**XXXXXXXXX**

|  |  |
| --- | --- |
| 文件编号 |  |
| 当前版本 | V0.2 |
| 作 者 | 吴泉 |
| 审 批 者 |  |
| 发布日期 |  |

**修订记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **日期** | **作者** | **修订版本** | **修改类型** | **修改描述** |
| 1 | 2016.6.24 | 吴泉 | V0.1 | A添加 | 新建 |
| 2 | 2016.06.24 | 吴泉 | V0.2 | 根据黄子华的检视意见进行修改 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

（备注：修改类型分为：A添加 M修改， D删除）

# 数据库编码规范

目录

[文档修订记录 1](#_Toc454547015)

[Sowell-数据库编码规范 2](#_Toc454547016)

[1. 目的 2](#_Toc454547017)

[2. 范围 3](#_Toc454547018)

[3. 术语 3](#_Toc454547019)

[4. 设计概要 3](#_Toc454547020)

[1) 设计原则 3](#_Toc454547021)

[2) 设计更新 3](#_Toc454547022)

[5. 命名总体原则 4](#_Toc454547023)

[6. 命名规范（逻辑对象） 4](#_Toc454547024)

[1) 数据库命名 4](#_Toc454547025)

[2) 数据库对象命名 4](#_Toc454547026)

[7. 数据库操作原则 5](#_Toc454547027)

[1) 建立、删除、修改库表操作 5](#_Toc454547028)

[2) 添加、删除、修改表数据 5](#_Toc454547029)

# 目的

为了统一公司软件开发的设计过程中关于数据库设计时的命名规范和具体工作时的编程规范，便于交流和维护，特制定此规范。

# 范围

本规范适用于研发部全体人员，作用于软件项目开发的数据库设计、维护阶段。

# 术语

* 数据库对象：在数据库软件开发中，数据库服务器端涉及的对象包括物理结构和逻辑结构的对象；
* 物理结构对象：是指设备管理元素，包括数据文件和事务日志文件的名称、大小、目录规划、所在的服务器计算极名称、镜像等，应该有具体的配置规划。一般对数据库服务器物理设备的管理规程，在整个项目/产品的概要设计阶段予以规划；
* 逻辑结构对象：是指数据库对象的管理元素，包括数据库名称、表、字段、视图、索引、序列、触发器、函数、数据类型、数据库安全性相关的设计、数据库配置有关的设计以及数据库中其他特性处理相关的设计等。

# 设计概要

## 设计原则

* 数据库部署上要防止单点故障；
* 最小化数据库依赖原则，特殊情况需要引入函数、触发器、视图等数据库高级功能，需要发起内部评审，不编写复杂的SQL；
* 表数据量和存储空间水位保持设计，对于会一直增长数据的表，例如用户行为记录表，需要设计相应的归档策略；
* 数据库整体存储空间设计，根据每一个表的水位，设计数据的存储空间；
* 表数据被查询索引设计，当表的数据量超过1万，并且根据某个字段查询的频率很高时，需要设计索引；
* 表与表之间不容许建立外键，数据库层外键关联，当并发量或者数据库表的数据比较大时，会直接影响数据库操作性能，因此关联关系应该放到程序中去实现，如果有关联查询时，需要针对关联字段建立索引；
* 每张表需建立主键，且主键字段越精简越好；
* 每个字段设置默认值，避免出现NULL；
* 设计表结构时，尽量遵循三范式，针对使用率较高的表，可考虑降低范式，增加冗余，利用空间换效率；
* 设计表字段控制内容存储长度，能少不多；
* 自增字段类型必须是整型，推荐类型为INT4 或者是INT8类型。并且自增字段必须是主键或者是主键的一部分。

## 设计更新

* 运行阶段，由数据库管理员进行维护，数据库日志做好归档；
* 如对表结构进行修改，应该走软件【测试发布流程】，并且同时更新数据库设计说明书，需要提供数据备份和回退脚本；
* 修改数据库要通过SQL，禁止其它方式对数据进行修改；
* 修改数据库的SQL要添加说明后保存备查。

# 命名总体原则

* 设定的前缀一律用小写字母；
* 标识名称命名全部小写；
* 整个命名的全长不得超过30个字母；
* 全部使用字母和下划线‘\_’，不能使用中文和其他字符，有特别情况允许使用末尾数字编号。例如：t\_finace1, t\_finace2...；
* 命名名称来自于业务，全部采用英文单词；
* 英文单词过长可以采用通用的缩写，尽量表达出业务的含义；
* 如需要两个以上的英文单词做标识名称，单词之间要用下划线‘\_’连接；
* 名称全是由名词组成的，名词由大范围到小范围排序取名；
* 完成某功能的名称，如函数和过程，以动宾形式取名；

# 命名规范（逻辑对象）

## 数据库命名

数据库的命名要求使用与数据库意义相关联的英文字母，即<业务系统名称>；例如：客户资料数据库的命名为customer\_info，dcmp数据库命名为dcmp。

## 数据库对象命名

* **表**

表的命名必须以“t\_”（Table缩写）开头，格式为：t\_[系统标识]\_<数据表类型标识>\_<表标识>。

其中，[ ]表示可选项，依据实际情况而增加；<表标识>要求与表意义相关联的英文字母，例如：t\_license\_info。

数据表大致分为：业务数据表、基本编码表、辅助编码表、系统信息表、累计数据表、结算数据表、决策数据表；

基本编码表用base标志

累计数据表用count标志

系统信息表用info标志 ...

例如：t\_trade\_base\_trade\_code,t\_trade\_info\_help,t\_license\_info ...

* **字段/域**

根据业务要求进行命名，不需设定固定的前缀。

* **索引**

针对数据库表中一个或多个字段建立的索引的命名格式应以“idx\_”开头，索引列名间用\_隔开，即为idx\_columnname1\_columnname2\_…

其中，columnname1是数据库表中（第一个）索引字段的名称或名称简写；columnname2是数据库表中（第二个）索引字段的名称或名称简写；索引名的总长必需符合数据库的规定。

例：idx\_client\_uuid (表示在字段client\_uuid上创建索引)

PostgreSQL数据库索引和表关联，原则上按上述规则进行索引建立，当长度超标时，可以对字段名进行缩写。

* **视图**

视图的命名必须以“v\_”(View 缩写)开头，格式为： v\_<视图类型>\_[系统标识] \_<视图标识>。

其中，视图类型参见“表的分类说明”；[系统标识\_]是可选项，依据情况而增加；<视图标识>应与视图意义相关联的英文字母。

例：v\_user\_detail\_info

*Note：原则上，不容许使用视图，如果需要使用，请发起研发部内部评审；*

* **触发器**

触发器的的命名必须符合tr\_<表名>\_<i,u,d的任意组合>格式。

其中,tr表示是触发器；<i,u,d的任意组合>是与触发器意义相关联的英文字母。

例：tr\_user\_info\_iu（表示对user\_info表进行插入、更新的触发器）

*Note：原则上，不容许使用触发器，如果需要使用，请发起研发部内部评审；*

* **函数**

函数的命名必须符合fn\_[系统标识]\_<函数标识>格式。

其中，fn表示是函数，[系统标识]为可选项，依情况而定；<函数标识>是与函数意义相关联的英文字母。

例：fn\_create\_id（以动宾方式取名）

*Note：原则上，不容许使用函数，如果需要使用，请发起研发部内部评审；*

# 数据库操作原则

## 建立、删除、修改库表操作

* 在开发环境中，对于自己的库表可任意进行修改、删除操作；但需保留相应的建表语句和说明，与建表人建表时间。转测试时，需要提供相应的SQL语句。

## 添加、删除、修改表数据

* 在开发环境中，开发人员所开发模块独自使用的库表，可自由操作表中数据；

对其他模块关联的库表，应取得其他模块的开发人员同意后再执行操作。