# 数据库设计规范

## 数据库表命名规范

此规范包括表名命名规范，索引命名规范，外键命名规范和字段名命名规范。

### 表名命名规则

数据库表的命名以是名词的复数形式且都为小写，如cities, categories, friends等等

如果表名由几个单词组成，则单词间用下划线("\_")分割，如subscribed\_pois,poi\_categories等

表名尽量用全名

表名限制在30个字符内。当表的全名超过30字符时，可用缩写来减少表名的长度，如description --> desc；information --> info；address --> addr等

表字段名命名规则

字段名为小写

字段名为有意义的单词，或单词的缩写

如果字段由几个单词组成，则单词间用下划线("\_")分割，如client\_id，post\_code等

字段名限制在30个字符内。当字段名超过30字符时，可用缩写来减少字段名的长度，如description --> desc；information --> info；address --> addr等

### 索引命名规则

索引须按照IDX\_table\_<column>\_<column>,其中<table>是建立索引的表名，<column>是建立索引的字段名

索引名限制在30个字符内。当索引名超过30字符时，可用缩写来减少索引名的长度，如description --> desc；information --> info；address --> addr等

### 主键、外键命名规则

主键按照PK\_<table>的规则命名，其中<table>为数据库表名

唯一键按照UK\_<table>\_<column>的规则命名，其中<table>为数据块表名，<column>为字段名

外键按照FK\_<pppp>\_<cccc>\_<nn>的规则命名，其中<pppp>为父表名，<cccc>为子表名，<nn>为序列号

### 数据库规范化设计的一些基本要求：

一、表中应该避免可为空的列。

虽然表中允许空列，但是，空字段是一种比较特殊的数据类型。数据库在处理的时候，需要进行特殊的处理。如此的话，就会增加数据库处理记录的复杂性。当表中有比较多的空字段时，在同等条件下，数据库处理的性能会降低许多。

解决方法：一是通过设置默认值的形式，来避免空字段的产生。二是若一张表中，允许为空的列比较多，接近表全部列数的三分之一。而且，这些列在大部分情况下，都是可有可无的。若数据库管理员遇到这种情况，建议另外建立一张副表，以保存这些列。

二、表不应该有重复的值或者列。

三、表中记录应该有一个唯一的标识符。

四、数据库对象要有统一的前缀名。

五、尽量只存储单一实体类型的数据。

六、尽量符合数据库的几个范式。

设计规范

所有字段在设计时，除以下数据类型timestamp、image、datetime、smalldatetime、uniqueidentifier、binary、sql\_variant、binary 、varbinary外，必须有默认值。字符型的默认值为一个空字符值串’’；数值型的默认值为数值0；逻辑型的默认值为数值0；

其中：系统中所有逻辑型中数值0表示为“假”；数值1表示为“真”。

datetime、smalldatetime类型的字段没有默认值，必须为NULL。

当字段定义为字符串形时建议使用varchar而不用nvarchar。

字段的描述

数据库中每个字段的描述(Description)如下：

表内的每一个值只能被表达一次

表内的每一行都应当被唯一的标示

表内不应该存储依赖于其他键的非键信息

如果字段事实上是与其它表的关键字相关联而未设计为外键引用，需建索引。

如果字段与其它表的字段相关联，需建索引。

## SQL 开发代码规范

我们规定sql语句编码的时候程序员需要遵循以下规范：

a 所有的关键字的所有字母必须大写。如果一个常量由几个词组成，我们规定两个词之间使用下划线（\_）来划分。表名、列名、视图名或它们的别名必须和它们的定义保持一致。

b 注释必须要规范。

其他设计技巧

)避免使用触发器

触发器的功能通常可以用其他方式实现。在调试程序时触发器可能成为干扰。假如你确实需要采用触发器，你最好集中对它文档化。

)使用常用英语(或者其他任何语言)而不要使用编码在创建下拉菜单、列表、报表时最好按照英语名排序。假如需要编码，可以在编码旁附上用户知道的英语。

)保存常用信息。让一个表专门存放一般数据库信息非常有用。在这个表里存放数据库当前版本、最近检查/修复、关联设计文档的名称、客户等信息。这样可以实现一种简单机制跟踪数据库，当客户抱怨他们的数据库没有达到希望的要求而与你联系时，这样做对非客户机/服务器环境特别有用。

)包含版本机制。在数据库中引入版本控制机制来确定使用中的数据库的版本。时间一长，用户的需求总是会改变的。最终可能会要求修改数据库结构。把版本信息直接存放到数据库中更为方便。

)编制文档。对所有的快捷方式、命名规范、限制和函数都要编制文档。采用给表、列、触发器等加注释的数据库工具。对开发、支持和跟踪修改非常有用。对数据库文档化，或者在数据库自身的内部或者单独建立文档。这样，当过了一年多时间后再回过头来做第2 个版本，犯错的机会将大大减少。

)测试、测试、反复测试。建立或者修订数据库之后，必须用用户新输入的数据测试数据字段。最重要的是，让用户进行测试并且同用户一道保证选择的数据类型满足商业要求。测试需要在把新数据库投入实际服务之前完成。

)检查设计。在开发期间检查数据库设计的常用技术是通过其所支持的应用程序原型检查数据库。换句话说，针对每一种最终表达数据的原型应用，保证你检查了数据模型并且查看如何取出数据。