# MYSQL数据库命名及设计规范

## 1 目的

规范思源软件研发部MySQL数据库的设计，保持各系统、各模块的统一风格。

## 2 适用范围

适用于思源软件所有使用MySQL作为DBMS的数据库。

## 3 内容

### 3.1 标准化和规范化

数据库设计尽量遵循第三范式（3NF）的要求。即某个表只包括其本身基本的属性，当不是它们本身所具有的属性时需进行分解。表之间的关系通过外键相连接。它具有以下特点：有一组表专门存放通过键连接起来的关联数据。

对于因效率等原因，无法遵循第三范式的，应该在设计文档中明示原因，并说明如何保持数据的一致性。

### 3.2 数据库字符集规范

统一使用UTF8字符集。

### 3.3 数据库的字符规范

采用26个英文字母(区分大小写)和0-9这十个自然数,加上下划线'\_'组成,共63个字符.不能出现其他字符(注释除外)。

### 3.4 数据库的命名规范

注意事项:

使用常用英语而不要使用拼音首字母缩写；

数据库命名都不得超过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@)；

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；

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

例如：

access add all alter and any as asc audit between by char check cluster column comment compress connect create current date decimal default delete desc distinct drop else exclusive exists file float for from grant group having identified immediate in increment index initial insert integer intersect into is level like lock long maxextents minus mlslabel mode modify noaudit nocompress not nowait null number of offline on online option or order p ctfree prior privileges public raw rename resource revoke row rowid rownum rows select session set share size smallint start successful synonym sysdate table then to trigger uid union unique update user validate values varchar varchar2 view whenever where with ……

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。

#### 3.4.1 数据库名

全部使用小写字母，

ETS数据库以ets为前缀，后面为"\_"模块，如ets\_platform, ets\_budget，使用各模块的项目名称。如果出现超过30字符的情况，进行合理意义的裁剪。

#### 3.4.2 表名

数据库表的命名采用如下规则：

表名长度不能超过30个字符，表名中含有单词全部采用单数形式，单词要小写；

多个单词间用下划线’\_’进行连接；

表中含有的单词建议用完整的单词。如果导致表名长度超过30个字符，则从最后一个单词开始，依次向前采用该单词的缩写。（如果没有约定的缩写，则采用该单词前4个字母来表示）

如果不同的表属于不同业务模块，建议在表名前增加模块缩写。示例：

客户表：t\_cust\_customer

客户联系方式：ct\_cust\_ contect

分表命名规则：原表明\_数字，示例：t\_cust\_customer \_01

关联表命名规则：表a\_表b，如果存在模块缩写，根据实际的业务需要保留一个模块缩写

示例：

房客关系表： t\_prt\_property\_customer

#### 3.4.3 字段名

数据库表的字段命名采用如下规则：

字段名称使用单词组合完成，全部使用小写名称，各单词之间使用下划线”\_”连接；

特别的，实体ID字段名以去除前后缀及模块缩写之后的数据库名+”\_id”命名，例如：客户编码ID：customer\_id

表与表之间的相关联字段要用统一名称；

如果该字段表示的是时间，则使用\_time为后缀。一般用于系统生成的时间字段；

如果该字段表示的是不含时间的日期，则使用\_date为后缀。一般用于业务录入的日期字段；

相关业务表要增加创建时间和最后修改时间

示例：

创建时间：create\_time

最后修改时间：update\_time

#### 3.4.4 主键名

主键名为前缀pk\_+表名。

#### 3.4.5 外键名

外键名为fk\_<主表前12个字符>\_r\_<被引用表前12个字符>，如果一个表对另一个表多次引用，使用序号标识。

#### 3.4.6 索引名

索引名的基本规则为ix\_<表名>\_<核心字段名>，超长时进行合适的截取。

### 3.5 字段类型的规范

字段类型尽量使用如下类型，如果用到特殊类型需要在概要设计文档中特别说明：

int

bigint

varchar

datetime

timestamp

### 3.6 存储过程

存储过程的命名采用如下规则：

存储过程名用p\_开头，存储过程名长度不能超过30个字符。存储过程名用小写的英文单词来表示。多个单词间用下划线’\_’进行连接

### 3.7 函数

函数的命名采用如下规则：

函数名用f\_开头，函数名长度不能超过30个字符。函数名用小写的英文单词来表示。多个单词间用下划线’\_’进行连接

### 3.8 视图

数据库视图的命名采用如下规则：

视图名用v\_开头，视图名长度不能超过30个字符。视图名用小写的英文单词来表示；

视图其他部分参照表的命名要求。

### 3.9 其他事项

系统尽量避免触发器的使用！

# MySql开发设计规范

数据库设计尽量遵循第三范式（3NF）的要求。即某个表只包括其本身基本的属性，当不是它们本身所具有的属性时需进行分解。表之间的关系通过外键相连接。它具有以下特点：有一组表专门存放通过键连接起来的关联数据。对于因效率等原因，无法遵循第三范式的，应该在设计文档中明示原因，并说明如何保持数据的一致性。

## 命名规范

1. 库名、表名、字段名采用26个英文字母和0-9这十个数字,加上下划线'\_'组成,不能出现其他字符(注释除外).

2. 库名、表名、字段名禁止超过30个字符。

3. 库名、表名、字段名禁止使用MySQL保留字

4. 数据对象、变量的命名都采用英文字符,禁止使用中文命名.

5. 临时库、表名必须以tmp为前缀，并以日期为后缀. 如tmp\_tablea\_20151130

6. 备份库、表必须以bak为前缀，并以日期为后缀。如bak\_Databasename\_20151130

保留字：access add all alter and any as asc audit between by char check cluster column comment compress connect create current date decimal default delete desc distinct drop else exclusive exists file float for from grant group having identified immediate in increment index initial insert integer intersect into is level like lock long maxextents minus mlslabel mode modify noaudit nocompress not nowait null number of offline on online option or order p ctfree prior privileges public raw rename resource revoke row rowid rownum rows select session set share size smallint start successful synonym sysdate table then to trigger uid union unique update user validate values varchar varchar2 view whenever where with。。。

## 基础规范

1. MySQL5.6版本建议使用INNODB存储引擎

2. 表字符集使用UTF8,需要支持emoji 表情的列设置为utf8mb4

3. 所有表和列原则上都需要添加注释

4. 单表数据量建议控制在5000W以内,或不超过2G

5. 不在数据库中存储图片、文件.

6. 禁止在线上做数据库压力测试,如有特殊需要，需提前报备。

7. 禁止客户端直接操作测试，生产数据库

## 库表设计

1. 禁止使用分区表

2. 拆分大字段和访问频率低的字段，分离冷热数据

3. 用HASH进行散表，表名后缀使用十进制数，下标从0开始

4. 按日期时间分表需符合YYYY[MM][DD][HH]格式

5. 采用合适的分库分表策略。例如千库十表、十库百表等

## 字段设计规范

1. 尽可减少使用TEXT、BLOB类型。如必要需将TEXT,BLOB字段拆分后单独存储。

2. 不建议使用FLOAT和DOUBLE，建议用DECIMAL或bigint替代。

3. 不建议使用ENUM类型，考虑TINYINT代替。

4. 原则上禁止DEFAULT NULL，建议NOT NULL 设置默认值。

5. 如果不存储负数，建议使用UNSIGNED存储非负整数

6. 使用bigint存储时间

7. 使用INT UNSIGNED存储IPV4

8. InnoDB引擎优先考虑使用VARBINARY存储大小写敏感的变长字符串

9. 禁止在数据库中存储明文密码

## 索引规范

1. 单张表中索引数量不超过5个

2. 单个复合索引中的字段数不超过5个

3. 对字符串使用前缀索引，前缀索引长度不超过8个字符。

4. 建议优先考虑前缀索引，必要时可添加伪列并建立索引。

5. InnoDB表必须有主键，即使主键是无意义的。

6. 不使用更新频繁的列做为索引。

7. 频繁执行的重要SQL必须被索引

8. UPDATE、DELETE语句的WHERE条件列，ORDER BY、GROUP BY、DISTINCT的字段，多表JOIN的字段 建议设置索引

9. 选择度最大的字段放在前面

10. 核心SQL优先考虑覆盖索引

11. 避免冗余和重复索引

## 索引禁忌

1. 不在选择性低的列上建立索引，例如“性别”, “状态”， “类型”

2. 不在索引列进行数学运算和函数运算

3. 尽量不使用外键

4. 不使用前导查询，如like “%ab”,like “%ab%”

## SQL语句

1. 要求所有的SQL提交测试前，使用EXPLAIN 查看执行计划 ，避免全表扫描，避免生成临时表

2. 使用预编译语句. 高效，相对安全。

3. 禁止不同类型字段做比较，避免隐式转换

4. 尽量不使用存储过程、触发器、UDF、events等

5. 避免使用大表的JOIN

6. 避免在数据库中进行数学运算，尽量在程序端做处理，传给数据库。

7. 尽量将大SQL拆分成小SQL

8. 禁止单条SQL语句同时更新多个表

9. 禁用select \*

10. insert语句必须显示指定字段名

11. 使用in代替or，in的值不超过1000个

12. 禁止使用order by rand()

13. 用union all而不是union

14. 程序应有捕获SQL异常的处理机制

15. 减少与数据库的交互次数

16. 合理的使用分页

## 操作规范

1. 批量导入、导出数据必须提前通知DBA协助观察.

2. 禁止在线上从库执行后台管理和统计类查询。尽量放在备份库操作.

3. 最小化权限原则。只分配应用必要的权限。Delete权限需按表分配.

4. 尽量不在MySQL数据库中存放业务逻辑.

5. 重大项目的数据库方案选型和设计提前通知DBA参与.

6. 大型推广活动或上线新功能须提前通知DBA进行评估.

7. 对特别重要的库表，提前与DBA沟通确定维护和备份优先级.

8. 避开在业务高峰期批量更新、查询数据库.

9. 线上数据变更，须详细注明所有相关SQL语句.

## 注意事项：

1. 需要建库的应用，需要提供应用的IP,以便对数据库用户进行授权。

2. MySQL线上环境涉及到数据库的DDL（Create,Alter,Drop,DECLARE）操作，请抄送给DBA。