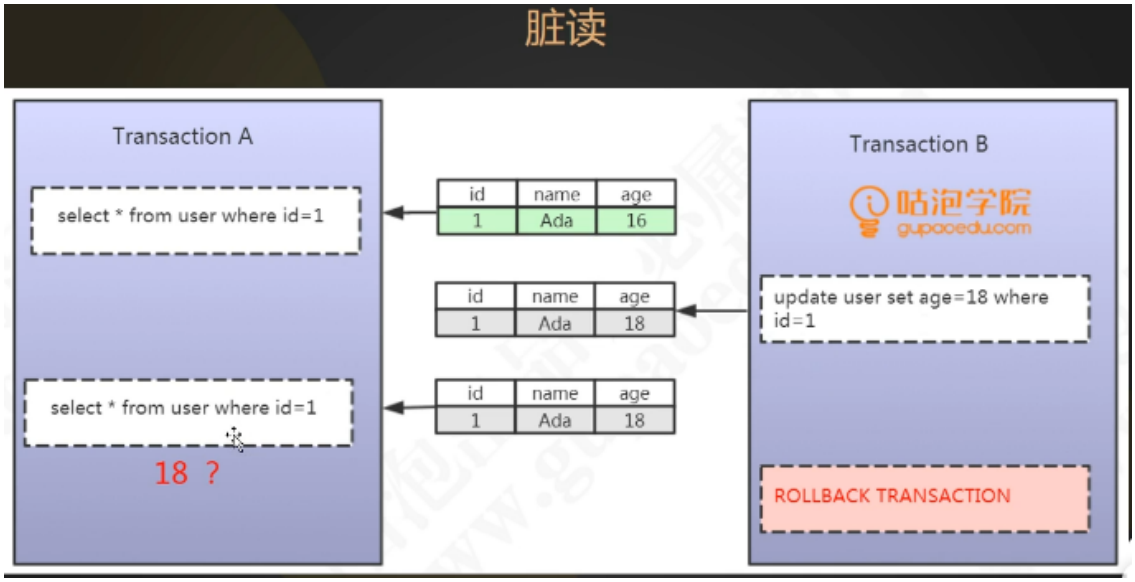
Mysql 事务

什么叫事务：

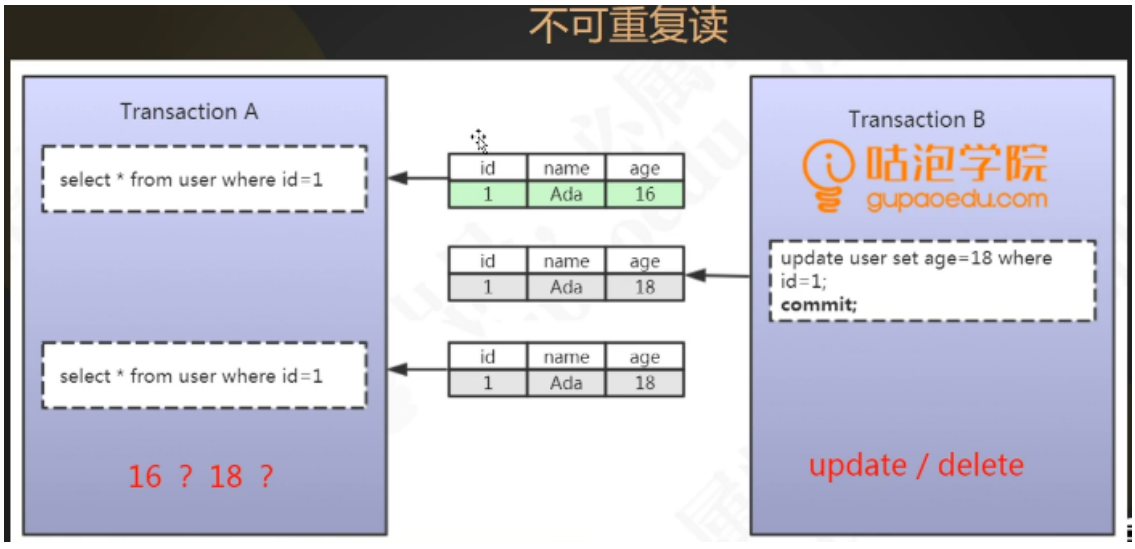
- 1.数据库事务可以包含**一个或多个**数据库操作,但这些操作构成一个**逻辑上**的整体。
- 2.构成逻辑整体的这些数据库操作,要么全部执行成功,要么全部不执行。（原子性）
- 3.构成事务的所有操作,要么全都对数据库产生影响,要么全都不产生影响,即不管事务是否执行成功,数据库总能保持一致性状态。事务的执行结果必须使数据库从一个一致性状态到另一个一致性状态。（一致性）
- 4.以上即使在数据库出现故障（持久性）以及并发事务存在的情况下（隔离性）依然成立。

事务并发的问题：

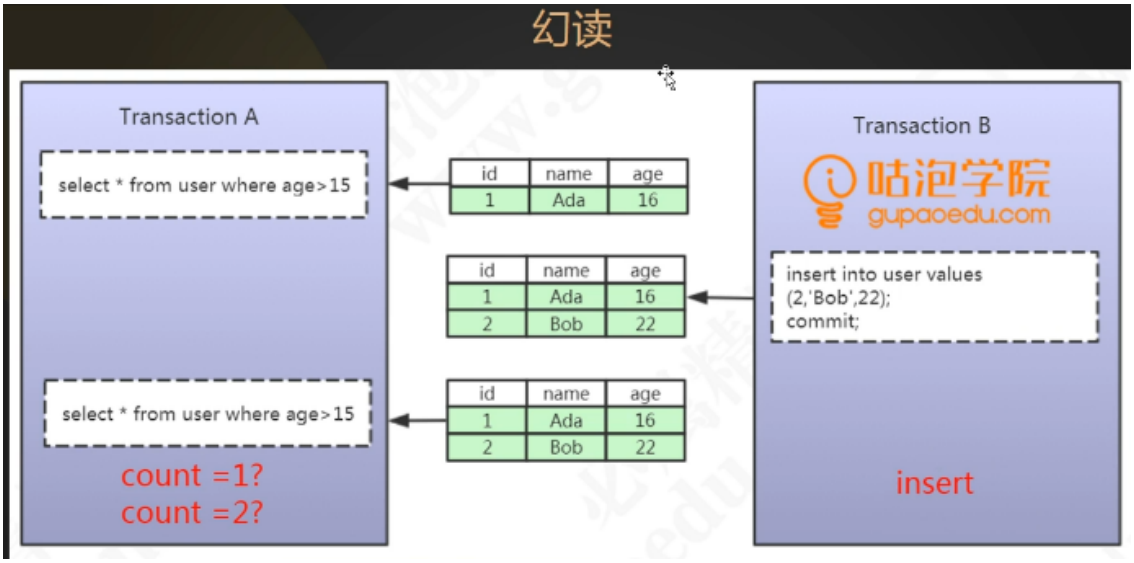
1. 脏读：事务A读取到了并修改了事务B未提交的数据，导致事务一致性被破坏



2. 不可重复读：因为事务B对数据做了修改或删除，导致事务A的两次查询结果不一致。



3. 幻读：因为事务B对数据做了插入，导致事务A的两次查询结果不一致。



4. 脏写：事务回滚了其他事务对数据项的已提交修改

事务1	事务2
read(A)=10	
	read(A)=10
	write(A)=30
	Commit
write(A)=20	
Rollback(A=10)	

5. 丢失更新：事务覆盖了其他事务对数据的已提交修改,导致这些修改好像丢失了一样。

事务1	事务2
	Read(A)=10
Read(A)=10	
	A:=A+10
	Commit
A:=A-10	
Commit(A=0)	

事务隔离级别

SQL标准为事务定义了不同的隔离级别,从低到高依次是

- **读未提交(READ UNCOMMITTED)**

  如果一个事务读取到了另一个未提交事务修改过的数据，那么这种隔离级别就称之为读未提交。

- **读已提交(READ COMMITTED)**

  如果一个事务只能读取到另一个已提交事务修改过的数据，并且其它事务每对该数据进行一次修改并提交后，该事务都能查询得到最新值，那么这种隔离级别就称之为读提交。

  该隔离级别满足了隔离的简单定义：一个事务从开始到提交前所做的任何改变都是不可见的，事务只能读取到已经提交的事务所做的改变。

- **可重复读(REPEATABLE READ)**

  http://c.biancheng.net/view/7265.html

  https://www.cnblogs.com/myseries/p/10931595.html

  这里涉及到MVCC的知识，InnoDB在每行记录后面保存两个隐藏的列来，分别保存了这个行的创建时间和行的删除时间。这里存储的并不是实际的时间值,而是系统版本号，当数据被修改时，版本号加1

  - 在读取事务开始时，系统会给当前读事务一个版本号，事务会读取版本号<=当前版本号的数据
  - 此时如果其他写事务修改了这条数据，那么这条数据的版本号就会加1，从而比当前读事务的版本号高，读事务自然而然的就读不到更新后的数据了

  总之：因为MySQL的可重复读，对事务B进行查询时，事务A提交的更新不会影响到事务B。但是对事务B进行更新时，事务A提交的更新会影响到事务B。

- **串行化(SERIALIZABLE)**

  如果一个事务先根据某些条件查询出一些记录，之后另一个事务又向表中插入了符合这些条件的记录，原先的事务再次按照该条件查询时，能把另一个事务插入的记录也读出来。那么这种隔离级别就称之为串行化。

  SERIALIZABLE 是最高的事务隔离级别，主要通过强制事务排序来解决幻读问题。简单来说，就是在每个读取的数据行上加上共享锁实现，这样就避免了脏读、不可重复读和幻读等问题。但是该事务隔离级别执行效率低下，且性能开销也最大，所以一般情况下不推荐使用。

事务的隔离级别越低,可能出现的并发异常越多,但是通常而言系统能提供的并发能力越强。

不同的隔离级别与可能的并发异常的对应情况如下表所示,有一点需要强调,这种对应关系只是理论上的,对于特定的数据库实现不一定准确,比如mysql的Innodb存储引擎通过Next-Key Locking技术在**可重复读级别就消除了幻读**的可能。

	可能导致的并发异常				
事务的隔离级别	**脏写**	**脏读**	**不可重复读**	**幻读**	**丢失更新**
读未提交(READ UNCOMMITTED)		可能	可能	可能	可能
读已提交(READ COMMITTED)			可能	可能	可能
可重复读(REPEATABLE READ)				可能	
串行化(SERIALIZABLE)					