

CMPE321 Introduction to Database Systems
2016/2017-2

Berk Erol

Storage Manager Design

Berk Erol
2014400189

Table of Contents

Introduction	3
Assumptions and Constraints.....	4
System Catalog	4
System Catalog Page	4
Type	4
Page	5
Record.....	5
Data Structures	6
System Catalog	6
Type	6
Page	7
Record	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 designing 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

Assumptions and Constraints

System Catalog

- There is only one system catalog.
- Creating or deleting system catalogs are not allowed.
- System Catalog file shall have the following format: SystemCatalog.txt
- System Catalog header
 - Name of the storage system (50 bytes)
 - # of pages (2 bytes)
- Storage system name
 - Storage system name shall be alphanumeric.
 - The length of storage system name shall be at most 50 bytes.
- Size of System Catalog shall be at most 101428 bytes.
- System Catalog shall have at most 99 pages. $(101428 - 50 - 2) / 1024$
- System Catalog shall have at most 3465 types. $(99 * 35)$

System Catalog Page

- Each page stores type names.
- Page header
 - Page number (2 bytes)
 - # of type names (2 bytes)
- Line endings $(35 + 1) * 2 = 72$ bytes.
- Unused space 3 bytes.
- Size of each page shall be 1024 bytes.
- Each page shall have at most 35 type names. $(1024 - 2 - 2 - 72 - 3) / 27$
- Pages are deleted physically because type names are deleted physically.

Type

- Each type is stored in a different file.
- Each file shall be able to store all pages for a type.
- File shall have the following format: <TypeName>.txt
- Type name
 - Each type name shall be unique.
 - Type names shall be alphanumeric.
 - The length of type names shall be at most 27 bytes.
- File header
 - Usage status (1 byte)
 - # of pages (2 bytes)
 - # of fields (2 bytes)
 - Field names (112 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.
- Size of each type shall be at most 101493 bytes.

- Each type shall have at most 99 pages. $(101493-1-2-2-112)/1024$
- Each type shall have at most 1485 records. $(99*15)$
- Each type shall have at most 23760 fields. $(1485*16)$
- Types cannot be deleted physically (only their status change).

Page

- Each page stores records.
- Page header
 - Page number (2 bytes)
 - # of active records (2 bytes)
 - # of deleted records (2 bytes)
- Line endings $(15+1) * 2 = 32$ bytes.
- Unused space 11 bytes.
- Size of each page shall be 1024 bytes.
- Each page shall have at most 15 records. $(1024-2-2-2-32-11)/65$
- Each page shall have at most 240 fields. $(15*16)$
- Pages do not need to be deleted physically because records are not deleted physically.

Record

- Each record stores fields.
- Record header
 - Usage status (1 byte)
- Key field
 - Each key field shall be unique.
 - There shall be one and only one key field in a record.
 - Key field shall always be the first field in its record.
- Size of each record shall be at most 65 bytes.
- Each record shall have at most 16 fields. $(65-1)/4$
- Records cannot be deleted physically (only their status change).
- Size of each field shall be at most 4 bytes.
- Each field stores data as integers.
- Fields can be deleted physically (except key field).

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. Users first reach this file if they want to make an operation in the storage system. There is only one system catalog. Creating or deleting system catalogs are not allowed. Type names are stored in System Catalog pages. Each page shall have a fixed size of 1024 bytes the System Catalog can have multiple pages. There shall always be enough number of pages in System Catalog for all type names (Pages are created after the last page of the type when there is no empty type name slot in the last page and last page is deleted when there is no type name in the last page). Pages that are not completely full are filled with blank spaces to reach 1024 bytes.

System Catalog		System Catalog Header		System Catalog Page	Page Header	
System Catalog Header				Page Header		
System Catalog Page 1		System name	# of pages	Type Name 1	Page number	# of types
System Catalog Page 2				Type Name 2		
System Catalog Page ...				Type Name ...		
System Catalog Page 99				Type Name 35		

Type

The storage system consists of several types. Each type is stored in a different file. Each file stores all pages for a type. Usage status specifies whether the type is deleted or not (1 for active, 0 for deleted). Types cannot be deleted physically (only their status change). This property speeds up the type deletion process. Types can have different number of pages, records, fields and different sizes but the maximum amount for these values are the same for all types as specified in Assumptions and Constraints section.

File		File Header				
File Header		Usage status	# of pages	# of fields	Field Name 1	Field Name 16
Page 1						
Page 2						
Page ...						
Page 99						

Page

Each page stores records. Each page shall have a fixed size of 1024 bytes but a type (file) can have multiple pages. There shall always be enough number of pages in a file for all records of a type (Pages are created after the last page of the type when there is no empty record slot in the last page but pages do not need to be deleted physically because records are not deleted physically). This property speeds up the record creation and deletion processes. Pages that are not completely full are filled with blank spaces to reach 1024 bytes.

Page

Page Header
Record 1 (Status 1)
Record ... (Status 1)
Record 10 (Status 1)
Record 1 (Status 0)
Record ... (Status 0)
Record 5 (Status 0)

Page Header

Page number	# of active records	# of deleted records
-------------	---------------------	----------------------

Record

Each record stores fields. Usage status specifies whether the record is deleted or not (1 for active, 0 for deleted). New records are created at the right position in active records (to keep active records sorted according to key field). Records cannot be deleted physically (only their status change) and newly deleted records are moved to the beginning of deleted records (to shift less records to the left). This way active records remain sorted and deleted records come after active records. These properties speed up the record searching and listing processes. Records of different types can have different number of fields but the maximum amount for this value is the same for all records as specified in Assumptions and Constraints section. 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. Fields can be deleted physically (except key field).

Record

Record Header
Field 1 (Key Field)
Field 2
Field ...
Field 16

Record Header

Usage status

Operations

DDL Operations

Create a Type

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

Delete a Type

```
FUNCTION deleteType
  GET type name from user
  OPEN system catalog
  read number of pages from system catalog header
  FOREACH page in system catalog
    read number of types from page header
    FOREACH type name in that page
      IF type name is deleted
        shift the current type name by one type name to the left
      ELSE IF type name matches the user input
        delete this type name from system catalog
      END IF
    END FOREACH
    update number of types in page header
  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
  OPEN the file
  change usage status to deleted in file header
  delete all pages and records of that file
  SAVE and CLOSE the file
  SAVE and CLOSE system catalog
END deleteType
```


List all Types

```
FUNCTION listAllTypes
  GET type name from user
  OPEN system catalog
  read number of pages from system catalog header
  FOREACH page in system catalog
    read number of types from page header
    FOREACH type name in that page
      PRINT type name
    END FOREACH
  END FOREACH
  CLOSE system catalog
END listAllTypes
```

DML Operations

Create a Record

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

Delete a Record

```
FUNCTION deleteRecord
  OPEN the file
  GET key field from user
  read number of pages from file header
  FOREACH page in that file
    read numbers of active and deleted records from page header
    FOREACH active record in that page
      IF old record is deleted
        shift the current active record by one record to the left
      ELSE IF key field of the current active record matches the user input
        extract old record
        change usage status to deleted in record header
      END IF
    END FOREACH
  IF there is a deleted record in that page
    insert old record before current deleted record
    update numbers of active and deleted records in page header
  BREAK
END IF
update numbers of active and deleted records in page header
END FOREACH
SAVE and CLOSE the file
END deleteRecord
```

Update a Record

```
FUNCTION updateRecord
  OPEN the file
  read number of fields from file header
  GET key field from user
  read number of pages from file header
  FOREACH page in that file
    read number of active records from page header
    FOREACH active record in that page
      IF key field of the current active record matches the user input
        FOREACH field except key field
          GET value from user
          add field to the record
        END FOREACH
      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 active 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
  IF there is a deleted record in that page
    BREAK
  END IF
END FOREACH
CLOSE the file
END searchRecord
```

List all Records

```
FUNCTION listAllRecords
  OPEN the file
  FOREACH page in that file
    FOREACH active record in that page
      PRINT all fields of that record
    END FOREACH
  IF there is a deleted record in that page
    BREAK
  END IF
END FOREACH
CLOSE the file
END listAllRecords

FUNCTION chooseDML(operation type)
  GET type name from user
  OPEN system catalog
  read number of pages from system catalog header
  FOREACH page in system catalog
    read number of types from page header
    FOREACH type name 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
```

Conclusion

Advantages

- Records are sorted by key fields.
 - Efficient for searching.
 - Decreases page accesses.
- 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.
- No overwriting deleted records.
 - Solution: Pages that are full with deleted records can be deleted periodically depending on the usage (once a week or once a month).
- 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.