1. 数据库结构优化
2. 优化表结构
3. 尽量定义表字段为NOT NULL，因为在MYSQL中含有空值的列很难查询优化；
4. 对于只包含特定类型的字段，可以使用enum, set等数据类型；
5. 数值型字段的比较比字符串效率高的多，字段类型尽量使用最小的数据类型；
6. 尽量使用TINYINT, SMALLINT, MEDIUM\_INT这种作为整数类型减少字段的字节数，指定宽度是没有用的，因为指定的类型标识范围时确定的；
7. 单表的字段建议在20以内；
8. 合理地加入冗余字段可以提高查询速度；
9. 表拆分

当一个表的数据量过多时，可以将其拆为多个表，有两种拆分方式，一种是垂直拆分，一种是水平拆分。

1. 垂直拆分

按照子弹进行拆分，即把不同的列拆分到不同的表中。如用户表中的一些字段可能经常访问，则可以将这些字段放进一张表中，其他的字段放到另一张表中。

1. 水平拆分

按照行进行拆分，常见的就是分库分表，将不同的行分到不同的表中，缓解表的访问压力。在查找时，在这些分表中同时查找。

1. 表分区

分区是用于例如日志记录，查询少的情景。一般用于后台的数据报表分析。Mysql主要支持四种模式的分区： range分区， List预定义列表分区， hash分区， key键值分区。

1. 读写分离

对于高并发的数据库访问，读操作一定比写操作更多，因此可以采用数据库集群的方式，设置一个主库为写库，其他从库为读库，缓解服务器的访问压力。Mysql自带了主从复制的功能，完成读写分离。

1. 数据库集群

使用MyCat搭建Mysql集群，使用多台服务器实现写操作，由MyCat负责维护集群节点。

1. 硬件优化
2. 内存

内存的IO比硬盘快多了，可以增加系统的缓冲区容量，减少磁盘的IO。

1. 磁盘

磁盘IO是制约MYSQL性能的最大因素之一，因此使用SSD或者PCIe SSD设备，购置阵列卡，选用RAID-10，使用高转速的机械盘。

1. CPU

CPU仅能决定运算速度，即使是运算速度也要取决于与内存之间的总线带宽，以及内存本身的速度。

1. 网络

选择网络延时低，吞吐量高的设备。

1. 缓存优化
2. 查询缓存

缓存MYSQL中的ResultSet，即一条SQL语句执行的结果集，只针对select语句。在接受到一条select语句的请求后，如果该语句满足查询缓存的要求，mysql会 直接根据预设的Hash算法将select语句以字符串的方式进行Hash计算，然后将得到的Hash值与搜索结果存入缓存中。

同时，查询缓存也有一个缺陷，为了保证数据的一致性，当某个表的数据有任何变化时，都会导致所有引用了该表的select语句在查询缓存中的缓存数据失效。因此，对于数据变动十分频繁的情况，不能使用查询缓存。

1. 全局缓存

数据库是IO密集型应用，从内存中读取数据比硬盘读取要快多了，因此使用缓存是优化IO的首选。启动MYSQL时就需要分配总是存在的全局缓存，也可以在my.conf配置文件的[mysqld]中配置，查询缓存属于全局缓存。

1. 局部缓存

mysql还会为每个连接分配连接缓存，每个连接到服务器的线程都有自己的缓存，大概256k。

还有很多内容，看笔记吧！！！