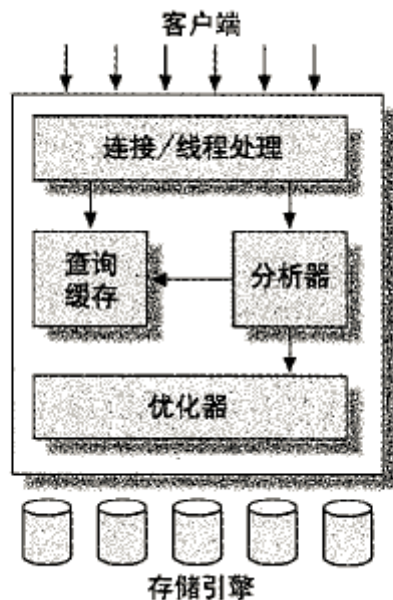


# mysql架构

存储引擎架构可以将查询处理和各类服务器任务与数据的存储/提取分离，这种性质可以基于每一张表选取存储引擎，以满足性能，特征,数据存储和其他特行的需要。



**顶层：**链接处理，授权认证，安全等待

**第二层：**大多数核心，查询解析，分析，优化，缓存还有内建函数（时间日期），引擎核心（触发器，存储过程，视图等等）

**最底层：**与操作系统进行交互

**认证方式：**主机信息+口令；安全套字层ssl，证书x.509

**优化：**mysql会解析查询语句，并创建内部存储结构（解析数），然后对其优化。

## 锁

读锁写锁，也是共享锁与排他锁

锁策略：是在锁开销和数据安全之间的平衡策略

表锁：开销最小的锁，将整个表锁起来

行级锁：支持并发说最多，但是开销最多

## 事物

可以用 `START TRANSACTION` 语句开始一个事务，用 `COMMIT` 语句提交整个事务，永久地修改数据，或者用 `ROLLBACK` 语句回滚整个事务，取消已做的修改。事务 SQL 样本如下：

## ACID

原子性 (Stomicity)

一致性 (Consisteney)

隔离性 (Isolation)

持久性 (Durability)

## 隔离级

READ UNCOMMITTED(读取未提交)

READ COMMITTED (读取已提交)

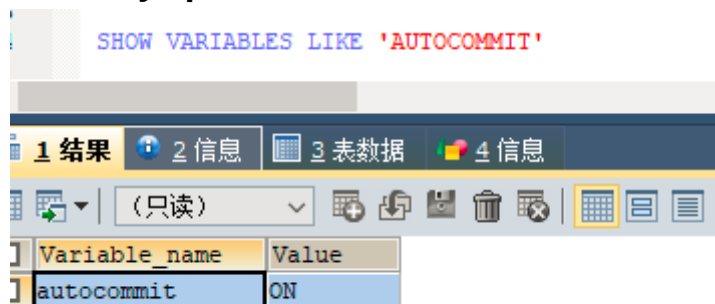
REPEATABLE READ(可重读)

SERIALIZABLE (可串行化)

**死锁：共同资源，相互竞争，导致无法获取资源**

**事物日志：**

**事物：mysql默认操作模式是AUTOCOMMIT模式，1与on，0与off效果相同**



InnoDB使用**二相锁定协议**；一个事物在任何时刻都可以获取锁，但只有在commit和rollback后才会释放锁

## 显示锁

`select ... LOCK IN SHARE MODE`

`SELECT ... FOR UPDATE`

多版本并发控制