**[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "_blank)事务的4个特性：  
原子性(atomic):** 都成功或者都失败；  
**一致性(consistency):**事务操作之后,数据库所处的状态和业务规则是一致的;比如a,b账户相互转账之后，总金额不变；  
**隔离性(isolation):**操作中的事务不相互影响;  
**持久性(durability):**事务提交后被持久化到数据库.

**脏读，不可重复读， 幻读  
幻读phantom read：**事务1读取记录时事务2增加了记录并提交，事务1再次读取时可以看到事务2新增的记录；  
**不可重复读unrepeatable read：**事务1读取记录时，事务2更新了记录并提交，事务1再次读取时可以看到事务2修改后的记录；  
**脏读dirty read：**事务1更新了记录，但没有提交，事务2读取了更新后的行，然后事务T1回滚，现在T2读取无效。

**事务隔离级别描述：**READ UNCOMMITTED：幻读，不可重复读和脏读均允许；  
READ COMMITTED：允许幻读和不可重复读，但不允许脏读；  
REPEATABLE READ：允许幻读，但不允许不可重复读和脏读；  
SERIALIZABLE:幻读，不可重复读和脏读都不允许；   
[**Oracle**](http://lib.csdn.net/base/oracle)默认的是 READ COMMITTED。

**如何感性理解这些“深奥”的术语？个人认为， 作为一个**[**Java**](http://lib.csdn.net/base/javase)**程序员，这些字眼比较唬人。**

**首先， 隔离级别描述了事务被隔离的程度。可以简单的想象一下， 一个人因为某种原因（比如国家安全）需要被隔离，则涉及安全的机密程度越高， 需要隔离级别就越高，他与外界的沟通渠道就越少。**

**对于数据库而言， 每个事务都要占用一些资源，比如对表/数据享有的操作权限，事务的隔离级别描述了事务对资源享用的程度。**

如果数据库的隔离级别为REAE\_UNCOMMITTED， 则其他线程可以看到未提交的数据， 因此就出现脏读；

如果数据库隔离级别设为READ\_COMMITTED，即没提交的数据别人是看不见的，就避免了脏读；但是，正在读取的数据只获得了读取锁，读完之后就解锁，不管当前事务有没有结束，这样就容许其他事务修改本事务正在读取的数据。导致不可重复读。

REPEATABLE READ因为对正在操作的数据加锁，并且只有等到事务结束才放开锁， 则可以避免不可重复读；  
REPEATABLE READ只能保证正在被本事务操作的数据不被其他事务修改，却无法保证有其他事务提交新的数据。 则有可能线程1在操作表T1的时候（特别是统计性的事务），其他线程仍然可以提交新数据到表T1，这样会导致线程1两次统计的结果不一致，就像发生幻觉一样。

SERIALIZABLE因为获得范围锁，且事务是一个接着一个串行执行，则保证了不会发生幻读。  
由此可见，隔离级别越高，受其他事物干扰越少，并发性能越差。

设置语法

Oracle:

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE|READ COMMITTED|READ UNCOMMITTED|REPEATABLE READ；

Sybase:

sql opreation at isolation read committed

 脏读就是指当一个事务正在访问数据，并且对数据进行了修改，而这种修改还没有提交到数据库中，这时，另外一个事务也访问这个数据，然后使用了这个数据。因为这个数据是还没有提交的数据，那么另外一个事务读到的这个数据是脏数据，依据脏数据所做的操作可能是不正确的。  
  
          不可重复读是指在一个事务内，多次读同一数据。在这个事务还没有结束时，另外一个事务也访问该同一数据。那么，在第一个事务中的两次读数据之间，由于第二个事务的修改，那么第一个事务两次读到的数据可能是不一样的。这样就发生了在一个事务内两次读到的数据是不一样的，因此称为是不可重复读。  
  
         幻觉读是指当事务不是独立执行时发生的一种现象，例如第一个事务对一个表中的数据进行了修改，这种修改涉及到表中的全部数据行。同时，第二个事务也修改这个表中的数据，这种修改是向表中插入一行新数据。那么，以后就会发生操作第一个事务的用户发现表中还有没有修改的数据行，就好象发生了幻觉一样。