

# 《C++现代程序设计》课程项目任务书

本文将作为《C++现代程序设计》的期末大作业的需求。同学们需对照本文，完成项目的分析、设计、编码和测试，并撰写项目报告。

## 一、项目开发组织形式

1. 项目开发必须以小组形式进行，每组 1~3 人。小组自由组合，推选出组长。
2. 要求小组成员分工明确，确定各自承担的任务量。
3. 成果由组长提交。

## 二、项目成果提交

1. 根据报告模板撰写项目报告。报告请保存为 PDF 格式。报告命名为：  
C++项目报告-小组名或姓名.pdf
2. 根据成员表现，填写项目评分表，保留.docx 格式，因为我还有在上面评分。
3. 将所有源代码（.cpp、.h、数据文件、Makefile）放在同一个目录中。请勿包含可执行代码，否则压缩包会很大。
4. 运行环境配置。配置信息写在一个文件里，命名为：config.pdf，存放在源代码目录中。其中包括操作系统、编译环境等信息。
5. 将报告、评分表和源代码目录打包在一个压缩包中，命名为：  
小组名.zip。
6. 将.zip 在 2023/12/31 23:59 前发送至：baizj@uestc.edu.cn，邮件主题为：xxx 小组的 C++项目成果。

## 三、评价依据

根据项目完成度、代码质量和报告质量进行。

1. 未按要求做的，得 0 分。
2. 过期提交的，得 0 分。
3. 项目代码和报告只要缺一样，得 0 分。
4. 项目代码明显是抄袭的，得 0 分。
5. 报告格式不合格的，得 0 分。

附件 1：项目要求

附件 2：项目报告模板（另附文件）

## 附件 1：项目要求

### 一、 内容和要求

运用 C++ 知识群和其它知识，对标 MySQL，开发一个微型数据库管理系统。

### 二、 系统功能要求

#### 一) 具体功能要求

##### 1. 存储功能

采用宿主操作系统的文件系统。数据表、索引（如果有）等用文件，目录结构和文件格式自定。

目录结构和文件命名方式示例：

```
data //目录
|---- person //目录，库名
|          |---- person.dat //表文件
|          +---- person.idx //同名表的索引文件
+----- other
```

##### 2. DDL

至少实现如下语句：

###### 1) create database

功能：创建数据库

语法：

```
create database <dbname>
```

*注：尖括号<>标记括起了参数。在使用时，须给出具体的名字。例如：*

```
create database person
```

*以下同此。*

###### 2) drop database

功能：删除数据库

语法：

```
drop database <dbname>
```

###### 3) use

功能：切换数据库

语法：

```
use <dbname>
```

#### 4) create table

功能：创建表

语法：

```
create table <table-name> (  
    <column> <type> [primary],  
    ...)
```

表名和列名：全英文小写，不含\_和特殊字符。

数据类型 **type**。作为简化，**type** 只考虑以下两种：

- ✧ **int**：采用 C++ 的默认长度，不考虑多种 **int**。
- ✧ **string**：最长 256 字符的定长串。编码类型：UTF-8。

如指定列是 **primary**（主键），则须为表建立索引。索引数据结构自定（例如二叉平衡树等）。

#### 5) drop table

功能：删除表

语法：

```
drop table <table-name>
```

注：删除表的同时，删除对应的索引（如果有的话）。

请自行设计所有的错误处理。以下同。

### 3. DML

至少实现如下语句：

#### 1) select

功能：根据条件（如果有）查询表，显示查询结果。

语法：

```
select <column> from <table> [ where <cond> ]
```

- ✧ **<column>**：<column-name> | \*。只需一个列名（如要更多，请自行设计）。\*表示所有列。
- ✧ **where 子句**：可选。如无，表示无条件查询。
  - **<cond>**：<column> <op> <const-value>
  - **<op>**：=、<、>三者之一

提示：如果表有索引，请使用它。以下同此。

异常：请自行设计可能出现的异常（例如表不存在等）及其处理方法。以下同此。

## 2) delete

功能：根据条件（如果有）删除表中的记录。

语法：

```
delete <table> [ where <cond> ]
```

✧ where 子句：语法同 select 的。

## 3) insert

功能：在表中插入数据。

语法：

```
insert <table> values (<const-value>[, <const-value>...])
```

注：字符串数据用双引号括起来

## 4) update

功能：根据条件（如果有）更新表中的记录。如无条件，则更新整张表。

语法：

```
update <table> set <column> = <const-value> [ where <cond> ]
```

✧ where 子句：语法同 select 的。

注：字符串数据用双引号括起来

## 二) 界面要求

模仿 MySQL 的交互式界面。例如：运行你设计的 DBMS，可能的操作序列如下：

```
> create table person (id int primary, name string)
```

[反馈信息在这里](#)

```
> use person
```

```
> insert person values(1001, "peter")
```

[反馈信息在这里](#)

```
> select name from person where id = 1001
```

[反馈信息在这里](#)

```
> exit
```

反馈信息请自行设计，但必须清晰明了。

### 三、 编码要求

1. 总体无特别要求，但 C++ 代码量必须占压倒性比例。
  - ✧ 可以使用 STL 库。注：多线程库属于 STL。
  - ✧ 不允许使用其他第三方库，包括网络库等。就是说，你的应用是个单机应用，无需采用 C/S 架构（但可以借鉴其思路）。
2. C++ 标准至少到 C++ 20。
3. 必须在 Linux 系统下开发。因为我在此系统下评估大家的代码。
4. 必须使用 make 来构建应用。
  - ✧ make 依赖文件用默认的名字：Makefile。
  - ✧ 可以使用 CMake 工具或其他类似工具。但请在打包发送给我前，先生成 Makefile。这样，我只需使用 make 来构建应用。

注：强烈建议文件和目录用全英文命名。
5. 测试（含单元测试）代码请自行设计编写。测试必须覆盖 5 条 DDL、4 条 DML 语句，以及其他你自行设计的语句。这同时意味着，你在项目报告中，必须阐述这些语句的实现原理，以及它们的测试结果（截图）。
6. 强烈建议使用 Git 进行代码托管。请自行研究 Git 的使用。推荐使用国产 Gitee。