**事务**：**事务内的语句要么全部执行，要么全部执行失败**

**事务必须满足4个条件（ACID）：**

* **原子性：事务是最小工作单元，一个事务（transaction）中的所有操作，要么全部完成，要么全部失败回滚**
* **一致性：数据库总是从一个一致性的状态转换到另外一个一致性的状态**
* **隔离性：事务的修改在提交前，其他事务是不可见**
* **持久性：事务处理结束后，对数据的修改永久保存在数据库中**

**并发问题**

* **丢失更新**

**在没有事务隔离的情况下，两个事务都同时更新一行数据，但是第二个事务却中途失败退出， 导致对数据的两个修改都失效了。**

     第一类：  张三的工资为5000，事务A中获取工资为5000，事务B获取工资为5000，汇入100，并提交数据库，工资变为5100，

       随后

       事务A发生异常，回滚了，恢复张三的工资为5000，这样就导致事务B的更新丢失了。

**有两个并发事务同时读取同一行数据，然后其中一个对它进行修改提交，而另一个也进行了修改提交。这就会造成第一次写操作失效。**

第二类：A事务覆盖B事务已经提交的数据，造成B事务所做操作丢失：



相反如果转账事务先提交，那么用户账户将损失100元。

* **脏读**

**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**

* **不可重复读**

**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**

在事务A中，读取到张三的工资为5000，操作没有完成，事务还没提交。  
　　与此同时，  
　　事务B把张三的工资改为8000，并提交了事务。  
　　随后，  
　　在事务A中，再次读取张三的工资，此时工资变为8000

**不可重复读侧重于修改，幻读侧重于新增或删除**

* **幻读**

**第一个事务对一个表中的数据进行了修改，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。操作第一个事务的用户再次查看时会发现表中还有没有修改的数据行**

目前工资为5000的员工有10人，事务A读取所有工资为5000的人数为10人。  
　　此时，  
　　事务B插入一条工资也为5000的记录。  
　　这是，事务A再次读取工资为5000的员工，记录为11人。此时产生了幻读





**四种隔离级别**

**Read Uncommitted（读取未提交内容）**

**在该隔离级别，所有事务都可以看到其他未提交事务的执行结果（脏读）**

**Read Committed（读取提交内容）（大多数默认）**

**满足了隔离的简单定义：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）**

程序员拿着信用卡去享受生活（卡里当然是只有3.6万），当他埋单时（程序员事务开启），收费系统事先检测到他的卡里有3.6万，就在这个时候！！程序员的妻子要把钱全部转出充当家用，并提交。当收费系统准备扣款时，再检测卡里的金额，发现已经没钱了（第二次检测金额当然要等待妻子转出金额事务提交完）。

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

**Repeatable Read（可重读）**

**这是MySQL的默认事务隔离级别，就是在开始读取数据（事务开启）时，不再允许修改操作**  
       程序员拿着信用卡去享受生活（卡里当然是只有3.6万），当他埋单时（事务开启，不允许其他事务的UPDATE修改操作），收费系统事先检测到他的卡里有3.6万。这个时候他的妻子不能转出金额了。接下来收费系统就可以扣款了

**Serializable 序列化**

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

