**命名规范**

1. **数据库涉及字符规范**

采用26个英文小写字母(不分大小写)和0-9这十个自然数，加上下划线'\_'组成，共37个字符。不能出现其他字符(注释除外)。

注意事项:

1) 以上命名都不得超过30个字符的系统限制。变量名的长度限制为29(不包括标识字符@)。

2) 数据对象、变量的命名都采用英文字符，禁止使用中文命名。绝对不要在对象名的字符之间留空格。

3) 小心保留词，要保证你的字段名没有和保留词、数据库系统或者常用访问方法冲突。

5) 保持字段名和类型的一致性，在命名字段并为其指定数据类型的时候一定要保证一致性。假如数据类型在一个表里是整数，那在另一个表里可就别变成字符型了。

1. **数据库命名规范**

数据库名使用小写英文以及下划线组成。比如:my\_db，snepr

备份数据库名使用正式库名加上备份时间组成，如:dbname\_20070403

1. **数据库表命名规范**

数据表名使用小写英文以及下划线组成，统一表前缀（多个单词的表名，主要关键单词放最后）

比如:info\_user，info\_destination

备份数据表名使用正式表名加上备份时间组成，如:

info\_user\_20070403

info \_destination\_20070403

关联表命名，各关联表的（表名的前面单词的首字母+最后单词的前3个字母）以下划线连接。

例如：表li\_location\_mark 与表li\_incident的关联表命名为：li\_locm\_inc。

1. **字段命名规范**

字段名称使用单词组合完成，带表名前缀。字段的表名前缀（取表名各单词的首字母）

如web\_user 表的字段:

wu\_id、wu\_name

表li\_locm\_inc的字段

li\_id

表与表之间的相关联字段要用统一名称，

如li\_user表里面的u\_id和li\_group表里面的u\_id相对应；

1. **外键命名规范**

外键名称为FK\_表名A\_表名B\_关联字段名；

其中表名和关联字段名如果过长，可以取表名、关联字段名的前5 个字母。

如果表名、关联字段为多个单词组合，可以取前一个单词，外加后续其它单词的

首字母作为字段名。

如：FK\_user\_token\_user\_phnum；

1. **字段类型规范**

规则:用尽量少的存储空间来存数一个字段的数据。

比如能用int 的就不用char 或者varchar

能用varchar(20)的就不用varchar(255)

时间戳字段尽量用int 型，如created:表示从'1970-01-01 08:00:00'开始

的int 秒数，采用英文单词的过去式；gmtCreated:表示datetime 类型的时间，

即形如'1980-01-01 00:00:00'的时间串，Java 中对应的类型为Timestamp

1. **索引使用原则：**

1) 逻辑主键使用唯一的成组索引，对系统键(作为存储过程)采用唯一的非成

组索引，对任何外键列采用非成组索引。考虑数据库的空间有多大，表如何进行访

问，还有这些访问是否主要用作读写。

2) 大多数数据库都索引自动创建的主键字段，但是可别忘了索引外键，它们

也是经常使用的键，比如运行查询显示主表和所有关联表的某条记录就用得上。

3) 不要索引blob/text 等字段，不要索引大型字段(有很多字符)，这样作会让

索引占用太多的存储空间。

4) 不要索引常用的小型表

不要为小型数据表设置任何键，假如它们经常有插入和删除操作就更别这样

作了。对这些插入和删除操作的索引维护可能比扫描表空间消耗更多的时间。

1. **sql 语句规范**

所有sql 关键词全部大写，比如SELECT，UPDATE，FROM，ORDER，BY 等，表名

与字段名不需要大写

如:

SELECT COUNT(\*) FROM cdb\_members WHERE userName= 'aeolus';

1. **其他设计技巧**

1) 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。

2) 避免使用存储过程

3) 使用常用英语(或者其他任何语言)而不要使用拼音首字母缩写\_\_

3.表设计规范：

(1)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。

[**数据库**](http://lib.csdn.net/base/mysql)设计表及字段命名规范(我整理的，望大家多多提建议)

1.数据库表命名规范:

(1)表名前应该加上前缀，表的前缀一个用系统或模块的英文名称缩写，前缀全部大写或首字母大写，表名中包含的单词首字母大写。

(2)数据库表名应该有意义，并且易于理解，最好使用可以表达功能的英文单词或缩写，如果用英文单词表示，建议使用完整的英文单词。

(3)表名不可以太长，最好不要超过3个英文单词长度（22个字母）。

(4)在数据库表命名时应该用英文单词的单数形式，如员工表命名：应该为Employee而不是Employees .

(5)如果是后台表命名时应该在表名基础上加上后缀  
\_b  
(back 首字母 )

(6)在表创建完成前，应该为表添加表的注释。

2.表字段命名规范：

(1)数据库表字段应该是有意义而且易于理解的，最好是能够表达字段含义的英文字母

（有人认为如果用英文单词作为字段，因为翻译工具不同，而字段不统一建议使用汉语拼音首字母缩写  
；有人认为用汉语拼音缩写看起来不直观，老半天也不知道到底这个字段是干什么的）

(2)系统中所有属于内码，即仅用于标识唯一性和程序内部用到的标识性字段，字段名称建议取为 ID ，采用类型为整型或长整型.

(3)系统中属于是业务内的编号字段，代表一定业务信息，建议字段命名为  
code , 如工作单编号  
wf\_code .

(4) 不要在数据库表字段（列名）中包含数据类型，如：datetime

(5)不要在数据库表字段（列名）命名时重复表名，可以使用表名首字母（不包含数据库表名前缀）

注意：  
不要在数据库表字段（列名）命名时  
不建议使用数据库关键字,如：name,time ,datetime password 等

3.表设计规范：

(1)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。

(2)当字段定义为字符串形时建议使用varchar而不用nvarchar。

注  
：在MySQL5.0以上的版本中，varchar数据类型的长度支持到了65535，也就是说可以存放65532个字节的数据，起始位和结束位占去了3个  
字节。

(3)建议在大多数表中（如工作单），应都有以下字段：

字段名说明类型默认值

CreatorID创建者int  
默认值为  
0

CreatedTime创建时间Datetime  
默认值为NULL

(4)字段的描述

a.字段必须填写描述信息（注释）

b.尽量遵守第三范式的标准（3NF）

表内的每一个值只能被表达一次（列名不重复）

表内的每一行都应当被唯一的标示（标识唯一性，如  
自动增长  
主键）

表内不应该存储依赖于其他键的非键信息

(5)加索引规则

a.表建好后数据库自动为表生成一个索引（为  
自动增长的列生成唯一索引），如果在对这列添加索引，数据库会给一个警告,内容大概是，已经为这列添加了索引，建议修改索引名称和自动增长列名保持一致，为了方便使用。

b.如果在添加索引时，建议索引名称和数据库列名保持一致，为了方便使用

c.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。

d.如果字段与其它表的字段相关联，需建索引。

e.如果字段需做模糊查询之外的条件查询，需建索引。

f.除了主关键字允许建立簇索引外，其它字段所建索引必须为非簇索引。

4.存储过程命名规范

(1)  
存贮过程的命名请遵循以下命名规范：USP\_ + 系统模块缩写（与表前缀类似）+\_ + 功能标识 + 代表存贮过程操作的主要表名（不带前缀）或功能的英文单词或英文单词缩写。

如果一个存贮过程只对一个表进行操作，建议存贮过程的名称就用存贮过程所操作的表的表名（不带前缀）。这样有利于根据表名找到相应的存贮过程。例如：

用于新增的存贮过程USP\_MESSAGE\_Add\_Model

用于修改的存贮过程USP\_ MESSAGE\_Upt\_Model

用于删除的存贮过程USP\_ MESSAGE\_Del\_ Modele

注：USP是user stored procedure  
缩写

5.存储过程设计规范

在存贮过程中必须说明以下内容：

(1)目的：说明此存贮过程的作用。

(2)作者：首次创建此存贮过程的人的姓名。在此请使用中文全名，不允许使用英文简称。

(3)创建日期：创建存贮过程时的日期。

(4)修改记录：

修改记录需包含修改顺序号、修改者、修改日期、修改原因，修改时不能直接在原来的代码上修改，也不能删除原来的代码，只能先将原来的代码注释掉，再重新增加正确的代码。修改顺序号的形式为：log1，log2，log3。。。，根据修改次数顺序增加，同时在注释掉的原来的代码块和新增的正确代码块前后注明修改顺序号。

(5)对存贮过程各参数及变量的中文注解。

建议：在数据库中创建一个文本文件保存创建脚本

6.视图命名规范

视图的命名请遵循以下命名规范：UV \_ + 系统模块缩写（与表前缀类似）+\_ + 功能标识 + 代表视图查询的主要表名（不带前缀）或功能的英文单词或英文单词缩写。

如果一个视图只对一个表进行查询，建议视图的名称就用视图所查询的表的表名（不带前缀）。这样有利于根据表名找到相应的视图。

注：UV是userView缩写

7.视图设计规范

在视图中必须说明以下内容：

(1)目的：说明此视图的作用。

(2)创建者：首次创建此视图的人的姓名。在此请使用中文全名，不允许使用英文简称。

(3)修改者、修改日期、修改原因：如果有人对此视图进行了修改，则必须在此视图的前面加注修改者姓名、修改日期及修改原因。

(4)对视图各参数及变量的中文注解

建议：在数据库中创建一个文本文件保存创建脚本

8.触发器命名规范

Insert触发器加'\_i'，Delete触发器加'\_d'，Update触发器加'\_u'

9.触发器设计规范

在视图中必须说明以下内容：

(1)目的：说明此触发器的作用。

(2)创建者：首次创建此触发器的人的姓名。在此请使用中文全名，不允许使用英文简称。

(3)修改者、修改日期、修改原因：如果有人对此触发器进行了修改，则必须在此触发器的前面加注修改者姓名、修改日期及修改原因。

(4)对触发器各参数及变量的中文注解

建议：在数据库中创建一个文本文件保存创建脚本