

SQL2008

数据库命名规范

2012. 1. 7

1. 数据库涉及字符规范

采用 26 个英文字母和 0—9 这十个自然数,加上下划线_组成,共 36 个字符。不出现其他字符。鉴于大小写规则难记忆,而且有些情况下,数据库不区分大小写,统一采用小写字母。

采用英文单词或英文短语(包括缩写)作为名称,不使用无意义的字符或汉语拼音。

名称应该清晰明了,能够准确表达事物的含义,最好可读,遵循“见名知意”的原则。

2. 数据库对象命名规范

数据库对象包括表、视图(查询)、存储过程(参数查询)、函数、约束。对象名字由前缀和实际名字组成,绝对不要在对象名的字符之间留空格,长度不超过 30。

使用单数(比如表名 tb_member, 不要使用 tb_members)。

前缀: 使用小写字母

表	tb_<表的内容分类>_<表的内容>
视图	vi
存储过程	sp
函数	fn
索引	idx_<表名>_<索引标识>
主键	pk_<表名>_<主键标识>
外键	fk_<表名>_<主表名>_<外键标识>
序列	seq

实际名字: 实际名字尽量描述实体的内容,由单词或单词组合,每个单词以

‘_’ 间隔, 字母小写, 不以数字开头。合法的对象名字类似如下。

表	tb_user_info	tb_message_detail
视图	vi_message_list	
存储过程	sp_message_add	

3. 数据库表设计规范

表名由前缀和实际名字组成。tb_<表的内容分类>_<表的内容>。

表的设计要符合 3NF, 然后, 可以根据效率的需要, 适当做一些冗余。

前缀: 使用小写字母 tb, 代表表。实际名字中, 一个系统尽量采取同一单词, 多个后面加_来连接区分。

合法的表名类似如下。

```
tb_member
tb_member_info
tb_forum_board
tb_blog_comment1
```

一些作为多对多连接的表, 可以使用两个表的前缀作为表名:

如: 用户表 tb_user, 用户分组表 tb_group_info, 这两个表建立多对多关系的表名为: tb_user_group_relation

4. 字段命名规范

字段由表的简称, 实际名字组组成。如果此字段关联另外的字段, 那么加下划线_连接关联表字段的字段名。布尔型的字段, 以一些助动词开头, 更加直接生动: 如, 用户是否有留言 has_message, 用户是否通过检查 is_checked 等。

字段名为英文短语、形容词+名词或助动词+动词时态的形式表示, 遵循“见名

知意”的原则。

因此，合法的字段名类似如下。

```
user_id
user_name
user_reg_date
```

5. 视图命名规范

由前缀和实际名字组成，中间用下划线连接。前缀：使用小写字母 vi，表示视图。合法的视图名类似如下。

```
vi_user
vi_user_info
```

6. 存储过程命名规范

字段由前缀和实际名字加操作名字组成，中间用下划线连接。

前缀：使用小写字母 sp，表示存储过程。

操作名字：insert|delete|update|caculate|confirm

例如：sp_user_insert（前缀+表名+操作名字）

```
/*-----
//
// 描述：*****
// 创建人：***
// 创建日期：2011.1.7
// 修改标识：
// 修改描述：
//-----*/
```

7. 数据库设计文档规范

所有数据库设计要写成文档,附有表关系图,可采用 PowerDesigner 设计; 表的具体描述文档以模块化形式表达。大致格式如下:

表名: tb_members

日期: 2008-12-17

版本: 1.0

描述: 保存用户资料

主键: mem_id

索引: idx_mem_name, unique(唯一索引)

外键: mem_id (对应 tb_member_info.meminfo_id)

具体内容样例

字段名	数据类型	可为空	默认值	备注
meminfo_id	int (4)	否	无	主键, 自增长 (如果是 ORACLE 要指定序列对象名)
name	varchar (50)	否	无	表示该会员名, 会员名不重复
mem_id	int (4)	是	0	外键
address	varchar (100)	是	无	表示所在地址
phone	varchar (50)	是	空串	表示联系电话
QQ	varchar (50)	否	无	表示会员 QQ

8. sql 语句规范

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

9. 数据完整性规范

1、为便于在程序的编码期查错, 可以在设计数据库的时候尽可能多的加上约束 (check)。如, 整型的字段的取值范围等, 常常为 field>0。

- 2、同理，尽可能地在开发期间使用触发器来验证数据的完整性。
- 3、如果字段之间存在冗余，应该编写触发器来管理冗余的字段
- 3、在开发阶段保存完整的主键、外键和唯一索引的约束。
- 4、原则：编码期间，数据完整性优先于性能。在保障系统正确运行的前提下尽可能的提高效率。
- 5、数据库分开发期和运行期两个版本：开发期版本主要体现完整的约束，运行期版本体现效率的优化。

10. 数据库性能优化规范（运行期）

- 1、在运行阶段删除不必要的约束(check)。
- 2、尽量不要使用触发器
- 3、尽量保留主键约束
- 4、适当删除外键，以提高性能
- 5、在运行期间，通过分析系统的访问量，创建索引来优化性能
- 6、分析每个表可能的数据增长量，定义自动拆分表规则。将大表进行拆分来提高性能。
- 7、预先考虑数据清理规则：在什么情况下删除数据库中的旧数据，以此来提高性能。
- 8、制定数据库备份和灾难恢复计划。
- 9、为效率考虑，可以在系统测试阶段适当增加冗余字段，或者冗余表。
- 10、分页的记录输出建议通过存储过程来实现，不要使用 API 游标来分页，这样可以提高分页的效率。