*Microsoft*Press
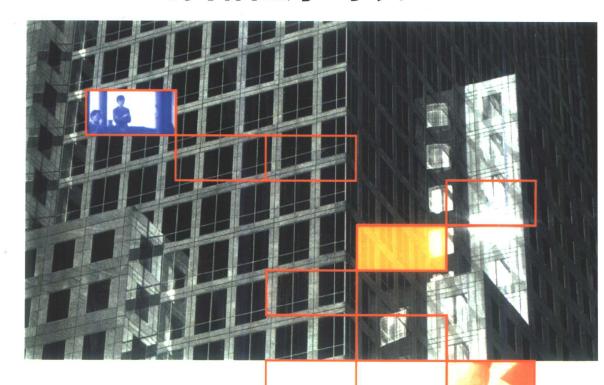
T-SQL 存储程序与表 （影印版）

T-SQL Stored Procedures and Tables Reference

David Iseminger
Series Editor

北京大学出版社
http://cbs.pku.edu.cn

T-SQL 存储程序与表
（影印版）

[美] David Iseminger 著

内 容 简 介

SQL Server 2000 存储程序通过管理 Active Directory 服务、核心维护任务、复制、全文检索、安全、系统维护、网络辅助任务、XML 文本、SQL 代理服务器等，可以实现 Microsoft SQL Server 的许多例程管理和信息活动；本书还提供了 SQL Server 2000 存储程序的有关内容，通过对本书的学习，可以掌握其使用方法。

本书由微软公司专家编写，实用性强，是系统管理员和程序开发人员的必备参考书。

出 版 前 言

如果用一个成语来概括国内计算机图书市场的现状，当谓之"汗牛充栋"。然而，如果您是一位从事计算机应用系统开发或管理的中、高级专业人士，很可能发现这貌似种类齐全的计算机图书中，为您量身定做的并不多见。

依据多年从事计算机图书工作所积累的经验，以及与 IT 领域广泛而深入的接触所获取的信息，我们认识到，具有相当的专业深度和技术前沿性的图书，是计算机专业人员的迫切需要，当然，也是我们从事计算机图书工作、服务专业领域的一大着眼点。

基于这一点，2000 年元月，我们与微软出版社(Microsoft Press)达成合作协议，成立微软图书影印中心，独家代理微软出版社图书影印版在中国大陆的出版、发行，为 IT 业界提供及时的专业技术服务。选题和策划上的匠心独运，使得我们的影印书成为计算机图书中的标新立异者。这里，有四大特色值得读者朋友予以关注：

首先，这是微软出版社第一次授权在中国大陆影印、发行它的版权书。在选题上，可以说独辟蹊径。在内容上，立足技术广度和深度，系统推介微软产品。所有这些，都是目前国内一般计算机图书所无法比拟的。

其次，我们的理念是为国内计算机专业人员学习前沿性的微软技术提供服务。为此，我们不但与微软公司紧密协作与沟通，及时掌握微软最新技术动向，而且组织了精干的工作人员，倾力于微软影印书的出版和发行。

再者，微软影印书主要面向中、高级专业人员，印量有限。这类书的读者对象有较强的针对性，一般来说，包括 IT 决策人员，中、高级开发人员，以及中、高级系统管理人员。因而，我们将每套书的印数控制在 1000~2000 册之间。

最后，微软图书影印版几乎与原版书保持同步发行，最大限度地满足了国内读者跟踪微软最新技术的需求。软件升级越来越快，新软件令人目不暇接。作为技术载体之一的图书，只有迅速作出反应，把新软件介绍给读者，才能赢得他们的青睐。总之，兵贵神速，这是我们的目标。

正应验了前人的预言，21 世纪是一个信息时代。软件作为信息系统的神经，在我们生活的这个时代里发挥着举足轻重的作用，而微软公司和它推出的各种软件，更是令世人为之瞩目。我们将立足图书，继续并扩大与微软公司的合作，在中国信息产业的发展道路上留下自己的足迹。

出版者
2000 年 10 月

Contents

Introduction

Imagine buying your brand-new washing machine and dryer at Costco, only to remember you brought the Miata—that's how I felt while putting together the *SQL Server 2000 Reference Library*. All told, there was about three times as much SQL Server 2000 Books Online reference material as there was room in the *SQL Server 2000 Reference Library*, so I had to figure out the most widely used, appropriate, and pertinent parts of Books Online, and have included those parts in the library you now have in your possession.

Despite those woes, I believe you'll find the *SQL Server 2000 Reference Library* full of must-have information about SQL Server 2000. From architecture and XML, to Analysis Services (formerly OLAP) and T-SQL, this reference library contains the essential reference information you need to program, administer, deploy, or optimize your SQL Server 2000 solution... without monitor-induced eyestrain.

The SQL Server 2000 Reference Library is part of the Windows Programming Reference Series (WPRS), a series of libraries dedicated to providing printed development and IT material in a timely, intelligently organized, and well-conceived manner. You can find out more about WPRS and other available reference libraries (including the *COM+ Developer's Reference Library* and the *Active Directory Developer's Reference Library*) at www.iseminger.com—a website dedicated to providing additional information about the series, and other books also by yours truly.

How the SQL Server 2000 Reference Library is Structured

The *SQL Server 2000 Reference Library* consists of six volumes, each of which focuses on one or more specific areas of SQL Server 2000. These guides and programming reference volumes have been divided into the following:

Volume 1: SQL Server 2000 Architecture and XML/Internet Support

Volume 2: Database Creation, Warehousing, and Optimization

Volume 3: Analysis Services

Volume 4: Replication and English Query

Volume 5: T-SQL Language Reference

Volume 6: T-SQL Stored Procedures and Tables Reference

Dividing the *SQL Server 2000 Reference Library* into these categories enables you, the reader, to quickly identify the volume you need, based on your task, and facilitates your maintenance of focus for that task. This approach allows you to keep one reference book open and handy, or tucked under your arm while running between server racks.

In addition to the overall library structure, each volume in the *SQL Server 2000 Reference Library* is divided into parts that concentrate on a given subject. In order to provide a quick overview of a part's contents, each begins with a Part Introduction page that outlines what you'll find therein.

Finding Related Topics and Working With the Topic Index

Throughout this library, you're going to see references to related topics; some of them within the text of a given paragraph, and others placed in a special section called **Related Topics**. Since the entire body of SQL Server 2000 Books Online constitutes more topics than what you'll find in these volumes, you may occasionally come across a referenced topic that doesn't correspond to a section in this library. Don't worry; you can get to that information through the SQL Server 2000 Books Online. Most of the references *will* pertain to items in these volumes, but there might be some reference you're interested in that will lead you online.

To make locating topics as easy as possible, and to enable you to quickly identify which topics are found in this library (versus which are found only in Books Online), Volumes 1 through 4 include a special index called **Topics in the SQL Server 2000 Reference Library**. There's some good information about the topic index (as I'll refer to it from now on) that will help you understand how to get the most use out of it and out of the *SQL Server 2000 Reference Library*.

For starters, the topic index contains *topics only found in the SQL Server 2000 Reference Library*. So, say you're reading through a chapter in this library, and you come across text that refers you to another topic, such as the following:

Related Topics

Building and Processing Cubes, Updating Cubes and Dimensions

Or something like the following:

> …For more information about which editions support which features, see Features Supported by the Editions of SQL Server 2000.

You can then look through this alphabetical listing of topics included in this library for the referenced topic, and when you find it, you'll be directed to the Volume and Chapter where that topic can be found, as shown here:

If two topics with the same title exist (as shown here), each will be listed separately. As previously mentioned, most references refer to topics also found in this library, but for those that don't, you can go to Books Online to get more information.

There are important exceptions to topic references and the topic index: *T-SQL statements and other programming elements are **not** included in the topic index.* Each volume that contains programming elements (such as T-SQL stored procedures, or English Query statements) has its own index of programming elements. Rather than cluttering this index up with programming elements (or vice versa), I've provided separate indexes to help you find the information you need faster.

Since Volumes 5 and 6 are almost entirely dedicated to statement definitions, including the topic index in those volumes didn't seem like a good use of (precious) pages.

The Idea Behind SQL Server 2000 Reference Library

The *SQL Server 2000 Reference Library*, like all libraries in the Windows Programming Reference Series, is designed to deliver the most pertinent information in the most accessible way possible. The *SQL Server 2000 Reference Library* is also designed to integrate seamlessly with SQL Server 2000 Books Online (and with MSDN Online) by providing a look-and-feel that is consistent with the electronic means of disseminating SQL Server 2000 reference information. In other words, the way that a given function reference appears on the pages of this book has been designed specifically to emulate the way that Books Online presents its reference pages.

The reason for maintaining such integration is simple: make it easy for you—the administrator or developer of SQL Server 2000 solutions—to use the tools and get the ongoing information you need to do your job. By providing a "common interface" among reference resources, your familiarity with the *SQL Server 2000 Reference Library* reference material can be immediately applied to Books Online, and vice-versa. In a word, it means *consistency*.

You'll find this philosophy of consistency and simplicity applied throughout Windows Programming Reference Series publications. I've designed the series to go hand-in-hand with online resources. Such consistency lets you leverage your familiarity with electronic reference material, and then apply that familiarity to enable you to get away from your computer if you'd like, take a book with you, and—in the absence of keyboards and e-mail and upright chairs—get your reading and research done. Of course, each of the *SQL Server 2000 Reference Library* volumes fits nicely right next to your mouse pad as well, even when opened to a particular reference page.

With any job, the simpler and more consistent your tools are, the more time you can spend doing work rather than figuring out how to use your tools. The structure and design of the *SQL Server 2000 Reference Library* provides you with a comprehensive, pre-sharpened toolset to quickly program, administer, or optimize SQL Server 2000 deployments.

T-SQL Language Reference Overview

Transact-SQL is central to the use of Microsoft SQL Server. All applications that communicate with SQL Server do so by sending Transact-SQL statements to the server, regardless of an application's user interface.

Transact-SQL is generated from many kinds of applications, including:

- General office productivity applications.
- Applications that use a graphical user interface (GUI) to allow users to select the tables and columns from which they want to see data.
- Applications that use general language sentences to determine what data a user wants to see.
- Line of business applications that store their data in SQL Server databases. These can include both applications from other vendors and applications written in-house.
- Transact-SQL scripts that are run using utilities such as **osql**.
- Applications created with development systems such as Microsoft Visual C++, Microsoft Visual Basic, or Microsoft Visual J++ that use database application programming interfaces (APIs) such as ADO, OLE DB, and ODBC.
- Web pages that extract data from SQL Server databases.
- Distributed database systems from which data from SQL Server is replicated to various databases or distributed queries are executed.
- Data warehouses in which data is extracted from online transaction processing (OLTP) systems and summarized for decision-support analysis.

Transact-SQL Syntax Conventions

The syntax diagrams in the Transact-SQL Reference use these conventions.

Convention	Used for
UPPERCASE	Transact-SQL keywords.
italic	User-supplied parameters of Transact-SQL syntax.
I (vertical bar)	Separating syntax items within brackets or braces. You can choose only one of the items.

(continued)

(continued)

Convention	Used for
[ ] (brackets)	Optional syntax items. Do not type the brackets.
{ } (braces)	Required syntax items. Do not type the braces.
[,...*n*]	Indicating that the preceding item can be repeated *n* number of times. The occurrences are separated by commas.
[...*n*]	Indicating that the preceding item can be repeated *n* number of times. The occurrences are separated by blanks.
bold	Database names, table names, column names, index names, stored procedures, utilities, data type names, and text that must be typed exactly as shown.
<label> ::=	The name for a block of syntax. This convention is used to group and label portions of lengthy syntax or a unit of syntax that can be used in more than one place within a statement. Each location in which the block of syntax can be used is indicated with the label enclosed in chevrons: <label>.

Unless specified otherwise, all Transact-SQL references to the name of a database object can be a four-part name in the form:

```
[
    server_name.[database_name].[owner_name].
    | database_name.[owner_name].
    | owner_name.
    ]
]
object_name
```

- *server_name* specifies a linked server name or remote server name.
- *database_name* specifies the name of a Microsoft SQL Server database when the object resides in a SQL Server database. It specifies an OLE DB catalog when the object is in a linked server.
- *owner_name* specifies the user that owns the object if the object is in a SQL Server database. It specifies an OLE DB schema name when the object is in a linked server.
- *object_name* refers to the name of the object.

When referencing a specific object, you do not always have to specify the server, database, and owner for SQL Server to identify the object. Intermediate nodes can be omitted; use periods to indicate these positions. The valid formats of object names are:

server.database.owner.object
server.database..object
server..owner.object
server...object
database.owner.object
database..object
owner.object
object

Code Example Conventions

Unless stated otherwise, the examples were tested using SQL Query Analyzer and its default settings for these options:

- QUOTED_IDENTIFIER
- ANSI_NULLS ·
- ANSI_WARNINGS
- ANSI_PADDING
- ANSI_NULL_DFLT_ON
- CONCAT_NULL_YIELDS_NULL

Most code examples in the Transact-SQL Reference have been tested on servers running a case-sensitive sort order. The test servers were usually running the ANSI/ISO 1252 code page.

Transact-SQL Data Type Categories

Data types with similar characteristics are classified into categories. Categories that contain two or three data types generally have a category name derived from the data types in that category. For example, the **money** and **smallmoney** category contains the **money** data type and the **smallmoney** data type. Data type names always appear in bold, even when used as part of a category name.

Transact-SQL Data Type Hierarchy

The following data type hierarchy shows the SQL Server data type categories, subcategories, and data types used in the SQL Server documentation. For example, the exact numeric category contains three subcategories: integers, **decimal**, and **money** and **smallmoney**.

The exact numeric category also contains all of the data types in these three subcategories: **bigint, int, smallint, tinyint, bit, decimal, money**, and **smallmoney**. Any reference to exact numeric in the Transact-SQL Reference refers to these eight data types.

In this hierarchy the category names built from two or more data types use the conjunction "and." The conjunction "or" may be used in the Transact-SQL Reference if it is more appropriate for the context in which the name is used.

The data types specified in this hierarchy also pertain to synonyms. For example, **int** refers to both **int** and its synonym **integer.** For more information, see Data Types.

numeric
    exact numeric
        integer
            bigint
            int
            smallint
            tinyint
        bit
        decimal and **numeric**
            decimal
            numeric
        money and **smallmoney**
            money
            smallmoney
    approximate numeric
        float
        real
    datetime and **smalldatetime**
        datetime
        smalldatetime

character and binary string
    character string
        char, varchar, and **text**
            char and **varchar**
                char
                varchar
            text
        Unicode character string
            nchar and **nvarchar**
                nchar
                nvarchar
            ntext
    binary strings
        binary and **varbinary**
            binary
            varbinary
        image

cursor

sql_variant

table

timestamp

uniqueidentifier

Additional data type categories used in the Transact-SQL Reference are described in these two hierarchies:

text, **ntext**, and **image**
      text and **ntext**
            text
            ntext
      image

short string
      short character
            char and **varchar**
                  char
                  varchar
            nchar and **nvarchar**
                  nchar
                  nvarchar
            binary and **varbinary**
                binary
                varbinary

New and Enhanced Features in Transact-SQL

Transact-SQL in Microsoft SQL Server 2000 provides new and enhanced statements, stored procedures, functions, data types, DBCC statements, and information schema views.

Data Types

New data types
bigint
table
sql_variant

Database Console Commands (DBCC)

New commands

DBCC CHECKCONSTRAINTS	DBCC DROPCLEANBUFFERS
DBCC CLEANTABLE	DBCC FREEPROCCACHE
DBCC CONCURRENCYVIOLATION	DBCC INDEXDEFRAG

Enhanced commands

DBCC CHECKALLOC	DBCC CHECKFILEGROUP
DBCC CHECKDB	DBCC SHOWCONTIG
DBCC CHECKTABLE	

Functions

New functions

BINARY_CHECKSUM	COUNT_BIG
CHECKSUM	DATABASEPROPERTYEX
CHECKSUM_AGG	fn_helpcollations
COLLATIONPROPERTY	fn_listextendedproperty

(continued)

(continued)

New functions

fn_servershareddrives	INDEXKEY_PROPERTY
fn_trace_geteventinfo	OBJECTPROPERTY
fn_trace_getfilterinfo	OPENDATASOURCE
fn_trace_getinfo	OPENXML
fn_trace_gettable	ROWCOUNT_BIG
fn_virtualfilestats	SCOPE_IDENTITY
GETUTCDATE	SERVERPROPERTY
HAS_DBACCESS	SESSIONPROPERTY
IDENT_CURRENT	SQL_VARIANT_PROPERTY

Information Schema Views

New information schema views
PARAMETERS
ROUTINES
ROUTINE_COLUMNS

Replication Stored Procedures

New replication stored procedures

sp_addmergealternatepublisher	sp_getqueuedrows
sp_addscriptexec	sp_getsubscriptiondtspackagename
sp_adjustpublisheridentityrange	sp_helparticledts
sp_attachsubscription	sp_helpmergealternatepublisher
sp_browsesnapshotfolder	sp_helpreplicationoption
sp_browsemergesnapshotfolder	sp_ivindexhasnullcols
sp_changesubscriptiondtsinfo	sp_marksubscriptionvalidation
sp_copysnapshot	sp_mergearticlecolumn
sp_disableagentoffload	sp_repladdcolumn
sp_dropanonymouseagent	sp_repldropcolumn
sp_dropmergealternatepublisher	sp_restoredbreplication
sp_enableagentoffload	sp_resyncmergesubscription
sp_getagentoffloadinfo	sp_vupgrade_replication