

写在前面：无规矩不成方圆。对于刚加入互联网的朋友们，肯定会接触到MySQL，MySQL作为互联网最流行的关系型数据库产品，它有它擅长的地方，也有它不足的短板，针对它的特性，结合互联网大多应用的特点，笔者根据自己多年互联网公司的MySQL DBA经验，现总结出互联网MySQL的一些开发规范，仅供参考。

## 一、基础规范

- (1) 使用INNODB存储引擎
- (2) 表字符集使用UTF8
- (3) 所有表都需要添加注释
- (4) 单表数据量建议控制在5000W以内
- (5) 不在数据库中存储图片、文件等大数据
- (6) 禁止在线上做数据库压力测试
- (7) 禁止从测试、开发环境直连数据库

## 二、命名规范

- (1) 库名表名字段名必须有固定的命名长度，12个字符以内
- (2) 库名、表名、字段名禁止超过32个字符。须见名之意
- (3) 库名、表名、字段名禁止使用MySQL保留字
- (4) 临时库、表名必须以tmp为前缀，并以日日期为后缀
- (5) 备份库、表必须以bak为前缀，并以日期为后缀

## 三、库、表、字段开发设计规范

- (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
- (6) 尽可能不使用TEXT、BLOB类型
- (7) 用DECIMAL代替FLOAT和DOUBLE存储精确浮点数
- (8) 越简单越好：将字符转化为数字、使用TINYINT来代替ENUM类型
- (9) 所有字段均定义为NOT NULL
- (10) 使用UNSIGNED存储非负整数

(11) INT类型固定占用4字节存储

(12) 使用timestamp存储时间

(13) 使用INT UNSIGNED存储IPV4

(14) 使用VARBINARY存储大小写敏感的变长字符串

(15) 禁止在数据库中存储明文密码，把密码加密后存储

(16) 用好数值类型字段

Tinyint (1Byte)

smallint (2Byte)

mediumint (3Byte)

int (4Byte)

bigint (8Byte)

类型	字节	最小值	最大值
		(带符号的/无符号的)	(带符号的/无符号的)
TINYINT	1	-128	127
无符号		0	255
SMALLINT	2	-32768	32767
无符号		0	65535
MEDIUMINT	3	-8388608	8388607
无符号		0	16777215
INT	4	-2147483648	2147483647
无符号		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
无符号		0	18446744073709551615

如果数值字段没有那么大，就不要用 bigint

(17) 存储ip最好用int存储而非char(15)

(18) 不允许使用ENUM

(19) 避免使用NULL字段

NULL字段很难查询优化，NULL字段的索引需要额外空间，NULL字段的复合索引无效

(20) 少用text/blob，varchar的性能会比text高很多，实在避免不了blob，请拆表

(21) 数据库中不允许存储大文件，或者照片，可以将大对象放到磁盘上，数据库中存储它的路径

#### 四、索引规范

##### 1、索引的数量要控制：

- (1) 单张表中索引数量不超过5个
- (2) 单个索引中的字段数不超过5个
- (3) 对字符串使用用前缀索引，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

##### 2、主键准则

- (1) 表必须有主键
- (2) 不使用更新频繁的列作为主键
- (3) 尽量不要选择字符串列作为主键
- (4) 不使用UUID MD5 HASH这些作为主键(数值太离散了)
- (5) 默认使用非空的唯一键作为主键
- (6) 建议选择自增或发号器

##### 3、重要的SQL必须被索引，比如：

- (1) UPDATE、DELETE语句的WHERE条件列
- (2) ORDER BY、GROUP BY、DISTINCT的字段

##### 4、多表JOIN的字段注意以下：

- (1) 区分度最大的字段放在前面
- (2) 核心SQL优先考虑覆盖索引
- (3) 避免冗余和重复索引
- (4) 索引要综合评估数据密度和分布以及考虑查询和更新比例

##### 5、索引禁忌

- (1) 不在低基数列上建立索引，例如“性别”
- (2) 不在索引列进行数学运算和函数运算

##### 6、尽量不使用外键

- (1) 外键用来保护参照完整性，可在业务端实现
- (2) 对父表和子表的操作会相互影响，降低可用性

7、索引命名：非唯一索引必须以 idx\_字段1\_字段2命名，唯一所以必须以uniq\_字段1\_字段2命名，索引名称必须全部小写

##### 8、新建的唯一索引必须不能和主键重复

9、索引字段的默认值不能为NULL，要改为其他的default或者空。NULL非常影响索引的查询效率

10、反复查看与表相关的SQL，符合最左前缀的特点建立索引。多条字段重复的语句，要修改语

句条件字段的顺序，为其建立一条联合索引，减少索引数量

11、能使用唯一索引就要使用唯一索引，提高查询效率

12、研发要经常使用explain，如果发现索引选择性差，必须让他们学会使用hint

## 五、SQL规范

(1) sql语句尽可能简单

大的sql想办法拆成小的sql语句(充分利用QUERY CACHE和充分利用多核CPU)

(2) 事务要简单，整个事务的时间长度不要太长

(3) 避免使用触发器、函数、存储过程

(4) 降低业务耦合度，为scale out、sharding留有余地

(5) 避免在数据库中进行数学运算(MySQL不擅长数学运算和逻辑判断)

(4) 不要用select \*，查询哪几个字段就select 这几个字段

(5) sql中使用到OR的改写为用 IN() (or的效率没有in的效率高)

(6) in里面数字的个数建议控制在1000以内

(7) limit分页注意效率。Limit越大，效率越低。可以改写limit，比如例子改写：

```
select id from t limit 10000, 10; => select id from t where id > 10000 limit 10;
```

(9) 使用union all替代union

(10) 避免使用大表的JOIN

(11) 使用group by 分组、自动排序

(12) 对数据的更新要打散后批量更新，不要一次更新太多数据

(13) 减少与数据库的交互次数

(13) 注意使用性能分析工具

Sql explain / showprofile / mysqlsla

(14) SQL语句要求所有研发，SQL关键字全部是大写，每个词只允许有一个空格

(15) SQL语句不可以出现隐式转换，比如 select id from 表 where id='1'

(16) IN条件里面的数据数量要少，我记得应该是500个以内，要学会使用exist代替in，exist在一些场景查询会比in快

(17) 能不用NOT IN就不用NOTIN，坑太多了。。会把空和NULL给查出来

(18) 在SQL语句中，禁止使用前缀是%的like

(19) 不使用负向查询，如not in/like

(19) 关于分页查询：程序里建议合理使用分页来提高效率limit，offset较大要配合子查询使用

(20) 禁止在数据库中跑大查询

(21) 使用预编译语句，只传参数，比传递SQL语句更高效；一次解析，多次使用；降低SQL注入

概率

(22) 禁止使用order by rand()

(23) 禁止单条SQL语句同时更新多个表

## 六、流程规范

(1) 所有的建表操作需要提前告知该表涉及的查询sql;

(2) 所有的建表需要确定建立哪些索引后才可以建表上线;

(3) 所有的改表结构、加索引操作都需要将涉及到所改表的查询sql发出来告知DBA等相关人员;

(4) 在建新表加字段之前, 要求研发至少要提前3天邮件出来, 给dba们评估、优化和审核的时间

(5) 批量导入、导出数据必须提前通知DBA协助观察

(6) 禁止在线上从库执行后台管理和统计类查询

(7) 禁止有super权限的应用程序账号存在

(8) 推广活动或上线新功能必须提前通知DBA进行行流量评估

(9) 不在业务高峰期批量更新、查询数据库