脏读dirty   reads：当事务读取还未被提交的数据时，就会发生这种事件。举例来说：Transaction   1   修改了一行数据，然后   Transaction   2   在   Transaction   1   还未提交修改操作之前读取了被修改的行。如果   Transaction   1   回滚了修改操作，那么   Transaction   2   读取的数据就可以看作是从未存在过的。       
      
  不可重复的读non-repeatable   reads：当事务两次读取同一行数据，但每次得到的数据都不一样时，就会发生这种事件。举例来说：Transaction   1   读取一行数据，然后   Transaction   2   修改或删除该行并提交修改操作。当   Transaction   1   试图重新读取该行时，它就会得到不同的数据值（如果该行被更新）或发现该行不再存在（如果该行被删除）。       
      
  虚读phantom   read：如果符合搜索条件的一行数据在后面的读取操作中出现，但该行数据却不属于最初的数据，就会发生这种事件。举例来说：Transaction   1   读取满足某种搜索条件的一些行，然后   Transaction   2   插入了符合   Transaction   1   的搜索条件的一个新行。如果   Transaction   1   重新执行产生原来那些行的查询，就会得到不同的行。

数据库带来的并发问题包括：

 　　1.丢失或覆盖更新。(幻像读)

 　　2.未确认的相关性（脏读）。

 　　3.不一致的分析（非重复读）。

详细描述如下：

## 一.丢失更新

当两个或多个事务选择同一行，然后基于最初选定的值更新该行时，会发生丢失更新问题。每个事务都不知道其它事务的存在。最后的更新将重写由其它事务所做的更新，这将导致数据丢失。

e.g.事务A和事务B同时修改某行的值，

 1.事务A将数值改为1并提交

       2.事务B将数值改为2并提交。

这时数据的值为2，事务A所做的更新将会丢失。

解决办法：对行加锁，只允许并发一个更新事务。

## 二.未确认的相关性（脏读）

　　当第二个事务选择其它事务正在更新的行时，会发生未确认的相关性问题。第二个事务正在读取的数据还没有确认并且可能由更新此行的事务所更改。

      e.g.

        1.Mary的原工资为1000, 财务人员将Mary的工资改为了8000(但未提交事务)        2.Mary读取自己的工资 ,发现自己的工资变为了8000，欢天喜地！

     3.而财务发现操作有误，回滚了事务,Mary的工资又变为了1000

      像这样,Mary记取的工资数8000是一个脏数据。

 解决办法：如果在第一个事务提交前，任何其他事务不可读取其修改过的值，则可  以避免该问题。

## 三.不一致的分析（非重复读）

　　当第二个事务多次访问同一行而且每次读取不同的数据时，会发生不一致的分析问题。不一致的分析与未确认的相关性类似，因为其它事务也是正在更改第二个事务正在读取的数据。然而，在不一致的分析中，第二个事务读取的数据是由已进行了更改的事务提交的。而且，不一致的分析涉及多次（两次或更多）读取同一行，而且每次信息都由其它事务更改；因而该行被非重复读取。

      在一个事务中前后两次读取的结果并不致，导致了不可重复读。

e.g.

1.在事务1中，Mary 读取了自己的工资为1000,操作并没有完成

2.在事务2中，这时财务人员修改了Mary的工资为2000,并提交了事务.

3.在事务1中，Mary 再次读取自己的工资时，工资变为了2000

解决办法：如果只有在修改事务完全提交之后才可以读取数据，则可以避免该问题。

**四.幻像读**

 　当对某行执行插入或删除操作，而该行属于某个事务正在读取的行的范围时，会发生幻像读问题。事务第一次读的行范围显示出其中一行已不复存在于第二次读或后续读中，因为该行已被其它事务删除。同样，由于其它事务的插入操作，事务的第二次或后续读显示有一行已不存在于原始读中。

 　　e.g. 目前工资为1000的员工有10人。

1.事务1,读取所有工资为1000的员工。

2.这时事务2向employee表插入了一条员工记录，工资也为1000

3.事务1再次读取所有工资为1000的员工 共读取到了11条记录，

解决办法：如果在操作事务完成数据处理之前，任何其他事务都不可以添加新数据，则可避免该问题