Mysql事务的关键性技术：日志文件(redo log和undo log)、锁技术和MVCC

事务：是访问和更新数据库的程序执行单元；事务中包含多个sql语句，要么都执行，要么都不执行。

1. mysql中事务是由存储引擎实现的，mysql中支持事务的存储引擎有innodb和ndb cluster，其他存储引擎不支持事务。
2. mysql默认采用自动提交模式，通过autocommit控制，autocommit参数是针对连接的，在一个连接中修改该参数，不会对其他连接产生影响。
3. mysql中ddl语句(create table/drop table/alter table)、lock table会强制执行commit提交事务。
4. ndb cluster事务不满足持久性和隔离性；innodb默认事务隔离级别是可重复读，不满足隔离性
5. 原子性的实现原理（要么都做，要么都不做）
6. undo log(回滚日志)：原子性和隔离性实现的基础

undo log属于逻辑日志，记录sql执行相关的信息，当发生回滚时，innodb根据日志中的内容做与之前相反的操作，对insert做delete；对delete做insert；对update执行相反的update，把数据改回去；

1. 持久性实现原理(一旦提交，对数据库的改变就是永久的)
2. innodb使用buffer pool作为读写缓存，当写数据时，先写入buffer pool中，buffer pool中修改的数据会定期刷新到磁盘中，但是当mysql宕机时，buffer pool中的数据还没有刷新到磁盘，导致数据丢失，事务的持久性无法保证。Mysql使用redo log解决这个问题，当数据修改时，除了修改buffer pool中的数据，还会在redo log中记录这次操作；当事务提交时，会调用fsync接口对redo log进行刷盘。如果mysql宕机，重启时可以读取redo log中的数据，对数据库进行恢复。Redo log是先写入日志，再更新到buffer pool，保证数据不会丢失。
3. 刷脏是随机io，每次修改数据的位置随机，但redo log是追加操作，属于顺序io；刷脏是以page为单位，一个page上的小修改都要整页写入，但是redo log中只包含真正需要写入的部分，减少了无效io
4. 隔离性实现原理（事务内的操作与其他事务隔离）
5. 写操作对写操作的影响：锁机制保证
6. 写操作对读操作的影响：MVCC保证
7. 锁机制

同一时刻只能有一个事务对数据进行写操作。事务修改数据前，需要先获得锁，该事务操作期间，这部分数据是锁定的，其他事务如果需要修改数据，需要等待当前事务提交或回滚后释放锁。按照粒度可分为行锁和表锁。表锁在操作数据时会锁定整张表，并发性能较差；行锁则只锁定需要操作的数据。Myisam只支持表锁，innodb同时支持表锁和行锁

1. 脏读：当前事务中可以读到其他事务未提交的数据(脏数据)
2. 不可重复读：在事务A中先后读取同一个数据，两次读取的结果不一样
3. 幻读：在事务A中按照某个条件先后查询数据库，两次查询结果的条数不同。

4、SQL标准中定义了四种隔离级别：读未提交，读已提交，可重复读，可串行化



Innodb的默认隔离级别是可重复读，需要注意的是，在SQL标准中，RR是无法避免幻读问题的，但是InnoDB实现的RR避免了幻读问题。

四、MVCC

1、可重复读解决脏读、不可重复读、幻读等问题，使用的是MVCC：MVCC全称Multi-Version Concurrency Control，即多版本的并发控制协议。同一时刻，不同的事务读取到的数据可能是不同的

<https://www.cnblogs.com/kismetv/p/10331633.html>

锁技术：

当多个请求中既有读又有写时，对并发请求进行控制。

1. 共享锁(shared lock)—读锁

多个读请求共享一把锁读数据

1. 排他锁(exclusive lock)—写锁