### 实验一 SQL SERVER 2008的基本知识与操作

实验日期：

### 实验目标：

1. 以SQL Server2008为实验平台掌握SQL Server服务器的注册、启动、暂停、查看与设置服务器的属性等操作；
2. 了解SQL Server Management Studio的功能
3. 学习SQL Server2008所提供的系统数据库；
4. 学习利用SQL Server Management Studio创建数据库。掌握数据库中这主数据文件、辅助数据文件、日志文件三类文件的作用；
5. 学习利用SQL Server Management Studio数据库中创建表；创建表的字段、字段的数据类型、设置表的主键；
6. 学习利用SQL Server Management Studio修改数据库中表的数据，删除表的数据、向表中插入数据；
7. 利用SQL Server Management Studio删除数据库中的表；
8. 利用SQL Server Management Studio删除数据库、分离数据库、附加数据库库。

### 实验内容

SQL Server 2008将数据保存在数据库中，并为用户提供了访问这些数据的接口。对数据库的基本操作包括创建、查看、修改和删除数据库等。在学习这些操作之前，先来了解一下数据库文件和SQL Server系统数据库。

1. **数据库文件**

在SQL Server中数据库文件是存储数据的文件，其可以分为三类。

主数据文件：扩展名是.mdf，它包含数据库的启动信息以及数据库数据，每个数据库只能包含一个主数据文件。在SQL Server中数据的存储单位是页。

辅助数据文件：扩展名是.ndf,因为有些数据库可能非常大，用一个主数据文件可能放不下，因此就需要有一个多个辅助数据文件存储这些数据，可以和主数据文件放在相同的位置也可以存放在不同的位置。

日志文件：用来记录页的分配和释放以及对数据库数据的修改操作，扩展名为.ldf,包含用于恢复数据库的日志信息。每个数据库必须至少有一个日志文件，也可以有多个。

创建数据库时，一个数据库至少包含一个主数据文件和一个或多个日志文件，还可能包含一些辅助数据文件。这些文件默认的位置为：\program files\Microsoft SQL Server\MSSQL\Data文件夹。

1. **SQL Server系统数据库**

在 SQL Server 2008有两类数据库：系统数据库和用户数据库。系统数据库存储有关SQL Server 的系统信息，它们是SQL Server 2008管理数据库的依据。如果系统数据库遭到破坏，那么SQL Server将不能正常启动。在安装了SQL Server 2008的系统共中将创建4个可见系统数据库。

（1）master数据库

master数据库是SQL Server中最重要的数据库，它是SQL Server的核心数据库，如果该数据库被损坏，SQL Server将无法正常工作，master数据库中包含所有的登录名或用户ID所属的角色、服务器中的数据库的名称及相关的信息、数据库的位置、SQL Server如何初始化四个方面的重要信息。

（2）model数据库

用户创建数据库时是以一套预定义的标准为模型。例如，若希望所有的数据库都有确定的初始大小，或者都有特定的信息集，那么可以吧这些信息放在model数据库中，以model数据库作为其他数据库的模板数据库。如果想要使用所有的数据库都有一个特定的表，可以把该表放在model数据库里。model数据库是tempdb数据库的基础。对model数据库的任何改动都将反映在tempdb数据库中，所以，在决定对model数据库有所改变时，必须预先考虑好。

（3）msdb数据库

msdb数据库给SQL Server代理提供必要的信息来运行作业，其供SQL Server 2008代理程序调度警报作业以及记录操作时使用。

（4）tempdb数据库

tempdb数据库用作系统的临时存储空间，其主要作用是存储用户建立的临时表和临时存储过程，存储用户说明的全局变量值，为数据排序创建临时表，存储用户利用游标说明所筛选出来的数据。

1. **创建数据库**

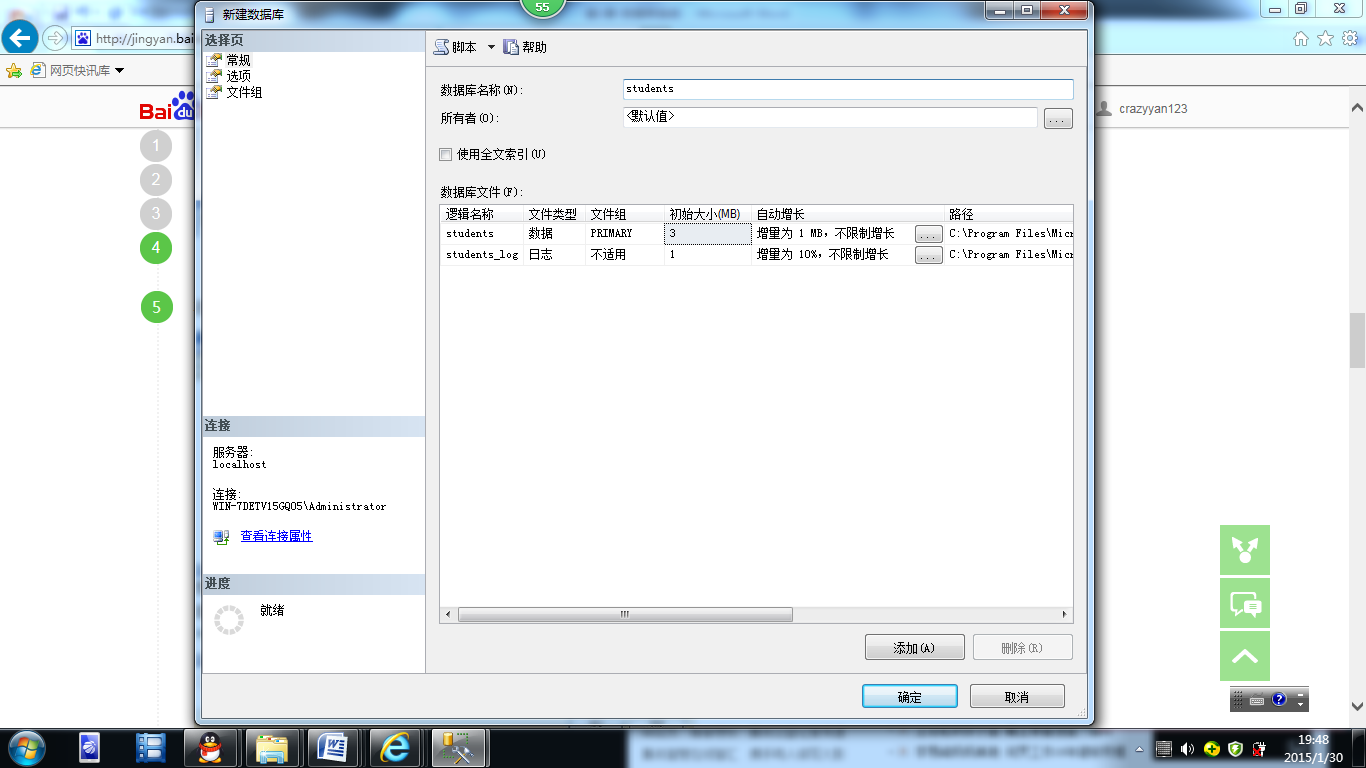
选择开始菜单中→程序→【Management SQL Server 2008】→【SQL Server Management Studio】命令，打开【SQL Server Management Studio】窗口，并使用Windows或 SQL Server身份验证建立连接。

图1.1 新建数据库

在【对象资源管理器】窗口中展开服务器，然后选择【数据库】节点。右键单击【数据库】节点，从弹出来的快捷菜单中选择【新建数据库】命令。执行上述操作后，会弹出【新建数据库】对话框。在对话框、左侧有3个选项，分别是【常规】、【选项】和【文件组】。完成这三个选项中的设置会后，就完成了数据库的创建工作，在【数据库名称】文本框中输入要新建数据库的名称。例如，这里以“Students”。

【数据库文件】列表中包括两行，一行是数据库文件，而另一行是日志文件。通过单击下面的【添加】、【删除】按钮添加或删除数据库文件，如图1.1。

【逻辑名称】指定该文件的文件名。

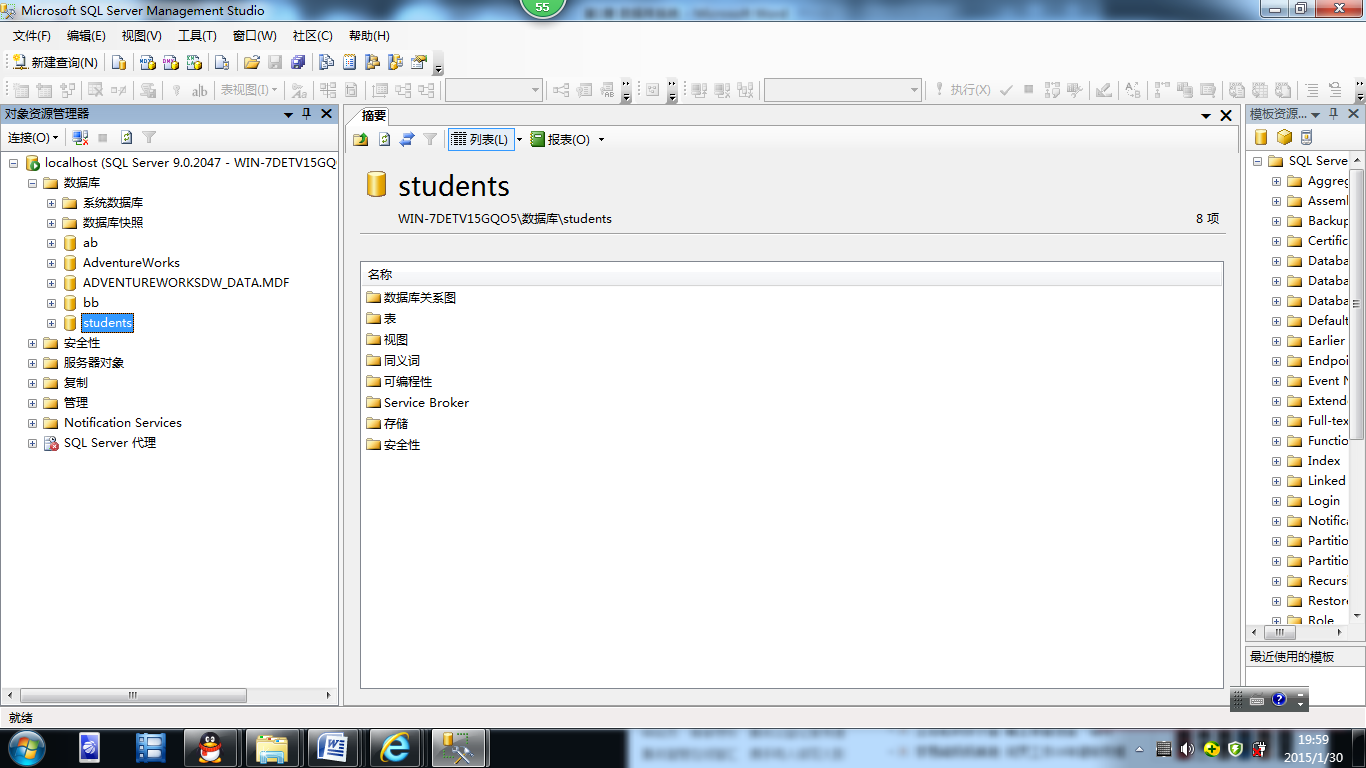
【文件类型】用于区别当前文件是数据文件还是日志文件。

【文件组】显示当前数据库文件所属的文件组。一个数据只能存在一个文件组里。

【初始大小】指定该文件的初始容量。默认值为3MB.日志默认值为1MB。

【自动增长】用于设置文件的容量不够用时，文件根据何中增长方式自动增长。

完成以上操作后，单击【确定】按钮关闭【新建数据库】对话框。至此“新建的数据”数据库创建成功。新建的数据库可以再【对象资源管理器】窗口看到，如下图1.2所示。

图1.2建好后的数据库Students

1. **修改数据库**

建立一个数据库之后，可以根据需要对该数据库的结构进行修改。

启动SSMS，在“对象资源管理器”窗格中展开数据库节点，右击要修改的数据库名称，在弹出的快捷菜单中选择“属性”命令，打开“数据库属性”对话框。可以通过修改数据库属性来修改数据库。修改数据库的操作包括增减数据库文件、修改文件属性（包括数据库的名称、大小和属性）、修改数据库选项等。

1. **删除数据库**

为了减少系统资源的消耗，对于不再需要的用户创建数据库，应当把它从数据库服务器中删除，从而将其所占的磁盘空间全部释放掉。

删除数据库的具体操作如下：

启动SSMS，在“对象资源管理器”窗格中展开数据库节点，右击要删除的数据库名称，在弹出的快捷菜单中选择“删除”命令，打开“删除对象”对话框，单击“确定”按钮，数据库就被删除。

1. **分离和附件数据库**

当数据库需要从一台计算机移到另一台计算机，或者需要从一个物理磁盘移到另一个物理磁盘时，常要进行数据库的附加与分离操作。

附加数据库是指将当前数据库以外的数据库附加到当前数据库服务器中。

附加数据库的具体操作如下：

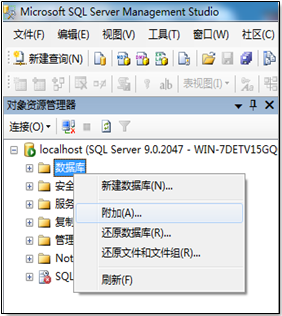
启动SSMS，在“对象资源管理器”窗格中右击“数据库”节点在快捷菜单中选择“附加”命令，打开“附加数据库”对话框，如图1.3所示，单击“添加”按钮，打开“定位数据库文件”对话框，选择要附加的数据库主数据文件（.mdf），单击“确定”按钮，返回上述“附加数据库”对话框，单击“确定”按钮，完成数据库的附加操作。

图1.3 附加数据库

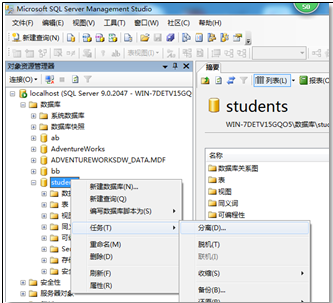


图1.4 分离数据库

分离数据库就是讲数据库从SQLserver2008服务器中卸载，但依然保存数据库的数据文件和日志文件。需要时，分离的数据库，可以重新附加到SQL server2008服务器中。

分离数据库的具体操作如下：

启动SSMS，在“对象资源管理器”窗格中展开数据库节点，右击要分离的数据库名称，在弹出的快捷菜单中单击“任务”→“分离”命令，如图1.4所示，打开“分离数据库”对话框，单击“确定”按钮，实现数据库的分离。

### 简答题：

1. 打开SQL Server Management Studio时，怎样进行数据库的登录认证？
2. 对象资源管理器的作用是什么？

### 实验二 管理数据库操作

实验日期：

### 实验目标：

（1）熟练掌握用SQL语句创建数据库的方法。

（2）熟练掌握数据库属性的设置。

（3）掌握数据库的修改和删除方法。

### 实验内容：

（1）利用SQL Server Management Studio创建一个”SM”的数据库，初始大小为3MB,最大为50MB，数据库自动增长，增长方式按10%；日志文件初始大小为2MB，最大值不受限制，按1MB增长。

（2）通过SQL语句创建一个名为“学籍”的数据库，指定主文件名为“学籍\_data”，存储路径为“d:\example\学籍\_data.mdf”，该数据文件的初始大小为10MB，最大为100MB，增长方式按10MB增长；指定主日志文件名为“学籍\_log”，存储路径为“d:\example\学籍\_log.ldf”，该日志文件初始大小为20MB，最大为200MB，按10MB增长。

1. 在工具栏单击“新建查询”按钮，打开代码编辑器，输入如下SQL语句。

Create database 学籍

On primary

(name=学籍\_data,

Filename=’d:\example\学籍\_data.mdf’,

Size=10,

Maxsize=100,

Filegrowth=10)

Log on

(name=学籍\_log,

Filename=’d:\example\学籍\_log.ldf’,

Size=20,

Maxsize=200,

Filegrowth=10)

请执行该语句，即可创建“学籍”数据库。

1. 使用对象资源管理器，可以修改“学籍”数据库。选择“学籍”数据库的节点，点击右键，在弹出的快捷菜单中选择“属性”命令，打开“数据库属性”对话框，即可进行数据库的修改操作。
2. 使用”SQL”语句，在“学籍”数据库中添加一个数据文件“学籍\_data1”，指定其初始大小为4MB，最大不受限制，增长方式按10%增长。

Alter database 学籍

Add file

(name=’学籍\_data1’,

Filename=’d:\example\学籍\_data1.mdf’,

Size=4,

Maxsize=unlimited,

Filegrowth=10%)

Go

1. 使用SQL语句删除“学籍”数据库。

Drop database 学籍

GO

提示：当数据库正在参与复制是不能删除，当数据库正在被复制时不能删除，当有用户正在使用数据库时不能删除。

### 简答题：

（1）SQL Server2008中的数据库能否只包含数据文件？

### 实验三 用T-SQL创建与修改表的操作

实验日期：

### 实验目标：

（1）熟练掌握各种数据类型。

（2）熟练运用T-SQL（Transact-SQL）语句正确创建表。

（3）掌握常用的数据完整性约束。

### 实验内容：

（1）创建“学籍”数据库，在该数据库中创建Student表。

表3.1 Student表的结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **长度** | **字段说明** | **能否为空** | **是否主键** |
| Sno | CHAR | 10 | 学号 | 否 | 是 |
| Sname | CHAR | 10 | 姓名 | 是 | 否 |
| Ssex | CHAR | 2 | 性别 | 是 | 否 |
| Sage | TINYINT |  | 年龄 | 是 | 否 |
| Sdept | CHAR | 10 | 系 | 是 | 否 |

利用T-SQL命令创建Student表，表的结构如表3.1所示。

代码如下：

CREATE TABLE Student

( Sno CHAR(10) primary key,

Sname CHAR(10),

Ssex CHAR(2) check (Ssex in (‘男’,’女’)),

Sage TINYINT,

Sdept CHAR(10)

)

（2）利用T-SQL命令创建Course表，表的结构如表3.2所示。

表3.2 Course表的结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **长度** | **字段说明** | **能否为空** | **是否主键** |
| Cno | CHAR | 5 | 课程号 | 否 | 是 |
| Cname | CHAR | 10 | 课程名 | 否 | 否 |
| Credits | TINYINT |  | 学分 | 是 | 否 |
| PCno | CHAR | 5 | 先修课程号 | 是 | 否 |

代码如下：

CREATE TABLE Course (

Cno char(5) NOT NULL,

Cname char(10) NOT NULL,

PCno char(5),

Credit tinyint CHECK (Credit > 0),

PRIMARY KEY(Cno),

FOREIGN KEY (PCno) REFERENCES Course(Cno),

)

（3）为已经创建了的Student表创建一个约束，规定在Student表中，如果学生没有提供所在系，就写入默认值“计算机系”。df\_dept为该约束的名。

代码如下：

Alter table Student

Add constraint df\_dept

Default ‘计算机系’ for Sdept

（4）利用T-SQL命令创建SC表，表的结构如表3.3所示。

表3.3 SC表的结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **长度** | **字段说明** | **能否为空** | **是否主键** |
| Sno | CHAR | 10 | 学号 | 否 | 是 |
| Cno | CHAR | 5 | 课程号 | 否 | 是 |
| Grade | TINYINT |  | 成绩 | 是 | 否 |

代码如下：

CREATE TABLE SC(

Sno char(10) FOREIGN KEY(Sno) REFERENCES Student(Sno) ON DELETE CASCADE,

Cno char(5) FOREIGN KEY(Cno) REFERENCES Course(Cno),

Grade tinyint CHECK(grade<=100 and grade>=0),

PRIMARY KEY(Sno,Cno)

)

执行并思考Sno char(10) FOREIGN KEY(Sno) REFERENCES Student(Sno) ON DELETE CASCADE语句中ON DELETE CASCADE的含义与作用。

### 问答题：

（1）T-SQL支持的数据类型有哪些？

（2）Tinyint数据类型定义的数据的取值范围是多少？

（3）写出创建以下表3.4-3.6所示的表结构的SQL语句，要求在定义表的同时定义数据的完整性约束。

表3.4 图书表

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **约束** |
| 书号 | 统一字符编码定长类型，长度为6 | 主键 |
| 书名 | 统一字符编码可变长类型，长度为30 | 非空 |
| 第一作者 | 普通编码定长字符，长度为10 | 非空 |
| 出版日期 | 小日期时间型 | 非空 |
| 价格 | 定点小数，小数部分1位，整数部分3位 |  |

表3.5 书店表

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **约束** |
| 书店编号 | 统一字符编码定长类型，长度为6 | 主键 |
| 店名 | 统一字符编码可变长类型，长度为30 | 非空 |
| 电话 | 普通编码定长字符类型，长度为8 |  |
| 地址 | 普通编码可变长字符类型，长度40 |  |
| 邮政编码 | 普通编码定长字符类型，长度6 |  |

表3.6 销售表

|  |  |  |
| --- | --- | --- |
| **列名** | **数据类型** | **约束** |
| 书号 | 统一字符编码定长类型，长度为6 | 主键，外键 |
| 书店编号 | 统一字符编码定长类型，长度为6 | 主键，外键 |
| 销售日期 | 小日期时间型 | 主键 |
| 销售数量 | 小整型 |  |
| 邮政编码 | 普通编码定长字符类型，长度6 |  |

（4）删除“销售表”中的“邮政编码”列；

（5）将“销售表”中的“销售数量”列的数据类型改为整型。

### 实验四 数据查询的操作（一）

实验日期：

### 实验目标：

（1）掌握SELECT子句以及WHERE子句的使用方法。

（2）学会应用ORDER BY子句。

（3）掌握5种基本的聚合函数。

（4）学会分组统计和二次查询。

（5）学会数据的导入与导出的方法。

### 实验内容：

在SELECT语句中使用WHERE子句，在WHERE子句后跟一个条件式，就可以查询满足条件的行。

WHERE子句的格式：WHERE〈条件〉。

表4.1 常用的查询条件

|  |  |  |
| --- | --- | --- |
| 查询条件 | 运算符 | 说明 |
| 比较 | =、<、<=、>、>=、<>、  !=、!<、!> |  |
| 范围谓词 | BETWEEN AND、  NOT BETWEEN AND | 介于二者之间（包括两端）  不介于二者之间 |
| 集合谓词 | IN  NOT IN | 在集合中  不在集合中 |
| 字符匹配谓词 | LIKE、  NOT LIKE | 匹配  不匹配 |
| 空值 | IS NULL  IS NOT NULL | 是空值  不是空值 |
| 逻辑运算 | NOT、AND、OR | 逻辑非、逻辑与、逻辑或 |

打开对象资源管理器，创建一个名为”lianxi1”的数据库，在该数据库中导入以下的仓库表4.2和职工表4.3。

表4.2 仓库表

|  |  |  |
| --- | --- | --- |
| **仓库号** | **城市** | **面积** |
| WH1 | 北京 | 370 |
| WH2 | 上海 | 500 |
| WH3 | 广州 | 200 |
| WH4 | 广州 | 300 |
| WH5 | 天津 | 340 |
| WH6 | 上海 | 350 |
| WH7 | 上海 | 600 |
| WH8 | 天津 | 300 |

注意：面积以平方为单位。

表4.3 职工表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **职工号** | **姓名** | **仓库号** | **月工资** | **性别** |
| E1 | 朱迪 | WH2 | 2220 | 女 |
| E2 | 牛丽丽 | WH1 | 1810 | 女 |
| E4 | 李安 | WH2 | 1850 | 男 |
| E5 | 王进步 | WH3 | 1530 | 男 |
| E6 | 李光铭 | WH1 | 1550 | 男 |
| E7 | 赵芙蓉 | WH4 | 2300 | 女 |
| E8 | 刘山 | WH4 | 2000 | 男 |
| E9 | 张尚琳 | WH5 | 2050 | 女 |
| E10 | 王玛丽 | WH5 | 1900 | 女 |
| E11 | 胡尼克 | WH6 | 2100 | 男 |
| E12 | 古梅 | WH7 | 1700 | 女 |
| E15 | 胡俊 | WH5 | 1780 | 男 |
| E16 | 胡轩 | WH1 | 1600 | 男 |

用SQL语句完成以下的练习题：

1. 从职工关系中检索所有工资值。

select 月工资

from zg

1. 检索仓库关系中的所有元组。

select \*

from ck

1. 检索工资多于2000元的职工对应的职工号，姓名。

select 职工号,月工资 from zg where 月工资>2000

1. 检索哪些仓库的面积在400到550平方之间，列出仓库号和所在的城市。

select 仓库号,城市 from ck where 面积>400 and 面积<550

1. 检索出有广州哪些仓库，列出仓库号。

select 仓库号 from ck where 城市='广州'

1. 检索出所在城市为广州的仓库的个数。

select count(仓库号)

from ck where 城市='广州

1. 检索出所有职工的平均工资。

select avg(月工资) from zg

1. 检索出在WH5仓库工作的职工的人数。

select count(仓库号) from zg where 仓库号='WH5'

1. 检索出职工表的所有的仓库号，去掉重复值。

select distinct 仓库号 from zg

1. 检索出每个仓库的人数。

select 仓库号,count(\*)人数 from zg group by 仓库号

1. 检索出上海的仓库有哪些，列出仓库号、面积。查询结果按面积降序排列。

select 仓库号,面积 from ck where 城市='上海' order by 面积

1. 检索出哪些仓库女职工的人数达到了3人。

select 仓库号,count(职工号)人数

from zg where 性别='女'

group by 仓库号 having count(职工号)>3

1. 检索出所有姓胡的职工的职工号、姓名、性别、工资。查询结果按性别排降序、工资排升序。
2. 检索出职工“王玛丽”的年工资。
3. 检索出“WH1”和“WH2”两个仓库工作的职工的基本信息。
4. 检索出年工资在24000以上的职工的姓名，年工资值。

### 问答题：

（1）WHERE子句与HAVING子句有和不同？

（2）在WHERE子句中可以使用的运算符有哪些？WHERE子句中能不能使用聚集函数？

### 实验五 数据查询的操作（二）

实验日期：

### 实验目标：

（1）熟练掌握基本的连接操作，掌握内连接与外连接的方法，学会应用自身连接。

（2）熟练掌握不相关子查询。

（3）掌握相关子查询。

（4）学会给表命别名。

（5）学会数据的导入与导出的方法。

### 实验内容：

打开对象资源管理器，创建一个名为”lianxi2”的数据库，在该数据库中导入以下的仓库表5.1、职工表5.2、订购表5.3和供应商表5.4。

表5.1仓库表

|  |  |  |
| --- | --- | --- |
| 仓库号 | 城市 | 面积 |
| WH1 | 北京 | 3700 |
| WH2 | 上海 | 5000 |
| WH3 | 广州 | 2000 |
| WH4 | 武汉 | 4000 |
| WH5 | 上海 | 4560 |
| WH6 | 广州 | 6700 |
| WH7 | 珠海 | 4800 |

表5.2职工表

|  |  |  |
| --- | --- | --- |
| 仓库号 | 职工号 | 月工资 |
| WH2 | E1 | 2220 |
| WH1 | E2 | 2210 |
| WH2 | E3 | 4050 |
| WH3 | E4 | 3230 |
| WH1 | E5 | 3250 |
| WH1 | E6 | 2300 |
| WH4 | E7 | 5000 |
| WH5 | E8 | 4000 |
| WH5 | E9 | 3400 |
| WH6 | E10 | 3800 |

表5.3 订购表

|  |  |  |  |
| --- | --- | --- | --- |
| 职工号 | 供应商号 | 订购单号 | 订购日期 |
| E3 | S7 | OR091204 | 2009-12-4 |
| E1 | S4 | OR090101 | 2009-4-1 |
| E7 | S4 | OR100402 | 2010-4-2 |
| E6 | S6 | OR100121 | 2010-1-21 |
| E3 | S4 | OR091115 | 2009-11-15 |
| E1 | S6 | OR060201 | 2006-2-1 |
| E3 | S6 | OR100312 | 2010-3-12 |
| E3 | S3 | OR090302 | 2009-3-2 |
| E8 | S7 | OR100928 | 2010-9-28 |
| E6 | S7 | OR100712 | 2010-7-12 |
| E5 | S3 | OR100507 | 2010-5-7 |

表5.4 供应商表

|  |  |  |
| --- | --- | --- |
| 供应商号 | 供应商名 | 地址 |
| S3 | 振华电子厂 | 西安 |
| S4 | 华通电子公司 | 北京 |
| S6 | 607厂 | 郑州 |
| S7 | 爱华电子厂 | 北京 |

用SQL语句完成以下的练习题：

1. 查询每个城市的仓库总面积。
2. 查询每个仓库的职工人数，如果该仓库没有职工，也要列出人数为0。
3. 查询在上海工作的职工有多少个。
4. 查询哪些职工在上海工作，列出他们的职工号，仓库号。
5. 查询与E3号职工有订购业务联系的供应商号，供应商名。
6. 查询哪些职工与爱华电子厂有订购业务联系，列出他们的职工号，仓库号。
7. 查询每个供应商的订购单数目，列出供应商号和他们的订购单数目。
8. 查询工资在3000元以上，并在北京或上海工作的职工，列出他们的职工号和工资。
9. 查询与供应商为S3的供应商有业务联系的职工他们的职工号、订购单号、仓库号、城市。
10. 查询出哪些仓库没有分配职工。
11. 查询出哪些职工的工资高于全体职工平均工资。
12. 查询出哪些职工的工资高于他所在仓库的职工的平均工资。
13. 求出哪个职工所发出的订购单最多。

### 实验六 数据查询的操作（三）

实验日期：

### 实验目标：

（1）熟练掌握基本的连接操作，掌握内连接与外连接的方法，学会应用自身连接。

（2）熟练掌握不相关子查询。

（3）掌握相关子查询。

（4）学会给表命别名。

（5）学会在FROM子句中派生关系，构造查询。

### 实验内容：

假设“教学管理”数据库中有4个基本表（关系）：

学生关系：学生（学号，姓名，性别，所在系）

课程关系：课程（课程号，课程名，开课学期，教师号）

选课关系：选课（学号，课程号，成绩）

教师关系：教师（教师号，姓名，职称，部门）

每个关系的当前值分别如表所示。

打开对象资源管理器，创建一个名为”教学管理”的数据库，在该数据库中导入以下的表6.1—6.4。

表6.1 学生关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号** | **姓名** | **性别** | **年龄** | **所在系** |
| S101101 | 陈名军 | 男 | 18 | 计算机系 |
| S101102 | 吴小晴 | 女 | 19 | 计算机系 |
| S101103 | 王明燕 | 女 | 19 | 计算机系 |
| S101104 | 严利 | 男 | 20 | 计算机系 |
| S101105 | 朱欣 | 男 | 20 | 计算机系 |
| S101201 | 李国庆 | 男 | 21 | 信息系 |
| S101202 | 李祥 | 男 | 21 | 信息系 |
| S101203 | 孙渝研 | 男 | 20 | 信息系 |
| S101204 | 赵艳 | 女 | 18 | 信息系 |
| S101205 | 刘唯 | 女 | 19 | 信息系 |
| S101206 | 林玉霞 | 女 | 20 | 信息系 |
| S101207 | 王江 | 男 | 21 | 信息系 |
| S101301 | 王成 | 男 | 20 | 会计系 |
| S101302 | 张平安 | 男 | 18 | 会计系 |
| S101401 | 钟琴 | 女 | 19 | 会计系 |
| S101402 | 吴娟娟 | 女 | 21 | 会计系 |
| S101403 | 李月 | 女 | 22 | 会计系 |
| S101404 | 陈名军 | 男 | 23 | 会计系 |
| S101405 | 赵艳 | 女 | 21 | 会计系 |

表6.2课程关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **课程号** | **课程名** | **开课学期** | **学分** | **教师号** |
| 101 | 计算机基础 | 1 | 3 | T1 |
| 102 | 体育 | 2 | 4 | T2 |
| 201 | 英语 | 1 | 4 | T3 |
| 202 | 大学语文 | 3 | 4 | T4 |
| 305 | 操作系统 | 4 | 4 | T5 |
| 304 | 计算机原理 | 4 | 4 | T5 |
| 301 | 计算机网络 | 3 | 3 | T6 |
| 302 | 电子技术 | 3 | 4 | T6 |
| 303 | 数据库应用 | 4 | 3 | T7 |

表6.3 选课关系

|  |  |  |
| --- | --- | --- |
| **学号** | **课程号** | **成绩** |
| S101101 | 101 | 60 |
| S101101 | 102 | 83 |
| S101101 | 201 | 78 |
| S101101 | 202 | 87 |
| S101101 | 305 | 79 |
| S101101 | 304 | 89 |
| S101101 | 303 | 64 |
| S101101 | 302 | 90 |
| S101101 | 301 | 83 |
| S101102 | 101 | 84 |
| S101102 | 102 | 75 |
| S101102 | 202 | 86 |
| S101102 | 303 | 67 |
| S101201 | 101 | 78 |
| S101201 | 102 | 72 |
| S101201 | 303 | 76 |
| S101201 | 201 | 50 |
| S101301 | 101 | 90 |
| S101302 | 101 | 90 |
| S101302 | 303 | 83 |

表6.4教师表

|  |  |  |  |
| --- | --- | --- | --- |
| 教师号 | 姓名 | 职称 | 部门 |
| T1 | 胡美丽 | 讲师 | 公共教学 |
| T2 | 王珊珊 | 讲师 | 公共教学 |
| T3 | 王新 | 讲师 | 公共教学 |
| T4 | 李再敏 | 副教授 | 公共教学 |
| T5 | 李红玉 | 教授 | 计算机系 |
| T6 | 周进 | 助教 | 计算机系 |
| T7 | 张丽丽 | 助教 | 计算机系 |
| T8 | 王晓舟 | 副教授 | 计算机系 |
| T9 | 周樱 | 讲师 | 信息系 |

1.查询“101”课程成绩比“102”课程成绩高的所有学生的学号。

提示：

select a.学号

From (select 学号,成绩 from 选课 where 课程号='101') a,(select 学号,成

from 选课 where 课程号='102') b

where a.成绩>b.成绩 and a.学号=b.学号;

2.查询全部选修课程表中所有课的同学的学号、姓名。

3.查询哪些老师没有教授任何课程。列出老师的全部列。（not in和 not EXISTS 两种方式实现）。

4.查询出只选修了一门课程的学生的学号和姓名。

5.查询张丽丽老师教过的学生的学号、姓名。

6.查询相同姓名的学生，列出相同的姓名及相应的同名人数。（有几个学生同名）

提示：考虑按姓名分组，分组内元组的个数。

7.查询每门功成绩最好的前两名。

提示：

SELECT t1.学号,t1.课程号,成绩

FROM 选课 t1

WHERE 成绩 IN (SELECT TOP 2 成绩

FROM 选课

WHERE 课程号= t1.课程号

ORDER BY 成绩 DESC

)

这道题每次父亲查询都提供课程号的值给子查询，其子查询的功能就是计算父查询对应课程的前两名。

8.查询学过“101”并且也学过编号“102”课程的同学的学号、姓名。

9.查询没有选修课程表中所有课的同学的学号、姓名。

10.查询全部学生都选修过的课程对应的课程号和课程名。

提示：

意味着该课程的选课人数与学生总人数相等。考虑按课程号分组，查询哪个分组中学号的个数与学生个数相等，得到这样的分组它的课程号。

11.查询没学过“李红玉”老师讲授的任一门课程的学生姓名。（not in不相关即可实现）

### 实验七 视图与索引

实验日期：

### 实验目标：

（1）掌握创建视图、删除视图；

（2）查询视图。

（3）更新视图、修改视图对应的数据。

（4）理解索引的概念和索引的作用，学会使用索引。

（5）了解聚簇索引和非聚簇索引。

### 实验内容：

假设“学生选课”数据库中有3个基本表（关系）：

学生关系：学生（学号，姓名，性别，所在系）

课程关系：课程（课程号，课程名，开课学期，教师号）

选课关系：选课（学号，课程号，成绩）

每个关系的当前值分别如表所示。

打开对象资源管理器，创建一个名为”学生选课”的数据库，在该数据库中导入以下的表7.1—7.3。

表7.1 学生关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号** | **姓名** | **性别** | **年龄** | **所在系** |
| S101101 | 陈名军 | 男 | 18 | 计算机系 |
| S101102 | 吴小晴 | 女 | 19 | 计算机系 |
| S101103 | 王明燕 | 女 | 19 | 计算机系 |
| S101104 | 严利 | 男 | 20 | 计算机系 |
| S101105 | 朱欣 | 男 | 20 | 计算机系 |
| S101201 | 李国庆 | 男 | 21 | 信息系 |
| S101202 | 李祥 | 男 | 21 | 信息系 |
| S101203 | 孙渝研 | 男 | 20 | 信息系 |
| S101204 | 赵艳 | 女 | 18 | 信息系 |
| S101205 | 刘唯 | 女 | 19 | 信息系 |
| S101206 | 林玉霞 | 女 | 20 | 信息系 |
| S101207 | 王江 | 男 | 21 | 信息系 |
| S101301 | 王成 | 男 | 20 | 会计系 |
| S101302 | 张平安 | 男 | 18 | 会计系 |
| S101401 | 钟琴 | 女 | 19 | 会计系 |
| S101402 | 吴娟娟 | 女 | 21 | 会计系 |
| S101403 | 李月 | 女 | 22 | 会计系 |
| S101404 | 陈名军 | 男 | 23 | 会计系 |
| S101405 | 赵艳 | 女 | 21 | 会计系 |

表7.2课程关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **课程号** | **课程名** | **开课学期** | **学分** | **教师号** |
| 101 | 计算机基础 | 1 | 3 | T1 |
| 102 | 体育 | 2 | 4 | T2 |
| 201 | 英语 | 1 | 4 | T3 |
| 202 | 大学语文 | 3 | 4 | T4 |
| 305 | 操作系统 | 4 | 4 | T5 |
| 304 | 计算机原理 | 4 | 4 | T5 |
| 301 | 计算机网络 | 3 | 3 | T6 |
| 302 | 电子技术 | 3 | 4 | T6 |
| 303 | 数据库应用 | 4 | 3 | T7 |

表7.3 选课关系

|  |  |  |
| --- | --- | --- |
| **学号** | **课程号** | **成绩** |
| S101101 | 101 | 60 |
| S101101 | 102 | 83 |
| S101101 | 201 | 78 |
| S101101 | 202 | 87 |
| S101101 | 305 | 79 |
| S101101 | 304 | 89 |
| S101101 | 303 | 64 |
| S101101 | 302 | 90 |
| S101101 | 301 | 83 |
| S101102 | 101 | 84 |
| S101102 | 102 | 75 |
| S101102 | 202 | 86 |
| S101102 | 303 | 67 |
| S101201 | 101 | 78 |
| S101201 | 102 | 72 |
| S101201 | 303 | 76 |
| S101201 | 201 | 50 |
| S101301 | 101 | 90 |
| S101302 | 101 | 90 |
| S101302 | 303 | 83 |

用学生、课程、选课这三张表，利用SQL语句完成以下练习题。

（1）创建V1视图，学生的学号、姓名、所在系、课程号、课程名、课程学分的视图。

（2）创建V2视图，查个学生的平均成绩的视图，要求列出学生学号及平均成绩。

（3）创建V3视图，每个学生的修课学分的视图，要求列出学生学号及总学分。

（4）上面的视图V3能否对其总学分对应的数据进行修改？

（5）创建V4视图，计算机系的学生基本信息的视图。该视图能否更新学生的姓名？（无需选课的信息）。

（6）创建V5视图，每个学生获得的最高成绩，要求列出学号和最高成绩

（7）借助视图V5，实现该查询，对每个学生找出他获得最高成绩的课程号。

（8）删除视图V1。

（9）为学生关系的姓名列创建一个非聚簇索引。

（10）为课程表的课程名创建一个聚簇索引。

### 简答题：

1. 视图可以加快数据的查询速度，这句话对吗？为什么？
2. 视图可以简化查询，这句话对吗？为什么？
3. 视图是一个虚表，数据库中只存放视图的定义，而不存放视图包含的数据，这些数据仍存放在原来基本表中。这句话对吗？
4. 基本表中的数据如果发生变化，视图中查询出数据也随之变化。这句话对吗？
5. 索引是否越多越好？
6. 哪些视图不能更新？
7. 聚簇索引和非聚簇索引有何区别，为什么每个表只能有一个聚簇索引？

### 实验八 数据操作

实验日期：

### 实验目标：

（1）掌握插入数据到表中。

（2）掌握删除表中的数据。

（3）掌握修改表中的数据。

### 实验内容：

建立一个名叫“仓库职工”的数据库，接下来请利用实验五的数据，将表中的数据导入到“仓库职工”数据库中。用T-SQL语句完成以下练习题。

1. 查询每个城市的仓库的总面积，将查询的结果插入到新表t1中，该表需要自己创建。

创建表t1的代码如下：

Create table t1

(cityname char(20),

Sumarea int

)

运行代码创建了表t1。

接下来向t1中插入数据，其中数据为某个子查询的结果。代码提示如下：

Insert into t1

Select 城市,sum(面积)

From 仓库

Group by 城市

1. 插入一个新的供应商元组（S9，智通公司，沈阳）
2. 北京的所有仓库增加100M2的面积。
3. 给低于所有职工平均工资的职工提高5%（注意要用0.05表示5%）的工资。
4. 给北京的职工加900元工资。（相关子查询，不相关子查询，多表三种方法实现 ）。

用相关子查询实现的代码如下：

Update 职工

Set 月工资=月工资+900

Where ‘北京’=(select 城市

From 仓库

Where 仓库.仓库号=职工.仓库号)

用不相关子查询实现的代码如下：

Update 职工

Set 月工资=月工资+900

Where 仓库号=(select 仓库号

From 仓库

Where 城市=’北京’)

1. 删除目前没有任何订购单的供应商。
2. 删除由在上海仓库工作的职工发出的所有订购单。

### 实验九 流程控制语句（选做）

实验日期：

### 实验目标：

（1）掌握流程控制关键字。

（2）查询流程控制关键字的使用方法。

### 实验内容：

流程控制语句是指那些用来控制程序执行和流程分支的语句，在SQL Server 2008中，流程控制语句用来控制SQL语句、语句块或者存储过程的执行流程。

Transact-SQL 语言使用的流程控制命令与常见的程序设计语言类似，主要有以下几种控制命令。

假设有“学生选课”的数据库，数据如实验七。

（1）使用IF...ELSE 语句

IF**...**ELSE语句是条件判断语句，其中，ELSE子句是可选的，最简单的IF语句没有ELSE子句部分。IF**...**ELSE语句用来判断当某一条件成立时执行某段程序，条件不成立时执行另一段程序。SQL Server允许嵌套使用IF**...**ELSE语句，而且嵌套层数没有限制。

IF**...**ELSE语句的语法形式为：

IF <布尔表达式>

<SQL语句>|<语句块>

[ELSE

<SQL语句>|<语句块>]

**例** 在Student表中查询是否有‘张力’这个学生。如果有，则显示这个学生的姓名和系，否则显示没有此人。

USE 学生选课

GO

DECLARE @message VARCHAR(20)

IF EXISTS(SELECT \* FROM Student WHERE SNAME='张力')

SELECT SNAME,SDEPT FROM Student WHERE SNAME='张力'

ELSE

BEGIN

SET @message='没有此人'

PRINT @message

END

**例** 在SC表中查询是否有成绩大于90分的学生，有则输出有学生的成绩高于90分，否则输出没有学生的成绩高于90分。

USE STUDENTS

GO

DECLARE @message VARCHAR(20)

IF EXISTS(SELECT \* FROM SC WHERE GRADE>90)

PRINT '有学生的成绩高于90分'

ELSE

BEGIN

SET @message='抱歉，没有学生的成绩高于90分' PRINT @message

END

（2）使用BEGIN...END 语句

在控制流程中需要执行两条或两条以上的语句，应该将这些语句定义为一个语句块（称为复合语句）。BEGIN和END必须成对实现。

语法格式：

BEGIN

<SQL语句>|<语句块>

END

（3）使用CASE语句

CASE结构提供比IF……ELSE结构更多的选择和判断的机会。使用CASE表达式可以很方便的实现多重选择的情况，从而可以避免编写多重的IF……ELSE嵌套循环。CASE语句按照使用形式不同，可以分为简单CASE语句和搜索CASE语句，它们的语法形式分别为：

**1. 简单CASE函数：**

CASE <表达式>

WHEN <表达式> THEN <表达式>

**…**

WHEN <表达式> THEN <表达式>

[ELSE <表达式>]

END

**例** 从学生表Student中，选取SNO，SSEX，如果SSEX为“男”则输出“M”，如果为“女”输出“F”。

SELECT SNO,SSEX=

CASE SSEX

WHEN '男' THEN 'M'

WHEN '女' THEN 'F'

END

FROM Student

运行结果如图8.10所示。

**2. CASE搜索函数：**

CASE

WHEN <条件表达式> THEN <表达式>

**…**

WHEN <条件表达式>

THEN <表达式>

[ELSE <表达式>]

END

**例** 从SC表中查询所有同学选课成绩情况，凡成绩为空者输出“未考”、小于60分输出“不及格”、60分至70分输出“及格”、70分至90分输出“良好”、大于或等于90分时输出“优秀”。

SELECT SNO,CNO,GRADE,

GRADE=CASE

WHEN GRADE IS NULL THEN '未考'

WHEN GRADE<60 THEN '不及格'

WHEN GRADE>=60 AND GRADE<70 THEN '及格'

WHEN GRADE>=70 AND GRADE<90 THEN '良好'

WHEN GRADE>=90 THEN '优秀'

END

FROM SC

（4）使用循环：WHILE语句

WHILE语句用来处理循环。在条件为TRUE的时候，重复执行一条或一个包含多条T-SQL语句的语句块，直到条件表达式为FALSE时退出循环体.

其语法如下：

WHILE <条件表达式>

[BEGIN]

<程序块>

[BREAK]

[CONTINUE]

[程序块]

[END]

说明：CONTINUE 命令可以让程序跳过CONTINUE 命令之后的语句，回到WHILE 循环的第一行，继续进行下一次循环。BREAK 命令则让程序完全跳出循环，结束WHILE 命令的执行。WHILE 语句也可以嵌套。

**例** 编程求1到100的和。

declare @i int

declare @sum int

set @i=1

set @sum =0

while @i<=100

begin

set @sum =@sum+@i

set @i=@i+1

end

select @sum as 合计 ,@i as 循环数 图9.1

运行结果如图9.1所示。

请读下列程序并回答下列程序的功能。

DECLARE @i INT

SET @i = 1

WHILE (@i < 11)

BEGIN

IF(@i < 5)

BEGIN

SET @i = @i + 1

CONTINUE

END

PRINT @i

SET @i = @i + 1

END

（5）使用GOTO 语句

GOTO：可以将执行流程改变到由标签指定的位置。系统跳过GOTO关键字之后的语句，并在GOTO 语句中指定的标签处继续执行操作。

语法：

GOTO 标识符

求1＋2＋3＋…＋10的总和。

DECLARE @S SMALLINT,@I SMALLINT

SET @I=1

SET @S=0

BEG:

IF (@I<=10)

BEGIN

SET @S=@S+@I

 SET @I=@I+1

GOTO BEG

END

PRINT @S

运行结果如图9.2所示。 图9.2

（6）调度执行：WAIT FOR

该语句可以指定它以后的语句在某个时间间隔之后执行，或未来的某一时间执行。

语法如下：

WAIT FOR{DELAY 'time'|TIME 'time'}

参数含义：

DELAY 'time'是指定SQL Server 等待的时间间隔，最长可达24小时。

TIME 'time'是指定SQL Server等待到某一时刻。

**例** 若变量“@等待”的值等于“间隔”，查询Studnet表是在等待2分钟后执行，否则在下午2:10执行。

DECLARE @等待 CHAR(10)

SET @等待= '间隔'

IF @等待= '间隔'

BEGIN

WAITFOR DELAY '00:02:00'

SELECT \* FROM STUDENT

END

ELSE

BEGIN

WAITFOR TIME '14:10:00'

SELECT \* FROM STUDENT

END

### 实验十 SQL SERVER事务设计（选做）

实验时间：

### 实验目的：

（1）通过实验理解事务的概念、特性，掌握事务的设计思想和事务创建、执行的方法；

（2）掌握事务的提交COMMIT；

（3）掌握事务的回滚ROLLBACK；

（4）了解事务的锁。

### 实验内容：

创建一个名为“仓库职工”的数据库，导入以下的四张表到该数据库中，完成以下各题：

仓库表

|  |  |  |
| --- | --- | --- |
| 仓库号 | 城市 | 面积 |
| WH1 | 北京 | 370 |
| WH2 | 上海 | 500 |
| WH3 | 广州 | 200 |
| WH4 | 武汉 | 400 |

职

职工表

|  |  |  |
| --- | --- | --- |
| 仓库号 | 职工号 | 工资 |
| WH2 | E1 | 1220 |
| WH1 | E2 | 1210 |
| WH2 | E3 | 1250 |
| WH3 | E4 | 1230 |
| WH1 | E5 | 1250 |
| WH3 | E6 | 2000 |
| WH1 | E7 | 2080 |

订购表

|  |  |  |  |
| --- | --- | --- | --- |
| 职工号 | 供应商号 | 订购单号 | 订购日期 |
| E3 | S7 | OR67 | 2009-12-4 |
| E1 | S4 | OR73 | 2009-4-1 |
| E7 | S4 | OR76 | 2009-4-2 |
| E6 | S6 | OR77 | 2009-1-21 |
| E3 | S4 | OR79 | 2009-11-15 |
| E1 | S6 | OR80 | 2009-2-1 |
| E3 | S6 | OR90 | 2009-3-12 |
| E3 | S3 | OR91 | 2009-3-2 |

供应商表

|  |  |  |
| --- | --- | --- |
| 供应商号 | 供应商名 | 地址 |
| S3 | 振华电子厂 | 西安 |
| S4 | 华通电子公司 | 北京 |
| S6 | 607厂 | 郑州 |
| S7 | 爱华电子厂 | 北京 |
| S8 | 胖熊公司 | 广州 |
| S9 | 巧姑娘日化 | 北京 |

1. **创建事务,并执行。功能为实现广州的职工加10%的工资.**

Begin transaction

Use cangku

Go

Update 职工 set 工资=工资\*1.1

From 职工,仓库

Where 职工.仓库号=仓库.仓库号 and 城市=’广州’

Go

Commit

Go

（2）在select 语句中加锁，悲观锁定职工表。

Begin transaction

Select \* From职工表with(xlock)

Insert into 职工表(仓库号,职工号) values(‘WH2’,’E10’)

Update 职工表 set 工资=2980

Where 仓库号=‘WH2’ and 职工号=‘E10’

Commit transaction

（3）使用HOLDLOCK对供应商表加共享锁。

Begin transaction t2

Select 供应商号 From 供应商表with(holdlock)

Select count(供应商号) From 供应商表

Commit

**思考题：**

1.什么是事务？事务和程序是一个概念吗？

2.解释语句rollback和commit的作用。

3.为何使用事务？

4.乐观锁和悲观锁的优缺点各是什么？

### 实验十一 SQL SERVER的存储过程（选做）

实验时间：

### 实验目的：

（1）掌握存储过程的定义；

（2）掌握存储过程的有点；

（3）掌握定义带输入与输出参数的存储过程；

（4）调用存储过程。

（5）删除存储过程

### 实验内容：

注意：这次实验的示例数据库为学生选课数据库，请运行例题中的代码。

存储过程（procedure）类似于C语言中的函数，JAVA的方法。它可以重复调用。当存储过程执行一次后，可以将语句缓存中，这样下次执行的时候直接使用缓存中的语句。这样就可以提高存储过程的性能。

存储过程是一组编译在单个执行计划中的Transact-SQL语句，将一些固定的操作集中起来交给SQL Server数据库服务器完成，以实现某个任务。

存储过程的优点：

（1）与其他应用程序共享应用程序逻辑，因而确保了数据访问和修改的一致性。

（2）防止数据库中表的细节暴露给用户。

（3）提供安全机制。

（4）改进性能。

（5）减少网络流量。

1. **存储过程的分类**

（1）用户定义的存储过程

用户定义的Transact-SQL存储过程中包含一组Transact-SQL 语句集合，可以接受和返回用户提供的参数。

（2）扩展存储过程

扩展存储过程是指 Microsoft SQL Server 的实例可以动态加载和运行的 DLL，是由用户使用编程语言（例如C）创建的自己的外部例程，扩展存储过程一般使用sp\_或xp\_前缀。

（3）系统存储过程

由系统提供的存储过程，可以作为命令执行各种操作。系统存储过程定义在系统数据库master中，其前缀是sp\_，例如常用的显示系统信息的sp\_help存储过程。

1. **如何创建存储过程**

创建简单的存储过程的语法：

CREATE PROC[EDURE] 存储过程名

[WITH ENCRYPTION]

[WITH RECOMPILE]

AS

SQL语句

说明：[WITH ENCRYPTION]--对存储过程进行加密，加密的存储过程用sp\_helptext查看不到存储过程的原码；[WITH RECOMPILE]--对存储过程重新编译。

执行存储过程的语法：

EXEC 存储过程名

**例** 创建一个名为GetInfo存储过程，用于获取所有学生信息。

CREATE PROCEDURE GetInfo

AS

SELECT \* FROM Student

执行存储过程：

EXEC GetInfo

接下来学习编写带参数的存储过程。上例中的存储过程可以获取所有学生信息，如果要获取指定学生的信息怎么做？这里就需要创建带参数的存储过程。

存储过程的参数分两种：输入参数和输出参数。输入参数用于向存储过程传入值，类似C语言的按值传递；输出参数用于在调用存储过程后返回结果，类似C语言的按引用传递；

带参数的存储过程的语法：

CREATE PROC[EDURE] 存储过程名

@参数1 数据类型 = 默认值[OUTPUT],

**…,**

@参数n 数据类型 = 默认值 [OUTPUT]

AS

SQL语句

**例** 创建一个带输入参数的存储过程，要求用于获取指定学生的信息。

CREATE PROCEDURE StuInfo

@name CHAR(10)

AS

SELECT \* FROM STUDENT WHERE SNAME=@name

执行存储过程

EXEC StuInfo @name='李晨'

或按位置传递参数值

EXEC StuInfo '李晨'

执行完毕的结果如图11.1所示：

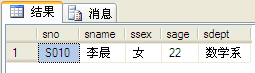


图11.1执行存储过程结果

**例** 创建一个带输入和输出参数的存储过程GetScore，获取指定课程的平均成绩、最高成绩、最低成绩，并返回结果。

CREATE PROCEDURE GetScore

@kcID CHAR(10),@AVGScore INT OUTPUT,

@MAXScore INT OUTPUT,@MINScore INT OUTPUT

AS

SELECT @AVGScore=AVG(Grade),@MAXScore=MAX(Grade),@MINScore=MIN(Grade)

FROM SC

WHERE Cno=@kcID

SELECT @AVGScore as 平均成绩,@MAXScore as 最高成绩,@MINScore as 最低成绩

执行存储过程

DECLARE @kcID CHAR(10),@AVGScore INT,@MAXScore INT,@MINScore INT

SET @kcID='C001'

EXEC GetScore @kcID,@AVGScore,@MAXScore,@MINScore

执行完毕的结果如图8.20所示：

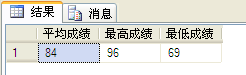


图11.2 执行存储过程结果

1. **查看存储过程**

在SQL Server中，根据不同需要，可以使用sp\_helptext、sp\_help、sp\_depends系统存储过程来查看用户自定义函数的不同信息。

**例29.** 查看Students数据库中存储过程GetInfo信息。

代码如下：

EXEC sp\_helptext GetInfo

EXEC sp\_help GetInfo

EXEC sp\_depends GetInfo

运行后得到存储过程的定义、参数和依赖信息。

1. **存储过程的修改**

修改存储过程是由ALTER语句来完成的，其语法如下：

ALTER PROCEDURE procedure\_name

[WITH ENCRYPTION]

[WITH RECOMPILE]

AS

Sql\_statement

**例** 修改存储过程StuInfo，根据用户提供的系名进行统计这个系的人数，并要求加密。

ALTER PROCEDURE StuInfo

@dept CHAR(10),

@num INT OUTPUT

WITH ENCRYPTION

AS

SELECT @num=COUNT(\*) FROM Student WHERE SDEPT=@dept

PRINT @num

执行存储过程

DECLARE @dept CHAR(10),@num INT

SET @dept='CS'

EXEC StuInfo @dept,@num

1. **存储过程的删除**

存储过程的删除是通过DROP语句来实现的。

**例31.** 使用Transact-SQL语句来删除存储过程StuInfo。

DROP PROCEDURE StuInfo