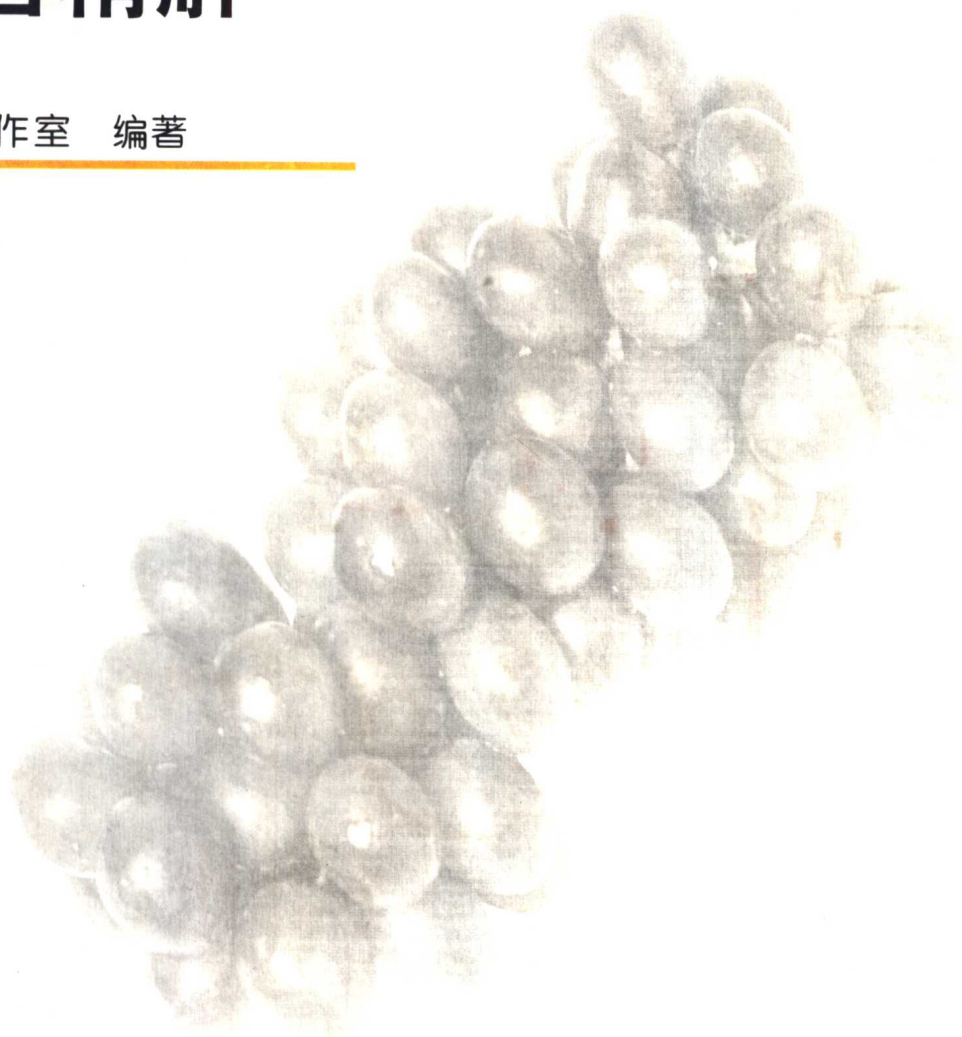


Transact -SQL

语言精解

中创工作室 编著



中国电力出版社

www.cepp.com.cn

Transact-SQL

语言精解

中创工作室 编著



中国电力出版社
www.cepp.com.cn

TP3

内 容 提 要

本书是《SQL Server 7.0 系统管理入门》一书的姐妹篇。Transact-SQL 是微软公司的数据库产品 SQL Server 7.0 中对标准 SQL 的扩展版本,是所有开发工具和应用程序访问数据的核心。

本书以深入浅出的方式,围绕一个样本数据库进行讲述,每章还精心设计了一个练习。读者定能在这种理论与实践紧密结合的学习环境中轻松掌握 Transact-SQL。

图书在版编目(CIP)数据

Transact-SQL 语言精解/中创工作室 编著.-北京:中国电力出版社,2000.1

ISBN 7-5083-0229-X

I .T… II .中… III.数据库管理系统, Transact-SQL IV. TP311.13

中国版本图书馆 CIP 数据核字(1999)第 75035 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.cepp.com.cn>)

北京密云红光印刷厂印刷

各地新华书店经售

*

2000 年 1 月第一版 2000 年 1 月北京第一次印刷

787 毫米×1092 毫米 16 开本 19 印张 432 千字

定价 29.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题,我社发行部负责退换)

目 录

第一章 Transact-SQL概述	1
1.1 数据库对象	1
1.2 Transact-SQL语言的内容	2
1.3 Transact-SQL的用途	4
第二章 本书中的样本数据库	5
2.1 数据库中的实体	5
2.2 数据库中的其他对象及练习	7
2.3 附录	10
第三章 有关表的基本操作	16
3.1 简单表的创建和修改	16
3.2 表中数据的更新	25
3.3 表中数据的查询	30
练习一 建表并对表进行简单的插入、修改、删除和查询操作	71
第四章 有关表的复杂特性	75
4.1 定义约束	75
4.2 自定义类型	83
4.3 缺省	84
4.4 规则	88
4.5 索引	92
4.6 触发器	97
练习二 表的复杂特征：自定义数据类型、缺省、规则、索引和触发器	101
第五章 批处理与流控制语言	107
5.1 批处理概述	107
5.2 流控制语言	109
练习三 学习使用批处理与流控制语言	125
第六章 事务	128
6.1 事务概述	129
6.2 嵌套使用事务	134
6.3 锁的使用	136
练习四 学习如何使用事务	137
第七章 存储过程	139
7.1 存储过程的创建	139

7.2 有关存储过程的其他操作	144
练习五 学习创建存储过程	145
第八章 安全管理	148
8.1 授予对象权限	148
8.2 拒绝对象权限	149
8.3 收回对象权限	150
8.4 综合使用各种命令	152
第九章 表中数据的复杂操作	154
9.1 复杂查询	154
9.2 表中数据的复杂更新	188
练习六 连接查询、子查询、带子句的修改、删除语句及大文本数据的插入 ...	195
第十章 使用索引提高查询效率	199
10.1 创建索引	199
10.2 使用索引	207
10.3 更新和维护索引	209
练习七 合理地创建与使用索引	210
第十一章 数据操纵中的视图使用技术	213
11.1 视图的创建	213
11.2 视图的使用	218
11.3 视图的修改和删除	223
练习八 体会通过视图带来的安全性和简单性	224
第十二章 密集型记录的游标访问技术	226
12.1 游标的创建和使用	227
12.2 游标使用中应注意的问题	233
练习九 通过游标访问表中数据	234
第十三章 应用逻辑的存储过程实现技术	236
13.1 复杂的存储过程的创建	236
13.2 存储过程的应用	240
13.3 系统存储过程	242
练习十 学习、使用分组存储过程和系统存储过程	244
第十四章 触发器在维护数据完整性的应用	246
14.1 触发器的创建、修改和使用	246
14.2 触发器的嵌套与递归使用	253
14.3 触发器的限制	255

练习十一 触发器的特性：嵌套与递归触发	256
第十五章 自然语言查询	260
15.1 安装Microsoft English Query.....	261
15.2 创建基于English Query的应用	265
练习十二 创建基于English Query应用查询数据库中的数据.....	290



第一章 Transact-SQL 概述

SQL (Structured Query Language) 是关系数据库管理系统 (RDBMS) 的操作语言, 用来设置、使用和维护关系数据库, 使定义和管理关系数据库的操作规范化。

最早提出 SQL 的是 E. F. Codd 博士, 他开发的关系数据库模型证明关系代数的逻辑操作能够从关系数据库中提取出任何数据, 并且完成这一任务不需要任何过程代码。1986 年 10 月, 美国国家标准局 (ANSI) 推出了第一个 SQL 语言的标准 SQL-86, 随后该标准被 SQL-89 代替。当前的执行标准是 SQL-92。目前大多数数据库厂商积极地把 SQL-92 的标准结合到了他们的数据库中。

Microsoft SQL Server 7.0 中的 Transact-SQL 是 SQL Server 对标准 SQL 的扩展版本, 它不仅与 ANSI SQL 标准兼容, 而且还提供了大量的命令和函数, 使得对数据库的操作变得更为灵活。

下面首先介绍 Transact-SQL 操作的数据库对象, 然后介绍 Transact-SQL 语言的内容, 最后介绍 Transact-SQL 在整个 SQL Server 数据库中的作用。

1.1 数据库对象

在关系数据库系统 (RDBMS) 中, 数据都存储在表中。而数据库中的其他对象, 诸如: 用户自定义数据类型、缺省、规则、视图、索引、触发器和存储过程等, 都是为表服务的。下面我们将简单地介绍一下这些数据库对象。

1. 表 (Table)

在关系数据库中, 数据的逻辑结构是二维表: 即由表中的行和列来存储数据。其中, 表中的每行描述实体的一个实例; 一列描述实体的一个属性。例如, 图 1-1 是表 ZGXXB 的实例, 此表用于存储职工信息, 包括职工代码、录入序号、姓名和性别等。

在 Microsoft SQL Server 7.0 中, 还提供了大量的系统表, 用于存储系统信息。

2. 用户自定义数据类型 (User Defined Data Types)

数据类型用于约束表的列, 规定列能够存放什么样的信息以及数据是如何存储的。用户自定义数据类型是用户根据具体需要, 在系统数据类型的基础上派生而来的。

3. 缺省 (Default)

缺省是绑定到用户自定义数据类型上或直接绑定到列上的一种数据库对象, 用于为列指定一个缺省值。当在表中插入数据时, 如果没有为该列指定插入值, 则在该列中插入缺省值。

表 ZGXXB

ZGXX DM	ZGXX XH	ZGXX XM	ZGXX XB	...
A00001	1	刘晨	0	...
A00002	2	刘建元	1	...
A00003	3	余莲	1	...
A00004	4	麦中凡	1	...

表的一行

表的一列

图 1-1 表的实例

4. 规则 (Rule)

规则是一种绑定到用户自定义数据类型上或直接绑定到列上的一种数据库对象，主要用于指定表列的取值必须遵守的规范。也就是说，当在表中插入数据时，插入的数据必须遵循规则指定的条件。

5. 索引 (Index)

索引是一种建立在表上的数据对象，用于帮助 SQL Server 定位要查找的数据，加快查询速度。

6. 视图 (View)

视图是一种逻辑意义上的表，它由其他的表或视图的特定列和行（或它们的变换）组成。可以用视图改变用户对数据库观察的角度和透明度。

7. 存储过程 (Stored Procedure)

存储过程是在建立时就已经编译和优化的一段程序，它存储在数据库中，执行速度要比独立地执行这段程序快得多。在数据库中，有系统提供的存储过程（称为系统存储过程），也有用户自己创建的存储过程。系统存储过程是随着 SQL Server 一起在安装时自动生成的，主要用于系统管理。

8. 触发器 (Trigger)

触发器是创建在表上的一种数据库对象，是一种特殊的存储过程。当用户修改、删除或插入数据时，触发器被自动地触发。触发器的作用是强制维护数据库中数据的完整性。

1.2 Transact-SQL 语言的内容

SQL (Structured Query Language) 的英语名称为结构化查询语言，在标准 SQL 中，它的实际功能还包括：数据操纵、数据定义和数据控制三方面。在 Microsoft SQL Server 7.0

中, Transact-SQL 不仅包含标准 SQL 的三类功能, 而且还包含支持结构化编程的语言: 流控制语言。

Transact-SQL 的内容如下:

✧ 数据定义语言 (Data Definition Language): 主要用于定义、修改和删除数据库中的对象 (如: 表、缺省、规则、索引、视图、触发器和存储过程)

✧ 数据操纵语言 (Data Manipulation Language): 主要用于对表中的数据进行操作, 包括插入数据 (Insert)、修改数据 (Update)、删除数据 (Delete) 和查询数据 (Select)

✧ 数据控制语言 (Data Control Language): 主要用于限制用户的访问权限, 包括 Grant (授予权限) 和 Revoke (收回权限) 两种语句。

✧ 流控制语言 (Control-of-Flow Language): 主要用于结构化编程中, 控制 SQL 语句的执行流向。

在本书中, 我们主要介绍在 Microsoft SQL Server 7.0 中, 如何使用 Transact-SQL 语言对数据库中的数据进行操作。与前一个版本 (Microsoft SQL Server 6.5) 相比, Microsoft Transact-SQL 还提供了很多新的扩展功能, 现简述如下:

1. 新的服务器选项和数据库选项

增加的服务器选项有: cost threshold for parallelism、query wait(s)、index create memory 和 remote proc trans 等。

增加的数据库选项有: autoclose、cursor close on commit、autoshrink merge publish、ANSI nulls 等。

2. 新的数据类型

新增加的数据类型有: nchar、nvarchar、ntext 和 cursor。

3. 新的 Transact-SQL 函数

新增加的函数有: columnproperty()、containstable()、cursor_status() 和 databaseproperty() 等。

4. 新的 Transact-SQL 语句

新增加的 SQL 语句有: alter procedure、alter view、alter trigger 和 create statistics 等。

5. 新的系统存储过程

新增加的系统存储过程有: sp_add_agent_parameter、sp_add_alert、sp_addapprole、sp_adddistpublisher 和 sp_adddistributiondb 等。

6. 新的系统表

新增加的系统表有: backupfile、backupmediafamily、backupmediaset、backupset、MSagent_parameters 和 MSagent_profiles 等。

此外, 在 Microsoft SQL Server 7.0 中, 还提供了一种称为英语查询的环境, 在此环境下, 用户可以使用自然英语查询语句来查询数据库中的信息。

1.3 Transact-SQL 的用途

对于所有的开发工具和应用程序来说，无论前台开发环境是什么，它们与数据库的交互语言只有一种：Transact-SQL 语言。Microsoft SQL Server 7.0 提供了大量的图像和命令行实用程序，使得客户能够以各种方式访问数据，其核心仍然是 Transact-SQL 语言，如图 1-2 所示。

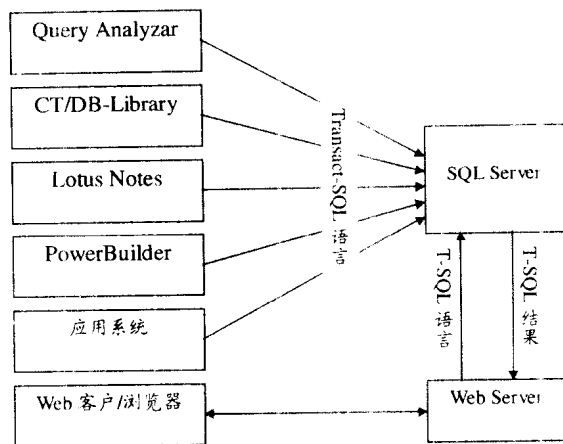


图 1-2 Transact-SQL 语言的核心地位

由图 1-2 可知，要想灵活地操作数据库中的数据，必须掌握 Transact-SQL 语言；Transact-SQL 在整个 SQL Server 中的关键性作用，也要求必须学好 Transact-SQL 语言。

第二章 本书中的样本数据库

在本书中，我们将以样本数据库 ms_database 为例，介绍在 Microsoft SQL Server 7.0 中如何使用 Transact-SQL 语言操纵数据。有关样本数据库 ms_database 的创建，请参看本书姐妹篇《SQL Server 7.0 系统管理入门》（中国电力出版社出版）的相关章节。

本章将介绍样本数据库的物理设计及要做的练习。

2.1 数据库中的实体

在样本数据库中，我们将以一个职工考核系统为例，介绍如何使用 Transact-SQL 语言。下面介绍该系统中的实体及实体间的关系，然后介绍表示这些实体和实体关系的表，并给出建表语句。

1. 实体 (Entities) 及实体间的关系 (Relations)

在职工考核系统中，共包含五个实体，如下所示：

1) 职工实体：用于记录职工的情况，其属性包括职工代码、名字、录入序号、录入日期、出身日期、地址、邮政编码、工资、性别、照片和个人简介。

2) 工作评定实体：用于记录工作考核的主要信息，其属性包括评定编号、标题、评定日期和评定内容。

3) 工作评定分类实体：用于记录评定与考核分类信息，其属性包括分类代码及分类备注。

4) 评级工作岗位实体：用于记录评级所在岗位，其属性为岗位。

5) 被评级别实体：用于记录被评定级别，其属性为被评级别。

这些实体之间的主要关系如下：

✧ 职工实体和工作评定实体：一个职工可进行多次评定，一种评定必须针对且只针对一个职工。

✧ 工作评定实体与评级工作岗位实体：多次评定可针对一个相同的岗位，一次评定可针对多个岗位。

✧ 工作评定实体与工作评定分类实体：多次评定可有一个共同的评定分类，一次评定可有多个评定分类。

✧ 工作评定实体与被评级别实体：一次评定有且必须有一个被评级别，一个被评级别可以对应多次评定。

图 2-1 显示了这些实体及实体之间的关系。

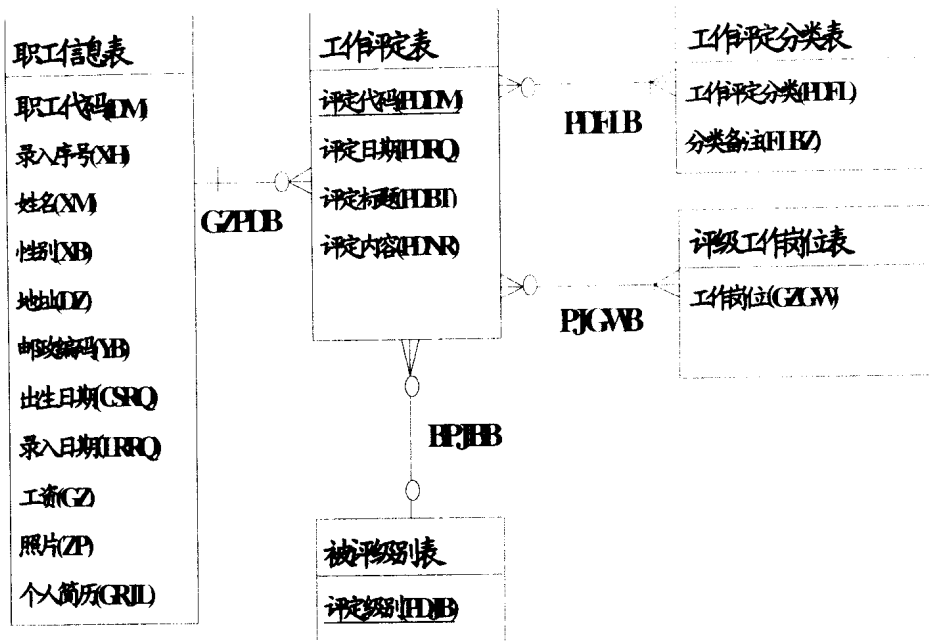


图 2-1 实体关系图

2. 表 (Table)

由于在工作评定实体与评级工作岗位实体、工作评定实体与工作评定分类实体之间存在多对多关系，所以，在样本数据库中，除了包含五个表示实体的表之外，还要包含两个关系表：评级岗位表和评定分类表。在样本数据库中，共包括如下的七个表：

- 1) 职工信息表：ZGXIB
- 2) 工作评定表：GZPDB
- 3) 工作评定分类表：PDFLB
- 4) 评级工作岗位表：PJGWB
- 5) 被评级别表：BPJBB
- 6) 评级岗位表：GZPDB_PJGWB
- 7) 评定分类表：GZPDB_PDFLB

图 2-2 显示了数据库中这些表的名称、表中的列名、列的数据类型和表之间的关系。

3. 建表语句

在样本数据库 ma_database 中使用批命令文件可以生成职工考核系统中的所有表，而且还可以在各个表中插入数据。在本书中，我们将以这些表及表中的数据为例，介绍 Transact-SQL 的使用方法。生成此数据库的 SQL 文件参见本章附录。

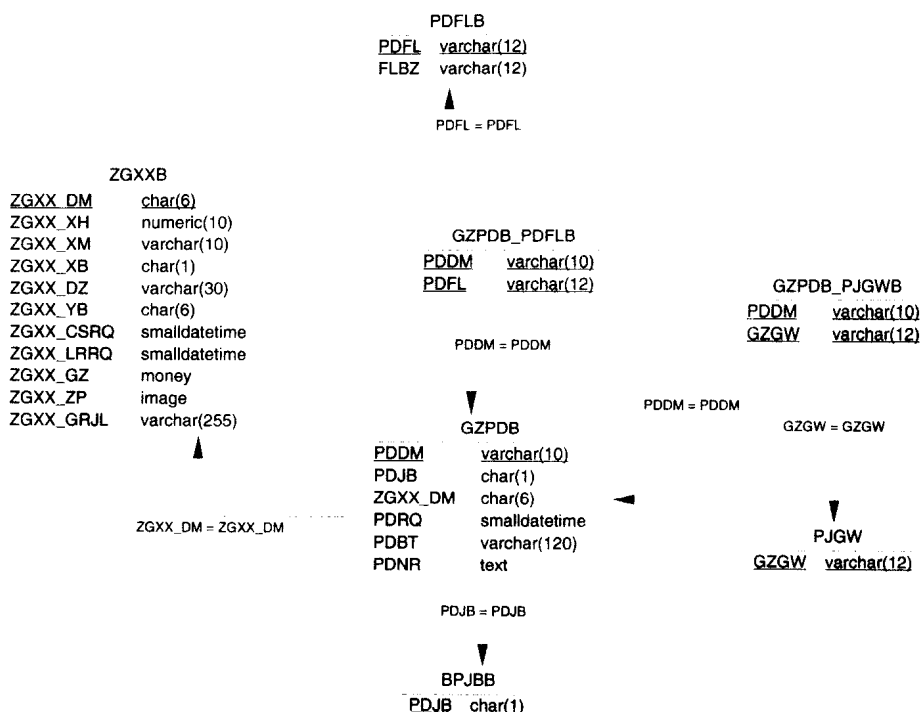


图 2-2 物理模型

2.2 数据库中的其他对象及练习

通过学习前面的内容，我们在样本数据库 `ms_database` 中创建了职工考核系统中的所有表，并在表中插入了数据。另外，我们还需要在样本数据库中创建为表服务的数据库对象，以维护数据的完整性和一致性。这些数据库对象将在本书各章的练习中实现。下面介绍这些要创建的数据库对象，以及本书中的练习。

1. 数据库中的对象

在样本数据库中，除了包含表以外，还要包含如下的数据库对象来维护表中的数据。

1) 用户自定义数据类型

在上面创建的表中，由于在两个表（`ZGXXB` 表和 `GZPDB` 表）中都有相同的列 `ZGXX_DM` 所以，为了统一维护和修改，我们在数据库中创建用于指定 `ZGXX_DM` 列的用户自定义数据类型 `T_code`。另外，由于职工性别是一个特殊的列，所以，我们为该列创建一个用户自定义数据类型 `T_sex`。

2) 缺省

在数据操作中，使用缺省可以简化数据的输入。在表 `ZGXXB` 中，由于男职工占多数，所以，为 `ZGXX_XB` 列创建缺省 `Def_sex`，指定该列的缺省值为“1”（“1”代表男性，“0”

代表女性);而职工记录的录入时间总是当前的系统时间,所以,我们为列 ZGXX_LRRQ 创建缺省 Def_entrydate,以指定该列的缺省值为当前系统时间。

3) 规则

在数据库中创建规则可以限制插入到表中的数据,使这些数据满足企业规则。在表 ZGXXB 中,由于邮政编码只能由“0”到“9”的数字字符组成,且字符串的长度必须为 6。所以,我们为 ZGXX_YB 列创建规则 Rul_postcode。

4) 索引

在表中创建索引后,当按照索引列查询表中的数据时,可以提高查询速度。在职工考核系统中,用户经常要查询表中的数据。我们在表 ZGXXB、ZGPDB、PDFLB 和 PJGWB 上定义索引。

5) 视图

在实际应用中,使用视图可以简化用户的操作,同时,还可以维护数据的安全性。在职工考核系统中,我们使用视图来限制数据库用户能够查看的评定信息。所以,我们创建视图 User_view 来限制用户,使用户只能查看到具有特殊被评级别的评定信息。

6) 触发器

在数据库中创建触发器可以保证数据更新过程中数据的一致性。在职工考核系统中,工作评定表 ZGPDB 中 ZGXX_DM 列的值必须在职工表 ZGXXB 的 ZGXX_DM 列中存在。所以,我们需要创建触发器 Trg_del 来维护表 ZGXXB 与表 ZGPDB 之间数据的一致性。

7) 存储过程

在数据库中创建存储过程可以提高程序的执行速度。所以,可以将经常执行的操作创建为存储过程,当执行操作时调用该存储过程。在职工考核系统中,用户经常要查看工作评定表中的数据和计算职工的年龄,所以,我们在数据库中创建分别用于实现这两个功能的存储过程 Proc_look 和 Proc_age。

在以后的各章中,我们将介绍这些数据库对象的详细内容。关于如何在样本数据库 ms_database 中创建上面的这些数据库对象,我们将在各章的练习中来实现。下面介绍本书中的练习及其主要内容。

2. 本书中练习的主要内容

在本书中共包含了十二个练习,这些练习都是在样本数据库 ms_database 中进行的。通过对样本数据库中数据的操作,可以使用户基本掌握 Transact-SQL 语言的使用方法。在这些练习中,包含了如何创建样本数据库 ms_database 中的数据库对象,以及对表中数据的各种操作。

下面是这些练习的主要内容。

练习一:建表并对表进行简单的插入、修改、删除和查询操作

✧ 创建职工表 ZGXXB。

✧ 为 ZGXXB 表添加数据。

✧ 查看和修改 ZGXXB 表中的数据。

练习二:表的复杂特征,即自定义数据类型、缺省、规则、索引和触发器

- ✧ 定义两个用户自定义数据类型 “T_code” 和 “T_sex”。
- ✧ 重新创建使用自定义数据类型 T_code 和 T_sex 的 ZGXXB 和 GZPDB 表。
- ✧ 创建缺省 def_sex 和缺省 def_entrydate。
- ✧ 定义规则 rul_postcode。
- ✧ 为 ZGXXB 表的 ZGXXB_xm 列定义唯一非聚簇索引 idx_xm。
- ✧ 为 GZPDB 表创建插入触发器和更新触发器来维护数据的完整性。

练习三：学习使用批命令与流控制语言

- ✧ 在批命令中使用 USE 命令。
- ✧ 创建批命令文件。
- ✧ 用流控制语句编一个程序段。

练习四：学习如何使用事务

- ✧ 编写一个流控制语句与事务混合使用的程序段。
- ✧ 如何嵌套使用事务。
- ✧ SELECT INTO 语句在事务中的使用。

练习五：学习创建存储过程

✧ 定义 proc_look 和 proc_age 两个存储过程，proc_look 用来查看有关工作评定的一些主要信息，proc_age 用来计算职工的年龄。

- ✧ 执行这两个存储过程，通过执行结果来验证它们的正确性。

练习六：连接查询、子查询、带子句的修改和删除语句，大文本数据的插入

- ✧ 一些连接查询、子查询、带子查询的修改和删除等语句的练习。
- ✧ 用 WRITETEXT 插入大文本数据。

练习七：合理地创建索引

- ✧ 把 ZGXXB 和 GZPDB 表创建的索引改为唯一非聚簇索引。
- ✧ 为 PDFLB 和 PJGWB 表定义唯一聚簇索引。

练习八：视图带来的安全性和简单性

✧ 创建视图 user_view，该视图可以限制用户对数据的访问，当以不同的用户登录后，通过该视图可以查看到不同的工作评定信息。

✧ 创建视图 vi_info，应用程序可以通过该视图直接查看到职工的考核情况，从而使应用程序变得简单。

✧ 创建视图 vi_title，通过该视图可以直接查到具有工作评定分类的评定信息的评级工作岗位、分类备注和被评级别。

练习九：使用游标操作表中的数据

- ✧ 定义一个只读游标，用来查看工作评定表中的评定内容。
- ✧ 定义可修改游标，查看并修改评级岗位。

练习十：学习、使用分组存储过程和系统存储过程

✧ 分组存储过程的特点是名称相同，便于管理。在职工考核系统中，职工表与工作评定表中的信息经常被查询。该练习为这两个表定义了两个用来查询信息的分组存储过

程。

✧ 练习使用系统存储过程：Sp_help、Sp_helptext、Sp_who。

练习十一：触发器的特性：嵌套与递归触发

该练习定义了下面两个触发器：

✧ 级联修改触发器 up_zgxx。该触发器的作用是当修改 ZGXXB 表的职工代码时，级联修改 GZPDB 表中的职工代码。

✧ 限制修改触发器 li_gzpd。该触发器只允许有访问相应工作评定权限的用户才可以修改 GZPDB 表中的职工代码。该触发器可以和第一个触发器配合使用。

练习十二：创建 English Query 应用查询数据库中的数据

✧ 创建 English Query 应用查询职工的工作评定情况。

2.3 附录

生成样本数据库的 SQL 文件如下：

```
/*===== */
/* Database name: MODEL_1 */
/* DBMS name: Microsoft SQL Server 7.x */
/* Created on: 99-7-3 23:34 */
/*===== */
```

```
if exists (select 1
           from sysobjects
           where id = object_id('GZPDB_PDFLB')
           and type = 'U')
    drop table GZPDB_PDFLB
go
```

```
if exists (select 1
           from sysobjects
           where id = object_id('GZPDB_PJGWB')
           and type = 'U')
    drop table GZPDB_PJGWB
go
```

```
if exists (select 1
           from sysobjects
```

```

        where id = object_id('GZPDB')
        and type = 'U')
drop table GZPDB
go

if exists (select 1
        from sysobjects
        where id = object_id('BPJBB')
        and type = 'U')
drop table BPJBB
go

if exists (select 1
        from sysobjects
        where id = object_id('PDFLB')
        and type = 'U')
drop table PDFLB
go

if exists (select 1
        from sysobjects
        where id = object_id('PJGW')
        and type = 'U')
drop table PJGW
go

if exists (select 1
        from sysobjects
        where id = object_id('ZGXXB')
        and type = 'U')
drop table ZGXXB
go

/* ===== */
/* Table: ZGXXB */
/* ===== */
create table ZGXXB

```