**数据库设计文档**

|  |  |
| --- | --- |
| **文档编号** | 数据库设计文档-V1.0 |
| **文档类别** | XXX |
| **密 级** | 机密 |
| **版本信息** | V1.0 |
| **建立日期** | 2025-06-28 |
| **创 建 人** | 第1小组全体成员 |
| **审 核 者** |  |
| **批 准 人** |  |
| **批准日期** | 2025-06-28 |
| **存放位置** | SQLITE |
| **编辑软件** | WPS |

更新记录

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 日期 | 作者 | 版本 | 变更内容 | 类型 | 批准人 | 批准日期 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

[类型：修改Modified，新建New，增加Add，删除Deleted]

目 录

[更新记录 II](#_Toc25104)

[目 录 III](#_Toc22805)

[1. 前言 1](#_Toc2954)

[1.1. 编写目的 1](#_Toc399)

[1.2. 术语说明 1](#_Toc15236)

[1.3. 数据库结构 1](#_Toc11523)

[1.3.1. 数据库文件位置 1](#_Toc24606)

[1.3.2. 数据库表说明 1](#_Toc16120)

[1.4. 数据库操作 2](#_Toc19737)

[1.4.1. 创建数据库连接 2](#_Toc10520)

[1.4.2. 查询用户信息 2](#_Toc11938)

[1.4.3. 更新用户信息 2](#_Toc2916)

[1.4.4. 插入新用户 3](#_Toc16175)

[1.4.5. 查询更新是否成功 3](#_Toc20227)

[1.4.6. 数据库关闭 4](#_Toc16792)

[1.5. 错误处理 4](#_Toc22681)

[1.6. 安全与性能考虑 4](#_Toc2187)

[2. 总结 4](#_Toc5702)

1. 前言

## 编写目的

该文档主要描述“小雅Plus”在线考试系统中用户、课程、考试、题目等信息在数据库中的实现，并实现数据库中相关信息的实时更新。本项目使用 Mysql/ODBC 数据库来存储与用户相关的信息。此外，还介绍了数据库表的详细说明。

## 术语说明

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| 小雅Plus | 在线考试系统。 |
| 考频 | 题目被用于组卷的次数统计指标，用于衡量题目使用频率。 |
| 自动组卷 | 系统根据设定的题型、难度、数量等条件自动从题库中选取题目生成试卷的功能。 |
| 主观题 | 需要人工阅卷的题目类型，如简答题、论述题等。 |
| 客观题 | 可由系统自动阅卷的题目类型，如单选题、多选题等。 |

[编写说明：列出文档中的专业属于和缩写，并给出其解释。]

## 数据库结构

### 数据库文件位置

云端服务器的sql数据库中。

### 数据库表说明

#### **<STUDENTS>数据表**

这个表用于存储学生的基本信息，包括学生ID、学生姓名，年级、学院和密码。

#### **数据列描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULLABLE** | **KEY** | **COMMENTS** |
| student\_id | INT | N | PK | 学生ID |
| name | VACHAR | N |  | 姓名 |
| grade | VACHAR | N |  | 年级 |
| college | VACHAR | N |  | 学院 |
| password | VACHAR | Y |  | 密码 |

#### **主键描述**

[编写说明：描述主键名称和对应数据列。说明栏中可以书写一些设计因素。]

|  |  |  |
| --- | --- | --- |
| **主键名称** | **数据列名称** | **说明** |
| student\_id | student\_id | 学生ID，用于唯一标识一名学生 |

#### **<TEACHERS>数据表**

这个表用于存储教师的基本信息，包括教师ID、教师姓名，学院和密码。

#### **数据列描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULLABLE** | **KEY** | **COMMENTS** |
| teacher\_id | INT | N | PK | 教师ID |
| name | VACHAR | N |  | 姓名 |
| college | VACHAR | N |  | 学院 |
| password | VACHAR | Y |  | 密码 |

#### **主键描述**

|  |  |  |
| --- | --- | --- |
| **主键名称** | **数据列名称** | **说明** |
| teacher\_id | teacher\_id | 教师ID，用于唯一标识一名教师 |

#### **<COURSES>数据表**

这个表用于存储课程的基本信息，包括课程ID、教师ID、学院和课程名称。

#### **数据列描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULLABLE** | **KEY** | **COMMENTS** |
| course\_id | INT | N | PK | 课程ID |
| teacher\_id | INT | N | FK | 教师ID |
| college | VACHAR | N |  | 学院 |
| course\_name | VACHAR | N |  | 课程名称 |

#### **主键描述**

|  |  |  |
| --- | --- | --- |
| **主键名称** | **数据列名称** | **说明** |
| course\_id | course\_id | 课程ID，用于唯一标识一节课程 |

#### **<QUESTIONS>数据表**

这个表用于存储题目的基本信息，包括题目ID、题型、分值、题干、答案等。

#### **数据列描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULLABLE** | **KEY** | **COMMENTS** |
| question\_id | INT | N | PK | 题目ID |
| question\_type | VACHAR | N |  | 题型 |
| course\_name | VACHAR | N |  | 课程名称 |
| score | INT | N |  | 分值 |
| content | TEXT | N |  | 题干 |
| answer | TEXT | N |  | 答案 |
| creator\_id | INT | N | FK | 出题人ID |
| modifier\_id | INT | N | FK | 最后修改人ID |
| frequency | INT | N |  | 考频 |
| is\_public | BOOLEAN | N |  | 是否公开 |

#### **主键描述**

|  |  |  |
| --- | --- | --- |
| **主键名称** | **数据列名称** | **说明** |
| question\_id | question\_id | 题目ID，用于唯一标识一个题目 |

#### **<EXAMS>数据表**

这个表用于存储考试的基本信息，包括考试ID、考试名称、考试日期、开始时间、结束时间等。

#### **数据列描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULLABLE** | **KEY** | **COMMENTS** |
| exam\_id | INT | N | PK | 考试ID |
| exam\_name | VACHAR | N |  | 考试名称 |
| course\_id | INT | N | FK | 课程ID |
| exam\_data | DATA | N |  | 考试日期 |
| start\_time | TIME | N |  | 开始时间 |
| end\_time | TIME | N |  | 结束时间 |
| total\_score | INT | N |  | 总分 |
| is\_published | BOOEAN | N |  | 是否发布 |
| is\_completed | BOOLEAN | N |  | 是否组卷完成 |

#### **主键描述**

|  |  |  |
| --- | --- | --- |
| **主键名称** | **数据列名称** | **说明** |
| exam\_id | exam\_id | 考试ID，用于唯一标识一场考试 |

#### **<STUDENTS\_COURSES>数据表**

这个表用于存储学生的课程成绩。

#### **数据列描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULLABLE** | **KEY** | **COMMENTS** |
| student\_id | INT | N | PK,FK | 学生ID |
| course\_id | INT | N | PK,FK | 课程ID |
| exam\_score | DECIMAL | N |  | 考试成绩 |

#### **主键描述**

|  |  |  |
| --- | --- | --- |
| **主键名称** | **数据列名称** | **说明** |
| student\_id | student\_id | 学生ID，用于唯一标识一名学生 |
| course\_id | course\_id | 课程ID，用于唯一标识一节课程 |

#### **<STUDENTS\_ANSWERS>数据表**

这个表用于存储学生的答案信息。

#### **数据列描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULLABLE** | **KEY** | **COMMENTS** |
| exam\_id | INT | N | PK,FK | 考试ID |
| question\_id | INT | N | PK,FK | 题目ID |
| student\_id | INT | N | PK,FK | 学生ID |
| student\_aswer | TEXT | N |  | 学生答案 |
| score | DECIMAL | N |  | 得分 |

#### **主键描述**

|  |  |  |
| --- | --- | --- |
| **主键名称** | **数据列名称** | **说明** |
| exam\_id | exam\_id | 考试ID，用于唯一标识一场考试 |
| question\_id | question\_id | 题目ID，用于唯一标识一个题目 |
| student\_id | student\_id | 学生ID，用于唯一标识一名学生 |

#### **<EXAM\_QUESTIONS>数据表**

这个表用于存储考试与题目的关系。

#### **数据列描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULLABLE** | **KEY** | **COMMENTS** |
| exam\_id | INT | N | PK,FK | 考试ID |
| question\_id | INT | N | PK,FK | 题目ID |
| question\_order | INT | N |  | 题目序号 |

#### **主键描述**

|  |  |  |
| --- | --- | --- |
| **主键名称** | **数据列名称** | **说明** |
| exam\_id | exam\_id | 考试ID，用于唯一标识一场考试 |
| question\_id | question\_id | 题目ID，用于唯一标识一个题目 |

#### **<EXAM\_PERMISSIONS >数据表**

这个表用于存储考试权限信息。

#### **数据列描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULLABLE** | **KEY** | **COMMENTS** |
| exam\_id | INT | N | PK,FK | 考试ID |
| teacher\_id | INT | N | PK,FK | 教师ID |
| is\_creator | BOOLEAN | N |  | 是否为创建者 |

#### **主键描述**

|  |  |  |
| --- | --- | --- |
| **主键名称** | **数据列名称** | **说明** |
| exam\_id | exam\_id | 考试ID，用于唯一标识一场考试 |
| question\_id | question\_id | 题目ID，用于唯一标识一个题目 |

#### **<QUESTION\_FAVORITES >数据表**

这个表用于存储教师收藏题库的信息。

#### **数据列描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NAME** | **TYPE** | **NULLABLE** | **KEY** | **COMMENTS** |
| question\_id | INT | N | PK,FK | 题目ID |
| teacher\_id | INT | N | PK,FK | 教师ID |

#### **主键描述**

|  |  |  |
| --- | --- | --- |
| **主键名称** | **数据列名称** | **说明** |
| question\_id | question\_id | 题目ID，用于唯一标识一个题目 |
| teacher\_id | teacher\_id | 教师ID，用于唯一标识一名教师 |

## 数据库操作

### 创建数据库连接

在操作数据库时，首先需要创建并打开数据库连接，在数据库打开后进行查询和数据更新操作。

### 查询用户信息

通过以下 SQL 查询操作，可以查找数据库中是否已经存在指定用户名的数据。

1.查询用户名是否存在：

query.prepare("SELECT ip\_address FROM text WHERE username = :username");

query.bindValue(":username", ui->nameLineEdit->text());

如果查询返回结果，并且 IP 地址匹配，则允许进入聊天室。

2.查询字段的SQL示例：

SELECT \* FROM text WHERE username = :username;

SELECT \* FROM text WHERE hostname = :hostname;

SELECT \* FROM text WHERE ip\_address = :ip\_address;

### 更新用户信息

当用户修改了 username、hostname 或 ip\_address 时，使用以下 SQL 更新对应字段。

1.更新用户名

query.prepare("UPDATE text SET username = :newname WHERE username = :oldname");

query.bindValue(":newname", newData);

query.bindValue(":oldname", oldData);

2.更新主机名

query.prepare("UPDATE text SET hostname = :newname WHERE hostname = :oldname");

query.bindValue(":newname", newData);

query.bindValue(":oldname", oldData);

3.更新IP地址

query.prepare("UPDATE text SET ip\_address = :newname WHERE ip\_address = :oldname");

query.bindValue(":newname", newData);

query.bindValue(":oldname", oldData);

### 插入新用户

当新用户注册时，通过以下 SQL 插入新的用户记录到数据库。

query.prepare("INSERT INTO text (username, hostname, ip\_address) VALUES (:username, :hostname, :ip\_address)");

query.bindValue(":username", ui->nameLineEdit->text());

query.bindValue(":hostname", QHostInfo::localHostName());

query.bindValue(":ip\_address", getIP());

if (query.exec()) {

qDebug() << "insert success";

} else {

qDebug() << "insert failed: " << query.lastError().text();

}

### 查询更新是否成功

在更新操作之后，可以通过再次查询新数据来验证更新是否成功。例如，更新用户名后，可以通过以下 SQL 查询验证。

query.prepare("SELECT \* FROM users WHERE username = :newname");

query.bindValue(":newname", newData);

if (query.next() && query.value("username").toString() == newData) {

qDebug() << "更新成功！";

} else {

qDebug() << "更新未找到。";

}

### 数据库关闭

每次数据库操作完成后，应该关闭数据库连接以释放资源。

db.close();

## 错误处理

1.SQL 异常处理：在所有数据库操作中，通过捕获 SQL 执行异常，使用LAST\_ERROR()函数获取错误信息，并记录到系统日志中。

2.数据完整性约束：利用数据库外键约束、唯一索引等机制，确保数据完整性，避免无效数据插入。

3.并发冲突处理：对关键操作（如成绩计算、组卷）使用数据库事务和锁机制，防止并发冲突。

## 安全与性能考虑

1.防止 SQL 注入：通过使用 bindValue 方法来绑定变量，防止 SQL 注入攻击。

2.数据库连接：将数据库部署到云端，是客户端在访问的时候可以随时访问，只需要一个驱动就可以进行数据操作，而且将操作传送至云端，减少内存开销，提高操作效率。

1. 总结

本数据库设计基于项目的基本需求，构建了包含用户管理、题库管理、考试管理、成绩分析等功能的完整数据模型。数据库结构遵循第三范式（3NF），确保数据完整性和一致性，同时通过索引设计和性能优化措施满足系统的响应时间和吞吐量要求。后续可根据实际使用情况进一步优化数据库结构和查询性能，以支持系统的持续迭代和扩展。