

SQLServer存储过程编程经验技巧Microsoft认证考试 PDF转换
可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao_ti2020/562/2021_2022_SQLServer_E](https://www.100test.com/kao_ti2020/562/2021_2022_SQLServer_E5_c100_562278.htm)

[5_c100_562278.htm](https://www.100test.com/kao_ti2020/562/2021_2022_SQLServer_E5_c100_562278.htm) S Q L S e r v e r 目前正日益成为W i n d o w N T 操作系统上面最为重要的一种数据库管理系统，随着 S Q L S e r v e r 2 0 0 0 的推出，微软的这种数据库服务系统真正的实现了在W i n d o w s N T / 2 0 0 0 系列操作系统一统天下的局面，在微软的操作系统上面，没有任何一种数据库系统能与之相抗衡，包括数据库领域的领头羊甲骨文公司的看家数据库系统O r a c l e 在内。不可否认，S Q L S e r v e r 最大的缺陷就是只能运行在微软自己的操作系统上面，这一点是S Q L S e r v e r 的致命伤口。但在另一方面却也成了最好的促进剂，促使S Q L S e r v e r 在自己仅有的“土地”上面将自己的功能发挥到了极至，最大限度的利用了N T 系列操作系统的各种潜能！作为S Q L S e r v e r 数据库系统中很重要的一个概念就是存储过程，合理的使用存储过程，可以有效的提高程序的性能；并且将商业逻辑封装在数据库系统中的存储过程中，可以大大提高整个软件系统的可维护性，当你的商业逻辑发生了改变的时候，不再需要修改并编译客户端应用程序以及重新分发他们到为数众多的用户手中，你只需要修改位于服务器端的实现相应商业逻辑的存储过程即可。合理的编写自己需要的存储过程，可以最大限度的利用S Q L S e r v e r 的各种资源。下面我们来共同看看各种编写S Q L S e r v e r 存储过程和使用存储过程的技巧经验！下面我们讨论的前提都是您有过一定的S Q L S e r v e r 存储过程的

编写经验，下面的各种技巧如果没有特别注明，则同时适用于SQL Server 7.0和SQL Server 2000。

1、使用OUTPUT类型参数的存储过程的技巧 一般的存储过程都是直接返回一段记录集给调用者，但是有的时候我们只需要一些存储过程返回的一些参数的值，这时候可以指定存储过程的OUTPUT参数，比如：
`Create procedure GetName @uid nvarchar(1), @username nvarchar(10) = '' output AS set @username = 'hongchao'` GO 在上面的存储过程当中我们传递进的参数是@uid，而参数@username则在调有的时候不需要传递进去。这样，存储过程会返回给我们参数@username的值是'hongchao'。上面的都比较的容易，需要注意的是，当在SQL 2000中的时候，如果你的存储过程只有一个参数，并且这个参数是OUTPUT类型的，你必须在调用这个存储过程的时候给这个参数一个初始的值，否则会出现调用错误的情况！

2、在存储过程中的书写注意事项 这一点在SQL Server 7.0和SQL Server 2000中有些地方是不一样的，也不知道是不是微软的遗漏，那就是有些系统的关键字在不同的版本之间有所不同，比如关键字level，同样的一句话：`select * from users where level = 1` 在SQL Server 7当中的存储过程当中运行没有丝毫的问题，但是到了SQL Server 2000当中则会出现运行的错误，原因就是在于在SQL Server 2000中“level”被当作了关键字（奇怪的是

SQL 7 当中也同样是关键字，却没问题），所以在 SQL 2000 当中，上面的语句应当改为：`select * from users where [level] = 1` 从上面的例子中我们可以看到，在你编写存储过程的时候，最好在有可能和系统关键字的地方使用“[”和“]”将他包围起来，以避免在移植过程中出现的运行错误问题。

3、在存储过程中使用系统存储过程 `SP__Executesql` 的注意事项 我们在编写自己的存储过程的时候，往往在很多的情况下，会使用到系统的存储过程 `SP__Execute`。但是需要的是，如果你在这个存储过程的参数（一般是一段 SQL 语句）当中进行了临时 `Table` 的操作，那末对于调用者来说，这个临时 `Table` 是不可见的，也就是说你无法通过临时 `Table` 来在调用者和被调用者之间传递值。解决的方法是使用全局临时 `Table`，也就是“##”开头的 `Table`。

4、在存储过程中使用临时 `Table` 和游标的注意事项 如果我们的商业逻辑比较复杂，在存储过程当中，就需要一些媒介作为中转台，这时候临时表就发挥了作用，但是请务必记得在使用完之后，即时删除使用到的临时 `Table`。而在存储过程当中想要依次遍历一个记录集的唯一方法就是使用系统游标，同样要注意的是，在使用完成之后及时关闭和销毁游标对象释放他用到的资源。并且不在万不得已的情况下，不要随意使用游标，因为他会占用较多的系统资源，尤其是对于大并发量的情况下，很容易使得系统资源耗尽而崩溃。使用临时 `Table` 和游标各有利弊，在使用的过程中要适当的利用即可！

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com