项目名称

（The English Name）

**数据库设计**

**XXX项目小组**

**修订表**

| 编号 | 生成版本 | 修订人 | 修订章节与内容 | 修订日期 |
| --- | --- | --- | --- | --- |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

**审批记录**

| 版本 | 审批人 | 审批意见 | 审批日期 |
| --- | --- | --- | --- |
| 1.0.0 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目 录**

[1. 引言 5](#_Toc200457859)

[1.1 目的 5](#_Toc200457860)

[1.2 读者对象 5](#_Toc200457861)

[1.3 参考资料 5](#_Toc200457862)

[1.4 术语与缩略语 5](#_Toc200457863)

[2. 数据库环境说明 5](#_Toc200457864)

[3. 数据库的命名规则 5](#_Toc200457865)

[4. 逻辑设计 5](#_Toc200457866)

[5. 物理设计 6](#_Toc200457867)

[5.1 表汇总 6](#_Toc200457868)

[5.2 表A 6](#_Toc200457869)

[5.3 表B 6](#_Toc200457870)

[6. 数据库对象设计 6](#_Toc200457871)

[7. 安全性设计 7](#_Toc200457872)

[7.1 防止用户直接操作数据库的方法 7](#_Toc200457873)

[7.2 用户账号密码的加密方法 7](#_Toc200457874)

[7.3 角色与权限 7](#_Toc200457875)

[8. 优化 7](#_Toc200457876)

[9. 数据库管理与维护说明 7](#_Toc200457877)

# 引言

## 目的

## 读者对象

## 参考资料

|  |  |  |  |
| --- | --- | --- | --- |
| 资料名称[标识符] | 出 版 单 位 | 作 者 | 日 期 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

*格式说明：列出本文档的所有参考资料（可以是非正式的出版物），对于不为本单位所拥有的资料，要注明来源、引用章节，如果必要需将部分章节复印做为附件附在文档后。*

## 术语与缩略语

|  |  |
| --- | --- |
| **术语、缩略语** | **解 释** |
|  |  |
|  |  |
|  |  |
|  |  |

# 数据库环境说明

*1．说明所采用的数据库系统，设计工具，编程工具等。*

*2．详细配置。*

# 数据库的命名规则

*1．完整并且清楚的说明本数据的命名规则。*

*2．若本数据库的命名规则与机构的标准不完全一致，请做出解释。*

# 逻辑设计

*数据库设计人员根据需求文档，创建与数据库相关的那部分实体关系图（ERD）。如果采用面向对象方法（OOAD），这里的实体相当于类（Class）。*

# 物理设计

1．主要是设计表结构。一般地，实体对应于表，实体的属性对应于表的列，实体之间的关系成为表的约束。逻辑设计中的实体大部分可以转换成物理设计中的表，但是它们并不一定是一一对应的。

*2．对表结构进行规范化处理（使用第一、二、三范式）。*

## 表汇总

|  |  |
| --- | --- |
| 表 名 | 功能说明 |
| 表A |  |
| 表B |  |
| 表C |  |
| … | … |

## 表A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表 名 |  | | | | |
| 功能说明 |  | | | | |
| 中文列名 | 字段名 | 数据类型  （精度范围） | 空否 | 约束条件 | 含义说明 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 补充说明 |  | | | | |

## 表B

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表 名 |  | | | | |
| 功能说明 |  | | | | |
| 中文列名 | 字段名 | 数据类型  （精度范围） | 空否 | 约束条件 | 含义说明 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 补充说明 |  | | | | |

......

# 数据库对象设计

*主要对象包括视图、触发器、存储过程、函数、索引等。*

# 安全性设计

*提高软件系统的安全性应当从“管理”和“设计”两方面着手。这里仅考虑数据库的安全性设计。*

## 防止用户直接操作数据库的方法

*用户只能用账号登录到应用软件，通过应用软件访问数据，而没有其他途径可以操作数据库。*

## 用户账号密码的加密方法

*对用户账号的密码进行加密处理，确保在任何地方都不会出现密码的明文。*

## 角色与权限

|  |  |  |
| --- | --- | --- |
| 角 色 | 可以访问的表与列 | 操 作 权 限 |
| 角色A |  |  |
|  |  |
| 角色B |  |  |
|  |  |

确定每个角色对数据库表的操作权限，如创建、检索、更新、删除等。每个角色，则每个角色拥有刚好能够完成任务的权限，不多也不少。在应用时再为用户分配角色，则每个用户的权限等于他所兼角色的权限之和。

# 优化

|  |  |  |
| --- | --- | --- |
| 优 先 级 | 优化对象（目标） | 措 施 |
|  |  |  |

*分析并优化数据库的“时-空”效率，尽可能地“提高处理速度”并且“降低数据库占用的空间”。*

*1。分析“时-空”效率的瓶颈，找出优化对象（目标），并确定优先级。*

*2．当优化对象（目标）之间存在对抗时，给出折衷方案。*

*3．给出优化的具体措施，例如优化数据库环境参数，对表格进行反规范化处理等。*

# 数据库管理与维护说明

在设计数据库的时候，及时给出管理与维护本数据库的方法，有利于将来撰写出正确完备的用户手册。