mysql> show variables like '%slow\_query\_log%';查询慢查询是否开启

mysql> set global slow-query\_log=1;开启本次数据库连接慢查询

mysql> show variables like 'long\_query\_time%';

默认大于10s的查询会被记录

mysql> set global long\_query\_time=4;

对于MySQL系统本身，我们可以使用工具来优化数据库的性能，通常有三种：**使用索引，使用EXPLAIN分析查询以及调整MySQL的内部配置。**

**性能瓶颈定位**

**Show命令**

我们可以通过show命令查看MySQL状态及变量，找到系统的瓶颈：

Mysql> show status ——显示状态信息（扩展show status like 'XXX'）

Mysql> show variables ——显示系统变量（扩展show variables like 'XXX'）

Mysql> show innodb status ——显示InnoDB存储引擎的状态

Mysql> show processlist ——查看当前SQL执行，包括执行状态、是否锁表等

Shell> mysqladmin variables -u username -p password——显示系统变量

Shell> mysqladmin extended-status -u username -p password——显示状态信息

查看状态变量及帮助：

Shell> mysqld --verbose --help [|more #逐行显示]

**慢查询分析**

**mysqldumpslow**

**我们可以通过打开log文件查看得知哪些SQL执行效率低下**

**explain分析查询**

使用 EXPLAIN 关键字可以模拟优化器执行SQL查询语句，从而知道MySQL是如何处理你的SQL语句的。这可以帮你分析你的查询语句或是表结构的性能瓶颈。**通过explain命令可以得到:**

– 表的读取顺序

– 数据读取操作的操作类型

– 哪些索引可以使用

– 哪些索引被实际使用

– 表之间的引用

– 每张表有多少行被优化器查询

**profiling分析查询**

通过**慢日志查询可以知道哪些SQL语句执行效率低下**，通过**explain我们可以得知SQL语句的具体执行情况，索引使用等**，还可以结合**show命令查看执行状态。**

如果觉得explain的信息不够详细，可以同通过**profiling命令得到更准确的SQL执行消耗系统资源的信息。**

举例：又发现一个有趣的现象：

**大表 左关联 小表，很慢；小表 左关联 大表，很快。**

当然，这些我们理论上都知道，但实际开发会忘却。又或者一开始两个表都为空时，而又没考虑到后期这两个表增长的速度时，日后就会埋下坑了。

**总结**

首先，嵌套的子查询是很慢的。

是可以通过慢查询搜索到的，应该对其进行再一步的优化。