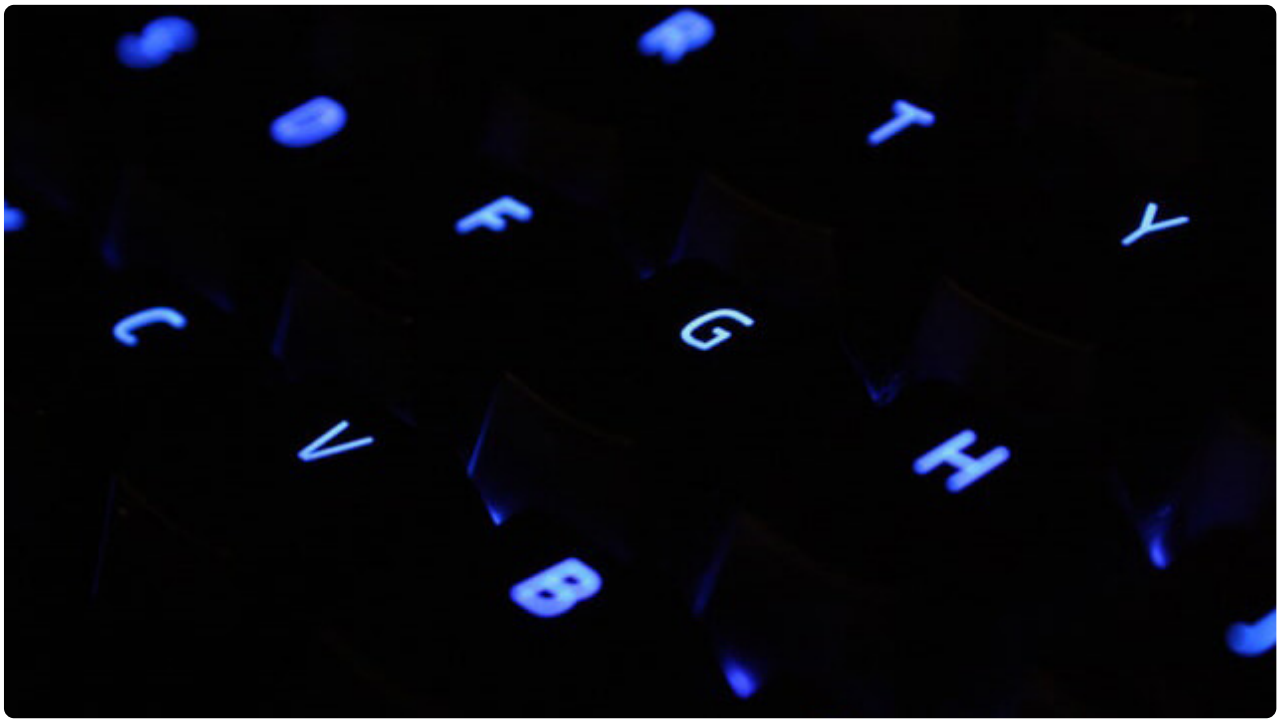
32 MySQL操作规范

更新时间：2019-10-31 10:33:25



> 自信和希望是青年的特权。
>
> ——大仲马

在上一节中，我们讲到了 MySQL 的整体优化思路。但是，有时即使我们在各个环节都优化的很好了，如果一些操作不规范，还是会出现一些问题。因此在本节，我们来聊聊 MySQL 的一些操作规范。当然，这也仅供参考，可能不一定适合每个公司。

1 命名规范

1.1、表名建议使用有业务意义的英文词汇，必要时可加数字和下划线，并以英文字母开头；

1.2、库、表、字段全部采用小写；

MySQL 在 Linux 下默认是区分大小写的，而在 Windows 下不区分大小写。因此，防止出现问题，建议都设置为小写。

1.3、避免用 MySQL 的保留字， MySQL 保留字请参考官方手册：9.3 Keywords and Reserved Words；

1.4、命名（包括表名、列名）禁止超过 30 个字符；

1.5、临时库、表名必须以 tmp 为前缀，并以日期为后缀，如：tmp_shop_info_20190404；

1.6、备份库、表必须以 bak 为前缀，并以日期为后缀，如：bak_shop_info_20190404；

1.7、索引命名：

- 非唯一索引必须按照“idx_字段名称”进行命名；
- 唯一索引必须按照“uniq_字段名称”进行命名。

2 设计规范

2.1、主键：

- 表必须有主键；
- 不使用更新频繁的列做主键；
- 尽量不选择字符串列做主键；
- 不使用 UUID MD5 HASH 做主键；
- 默认使用非空的唯一键。

2.2、如无特殊要求，建议都使用 InnoDB 引擎；

2.3、默认使用 utf8mb4 字符集，数据排序规则使用 utf8mb4_general_ci；

原因：utf8mb4 为万国码，无乱码风险；与 utf8 编码相比，utf8mb4 能支持 Emoji 表情。

2.4、所有表、字段都需要增加 comment 来描述此表、字段所表示的含义；

比如：data_status TINYINT NOT NULL DEFAULT ‘1’ COMMENT ‘1代表记录有效，0代表记录无效’。

2.5、如无说明，表必须包含 create_time 和 update_time 字段，即表必须包含记录创建时间和修改时间的字段；

2.6、用尽量少的存储空间来存数一个字段的数据：

- 能用 int 的就不用 char 或者 varchar；
- 能用 tinyint 的就不用 int；
- 使用 UNSIGNED 存储非负数值；
- 只存储年使用 YEAR 类型；
- 只存储日期使用 DATE 类型。

2.7、存储精确浮点数必须使用 DECIMAL 替代 FLOAT 和 DOUBLE；

原因：在存储的时候，FLOAT 和 DOUBLE 都存在精度损失的问题，很可能在比较值的时候，得到不正确的结果。

2.8、尽可能不使用 TEXT、BLOB 类型；

原因：会浪费更多的磁盘和内存空间，非必要的大量大字段查询会淘汰掉热数据，导致内存命中率急剧降低，影响数据库性能。如果实在有某个字段过长需要使用 TEXT、BLOB 类型，则建议独立出来一张表，用主键来对应，避免影响原表的查询效率。

2.9、禁止在数据库中存储明文密码；

2.10、索引设计规范：

a、需要添加索引的字段

- UPDATE、DELETE 语句的 WHERE 条件列；

- ORDER BY、GROUP BY、DISTINCT 的字段（原因可复习第 6 节）；
- 多表 JOIN 的字段（原因可复习第 8 节）。

b、单表索引建议控制在 5 个以内；

c、适当配置联合索引；

比如方便查询能走覆盖索引，或者几个字段同时作为条件的概率很高时，当然还有其他很多种情况可以设置联合索引，具体可以复习第 13 节。

d、业务上具有唯一性的字段，添加成唯一索引；

遇到过几次字段在业务场景上要求唯一，但是该字段在数据库里的数据却出现了重复。因此在代码层考虑外，还需要在 MySQL 上的对应字段添加唯一索引。

e、在 varchar 字段上建立索引时，建议根据实际文本区分度指定索引长度；

原因：可以降低索引所占用的空间，并且很多时候，比如字符串基本是长度大于 20，但是只要建立长度为 20 的索引，就已经有很高的区分度了。可以使用 count(distinct left(列名, 索引长度))/count(*) 的区分度来确定。

f、索引禁忌：

- 不在低基数列上建立索引，例如：性别字段。
- 不在索引列进行数学运算和函数运算（原因，做函数操作可能会导致使用不了索引，具体可以复习第 4 节）

2.11、不建议使用外键；

原因：外键会导致表与表之间耦合，update 与 delete 操作都会涉及相关联的表，十分影响 sql 的性能，甚至会造成死锁。高并发情况下容易造成数据库性能。

2.12、禁止使用存储过程、视图、触发器、Event ；

原因：高并发的情况下，这些功能很可能将数据库拖死，业务逻辑放到服务层具备更好的扩展性。

2.13、单表列数目建议小于 30；

2.14、表示例：

```
CREATE TABLE student_info (
 id INT (11) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '主键',
 stu_name VARCHAR (10) NOT NULL DEFAULT '' COMMENT '姓名',
 stu_class VARCHAR (10) NOT NULL DEFAULT '' COMMENT '班级',
 stu_num INT (11) NOT NULL DEFAULT '0' COMMENT '学号',
 stu_score SMALLINT UNSIGNED NOT NULL DEFAULT '0' COMMENT '总分',
 tuition DECIMAL (5, 2) NOT NULL DEFAULT '0' COMMENT '学费',
 phone_number VARCHAR (20) NOT NULL DEFAULT '0' COMMENT '电话号码',
 create_time datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '记录创建时间',
 update_time datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '记录更新时间',
 status TINYINT NOT NULL DEFAULT '1' COMMENT '1代表记录有效，0代表记录无效',
 PRIMARY KEY (id),
 UNIQUE KEY uniq_stu_num (stu_num),
 KEY idx_stu_score (stu_score),
 KEY idx_update_time_tuition (update_time, tuition)
) ENGINE = INNODB charset = utf8mb4 COMMENT '学生信息表';
```

3 SQL语句规范

3.1、避免隐式转换；

具体原因可以复习第 4 节中第 2 部分：隐式转换。

3.2、尽量不使用select *,只 select 需要的字段 ；

原因：读取不需要的列会增加 CPU、IO、NET 消耗，并且不能有效的利用覆盖索引。使用 SELECT * 容易在增加或者删除字段后导致程序报错。

3.3、禁止使用 INSERT INTO t_xxx VALUES (xxx)，必须显示指定插入的列属性 ；

原因：容易在增加或者删除字段后导致程序报错。

3.4、尽量不使用负向查询；

比如 not in/like。

3.5、禁止以 % 开头的模糊查询。

原因：使用不了索引（具体例子可以复习第 4 节中的第 3 部分：模糊查询）。

3.6、禁止单条 SQL 语句同时更新多个表；

3.7、统计记录数使用 select count(*)，而不是 select count(primary_key)或者 select count(普通字段名)；

原因：可能会导致走的索引不是最优的或者导致统计数字不准确。（具体例子可以复习第 9 节）。

3.8、建议将子查询转换为关联查询；

3.9、建议应用程序捕获 SQL 异常，并有相应处理；

3.10、SQL 中不建议使用 sleep()，如特殊需求需要用到 sleep()，请提前告知 DBA；

3.11、避免大表的 join。

4 行为规范

4.1、批量导入、导出数据必须提前通知 DBA 协助观察；

4.2、有可能导致 MySQL QPS 上升的活动，提前告知DBA；

4.3、同一张表的多个 alter 合成一次操作；

4.4、不在业务高峰期批量更新、查询数据库；

4.5、删除表或者库要求尽量先 rename，观察几天，确定对业务没影响，再 drop。

5 总结

本节讲解了 MySQL 的一些操作规范，主要讲解了下面这些场景的规范：

1. 命名规范；
2. 设计规范；
3. SQL 语句规范；
4. 行为规范。

当然，各个公司可能都有自己独有的 MySQL 使用规范，因此这篇文章仅供参考。

当制定出合理的 MySQL 使用规范，并严格按照规范操作，很多问题都可以在源头上避免掉。

6 问题

这是专栏的最后一节正式内容，对于 MySQL 的规范，相信有些方面我没考虑到。因此，这节的问题就是：在这篇文章中，还有哪些规范需要补充的？哪些规范又不是很合理的？欢迎在留言区补充和讨论。

7 参考资料

《MySQL 工作笔记》第 4 章：SQL 开发规范和基础

《阿里巴巴Java开发手册》第五章：MySQL 数据库

}

31 MySQL整体优化思路

33 加餐：答疑篇（二）