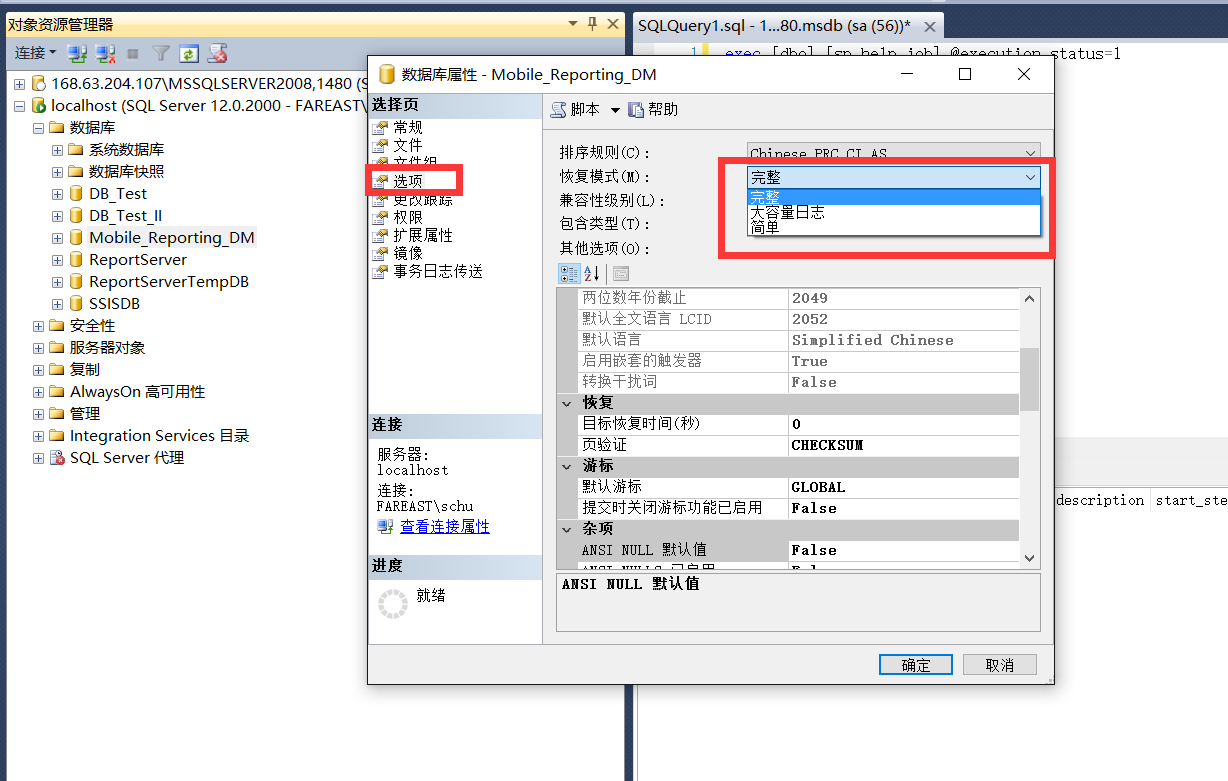
# SQL Server维护能力提升

# 创建新库

## 三种模式

SQL Server数据库有三种恢复模式：简单恢复模式、完整恢复模式和大容量日志恢复模式：

****

1. Simple 简单恢复模式,

Simple模式的旧称叫”Checkpoint with truncate log“，其实这个名字更形象，在Simple模式下，SQL Server会在每次checkpoint或backup之后自动截断log，也就是丢弃所有的inactive log records，仅保留用于实例启动时自动发生的instance recovery所需的少量log，这样做的好处是log文件非常小，不需要DBA去维护、备份log，但坏处也是显而易见的，就是一旦数据库出现异常，需要恢复时，最多只能恢复到上一次的备份，无法恢复到最近可用状态，因为log丢失了。 Simple模式主要用于非critical的业务，比如开发库和测试库，但是道富这边的SQL Server（即使是生产库）大都采用Simple模式，是因为这边的SQL Server大都用于非critical的业务（critical的数据库大都采用Oracle和DB2)，可以忍受少于1天的数据丢失(我们的job每天都会定时备份全库）。

如果需要压缩数据库日志（Shrink语句），将数据库模式切换到简单恢复模式后压缩率才是最高的，如果你的数据库在完整恢复模式或大容量日志回复模式下采用日志压缩，压缩后的日志大小并不会很理想。

1. Full 完整恢复模式,

和Simple模式相反，Full模式的旧称叫”Checkpoint without truncate log“，也就是SQL Server不主动截断log，只有备份log之后，才可以截断log，否则log文件会一直增大，直到撑爆硬盘，因此需要部署一个job定时备份log。Full的好处是可以做point-in-time恢复，最大限度的保证数据不丢失，一般用于critical的业务环境里。缺点就是DBA需要维护log，增加人员成本（其实也就是多了定时备份log这项工作而已）。

1. Bulk-logged 大容量日志恢复

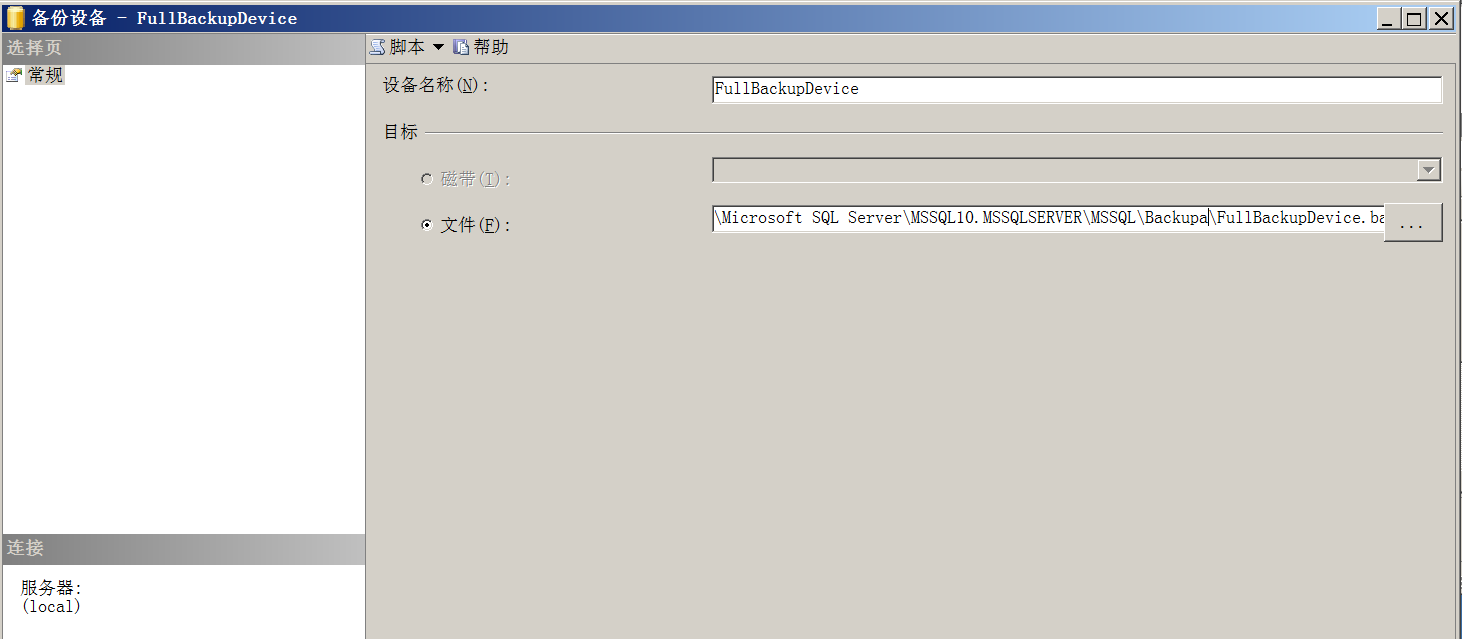
Bulk-logged模式和full模式类似，唯一的不同是针对以下Bulk操作，会产生尽量少的log： 1) Bulk load operations (bcp and BULK INSERT). 2) SELECT INTO. 3) Create/drop/rebuild index 众所周知，通常bulk操作会产生大量的log，对SQL Server的性能有较大影响，bulk-logged模式的作用就在于降低这种性能影响，并防止log文件过分增长，但是它的问题是无法point-in-time恢复到包含bulk-logged record的这段时间。 Bulk-logged模式的最佳实践方案是在做bulk操作之前切换到bulk-logged，在bulk操作结束之后马上切换回full模式。

# 备份和还原

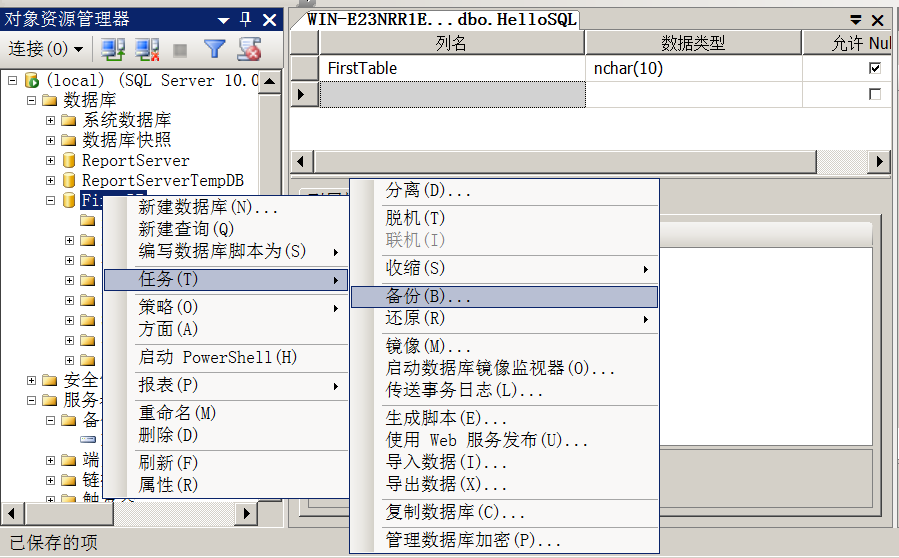
## 关于备份的9种情况

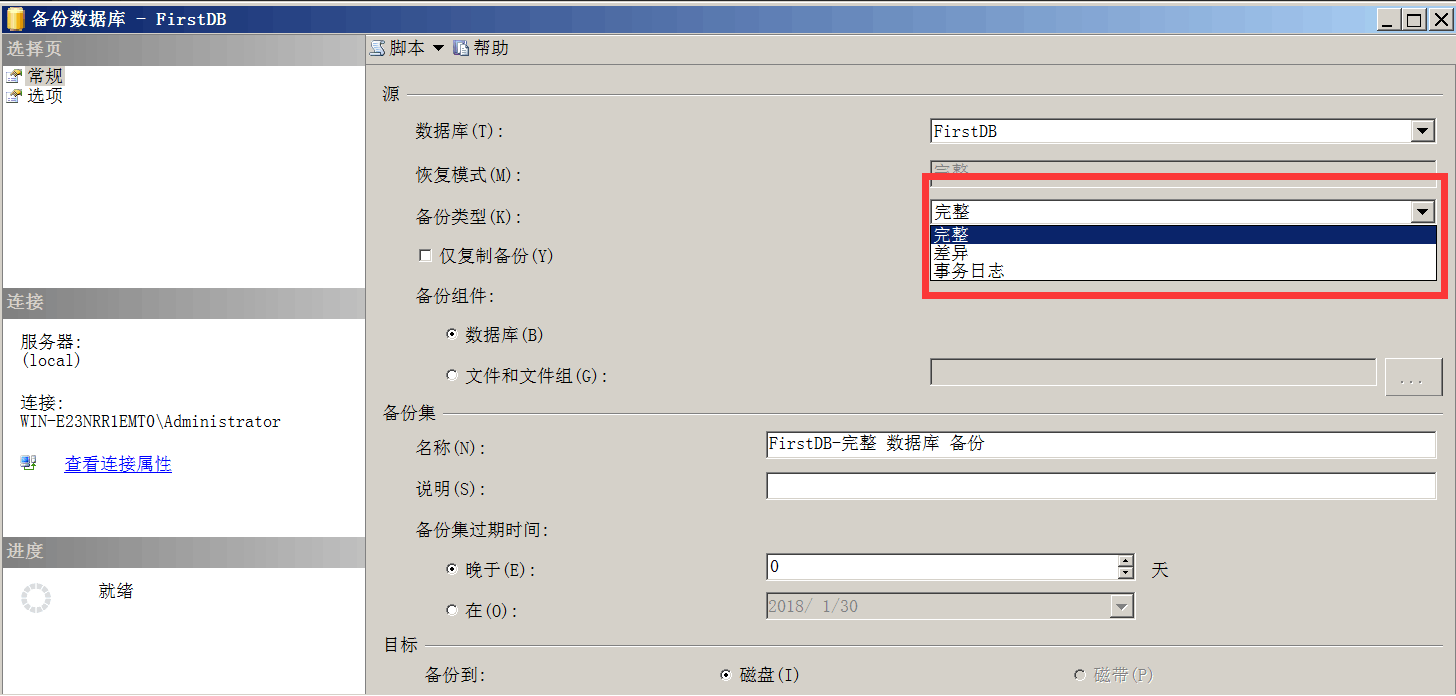
(1) 有RAID，还需要做数据库备份吗？   
回答:需要。有了RAID，万一部份磁盘损坏，可以修复数据库，有的情况下数据库甚至可以继续使用。但是，如果哪一天，你的同事不小心删除了一条重要的记录，怎么办？RAID是无能为力的。你需要合适的备份策略，把那条被误删的数据恢复出来。所以有了RAID，仍需要做备份集群，磁盘镜像同理。   
 (2) 我们需要全备份+日志备份？   
解答:如果你只做全备份，那么受限于全备份的大小和备份时间，不可能常做。而且只有全备份，不能将数据库恢复至某个时间点。所以，我们需要全备份+日志备份。比如每天一个全备份，每隔1小时或若干分钟一个日志备份。说到差异备份，因为微软的差异备份记录的是上一次全备份以来发生的变化，所以，如果数据库的改动很频繁的话，没过多久，差异备份就会和全备份的大小接近，因此这种情况下就不合适了。因此，全备份+日志备份的方案适合绝大多数的用户。   
 (3) 如果你仅在数据库本地做备份，万一磁盘损坏，或者整个服务器硬件损坏，备份也就没了，就没法恢复数据库。   
解答:因此，你需要把备份文件传送至另一个物理硬件上。大多数用户不用磁带机，因此不考虑。一般，我们需要另一台廉价的服务器或者PC来存放数据库的备份，来防止硬件损坏造成的备份丢失。   
 (4) 你可以在数据库服务器本地做完备份，然后使用某些方式将备份文件传送至备机。你是在备份完成后就马上穿送的吗？其实可以考虑将传送备份的脚本用T-SQL语句来写。   
 (5) 备份文件传送至备机后，就可以高枕无忧了吗？   
解答:不。作为DBA的你还需要检查备机上的备份文件是否能将数据库恢复至最新，如果采用日志备份，会不会因为丢失某一个日志备份文件而导致数据库不能恢复至最新？如何检查日志备份文件之间存在断档？   
 (6) 为了将数据库尽可能的恢复到最新，你可能会每隔10分钟（甚至1分钟）执行一次日志备份，那么万一数据库坏了，在恢复的时候，手动恢复成百上千个日志文件，是不是不太现实？   
 (7) 如果你所在公司有很多的数据库服务器（就像我所在的公司），而且磁盘空间有限，那么你不得不经常登录服务器来删除旧的备份文件，如果哪天忘了，或者五一十一长假，磁盘空间用完了，就麻烦了。   
 (8) 数据库在备份的时候，并不会检查数据页面的完整性，如果数据页坏了，备份作业仍会执行，而且不会报错，等到你发现数据页有错误的时候，你也很可能已经因为磁盘空间不足，而删除了早期的备份，而此时剩下的那些备份可能都是包含损坏的数据页，如果损坏的数据页是某个表的表头的话，那这个表你就再也没办法恢复了。所以你需要定期执行DBCC检查，来尽早发现数据库页面的完整性。在未作完DBCC检查之前，你不能删除旧的备份，以防止新的备份存在问题。所以，删除备份文件的工作变的有些麻烦。   
 (9) 你可能知道SQL Server提供了数据库维护计划。没错，使用它可以定期做备份，执行DBCC检查，但这一切仅限于本机操作。为了使数据库可靠，你还是需要自己把本地备份传送至备机。

## 手工备份

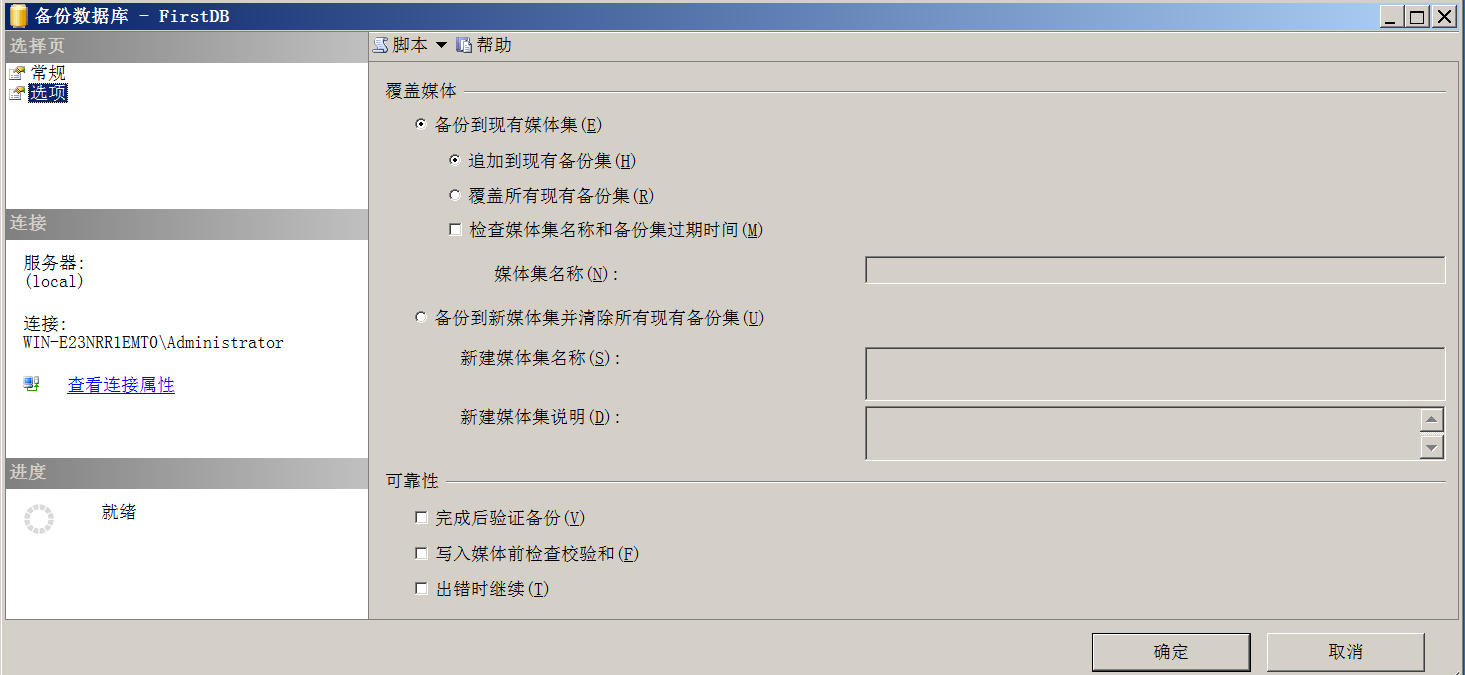


创建备份设备。

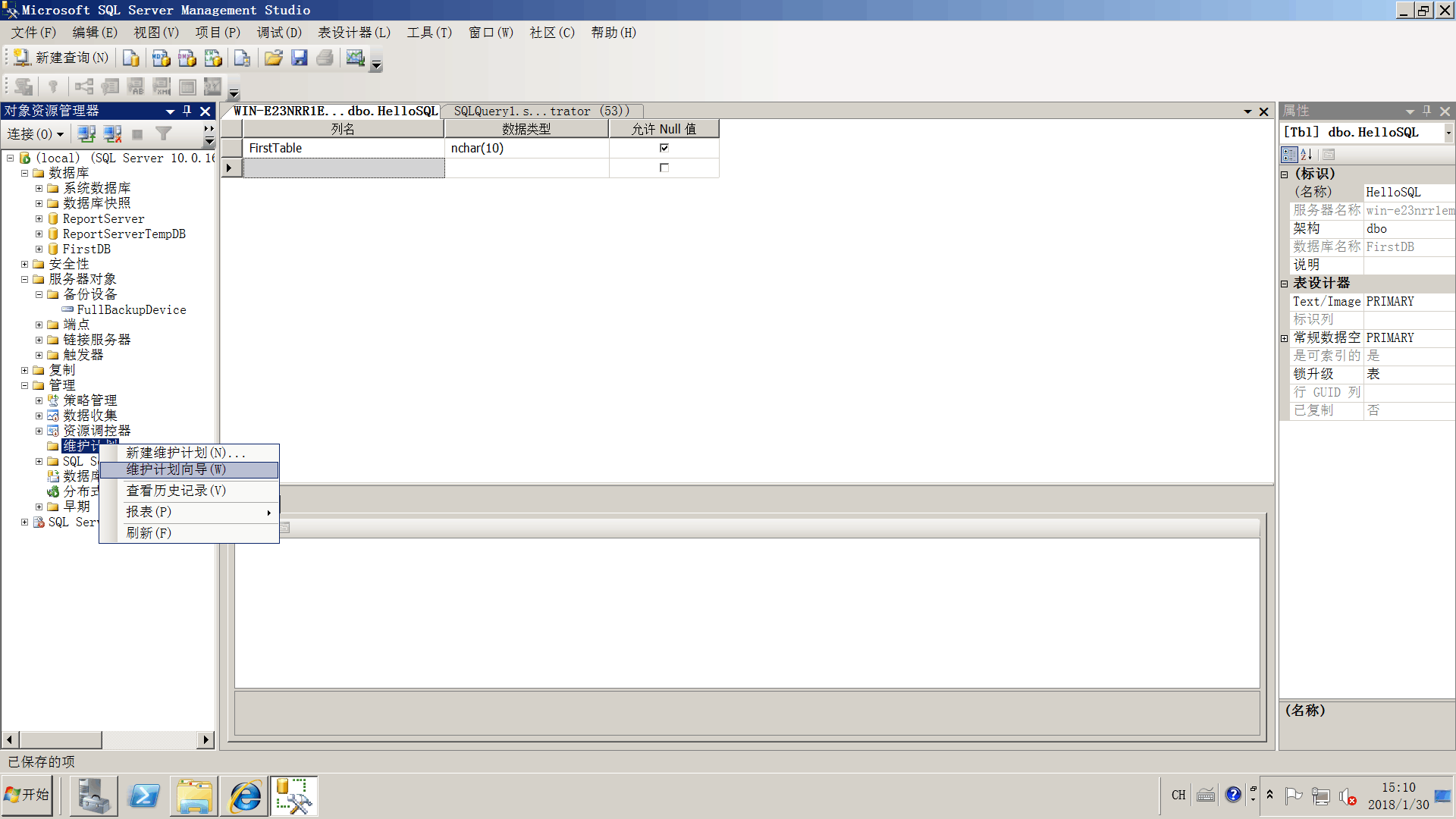


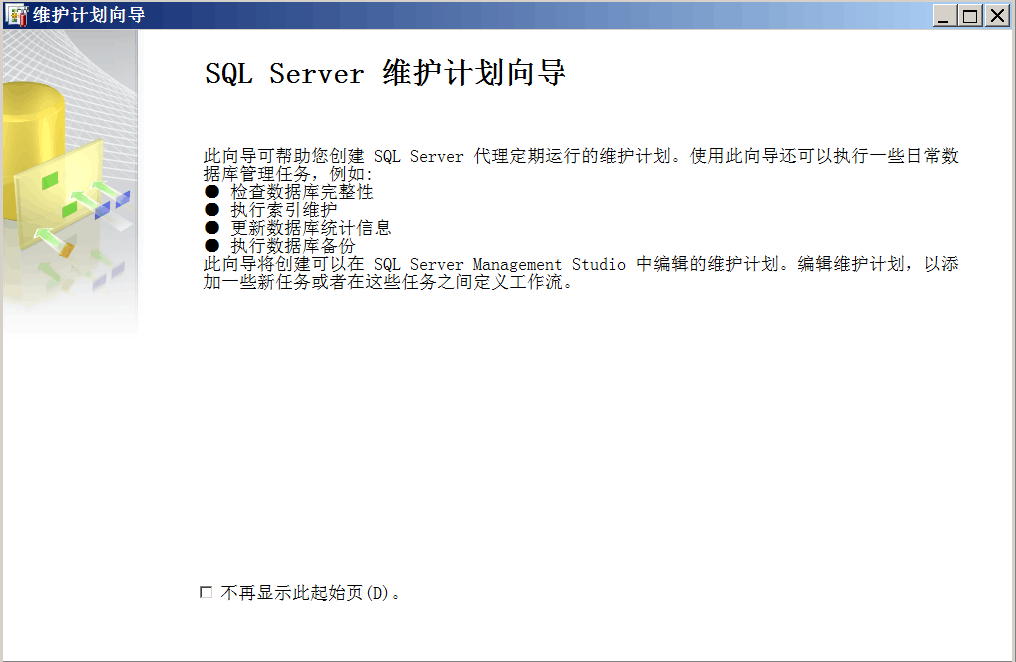


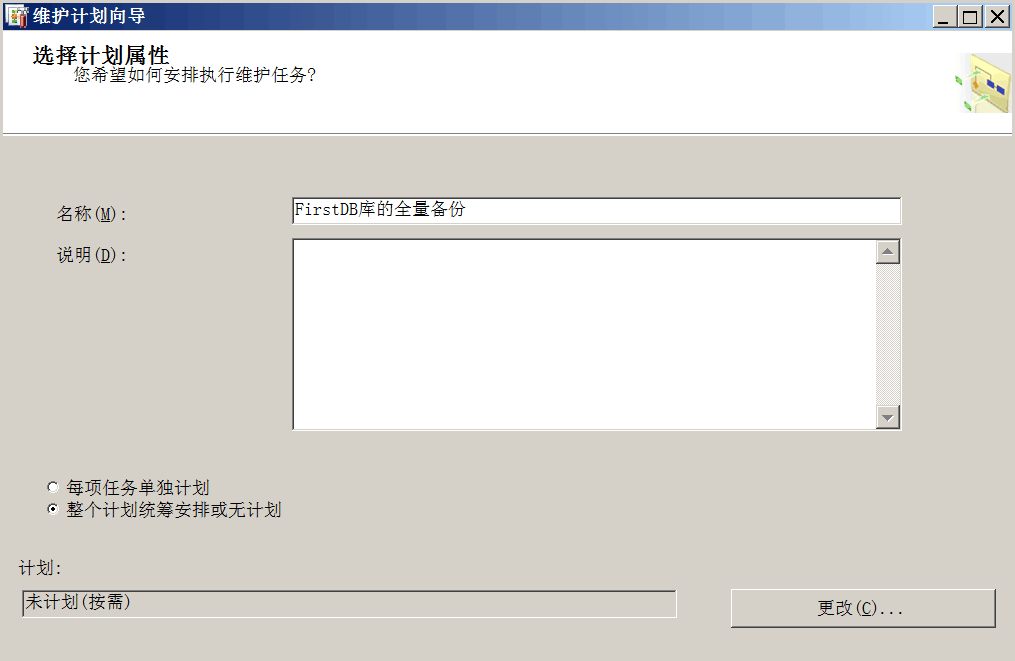
这三个选项，前两个都是备库的数据不备日志，第三个是专门备日志不备数据。

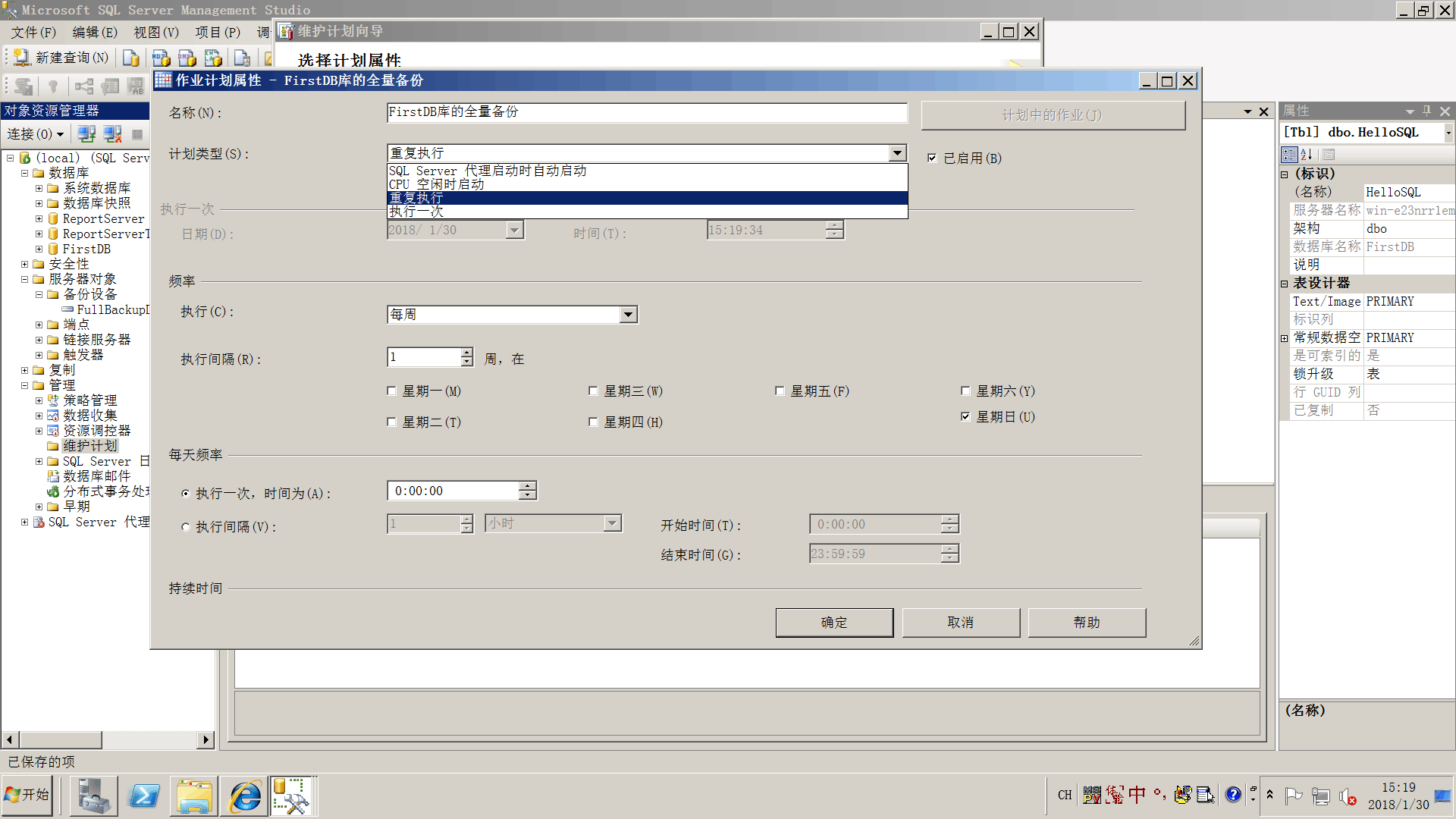


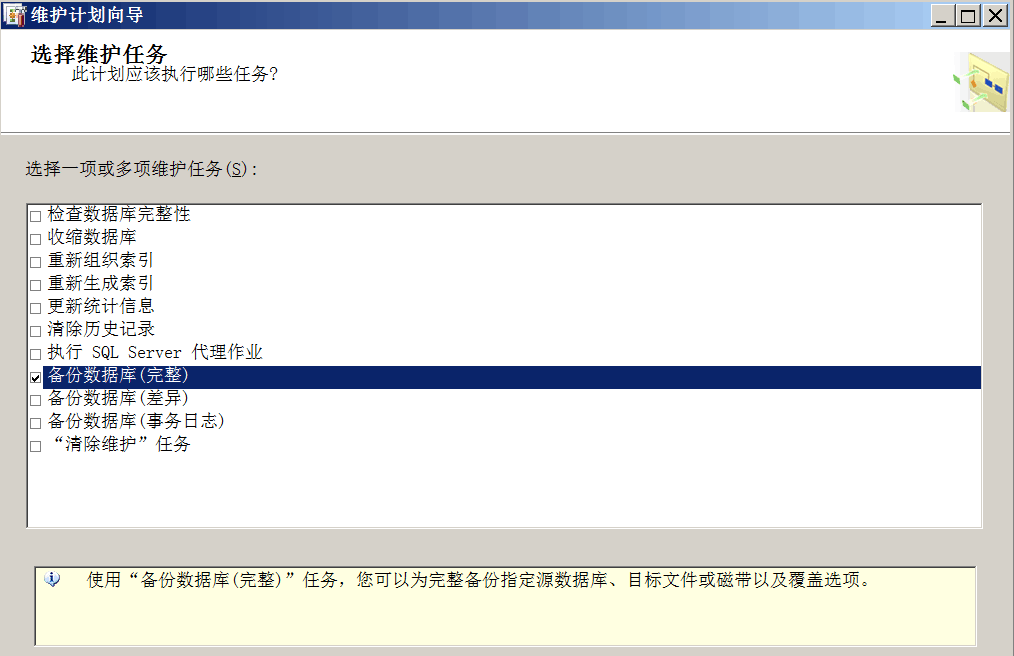
## 自动备份

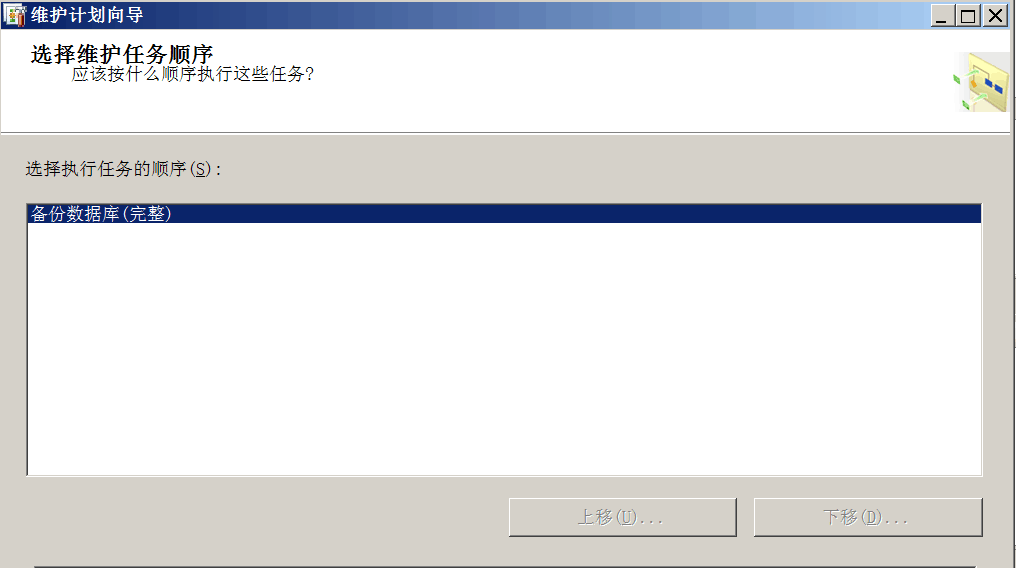


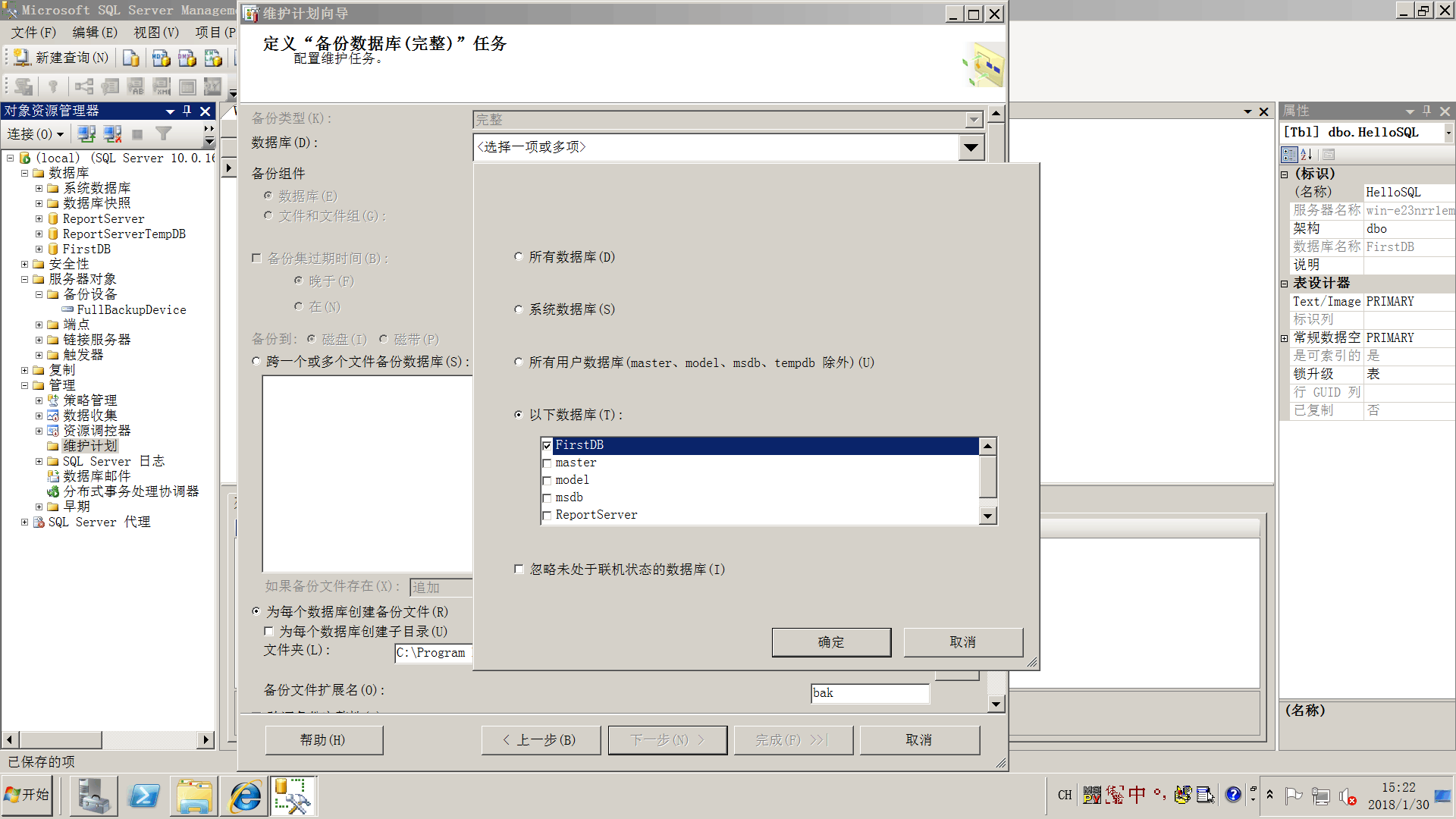


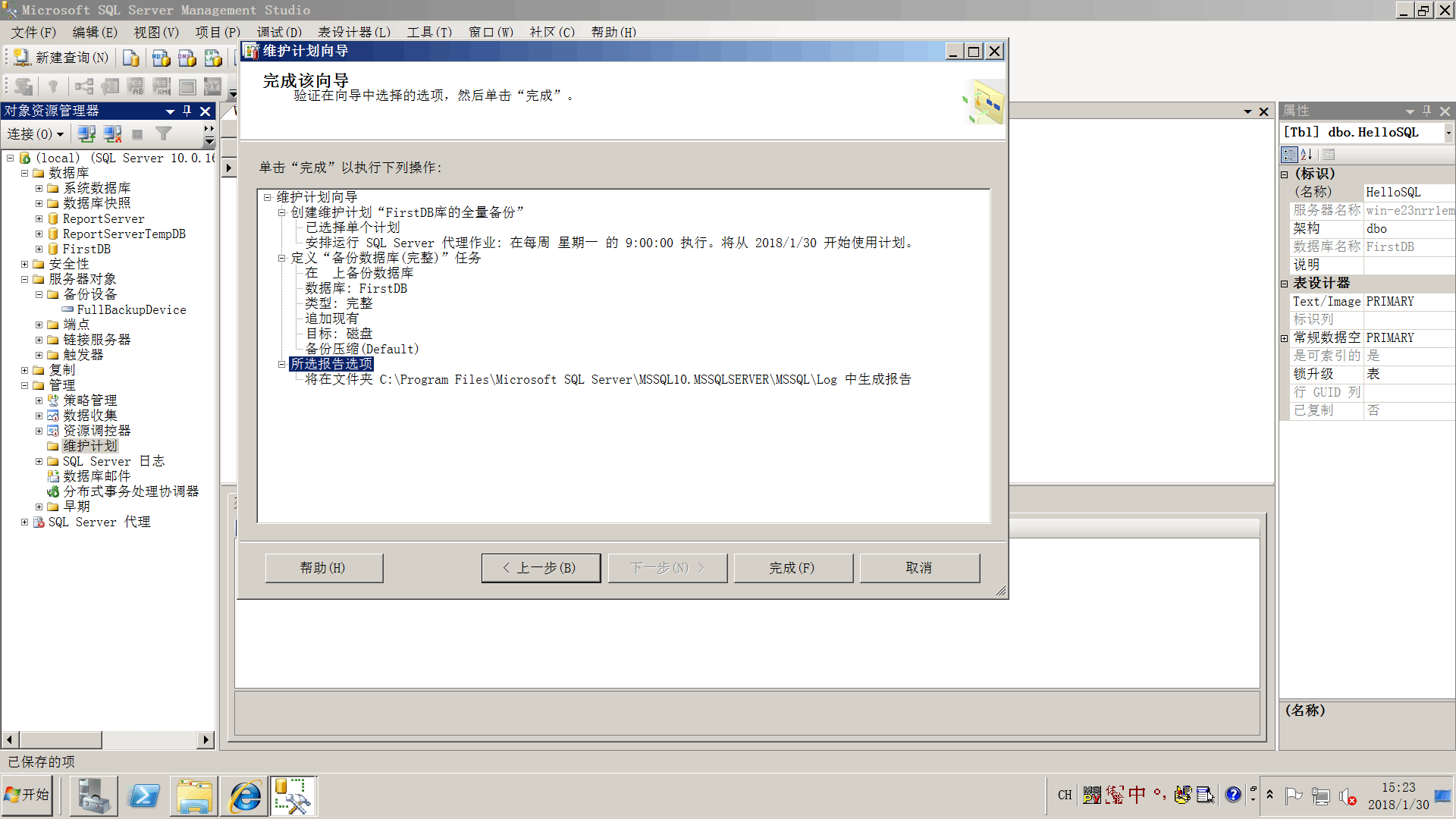


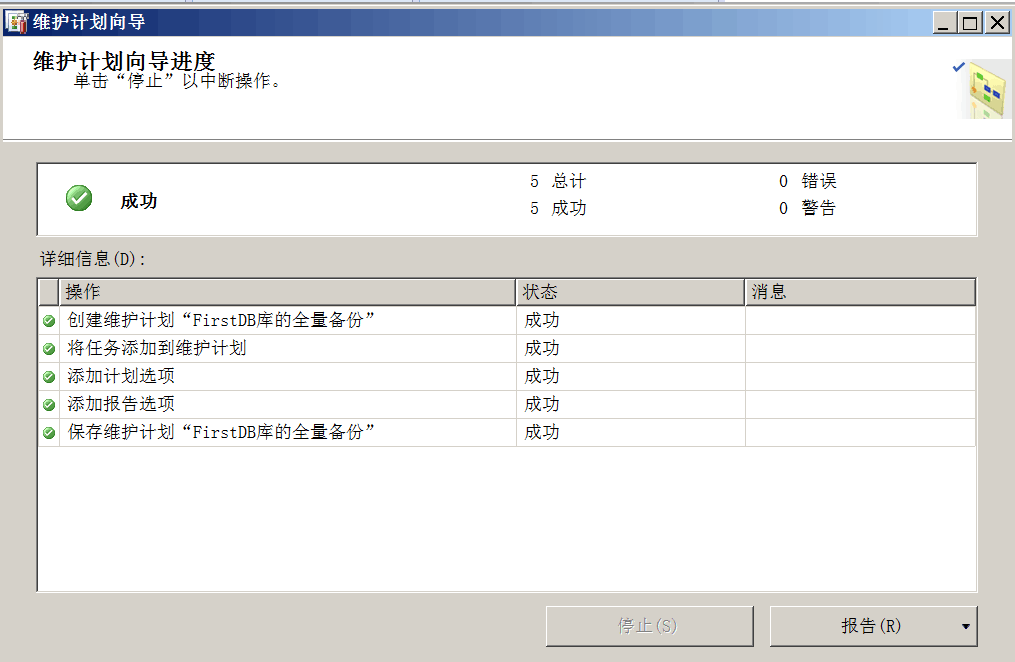










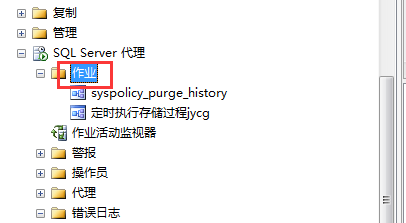


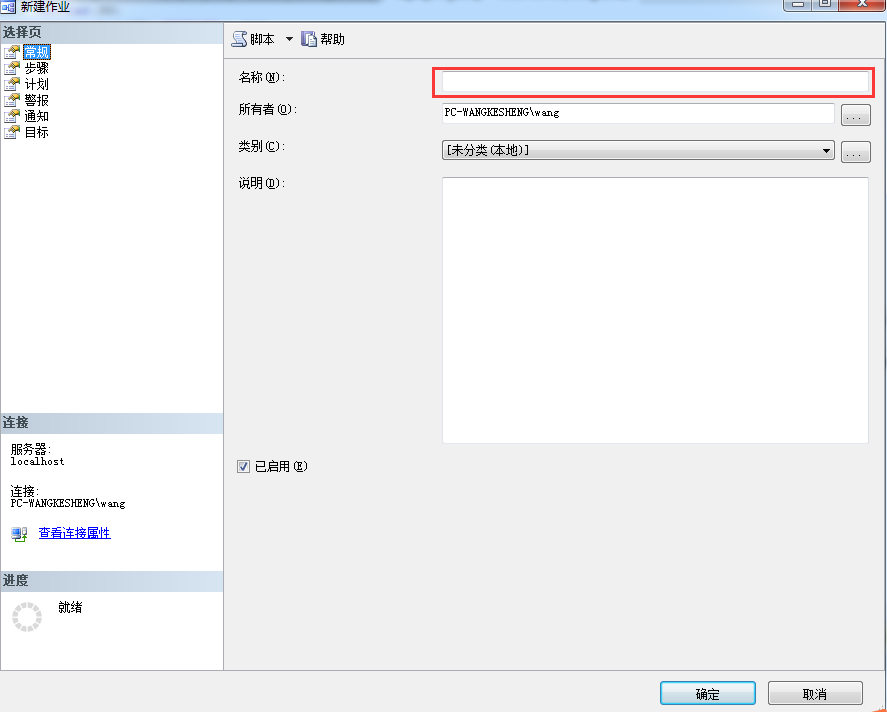
## 备份执行情况

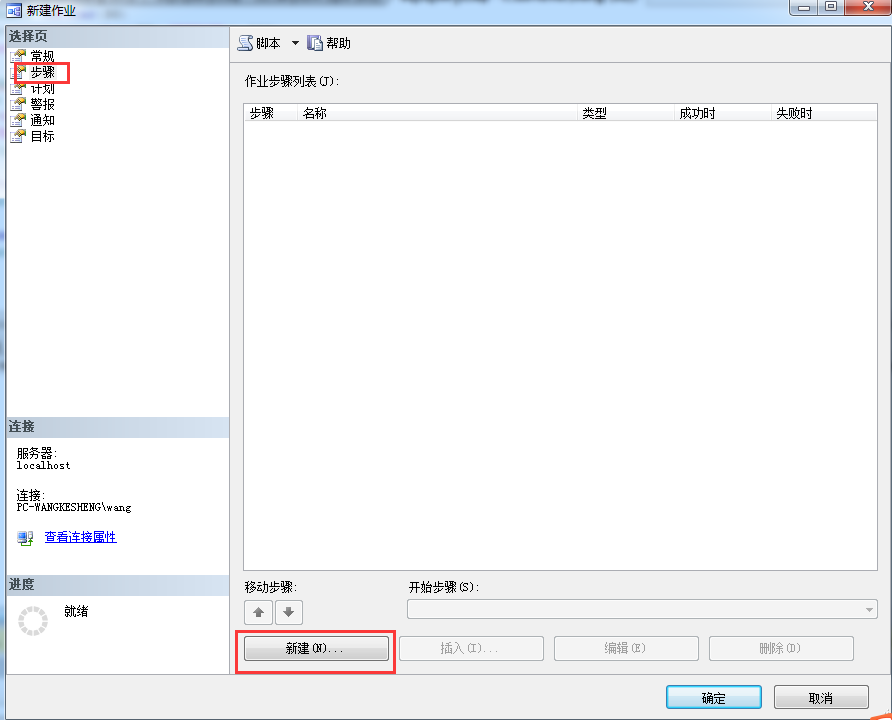
## 优化备份和还原的性能

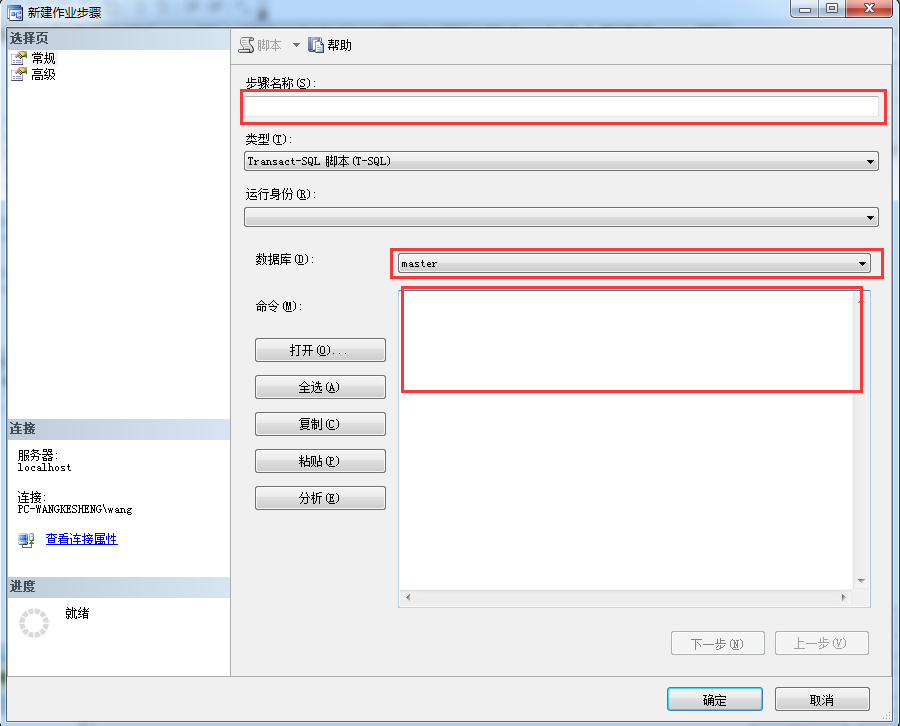
# 维护任务

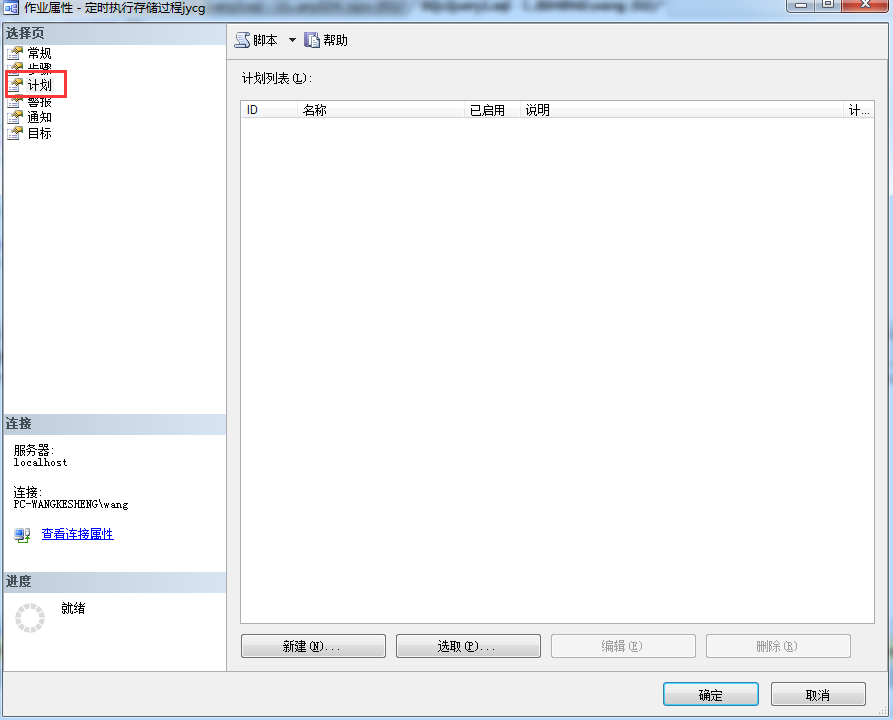
## 定时任务的创建

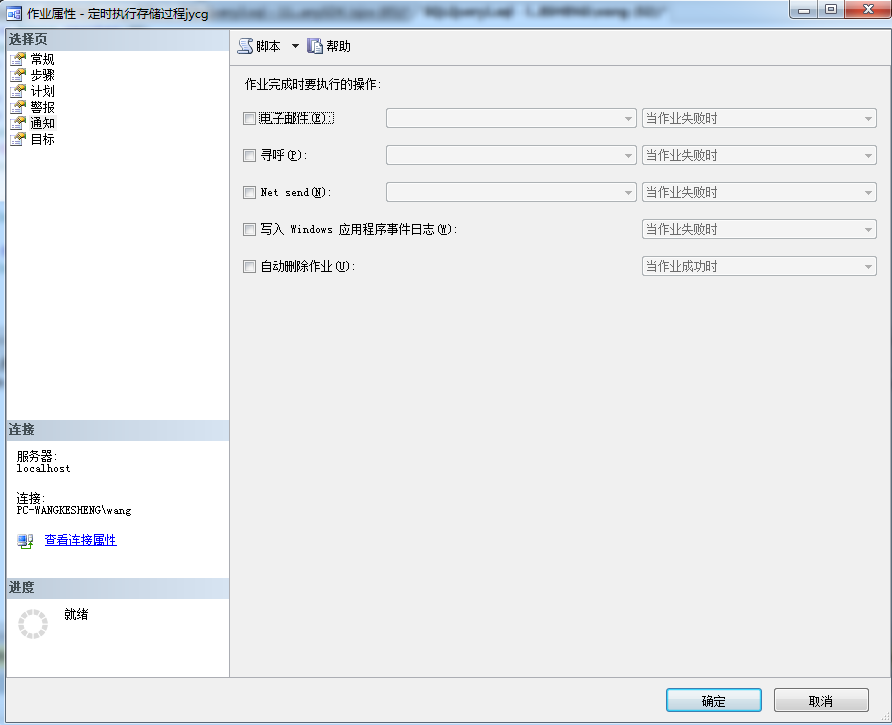
进入数据库，选择SQL server 代理，选择作业，右击新建  


定义名称，其他就按照正常填写就可以了  


选择步骤，新建  


填写常规资料，  
  
这里命令可以输入SQL语句，也可以打开已经写好的角本，确定完成，有多个步骤重复新建就可以了。

接下来，通过“计划”选项，设定执行时间，选择新建  
  
按照需求填写执行时间  

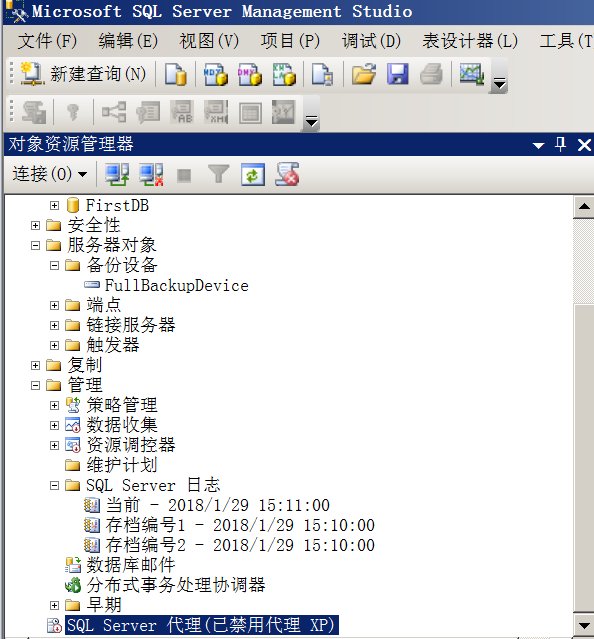

到这里创建一个定时执行任务基本完成。  
最后也可以设置通知，每次任务执行完成后，我们可以了解相关信息或者设置警报信息。  


## 定时任务的执行结果

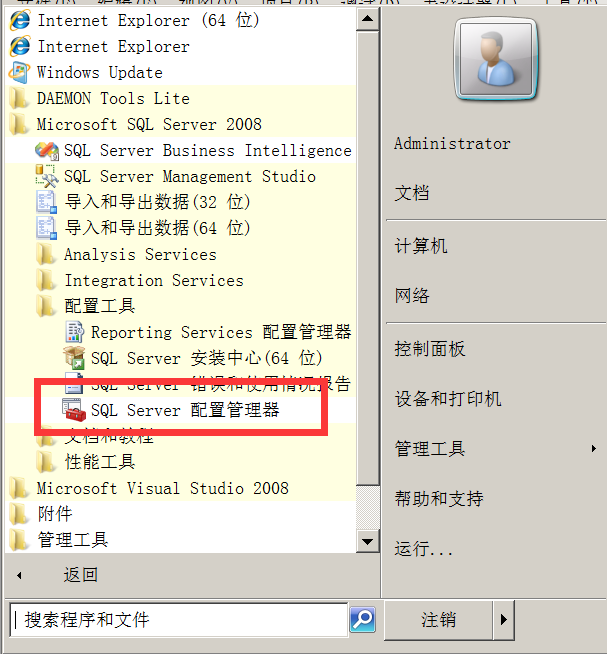


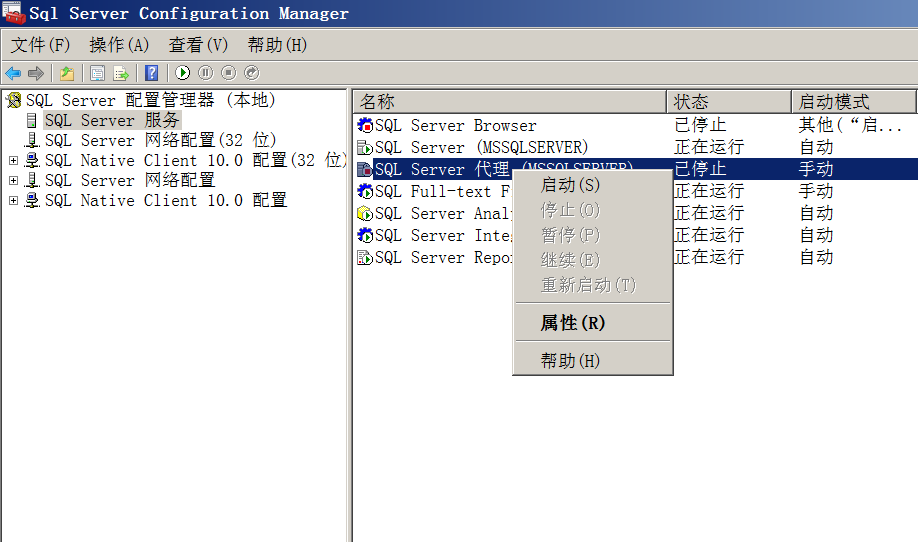
# SQL Server代理

## 判断SQL Server代理是否打开



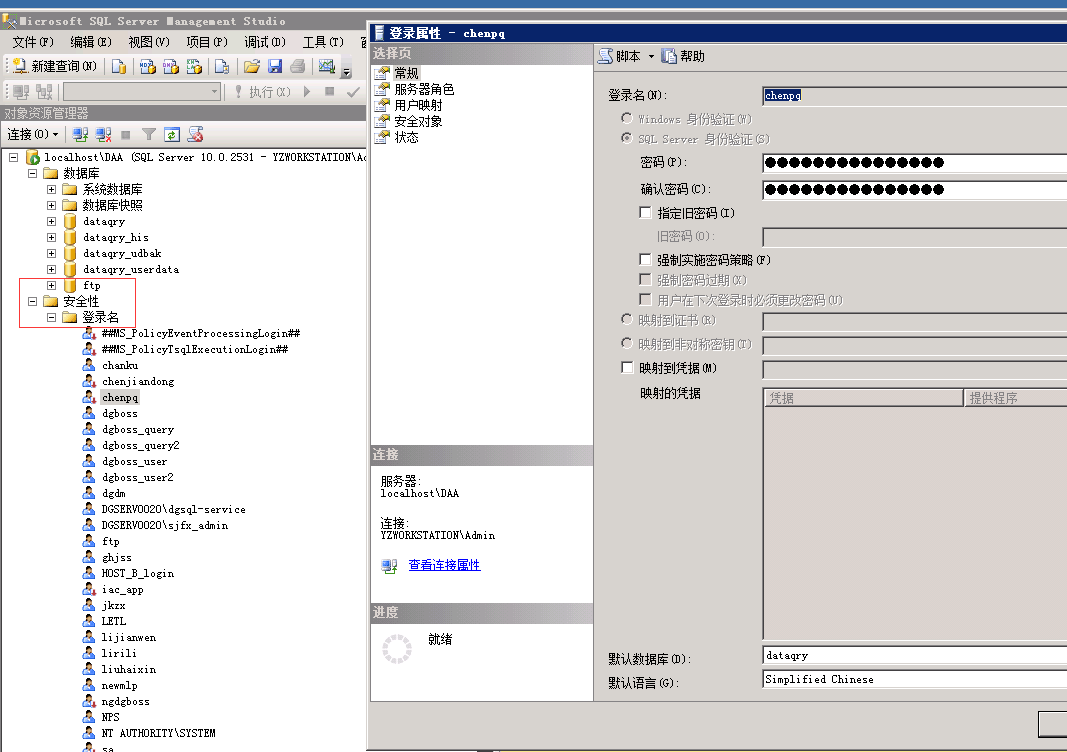
## 启停SQL Server代理





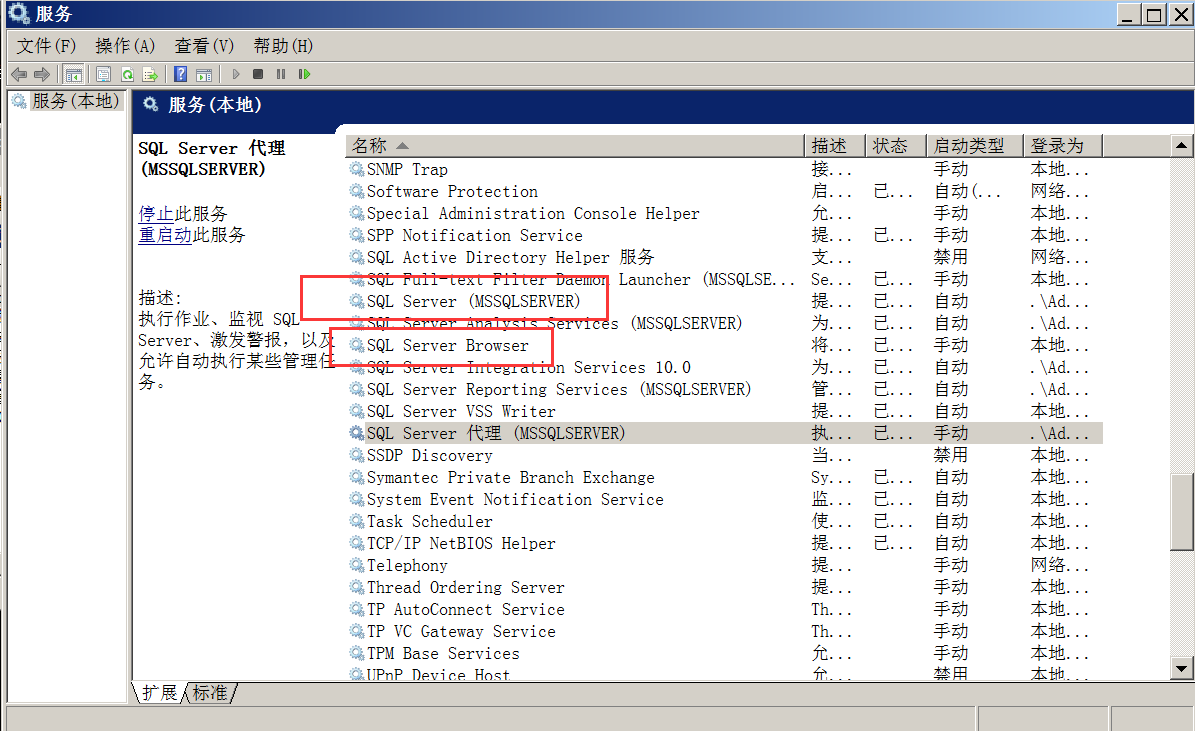
# 用户管理

## 创建用户并授权

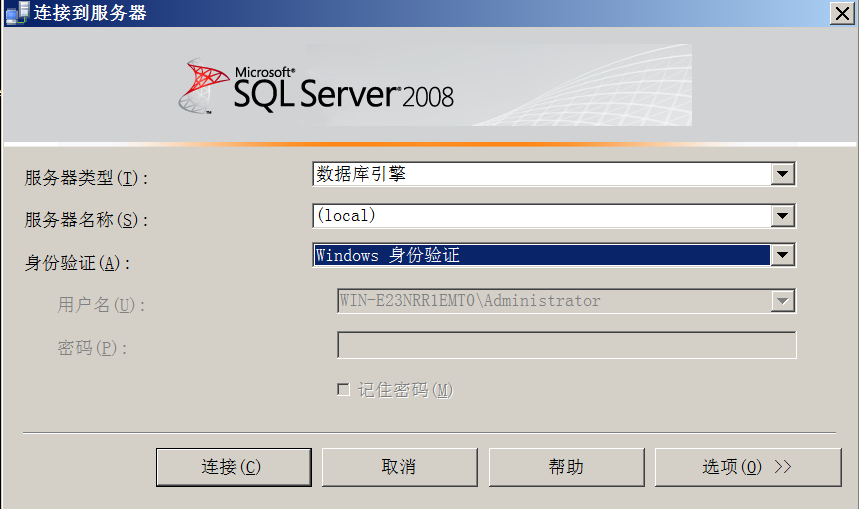


## 重置sa的密码

保证这两个服务要起来

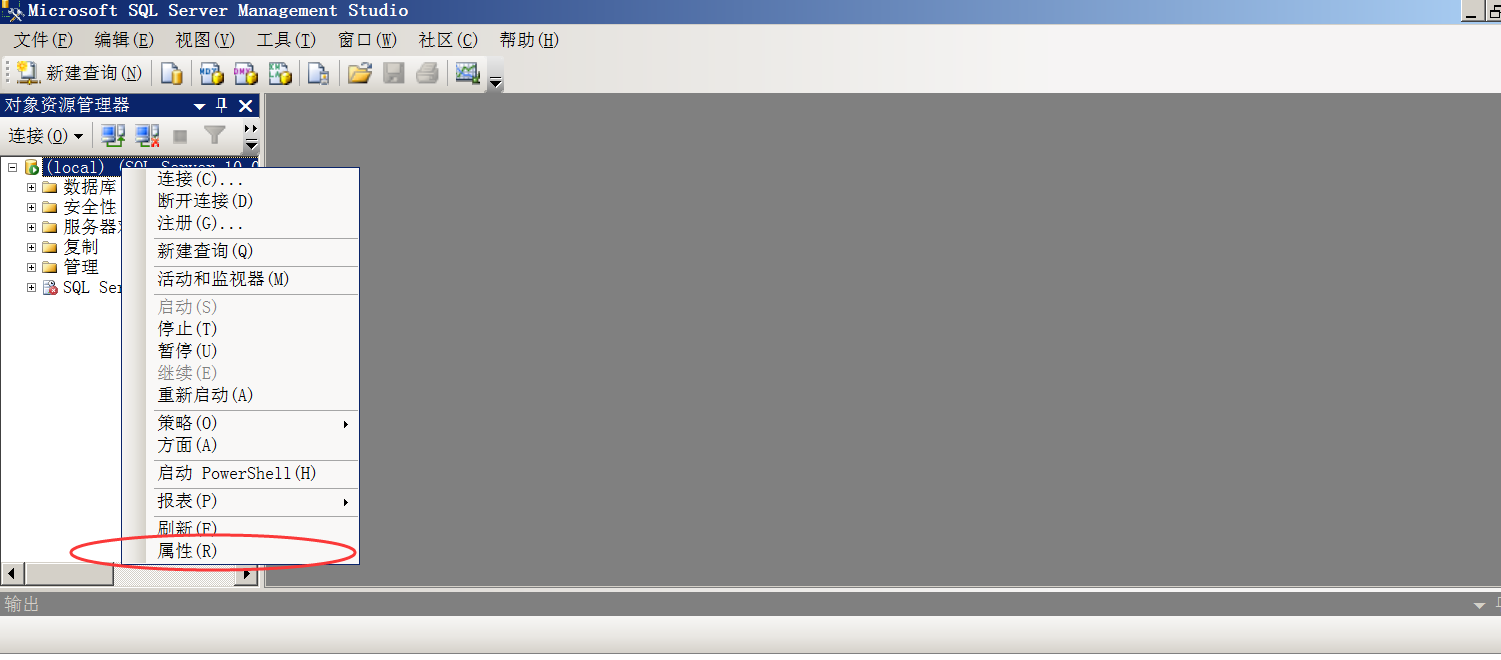


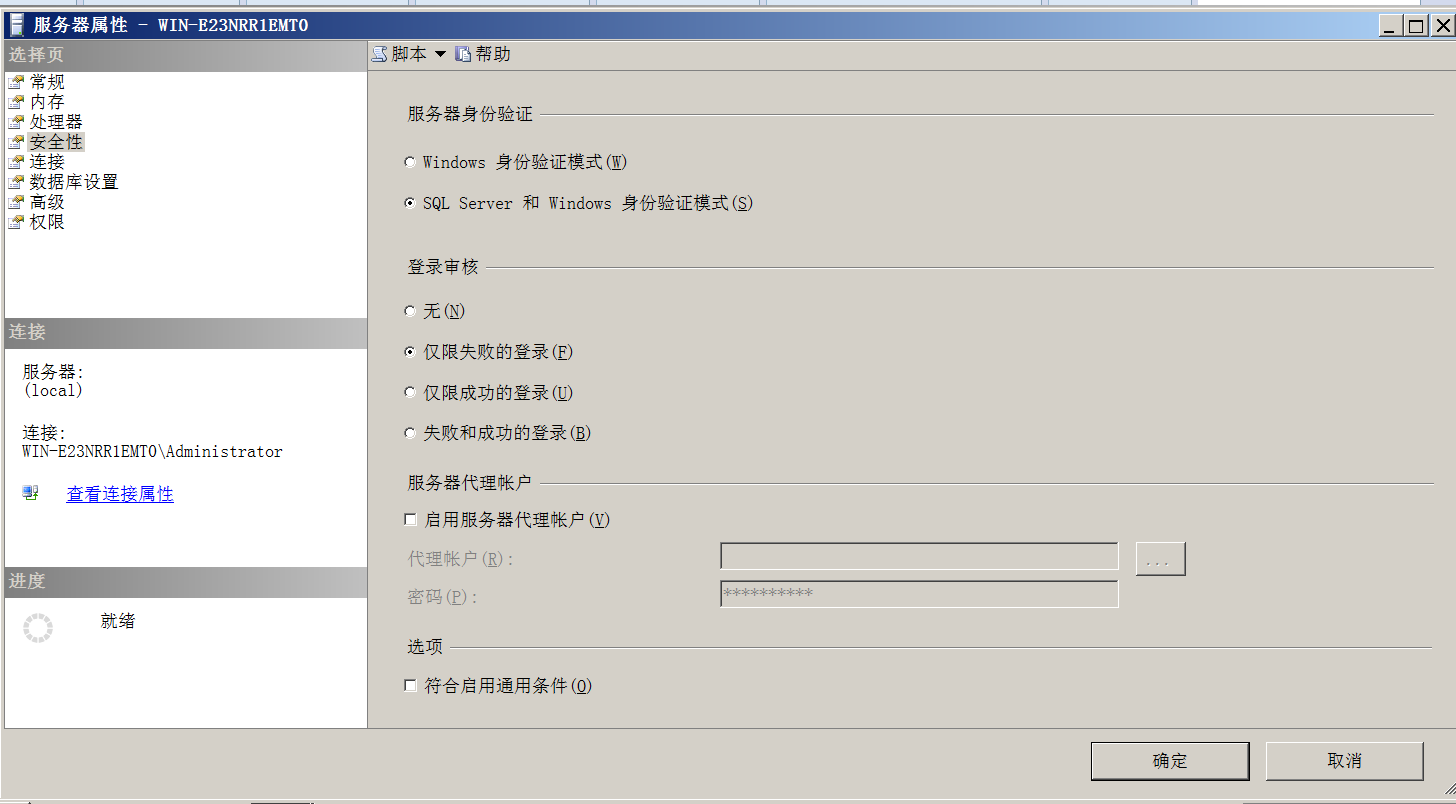
Windows身份认证看服务器名称

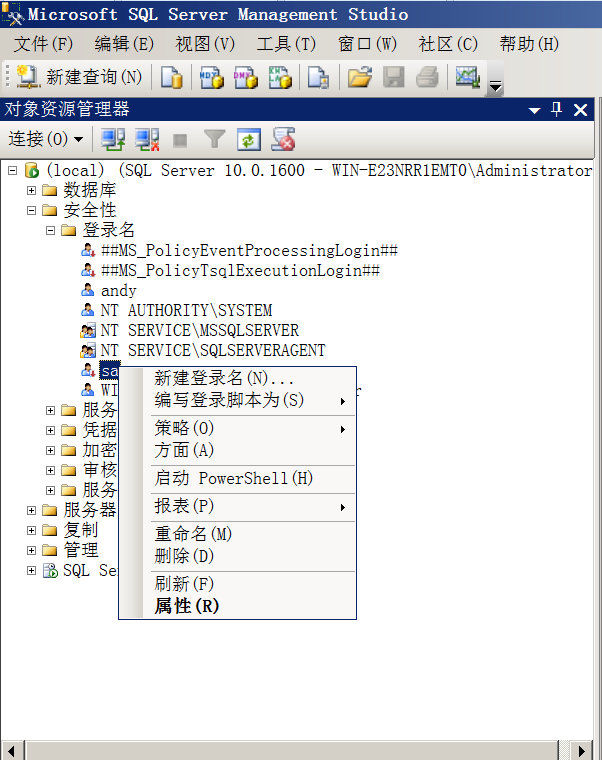


在“运行”里输入 sqlcmd，在命令提示符窗口输入EXEC sp\_addsrvrolemember 'WIN-E23NRR1EMT0/Administrator', 'sysadmin'，然后回车。

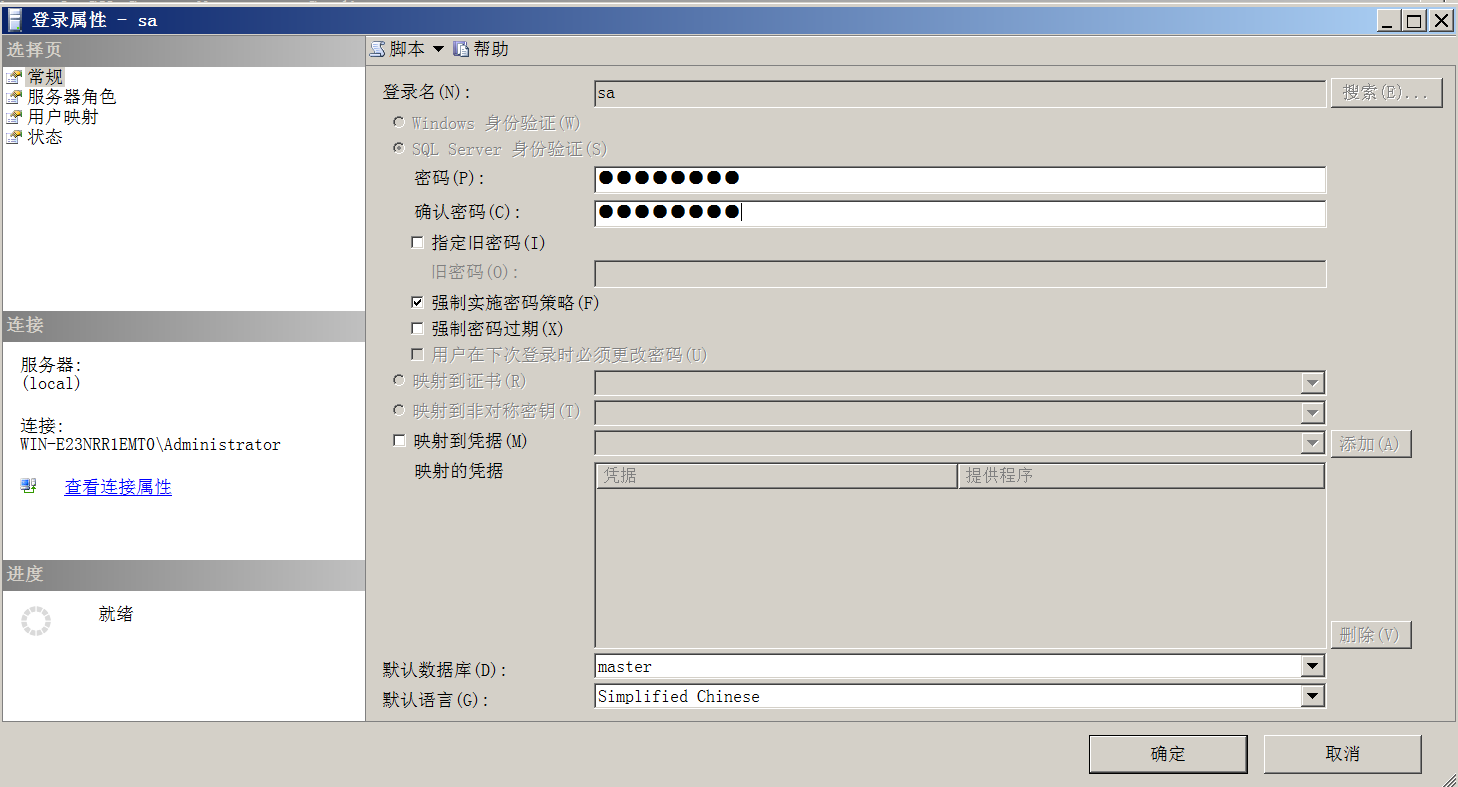
用Windows身份认证登录进去







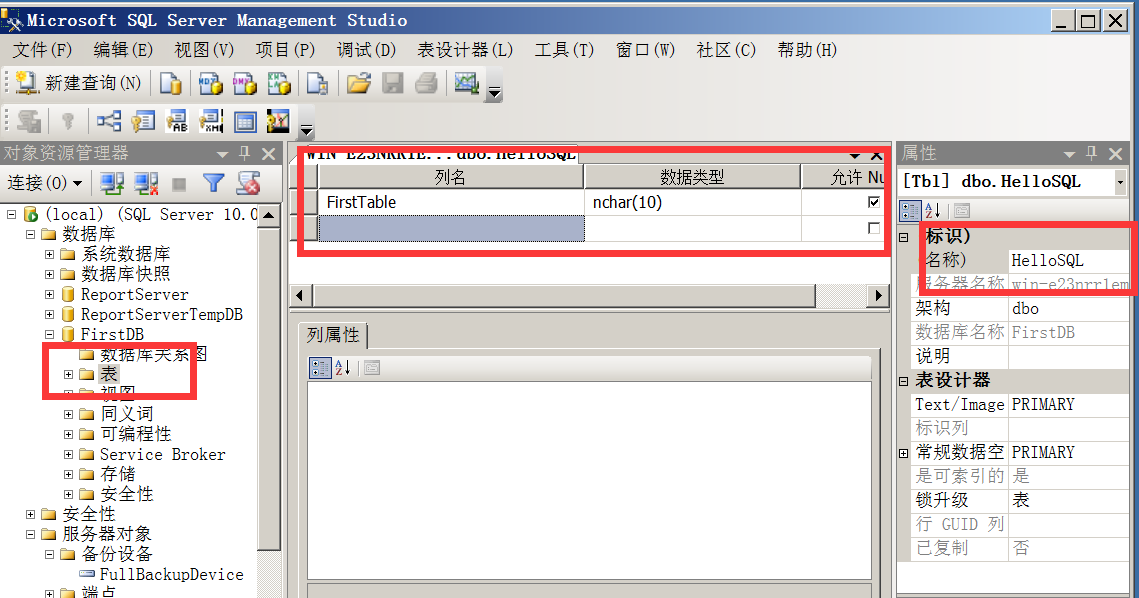
譬如改成9Mcc@2018



重启后密码生效

# 数据库对象维护

## 新建表



## 各个数据库文件空间使用统计

|  |
| --- |
| SELECT  DB\_NAME([database\_id]) AS [Database\_Name],  [file\_id],  name,  physical\_name,  type,  state,  is\_percent\_growth,  growth,  CONVERT(bigint, growth/128.0) AS [Growth\_in\_MB],  CONVERT(bigint, size/128.0) AS [Total\_Size\_in\_MB],  getdate() AS SAMPLETIME  INTO DGLJW\_DB\_FILE\_USEDRATIO  FROM  sys.master\_files WITH (NOLOCK)  ORDER BY  DB\_NAME([database\_id]) OPTION (RECOMPILE);  -- 定时统计插入  INSERT INTO DGLJW\_DB\_FILE\_USEDRATIO SELECT  DB\_NAME([database\_id]) AS [Database\_Name],  [file\_id],  name,  physical\_name,  type,  state,  is\_percent\_growth,  growth,  CONVERT(bigint, growth/128.0) AS [Growth\_in\_MB],  CONVERT(bigint, size/128.0) AS [Total\_Size\_in\_MB],  getdate() AS SAMPLETIME  FROM  sys.master\_files |

## 统计一个数据库里的各个表的统计信息

|  |
| --- |
| CREATE TABLE DGLJW\_SPACE  (  name VARCHAR(255),  ROWS BIGINT,  reserved VARCHAR(20),  DATA VARCHAR(20),  index\_size VARCHAR(20),  unused VARCHAR(20)  )  EXEC sp\_MSforeachtable "insert into DGLJW\_SPACE exec sp\_spaceused '?'"  SELECT \* FROM DGLJW\_SPACE ORDER BY name  DROP TABLE DGLJW\_SPACE |

## 查看一个数据库里有哪些对象

|  |
| --- |
| 跟Oracle不一样的是，要先选定数据库  select \* from INFORMATION\_SCHEMA.COLUMNS where table\_name = 'userdata\_1061943'  select top 10 \* from INFORMATION\_SCHEMA.TABLES where table\_name = 'userdata\_1061943'  select top 10 \* from INFORMATION\_SCHEMA.VIEWS where table\_name = 'userdata\_1061943' |

# 事务管理

## 查看正在运行的事务

|  |
| --- |
| select 'PercentComplete'=percent\_complete,  'OperationStartTime'=CONVERT(varchar(26),start\_time,120),  'EstimatedEndTime' = CONVERT(varchar(26),dateadd(ms,estimated\_completion\_time,getdate()),120),  session\_id,start\_time,status,  command,DB\_NAME(database\_id),  USER\_NAME(user\_id),blocking\_session\_id,  wait\_type,wait\_time,transaction\_id,  row\_count  from sys.dm\_exec\_requests  where estimated\_completion\_time <>0; |

# 访问

## 跨数据库访问

|  |
| --- |
| exec sp\_addlinkedserver 'DB\_FirstDB','','SQLOLEDB','192.168.0.2'  exec sp\_addlinkedsrvlogin 'DB\_FirstDB','false',null,'dbUser','dbPassword' |