

面试-Mysql

github: <https://github.com/Harry-Yu-Shuhang>

2025 年 1 月 18 日

1 基础篇

1.1 执行一条 select 语句，期间发生了什么？

具体见图 1.

MySQL 的架构共分为两层：Server 层和存储引擎层

Server 层负责建立连接、分析和执行 SQL,MySQL 大多数的核心功能模块都在这实现

存储引擎层负责数据的存储和提取,支持 InnoDB、MyISAM、Memory 等多个存储引擎,不同的存储引擎共用一个 Server 层。不同的存储引擎支持的索引类型也不相同,比如 InnoDB 支持索引类型是 B+ 树,且是默认使用,也就是说在数据表中创建的主键索引和二级索引默认使用的是 B+ 树索引。

1.1.1 第一步：连接器

连接的过程需要先经过 TCP 三次握手,因为 MySQL 是基于 TCP 协议进行传输的。

1.1.2 第二步：查询缓存

查询缓存是以 key-value 形式保存在内存中的,命中率低(因为只要一个表有更新操作,那么这个表的查询缓存就会被清空。),所以 8.0 已移除。

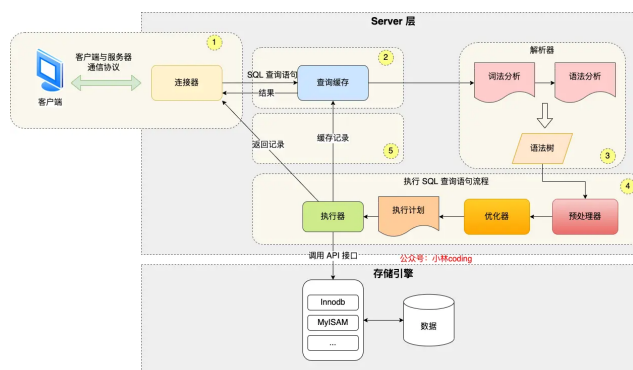


图 1: Mysql 架构 (Mysql8.0 中把查询缓存删掉了, 因为命中率很低)

1.1.3 第三步：解析 SQL

解析器会做如下两件事情。

第一件事情，词法分析。MySQL 会根据你输入的字符串识别出关键字出来，比如 select 和 from 就是关键字

第二件事情，语法分析。根据词法分析的结果，语法解析器会根据语法规则，判断你输入的这个 SQL 语句是否满足 MySQL 语法，如果没问题就会构建出 SQL 语法树。

1.1.4 第四步：执行 SQL

每条 SELECT 查询语句流程主要可以分为下面这三个阶段：

prepare 阶段，也就是预处理阶段；预处理阶段主要检查 SQL 查询语句中的表或者字段是否存在。

optimize 阶段，也就是优化阶段；优化器主要负责将 SQL 查询语句的执行方案确定下来，比如选取哪个索引。

execute 阶段，也就是执行阶段；负责执行语句。

2 事务篇

事务是由 MySQL 的引擎来实现的，我们常见的 InnoDB 引擎它是支持事务的。

事务必须要遵守 4 个特性：

原子性 (Atomicity)：一个事务中的所有操作，要么全部完成，要么全部不完成。事务在执行过程中发生错误，会被回滚到事务开始前的状态。

一致性 (Consistency)：是指事务操作前和操作后，数据满足完整性约束，数据库保持一致性状态。比如转账前后两人资金和是一样的。

隔离性 (Isolation)：数据库允许多个并发事务同时对其数据进行读写和修改的能力，隔离性可以防止多个事务并发执行时由于交叉执行而导致数据的不一致。也就是说，消费者购买商品这个事务，是不影响其他消费者购买的。

持久性 (Durability)：事务处理结束后，对数据的修改就是永久的，即便系统故障也不会丢失。

InnoDB 引擎通过什么技术来保证事务的这四个特性的呢？

- 持久性是通过 redo log（重做日志）来保证的；
- 原子性是通过 undo log（回滚日志）来保证的；
- 隔离性是通过 MVCC（多版本并发控制）或锁机制来保证的；
- 一致性则是通过持久性 + 原子性 + 隔离性来保证；

MySQL 服务端是允许多个客户端连接的，这意味着 MySQL 会出现同时处理多个事务的情况。也就是说，在同时处理多个事务的时候，就可能出现脏读 (dirty read)、不可重复读 (non-repeatable read)、幻读 (phantom read) 的问题。

- 脏读：如果一个事务「读到」了另一个「未提交事务修改过的数据」，就意味着发生了「脏读」现象。

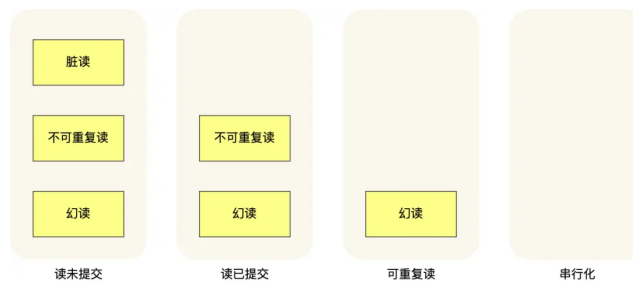


图 2: 不同事务隔离级别可能发生的现象

- 不可重复读: 在一个事务内多次读取同一个数据, 如果出现前后两次读到的数据不一样的情况, 就意味着发生了「不可重复读」现象。
- 在一个事务内多次查询某个符合查询条件的「记录数量」, 如果出现前后两次查询到的记录数量不一样的情况, 就意味着发生了「幻读」现象。

SQL 标准提出了四种隔离级别来规避这些现象, 隔离级别越高, 性能效率就越低。隔离级别从低到高如下:

- 读未提交 (read uncommitted), 指一个事务还没提交时, 它做的变更就能被其他事务看到;
- 读提交 (read committed), 指一个事务提交之后, 它做的变更才能被其他事务看到;
- 可重复读 (repeatable read), 指一个事务执行过程中看到的数据, 一直跟这个事务启动时看到的数据是一致的, **MySQL InnoDB 引擎的默认隔离级别**;
- 串行化 (serializable); 会对记录加上读写锁, 在多个事务对这条记录进行读写操作时, 如果发生了读写冲突的时候, 后访问的事务必须等前一个事务执行完成, 才能继续执行;