宏天软件

X5数据库设计和管理规范

Winston Yan

2013-11-11

# 命名规范

## 字段命名规范

1. 格式概述

[单词1]\_[单词2]\_[单词3～N]\_

采用1～多个单词，每个单词之间用下划线分隔，并以下划线作为后缀。

1. 纯大小写

表名和字段名全大写（对oracle）或全小写（对mysql），不允许出现大小写混合的情况。

1. 下划线后缀

字段后面增加下划线后缀。好处是：

* 1. 对单个单词时，可避免和关键字冲突。

如：id/name/version/desc/comment/remark/key等。

* 1. 可大量使用单个单词，字段名称总长度缩短。
  2. 容易设计实体基类的公共方法或成员变量。

1. 实体字段命名规则
2. 实体自己字段

对于该实体自己的字段，能够用单个单词或缩写尽量用，如id\_、key\_、name\_、desc\_等，格式统一以及方便写sql。

1. 业务意义完全一致的的字段统一命名

如alias、code、key等，同一用key\_表示。

1. 实体关联字段

关联的外键 或 用到的关联表字段，用2到3个单词表达，最好加上关联表的表名，格式参考这样：[关联表表名（去掉模块前缀）]\_[关联字段名]\_

如user\_group\_key\_（关联表名为sys\_user\_group，去掉sys\_前缀）。

1. 公共结构的字段
   1. 树形结构

采用统一的字段来定义树形结构，在不同的表中，如果需要支持树形，均使用相同的字段和类型，这样可以方便统一设计和开发。

## 表命名规范

1. 格式：[子系统缩写]\_[模块缩写（可选）]\_[表名]

* 子系统缩写

采用2～3个字母，可以是约定俗成的缩写，也可以是单词前几个字母。

如HI表示HISTORY，SYS表示SYSTEM，BPM表示Business Process Management等。

* 模块缩写

为了区分属于同一模块的几张表。可选，不强制要求。

如工作日历相关的表，可以是SYS\_CAL\_SETTING，SYS\_CAL\_USER、SYS\_CAL\_SHIFT 等等。

1. 表名

由1～多个单词构成，为了避免表名过长，一般采用较短的单词或者单词缩写。

## 其它命名规范

1. 外键

格式：[A表名（去掉子系统前缀）]\_R\_[B表名（去掉子系统前缀）]

如：DEFN\_R\_GL\_TYPE、NODE\_SET\_R\_NODE\_USR

# 数据类型规范

1. 自定义数据类型（Domain）

在powerdesigner中，通过Domain定义常用的字段类型，在维护字段类型时，从已有的Domain中选择。

对于具有特殊业务意义的字段，设置专门的Domain进行维护，避免同样字段在不同的表中数据类型或长度不一致。

1. 常用的字段类型

状态：smailint，考虑到状态可能存在等级比较，用短整数比较方便编程。

类型：varchar(20)，类型用英文的单词（1个或多个）的缩写表示，类型之间无 关联，类型通过equal来查询。

是否：char(1)，1/0，最省空间。

1. 由DBA制订Domain

所有的Domain均有DBA负责制订，其他的用户只能从Domain中选择数据类型（含长度），而不能自行定义数据类型。这样可以避免出现奇怪的数据结构和长度，或同一业务的字段在不同的表被设计了不同的数据结构。

# 注释规范

1. 表注释

在PowerDesigner的Table Properties中的comment中写表的注释。

格式如下（其中修改者及时间和修改内容可以有多组）：

@数据表名称：

@功能描述：（1～n行）

@创建者及时间：

@修改者及时间：

@修改内容：

例子：

@数据表名称：班次表 cal\_shift

@功能描述：维护日班、夜班等班次，每个班次有工时，方便统计工作量。

@创建者及时间：winstonyan@2013-11-05 10:20

@修改者及时间：winstonyan@2013-11-06 09:30

@修改内容：增加了字段“is\_actived\_ 是否有效 char(1)”

@修改者及时间：mansan@2013-11-07 09:30

@修改内容：增加了字段“is\_night\_ 是否夜班 char(1)”

根据这样的注释，生成的Java实体的头部注释可读性会很好。

1. 字段注释

格式如下（修改者及时间和修改内容可以重复多条）：

@字段名：

@业务说明：

@修改者及时间：

@修改内容：

例子：

@字段名：abs\_reason\_ 缺勤事由

@业务说明：记录该缺勤的原因

@修改者及时间：winstonyan@2013-11-06 10:30

@修改内容：从varchar(128)更改为varchar(255)

# 数据库管理

## PowerDesigner使用规范

1. 用Domain管理数据类型

详见第1章的第4节。

1. 按模块划分表，一个模块一个物理图。

系统的数据表很多，按模块进行表划分比较清晰。

一个模块一个物理图。

1. 表或视图颜色

最多使用三种颜色：

* 模块内部的表用蓝色。
* 模块外部的表用黄色。
* 废弃的表用紫色。去掉Generate勾选，并在合适的时候删除掉。

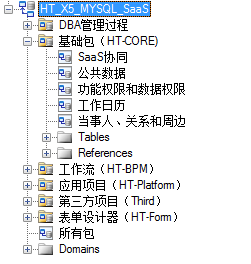
1. 一对一关联

表的主键是另一个表的主键，一对一关联；

通过Traceability Link（虚线箭头）关联加上描述处理。

## 按包结构来管理数据表

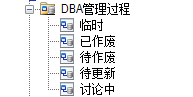
结构如下图：



1. 按子项目来分包。
2. 每个包下面按大的功能模块进行再次划分。
3. 如果功能模块的表比较少，则将几个模块合并到一个物理模型图中。
4. 采用全局的Domain，供所有的数据表使用。

## 表生命周期

增加一个DBA管理使用的包，用于管理下列生命周期的表。



### 临时测试数据结构

是一些当前业务模块测试时使用的数据结构，可以随时被删除。

该周期的表要去掉Generate的勾选。

可以用于开发或演示时，临时表的设计。

### 讨论中数据结构

是数据结构处于讨论中，还没有确定下来。

这类表移到“待更新”物理模型图中要从“讨论中”物理模型图中删除掉。

该周期的表要去掉Generate的勾选。

### 待更新的数据结构

是数据结构已经确定下来，但是还没有更新到数据库中。

这类表移到相应的正式模块下的物理模型图中要从“待更新”物理模型图中删除掉。

该周期的表要去掉Generate的勾选。

### 待作废的数据结构

在数据库中以前被业务正常使用，现在已经不再使用，但是还没有进行被作废的数据表中数据的迁移，没有完全作废的数据结构。如果要把这些数据结构进行作废，需要先进行数据迁移，以及其他相关处理。

在“待作废”物理模型图中只是保存一个复制，在实际的正式模块的物理模型图中依然存在。

### 已作废的数据结构：

在数据库已经不再被使用的业务数据表，相关的数据迁移已经完成，但是数据表还没有删除，相关的文档没有更新。

在实际的正式模块的物理模型图中该表已经不存在了，只在“已作废”物理模型图中存在。