**mysql大事务带来的问题**

**一、什么是大事务**

* 定义：运行时间比较长，操作的数据比较多的事务。
* 大事务风险：
  1. 锁定太多的数据，造成大量的阻塞和锁超时，回滚所需要的时间比较长。
  2. 执行时间长，容易造成主从延迟。
* 如何处理大事务：
  1. 避免一次处理太多大数据。
  2. 移出不必要在事务中的select操作
  3. 事务尽可能的短，事务只做单纯的数据库存储，修改逻辑，调用第三方及其他业务放在事务外处理。

**二、什么是事务**

* 1.原子性：一个事务必须被视为一个不可分割的最小工作单元整个事务的所有操作要么全部提交成功，要么全部失败，对于一个事务来说不可能只执行其中一部分。例如：银行卡转账（整个事务中的所有操作要么全部提交成功要么全部失败）。
* 2.事务的一致性：一致性是指事务将数据库从一种一致性状态转换到另外一种一致性状态，在事务开始之前和事务结束后数据库中数据的完整性没有被破坏。例如：银行卡转账（一张减去100元，一张增加100元数据没有变化）。
* 3.事务的隔离性：一个事务对数据库中数据的修改，在未提交完成前对于其他事务不可见的。
  1. 读未提交（read-uncommitted）：就是一个事务可以读取另一个未提交事务的数据脏读（脏读）
  2. 读已提交（read-committed）：就是一个事务要等另一个事务提交后才能读取数据（不可重复读）
  3. 可重复读（repeatable-read）：就是在开始读取数据（事务开启）时，不再允许修改操作
  4. 可串行化（serializable）：是最高的事务隔离级别，在该级别下，事务串行化顺序执行，可以避免脏读、不可重复读与幻读。但是这种事务隔离级别效率低下，比较耗数据库性能，一般不使用。
  5. 分析：若有事务对数据进行更新（UPDATE）操作时，读操作事务要等待这个更新操作事务提交后才能读取数据，可以解决脏读问题。但是会出现了一个事务范围内两个相同的查询却返回了不同数据，这就是不可重复读。
  6. 注意：
     1. 并发性由高到低1>2>3>4
     2. 隔离性由高到低4>3>2>1
     3. show variables like '%iso%'(查看当前隔离性级别)
     4. set session tx\_isolation （设置隔离级别）
* 4.事务的持久性：一旦事务提交，则其所做的修改就会永久的保存到数据库中，此时即使系统崩溃，已经提交的修改数据也不会丢失。

**三、大表带来的问题**

1. 大表定义：单表超过千万行、文件超过10G
2. 大表对查询的影响：
   * 慢查询、
   * 区分度底、
   * 大量磁盘IO、
   * 建立索引需要很长的时间、
   * 修改表结构需要长时间锁表、
   * 影响正常的数据操作
3. 如何处理大表问题：
   * 分库分表把一张表分成多个小表。难点：分表主键的选择、分表后夸分区数据的查询和统计。
   * 大表的历史数据归档（前端增加历史查询）难点：时间点选择，如何进行归档操作

**四、在大促中数据库服务器**

1. 数据库架构：主从复制、读写分离、集群等。
2. TPS:每秒处理事务的速度（一个事务三个过程）。
   1. 用户请求服务器
   2. 服务器内部处理
   3. 服务器返回给用户。
3. QPS:是一台指定服务器每秒能够相应的查询次数。
4. 并发量：同一时间处理的请求的数量。
5. 连接数：和服务器进行连接，但大部分处于sleep状态，只有少部分在运行。
6. 并发量大，连接数大说明cpu空闲少繁忙。
7. 磁盘IO读写过高会对服务器性能能造成影响。
8. 不要在主库上数据库备份（磁盘读压力增大）。