专业技能

熟悉 MySQL 及其索引、事务、日志、锁等机制，了解读写分离、分库分表

MySQL基础

详细描述一条 SQL 语句在 MySQL 中的执行过程。

先通过连接器校验权限

利用分析器进行 SQL 语句的词法分析和语法分析，构建解析树

使用优化器选择合适的索引和表连接顺序，最终选择一个最佳的执行计划

利用执行器，调用引擎层查询数据，返回结果集给客户端

MySQL 的存储引擎有哪些？它们之间有什么区别？

1）**InnoDB**：

* 支持事务和外键，使用行级锁。
* 提供高并发性能，适用于高负载的 OLTP 应用。
* 数据以聚集索引的方式存储，提高查询效率。

2）**MyISAM**：

* 不支持事务和外键，使用表级锁。
* 适合读取多、更新少的场景，如数据仓库。
* 具有较高的读性能和较快的表级锁定。

3）**MEMORY**：

* 数据存储在内存中，速度快，但数据在服务器重启后丢失。
* 适用于临时数据存储或快速缓存。

4）**NDB (NDBCluster)：**

* 支持高可用性和数据分布，适合大规模分布式应用
* 提供行级锁和自动分区。

5）**ARCHIVE**：

* 用于存储大量历史数据，支持高效的插入和压缩。
* 不支持索引，适合日志数据存储。

MySQL 中 InnoDB 存储引擎与 MyISAM 存储引擎的区别是什么？

如何使用 MySQL 的 EXPLAIN 语句进行查询分析？

MySQL 中如何进行 SQL 调优？

MySQL 中 count(\*)、count(1) 和 count(字段名) 有什么区别？

MySQL 中 DELETE、DROP 和 TRUNCATE 的区别是什么？

MySQL 中 varchar 和 char 有什么区别？

如何在 MySQL 中监控和优化慢 SQL？

MySQL 数据排序是怎么实现的？

MySQL分库分表

什么是分库分表？分库分表有哪些类型（或策略）？

**分库分表**是数据库性能优化的一种方法。

**分库分表的类型（或策略）** 包括：

1）**水平分表（表结构相同）**：

* 将同一张表的数据按行划分，分散到多个表中。例如，可以按用户 ID 的范围将数据分为多个表（如 user\_1、user\_2）。

2）**垂直分表（字段拆分）**：

* 将一张表的不同列拆分到多个表中，以减少每张表的字段数量和提高查询效率。例如，用户表可以分为基本信息表和详细信息表。

3）**水平分库（表结构相同）**：

* 将相同的表结构复制一份到另一个库中，每个库的表结构是一样的，可以减少单一数据库的读写压力，在大数量的情况下提高读写性能。例如，database1、 database2。

4）**垂直分库（字段拆分）**：

* 将数据分散到不同的数据库实例中。可以根据业务功能或模块进行分库，如将用户数据、订单数据分别存储在不同的数据库中。

如果组长要求你主导项目中的分库分表，大致的实施流程是？

1）**先分析业务需求**：

* 确定数据量及增长趋势，评估分库分表的必要性。（需要一定的预判但是不要过度设计）

2）**设计分库分表方案**：

* 选择适合的分库和分表策略（水平、垂直、哈希、范围等），并规划分库分表的结构。

3）**实现数据路由**：

* 根据分库分表策略设计数据路由机制，一般通过应用层代码或数据库中间件来实现，将请求路由到相应的数据库或表。

4）**数据迁移**：

* 将现有数据迁移到新的分库分表结构中，可以通过批量导入、ETL 工具或脚本来完成。

对数据库进行分库分表可能会引发哪些问题？

1）**事务问题**

我们使用关系型数据库，有很大一点在于它**保证事务的完整性**。

而分库之后单机事务就用不上了，必须使用分布式事务来解决，而分布式事务相对而言就比较重了，而且大部分的分布式事务只能保证最终一致性，所以**业务上会存在数据不一致的场景**。

2）**连表 JOIN 问题**

在一个库中的时候我们还可以利用 JOIN 来连表查询，而跨库了之后就无法使用 JOIN 了。

此时的解决方案就是在业务代码中进行关联，也就是先把一个表的数据查出来，然后通过得到的结果再去查另一张表，然后利用代码来关联得到最终的结果。

这种方式实现起来稍微比较复杂，不过也是可以接受的。

还有可以适当的冗余一些字段。比如以前的表就存储一个关联 ID，但是业务时常要求返回对应的 Name 或者其他字段。这时候就可以把这些字段冗余到当前表中，来去除需要关联的操作。

或者**通过宽表的形式查询**，比如将数据全量存储至 ES 中，利用 ES 来查询数据。

3）**全局 ID 唯一性问题**

以前单库单表直接使用数据库的自增 ID 即可，但是分库分表之后，使用自增 ID 会导致重复主键的情况，此时需要利用雪花算法或者其他全局唯一 ID 发号器来生成唯一主键。

4）**排序问题**

单表直接通过 order by 进行排序即可，分库分表后直接利用数据库是无法实现排序的。要么利用分库分表中间件的能力进行汇总排序，要么自己在业务代码中排序，要么利用 ES 存储全量数据排序查询。

5）**count 问题**

其实和排序问题类似，单表可以直接 count，分库分表后无法支持，只能多表 count 然后业务代码中累加，或者单独搞一个地方来维护总数，要么还是利用 ES。

MySQL读写分离

如何在 MySQL 中实现读写分离？

**做法一：代码封装**

利用代理类，对外暴露正常的读写接口，里面封装了逻辑，将读操作指向从库的数据源，写操作指向主库的数据源。

缺点：如果数据库宕机了，发生主从切换了之后，就得修改配置重启。如果系统是多语言的话，需要为每个语言都实现一个中间层代码，重复开发。

**做法二：使用中间件**

中间件一般而言是独立部署的系统，客户端与这个中间件的交互是通过 SQL 协议的。所以在客户端看来连接的就是一个数据库，通过 SQL 协议交互也可以屏蔽多语言的差异。常见的开源数据库中间件有： MySQL-Proxy、Atlas、ShardingSphere、Mycat 等。

缺点：整体架构多了一个系统需要维护，并且可能成为性能瓶颈，毕竟交互都需要经过它中转。

什么是 MySQL 的主从同步机制？它是如何实现的？

如何处理 MySQL 的主从同步延迟？

如何在 MySQL 中避免单点故障？

MySQL索引

MySQL 的索引类型有哪些？

MySQL InnoDB 引擎中的聚簇索引和非聚簇索引有什么区别？

请详细描述 MySQL 的 B+ 树中查询数据的全过程。

为什么 MySQL 选择使用 B+ 树作为索引结构？

MySQL 中的回表是什么？

MySQL 索引的最左前缀匹配原则是什么？

MySQL 的覆盖索引是什么？

MySQL 的索引下推是什么？

在什么情况下，不推荐为数据库建立索引？

在 MySQL 中建索引时需要注意哪些事项？

MySQL 中使用索引一定有效吗？如何排查索引效果？

MySQL 中的索引数量是否越多越好？为什么？

MySQL 三层 B+ 树能存多少数据？

MySQL事务

MySQL 中的事务隔离级别有哪些？MySQL 默认的事务隔离级别是什么？为什么选择这个级别？

数据库的脏读、不可重复读和幻读分别是什么？

MySQL 是如何实现事务的？

MySQL 中的 MVCC 是什么？

如果 MySQL 中没有 MVCC，会有什么影响？

MySQL 事务的二阶段提交是什么？

MySQL 中长事务可能会导致哪些问题？

你们生产环境的 MySQL 中使用了什么事务隔离级别？为什么？

MySQL日志

什么是 Write-Ahead Logging (WAL) 技术？它的优点是什么？MySQL 中是否用到了 WAL？

MySQL 插入一条 SQL 语句，redo log 记录的是什么？

MySQL 的 Change Buffer 是什么？它有什么作用？

MySQL 的 Doublewrite Buffer 是什么？它有什么作用？

MySQL 中的 Log Buffer 是什么？它有什么作用？

MySQL锁

MySQL 中有哪些锁类型？

MySQL 中如果发生死锁应该如何解决？

MySQL 的乐观锁和悲观锁是什么？