

# 数据库规范文档

---

版本V1.0.0

## 1 建表

1.1 表达是与否概念的字段，必须使用is\_xxx的方式命名，数据类型是unsigned tinyint（1表示是，0表示否）。

1.2 表名、字段名必须使用小写字母或数字，禁止出现数字开头，禁止两个下划线中间只出现数字，数据库字段名的修改代价很大，因为无法进行预发布，所以字段名称需要慎重考虑。

1.3 禁用保留字，如desc、range、match、delayed等，请参考MySQL官方保留字。

1.4 唯一索引名为uk字段名，普通索引名则为idx字段名（uk\_即unique key，idx\_即index的简称）。

1.5 小数类型为decimal，禁止使用float和double。

1.6 varchar是可变长字符串，不预先分配存储空间，长度不要超过5000。

1.7 全局唯一id均采用36位字符串，不使用整型int作为唯一标识。

1.8 所有created.id和updated.id均关联admin\_users.uid。

1.9 created.at, updated.at采用GMT时间，类型均为timestamp类型。

1.10 如果修改字段含义或对字段表示的状态追加时，需要及时更新字段注释。

1.11 字段允许适当冗余，以提高性能，但是必须考虑数据同步的情况。

1.12 单表行数超过500万行或者单表容量超过2GB，才推荐进行分库分表。

1.13 合适的字符存储长度，不但节约数据库表空间、节约索引存储，更重要的是提升检索速度。

1.14 使用utf8mb4 进行存储，排序使用utf8mb4.general.ci排序。

## 2 索引

2.1 业务上具有唯一特性的字段，即使是组合字段，也必须建成唯一索引。

2.2 超过三个表禁止join。需要join的字段，数据类型保持绝对一致，多表关联查询时，保证被关联的字段

需要有索引，即使双表join也要注意表索引、SQL性能。

2.3 在varchar字段上建立索引时，必须指定索引长度，没必要对全字段建立索引。

2.4 页面搜索严禁左模糊或者全模糊，如果需要请走搜索引擎来解决。

2.5 利用覆盖索引来进行查询操作，来避免全表操作。

2.6 利用延迟关联或者子查询优化超多分页场景。

2.7 SQL性能优化的目标：至少要达到range 级别，要求是ref级别，如果是consts最好。

2.8 建组合索引的时候，区分度最高的在最左边。

### 3 规约

3.1 不要使用count(列名)或count(常量)来替代count(), *count()*就是SQL92定义的标准统计行数的语法，跟数据库无关，跟NULL和非NULL无关。

3.2 当某一列的值全是NULL时，count(col)的返回结果为0，但sum(col)的返回结果为NULL。

3.3 使用ISNULL()来判断是否为NULL值，NULL与任何值的直接比较都为NULL。

3.4 在代码中写分页查询逻辑时，若count为0应直接返回，避免执行后面的分页语句。

3.5 不得使用外键与级联，一切外键概念必须在应用层解决，外键与级联更新适用于单机低并发，不适合分布式、高并发集群，级联更新是强阻塞，外键影响数据库的插入速度。

3.6 禁止使用存储过程，存储过程难以调试和扩展，更没有移植性。

3.7 数据订正时，删除和修改记录时，要先select，避免出现误删除，确认无误才能执行更新语句。

3.8 in操作能避免则避免，若实在避免不了，需要仔细评估in后边的集合元素数量，控制在1000个之内。

3.9 如果在数据库中查找某记录，尽量用sql精确定位，再取数据，而不是sql返回所有数据，再进行遍历定位。

3.10 TRUNCATE TABLE 比 DELETE 速度快，且使用的系统和事务日志资源少，但TRUNCATE无事务且不触发trigger，有可能造成事故，故不建议在开发代码中使用此语句。

### 4 其他

4.1 未完待续，欢迎补充。

*Author:KSLab*

*Date:2018-08-28*