SQL Server 2005
数据库技术——从应用到原理

张蕊 著

中国水利水电出版社
www.waterpub.com.cn

SQL Server 2005
数据库技术——从应用到原理

张蕊 著

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书主要内容包括：数据库基础知识、数据库（关系模型）、基本表（关系模式、关系完整性、关系规范化理论）、视图与索引（数据库系统的三级模式与二级映像）、数据查询（关系操作）、高级数据库对象（数据库安全性、数据库完整性）、数据库安全与恢复技术等。内容以应用技术为主线，括号内分别为与之对应的数据库原理，加深理解，不显枯燥，达到理论和实践的紧密结合。

本书可作为数据库系统管理员、信息系统管理员、计算机专业开发人员、广大科技工作者和研究人员参考的工具书。内容从基础语法格式入门，逐步深入，还可供零基础的计算机专业爱好者自学使用。

图书在版编目（CIP）数据

SQL Server 2005数据库技术 : 从应用到原理 / 张蕊著. -- 北京 : 中国水利水电出版社, 2015.5
ISBN 978-7-5170-3103-1

Ⅰ. ①S… Ⅱ. ①张… Ⅲ. ①关系数据库系统 Ⅳ. ①TP311.138

中国版本图书馆CIP数据核字(2015)第081057号

书　　名	**SQL Server 2005 数据库技术——从应用到原理**
作　　者	张蕊　著
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路1号D座　100038） 网址：www.waterpub.com.cn E-mail：sales@waterpub.com.cn 电话：（010）68367658（发行部）
经　　售	北京科水图书销售中心（零售） 电话：（010）88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排　　版	中国水利水电出版社微机排版中心
印　　刷	三河市鑫金马印装有限公司
规　　格	170mm×240mm　16开本　9.25印张　176千字
版　　次	2015年5月第1版　2015年5月第1次印刷
印　　数	0001—1500册
定　　价	**26.00**元

前言 QIANYAN

数据库技术是信息系统的一个核心技术，是一种计算机辅助管理数据的方法，主要研究如何组织和存储数据，如何高效地获取和处理数据。从20世纪60年代末期开始到如今，人们在数据库技术的理论研究和系统开发上都取得了辉煌的成就，不断研究新一代数据库系统。数据库系统已经成为现代计算机系统的重要组成部分，不仅应用于日常事务处理，且逐步应用到情报检索、人工智能、专家系统、计算机辅助设计等各个领域。

本书从数据库系统日常维护人员和管理员角度出发，详细地介绍了数据库系统管理和日常维护的方法与技巧。以SQL Server 2005为平台，在详细深入介绍数据库应用技术的基础上，逐步引出与之相关联的数据库原理知识，内容循序渐进、深入浅出、层层递进，并结合图、表、提示等形式将应用与原理有机结合，使读者知其然，还知其所以然。避免只会操作，不知缘由；或只具有理论知识，不具备实践能力的现象。

作者具有丰富的数据库技术讲授与使用经验，同时参阅和借鉴了国内外许多优秀的数据库技术相关书籍。从DBA的角度出发，叙述上主要以T-SQL语言的形式介绍数据库及各种数据库对象的创建与维护方法，SQL Server Management Studio的形式不作为本书重点，内容适当取舍，突出重点。

本书可作为数据库系统管理员、信息系统管理员、计算机专业开发人员、广大科技工作者和研究人员参考的工具书。内容从基础

语法格式入门，逐步深入，还可供零基础的计算机专业爱好者自学使用。

本人才疏学浅，必有许多不完善或错误，恳请各位读者不惜赐教。

张蕊

2014 年 12 月于郑州

目录 MULU

第1章 数据库基础知识

1.1 数据库系统概述

数据库技术产生于20世纪60年代末，是数据管理的最新技术，是计算机科学的重要分支。从小型单项事务处理到大型信息系统，从一般企业管理到计算机辅助设计与制造（CAD/CAM）、办公信息系统（OIS）、地理信息系统（GIS）、全球定位系统（GPS）、北斗卫星导航系统（BDS）等，越来越多新的应用领域采用数据库存储和处理他们的信息资源。对于一个国家来说，数据库的建设规模、数据库信息量的大小和使用频率已成为衡量这个国家信息化程度的重要标志。因此，数据库课程是计算机科学与技术专业、信息管理专业的重要课程。

1.1.1 数据库技术的产生与发展

数据库系统的产生和发展与数据库技术的发展相辅相成。数据库技术就是数据管理技术，是对数据的分类、组织、编码、存储、检索和维护的技术。数据库技术的产生和发展与计算机技术及其应用的发展紧密联系在一起。主要经历了三个基本阶段。

1. 人工管理阶段

20世纪50年代中期以前，计算机主要用于科学计算。外存只有磁带、卡片、纸带，没有磁盘等直接存取的存储设备，而且计算机没有操作系统，没有管理数据的软件，数据处理方式是批处理。基本特点是：数据不保存、数据无专门软件进行管理、数据不共享（冗余度大）、数据不具有独立性（完全依赖于程序）、数据无结构。

2. 文件系统阶段

从20世纪50年代后期到20世纪60年代中期，计算机硬件和软件都有了一定的发展。计算机不仅用于科学计算，还大量用于管理。这时，硬件方面已经有了磁盘、磁鼓等直接存取的存储设备。在软件方面，操作系统中已经有了数据管理软件，一般称为文件系统。处理方式上不仅有了文件批处理，而且能够联机实时处理。基本特点是：数据可以长期保存、由文件系统管理数据、程

序与数据有一定的独立性、数据共享性差（冗余度大）、数据独立性差、记录内部有结构（但整体无结构）。

3. 数据库系统阶段

从20世纪60年代中期至今。随着计算机硬件和软件技术的飞速发展，计算机用于管理的规模更为庞大，应用越来越广泛，数据量急剧增长，数据的共享要求越来越高，数据库技术应运而生。

1.1.2 数据库常用术语

1. 数据（Data）

数据是描述现实世界事物的符号记录，是用物理符号记录的可以鉴别的信息。物理符号有多种表现形式，包括数字、文字、图形、图像、声音及其他特殊符号。数据的各种表现形式都可以通过数字化存入计算机。

2. 数据库（DataBase，DB）

数据库是长期存储在计算机内、有组织、可共享的数据集合。这种集合具有如下特点：

（1）最小的冗余度。以一定的数据模型来组织数据，数据尽可能不重复。

（2）应用程序对数据资源共享。为某个特定的组织或企业提供多种应用服务。

（3）数据独立性高。数据结构较强地独立于使用它的应用程序。

（4）统一管理和控制。对数据的定义、操纵和控制，由数据库管理系统统一进行。

3. 数据库管理系统（DataBase Management System，DBMS）

数据库管理系统是位于用户与操作系统之间的一个数据管理软件，它的基本功能包括以下几个方面：

（1）数据定义功能：DBMS提供数据定义语言（Data Definition Language，DDL），通过它可以方便地对数据库中的数据对象进行定义，如CREATE DATABASE是创建数据库命令，CREATE TABLE是创建数据表命令等。

（2）数据操纵功能：DBMS还提供数据操纵语言（Data Manipulation Language，DML），可以使用DML操纵数据，实现对数据的基本操作，例如查询、插入、删除和修改。

（3）数据库的运行管理功能：数据库在建立、运行和维护时，由数据库管理系统统一管理和控制，以保证数据的安全性、完整性，以及对并发操作的控制以及发生故障后的系统恢复等。

（4）数据库的建立和维护功能：包括数据库初始数据的输入、转换功能，数据库的转存、恢复功能，以及数据库的重组织功能和性能监视、分析功

能等。

4. 数据库系统（DataBase System，DBS）

数据库系统是指在计算机系统中引入数据库后构成的系统。一般由数据库、操作系统、数据库管理系统（及其开发工具）、应用系统、数据库管理员和用户构成。应该指出的是，数据库的建立、使用和维护等工作只有 DBMS 远远不够，还要有专门的数据库管理员（DataBase Administrator，DBA）来完成。

1.1.3 数据库技术的发展趋势

随着信息技术和市场的发展，人们发现关系数据系统虽然技术很成熟，但其局限性也是显而易见的：它能很好地处理所谓的“表格型数据”，却对越来越多复杂类型的数据无能为力。在相当一段时间内，人们把大量的精力花在研究“面向对象的数据库系统”，然而产品的市场发展情况并不理想。理论上的完美性并没有带来市场的热烈反映。主要原因在于，其主要设计思想是企图用新型数据库系统来取代现有的数据库系统。这对许多已经运用数据库系统并积累了大量工作数据的客户，尤其是大客户来说，无法承受新旧数据间的转换而带来的巨大工作量及巨额开支。另外，面向对象的关系型数据库系统使查询语言变得极为复杂，从而使得无论是数据库的开发商家还是应用客户都视其复杂的应用技术为畏途。

数据库现在的发展方向，是新近出现的将原有的关系数据库与许多其他的功能，如电子邮件、个人通信等等相结合的趋势。而在企业自动化、电子政务等应用领域，人们相互进行的协同工作，也在与数据库技术融合。现在的数据库技术已经发展到了一个全新的阶段，即数据采集的多样化。这一变化给数据库技术带来很多的挑战，纵观数据库发展，整个数据库发展呈现出了以下几个特征。

1. 传感器数据库技术

随着微电子技术的发展，传感器的应用越来越广泛。可以使小鸟携带传感器，根据传感器在一定的范围内发回的数据定位小鸟的位置，从而进行其他的研究；还可以在汽车等运输工具中安装传感器，从而掌握其位置信息；甚至在微型的无人间谍飞机上也开始携带传感器，在一定的范围内收集有用的信息，并且将其发回到指挥中心。

当有多个传感器在一定的范围内工作时，就组成了传感器网络。传感器网络由携带者所捆绑的传感器及接收和处理传感器发回数据的服务器所组成。传感器网络中的通信方式可以是无线通信，也可以是有线通信。

传感器数据库必须利用系统中的所有传感器，而且可以像传统数据库那样方便、简洁地管理传感器数据库中的数据；建立可以获得和分配源数据的机

制；建立可以根据传感器网络调整数据流的机制；可以方便地配置、安装和重新启动传感器数据库中的各个组件等。传感器网络越来越多地应用于对很多新应用的监测和监控。在这些新的应用中，用户可以查询已经存储的数据或者传感器数据，但是，这些应用大部分建立在集中的系统上收集传感器数据。在这样的系统中数据是以预定义的方式抽取的，因此缺乏一定的灵活性。

2. 微小型数据库技术

数据库技术一直随着计算的发展而不断进步，随着移动计算时代的到来，嵌入式操作系统对微小型数据库系统的需求为数据库技术开辟了新的发展空间。微小型数据库技术目前已经从研究领域逐步走向应用领域。随着智能移动终端的普及，人们对移动数据实时处理和管理要求也不断提高，嵌入式移动数据库越来越体现出其优越性，从而被学界和业界所重视。

3. 信息集成

信息系统集成技术已经历了20多年的发展过程，研究者已提出了很多信息集成的体系结构和实现方案，然而这些方法所研究的主要集成对象是传统的异构数据库系统。随着Internet的飞速发展，网络迅速成为一种重要的信息传播和交换的手段，尤其是在Web上，有着极其丰富的数据来源。如何获取Web上的有用数据并加以综合利用，即构建Web信息集成系统，成为一个引起广泛关注的研究领域。

4. 数据流管理

测量和监控复杂的动态的现象，如远程通信、Web应用、金融事务、大气情况等，产生了大量、不间断的数据流。数据流处理对数据库、系统、算法、网络和其他计算机科学领域的技术挑战已经开始显露。这是数据库界一个活跃的研究领域，包括新的流操作、SQL扩展、查询优化方法、操作调度(Operator Scheduling)技术等。

5. “大数据”正步入实质性阶段

随着互联网业务的迅猛发展，数据规模急剧的膨胀，与之对应的IT硬件更新速度完全无法与之相比，存储和管理海量数据已越来越成为亟待解决的问题，大数据的概念也是由此应运而生，在这方面，NoSQL所具有的高性能、高可用性、高扩展能力非常适合TB、PB甚至ZB级数据的需求，也是目前“大数据”应用的主力。

1.2 关系数据库标准语言SQL

SQL (Structured Query Language)，即结构化查询语言，是关系数据库的标准语言，SQL是一个通用的、功能极强的关系数据库语言。其功能并不

仅仅局限在查询上。当前，几乎所有的关系数据库管理系统软件都支持 SQL。当然，许多软件厂商对 SQL 基本命令也进行了不同程度的扩充和修改，但是，大多数数据库均使用 SQL 作为共同的数据存取语言和标准接口已成为不争的事实，SQL 已成为数据库领域中的主流语言。

1.2.1 SQL 的产生和发展

SQL 语言是在 1974 年由 Boyce 和 Chamberlin 联合提出的。1975—1979 年 IBM 公司 San Jose Research Laboratory 研制了著名的关系数据库管理系统原型 System R，并实现了这种语言。1986 年 10 月美国国家标准局（American National Standard Institute，ANSI）的数据委员会 X3H2 批准了 SQL 作为关系数据库语言的美国标准，同时公布了 SQL 标准文本（简称 SQL－1986）。1987 年国际标准化组织（International Organization for Standardization，ISO）也通过了这一标准。此后 ANSI 不断修改和完善 SQL 标准，并与 1989 年公布了 SQL－1989 标准，1992 年又公布了 SQL－1992 标准。1999 年公布了 ANSI SQL－1999，也称作 SQL3，2003 年公布了 SQL2003。随着版本的不断增加，从最初的单文档到 SQL2003 的 3600 多页，SQL 标准的内容越来越多，规则越来越细化。

SQL 标准的影响超出了数据库领域。SQL 成为国际标准后，它在数据库以外的其他领域中也得到了重视和采用。有不少软件产品将 SQL 语言的数据查询功能与图形工具、软件工程工具、软件开发工具、人工智能程序结合起来。

1.2.2 SQL 的特点

SQL 之所以能够成为用户和业界的国际标准，并被广大用户所接受，是因为它是一个综合的、功能极强同时又简单易学的语言。SQL 集数据查询（Data Query）、数据操纵（Data Manipulation）、数据定义（Data Definition）和数据控制（Data Control）功能于一体，主要特点如下。

1. 综合统一

SQL 集数据定义语言 DDL、数据操纵语言 DML、数据控制语言 DCL 的功能于一体，语言风格统一，可以独立完成数据库生命周期中的全部活动。

（1）定义数据模式、插入数据，建立数据库。

（2）对数据库中的数据进行查询和更新。

（3）数据库重构和维护。

（4）数据库安全性和完整性控制等一系列操作要求。

这就为数据库应用系统的开发提供了良好的环境。特别是用户在数据库系统投入运行后，还可根据需要随时、逐步地修改模式，并不影响数据库的运行，从而使系统具有良好的可扩展性。

2. 以一种语法提供两种使用方式

SQL 具有自主式语言和嵌入式语言两种使用方式。自主式 SQL 能够独立地进行联机交互，用户只需在终端键盘上直接键入 SQL 命令即可对数据库进行操作。嵌入式 SQL 能够嵌入到高级语言（常用的主语言有 C、Visual Basic、PowerBuilder、Dehpi 等）的程序中，以实现对数据库的数据进行存取操作，给程序员设计程序提供了很大方便。在两种不同的使用方式中，SQL 的语法结构基本一致，使用方法大致相同。统一的语法结构的特点，为使用 SQL 提供了极大地灵活性和方便性。

3. 高度非过程化

非关系数据模型的数据操纵语言是“面向过程”的语言，用“过程化”语言定义完成某项请求，必须指定存取路径。而用 SQL 进行数据操作，只要提出“做什么”，而无须指明“怎么做”，因此无须了解存取路径。存取路径的选择以及 SQL 的操作过程由系统自动完成。这不但大大减轻了用户负担，而且有利于提高数据独立性。

4. 语言简洁、易学易用

尽管 SQL 语言功能极强又有两种使用方式，由于设计巧妙，其语言十分简洁，四大功能的完成仅用了 9 个动词：CREATE、DROP、ALTER、SELECT、INSERT、UPDATE、DELETE、GRANT 和 REVOKE。此外，SQL 语法接近英语口语，因此容易学习，容易使用。

1.3　Transact-SQL 语言

Transact-SQL 是在 MS SQL Server 中使用的 SQL，简称 T-SQL。它是微软在标准 SQL 语言基础上创建的符合 SQL Server 特点的数据库访问语言，一直以来都是 SQL Server 的开发、管理工具。SQL Server 2005 版本提供了很多增强功能，包括错误处理，递归查询，在 SQL Server 2005 中的很多操作都是使用 T-SQL 语言来实现的。大部分的可视化操作都可以由 T-SQL 完成，而且很多的高级管理必须由它完成。

T-SQL 虽然具备许多与程序设计语言类似的功能，然而 T-SQL 并非是编程语言。T-SQL 主要是为操作关系数据库而设计的，同时也包含许多可用的其他结构化语言所具有的逻辑运算、数学计算、条件表达式、流程控制结构等。

Transact-SQL 是应用于数据库的语言，本身不能独立存在。它是一种非过程性语言，与一般的高级语言（如 C、C++）是不同的。一般的高级语言在使用数据库时，需要依照每一行程序的顺序处理很多的操作。但是 SQL 语

言，用户只需告诉数据库需要什么数据，怎么显示，具体的内部操作有数据库系统来完成。

Transact-SQL 语言简单明了，易学易用，很容易掌握。按照用途包括以下内容：

（1）数据定义语言（DDL）：每个数据库、数据库中的表、视图、索引和完整性约束等都是数据库对象，要建立这些对象，即可通过 SQL 语言来完成。

（2）数据操纵语言（DML）：对已经创建了的数据库对象中的数据进行添加、修改和删除的语句，有 INSERT（插入）、DELETE（删除）和 UPDATE（更新）等操作。

（3）数据查询语言：数据库的查询过程通过 SQL 语言来实现，例如 SELECT 命令。

（4）数据控制语言（DCL）：用于设置或者更改数据库用户的权限。

1.3.1 T-SQL 语法

在书写 T-SQL 语言时要遵循一定的约定，表 1.1 给出了 T-SQL 语言使用时的规则。

表 1.1　　T-SQL 语法约定

规　则	说　明
UNION（大写）	T-SQL 关键字
|（竖线）	分隔括号和大括号的语法
[ ]（方括号）	可选语法项
{ }（大括号）	必选语法项
[，…n]	指示前面的项可以重复 n 次，每一项用逗号分隔
[…n]	指示前面的项可以重复 n 次，每一项用空格分隔
[;]	可选的 T-SQL 语句终止符
〈label〉::=	语法块的名称

1. 数据类型

SQL Server 提供两大类数据类型：系统数据类型、用户自定义数据类型。

系统数据类型在第 3 章详细介绍。

用户自定义数据类型是在已有的系统数据类型基础上扩充或限定，并非定义一个新的存储结构类型。当多个表中的列需要存储相同的数据类型时，并且想确保这些列具有完全相同的类型、长度和是否为空，这时用户就可以定义数据类型，并在创建表的这些列时使用这些数据类型。用户自定义数据类型的创建方法有两种：

（1）SQL Server Management Studio。展开某一数据库节点→可编程性→

右击“类型”→新建→用户定义数据类型→打开“新建用户定义数据类型”对话框→设置各个参数。

(2) 使用 T-SQL 语言。使用系统存储过程 sp_addtype 创建，语法格式为：

```
sp_addtype [@typename =] type,
[@phystype =] system_data_type
[, [@nulltype=] 'null_type']
```

参数说明：

① [@typename =] type：用户自定义数据类型的名称，数据类型的名称要遵照标识符的规则，且在数据库中是唯一的。

② [@phystype =] system_data_type：用户自定义数据类型所基于的系统数据类型，例如 int 或者 bit 型。

③null_type：该参数指明用户定义的数据类型处理空值的方式。该参数有三个取值：null、not null 或者 nonull，其默认值为 null。

2. 变量

T-SQL 变量命名规则如下：

1) 以 ASCII 字母、Unicode 字母、下划线、@或者 # 开头，后续可以为一个或多个 ASCII 字母、Unicode 字母、下划线、@、# 或者 $，但整个标识符不能全部是下划线、@或者 #。

2) 标识符不能是 T-SQL 的关键字。

3) 标识符中不能嵌入空格，或者其他的特殊字符。

4) 如果要在标识符中使用空格或者 T-SQL 的关键字以及特殊字符，则要使用双引号或者方括号将该标识符括起来。

SQL Server 2005 中，变量分为两种：局部变量和全局变量。

(1) 局部变量。局部变量是由用户声明的，声明的同时可以指定变量的名称（以@开头）、数据类型和长度，并同时设该变量的值为 NULL。局部变量仅在声明它的批处理、存储过程或者触发器中有效，当批处理、存储过程或者触发器执行结束后，局部变量将变成无效。

局部变量用 DECLARE 定义，语法格式如下：

```
DECLARE {@local_variable data_type } [, …n]
```

各参数说明如下：

- @local_variable：是变量的名称。变量名必须以@符号开头，符合变量命名规则。
- data_type：由系统提供的数据类型或者用户自定义的数据类型。

给局部变量的赋值可以使用 SET 语句直接赋值，也可以使用 SELECT 在查询中给变量赋值。其语法格式分别如下：

```
SET @local_variable = expr
SELECT {@local_variable = expr} [, …n]
```

各参数说明如下：

- @local_variable：是变量的名称。
- expr：相应类型的表达式。

【例 1.1】 通过 DECLARE 声明一个局部变量。

```
DECLARE @cno varchar (9);
```

★提示： 可同时声明多个变量，中间用“,”隔开。

【例 1.2】 通过 DECLARE 声明两个局部变量，并用 SET 给变量赋值。

```
DECLARE @var1 nvarchar (20), @var2 nchar (10)
SET @var1='学号'
SET @var2='年龄'
```

【例 1.3】 用 SELECT 语句给变量赋值。

```
DECLARE @cj float
SELECT @cj=90
SELECT * FROM sc WHERE cgrade<=@chengji
```

（2）全局变量。全局变量是 SQL Server 系统内部使用的变量，其作用范围并不局限于某个程序，而是任何程序任何时间都可以调用。全局变量通常用于存储一些 SQL Server 的配置设定值和效能统计数据。可以利用全局变量来测试系统的设定值或者 T-SQL 的命令执行后的状态值。

常用的全局变量如表 1.2 所示。

表 1.2　　SQL Server 2005 中的全局变量

名　称	变　量　说　明
@@CONNECTIONS	返回自最近一次启动 SQL Server 以来连接或试图连接的次数
@@DATEFIRST	返回 SET DATEFIRST 参数的当前值，SET DATEFIRST 参数用于指定每周的第一天是周几。例如 1 对应周一，7 对应周日
@@CUP_BUSY	返回自 SQL Server 最近一次启动以来 CPU 的工作时间其单位为毫秒
@@CURSOR_ROWS	返回最后连接上并打开的游标中当前存在的合格行的数量
@@SERVERNAME	返回运行 SQL Server 2000 本地服务器的名称

续表

名　称	变　量　说　明
@@REMSERVER	返回登录记录中记载的远程 SQL Server 服务器的名称
@@ERROR	返回最后执行的 Transact-SQL 语句的错误代码
@@ROWCOUNT	返回受上一语句影响的行数，任何不返回行的语句将这一变量设置为 0
@@VERSION	返回 SQL Server 当前安装的日期、版本和处理器类型
@@DBTS	返回当前数据库的时间戳值必须保证数据库中时间戳的值是唯一的
@@FETCH_STATUS	返回上一次 FETCH 语句的状态值
@@IDENTITY	返回最后插入行的标识列的列值
@@IDLE	返回自 SQL Server 最近一次启动以来 CPU 处于空闲状态的时间长短，单位为毫秒
@@IO_BUSY	返回自 SQL Server 最后一次启动以来 CPU 执行输入输出操作所花费的时间，单位为毫秒
@@LANGID	返回当前所使用的语言 ID 值
@@LANGUAGE	返回当前使用的语言名称
@@LOCK_TIMEOUT	返回当前会话等待锁的时间长短，单位为毫秒
@@MAX_CONNECTIONS	返回允许连接到 SQL Server 的最大连接数目
@@MAX_PRECISION	返回 decimal 和 numeric 数据类型的精确度
@@NESTLEVEL	返回当前执行的存储过程的嵌套级数，初始值为 0
@@OPTIONS	返回当前 SET 选项的信息
@@PACK_RECEIVED	返回 SQL Server 通过网络读取的输入包的数目
@@PACK_SENT	返回 SQL Server 写给网络的输出包的数目
@@PACKET_ERRORS	返回网络包的错误数目
@@PROCID	返回当前存储过程的 ID 值
@@SERVICENAME	返回 SQL Server 正运行于哪种服务状态之下：如 MS SQL Server、MSDTC、SQL Server Agent
@@SPID	返回当前用户处理的服务器处理 ID 值
@@TEXTSIZE	返回 SET 语句的 TEXTSIZE 选项值 SET 语句定义了 SELECT 语句中 text 或 image。数据类型的最大长度基本单位为字节
@@TIMETICKS	返回每一时钟的微秒数
@@TOTAL_ERRORS	返回磁盘读写错误数目
@@TOTAL_READ	返回磁盘读操作的数目
@@TOTAL_WRITE	返回磁盘写操作的数目
@@TRANCOUNT	返回当前连接中处于激活状态的事务数目