# mysql慢查询日志

## 1.1慢查询日志相关命令

(1)查询慢查询是否开启:show variables like 'slow\_query\_log';

(2)查询日志相关参数: show variables like '%log%';

(3)设置没有索引的记录到慢查询日志：

set global log\_queries\_not\_using\_indexes=on;

(4)查询慢查询超时时间:show variables like 'long\_query\_time';

(5)设置慢查询超时时间:set long\_query\_time=1;

(6)开启慢查询日志:set slow\_query\_log=on;

(7)查看慢查询日志的存储位置:show variables like 'slow%';

(8)改变慢查询输出方式:set log\_output=‘FILE,TABLE’;

show variables like 'log\_output';

1. mysql慢查询日志工具:mysqldumpslow
2. show variables like '%general\_log%';

-- show variables like 'log\_output';

-- select \* from mysql.general\_log;

# 性能优化

## SQL语句优化.

(1)子查询的优化:通常优化为join连接的方式.

(2)gpoup by语句优化:尽量用一个表中的字段,避免用临时表.

(3)limite语句优化:

·使用order by语句的时候尽量用主键排序,减少IO操作

·当表主键id连续得时候可用记录最大索引位置得方法优化limit语句,索引不连续得时候使用范围查询时返回每页数据会减少.

(4)索引:

·索引利于查询,影响insert,update语句

·查询冗余索引和重复索引:

select

a.TABLE\_SCHEMA AS '数据名',

a.TABLE\_NAME AS '表名',

a.INDEX\_NAME AS '索引1',

b.INDEX\_NAME AS '索引2',

a.COLUMN\_NAME as '重复列名'

from STATISTICS a JOIN STATISTICS b ON

a.TABLE\_SCHEMA = b.TABLE\_SCHEMA

AND a.TABLE\_NAME = b.TABLE\_NAME

AND a.SEQ\_IN\_INDEX = b.SEQ\_IN\_INDEX

AND a.COLUMN\_NAME = b.COLUMN\_NAME;

需要在information\_scheme数据库中查询

## 数据库结构的优化.

(1)选择合适的数据类型,尽量使用not null定义字段,尽量少用text类型非用不可时用附加表

(2)IP地址：插入的时候：INET\_ATON(192.158.2.114).读取的时候要INET\_NTOA(ip)转化。而将ip地址的存贮形式改为BIGINT

(3)表结构符合第三范式结构查询时会关联多张表,可合理的进行反范式化设计表.

(4)表的垂直拆分,把拥有多个列的表拆成几张表:

·把不常用的字段单独放到一个表中

·大字段独立存放一个表中

·经常用的字段放到一张表中