

# Replication还可以这样玩

利用replication实现字段数据类型快速变更

## DTCC

### 2015中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2015

大数据技术探索和价值发现



HELLO.  
I AM 大菠萝.



京东商城 ( JD.COM ) SQLSERVER DBA

<http://www.cnblogs.com/diablox/>

1

## 一个需求引发的血案

变更字段类型 ; int  $\rightarrow$  bigint



直接在原表上修改

创建新表，导数据，交换表名

直接新表写入，旧表数据归档方式处理



## 停写时间受数据量影响大

方案	优点	缺点	适用范围
A(直接修改)	操作最简单	执行时间、风险不可控	数据量较小的表
B(建表导数据)	不影响读取	受数据量影响较大	接受短时停写
C(建表不导数据)	执行时间短	应用需要变更逻辑	日志类型场景



前期概述

方案形成

成果输出

3

需求升级



一次，全搞定！





# MISSION IMPOSSIBLE

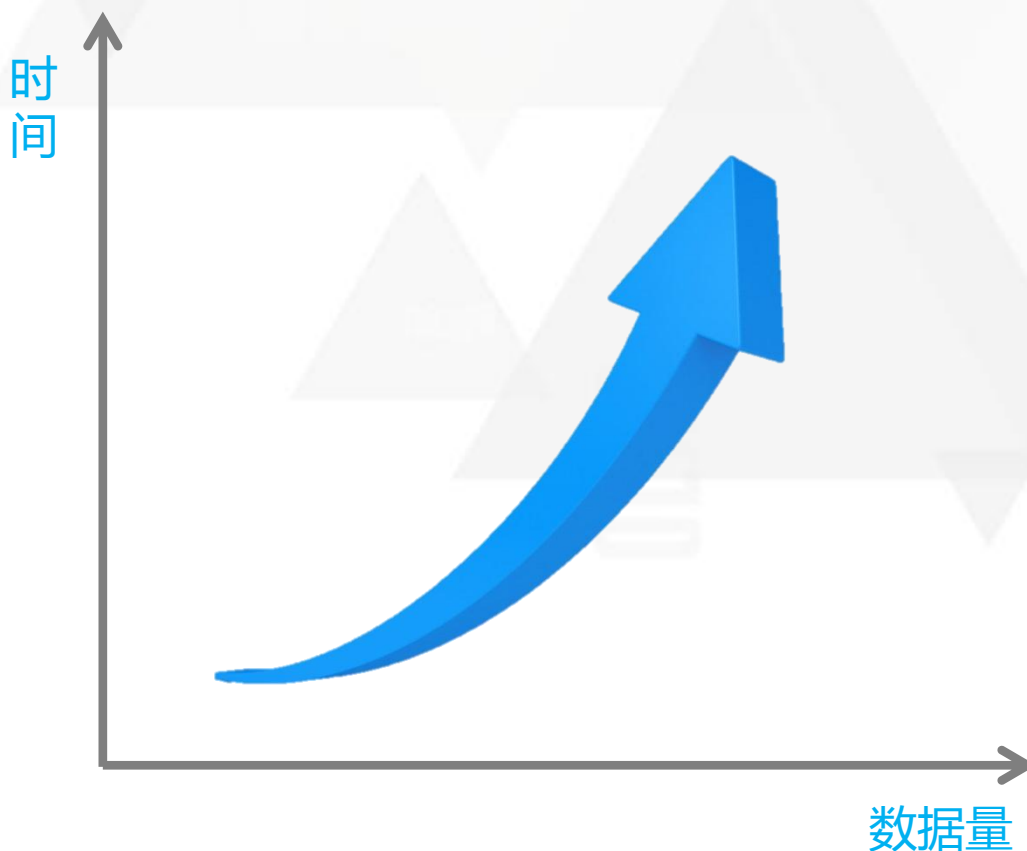




1

变通思路

关键点：数据量与执行时间的关系





有没有更好的方案可以控制停机后的**影响数据量**？

方案A(直接修改)

Sch-M锁（被动停写）

方案B(建表导数据)

数据初始化（主动停写）

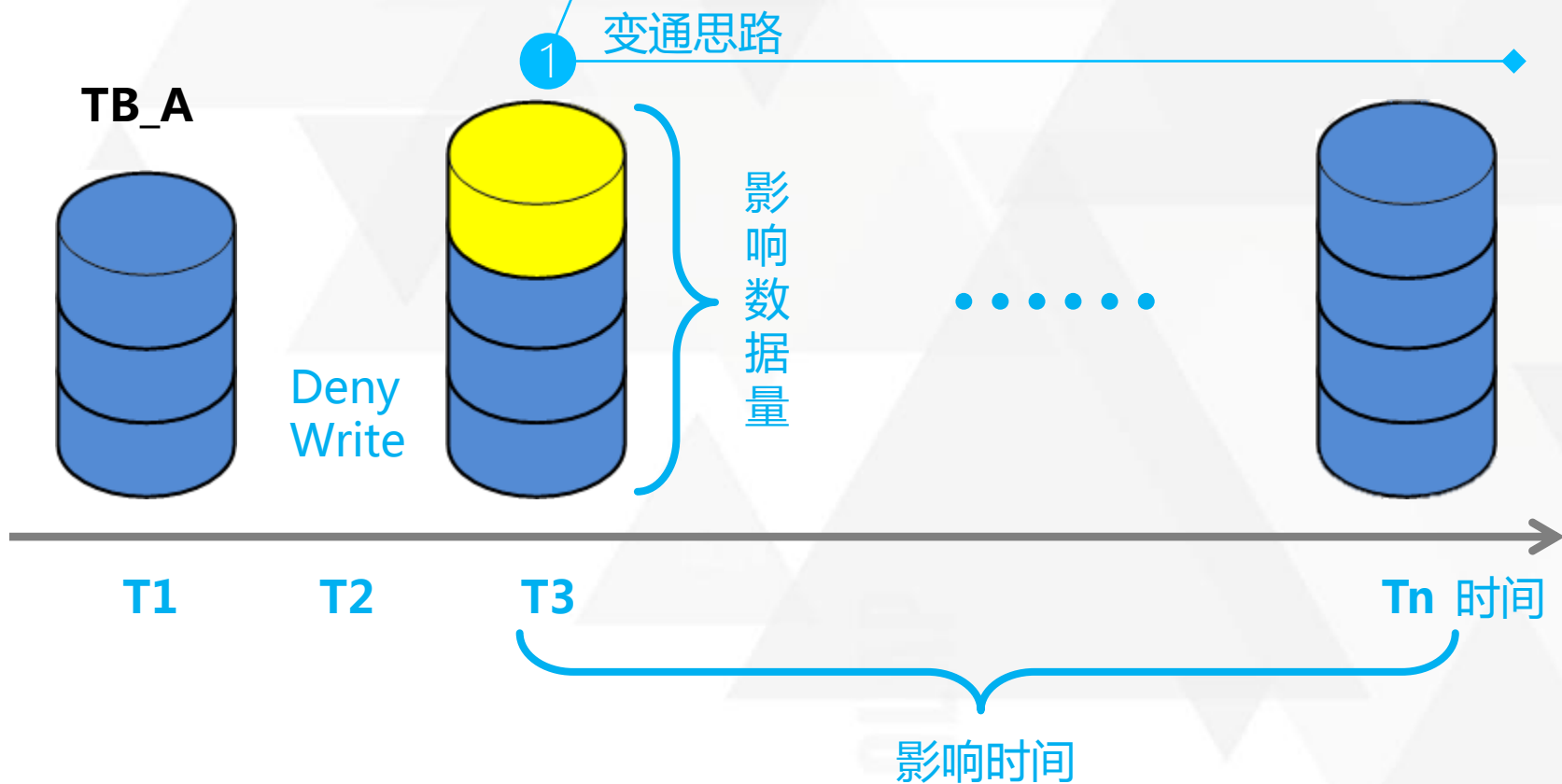
影响数据量为**当前快照**的数据量

影响数据在**停写后伴随结构变动**的整个过程



方案C是否真的应该被pass呢？



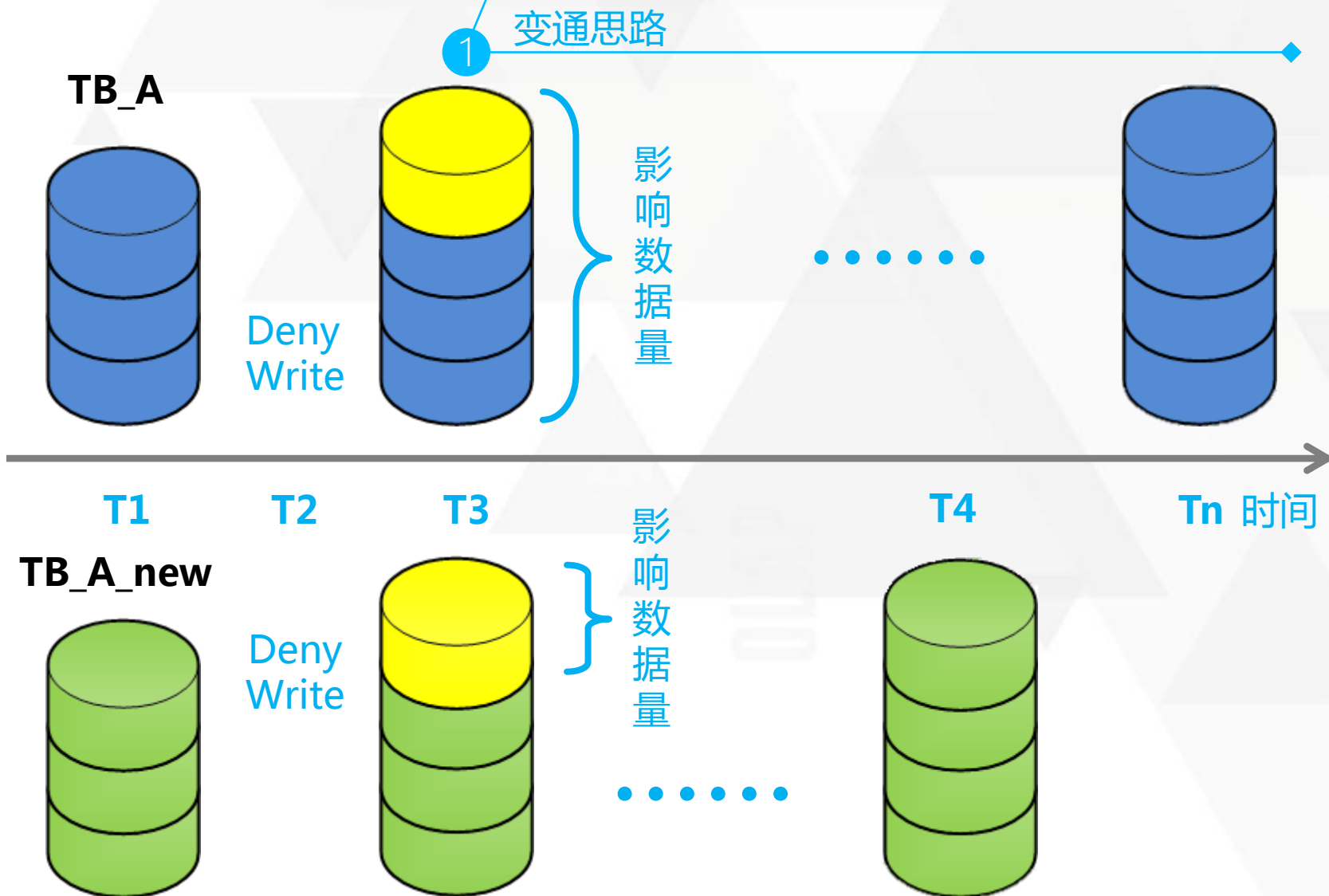




前期概述

方案形成

成果输出



如何**捕获变更**，持续应用到TB\_A\_new中，在停写后较短的时间内达到TB\_A和TB\_A\_new数据一致

# Transaction Replication



## 利用Transaction Replication实现TB\_A到TB\_A\_new的“准实时同步”

优点：

- 1、将数据初始化的动作前置
- 2、有效降低 “影响数据量”



- 1、创建“同库异构”的复制链路
- 2、其他对象的一致性（索引、权限.....）
- 3、停写、删除复制关系、交换表名、重建复制关系





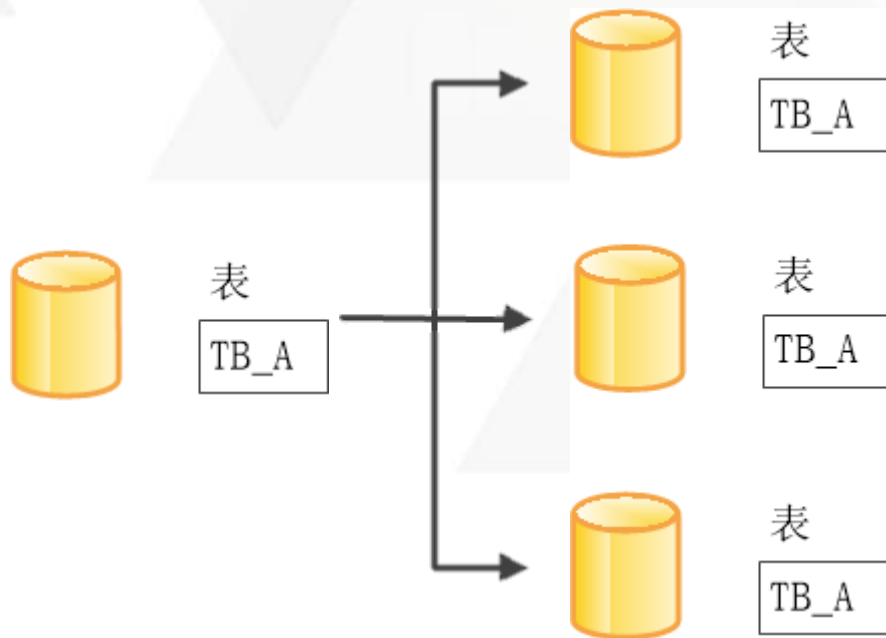
目的：

- 实现TB\_A到TB\_A\_new的数据同步关系
- 每个库均有TB\_A、TB\_A\_new两个“同库异构”数据表



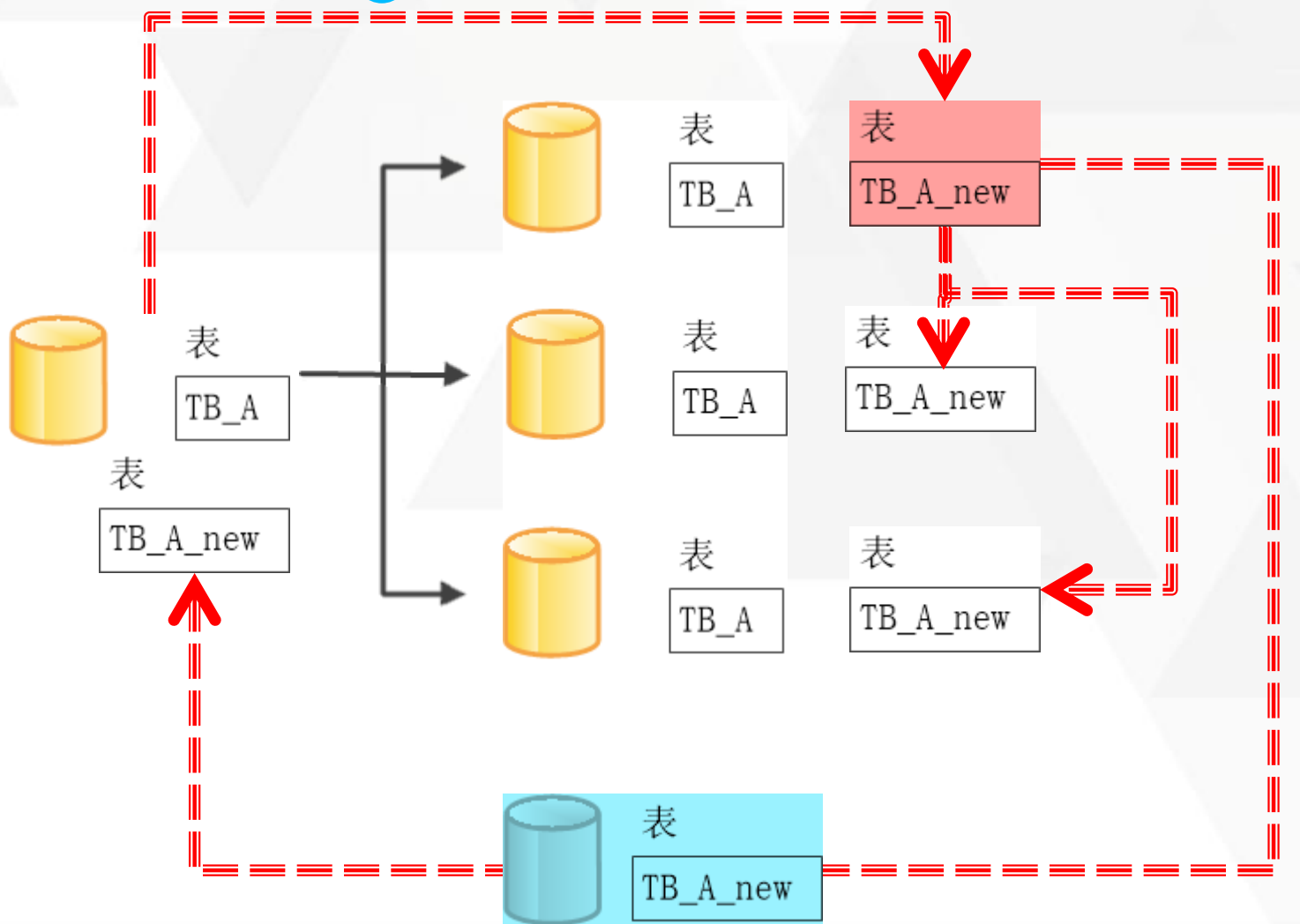
4

复制环路



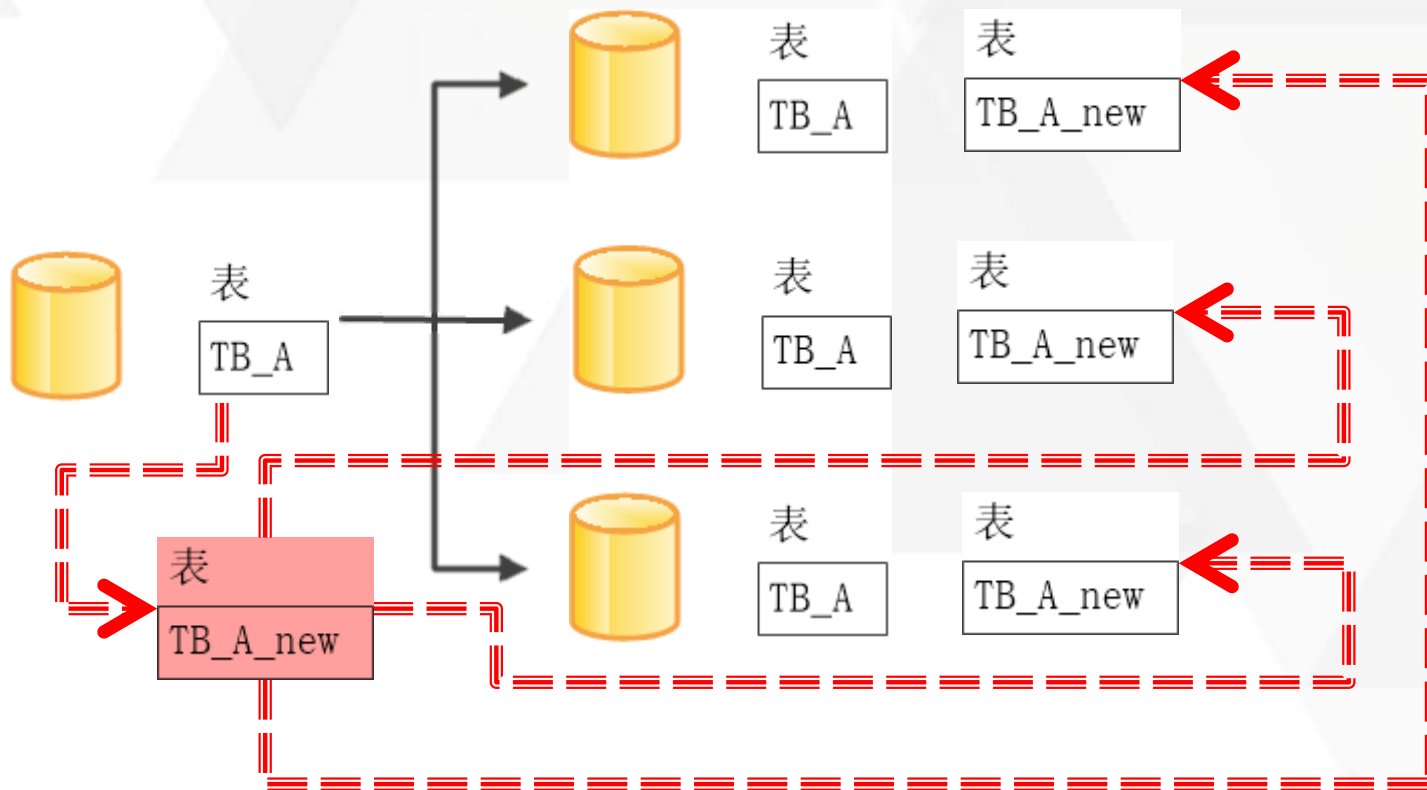
4

复制环路



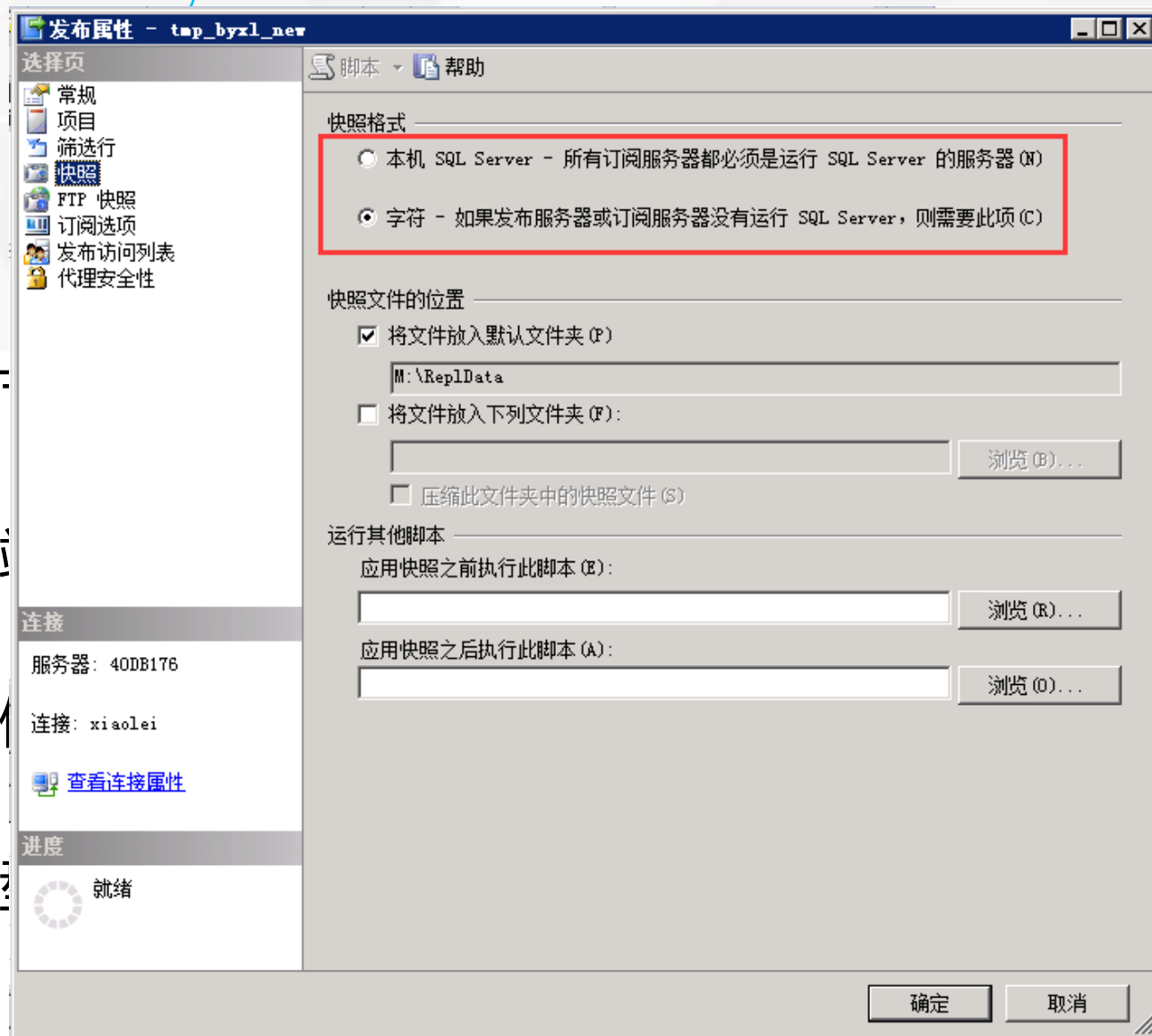
4

复制环路



## 准备阶段

- 手动创建根目录
- 修改“订阅选项”
- 修改复制存储
- 使用“字符型”快照格式



## 维护阶段

- 停写，检测复制延迟状态
- 删除复制链路，交换表名后重建（不初始化订阅）
- 检测新复制链路，清理旧表





人·工时





前期概述

方案形成

成果输出

1

实验成果

**NOTHING IS IMPOSSIBLE**







思想有多远

我们就能走多远



**DTCC** 2015年中国数据库技术大会  
DATABASE TECHNOLOGY CONFERENCE CHINA 2015





THANKS