

SQL建议

1. 尽量不要用 selcet *

2. 如果已知查询结果只有一条或只要最大/最小一条，建议 limit

3. 避免在where中使用or

or 导致索引失效

使用 union 替代 or

4.优化limit分页

• `select id, name from employee where id>10000 limit 10.`

• `select id, name from employee order by id limit 10000, 10`

• `select id, name, age from employee limit 10000, 10 - 反例`

5. 优化like

%123 - 不走索引

123% - 走索引

6. 使用where条件限定要查询的数据，避免返回多余的行

`select userId from user where isVip=1` 返回多条

`select userId from user where userId='userId' and isVip='1'` 返回一条

7. 避免在索引列上使用mysql内置函数

内置函数导致索引失效

8. 避免在where中使用表达式操作

where age-1 = 10 之类的

9. 避免在where中使用 != 或 <> 操作符

10. 使用联合索引时，遵从最左匹配原则

11. 查询优化，考虑在where和order by 涉及的列上建立索引

12. 如果插入过多，考虑批量插入

- 一次500批量插入，分批进行
- `insert into user(name,age) values`
- `<foreach collection="list" item="item" index="index" separator=",">`
- `(#{item.name},#{item.age})`
- `</foreach>`

13. 慎用 distinct

过滤重复记录时，如果字段很多，则会降低效率

• `SELECT DISTINCT * from user;`

• `select DISTINCT name from user;`

14. 删除冗余和重复索引

15. 一次修改/删除语句不要太大

如果数据量过大，导致长时间占用CPU，影响他人使用

可以分批进行删除

16. where子句中考虑使用默认值代替 null

• `select * from user where age is not null;`

• `select * from user where age>0;`

17. exist & in 的合理使用

小表驱动大表，让外层循环次数尽量小

链接两次，每次做上百万次的查询

链接上百万次，每次做两次查询

链接的申请和释放很耗费性能

根据两表数据量大小选择 in 和 exist

18. 尽量使用数字型字段

反例: ``king_id` varchar (20) NOT NULL COMMENT '守护者Id'`

正例: ``king_id` int(11) NOT NULL COMMENT '守护者Id'`

字符型会降低查询和连接的性能，增加存储开销

19. 索引不合适建在有大量重复数据的字段上

20. 尽量避免向客户端返回过多数据

21. 尽量使用varchar/nvarchar 替代 char/nchar

变长字段存储空间小

22. 在执行group by 语句前，先把不需要的记录过滤掉

反例: `select job, avg (salary) from employee group by job having job = 'president' or job = 'managent'`

正例: `select job, avg (salary) from employee where job = 'president' or job = 'managent' group by job;`

23. 使用 explain