Project Name		Confidentiality Level
Database Management System		Public
Project ID	Version	Document Code
00001	1.0	00001_1.0_03

Database Management System
Software Requirements Specification

Revision Record

Date	Revision Version	CR ID /Defect ID	Sec No.	Change Description	Author
4.26	1.0			First	15301018

Content

Keywords: software requirement specification DBMS users

Abstract:

List of abbreviations: *Describe abbreviations in this document, full spelling of the abbreviation and Chinese explanation should be provided.*

Abbreviations	Full spelling	Chinese explanation
DBMS	Database management system	数据库管理系统

1 Introduction

1.1 Purpose

This manual describes requirements from users for functions of the DBMS. The reader includes users, project administrators, testers, designers, developers. The manual focuses on function requirements of DBMS, performance and interfaces of developed software.

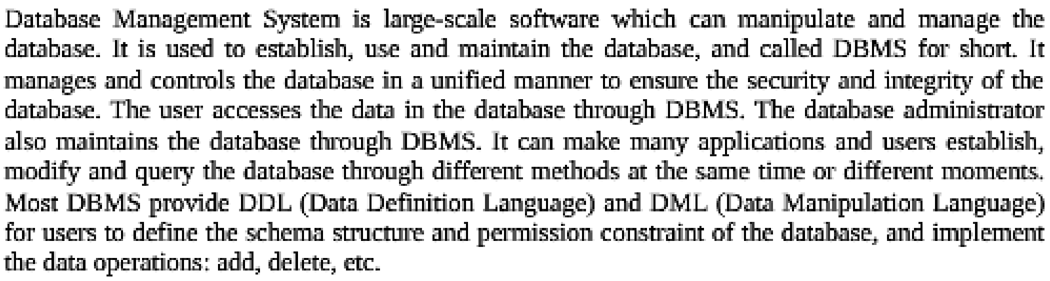
1.2 Scope

This manual analyzes the logical model of DBMS from users' perspective. It lists functions and services. The manual omits developments skills and describes user requirements.

2 General description

2.1 Software perspective

2.1.1 About the Project



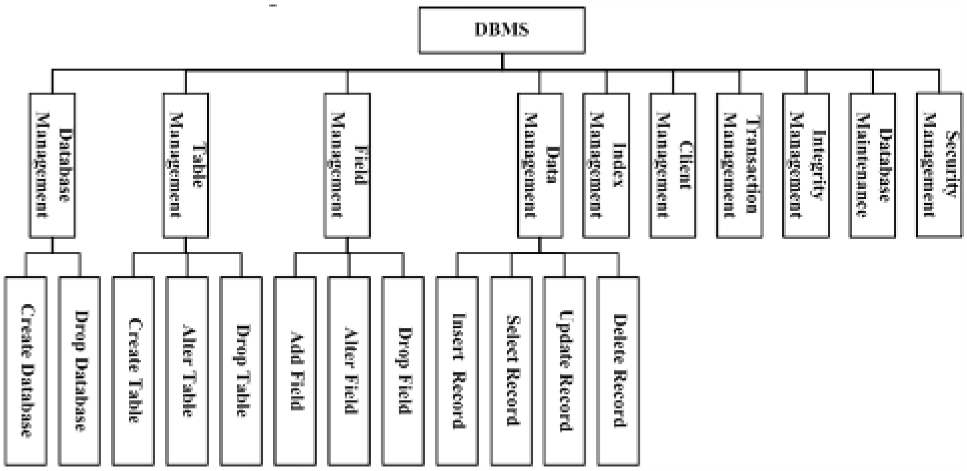
Database Management System is large-scale software which can manipulate and manage the database. It is used to establish, use and maintain the database, and called DBMS for short. It manages and controls the database in a unified manner to ensure the security and integrity of the database. The user accesses the data in the database through DBMS. The database administrator also maintains the database through DBMS. It can make many applications and users establish, modify and query the database through different methods at the same time or different moments. Most DBMS provide DDL (Data Definition Language) and DML (Data Manipulation Language) for users to define the schema structure and permission constraint of the database, and implement the data operations: add, delete, etc.

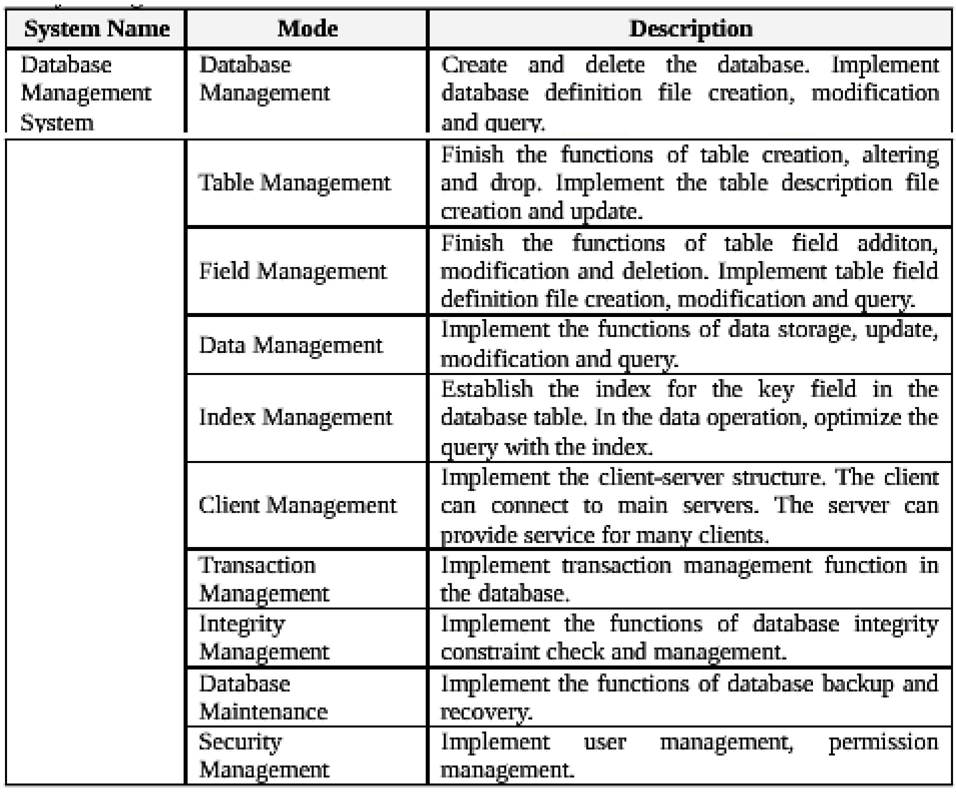
2.1.2 Environment of Product

This is a windows desktop window program. It stores the data with the file system of the operating system. It can implement the communication between the client and server through the network.

2.2 Software function

The function structure diagram is follows:





System Name	Mode	Description
Database Management System	Database Management	Create and delete the database. Implement database definition file creation, modification and query.
	Table Management	Finish the functions of table creation, altering and drop. Implement the table description file creation and update.
	Field Management	Finish the functions of table field additon, modification and deletion. Implement table field definition file creation, modification and query.
	Data Management	Implement the functions of data storage, update, modification and query.
	Index Management	Establish the index for the key field in the database table. In the data operation, optimize the query with the index.
	Client Management	Implement the client-server structure. The client can connect to main servers. The server can provide service for many clients.
	Transaction Management	Implement transaction management function in the database.
	Integrity Management	Implement the functions of database integrity constraint check and management.
	Database Maintenance	Implement the functions of database backup and recovery.
	Security Management	Implement user management, permission management.

2.3 Actors

The users of this system are the database administrators and the client users.

The database administrator requires some database management knowledge, and can install, deploy, create, manage and maintain the database.

The client use requires some application operation knowledge, and can use this database to implement the functions of data storage, modification, query, etc.

2.4 User characteristics

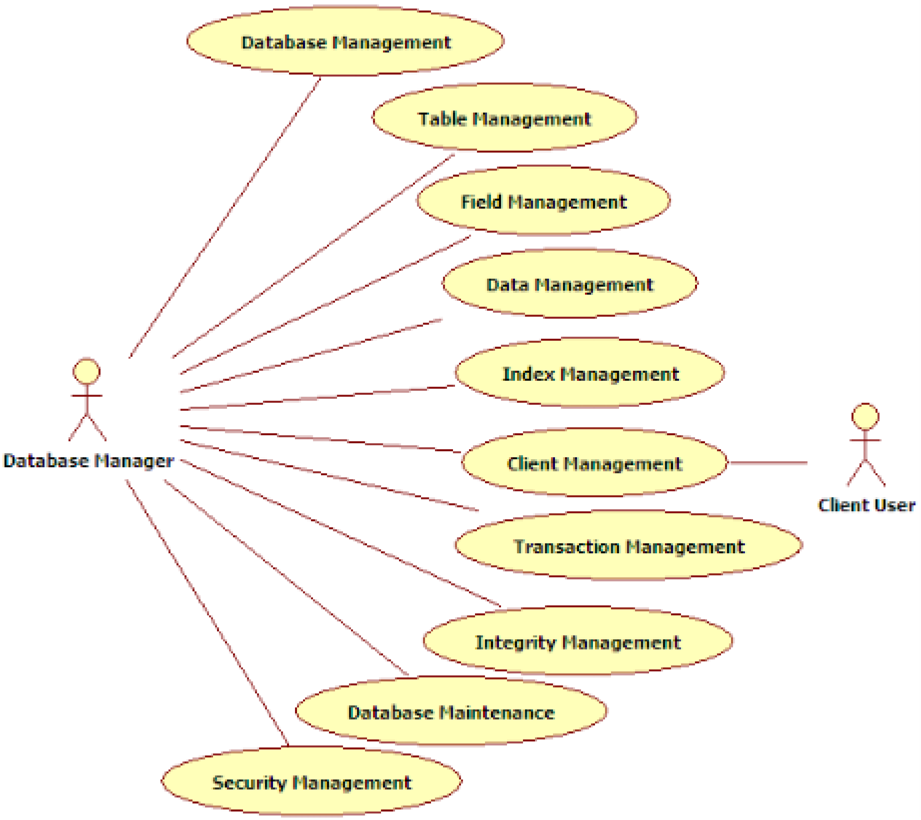
The users of this system are the database administrators and the client users.
The database administrator requires some database management knowledge, and can install, deploy, create, manage and maintain the database.
The client use requires some application operation knowledge, and can use this database to implement the functions of data storage, modification, query, etc.

2.5 Assumptions & Dependencies

This system generally uses the WINDOWS operating system, if users have additional operational requirements under the particular circumstances, the system should have a portable condition.

3 Specific Requirements

3.1 System Use Case

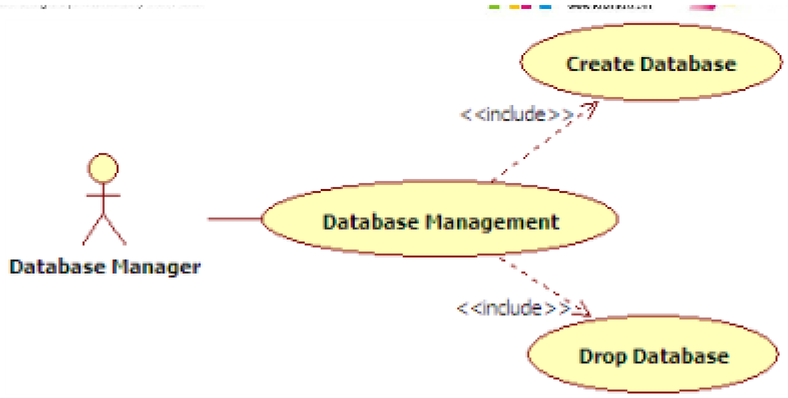


3.2 Database Management

1. Sub-module introduction

Create and delete the database. Implement database definition file creation, modification and query..

2. Sub-module use case diagram



3.2.1 Functional Requirements1

1. Introduction

Implement the function of database creation. Corresponding SQL statement: CREATE DATABASE <database name>. The database is divided into two types: user database and system database. The system database is database Ruanko created by the system when the system is created. This database can't be deleted.

2. Inputs

Database name, database date file storage path, database creation time, database type.

3. Process

(1) Determine the validity of the database name.
The length of the database name can't exceed 128 characters. If the name is too long, the database can't be created and prompt the user.
Determine whether the database name has existed. If it exists, the database can't be created and prompt the user.
(2) Save the database information into database file ruanko.db.
Determine whether ruanko.db file exists under the system directory. If it exists, append to the file. If it doesn't exist, create raunko.db file.

4. Output

Create or update database description file ruanko.db file, add or delete the database information.

3.2.2 Drop Database

1. Introduction

Implement the function of database deletion. Corresponding SQL statement: DROP DATABASE <database name>. The database is divided into two types: the user database and the system database. The user database can be deleted. The system database is database Ruanko created by the system when the system is created. This database can't be deleted.

2. Inputs

The database name, database data file storage path, database creation time, database type.

3. Process

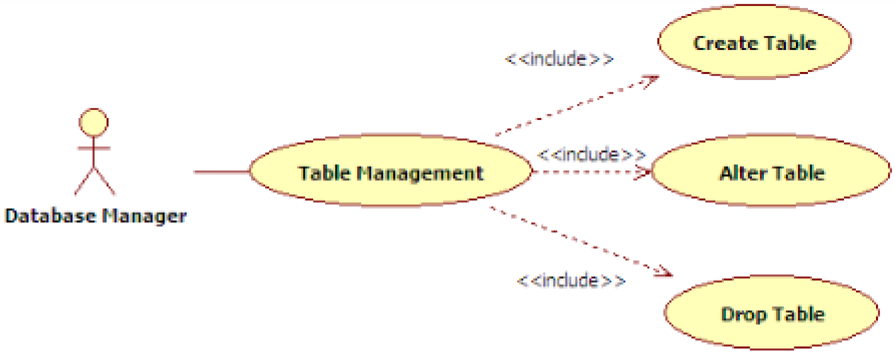
(1) Determine whether the database exists.
(2) Determine whether the database is used by the user. If there is a user that connects to it, it can not be deleted.
(3) Delete the database information from ruanko.db file.
(4) Delete all files related to the database.

4. Output

Delete the database description file in database description file ruanko.db.

3.3 Table Management

1. Sub-function Introduction
Finish the functions of table creation, modification and deletion. Implement the table description file creation and update.
2. Sub-function System Use Case



3.3.1 Create Table

1. Introduction

Implement database table creation function. Corresponding SQL statement: CREATE TABLE <table name>. When the table is created, add the basic information of the table into the table information file of the current database. The table description file is: [Database Name].tb.

2. Inputs

Database name, table name, table definition file path, table integrity file path, table record file path, table index file path, table creation time, last modification time (is creation time), table field count (initial is 0), total number of records in table (initial is 0).

3. Process

(1) Determine the validity of the table name.
The length of the table name can't exceed 128 characters. If the name is too long, the table can't be created and prompt the user.
Determine whether the table name has existed. If it exists, the table can't be created and prompt the user.
(2) Get the paths of four table definition files according to the rule.
Table definition file path: [Table Name].tdf.
Table integrity file path: [Table Name].tic.
Table record file path: [Table Name].trd.
Table index file path: [Table Name].tid.
(3) Save the table information to table description file "[Database Name].tb" file. If the table information file does not exist, create the file. If it exists, append the record to the end of the file.

4. Output

Create or update the table description file: "[Database Name].tb".

3.3.2 Alter Table

1. Introduction

Implement database table modification function. Corresponding SQL statement: ALTER TABLE <table name> <alter table action>. When the table field is added, update the field count, record count, modification time or path in the table.

2. Inputs

Database name, table name, table definition file path, table integrity file path, table record file path, table index file path, table creation time, last modification time (is current time), table field count, total number of records in table.

3. Process

(1) Through the database name and table name, find the table information file, get the table modification information.
(2) Set the modification time.
(3) Update the table information.

4. Output

Update table description file: "[Database Name].tb".

3.3.3 Drop Table

1. Introduction
Implement database table deletion function. Corresponding SQL statement: DROP TABLE <table name>.

2. Input
The database name and table name.

3. Processing
(1) Through the database name and table name, search the table information file to get the table information.
(2) Delete the current table information.
(3) Delete the definition file and record file corresponding to the table.

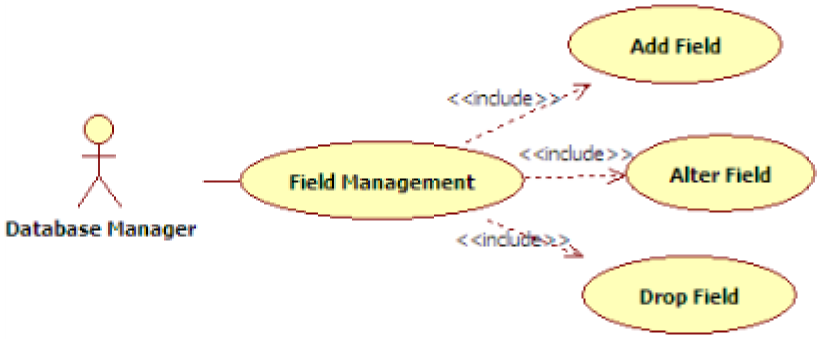
4. Output
Update the table description file: "[Database name].tb".

3.4 Field Managent

1. Sub-function Introduction

Finish the functions of table field additon, modification and deletion. Implement table field definition file creation, modification and query.

2. Sub-function System Use Case



3.4.1 Add Field

1. Introduction

In created table, add fields. Corresponding SQL statement: ALTER TABLE <table name> ADD COLUMN <column name> <column definition>.

2. Inputs

Database name, table name, field name, field order, field type, field type parameter, modification time (creation time), integrity constraint information (default is none).

3. Process

(1) Determine whether the table description file exists. If it doesn’t exist, create it.
(2) Determine whether the field name is valid. If it is invalid, it can’t be created.
(3) Update the field information.

4. Output

Create or update the table definition file. File name:[Table Name].tdf.

3.4.2 Alter Field

1. Introduction

Modify the field information in the table. Corresponding SQL statement: ALTER TABLE <table name> MODIFY COLUMN <column name> <alter column action>.

2. Inputs

Database name, table name, field name.
Modified field information: field name, field order, field type, field type parameter, modification time (creation time), integrity constraint information (default is none).

3. Process

(1) Determine whether the field exists. If it doesn’t exist, it fails and prompt the user.
(2) Read the original field information.
(3) Update the field information.
(4) Update the corresponding field information in the index.
(5) Determine whether there is a record. If there is a record, update all the records.

4. Output

Updated record file.
Updated table description file.
Updated index description file.

3.4.3 Drop Field

1. Introduction

Delete the field in the table. Corresponding SQL statement: ALTER TABLE <table name> DROP COLUMN <column name> <drop behavior>.

2. Inputs

The database name, table name and field name.

3. Process

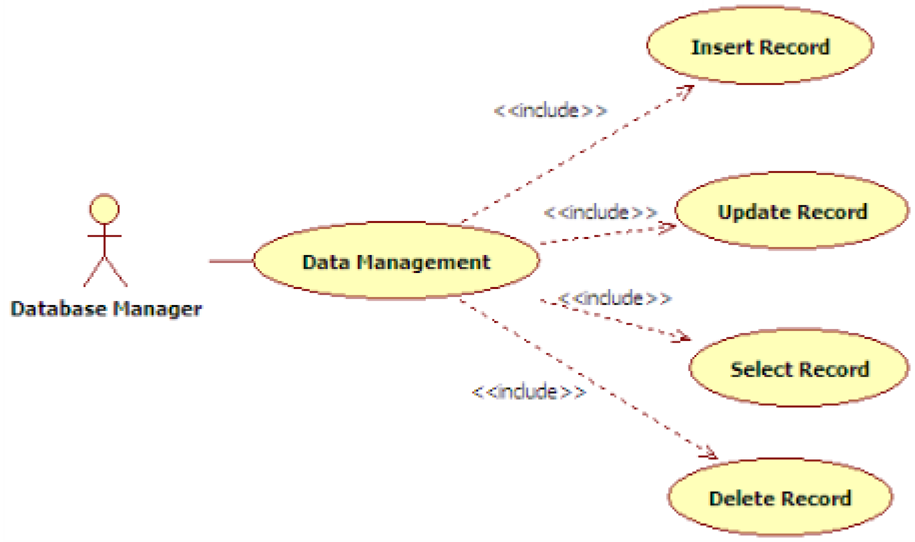
Determine whether the field exists. If it does not exist, prompt the user.
Determine whether the current field is used by the index. If it is used by an index, remove the corresponding index.
Update the table field information, and delete the field information.
Determine whether there are records. If there are records, update all records.

4. Output

Updated record files.
Updated table definition files.
Updated index description files.

3.5 Data management

1. Sub-function Introduction
Implement the functions of data storage, update, modification and query.
2. Sub-function System Use Case



3.5.1 Insert Record

1. Introduction

Insert a record into the database table. Corresponding SQL statement: INSERT INTO <table name> <column name list> VALUES <insert value list>.

2. Inputs

Database name, table name, key-value pair list of field name and field value.

3. Process

(1) Read the field record in the table.
(2) Check the integrity constraint.
(3) Assign values to the autoincrement field and default field.
(4) Save records into the record file.

4. Output

Record file, record file name: [Table Name].trd.

3.5.2 Update Record

1. Introduction

Update the record in the database table. Corresponding SQL statement is: UPDATE <table name> SET <column name> = <update value> [WHERE <search condition>].

2. Inputs

Database name, table name, key-value pair list of field name and field value.

3. Process

(1) Read the field record in the table.
(2) Check the integrity constraint.
(3) Find the record to be updated (find through the primary key).
(4) Update the record to the record file.

4. Output

Record file.

3.5.3 Select Record

1. Introduction

Query all the records in the table. Corresponding SQL statement is: SELECT * FROM <table name> [WHERE <search condition>].

2. Inputs

Database name,table name

3. Process

(1) Read the database table information.
(2) Read the record information in the table.
(3) Return all the record information.
(4) Display the records.

4. Output

Display the records in the view of the interface.

3.5.4 Delete Record

1. Introduction

Delete the record in the table. Corresponding SQL statement is: DELETE FROM <table name> [WHERE <search condition>].

2. Inputs

The database name, table name, key-value pair list of the field name and field value.

3. Process

(1) Read the database table information.
(2) Locate to the record of the table to be deleted.
(3) Update the record files.
(4) Updated the index files

4. Output

Updated index files.
Updated record files.

3.6 Index Management

Establish the index for the key field in the database table. In the data operation, optimize the query with the index.

3.7 Client Management

Implement the client-server structure. The client can connect to main servers. The server can provide service for many clients.

3.8 Transaction Management

Implement transaction management function in the database.

3.9 Integrity Management

Implement the functions of database integrity constraint check and management.

3.10 Database Maintenance

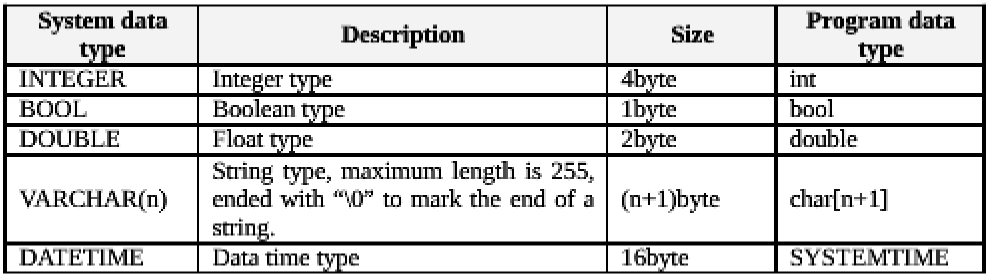
Implement the functions of database backup and recovery.

3.11 Security Management

Implement user management, permission management.

3.12 Data Information

3.12.1 Data type



System data type	Description	Size	Program data type
INTEGER	Integer type	4byte	int
BOOL	Boolean type	1byte	bool
DOUBLE	Float type	2byte	double
VARCHAR(n)	String type, maximum length is 255, ended with "\0" to mark the end of a string.	(n+1)byte	char[n+1]
DATETIME	Data time type	16byte	SYSTEMTIME

3.12.2 Integrity

3.12.2.1 Entity Integrity

PRIMARY KEY

3.12.2.2 Referential Integrity

FOREIGN KEY

3.12.2.3 User-defined Integrity

1. CHECK
2. UNIQUE
3. NOT NULL
4. DEFAULT
5. IDENTITY

3.12.3 Database file

This system is a relational database management system. It stores data with the binary file.

1. File Design

The files in DBMS are mainly divided into two types: data definition file and data file

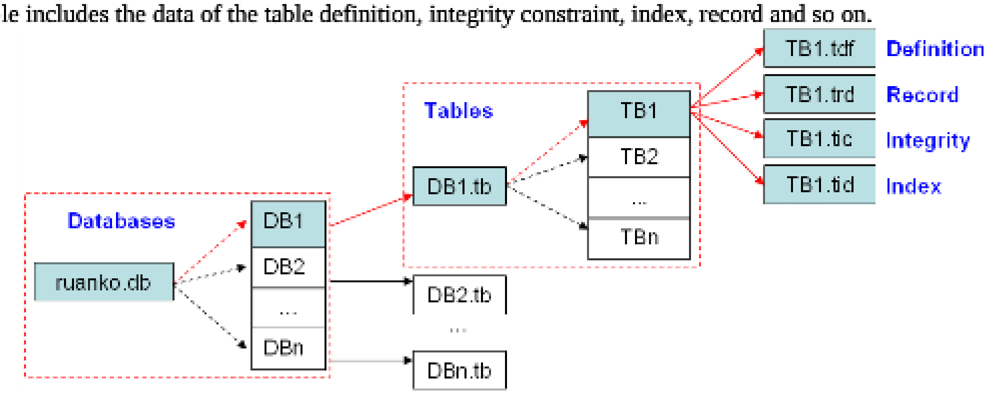
(1) Data definition file: saves the definitions of various objects in DBMS.

(2) Data file: saves various data in DBMS.

Type	File	Name	Remark
Data definition file	Database description file	ruanko.db	Save the database information.
	Table description file	*.tb	Save the table information.
	Table definition file	*.tdf	Save the field information.
	Integrity description file	*.tic	Save the integrity constraints.
	Index description file	*.tid	Save the definition of the index.
Data File	Record File	*.trd	
	Index data file	*.ix	
	Log file	*.log	

2. File Structure

The database management system supports multi-database. A database can include many tables. A table includes the data of the table definition, integrity constraint, index, record and so on.



3. Directory Structure

Taking [DBMS_ROOT] as the root directory, each database creates a folder to save various files in the database. Path :[DBMS_ROOT]\data\DB_NAME\.

Example: create a “Ruanko” database, and create "Student“ table in the database. The presented directory structure is as follows:

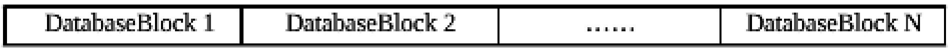
File	Path
Database description file	[DBMS_ROOT]\ruanko.db
Table description file	[DBMS_ROOT]\data\Ruanko\Ruanko.tb
Table definition file	[DBMS_ROOT]\data\Ruanko\Student.tdf
Record file	[DBMS_ROOT]\data\Ruanko\Student.trd
Integrity description file	[DBMS_ROOT]\data\Ruanko\Student.tic
Index description file	[DBMS_ROOT]\data\Ruanko\Student.tid

3.12.4 Database Description File

3.12.4.1 File Name

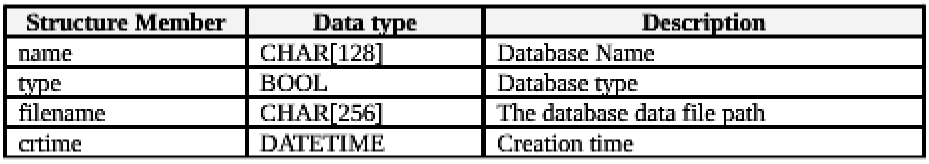
ruanku.db

3.12.4.2 File Structure



DatabaseBlock 1	DatabaseBlock 2	……	DatabaseBlock N

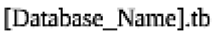
3.12.4.3 Database information structure



Structure Member	Data type	Description
name	CHAR[128]	Database Name
type	BOOL	Database type
filename	CHAR[256]	The database data file path
crtime	DATETIME	Creation time

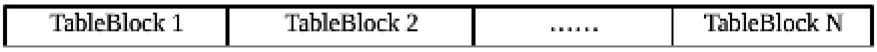
3.12.5 Table Description File

3.12.5.1 File Name



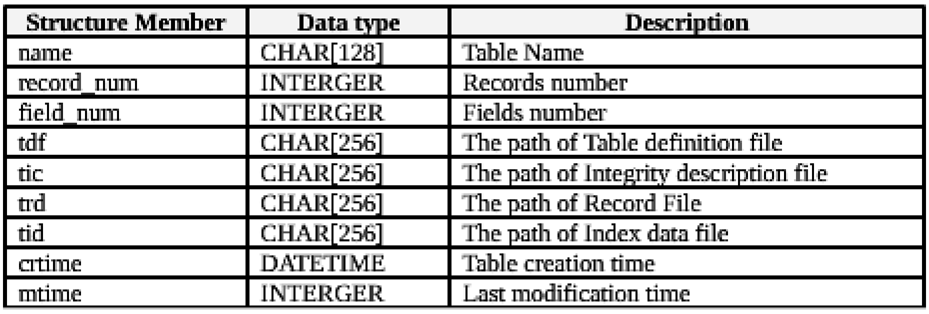
[Database_Name].tb

3.12.5.2 File Structure



TableBlock 1	TableBlock 2	……	TableBlock N

3.12.5.3 Table information structure



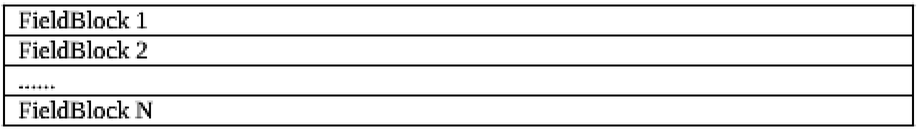
Structure Member	Data type	Description
name	CHAR[128]	Table Name
record_num	INTERGER	Records number
field_num	INTERGER	Fields number
tdf	CHAR[256]	The path of Table definition file
tic	CHAR[256]	The path of Integrity description file
trd	CHAR[256]	The path of Record File
tid	CHAR[256]	The path of Index data file
crtime	DATETIME	Table creation time
mtime	INTERGER	Last modification time

3.12.6 Table Definition File

3.12.6.1 File Name

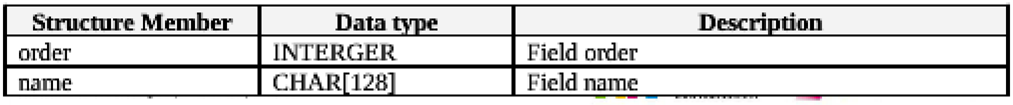
[Table_Name].tdf

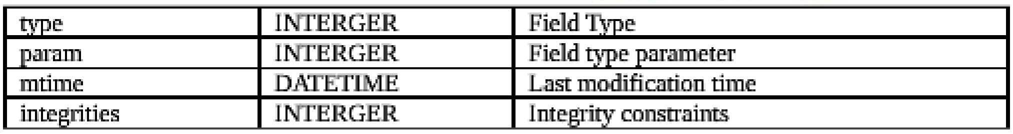
3.12.6.2 File Structure



FieldBlock 1
FieldBlock 2
……
FieldBlock N

3.12.6.3 Field information structure





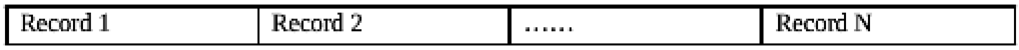
Structure Member	Data type	Description
order	INTERGER	Field order
name	CHAR[128]	Field name
type	INTERGER	Field Type
param	INTERGER	Field type parameter
mtime	DATETIME	Last modification time
integrities	INTERGER	Integrity constraints

3.12.7 Record File

3.12.7.1 File Name

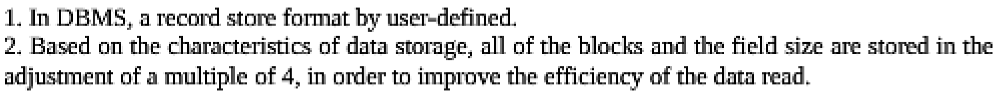
[Table_Name].trd

3.12.7.2 File Structure



Record 1	Record 2	……	Record N

3.12.7.3 Record information structure



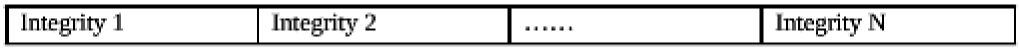
1. In DBMS, a record store format by user-defined.
2. Based on the characteristics of data storage, all of the blocks and the field size are stored in the adjustment of a multiple of 4, in order to improve the efficiency of the data read.

3.12.8 Integrity Description File

3.12.8.1 File Name

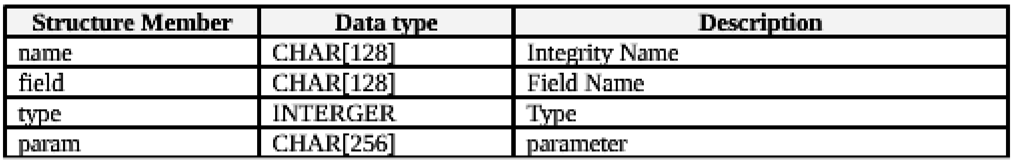
[Table_name].tic

3.12.8.2 File Structure



Integrity 1	Integrity 2	……	Integrity N

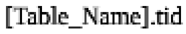
3.12.8.3 File information structure



Structure Member	Data type	Description
name	CHAR[128]	Integrity Name
field	CHAR[128]	Field Name
type	INTERGER	Type
param	CHAR[256]	parameter

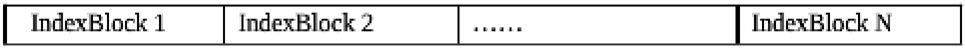
3.12.9 Index Description File

3.12.9.1 File Name



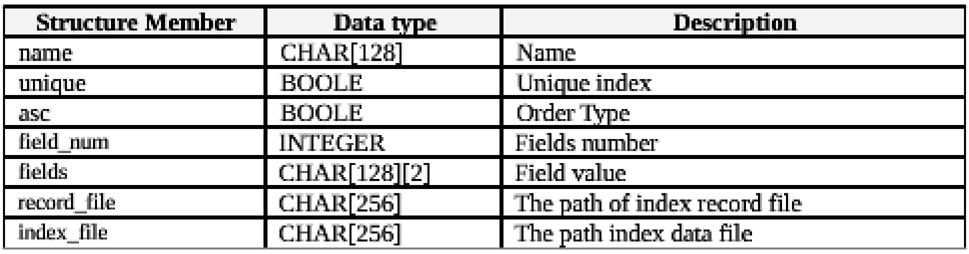
[Table_Name].tid

3.12.9.2 File Structure



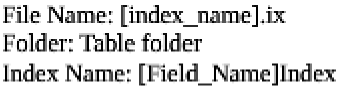
IndexBlock 1	IndexBlock 2	……	IndexBlock N

3.12.9.3 Index information structure



Structure Member	Data type	Description
name	CHAR[128]	Name
unique	BOOLE	Unique index
asc	BOOLE	Order Type
field_num	INTEGER	Fields number
fields	CHAR[128][2]	Field value
record_file	CHAR[256]	The path of index record file
index_file	CHAR[256]	The path index data file

3.12.9.4 Index data file



File Name: [index_name].ix
Folder: Table folder
Index Name: [Field_Name]Index

4 Interface Requirements

4.1 User Interface

The resolution is 1024*768 and software interfaces should adapt to the screen dimension.

4.2 Software Interface

This is a windows application program with GUI.

5 Overall Design Constraints

5.1 Hardware Limitations

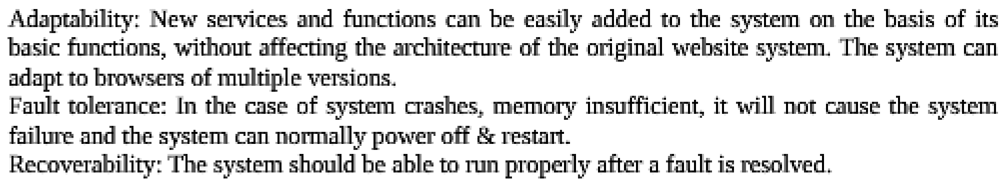
Minimum configuration on a computer:
CPU: 1 GHZ
Memory: 128 MB

5.2 Technology Limitations

File format: binary mode file format
Coding standard: C++ Coding standard

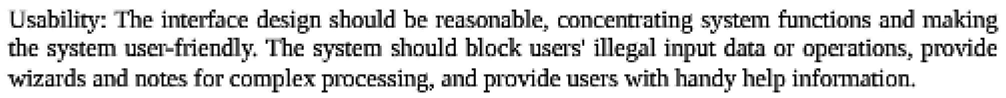
6 Software Quality Attributes

6.1 Reliability



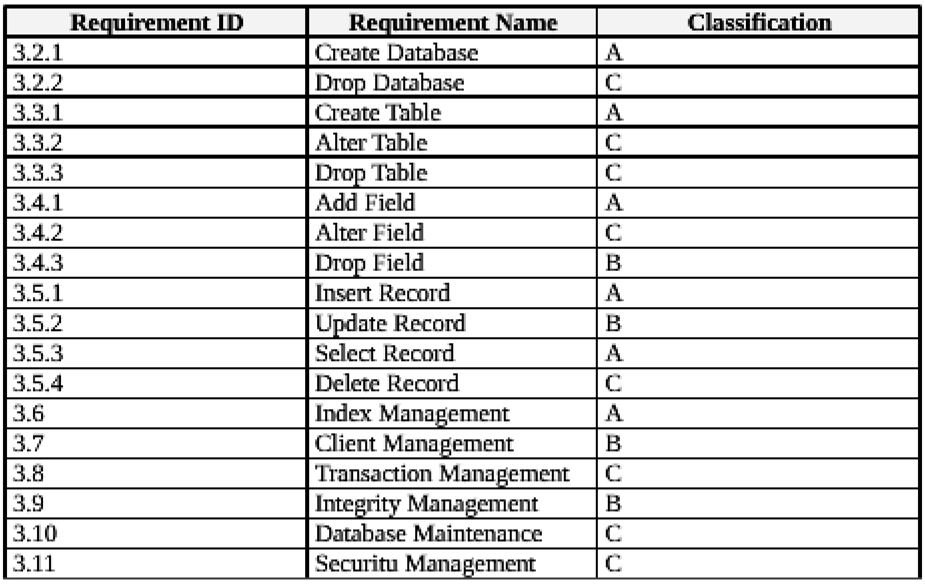
Adaptability: New services and functions can be easily added to the system on the basis of its basic functions, without affecting the architecture of the original website system. The system can adapt to browsers of multiple versions.
Fault tolerance: In the case of system crashes, memory insufficient, it will not cause the system failure and the system can normally power off & restart.
Recoverability: The system should be able to run properly after a fault is resolved.

6.2 Usability



Usability: The interface design should be reasonable, concentrating system functions and making the system user-friendly. The system should block users' illegal input data or operations, provide wizards and notes for complex processing, and provide users with handy help information.

7 Requirements Classification



Requirement ID	Requirement Name	Classification
3.2.1	Create Database	A
3.2.2	Drop Database	C
3.3.1	Create Table	A
3.3.2	Alter Table	C
3.3.3	Drop Table	C
3.4.1	Add Field	A
3.4.2	Alter Field	C
3.4.3	Drop Field	B
3.5.1	Insert Record	A
3.5.2	Update Record	B
3.5.3	Select Record	A
3.5.4	Delete Record	C
3.6	Index Management	A
3.7	Client Management	B
3.8	Transaction Management	C
3.9	Integrity Management	B
3.10	Database Maintenance	C
3.11	Securitu Management	C

Importance of requirements is classified as following:

A. Mandatory: absolutely essential features, without which the product development will be canceled.

B. Important: unessential features that may affect the viability of the product.

C. Nice to have: desired features, the absence of which will not affect the product viability.