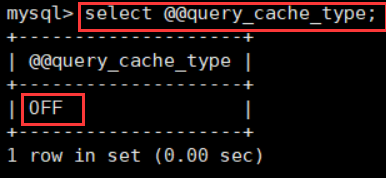
MySQL的SQL底层执行原理--图灵学院

# MySQL—Server端的缓存

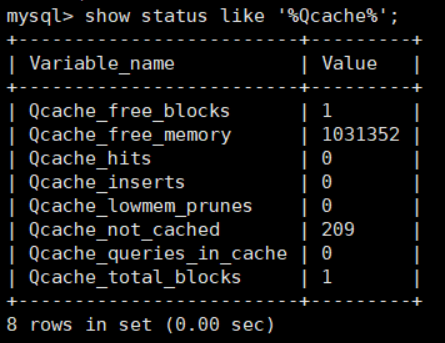
## 查看缓存是否开启

**select @@query\_cache\_type;**



## 查看缓存的状态信息

**show status like '%Qcache%';**



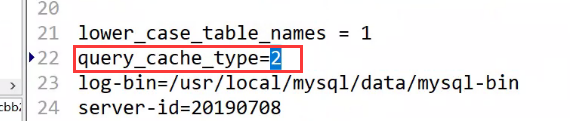
查询缓存记录数：**Qcache\_queries\_in\_cache** | 0

缓存 命中 次数：**Qcache\_hits**  | 0

## 如何开启缓存：

query\_cache\_type=0

默认为0，关闭状态；如果为1，缓存所有；如果为2，缓存设置。



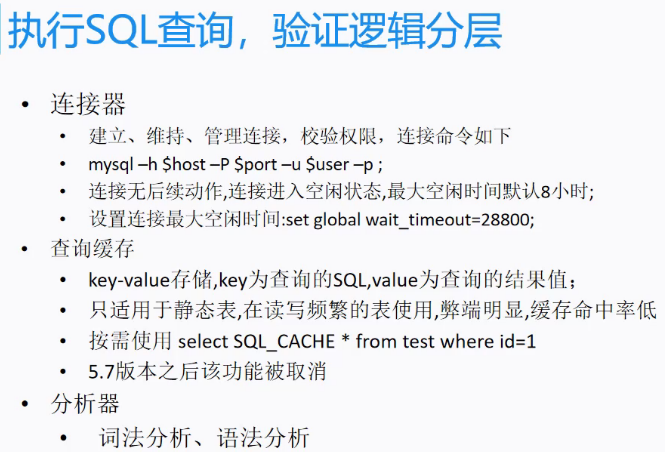
## 缺点分析及使用场景

缺点：读写频繁时，不适用。因为一旦发生写操作，缓存就失效，如果写操作比较频繁，缓存就起不到作用，缓存命中率就很低。

应用场景：字典表，不会经常改变的表，一次缓存，永久使用，大大提高速度。

在MySQL5.7版本之后，该功能被取消。

添加参数： SQL\_CACHE



# 日志模块

Redo Log 重做日志(物理日志)；

Bin Log 归档日志(逻辑日志)；

## Redo Log 重做日志(物理日志)；

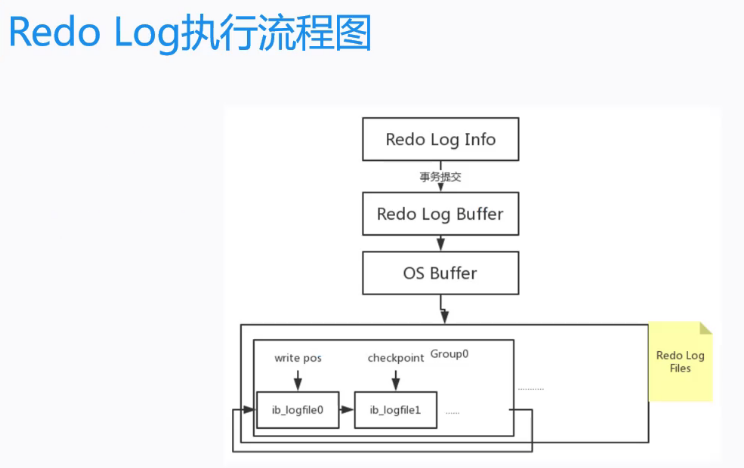


Redo Log 存在在ib\_logfile文件中。

## Redo Log执行流程图

经过两个缓存，Redo Log Buffer 和OS Buffer。

OS Buffer负责将日志写到磁盘中。





两个指针：writepos写指针；checkpoint为擦除指针。

## Redo Log参数设置



# Bin Log

见单独文件介绍。