



链滴

数据库设计规范

作者: [Hefery](#)

原文链接: <https://ld246.com/article/1641230162103>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

数据库命名规范

- 库名与应用名称尽量一致，所有数据库对象名称必须使用 **小写字母并用下划线分割**
- 所有数据库对象名称禁止使用MySQL保留关键字
- 表的命名好是遵循“业务名称_表的作用”
- **表名不使用复数名词**：表名应该仅仅表示表里面的实体内容，不应该表示实体数量
- **所有存储相同数据的列名和列类型必须一致**

数据库基本设计规范

- 数据库和表的字符集统一使用UTF-8
- 所有表和字段都需要添加注释
- 尽量控制单表数据量的大小，建议控制在500万以内（历史数据归档，分库分表）
- 禁止在表中建立预留字段
- 主键列选择：值顺序增长
- 不使用更新频繁的列作为主键，不使用多列主键
- 不使用UUID，MD5，HASH，字符串列作为主键
- 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是，0否）

数据索引规范

- 限制每张表上的索引数量，建议单张表索引不超过5个

索引列选择

SELECT、UPDATE、DELETE语句的WHERE从句中的列

包含在ORDER BY、GROUP BY、DISTINCT中的字段

多表JOIN的关联列

索引列顺序

区分度最高的列放在联合索引的最左侧

尽量把字段长度小的列放在联合索引的最左侧

使用最频繁的列放到联合索引的左侧

- 避免建立冗余索引和重复索引

重复索引：primary key(id)、index(id)、unique index(id)

冗余索引：index(a,b,c)、index(a,b)、index(a)

- 对于频繁查询优先考虑使用覆盖索引

覆盖索引：就是包含了所有查询字段的索引

- 尽量避免使用外键
- **主键索引名为 pk_字段名；唯一索引名为 uk_字段名；普通索引名则为 idx_字段名**
- 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

- 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，保证被关联的段需要有索引
- 利用覆盖索引来进行查询操作，避免回表。一本书要知道第 11 章的标题，会翻开第 11 章对应的一页吗？目录浏览一下就好，这个目录就是起到覆盖索引的作用

数据库字段设计规范

- 将字符串转化为数字类型存储

`INET_ATON('255.255.255.255')=4294967295`

`INET_NTOA(4294967295)='255.255.255.255'`

- 对于非负型的数据来说，要优先使用无符号整型来存储(无符号相对于有符号可以多出一倍的存储空间)
- 尽可能把所有列定义为NOT NULL

索引NULL列需要额外的空间来保存，所以要占用更多的空间

进行比较和计算时要对NULL值做特别的处理

- 同财务相关的金额类数据，必须使用 decimal 类型
- VARCHAR(N)中的N代表的是字符数，而不是字节数
- 使用UTF-8存储汉字VARCHAR(255)=765个字节
- varchar 是可变长字符串，不预先分配存储空间，长度不要超过 5000，如果存储长度大于此值，义字段类型为 text，独立出来一张表，用主键来对应，避免影响其它字段索引效率
- 过大的长度会消耗更多的内存
- 避免使用TEXT、BLOB数据类型
- 避免使用ENUM数据类型

数据库SQL开发规范

- 建议使用预编译语句进行数据库操作
- 避免数据类型的隐式转换（隐式转换会导致索引失效：select * from customer where id='111'）
- 充分利用表上已经存在的索引

避免使用双%号的查询条件。如a like '%123%'

使用left join 或not exists来优化not in 操作

- 禁止使用不含字段列表的 INSERT 语句

`insert into t values('a','b','c'); insert into t(c1,c2,c3) values('a','b','c');`

可减少表结构变更带来的影响

- 避免使用子查询，可以把子查询优化为 join 操作

子查询的结果集无法使用索引

子查询会产生临时表操作，如果子查询数据量大则严重影响效率

消耗过多的CPU及IO资源

- 避免使用 JOIN 关联太多的表

每 join 一个表会多占用一部分内存 (join_buffer_size)

会产生临时表操作，影响查询效率

MySQL最多允许关联61个表，建议不超过5个

- 减少同数据库的交互次数
- 使用in 代替or

in 的值不要超过500个

in 操作可以有效的利用索引

- 禁止使用order by rand()进行随机排序

会把表中所有符合条件的数据装载到内存中进行排序

会消耗大量的CPU和IO及内存资源

推荐在程序中获取一个随机值，然后从数据库中获取数据的方式

- WHERE从句中禁止对列进行函数转换和计算

对列进行函数转换或计算会导致无法使用索引

where date(createtime)='20160901' --> where createtime>='20160901' and createtime<'20160902'

- 在明显不会有重复值时使用UNIONALL而不是UNION

UNION会把所有数据放到临时表中后再进行去重操作

UNION ALL 不会再对结果集进行去重操作

- 拆分复杂的大SQL为多个小SQL

MySQL一个SQL只能使用一个CPU进行计算

数据库操作行为规范

- 超100万行的批量写操作，要分批多次进行操作

大批量操作可能会造成严重的主从延迟

binlog日志为row格式时会产生大量的日志

避免产生大事务操作

- 临时库表必须以tmp为前缀并以日期为后缀
- 备份库，备份表必须以bak为前缀并以日期为后缀
- 对于大表使用 pt-online-schema-change 修改表结构