer. Type and field names, however, should be alphanumeric.

You will need to determine some constraints and assumptions. These include, but are not limited to, the following:

1. Page size (Something between 1KB and 2KBs is optimal)
2. What information to store in your page headers and record headers
3. Max number of fields a type can have
4. Max length of a type name
5. Max length of a field name

The user can run 7 operations:

DDL Operations

1. Create a type
2. Delete a type
3. List all types

DML Operations

1. Create a record
2. Delete a record
3. Search for a record (by primary key)
4. List all types of a record

Make reference to various parts of storage structures (such as page header, record header etc) you defined in specifying your algorithms.

For this assignment, you can assume that the disk manager fetches a page when the page address is supplied.

Do not create a new file for a new page. A file should be able to hold multiple pages, but your algorithms should read a file page by page.

Do not employ any error checking. Try to make your design reasonably efficient but do not include extra storage structures for performance improvements. Assume that the user *always* inputs valid data.

Report You are supposed to submit a report that has to have the following sections:

1. Title Page (Course name, semester, assignment title, your name. Do not write the instructor's name)
2. Introduction (Briefly describe your project, its aim, etc.)
3. Assumptions & Constraints (e.g. max number of fields a record can have, max length for a field name)
4. Data Structures (Explain your page design, page header, record header, etc. with diagrams)
5. Operations (Algorithms written in pseudocode. You can use *algorithm2e* or *algorithmicx* packages in L^AT_EX)

6. Conclusions & Assessment (Evaluate your design, considering its ups and downs.)