# MySQL InnoDB引擎：

InnoDB特点：行锁设计，支持MVCC，支持外键，提供一致性非锁定读，同时尽可能高效的利用计算机硬件资源

后台线程 Master Thread:将内存数据异步刷新到磁盘，InnoDB的主要工作都是该线程完成。优先度最高。

IO Thread：处理IO请求回调

Purge Thread：在事务提交的时候回收Undo Log

缓存池是一块内存区域，在DB读取页的时候，首先从磁盘读到缓存池中。对于写操作，首先修改缓冲池中的页，再以一定的频率刷新到磁盘上。

InnoDB引擎通过B+树实现索引页，减小读取次数，有效提升速度，B+树搜索之后并不能直接找到数据，只能找到索引页，然后将该页加载入内存，通过key在其中二分查找，找到key对应的slot，在slot管理的记录中查找最终的数据。

自适应哈希索引：InnoDB存储引擎会监控对表上各索引页的查询，如果查询到建立哈希索引可以带来速度提升，则建立哈希索引

日志文件：

错误日志：错误日志记录所有错误信息和警告信息

慢查询日志：定位可能存在问题的SQL语句，可以设定一个阈值，将运行时间超过该值的SQL语句都记录到慢查询日志文件中

查询日志：记录了所有对MYSQL请求的信息，

二进制日志BinLog：BinLog记录了对MYSQL执行更改的操作（不包括select和show）的操作，主要用途有：数据恢复，集群同步，数据同步

InnoDB表存储：表都是根据主键顺序组织存放的，每张表都有一个主键，如果没有显式指定，InnoDB会使用第一个非空的唯一索引，如果没有唯一索引，InnoDB会自动创建一个6字节大小的指针做主键。所有的数据被存放在一个表空间，表空间由段，区，页组成

InnoDB实现了两种标准的行级锁：

共享锁 允许事务读一行事务

排他锁 允许事务删除或更新一行数据

意向锁：意向锁意味着事务希望在更细粒度上进行加锁。

一致性非锁定读：是指InnoDB通过多行版本控制来读取当前执行时间中的数据，如果当前数据正在执行写操作，这时读取操作不会因此阻塞，而是去读取一个快照数据。快照数据就是当前行数据对历史版本，每行记录可能有多个版本，在ReadCommitted 和repeatableread，都会使用一致性非锁定读，但是方式不同：

Read Committed下，一致性非锁定读总是读取被锁定行的最新一份快照

Repeatable Read下，一致性非锁定读总是读取事务开始时的行版本数据

事务的实现：

Redo Log 和Undo log

Redo Log重做日志用来实现事务的持久性，由内存中的重做日志缓冲和磁盘里的重做日志文件，当事务提交后必须将日志写入到磁盘中的重做日志做持久化，待事务提交结束后才算完成。

Undo log用于职务的回滚，同时用于MVCC，实现一致性非锁定读