

CMPE321 Introduction to Database Systems
2016/2017-2

Berk Erol

Storage Manager Implementation

Berk Erol
2014400189

Table of Contents

Introduction	3
Assumptions and Constraints.....	4
System Catalog	4
Type	5
Data Structures	6
System Catalog	6
Type	7
Operations	8
DDL Operations	8
Create a Type.....	8
Delete a Type.....	8
List all Types	9
DML Operations.....	9
Create a Record	9
Delete a Record	10
Update a Record.....	10
Search for a Record	11
List all Records.....	11
Conclusion	12

Introduction

This project is about implementing a storage manager. This document explains the assumptions and constraints required to manage the system & the data structures (with diagrams) and operations (with pseudocodes) required to store the data in the system. This DBMS enables users to run various DDL and DML operations as the followings

- 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. Update a record
 4. Search for a record (by key field)
 - < views all records whose key fields are less than a specific value.
 - > views all records whose key fields are more than a specific value.
 - = views all records whose key fields are equal to a specific value.
 5. List all records of a type

I changed my initial design as the followings

- There is no system catalog header. System name is not used and total number of system catalog pages is calculated from file size.
- System catalog pages store records instead of just type names. These records store number of fields and field names of that type in addition to type name.
- There is no file header, information in file headers is moved to system catalog records.
- Usage status is not a field of system catalog records because system catalog records are physically deleted.
- Number of type pages is not a field of system catalog records because it is calculated from file size.
- System catalog page headers contain page number and number of system catalog records (instead of number of types).
- Each system catalog page has at most 10 records (instead of 35 type names).
- Pages are physically deleted.
- Records are physically deleted. There is just one record type (instead of active and deleted). Deleted records are deprecated because I already shift them to right after deletion whether they have a usage status or not.
- Page header contains page number and number of records (instead of numbers of active and deleted records together).
- Record header contains key field (instead of usage status).
- Each record has at most 10 fields (instead of 16 fields).
- Each page has at most 24 records (instead of 15 records).

Assumptions and Constraints

System Catalog

- There is only one system catalog (Creating or deleting system catalog are not allowed).
- System Catalog file shall have the following format: SystemCatalog.txt
- System Catalog stores pages.
- Each page stores records.
- **Page header**
 - Page number (2 bytes)
 - # of records (2 bytes)
- **Record header**
 - Type name
- **Type name**
 - Each type name shall be unique.
 - Type names shall be alphanumeric.
 - The length of type names shall be at most 27 bytes.
- **Record**
 - # of fields (2 bytes)
 - Field names (70 bytes)
- **Field name**
 - Each field name shall be unique.
 - Field names shall be alphanumeric.
 - The length of field names shall be at most 7 bytes.
- Line endings (10+1) *2= 22 bytes.
- Unused space 8 bytes.
- Size of each record shall be 99 bytes.
- Each page shall have at most 10 records.
- Size of each page shall be 1024 bytes.
- System Catalog shall have at most 99 pages.

Type

- Each type is stored in a different file.
- File shall have the following format: <TypeName>.txt
- File stores all pages for a type.
- Each page stores records.
- **Page header**
 - Page number (2 bytes)
 - # of records (2 bytes)
- **Record header**
 - Key field (4 bytes)
- **Key field**
 - Each key field shall be unique.
 - There shall be one and only one key field in a record.
- Each record stores fields.
- Each field stores data as integers.
- Line endings (24+1) *2=50 bytes.
- Unused space 10 bytes.
- Size of each field shall be at most 4 bytes.
- Each record shall have at most 10 fields.
- Size of each record shall be at most 40 bytes.
- Each page shall have at most 24 records.
- Size of each page shall be 1024 bytes.
- Each type shall have at most 99 pages.

Data Structures

System Catalog

It is the main file of the storage system. It keeps all general information about the storage system and its files. It is reached first for all the operations in the system. There is only one system catalog. Creating or deleting system catalogs are not allowed. System Catalog stores pages, pages store records, records store type name, number of type pages, number of fields and field names. Pages and records can be deleted physically. Each page shall have a fixed size of 1024 bytes but System Catalog can have multiple pages. Pages that are not completely full are filled with spaces to reach 1024 bytes. There shall always be enough number of pages in System Catalog for all records (Pages are created after the last page when there is no empty record slot in the last page and last page is deleted when there is no record in the last page). New records are created at the end. When a record is deleted, all records after that, are shifted by one record to the left. Records that are not completely full (number of fields is less than maximum number of fields) are filled with spaces to reach the maximum record size and keep page size, record size and number of records in a page fixed.

System Catalog
System Catalog Page 1
System Catalog Page ...
System Catalog Page 99

System Catalog Page	Page Header	
Page Header		
System Catalog Record 1	Page number	# of records
System Catalog Record ...		
System Catalog Record 10		

System Catalog Record	Record Header	
Record Header		
# of fields	Type name	
Field Name 1		
Field Name ...		
Field Name 10		

Type

The storage system consists of several types. Each type is stored in a different file. Each file stores all pages for a type. Pages store records and records store fields. Pages and records can be deleted physically. Each page shall have a fixed size of 1024 bytes but a file can have multiple pages. Pages that are not completely full are filled with spaces to reach 1024 bytes. There shall always be enough number of pages in a file for all records of a type (Pages are created after the last page when there is no empty record slot in the last page and last page is deleted when there is no record in the last page). New records are created at the right position to keep all records sorted according to key field. This property speeds up record searching process. When a record is deleted, all records after that, are shifted by one record to the left. Records that are not completely full (number of fields is less than maximum number of fields) are filled with zeroes to reach the maximum record size and keep page size, record size and number of records in a page fixed. There is only one key field which is the first field. Each field stores data as integers.

Type
Page 1
Page ...
Page 99

Page	Page Header	
Page Header		
Record 1	Page number	# of records
Record ...		
Record 24		

Record	Record Header	
Record Header		
Field 2	Field 1 (Key Field)	
Field ...		
Field 10		

Operations

DDL Operations

Create a Type

```
FUNCTION createType
  create a new record
  GET type name from user
  add type name to the record
  GET number of fields from user
  FOREACH field
    GET name from user
    add field to the record
  END FOREACH
  OPEN system catalog
  go to last page in system catalog
  add this record after the last record in last page
  update number of records in page header
  IF there is no empty record slots in the last page
    create a new page
    add page number to newly created page header
  END IF
  SAVE and CLOSE system catalog
  create a file with type name
  OPEN the file
  create a new page
  add page number to newly created page header
  SAVE and CLOSE the file
END createType
```

Delete a Type

```
FUNCTION deleteType
  GET type name from user
  OPEN system catalog
  FOREACH page in system catalog
    read number of records from page header
    FOREACH record in that page
      IF old record is deleted
        shift the current record by one record to the left
      ELSE IF key field of the current record matches the user input
        extract old record
      END IF
    END FOREACH
  END FOREACH
  IF there is no type names in the last page
    delete this page
    update number of pages in system catalog header
  END IF
  IF old record is NOT deleted
    delete the last record
  END IF
  IF there is no records in the last page
    delete last page
  END IF
  update number of records in last page header
  SAVE and CLOSE system catalog
  OPEN the file
  delete all pages and records of that file
  SAVE and CLOSE the file
END deleteType
```


List all Types

```
FUNCTION listAllTypes
  OPEN system catalog
  FOREACH page in system catalog
    read number of records from page header
    FOREACH record in that page
      PRINT type name and number of fields and field names
    END FOREACH
  END FOREACH
  CLOSE system catalog
END listAllTypes
```

DML Operations

```
FUNCTION chooseDML(operation type)
  GET type name from user
  OPEN system catalog
  FOREACH page in system catalog
    read number of records from page header
    FOREACH record in that page
      IF type name matches the user input
        CALL the function according to the operation type
        BREAK ALL
      END IF
    END FOREACH
  END FOREACH
  CLOSE system catalog
END chooseDML
```

Create a Record

```
FUNCTION createRecord(number of fields)
  OPEN the file
  create a new record
  FOREACH field
    GET value from user
    add field to the record
  END FOREACH
  FOREACH page in that file
    read number of records from page header
    FOREACH record in that page
      IF new record is inserted
        shift the current record by one record to the right
      ELSE IF key field of the new record is less than key field of the current record
        insert new record before current record
      END IF
    END FOREACH
  END FOREACH
  IF new record is inserted
    insert the last shifted record
  ELSE
    insert new record
  END IF
  update number of records in last page header
  IF there is no empty record slots in the last page
    create a new page
    add page number to newly created page header
  END IF
  SAVE and CLOSE the file
END createRecord
```

Delete a Record

```
FUNCTION deleteRecord
  OPEN the file
  GET key field from user
  FOREACH page in that file
    read number of records from page header
    FOREACH record in that page
      IF old record is deleted
        shift the current record by one record to the left
      ELSE IF key field of the current record matches the user input
        extract old record
      END IF
    END FOREACH
  END FOREACH
  IF old record is NOT deleted
    delete the last record
  END IF
  IF there is no records in the last page
    delete last page
  END IF
  update number of records in last page header
  SAVE and CLOSE the file
END deleteRecord
```

Update a Record

```
FUNCTION updateRecord(number of fields)
  OPEN the file
  create a new record
  FOREACH field
    GET value from user
    add field to the record
  END FOREACH
  FOREACH page in that file
    read number of records from page header
    FOREACH record in that page
      IF key field of the current record matches the user input
        insert new record
        BREAK ALL
      END IF
    END FOREACH
  END FOREACH
  SAVE and CLOSE the file
END updateRecord
```

Search for a Record

```
FUNCTION searchRecord
  GET search operator from user
  GET searched value from user
  OPEN the file
  FOREACH page in that file
    FOREACH record in that page
      IF search operator is <
        IF key field is less than searched value
          PRINT all fields of that record
        ELSE
          BREAK ALL
        END IF
      ELSE IF search operator is > AND key field is more than searched value
        PRINT all fields of that record
      ELSE IF search operator is = AND key field is equal to searched value
        PRINT all fields of that record
        BREAK ALL
      END IF
    END FOREACH
  END FOREACH
  CLOSE the file
END searchRecord
```

List all Records

```
FUNCTION listAllRecords
  OPEN the file
  FOREACH page in that file
    FOREACH record in that page
      PRINT all fields of that record
    END FOREACH
  END FOREACH
  CLOSE the file
END listAllRecords
```

Conclusion

Advantages

- Records are sorted by key fields.
 - Efficient for searching.
 - Decreases page accesses.
- There are 3 different search options.
 - Data can be analyzed in detail.
- Sizes are reasonable.
 - Easy to use for different purposes.
- Number of fields can be selected.
 - Gives freedom to user.
- Unused space in pages is minimal.
- Linked list implementation would be fast.
 - Shifting process required for creation and deletion of records is fast with a linked list.

Drawbacks

- Fields are of type integer.
 - Prevents the user from using other data types thus limits the user.
- No error checking for invalid user inputs can be problematic in real life usage.
- Records that are not completely full (number of fields is less than maximum number of fields) are filled with zeroes to reach the maximum record size and keep page size, record size and number of records in a page fixed.
- Fixed page size requires more space sometimes but enables a faster and efficient system.
 - Solution: Different page sizes can be determined for different usage types.