解析用 MA移植Acceses到SQL几点问题 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/284/2021_2022__E8_A7_A3_E6_9E_90_E7_94_A8__c97_284919.htm 这些年来，Access数据库一直在PC平台占据主导地位，使用它建立了大量的部门数据库。随着这些数据库的应用，它们中的大多数已经慢慢地具有应急使命，现在需要的是加固成为一个安全的客户端服务器引擎。 在微软想要统治世界的伟大计划中，更希望这种引擎是SQL Server。随着这种想法，微软针对Access提供了免费的SQL Server移植工具SSMA。 对于开发者来说，移植工具已有很大的实惠。但期望这种工具能够移植整个应用程序是不现实的，因为Access有一些SQL Server所没有的简单工具（例如窗体和报表性能）。但是我们有理由相信这种工具能做大部分工作，比如建立适当的表，转移数据，把查询转换成视图等。 SSMA的运行需要在.NET Framework2.0版本以上，J#2.0可重组包以及至少1GB RAM。 SSMA具有一个清晰的图形用户界面，分成四个面板。在建立一个新工程之后，首先添加一个或多个Access数据库，然后连接到适当的SQL Server数据库，下一步就是把架构（schema）转换成SQL Server。 注意，这个过程并不是运行依靠SQL Server引擎的架构，而是简单地生成了一个在SSMA中可见的，可用的SQL Server架构，同时生成一个错误、警告和信息标记的集合。 从这点来看，该工具的能力就显而易见。作为一个开始，这些标记指出转换问题，例如：不支持Access的一些函数如DateDiff，所以不能转换（当然这些函数可以被转换，但SSMA不能实现）。 你可以浏览Access架构，观察正在计划

的类型映射等等，当然如果你不喜欢这种缺省映射，也完全可以改变它，或者根据特殊的工程甚至特殊的表来做改变。查询是一个比较特别的情形。它们被转换成SQL Server视图：你可以编辑Access查询然后产生适当的SQL Server代码。这样的编辑是发生在SSMA的架构中，而不是在Access数据库本身完成。 你可以使用SSMA运行依靠数据库的SQL Server架构，它建立了一种结构来保存数据以便你可以移植数据。理论上听起来很好，但是实际上是怎样的呢？虽说尝试从任意一个数据库引擎移植到另一个都是麻烦的，且这个工具可以免费的为你做90%的工作，但它还存在一些缺陷。 例如，虽然不是SQL标准的一部分，Access需要所有日期来包装到hash记号中。不幸的是，SSMA看起来没有考虑到这点，这个疏忽的结果就是所有涉及到日期的查询结果都不能成功转换。下面是一个错误信息的例子： /* * SSMA error messages: * A2SS0058: Following SQL statement is not supported and cannot be converted: * * SELECT DISTINCTROW EMPLOYEES.EmployeeNo, EMPLOYEES.FirstName, EMPLOYEES.LastName, EMPLOYEES.DateOfBirth, EMPLOYEES.DateEmployed * FROM EMPLOYEES * WHERE ((EMPLOYEES.DateOfBirth)>#1/1/1970#). * */PRINT ’ERROR: SSMA failed to convert the previous statement.’日期在数据库中是很常见的，所以这个疏忽将会影响大多数数据库转换。但要解决并不困难，如下： SELECT EmployeeNo, FirstName, LastName, DateOfBirth FROM dbo.EMPLOYEES WHERE (DateOfBirth > CONVERT(DATETIME, ’1970-01-01’)) 从例子中返回正确的数据集。 （我们可以讨论一下是否是这样，

例如：CONVERT(DATETIME, ’1970-01-01 00:00:00’, 102)可能更恰当，但是不管怎么说，我们可以转换数据处理），如果我们可以手动地做，SSMA就应该可以为我们做这件事。还有更糟糕的问题：Access默认是在文本周围使用双引号，例如： SELECT EMPLOYEES.EmployeeNo, EMPLOYEES.FirstName FROM EMPLOYEES WHERE ((EMPLOYEES.FirstName="Norma")). SQL Server不是这样，它使用单引号，如下：WHERE EMPLOYEES.FirstName= ’Norma’.然而，SSMA保留了上面这样的双引号代码，没做任何改变。而且在架构产生期间并没有引发错误提示，错误提示只发生在把架构加载到SQL Server数据库的过程中。那时，SSMA抛出一个错误提示说存在一个非法列名Norma，这样视图就不能加载到SQL Server中了。以上显示出SSMA并没有做足够的语法检查。 再强调一下，Access默认使用双引号，而SSMA却不能处理如此简单平常的Access语法。这就像一个法语到英语的翻译者可以处理语言中的大多数词，却为单词“ Bonjour ”感到束手无策一样。 再一个例子，Access允许为字段添加规则约束，例如一个名为“ Title ”的字段可以接受的值可能只有Mr., Mrs., Miss., Ms等。但SQL Server不能准确地支持同样的类型约束。非常明显SSMA转换这种约束规则为一个表约束。Brilliant，做的不错，只是在代码中丢失了字段名： ALTER TABLE [dbo].[NAMES] ADD CONSTRAINT [SSMA_CC$NAMES$Title$validation_rule] CHECK (In (’Mr.’, ’Mrs.’, ’Miss’, ’Ms’, ’Dr.’, ’Prof.’))这不仅不能在架构转载到SQL Server时运行，同时更不能产生一个错误消息。最后一行的正确语法应该是:CHECK (Title In (’Mr.’,

’ Mrs. ’ , ’ Miss ’ , ’ Ms ’ , ’ Dr. ’ , ’ Prof. ’)) 那么，我们是否应该从机器上删除SSMA呢？当然不，因为它确实完全自动地做了大量的工作，也提供了一个合理的环境，在那里可以看到问题区域并做出整理。指出它的缺陷，只是期望SSMA能更好。 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