**好生活MySQL数据库开发规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 版本 | 说明 | 创建人 |
| 2019.03.25 | V2.0 | 初始版本 | DBA |

## 一、目标

为了最大限度保证合生通MySQL数据库设计编码规划化，使开发可以写出高质量查询语句。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、命名规范

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。

* 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。
* 避免使用MySQL的保留字如LEVEL、关键字如TYPE。
* 各表之间相关列名尽量同名。
* 命名只能使用英文字母，数字和下划线。

**3.1 表名**

3.1.1规则

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**

3.2.1规则

命名规则为：proc\_xxx\_yyy\_StoredProcedureName。Proc表示存储过程前缀，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

3.2.2 StoredProcedureName规则如下：

* 用动词或动词短语来命名，并带有宾语
* 需要符合用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)
* StoredProcedureName内不使用下划线
* 当操作依赖条件时，一般结尾使用 By+条件

3.2.3 存储过程正确的命名

例如：

proc\_sys\_InsertUser

proc\_sys\_SearchUserByUserID

proc\_sys\_DeleteUserByUserID

**3.3 视图**

规则如下：

视图的命名采用v\_xxx\_yyy\_*ViewName\_v*。v\_表示视图前缀，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

○ 尽量谨慎地使用缩写

○ 尽量不要和关键字重合

○ ViewName中使用下划线

视图正确的命名，例如：

v\_sys\_UserView

v\_biz\_UserOrderView

**3.4 索引**

规则如下：

IDX\_table\_name\_N1 表示单列普通索引，多个索引序号递增

IDX\_table\_name\_U1 表示单列或多列唯一索引，多个索引序号递增

IDX\_table\_name\_C1 表示复合索引，多个索引序号递增

如：

tst\_sample表member\_id上的唯一索引: IDX\_tst\_sample\_u1

sys\_news表title上的普通索引: IDX\_sys\_news\_n1

**3.5 主键**

主键命名：PK\_table\_name

示例：tst\_sample表的主键: PK\_tst\_sample

**3.6 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、类型规则

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。

* 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。
* 避免使用MySQL的保留字如LEVEL、关键字如TYPE。
* 各表之间相关列名尽量同名。
* 命名只能使用英文字母，数字和下划线。

**4.1 整型**

规则：统一使用bigint数据类型

示例：id bigint not null

**4.2 浮点型**

规则：浮点类型数据统一使用decimal数据类型

示例：je decimal(20,2) ,表示整数为18位，小数2位，共20位。

**4.3 日期型**

规则：日期类型统一使用datetime数据类型

示例：created datetime not null default now()

禁用timestamp原因如下：

* 目前项目不存在跨时区问题，如果有此需求，则可以考虑使用该类型，同时使用该类型比datetime占用更少的空间。
* Timestamp类型不支持按日期分区(5.6版本测试),不方便日后表分区数据归档处理

**4.4 字符型**

规则:统一使用varchar(n)，n表示字符，大于4000字符使用text类型，大于8000字符使用longtext类型。

示例：name varchar(20)，content text,html longtext。

## 五、设计规范

**5.1 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now()
* 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now()
* 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 | bigint | 编号 |
| user\_code | VARCHAR2（30） | 代码 |
| created\_by | varchar(20) | 创建人 |
| creation\_time | datetime | 创建时间 |
| modified\_by | varchar(20) | 修改人 |
| modification\_time | datetime | 修改时间 |
| delete\_status | Varchar(1) | 删除标志 |

**5.2 字段规范**

* 一行记录必须表内唯一，表必须有主键，自增主键类型为:bigint。
* 每个字段数据类型选择满足业务最小类型
* 枚举类型使用 INT，且需要说明枚举类型的各个不同取值的含义
* 禁止使用CHAR类型，应使用VARCHAR类型，长度根据实际业务数据设置，不允许设置过大。
* VARCHAR(n)单列存储数据超过1000个字符，需要考虑拆表（水平或垂直），将常用列放在一张表中，不常用列及大字符列放在另一张表中；或者将大字符列以行的形式存储在另一张表中）
* 使用DECIMAL数据类型时请给定长度，例如: DECIMAL (5,2)　表示整数部分最大3位，小数部分为2位；
* 表中日期字符统一使用datetime类型，（禁用使用timestamp该类型不支持按日期进行表分区），初始化使用：created datetime not null default now()

## 五、使用规范

**5.1 综合**

* 不得使用触发器。
* 数据库更改(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。
* 日期类型必须使用date,datetime类型

**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，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字句中对列施以表达式或函数:
* 查询条件中禁止使用NOT IN子句可用NOT EXIST代替
* 大结果集标用UNION 操作
* 子查询中避免使用复杂查询（子查询中含有distinct ,group by，自定义函数）。

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’);

实际线上慢查询例子说明

* 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、修改**

* 删除或修改记录时，尽量通过主键来进行更新
* 当有主从表时，要先删除从表记录，在删除主表记录
* 批量大量数据时考虑如下方式

如果一次删除表中数据量较大，采用循环方式进行删除，通过主键一次删除一行，删除一批后再进行提交。

如果删除数据量相对较少，确保删除的列上有索引。