

MySQL优化

一、SQL语句优化

1. 不要把select子句写成select *

```
select * from test;
```

2. 谨慎使用模糊查询

```
select ename from test where ename like '%S%'
select ename from test where ename like 'S%'
```

3. 对order by 排序的字段添加索引 (索引是个二叉树)

```
select ename from test order by num;
```

4. 要用is null 或 is not null的表达式会跳过索引进行全表扫描

```
select ename from test where comm IS NOT NULL;
可用以下语句替换
select ename from test where comm >=0;
```

5. 尽量少用 != 运算符

```
select ename from test where num != 20 [全表扫描]
可用以下语句替换
select ename from test where num >20 and num <20;
```

6. 尽量少用 or 运算符【or之前的能用到索引、or后边的会跳过索引会进行全表扫描】

```
select ename from test where num = 20 or num = 33
可用以下语句替换
select ename from test where num =20
union all
select ename from test where num =30
```

7. 尽量少用 in 或 not in 运算符

```
select ename from test where num in(20,30)
可用以下语句替换
select ename from test where num =20
union all
select ename from test where num =30
```

8. 避免条件语句中的数据类型转换

```
select ename from test where num = '20'
可用以下语句替换
select ename from test where num =20
```

9. 在表达式左侧使用运算符和函数都会让索引失效

```
select ename from test where salary * 12 > 100000
可用以下语句替换
select ename from test where salary> 100000/12

select ename from test where year(addtime) >= 2019
可用以下语句替换
select ename from test where addtime >= '2019-01-01 00:00:00'
```

二、MySQL参数优化

1. 优化最大连接数

- max_connections 是MySQL最大并发连接数，默认值151
- MySQL允许的最大连接数量上线为16384
- 实际连接数是最大连接数的85%较为合适

```
show variables like 'max_connections';    # 最大连接数
show status like 'max_used_connections';  # 实际连接数
```

2. 优化请求堆栈

- back_log 是存放执行请求的堆栈大小，默认值为50，可设置为最大连接数的20%-30%

3. 修改并发线程数

- innodb_thread_concurrency 代表并发线程数，默认是0
- 并发线程数一般是CPU核心数量的两倍

4. 修改连接超时

- wait-timeout 是超时时间，单位是秒
- 默认连接超时为8h，连接长期不用又不销毁，浪费资源

```
# 十分钟有效期  
wait-timeout = 600
```

5. InnoDB的缓存机制

- 部分数据表中的数据缓存
- 部分索引中的索引缓存
- 修改InnoDB缓存大小

innodb_buffer_pool_size 是InnoDB的缓存容量，默认是128M,可设置为主机内存的70% - 80%

注意：以上配置修改都基于mysql的my.cnf 【默认/etc/my.cnf】,修改完成后请重启mysql

```
max_connections = 300  
back_log = 90  
innodb_thread_concurrency = 2  
wait-timeout = 600  
innodb_buffer_pool_size = 300M  
  
# 查看是否开启慢日志  
show variables like 'slow_query';  
slow_query_log=ON # 开启慢查询日志  
long_query_time=1 # 查询超时时间 1s
```

