

学校的理想装备

电子图书·学校专集

校园网上的最佳资源

将 Access 数据库移植到

Microsoft SQL Server 7.0



将 Access 数据库移植到 Microsoft SQL Server 7.0

1999 年 2 月

微软公司

摘要：本文描述了将 Access 数据库转换到 SQL Server 数据库的过程和工具

简介

随着用户对于企业级高性能数据库的需求的增长，用户时常要从 Microsoft Access Jet 引擎的文件-服务器环境下转换到 Microsoft SQL Server 的客户-服务器环境。Microsoft Office 2000 中的 Access 2000 Upsizing Wizard 可实现将数据表和查询转移到 SQL Server 7.0 中。如果您用的是 Access 的较早的版本，您可以先将它升级为 Access 2000，然后再使用其中的 Upsizing Wizard，从而将您的应用移植到 SQL Server 中。

如果您并不太愿意采用 Access 2000 和 Upsizing Wizard 来实现移植，本文可以作为将 Access 2000 移植到 SQL Server 的一个指南。转移一个 Access 上的应用首先需要将数据转移到 SQL Server，然后将查询转移进数据库，或是转移为 SQL 文件以备稍后执行。最后要采取的步骤是移植应用程序。

数据库移植中用到的 SQL Server 工具

SQL Server 管理器 (SQL Server Enterprise Manager)

SQL Server 管理器 允许对 SQL Server 以及 SQL Server 中的对象进行企业级的配置和管理。SQL Server 管理器提供一个强有力的 scheduling 引擎，高度的容错力和一个嵌入式的复制管理界面。使用 SQL Server 管理器可以实现以下功能：

管理连接和用户许可

创建脚本程序

管理 SQL Server 对象的备份

备份数据和事务处理日志

管理表、视图、存储过程、触发器、索引、规则、默认值以及用户定义的数据类型

建立全文本索引、数据库图表和数据库维护计划

输入和输出数据

转换数据

执行各种网络管理任务

在以 Microsoft Windows NT 为操作系统的计算机中，SQL Server Manager 由 SQL Server Setup 进行安装，并被默认为服务器组件，而在运行着 Windows NT 和 Microsoft Windows 95 的机器上，它将被默认为客户方组件。您将从 SQL Server Manager 的图形用户界面中启动数据转移服务 (DTS, Data Transformation Services)。

数据转移服务 (Data Transformation Services, DTS)

数据转移服务允许您在多种异构数据源之间输入和输出数据，这些数据源采用基于数据库的 OLE 体系结构；或在使用 SQL Server 7.0 的多个计算机之间转移数据库和数据库对象；您还可以通过运用数据转移服务，更便捷地在一个在线事务处理系统（OLTP）中建立数据仓库和数据中心。

DTS Wizard 允许您交互地创建 DTS 包，通过 OLE DB 和 ODBC 来输入、输出、验证和转移数据。DTS Wizard 还允许您在关系型数据库之间拷贝图解（schema）和数据。

SQL Server 查询分析器（Query Analyzer）

SQL Server 查询分析器是一种图形化的查询工具，通过它您可以分析一个查询，同时执行多个查询，查看数据和获取索引建议。SQL Server 查询分析器提供了 showplan 选项，用来显示 SQL Server 查询优化器所选择的数据提取方法。

SQL Server Profiler

SQL Server Profiler 可以实时地捕获数据库服务器活动的连续记录。SQL Server Profiler 允许您监控 SQL Server 产生的事件，过滤基于用户指定标准的事件，或将操作步骤输出到屏幕、文件或数据表。运用 SQL Server Profiler，您可以重新执行所捕获的上一次操作。这种工具可以帮助应用程序开发者识别那些可能会降低应用程序性能的事务处理。在将一个基于文件体系结构的应用程序移植到客户/服务器结构中时该特性是很有用的，因为它的最后一步包括对面向新的客户/服务器环境的应用程序进行优化。

转移表和数据

使用 DTS Wizard 将您的 Access 数据转移到 SQL Server，可采取以下步骤：

在 SQL Server Manager（Enterprise Manager）的工具菜单中，鼠标指向“Data Transformation Services”，然后点击“Import Data。”

在“选择数据源”（Choose a Data Source）的对话框中，选择 Microsoft Access 为数据源，然后输入您的 .mdb 文件名（mdb 为文件扩展名）或者选择浏览文件。

在“选择数据目标”（Choose a Destination）的对话框中，选择“Microsoft OLE DB Provider for SQL Server”，再选择好数据库服务器，然后点击所需的认证模式。

在“指定表备份或查询”（Specify Table Copy or Query）的对话框中，点击“拷贝表”（Copy tables）。

在“选择数据源”的对话框中，点击“选择所有项”（Select All）。

移植 Microsoft Access 查询

您可以将 Access 的查询以下面的格式之一转移至 SQL Server 中：

事务处理 SQL 脚本程序（Transact-SQL scripts）

事务处理 SQL 语句通常是由数据库程序调用的，但是您也可以使用 SQL Server 7.0 中包含的 SQL Server 查询分析器直接运行它们。SQL Server 查询分析器可帮助开发者测试事务处理 SQL 语句，或运行那些执行查询处理、数据操作（插入，修改，删除）和数据定义（创

建表)的事务处理 SQL 语句。

存储过程 (Stored procedures)

开发者可以将大部分产生自 Access 查询 (查找, 插入, 修改, 删除) 的事务处理 SQL 语句转移至存储过程。用事务处理 SQL 语句书写的存储过程可以用来对您的数据存取打包, 并使之标准化, 而且存储过程实际上是存储在数据库中的。存储过程可以带参数, 也可不带参数, 可以由数据库程序调用或者由 SQL Server 查询分析器手动执行。

视图 (Views)

视图是从一个或多个表中显示特定的行和列的虚拟表。它们允许用户可以不直接执行构成查询基础的复杂连接而建立查询。视图不支持参数的使用。连接多个数据表的视图不能用 INSERT, UPDATE 或 DELETE 语句来修改。视图由事务处理 SQL 语句调用, 也可用于 SQL Server 查询分析器中运行的程序段。SQL Server 视图和 SQL-92 标准不支持视图中的 ORDER BY 排序子句。如欲了解事务处理 SQL, 存储过程和视图的其他信息, 请参阅 SQL Server 在线参考书。

Access 查询类型的 SQL Server 移植选择与建议

一个 SELECT 语句可以存储在事务处理 SQL 文件、存储过程或是视图中。建立存储过程是将数据库应用开发与数据库设计的物理实施分开的最佳方法。存储过程可在一处创建而由应用程序调用。

如果存储过程所基于的数据库变化了, 而存储过程经过仔细的修改以反应这些变化, 则对存储过程的调用将不会受到破坏。

交叉表 (CROSSTAB)

交叉表经常用于总结报表。

一个 Access 的交叉表可以通过 SQL 程序段、存储过程或视图中的事务处理 SQL 语句来执行。每当发出一个查询时, 数据连接被重现执行以确保最近的数据得到使用。

根据实际应用情况, 比较合适的方法是将交叉表中的数据存储为一个临时表 (参考下面的 MAKE TABLE), 临时表对资源的需求比较少, 但是临时表在建立的同时只提供对数据的一个快照 (snapshot)。

创建表 (MAKE TABLE)

Access 中的 “MAKE TABLE”(创建表) 可以通过事务处理 SQL 脚本程序或存储过程中的事务处理 SQL 语言的建表语句 “CREATE TABLE” 来执行。语法如下所示:

```
SELECT [ ALL | DISTINCT ]
[ {TOP integer | TOP integer PERCENT} [ WITH TIES] ]
<select_list>
[ INTO new_table ]
[ FROM {<table_source>} [,...n] ]
[ WHERE <search_condition> ]
```

```
[ GROUP BY [ALL] group_by_expression [,...n]
[ WITH { CUBE | ROLLUP } ]
CREATE TABLE mytable (low int, high int)
```

UPDATE (修改)

UPDATE 语句可以存储在事务_SQL 脚本程序中，然而比较好地执行 UPDATE 语句的方法是创建一个存储过程。

APPEND (添加)

APPEND 语句可以存储在事务_SQL 脚本程序中，然而比较好地执行 APPEND 语句的方法是创建一个存储过程。

移植 Microsoft Access 的查询到存储过程和视图

每个 Access 查询都必须用以下的一系列语句替换：

```
CREATE PROCEDURE <NAME_HERE> AS
< SELECT, UPDATE, DELETE, INSERT, CREATE TABLE statement from Microsoft Access >
GO
```

```
CREATE VIEW <NAME_HERE> AS
<Place (SELECT only, with no parameters) Microsoft Access Query>
GO
```

对每个 Access 查询应执行：

打开 Access，然后在 SQL Server 中，打开 SQL Server 查询分析器。

在 Access 的数据库窗口中点击“Queries”tab 键，然后点击“Design”按钮。

在“View”菜单上点击“SQL”按钮。

将整个查询粘贴到 SQL Server 查询分析器中。

测试语法，保存事务处理 SQL 语句以备后用，或者在数据库中执行这些语句。您可以选择将事务处理 SQL 语句保存到一段脚本程序中。

移植 Microsoft Access 查询到事务处理 SQL 语句

大部分的 Access 查询应该转换成存储过程和视图。然而，有一些应用程序开发者不太常用的语句可以存储为事务处理 SQL 脚本，一种以 sql 为文件扩展名的文本文件。这些文件可以在 SQL Server 查询分析器中运行。

如果您打算将一些 Access 查询转换为 sql 文件的话，可以考虑根据它们使用的方式有区别地将这些事务处理 SQL 语句分别放在几个脚本程序中。例如，您可以将必须以同样频率运行的事务处理 SQL 语句归类到同一个脚本中。另一个脚本中则应包含所有只在某些条件下运行的事务处理 SQL 语句。此外，必须以一定顺序执行的事务处理 SQL 语句应当归类到一个不连续的脚本中。

将 Access 语句转移到事务处理 SQL 文件

将语句拷贝到 SQL Server 查询分析器中

使用蓝色的多选项图标分析语句

在适当时候执行该语句

要执行 Access 中的创建表 (MAKE TABLE) 的查询任务的开发者在 SQL Server 中有几种选择。开发者可创建下列对象之一：

一个视图

视图具有动态的虚拟表的效果，可提供最近的信息。这是一个输入/输出强化器，因为每当发出一个查询时它都要求对数据表重现建立连接。

一个临时表

临时表为已连接的用户会话建立一个快照。您可以建立局部的或全局的临时表。局部临时表只在当前会话中可见，而全局临时表则在全局都是可见的。在局部临时表的名字前加上单个数字的前缀 (#table_name)，而在全局临时表的名字前加上两位数字的前缀 (##table_name)。对临时表的查询执行起来非常快，因为它们取得一个结果集的时候通常只用一个表，而不是将多个表动态地连接在一起。

如欲了解临时表的其他信息，请参阅 SQL Server 在线参考书。

SQL Server 7.0 中的数据转换服务 (DTS) 允许您通过创建包来实现临时表建立的标准化、自动化和进度安排。例如，当您移植 Access 2.0 中的 Northwind 范例数据库时，用于季度数据报表的交叉表可转变为一个视图或者一个可在规范基础上建立临时表的数据转换。如欲了解关于 DTS 的其他信息，请参阅 SQL Server 在线参考书。

其他设计上的考虑

下面是当您您的 Access 应用移植到 SQL Server 时必须考虑的一些其他问题：

使用参数

带参数的 SQL Server 存储过程需要一种不同于 Access 查询的语法格式，例如：

Access 2.0 格式：

查询名：Employee Sales By Country, in NWIND.mdb:

```
PARAMETERS [Beginning Date] DateTime, [Ending Date] DateTime;
SELECT Orders.[Order ID], [Last Name] & ", " & [First Name] AS Salesperson,
Employees.Country, Orders.[Shipped Date], [Order Subtotals].Subtotal AS [Sale
Amount]
FROM Employees INNER JOIN (Orders INNER JOIN [Order Subtotals] ON Orders.[Order ID]
= [Order Subtotals].[Order ID]) ON Employees.[Employee ID] = Orders.[Employee ID]
WHERE (((Orders.[Shipped Date]) Between [Beginning Date] And [Ending Date]))
ORDER BY [Last Name] & ", " & [First Name], Employees.Country, Orders.[Shipped Date];
```

SQL Server 7.0 格式：

```
CREATE PROCEDURE EMP_SALES_BY_COUNTRY
@BeginningDate datetime,
@EndingDate datetime
AS
SELECT Orders.[Order ID], [Last Name] + ", " + [First Name] AS Salesperson,
Employees.Country,
Orders.[Shipped Date], [Order Subtotals].Subtotal AS [Sale Amount]
FROM Employees INNER JOIN (Orders INNER J
```

