1.

MySQL的日志大概可以分成三种，错误日志(error\_log)，查询日志(query\_log)，二进制日志(binary\_log)：

查询日志(query log)：一般的，查询日志可以分为两种，通用查询日志(general query log)和慢查询日志(slow query log)；其中，通用查询日志可以用来各客户端连接时的相关信息和在数据库上执行的SQL语句；慢查询日志记录了SQL语句时间超过了预设的long\_query\_time的语句，在数据量较大的情况下，可以看看慢查询日志中有哪些语句需要进行优化。

二进制日志(binary\_log)：简单来说，二进制日志记录了对MySQL更新的操作，主要目的是尽可能的将数据库恢复到数据库故障点，因为二进制日志包含备份后进行的所有更新。

2.

我们通过对my.cnf进行配置以后，启动mysql。我们可以使用以下语句进行验证是否配置被读取并且生效。

例如通过对数据库查询可以查看当前的log设置：

mysql> show variables like 'log\_error'; #展示log\_error日志的地址

+---------------+---------------------+

| Variable\_name | Value |

+---------------+---------------------+

| log\_error | /var/log/mysqld.log |

+---------------+---------------------+

mysql> show variables like "datadir"; #展示数据库文件存放的地址

mysql> show variables like "socket”; #展示socket文件存放的地址

3.

mysql的配置从my.cnf中读取。里面可以分为2个区域[mysqld]以及[mysqld\_safe] 。 [mysqld]的配置信息将用于mysql服务器通过mysqld启动的时候读取，mysqld\_safe用于mysql服务器通过mysqld\_safe启动的时候读取。

mysqld\_safe启动比mysqld启动会多一些安全特性，例如当mysql崩溃的时候自动重启等。

例如配置log error:

1、找到配置文件/etc/my.cnf，如果找不到find / -type f -name 'my.cnf'全局查找即可

2、将错误日志参数写入配置文件

[mysqld]

log-error=/var/lib/mysql/mysql.err

4.

Mysql数据库文件：

mysql数据库在系统上是以一个文件夹的形式出现的，假设创建了Test数据库，那么在磁盘上就会出现Test的目录，而Test数据库的全局配置则是Test文件夹中的db.opt文件来管理。mysql的数据文件在datadir下，你在数据库中执行 show variables like '%datadir%';就会显示[数据库文件](https://www.baidu.com/s?wd=%E6%95%B0%E6%8D%AE%E5%BA%93%E6%96%87%E4%BB%B6&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)的路径，你可以找到Test文件夹。

5.

MySQL储存数据，处理数据的依靠为数据引擎 (Storage Engine) 。在MySQL中，不同的数据引擎会带有不一样的支持形成，例如是否支持事务处理，是否支持并行处理，最大容纳多少数据量, 数据库锁的颗粒度是表格锁还是行锁, 是否支持外键限制等等。

查看本MySQL版本所支持的数据引擎以及所选用的数据引擎。

登陆进mySql控制台show engines

MySQL自20多年前成立以来一直支持可插拔存储引擎，但在一段相当长的时间里MyISAM一直是默认的存储引擎.MySQL刚开始是为小型网站的小型数据库设计的，许多应用已经习惯使用MyISAM存储引擎。

　　刚开始没什么问题，一切正常，但现在的问题是：MyISAM没有考虑到应用到高并发高负载，多核CPU和RAID阵列的场景，也不能弹性扩展。所以网站流量越来越多后，他们不能扩展，因为MySQL查询会在表级锁上等待数秒 (MyISAM只支持这种锁机制) 。他们不想每次MySQL崩溃时损坏他们的业务数据。

许多人并不知道，自MySQL存在以来MyISAM存储引擎就有一个兄弟叫InnoDB。并且高并发负载，性能和弹性(也包括原子性，一致性和隔离)正是它的特长。  
当然，在InnoDB发展过程中也有过一些问题(尤其是2006年5.0.30之前的版本的性能问题)，但在这之后的10年时间里，InnoDB已经在你能想到的领域(或者没有)得到了证明，而MyISAM已经很少被关注了。  
🡪**因此，从MySQL 5.5.5开始，InnoDB成为默认的存储引擎，**现在你几乎找不到大型MySQL数据库的安装使用MyISAM而不是InnoDB。

🡪MyISAM存储引擎不支持外键限制。

6.

查看mySql的版本mysql -V

7.

在mysql 5.1中，Alert操作具体流程

如果对表中的某个域进行alert操作，并且该域需要影响表格中的数据时（例如，修改某个域的类型并且需要所有的数据行上的该域数据都变化），则mysql实际上会根据目标表格首先复制一个临时表，然后在临时表中进行操作。并且在此期间内对原表的读操作不会有影响，对原表的写作操会锁住，并且等修改临时表操作完成后让暂停的写操作应用在新表当中。所以数据库在更改时所占用的空间会变大。如果这种临时表所需要的空间不够，则会修改失败。

如果对表中的某个域进行alert操作，并且该域不需要影响表格中的数据时（例如，修改某个域的名称，不需要每个数据行进行操作），则mysql不会对原表进行复制。Mysql只修改元数据马上完成。

在mysql 8.x中，Alert操作可以选择不同的算法。可以选择复制临时表处理方式， 可以选择直接对原表进行处理，可以执行只修改元数据不对具体数据行进行任何处理等方式。Mysql 8.0的默认行为是只修改元数据而不改变具体数据行。

8.