1. 事务的概念是什么?事务的特性是什么？

事务的概念：保证一组数据库操作，要么全部成功，要么全部失败。

事务的特性：原子性、一致性、隔离性、持久性。

2. 多事务同时执行的时候，可能会出现的问题：脏读、不可重复读、幻读。

脏读：事务1执行修改操作，还未提交，事务2读到事务1的修改后的值进行后续操作，此时事务1进行回滚，事务2读到的其实是脏数据。

不可重复读：事务1读表t的所有数据后，事务2修改了表t中的若干行数据并提交，事务1再次读表t的数据，发现和第一次读的时候不一致。

幻读：事务1读表t的所有数据后，事务2向表t中插入了若干行数据并提交，事务1再次读表t的数据，发现和第一次读的时候相比多了若干行。

3. mysql的事务隔离级别读未提交, 读已提交, 可重复读, 串行各是什么意思?

读未提交：一个事务还未提交，它所做的变更就可以被别的事务看到

读提交：一个事务提交之后，它所做的变更才可以被别的事务看到

可重复读：一个事务执行过程中看到的数据和该事务启动时看到的数据是一致的。

串行化：对应一个记录会加读写锁，出现冲突的时候，后访问的事务必须等前一个事务执行完成才能继续执行

4. 读已提交, 可重复读是怎么通过视图构建实现的?  
5. 可重复读的使用场景举例? 对账的时候应该很有用?  
6. 事务隔离是怎么通过read-view(读视图)实现的?  
7. 并发版本控制(MCVV)的概念是什么, 是怎么实现的?  
8. 使用长事务的弊病? 为什么使用常事务可能拖垮整个库?  
9. 事务的启动方式有哪几种?   
10. commit work and chain的语法是做什么用的?   
11. 怎么查询各个表中的长事务?  
12. 如何避免长事务的出现?

1、事务的特性：原子性、一致性、隔离性、持久性

2、多事务同时执行的时候，可能会出现的问题：脏读、不可重复读、幻读

3、事务隔离级别：读未提交、读提交、可重复读、串行化

4、不同事务隔离级别的区别：

读未提交：一个事务还未提交，它所做的变更就可以被别的事务看到

读提交：一个事务提交之后，它所做的变更才可以被别的事务看到

可重复读：一个事务执行过程中看到的数据是一致的。未提交的更改对其他事务是不可见的

串行化：对应一个记录会加读写锁，出现冲突的时候，后访问的事务必须等前一个事务执行完成才能继续执行

5、配置方法：启动参数transaction-isolation

6、事务隔离的实现：每条记录在更新的时候都会同时记录一条回滚操作。同一条记录在系统中可以存在多个版本，这就是数据库的多版本并发控制（MVCC）。

7、回滚日志什么时候删除？系统会判断当没有事务需要用到这些回滚日志的时候，回滚日志会被删除。

8、什么时候不需要了？当系统里么有比这个回滚日志更早的read-view的时候。

9、为什么尽量不要使用长事务。长事务意味着系统里面会存在很老的事务视图，在这个事务提交之前，回滚记录都要保留，这会导致大量占用存储空间。除此之外，长事务还占用锁资源，可能会拖垮库。

10、事务启动方式：一、显式启动事务语句，begin或者start transaction,提交commit，回滚rollback；二、set autocommit=0，该命令会把这个线程的自动提交关掉。这样只要执行一个select语句，事务就启动，并不会自动提交，直到主动执行commit或rollback或断开连接。

11、建议使用方法一，如果考虑多一次交互问题，可以使用commit work and chain语法。在autocommit=1的情况下用begin显式启动事务，如果执行commit则提交事务。如果执行commit work and chain则提交事务并自动启动下一个事务。