

# 为什么DBA要求MySQL表索引不能超过5个

原创 DB哥 DB哥 2025年08月05日 07:01 安徽

程序员小张：“我的表才8个索引，查询快得飞起！”  
DBA老王反手甩出监控图：“你管这叫快？写操作都堵成春运火车站了！”

一、从“图书馆灾难”说起  
想象你管理一个图书馆（MySQL表），每本书（数据行）都有索引卡（索引）记录位置：

- 书名索引 (name\_index)
  - 作者索引 (author\_index)
  - ISBN索引 (isbn\_index)
  - .....
- 当新增一本书时，你需要更新所有索引卡（索引维护成本）

某天小张要求增加：

- 封面颜色索引 (cover\_color\_index)
  - 出版社成立年份索引 (found\_year\_index)
  - 责编星座索引 (editor\_star\_sign\_index)
  - .....
- 馆长（DBA）崩溃拍桌：“你再加索引，新书入库速度比蜗牛还慢！”

## 二、索引超标的4大“罪状”（基于MySQL 8.0 InnoDB）

```
1  -- 创建测试表 dbbro_user
2  CREATE TABLE dbbro_user (
3      id INT AUTO_INCREMENT PRIMARY KEY,
4      name VARCHAR(50) NOT NULL,
5      email VARCHAR(100) NOT NULL,
6      age TINYINT,
7      reg_time TIMESTAMP,
8      vip_level TINYINT
9  ) ENGINE=InnoDB;
10 -- 疯狂加索引 (7个!)
11 ALTER TABLE dbbro_user
12     ADD INDEX idx_name (name),
13     ADD INDEX idx_email (email),
14     ADD INDEX idx_age (age),
15     ADD INDEX idx_reg_time (reg_time),
16     ADD INDEX idx_vip_level (vip_level),
17     ADD INDEX idx_name_email (name, email),
18     ADD INDEX idx_age_vip (age, vip_level);
```

### 1 写操作性能雪崩

每次INSERT/UPDATE/DELETE，索引都要更新！

INSERT/DELETE 必更新所有索引  
UPDATE 只更新被修改字段关联的索引  
索引越多，高频更新字段带索引时性能衰减越剧烈

### 2 查询优化器选择困难症

优化器（Optimizer）选择索引时，需要分析所有可能性：

索引太多 → 优化器计算成本飙升 → 生成执行计划变慢！

👉 通过 `OPTIMIZER_TRACE` 验证：

```
1 SET optimizer_trace = "enabled=on";
2 SELECT * FROM dbbro_user WHERE ...;
3 SELECT * FROM information_schema.optimizer_trace;
```

输出中可见：

```
1  "considered_execution_plans": [
2    {
3      "plan_prefix": [],
4      "table": "dbbro_user",
5      "index": "idx_name",
6      "cost": 356.2
7    },
8    {
9      "index": "idx_name_email",
10     "cost": 102.4
11   },
12   ... # 其他索引计算成本
13 ]
```

索引越多，计算量指数级上升！

3 空间刺客：磁盘与内存的噩梦

每个索引都是一颗B+树（InnoDB引擎）：

- 非叶子节点：存储索引键值 + 指针
- 叶子节点：存储主键值（二级索引）

假设 `dbbro_user` 表数据1GB：

- 5个索引 → 额外占用约3.5GB
- 8个索引 → 额外占用约5.6GB

后果：

- 磁盘撑爆：`SELECT @@innodb_temp_data_file_path;` 狂飙
- 内存不足：Buffer Pool被索引占据，热数据被挤出！

4 索引打架：多个索引可能“抢着干活”

```
1 SELECT * FROM dbbro_user
2 WHERE age > 20
3    AND vip_level = 3
4 ORDER BY reg_time DESC;
```

- 如果选 `idx_age_vip` → 能过滤 `age` 和 `vip_level`
- 如果选 `idx_reg_time` → 排序快但过滤全表

索引太多可能导致优化器“选错索引”（统计信息不准时更严重）！

破局之道：

当开发申请第6个索引时，DBA会灵魂三问：

1. 业务场景：这个索引解决什么性能问题？
2. 替代方案：能否用联合索引覆盖？
3. 代价评估：写性能下降30%能接受吗？

- 不放 → 查询慢得吃不下饭 (全表扫描)
- 放太多 → 写操作咸到齁嗓子 (维护代价)




**“5个索引是黄金分割点  
再想要？拿业务逻辑来换！”**

如有疑问，可以留言，DB哥看到后会及时回复，也可以加DB哥微信交流

加入「DB哥数据库帮」

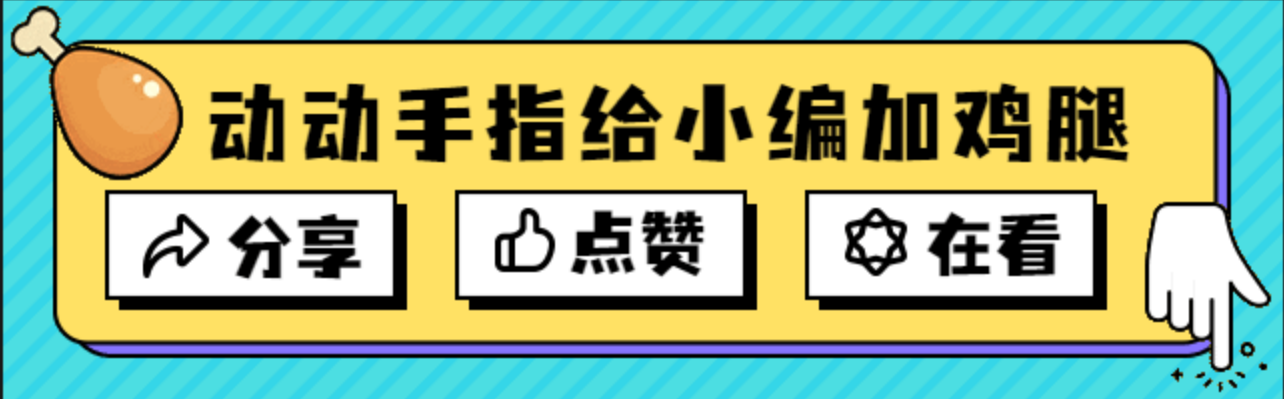
**DB哥数据库帮专属福利**  
▶ 授人以渔

DB哥  
10年数据库救火队老炮 | 用实战教你少熬三夜。遇到数据库别慌，DB哥专治数据库各种...  
207篇原创内容

- 1  10年数据库救火队老炮 | 用实战教你少熬三年夜
- 2  亲手调优3000+故障库 | 企业级数据库架构
- 3  库崩了？锁死了？SQL慢如🐢？CPU100%
- 4  别慌，DB哥专治数据库各种“不调”！

```
1 UPDATE salary SET bonus =0;-- 小张干的！
```

1. 不准在生产环境执行UPDATE不带WHERE，否则罚用触控板代替鼠标一周
2. 删库后不跑路，否则罚用Windows XP装 MySQL5.0（不兼容也要装）
3. 必须用 JOIN 代替子查询，否则罚直播用子查询实现复杂报表（不许用JOIN！）
4. 生产环境执行DDL必须测试，否则罚胸口碎大石（罪名：惊动监控告警）
5. 不用SELECT \* 横扫全表，否则罚罚抄《索引优化十诫》100遍（用毛笔写SQL语句）



DB哥

👍 稀罕作者

DB哥讲数据库 · 目录

← 上一篇

别让索引变备胎！MySQL 的“裁员”艺术

下一篇 →

为什么DBA怒吼：MySQL小数必须用 decimal？float/double是隐藏的财务刺客！