**事务隔离级别**

# What??

Mysql数据库：默认级别是**repeatable**

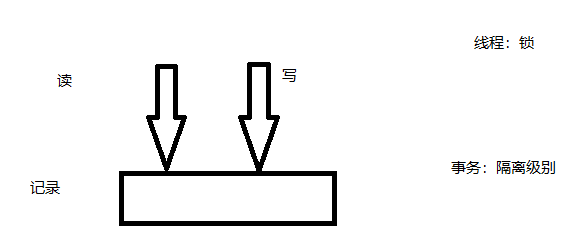


按照如上顺序：数据的安全性会越来越强，但是并发性会越来越低。

**根据具体的业务并发量，来灵活的设置这个隔离级别。**

并发量特别大，全是读，可以设置读未提交（这个不现实，因为一般没这样的业务），具体问题会从业务层再去解决并发的问题。

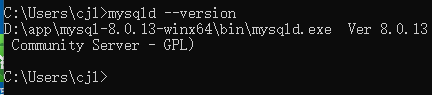
# Why??



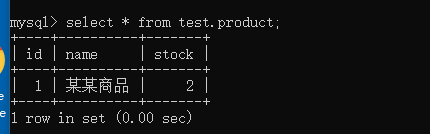
# Deep think

## 演示环境

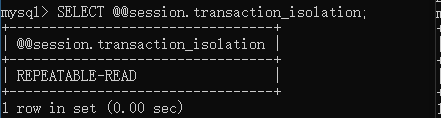
**Mysql版本是： 8**



初始数据：



隔离级别：



## 背景知识

登录：

mysql -uroot -p111111

查询数据库：

**show** **databases**;

切换数据库：

**use** test;

查询记录：

**select** \* **from** test.product;

事务执行语句：

**update** test.product **set** stock = stock-1 **where** id=1;

**INSERT** **INTO** `test`.`product` (`name`, `stock`) **VALUES** ('yy', '1');

恢复环境：

**update** test.product **set** stock = 2 **where** id=1;

**DELETE** **FROM** `test`.`product` **WHERE** name = 'yy';

隔离级别设置：

**SELECT** @@**session**.transaction\_isolation;

**SET** **session** **TRANSACTION** **ISOLATION** **LEVEL** READ UNCOMMITTED;

**SET** **session** **TRANSACTION** **ISOLATION** **LEVEL** READ COMMITTED;

**SET** **session** **TRANSACTION** **ISOLATION** **LEVEL** REPEATABLE READ;

**SET** **session** **TRANSACTION** **ISOLATION** **LEVEL** SERIALIZABLE;

事务控制：

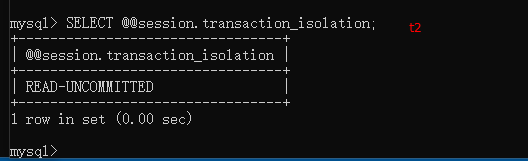
**start** **transaction**;

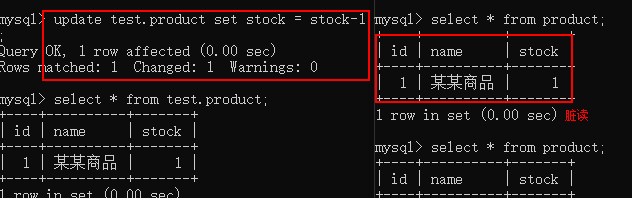
**commit**;

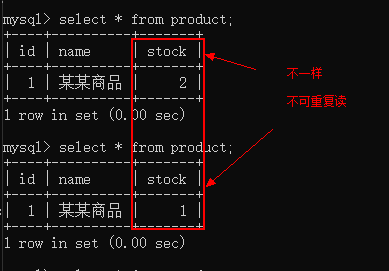
**rollback**;

## 读未提交

READ UNCOMMITTED;









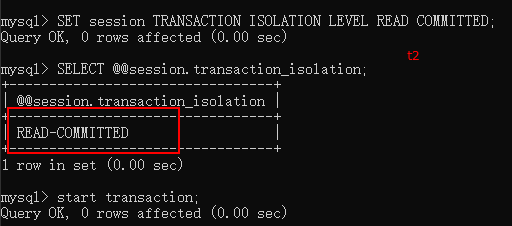
## 读已提交

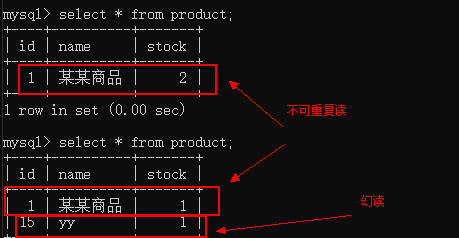
READ COMMITTED

第一步：恢复环境，最初的数据

**update** test.product **set** stock = 2 **where** id=1;

**DELETE** **FROM** `test`.`product` **WHERE** name = 'yy';





## 可重复读

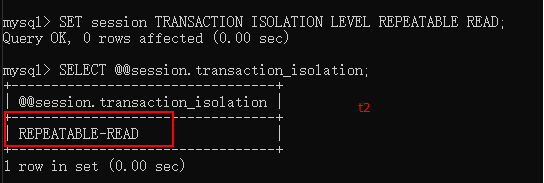
REPEATABLE READ

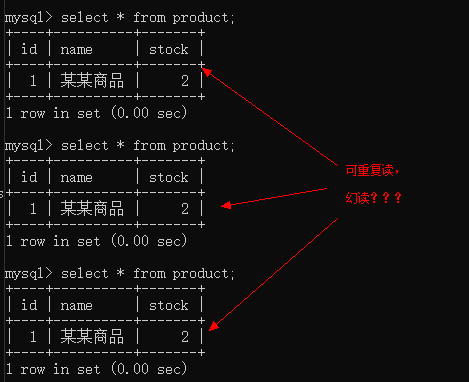
第一步：恢复环境

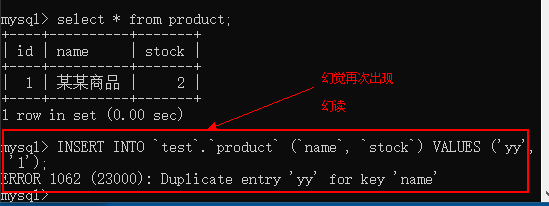
**update** test.product **set** stock = 2 **where** id=1;

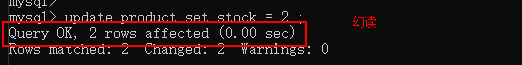
**DELETE** **FROM** `test`.`product` **WHERE** name = 'yy';

第二步：修改隔离级别









## 串行化

SERIALIZABLE

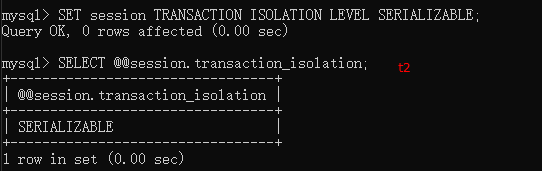
第一步：恢复环境

**update** test.product **set** stock = 2 **where** id=1;

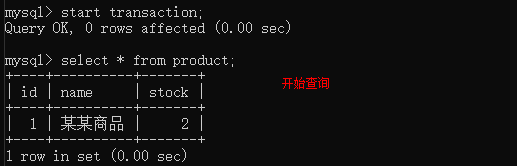
**DELETE** **FROM** `test`.`product` **WHERE** name = 'yy';

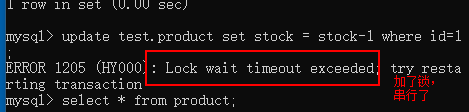
第二步：修改隔离级别

**SET** **session** **TRANSACTION** **ISOLATION** **LEVEL** SERIALIZABLE;



演示：





**不会出现脏读，不可重复读，以及幻读。**

# When??

根据具体业务情况，合理使用，一般选择中间2个级别：**读已提交，可重复读**。

资源层：隔离级别，数据库支持的，这种非常消耗性能

业务层：控制，线程。

# Practice

在一家企业里，现在要求你给一台线上的mysql数据库的表数据做“**热备份**“，请问你会选择什么样的事务隔离级别去备份呢？？为什么？？还需要注意哪些细节？？

**可重复读！！！**

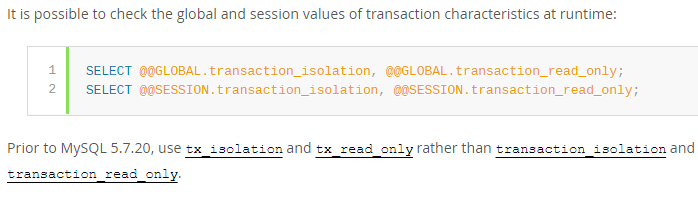
**START TRANSACTION WITH CONSISTENT SNAPSHOT.**

If the transaction [isolation level](https://dev.mysql.com/doc/refman/5.7/en/glossary.html#glos_isolation_level) is [REPEATABLE READ](https://dev.mysql.com/doc/refman/5.7/en/innodb-transaction-isolation-levels.html#isolevel_repeatable-read) (the default level), all consistent reads within the same transaction read the snapshot established by the first such read in that transaction. You can get a fresher snapshot for your queries by committing the current transaction and after that issuing new queries.

# 备注

**坑：注意版本区别**

**5.7以前的设置事务不一样**



参考：<https://dev.mysql.com/doc/refman/5.7/en/innodb-transaction-isolation-levels.html>

快照读：<https://dev.mysql.com/doc/refman/5.7/en/innodb-consistent-read.html>