DBMS设计文档

目录

[1简介 4](#_Toc481508432)

[1.1目的 4](#_Toc481508433)

[1.2范围 4](#_Toc481508434)

[2 第 0 层设计描述 4](#_Toc481508435)

[2.1软件系统上下文定义 4](#_Toc481508436)

[2.2设计思路 5](#_Toc481508437)

[3第 1 层设计描述 5](#_Toc481508438)

[3.1系统结构 5](#_Toc481508439)

[3.2分解描述 6](#_Toc481508440)

[3.3依赖性描述 8](#_Toc481508441)

[4第二层设计描述 8](#_Toc481508442)

[4.1 创建数据库 8](#_Toc481508443)

[4.2 创建表 9](#_Toc481508444)

[4.3 添加字段 9](#_Toc481508445)

[4.4 数据管理 10](#_Toc481508446)

关键词:MFC SDI Dialog C/S MVC 完整性 数据库 表 字段 记录数据

摘要 : 这是关于数据库管理系统的第一个版本，此数据库管理系统实现的功能包括数据库的建立和删除，表的管理，记录的增加、查看、删除，实现完整性约束，实现索引的建立和删除，实现多用户，进程同步，事务管理等

缩写列表:

| 缩写 | 全拼 | 中文解释 |
| --- | --- | --- |
| SDI | Single document interface | 单文档界面 |
| MFC | Microsoft foundation class | 微软基础类库 |
| DBMS | Database management system | 数据库管理系统 |
| RDBMS | Relational Database management system | 关系型数据库管理系统 |
| C/S | Client/server | 客户/服务器模式 |
| MVC | Model view controller | 模型、视图、控制器 |

# 1简介

### 1.1目的

本文档对数据库管理系统概要设计和详细设计进行说明，用于指导项目组下阶段的编码 实现和单元测试工作。本文档供项目组成员、客户项目代表、测试组成员、QA 等阅读。

### 1.2范围

#### 1.2.1软件名称

数据库管理系统。

#### 1.2.2软件功能

参考《DBMS需求分析文档》。

#### 1.2.3软件应用

本系统为关系数据库管理系统，数据存储使用文件，为小型的数据库系统。适合数据量 不大的小型数据存储与查询。

# 2 第 0 层设计描述

### 2.1软件系统上下文定义

数据库管理系统(DBMS)的体系结构包括：DBMS 体系结构、用户接口、语法分析、 查询处理、目录管理、并发控制、恢复机制、物理存储管理等。本系统主要实现模拟 DBMS 中 DDL、DML、DQL 等功能

### 2.2设计思路

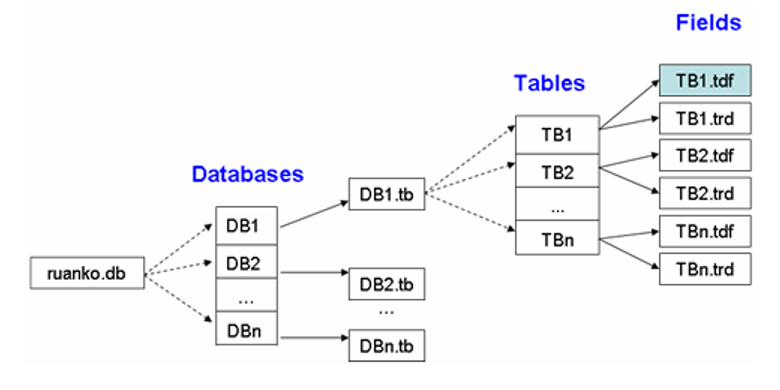
#### 2.2.1设计方案

1、程序结构设计 本系统为 MFC SDI 单文档工程。为提高程序的可读性，可维护性，对程序进行分层， 程序分为表示层，逻辑层和数据存储层。

程序软件结构可以按逻辑职责划分，分为"表示层"、 "业务逻辑层"、 "数据访问层"三层 。 各层之间使用"实体类"(数据对象) 进行数据的传递。另外，程序中各层中都可能用到且与 业务无关的公共类作为"工具类"。

2、数据存储结构 系统数据使用操作系统的二进制文件存储，以文件夹与文件来保存定义信息与数据信

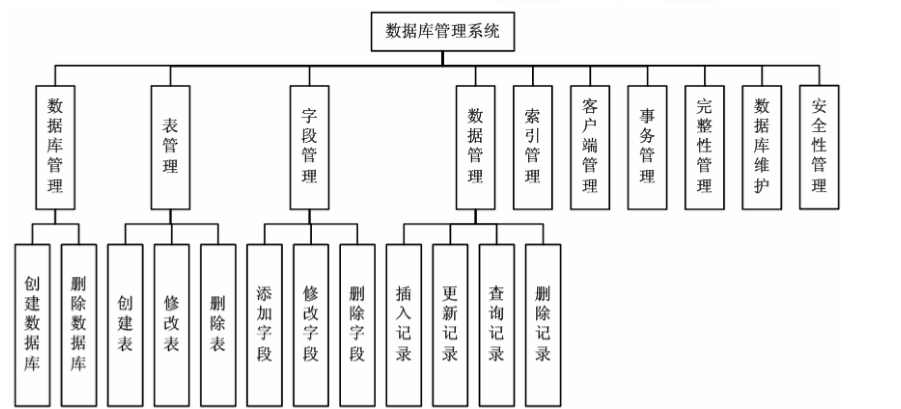
息。DBMS 系统定义文件包括：数据库描述文件(ruanko.db)、表描述文件(\*.tb)、表定义文件(保存字段信息，\*.tdf)、索引描述文件(\*.tid)、完整性描述文件(\*.tic)。DBMS 的数据文件 包括：记录文件(\*.trd)、索引数据文件(\*.ix)、日志文件(\*.log)、事务数据文件(\*.tac)、临时 文件(\*.tmp)等。 数据库描述文件、表描述文件、表定义文件、记录文件之间的关系图如下：



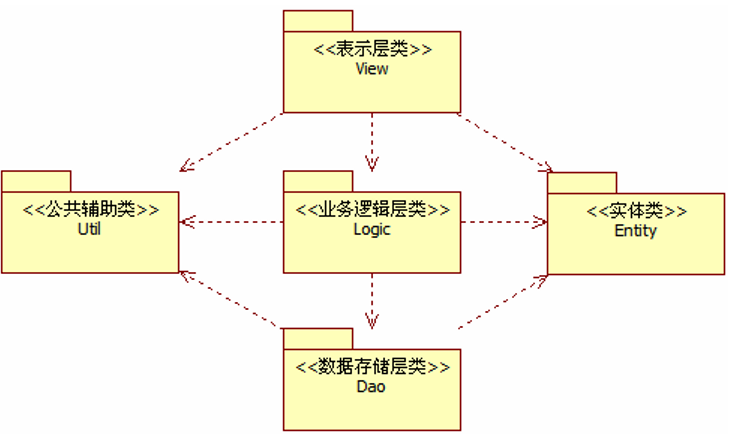
# 3第 1 层设计描述

### 3.1系统结构

系统按照分而治之的思想进行开发，将功能划分成若干个模块进行分别管理和开发， 系统详细的模块划分如下图所示：



各模块之间使用按三层结构进行划分，如下图所示：



工程在逻辑上使用文件夹进行划分，共分为：表示层、业务逻辑层、数据存储层、实

体类、公共辅助类。分别用文件夹过滤器：View、Logic、Dao、Entity、Util 来进行分隔。

### 3.2分解描述

#### 3.2.1 数据库管理

1、简介

完成数据库的创建与删除功能，实现数据库定义文件的创建、修改与查询。

2、功能列表

|  |  |  |
| --- | --- | --- |
| 模块名称 | 模块功能 | 功能描述 |
| 数据库管理 | 创建数据库 | 实现数据库的创建功能。对应 SQL 语句： CREATE DATABASE <database name>。 |
| 删除数据库 | 实现数据库的删除功能。对应SQL语句：DROP DATABASE <database name>。 |

#### 3.2.2 表管理

1、Overview 简介 完成表的添加、修改与删除功能，实现表定义文件的创建、修改与查询。

2、Functions 功能列表

|  |  |  |
| --- | --- | --- |
| 模块名称 | 模块功能 | 功能描述 |
| 表管理 | 创建表 | 实现数据库表的创建功能，对应的 SQL 语句： CREATE TABLE <table name>。 |
| 修改表 | 实现数据库表的修改功能，对应的 SQL 语句： ALTER TABLE <table name> <alter tableaction> |
| 删除表 | 实现数据库表的删除功能，对应的 SQL 语句： DROP TABLE <table name>。 |

#### 3.2.3 字段管理

1、Overview 简介 完成表字段的添加、修改与删除功能，实现表字段定义文件的创建、修改与查询。

2、Functions 功能列表

|  |  |  |
| --- | --- | --- |
| 模块名称 | 模块功能 | 功能描述 |
| 字段管理 | 添加字段 | 在已建的表中，添加字段。对应 SQL 语句： ALTER TABLE <table name> ADD COLUMN <column name> <column definition>。 |
| 修改字段 | 修改表中的字段信息。对应 SQL 语句：ALTER TABLE <table name> MODIFY COLUMN <column name> <alter column action>。 |
| 删除字段 | 删除表中的字段。对应 SQL 语句：ALTER TABLE <table name> DROP COLUMN <column name> <drop behavior>。 |

#### 3.2.4 数据管理

1、Overview 简介 实现数据的存储、更新、修改与查询的功能。

2、Functions 功能列表

|  |  |  |
| --- | --- | --- |
| 模块名称 | 模块功能 | 功能描述 |
| 数据管理 | 插入记录 | 向数据库表中插入一条记录。对应 SQL 语句： INSERT INTO <table name> <column name list> VALUES <insert value list>。 |
| 更新记录 | 更新数据库表中的记录。对应的 SQL 语句为： UPDATE <table name> SET <column name> = <update value> [ WHERE <search condition> ]。 |
| 查询记录 | 查询表中的全部记录。对应 SQL 语句为： SELECT \* FROM <table name> [ WHERE <search condition> ]。 |
| 删除记录 | 删除表中的记录。对应 SQL 语句为：DELETE FROM <table name> [ WHERE <search condition> ]。 |

### 3.3依赖性描述

本项目是Windows窗口程序，依赖于操作系统。数据存储依赖于操作系统的文件管理系

统。

# 4第二层设计描述

### 4.1 创建数据库

系统默认创建一个名为“dbsystem”的数据库。数据库基本信息保存在 CDBEntity 中， 使用树视图 CDBView 显示创建的数据库名。通过文档类 CRKDBMSDoc，将数据库信息传 递给视图类。数据库信息通过 CRKDBMSDoc 类传递给 CDBLogic 类，CDBLogic 类将数 据库信息传递给 CDBDao 类，CDBDao 类将数据库信息保存在“Ruanko.db”文件中。

1. CDBView

显示数据库结构视图类，在树视图中显示数据库名。

2、CRKDBMSDoc

文档类，设置默认数据库名，存储与读取数据，控制界面的显示与刷新。

3、CDBLogic

数据库业务逻辑类，判断默认的数据库是否存在，如果不存在则创建默认数据库。

4、CDBDao

数据库信息操作类，将数据库描述信息以二进制格式保存到“Ruanko.db”文件中。

5、CFileLogic

根据数据库名，获取数据库描述文件绝对路径。

6、CFileHleper

创建数据库描述文件路径中的文件夹和数据库描述文件。

### 4.2 创建表

根据数据库名 dbsystem 创建一个数据库表描述文件dbsystem.tb，用于保存数据库中的表 格信息。通过读取表描述文件(dbsystem.tb)，即可找到表结构定义文件(\*.tdf)和数据记录文 件(\*.trd)。

数据库信息通过 CDBEntity 保存，表信息通过 CTableEntity 保存。通过 CMainFrame 类响应创建表菜单事件，显示创建表对话框。通过创建表 CNewTableDlg 获取输入的表名。 将输入的表名传递给 CRKDBMSDoc。CRKDBMSDoc 将数据库名和表信息传递给 CTableLogic。CTableLogic 将表文件路径和表信息传递给 CTableDao。CTableDao 将表信息 以二进制格式保存到文件中。保存成功之后，在 CDBView 中显示表名，在 CTableView 中 显示表结构。

1. CMainFrame

框架窗口类，显示视图，响应创建表菜单事件。

1. CNewTableDlg

新建表对话框类，通过控件映射获取输入的表名。

1. CDBView

显示数据库结构视图类，在树视图的数据库结点下显示表名。

1. CTableView

显示表结构视图类。使用列表控件显示表结构。

1. CRKDBMSDoc

文档类，存储数据库和表信息，并控制界面显示和更新。

1. CTableLogic

表业务逻辑类。

1. CTableDao

表文件操作类，将表信息以二进制格式保存到文件中。

### 4.3 添加字段

根据表名创建表定义文件(\*.tdf)，保存用户定义的表结构。将文件路径保存到表描述 文件中，通过读取表描述文件即可找到表定义文件(\*.tdf)。将字段信息保存到表定义文件 中。

CMainFrame 类响应添加字段事件，在 CDBView 中用户选择表，在 CFieldDlg 中接收 输入的字段信息，字段信息保存在 CFieldEntity 中。CRKDBMSDoc 将数据库名、表和字段 传递给 CTableLogic。CTableLogic 将字段信息保存到表定义文件中。CDBView 将字段名显 示在表结点的列子结点下，CTableView 显示字段信息。

1. CFieldDlg

添加字段对话框类，接收输入的字段信息。

1. CMainFrame

框架窗口类，响应添加字段事件。

1. CRKDBMSDoc

文档类，保存数据库信息、表信息和字段信息，并控制界面显示和更新。

1. CDBView

数据库视图类，显示数据库中的表名以及表中的字段名。

1. CTableView

表结构视图类，显示选择的表中的字段信息。

1. CTableLogic

表业务逻辑类，调用 CTableDao 类方法，将字段信息保存到表定义文件(\*.tdf)中，并 修改表描述文件(\*.tb)。

1. CTableDao

表数据库处理类，将字段信息保存到表定义文件中。

1. CFieldEntity

字段信息实体类

### 4.4 数据管理

数据管理模块分为插入记录和查询记录。

#### 4.4.1 插入记录

在 CMainFrame 类响应插入记录消息响应函数，在 CRecordDlg 对话框中显示表中的字 段，编辑字段的值，组装成记录信息保存在 CRecordEntity 中。并传递给 CRKDBMSDoc， CRKDBMSDoc 将数据库名、表和字段信息传递给 CRecordLogic。CRecordLogic 将记录信 息传递给 CRecordDao 并调用 CTableDao 类的方法更新表描述信息。CRecordDao 类将记录 信息保存到记录数据文件中。

1. CRecordDlg

添加记录对话框类，接收输入的记录信息。

1. CMainFrame

框架窗口类，响应添加记录事件。

1. CRKDBMDoc

文档类，保存数据库信息、表信息和记录信息，并控制界面显示和更新。

1. CRecordLogic

记录信息业务逻辑类，调用 CRecordDao 类方法将记录信息保存到表记录文件(\*.trd) 中，并调用 CTableDao 类方法更新表描述信息。

1. CRecordDao

记录信息数据操作类，将记录信息保存到表记录文件中。

1. CTableDao

表信息操作类，更新表描述信息。

1. CRecordEntity

记录信息实体类。

#### 4.4.2 查询记录

查询记录时，先读取表定义文件(\*.tdf)查询表结构，然后根据表结构读取记录文件(\*.trd) 中的数据，返回给逻辑层。

CMainFrame 类响应查询时间，在 CDBView 中获取选择的表。CRKDBMSDoc 将当前 表信息传递给 CRecordLogic，CRecordLogic 将表信息传递给 CRecordDao 类。CRecordDao 读取“\*.trd”文件，获取记录信息，并通过 CReocrdEntity 数组返回给逻辑类。最后在 CRecordsView 中显示记录信息。

1. CMainFrame

框架窗口类，响应查询记录操作。

1. CRKDBMSDoc

文档类，保存数据库信息、表信息和记录信息，并控制界面显示和更新。

1. CDBView 类

数据库视图类，显示数据库中的表名和字段名。

1. CRecordsView

记录视图类，显示表中的记录信息。

1. CRecordLogic

记录信息业务逻辑类，调用 CRecordDao 方法，获取记录信息。

1. CRecordDao

记录信息数据操作类，读取记录文件(.trd)，获取文件中的记录信息。

1. CRecordEntity

记录信息实体类。