**优化sql语句的一般步骤**

**通过show status命令了解各种sql执行频率**

* mysql> show status like 'Com\_%'
* Com\_xxx表示每个xxx执行的次数，我们主要关心几个统计值。
* Com\_select执行select操作查询的次数，一次查询只累加1。
* Com\_insert执行insert操作的次数，批量操作只加1。
* Com\_update执行update操作的次数。
* Com\_delete执行delete操作的次数。  
  *以上对所有数据库引擎都会进行累计，下面这几个参数只针对Innodb引擎*
* Innodb\_rows\_read select查询返回的行数
* Innodb\_rows\_inserted 执行insert操作插入的行数
* Innodb\_rows\_updated 执行update操作更新的行数
* Innodb\_rows\_deleted 执行delete操作删除的行数  
  **查看以上参数可以看出当前数据库是以插入为主还是查询为主**
* show status 'connections' 连接数据库次数
* show status like 'uptime' 服务器工作时间
* show status like 'show\_queries' 慢查询次数

**通过查询慢查询日志**

* 通过long\_query\_time查询慢查询时间
* 查看慢查询日志log-slow-queries慢查询配置
* 通过查询 show processlist查看当前mysql进行线程

**通过explain分析执行效率低的语句**

* explain select count(1) from test
* select\_typeselect的查询类型
  + simple 简单表
  + primary 主见查询
  + union 第二个或者后面的查询
  + subquery 子查询中的第一个select
* table输出结果的表
* type 表示mysql在表中查询方式
  + all 扫描全表
  + index索引扫描
  + range索引扫描范围，常见于<,<=,>,>=,between
  + ref非唯一索引扫描
  + eq\_ref唯一索引扫描
  + const,system最多只有一行匹配，查询非常快，例如主键primary key/唯一索引unique index/表中只有一条记录
  + null不用访问表或者索引，直接就能得到结果explain select 1 from test where 1
* explain extened select count(1) from test where 1=1可以查看执行sql前，mysql做了哪些优化
* explain partitions select id from test where id = 15可以查看分区名称

**查看索引使用情况**

show status like 'Hanlder\_read%'  
如果Hanlder\_read\_rnd\_next值比较高索命表索引不正确

**分析表**

analyze table emp\_date

**检查表**

check table emp\_date

**优化表**

optimize table emp\_date

**修改数据库引擎**

alter table emp\_date engine = innodb

**大批量导入数据优化**

**对于MyIsAM优化**

* 关闭MyIsAM存储引擎表的非唯一索引

alert table emp\_date DISABLE KEYS; //关闭

load file infile 'xxx.txt' into table emp\_date; //导入数据

alert table emp\_date ENABLE KYES; //开启

**对于Innodb导入大数据优化**

* 对于Innodb数据库，导入的数据按照主键顺序要比无序数据快
* 关闭唯一性校验set unique\_checks = 0,导入后再开启set unique\_checks = 1
* 关闭自动增长set autocommit=0，导入后在开启set autocommit = 1

**优化Insert语句**

* 一次性执行多个insert语句，尽量使用insert into emp\_date values (1,2,3),(2,2,3),(3,3,4)
* 索引文件和数据文件建立在不同的磁盘上
* 当重一个文件载入到一个表时，使用load date infile 'xx.txt',比使用多个insert快20倍

作者：妖精的九尾  
链接：https://www.jianshu.com/p/ef3d00991c42  
來源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。