

SQL优化实战：从慢如蜗牛到快如闪电的必杀技

原创 farerboy 小林聊编程 2025年03月02日 10:00 福建

你点赞了吗？你关注了吗？每天分享干货好文。

大厂资深全栈开发，多年技术架构与技术管理经验，多年面试官经验。

还在为面试拿不到 offer 发愁吗？免费一对一面试指导，改进面试过程。

技术指导培训，带你从入门到精通，从crud到架构，从coding到管理，快速成长，快速拿 offer。

感兴趣的私我【farerboy】，免费领取学习资料，带你由入门到实战。



为什么你的SQL总是“跑不动”？

“这个查询怎么要10秒？！”——开发中最崩溃的瞬间，往往来自一条性能拉胯的SQL。

根据Oracle官方数据，80%的数据库性能问题源自低效SQL。而一次全表扫描的耗时可能是索引查询的100倍以上。本文将从执行原理到实战技巧，手把手教你成为SQL调优高手。

一、核心优化原则：让索引为你打工

1. 避免索引失效的六大禁忌

- 字段计算陷阱：WHERE子句中避免对索引字段进行运算
反例：`WHERE salary/12 > 5000` → 正例：`WHERE salary > 5000*12`
- 模糊查询黑洞：前导通配符导致索引失效
反例：`LIKE '%张三%'` → 正例：`LIKE '张%'`
- 类型转换灾难：隐式类型转换让优化器迷茫
反例：`WHERE id = '100'` (id为数值型) → 正例：`WHERE id = 100`

2. 索引设计的黄金法则

- 复合索引顺序：遵循最左前缀原则，高频查询字段放左侧
- 覆盖索引妙用：SELECT字段尽量包含在索引中，减少回表查询
- 索引数量控制：单表索引不超过5个，避免写操作性能下降

二、实战技巧：改写SQL的智慧

1. 拒绝“无脑查询”

- **SELECT**：只取所需字段，数据传输量减少50%
反例：`SELECT * FROM orders` → 正例：`SELECT order_id, amount FROM orders`
- **UNION ALL**替代**UNION**：避免重复数据过滤的开销

2. 复杂条件优化

- **OR**条件拆分：用**UNION ALL**替代**OR**连接
反例：`WHERE id=1 OR id=3` → 正例：`SELECT ... UNION ALL SELECT ...`
- **EXISTS**妙用：小表驱动大表时性能提升显著

3. 批量操作的艺术

- **批量插入**：单次提交1000条数据比逐条插入快20倍

```
1 INSERT INTO users (id, name) VALUES
2 (1, '张三'), (2, '李四'), ...;
```

- **分页优化**：避免 `LIMIT 100000, 20` 式深分页，改用ID范围查询

三、高阶武器：性能分析工具

1. EXPLAIN执行计划解读

具体请参看：SQL 优化工具使用之 explain 详解

- **关键指标**：
 - `type`：`ALL` 代表全表扫描，需优化为 `ref` 或 `range`
 - `rows`：扫描行数越少越好
 - `Extra`：出现 `Using filesort` 或 `Using temporary` 需警惕

2. 慢查询日志分析

- 开启方式：`SET GLOBAL slow_query_log = ON;`
- 分析工具：Percona Toolkit、pt-query-digest

3. SHOW PROFILE深度追踪

- 查看SQL各阶段耗时：

```
1 SET profiling = 1;
2 SELECT * FROM orders;
3 SHOW PROFILES;
4 SHOW PROFILE FOR QUERY 1;
5 ```:cite[3]
```

四、避坑指南：这些“优化”可能是毒药

1. **过度索引**：索引维护成本可能超过查询收益¹⁰
2. **盲目并行**：高并发下可能引发资源争用⁸
3. **游标滥用**：万行以上数据操作优先考虑集合运算

文末福利：

关注公众号回复“MySQL数据库设计规范”，领取《MySQL数据库设计规范》

如果有其它问题，欢迎评论区沟通。

感谢观看，如果觉得对您有用，还请动动您那发财的手指头，点赞、转发、在看、收藏

关注公众号       



小林聊编程

专注分享原创技术干货。大厂资深架构师，多年技术架构与技术管理经验，多年面试官经...

96篇原创内容

公众号

十年资深开发，带你从入门到精通，从crud到架构，从coding到管理，快速成长，快速拿offer。



farerboy

喜欢作者

数据库优化系列 · 目录

上一篇

SQL 优化工具使用之 explain 详解

下一篇

基于 Redis 分布式缓存实现：从理论到实践