**MySQL数据库开发规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 版本 | 说明 | 创建人 |
| 2019.03.25 | V1.0 | 初始版本 | DBA |

## 一、目标

为了最大限度保证合生通MySQK数据库设计编码规划化，避免后期不必要的由此引发的优化调整。制定本数据库开发设计规范。

## 二、适用对象及范围

适用的人员包括：使用MySQL数据库的研发人员

适用数据库版本：MySQL5.6及以后

## 三、命名原则

适用于数据库名、对象名如表、字段、索引、存储过程，函数等的命名约定。

|  |  |
| --- | --- |
| 序号 | 描述 |
| 1 | 命名使用富有意义的英文词汇，尽量避免使用缩写，多个单词组成的，中间以下划线分割； |
| 2 | 避免使用MySQL的保留字如LEVEL、关键字如TYPE（见MySQL保留字和关键字） |
| 3 | 各表之间相关列名尽量同名 |
| 4 | 除数据库名称长度为1－8个字符，其余为1－30个字符，Database link名称也不要超过30个字符 |
| 5 | 命名只能使用英文字母，数字和下划线 |

**3.1表名**

规则如下：

命名规则为:xxx\_yyy\_TableName。xxx表示子系统的名称，由三个字母构成；yyy表示子系统中的子模块的名称（可以没有）；TableName为表含义

TableName规则如下：

* 使用英文单词或词组作为表名，不得使用汉语拼音
* 用名词和名词短语作表名
* 不使用复数

正确的命名，例如：sys\_user，biz\_order

**3.2 存储过程**

规则如下：

命名规则为：xxx\_yyy\_StoredProcedureName。xxx表示子系统的名称，由三个字母构成；yyy表示子系统中的子模块的名称（可以没有）；StoredProcedureName为存储过程含义

StoredProcedureName规则如下：

⑴ 用动词或动词短语来命名，并带有宾语

⑵ 需要符合用Pascal 命名规则。

⑶ 尽量谨慎地使用缩写

⑷ 尽量不要和关键字重合

⑸ 不要用任何名前缀 (例如 U，B)

⑹ StoredProcedureName内不使用下划线

⑺ 当操作依赖条件时，一般结尾使用 By+条件

存储过程正确的命名，例如：

sys\_InsertUser

sys\_SearchUserByUserID

sys\_DeleteUserByUserID

**3.3 视图**

规则如下：

视图的命名采用xxx\_yyy\_*ViewName\_v*。xxx表示子系统的名称，由三个字母构成；yyy表示子系统中的子模块的名称（可以没有）；\_v后缀表示视图，ViewName部分表示视图的含义。

ViewName规则如下：

○ 用名词和名词短语，

○ 不使用复数

○ 用Pascal 命名规则

○ 尽量谨慎地使用缩写

○ 尽量不要和关键字重合

○ 不要用任何名前缀 (例如 U，B)

○ ViewName中使用下划线

视图正确的命名，例如：

sys\_UserView\_v

biz\_UserOrderView\_v

**3.4 索引**

规则如下：

IX\_table\_name\_column\_name

多单词组成的column name，取前几个单词首字母，加末单词组成column\_name。

如：

tst\_sample表member\_id上的索引: IX\_tst\_sample\_mid

sys\_news表title上的索引: IX\_sys\_news\_titile

**3.5修改**

修改记录时，必须有Where唯一条件

**3.6 主键**

主键命名：PK\_table\_name

如：

tst\_sample表的主键: PK\_tst\_sample

**3.7外键**

外键命名：FK\_table\_name\_column\_name

多单词组成的column name，取前几个单词首字母，加末单词组成column\_name。

如：

tst\_sample表user\_id字段的外键: FK\_tst\_sample\_uid

tst\_sample表type\_id字段的外键: FK\_tst\_sample\_tid

## 四、设计规范

**4.1规范约定**

1.尽可能遵守数据的设计规范3NF 规定，可适当存在冗余数据

* 表内的每一个值都只能被表达一次。
* 表内的每一行都应该被唯一的标识（有唯一键）。
* 表内不应该存储依赖于其他键的非键信息。

2.实体表尽可能包含以下字段

* 编号（ID）、代码（Code）
* 创建人（Created By）、创建时间（Creation Time）
* 修改人（Modified By）、修改时间（Modification Time）
* 删除标志（Delete Status）

其含义如下：

* 编号是系统内的唯一标识，一般为自增列
* 代码是客户为这个实体的编码，根据需要确定是否保留
* 创建人是是指数据库记录创建人，根据需要确定是否保留
* 创建时间是指数据库记录创建时间，默认为SYSDATE
* 修改人是是指数据库记录修改人，根据需要确定是否保留
* 修改时间是指数据库记录修改时间，默认为SYSDATE
* 删除标志是本条记录是否已删除的标志（0：没有删除；1：已经删除），数据中

的数据不做真正删除，只设置这个标志

这五个字段的命名规则如下

**实体名＋字段名称**，如user\_id、user\_code、created\_by、creation\_time、modified\_by 、modification\_time 、delete\_status。

这五个字段的类型和长度规则如下：

|  |  |  |
| --- | --- | --- |
| 字段名称 | 字段类型 | 描述 |
| user\_id | INTEGER | 编号 |
| user\_code | VARCHAR2（30） | 代码 |
| created\_by | INTEGER | 创建人 |
| creation\_time | DATE | 创建时间 |
| modified\_by | INTEGER | 修改人 |
| modification\_time | DATE | 修改时间 |
| delete\_status | CHAR(1) | 删除标志 |

**4.2字段规范**

* 一行记录必须表内唯一，表必须有主键。
* 枚举类型使用 NUMBER，且需要说明枚举类型的各个不同取值的含义
* ID结尾的字段为仅由数字组成，Code结尾的字段为仅由字母或数字组成
* 用CHAR（1）表示布尔值的取大写：“Y”，“N”。
* 应尽量使用VARCHAR2代替CHAR类型；
* VARCHAR(2)最多4000字符；
* DATE精确到秒，而非天；
* 使用CLOB代替LONG，BLOB代替LONG RAW；
* 使用NUMBER数据类型时请给定长度，例如:NUMBER(5,2)　表示整数部分最大3位，小数部分为2位；

## 五、使用规范

**5.1 综合**

* 如果开发过程中需要建立索引，需要提交书面的更改请求，说明所需索引的定义(名称、字段列表、顺序、索引类型)以及建立的理由。数据库管理员统一维护索引并将提交的请求更改。
* 数据库各表的初始数据(包含各代码表、配置表)需要提交给数据库管理员。
* 不得使用触发器。
* 涉及到数据库多表数据的更改(Insert/Delete/Update)必须使用数据库事务进行控制，并且必须有完整事务开始和提交/回滚机制。不能使用范围事务
* 尽量避免Union操作的使用，需要使用时，请向数据库管理员咨询使用Union操作的影响。
* SQL语句和存储过程，尽量使用PL/SQL Develper的代码美化器美化。
* 属于ORACLE的关键字大小，表名、列名等小写。

**5.2 查询**

* 在表查询中，一律不要使用\* 作为查询的字段列表，需要哪些字段必须显式写明
* 在表查询中，必须有Where条件，除非此表为非增长表
* 在表查询中，一次最多返回的记录条数不要超过1000条或记录内容不要大于1MB的数据。
* 在表查询中，作Order By排序时，优先使用主键列，索引列
* 多表关联查询时，优先使用Where条件，再作表关联，并且需要保证被关联的字段需要有索引。
* 避免在WHERE字句中对列施以函数:
* 错误：  
  SELECT　service\_id,service\_name　  
  FROM　service\_promotion  
  WHERE TO\_CHAR(gmt\_modified,’yyyy-mm-dd’)   
  = ‘20001-09-01’;  
  正确：  
  SELECT service\_id,service\_name　   
  FROM service\_promotion  
  WHERE gmt\_modified   
  >= TO\_DATE(‘2001-9-01’,’yyyy-mm-dd’)  
  AND gmt\_modified   
  < TO\_DATE(‘2001-9-02’,’yyyy-mm-dd’);
* 避免使用数据库的类型自动转换功能：

错误：  
SELECT category\_id, category\_name FROM category  
WHERE category\_id = ‘123’; -- id’s type is number

正确：  
SELECT category\_id, category\_name FROM category  
WHERE category\_id = 123; -- id’s type is number

**5.3 删除**

* 删除记录时，必须有Where唯一条件
* 当有主从表时，要先删除从表记录，在删除主表记录