**简介**

良好的使用规范，可以帮助我们更好更快的使用数据库，避免一些不好的使用习惯的养成。

**操作类规范**数据库攻击性语句，包括drop、rename、truncate、flush，必须发邮件给项目负责人以及架构组确认（需说明用途）（抄送研发组，运维组），然后再由运维组执行

* 新建数据库或表必须发邮件于项目负责人以及架构组确认（需说明用途）（抄送研发组，运维组），然后再由运维组执行
* 数据库变更操作，包括建表、增加/修改字段、增加/修改索引（需说明用途）（抄送研发组，运维组），然后再由运维组执行
* 申请的操作语句，先在测试库进行测试操作
* 不在业务高峰期执行大批量数据操作（增删改）
* 不允许直接drop table,要先rename,过一周再drop（rename规范：原表名\_del\_20161030 20161030为可执行删除的日期）
* 不允许直接delete数据，如果必须要建立临时表（表名参考上一条），将要移除的数据转移到临时表
* 不允许有不带where条件的select、update（不允许直接delete）
* 任何数据库相关的BUG，及时告知运维组

**设计类规范**

* 表设计
  + 表字符集明确使用UTF8
  + 存储引擎使用INNODB
  + 不在数据库中存储图片、文件，避免使用大文本类型
  + 不使用外键，由程序保证数据一致性
  + 禁止使用存储过程、视图
  + 每张表数据量控制在2000W以下 ，如预计会超出，需提前做好拆分或者归档迁移计划
  + 每张表必须要有无歧义的全面的注释
* 字段设计
  + 每个字段均定义为NOT NULL,且必须要有默认值（除自增id外），不允许默认null(不能被索引)
  + 每个字段必须要有无歧义的全面的注释（除自增id外），如有新的含义增加(如XX\_type)，请上线前告知运维组
  + 字段设计为合适的大小，使用UNSIGNED标记存储非负整数
  + 使用DECIMAL存储精确浮点数（Float、Double这些浮点数均是不精确的）
  + 使用TINYINT来代替ENUM类型
  + TEXT、BLOB等类型的大存储字段，需要进行拆分，或者独立建表
  + 尽可能将可枚举的字符转化为数字存储（单独拆分建表）
  + 建议：使用UNSIGNED INT存储ipv4 地址，而不是CHAR(15)
  + 建议：使用TIMESTAMP存储操作时间
* 命名
  + 库名、表名、字段名、索引名使用小写字母,以下划线\_分割
  + 库名、表名、字段名、索引名不使用保留字 （关键字）
  + 库名、表名、字段名不超过20个字符，需见名知意，避免使用缩写
* 主从
  + 尽量使用应用服务器时间而不是数据库时间，防止主从同步延迟导致的时间不一致。例如 ON UPDATE CURRENT\_TIMESTAMP

**索引规范**

* 单张表索引数量不超过5，单个索引字段数不超过5（如果超过，说明这个表为过度使用的表需要拆分了，或者索引设计不合理）
* 不能有冗余索引（例如A+B索引和A索引）
* 重要的SQL必须被索引。以下为可使用索引的场景：

- SELECT、UPDATE、DELETE语句的WHERE条件列

- ORDER BY、GROUP BY、DISTINCT的字段

- 多表JOIN的字段

* 用不了索引的情况（查询条件里有包含索引的字段的情况下）：

相关索引列进行了数学运算和函数运算

逻辑判断使用的是like ‘通配符+XXX’

与某个无索引的字段使用了逻辑或（or）关联查询

查询索引结果集过大，mysql放弃使用索引

**SQL语句设计规范**

* 避免使用大表join，会降低索引利用率，并可能会创建较大的内存临时表的创建
* 将复杂的sql在代码层面拆分设计，避免复杂难以理解和维护的SQL
* 避免在超过5次的循环体内执行sql，改为在循环外批量执行，减少与数据库的交互次数
* 不使用select \*