****问题****  
在存储过程P\_TEMP执行结束后，T\_TEMP的表各个业务指标的汇总值是否同取数源DBA.T\_DDW\_XY\_JJZB\_D的业务指标汇总值保持同步或者在特定过滤条件和计算规则下合理存在。

****设计****  
编写SQL脚本，测试年/月的累计指标是否保持同步，其中，测试用例的数据日期为20170531

****测试代码方案****tLogin','secret'  
go

exec sp\_grantdbaccess 'testLogin'  
go

grant select,insert,delete on tblTestCases to testLogin  
go

grant select,insert on tblResults to testLogin  
go

第一步设定当前使用的数据库为SQL Server的主数据库（master database）。当创建一个新的用户数据库的时候，这一步是必不可少的。如果要创建的数据库已经存在，系统就会给出一个错误，为了防止这种错误，可以在试图删除新数据库的旧版本之前通过查询sysdatabases表来检查旧版本的数据库是否存在。然后可以通过数据库创建语句来创建用于测试用例存储的数据库。用于创建数据库的语句有许多可选参数，但是对于轻量级的自动测试来说，接受这些参数的默认值通常就可以满足需要了。  
接下来我们使用use语句把当前数据库设为新创建的数据库。这个步骤很重要，因为如果省掉了这一步，接下来所有的SQL语句都将在SQL主数据库上执行，这可能会导致很糟糕的结果。现在你可以创建一个表用来保存测试用例的输入。这个表的结构完全取决于你所要进行的测试，但是它至少应该包含测试用例ID、测试用例的输入项以及测试用例的期望结果项。对于用于存储测试结果的表，至少需要一个用于表示测试用例ID的列以及用于表示测试结果成功或者失败的列。如果想要在这张表里存放多次测试运行的结果（绝大多数情况下都会有这种需求），则需要采用某种方法来区分不同批次的测试运行结果。其中一种方法是用一个列来存放每次测试结果产生的日期和时间。这个列实际上隐式地充当了测试运行批次的ID。你也可以创建一个显式的、专用的列来表示测试运行批次ID。

****注解****SQL数据库支持两种不同的安全模式：使用Windows认证可以通过Windows账号ID和密码连接数据库，使用混合模式认证可以通过SQL登录ID和SQL密码来连接数据库。如果想要通过SQL认证来连接测试数据库，应该使用系统存储过程sp\_addlogin()创建一个SQL登录账号以及相应的密码。如果要删除一个SQL登录账号，可以先查询sysxlogins表检查相应的账号是否存在，然后调用sp\_droplogin()来删除这个登录账号。创建好一个SQL登录账号以后，应该赋予登陆账号连接数据库的许可。然后需要针对数据库里的表，赋予登录账号与SQL语句相关的权限许可，比如SELECT、INSERT、DELETE以及UPDATE。  
SQL登录账号很容易与SQL用户搞混。SQL登陆账号是服务器范围的对象，它用来控制针对装有SQL Server的机器的连接许可。而SQL用户是数据库范围的对象，它用来控制数据库以及它所包含的表、存储过程和其他一些对象的权限许可。当为一个SQL登陆账号分配权限时，会自动创建一个同名的SQL用户。所以最终会有一个SQL登陆账号和一个SQL用户，两个名字相同并且相互关联。尽管也可以让不同名字的登陆账号和用户相互关联，但是这太容易让人迷惑了，所以最好还是使用名字相同的默认机制。  
当为了测试存储过程而创建测试用例存储结构的时候，你必须决定在什么时候以及如何往表中插入用于测试用例的那些数据。最简单的办法是在创建表的时候直接加入用于测试用例的数据。但是，在测试的过程中几乎肯定要在很多地方碰上需要添加或者移除测试用例数据的情况，所以更灵活的方法是使用BCP（Bulk Copy Program）、DTS（Data Transformation Services）或者一个辅助的C#程序来插入数据。如果想要插入以及删除测试用例数据，那么就应该针对存储测试用例数据的那张表为SQL登陆账号赋予INSERT和DELETE的权限许可。  
这个用于创建测试用例和存储测试结果的脚本可以通过几种方法来运行。一种方法是在查询分析器程序里打开并通过Execute命令（或者快捷键F5）执行这个脚本。第二种方法是使用osql.exe程序来执行这个SQL脚本。