Sql事务：保持数据完整性概念

数据库事务

程序事务（本质上也是数据库事务），分布式

（在SQL Server中，事务是一组数据库操作，这些操作被视为单个逻辑单元，并且要么全部成功，要么全部失败。事务通常用于确保数据库操作的一致性和完整性。如果一个事务中的任何操作失败，整个事务将被回滚，这意味着所有操作都将被撤消，数据库将回到事务开始之前的状态。事务还可以用于实现并发控制，以确保多个用户同时访问数据库时不会发生冲突。在SQL Server中，您可以使用BEGIN TRANSACTION、COMMIT TRANSACTION和ROLLBACK TRANSACTION语句来创建和管理事务）

数据库事务：三种

比如银行转账从A转到B，至少需要两步

账户A的资金减少

账户B资金增加

操作多张表，一个表有多个操作

数据库的事务四种特性：

1. 原子性：事务是一个完整的操作，事务的各步操作是不可分的（原子的），要么都执行，要么都不执行。
2. 一致性：当事务完成时，数据必须处于一致状态。
3. 隔离性：并发事务之间彼此隔离，独立，它不应该以任何方式依赖于或影响其他事务。
4. 永久性：事务完成之后，它对数据库的修改被永久保持。

-不合理 两条语句不是一个整体

--数据库三种事务

--对事物操作只有增删改，查询不会改变数据

--insert delete update select(不影响数据)

--数据库第一种，显示事务 begin transaction

--系统函数，捕获T-SQL语句执行错误编号@@error

use jgwnamedb

begin transaction--开始事务

declare @num int=0

update students

set stuname='张老六'

where stuname='张三'

set @num=@num+@@error--出现错误报为1

update students

set stuName='李老五'

where stuname='李四'

set @num=@num+@@error

if @num=0

begin

print '转帐成功'

commit transaction--提交业务

end

else --如果报错

begin

print '转帐失败'+cast(@num as varchar)

rollback transaction--就回滚信息

end

--数据库第二种，自动事务commit transaction：SQL server的默认模式

--每条单独T-SQL语句视为一个事务

--数据库第三种，隐式事务：通过设置set implicit\_transactions on语句

--将其隐式事务模式设置为打开，其后的T-SQL语句自动启动一个新事务

--提交或回滚一个事务后，下一个T-SQL语句又将启动一个新事务，保证事务的隔离性

set implicit\_transactions on--打开隐式事务，如果打开隐式事务，使用添加删除修改，事务锁表，看不到表中信息，但是表已经执行操作

--正在等待下一步指令，等待提交或者取消 rollback：取消 commit提交

commit

rollback

--oracle，默认隐式事务开启，sqlserver默认隐式事务关闭

--sqlconnection 连接对象 conn.close() 关闭汇话窗口