

# 数据库课程实验操作指导

(修订版)

华中科技大学计算机科学与技术学院

数据库系统课程教学组

二〇〇七年三月

# 目 录

## 一 . 概述

- 1 . SQL Server 2000 组成
- 2 . SQL Server 2000 的安装
- 3 . SQL Server 2000 的工具

## 二 . DDL 的使用方法

- 1 . 数据库建立
- 2 . 基本表的建立
- 3 . 视图的建立

## 三 . DML 的使用方法

- 1 . INSERT 命令
- 2 . DELETE 命令
- 3 . UPDATE 命令

## 四 . SELECT 命令 使用方法

## 五 . DCL 的使用方法

- 1 . SQL Server 登录管理
- 2 . 用户管理
- 3 . 授权用户 ( GRANT、REVOKE )

## 六 . 游标的使用

- 1 . 游标的定义
- 2 . 游标的操作

## 七 . 数据库的备份和恢复

## 八 . 实验练习

- 1 . 基本表的创建、数据插入
- 2 . 数据查询
- 3 . 数据修改、删除
- 4 . 视图的操作
- 5 . 库函数 , 授权控制
- 6 . 数据库备份、恢复

## 九 . 数据库课程设计基本要求

# 一. 概述

## 1. SQL Server 2000 组成

SQL Server 是可缩放的高性能基于 SQL 和客户/服务器体系结构的关系数据库管理系统服务器软件包，是由 Microsoft 公司推出的 SQL Server 数据库管理系统的最新版本。。从图 1 SQL Server 的体系结构示意图中看出，SQL Server 2000 由 4 部分组成，在实验中，我们要掌握基于 SQL Server 2000 的服务器的使用，也就是数据库管理员 DBA 的主要操作。

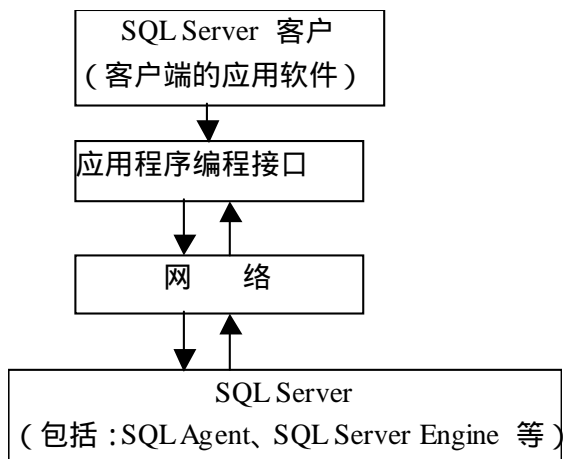


图 1 SQL Server 的体系结构示意图。

(注：本文所有内容均在 SQL Server 2000 上实现，读者也可在 SQL Server 2005 上得到类似结果。)

## 2. SQL Server 2000 的安装

SQL Server 2000 的常见版本有：企业版、标准版、个人版、开发人员版等。对硬件的最低需求为：CPU Pentium 166MHz，内存 64MB，硬盘 180 MB。SQL Server 2000 企业版和标准版只能在 windows2000 Server 版和 Professional 版操作系统下运行。下面介绍 SQL Server 2000 企业版在本地机上的安装过程。

1) 插入 SQL Server 2000 光盘，自动安装程序启动，屏幕上出现如图 2 所示画面，按图 2 所示**选择 SQL Server 2000 组件**；

2) 选择**安装数据库服务器**，如图 3 所示，进入 SQL Server2000 企业版安装向导；



图 2 SQL Server 2000 安装过程对话框



图 3 选择 SQL Server 2000 安装组件对话框

- 3) 在安装向导对话框中点击下一步，进入计算机名对话框；
- 4) 选择本地安装，点击下一步，进入安装选择对话框；
- 5) 选择创建新的 SQL Server 实例，点击下一步，进入用户信息对话框；
- 6) 输入用户信息，点击下一步，进入安装定义对话框；

7) 选择服务器和客户端工具，点击下一步，进入实例名对话框；

8) 输入实例名，点击下一步，进入安装类型选择对话框；

9) 选择典型安装，进入服务帐号设置对话框，如图 4 所示；

10) 选择对每一个用户使用同一个帐号，自动启动服务器，点击下一步，进入选择身份验证模式选择对话框，如图 5 所示；

11) 选择 Windows 身份验证模式。点击下一步，进入开始复制文件对话框，点击下一步，进入选择许可模式对话框，选择处理器许可证，点击继续，开始复制文件。

12) 文件复制完毕后，进入安装完毕对话框，点击完成。系统安装完毕。



图 4 选择服务帐号对话框



图 5 身份验证模式对话框

### 3. SQL Server 2000 的工具

#### 1) 服务管理器 (Server Manager)

在进行任何数据库操作前，都必须启动服务器，Server Manager 可以方便启动、停止、暂停本地或远程服务器。图 6 是打开的 Server Manager 窗口。适当配置窗口中的选项，点击“启动”按钮即可启动 SQL Server 服务器。SQL Server 2000 安装以后，每次开机时，Windows 都会自动启动服务管理器。在任务栏上有一个图标表示服务管理器的状态。如图 7 所示。



图 6 Server Manager 窗口



图 7 服务管理器在任务栏中的状态

#### 2) SQL Server 2000 企业管理器 (Enterprise Manager)

- 企业管理器的启动

进入 SQL Server2000 的企业管理器 从 **Program -> SQL Server -> Enterprise Manager**, 图 8 为企业管理器界面。企业管理器是 DBA 管理数据库的重要工具, 使用 Enterprise Manager 可以管理用户账号、登录服务器、备份和恢复数据库、启动和停止 SQL Server、创建和管理数据库对象和任务、处理访问控制列表、检查数据库的一致性。其中数据库 Master、Model、Msdb、Tempdb 是系统数据库。Pubs、Northwind 是 SQL Server 自带的样本数据库, 其中有大量的 Test 数据, 可以作为 SQL 语言实验的基础数据。

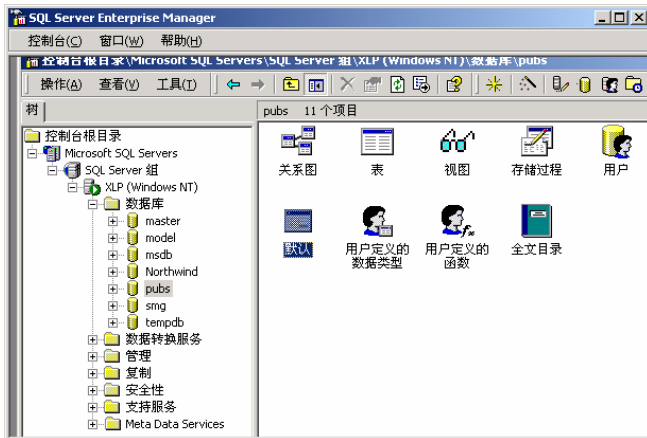


图 8 企业管理器

Tempdb 是系统数据库。Pubs、Northwind 是 SQL Server 自带的样本数据库, 其中有大量的 Test 数据, 可以作为 SQL 语言实验的基础数据。

### ● 数据库文件夹

数据库文件夹中显示 SQL Server 服务器管理的所有数据库信息。例如 SQL Server 提供的样本数据库 Pubs 的展开如图 9 所示。每个数据库均由这些项目来分别描述数据库的属性。其中关系图为 SQL Server 根据用户对数据库的定义, 自动绘制的数据库关系示意图 (E - R 图)。图 10 为 Pubs 的关系图, 表示出基本表之间的引用关系, 参照完整性定义的图示化表示。

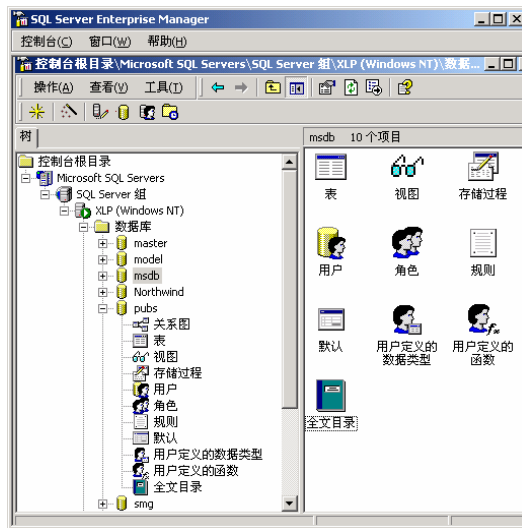


图 9 企业管理器中的数据库对象

表为数据库中包含的表的信息。表分为两类, 一类为用户创建的表, 另一类为系统创建的表。用户表由用户使用 DDL 语言创建, 系统表存放 DBMS 对用户数据库的管理信息。其它对象包含了与当前数据库有关的信息。

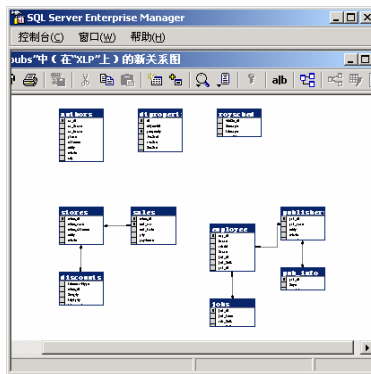


图 10 Pubs 的关系图

### 3) SQL Server 2000 查询分析器 (Query Analyzer)

查询分析器是一个重要工具, 实验中的所有 SQL 语言命令均需在查询分析器中输入、编辑运行。

从 Program -> SQL Server -> Query Analyzer 可以打开查询分析器, 如图 11 所示。它是一个多文档程序, 在其上可以同时打开多个查询程序 (窗口), 在查询分析器的工具栏中的绿色 ▶ 按钮为执行当前窗口中 SQL 语句按钮。点击它即可逐行执行。

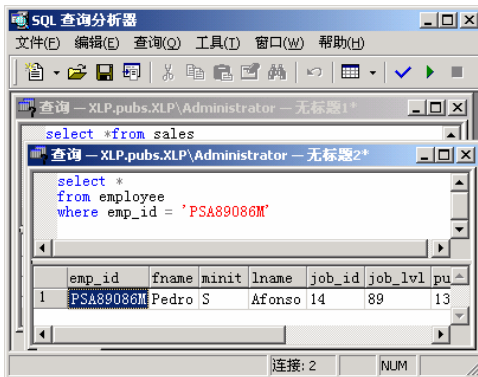


图 11 查询分析器

## 二. DDL 使用方法

Transact-SQL 是 SQL Server 内置的 SQL 语言, 它支持标准 SQL 语言, 但在许多方面进行了扩充, 其定义能力更为强大。Transact-SQL 对使用 Microsoft® SQL Server™ 非常重要。与 SQL Server 通讯的所有应用程序都通过向服务器发送 Transact-SQL 语句来进行通讯, 而与应用程序的用户界面无关。在此我们不对 Transact-SQL 的语法作详细介绍, 大家可以参考查询分析器中提供的帮助。

### 1. 数据库创建

在查询分析器中执行下列语句即可在默认的设备上创建新的数据库 ems。

```
CREATE DATABASE database_name
```

例 1: 创建名为 ems 的数据库: CREATE DATABASE ems

这是一个简单的人事管理数据库。本例中的所用数据库对象均为 ems 建立。执行结果如图 12 所示。

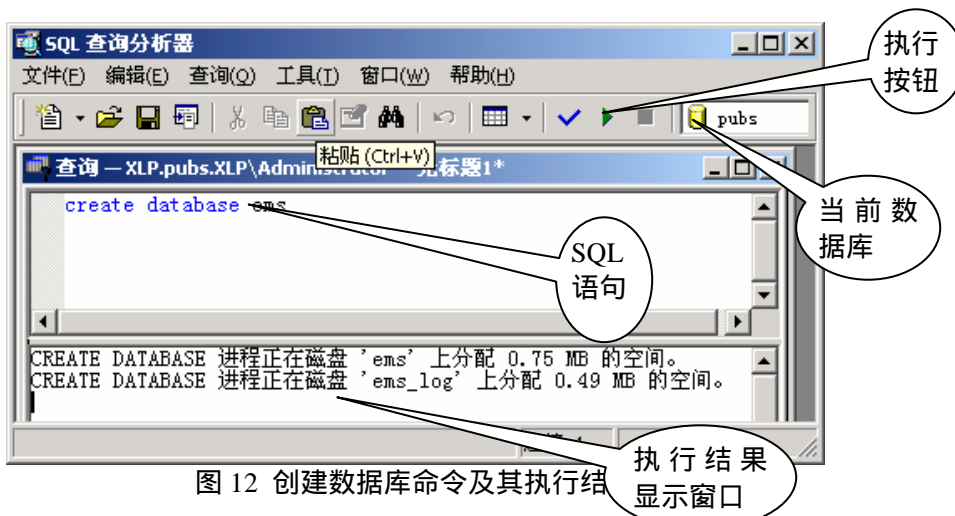


图 12 创建数据库命令及其执行结果

## 2. 基本表的建立

创建基本表的命令为：CREATE TABLE table\_name，在该命令中定义主码和外码时，可以使用列约束（Column Constraint）或表约束（Table Constraint）子句。

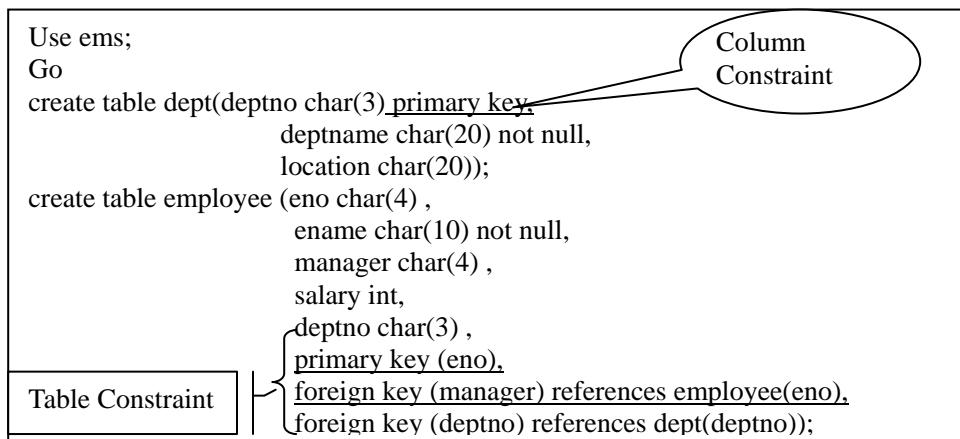
例 2：在 ems 中创建 employee（职员）表和 dept（部门）表：

employee（eno，ename，manager，salary，deptno）；主码：eno 外码：manager、deptno

dept（deptno，dname，location）主码：deptno

注：employee 中的 manager 为相应雇员的直接领导或上司的 eno。

创建基本表时，应先选择包含表的数据库，本例中选择当前数据库为 ems。查询分析器启动时的当前数据库为 pubs，如图 12 所示。改变当前数据库可以直接点击当前数据库列表框，选中所需数据库，也可使用 USE databasename 命令。图 13 为使用 SQL 建表命令创建上述基本表的源代码。在查询分析器中输入该代码，点击执行按钮即可。



```
Use ems;
Go
create table dept(deptno char(3) primary key,
                 deptname char(20) not null,
                 location char(20));
create table employee (eno char(4),
                      ename char(10) not null,
                      manager char(4),
                      salary int,
                      deptno char(3),
                      primary key (eno),
                      foreign key (manager) references employee(eno),
                      foreign key (deptno) references dept(deptno));
```

The image shows a screenshot of SQL code with two callouts. One callout, labeled 'Column Constraint', points to the 'primary key' text in the dept table definition. Another callout, labeled 'Table Constraint', points to the 'foreign key (manager) references employee(eno)' and 'foreign key (deptno) references dept(deptno)' lines in the employee table definition.

图 13 SQL 建表命令创建上述基本表的源代码

## 3. 视图的建立

视图是组成数据库体系结构——三级模式两级映像结构中的外模式的基本单元，SQL-Server 的视图定义命令为：CREATE VIEW view-name AS SELECT statement

视图是用于定义终端用户数据来源的。在视图定义中可以使用复杂的 SELECT 命令。

例 3：在 ems 中定义能够查询雇员年薪的视图 Annualsal 和统计雇员下属人数的视图 manager。视图定义为：

Annualsal（eno，ename，annualsal）；annualsal 为相应雇员的年薪。

manager（manager，name，clerknum）；clerknum 为相应雇员的直接下属个数。

在查询分析器中分别输入如图 14 所示的代码，即可创建要求的视图。

```
Use ems ;
create view annualsal(eno,ename,annualsal)
as select eno,ename,12*salary
from employee
```

```
Use ems;
create view manager (manager,name,clerknum)
as select e1.manager,e2.ename,e1.clerknum
from (select manager , count(*) as clerknum
from employee
group by manager
having count(*)>0) e1,employee e2
where e1.manager = e2.eno
```

图 14 : 视图定义源代码 其中: 在视图 manager 定义中的 e1 为联机视图。

### 三 . DML 使用方法

SQL 的 DML 包括插入 (INSERT)、删除(DELETE)、修改(UPDATE)等命令。DML 命令的执行是可能造成数据库不一致的根源。因此, 每一条语句在执行前, SQL-Server 都要验证语句是否符合完整性要求, 包括实体完整性、参照完整性、用户定义完整性。

#### 1 . INSERT 命令

SQL 语言的插入命令:

INSERT INTO table-name (column-list) VALUES(values-list), 可以完成数据输入功能。

例 4: 在 ems 中的 dept 表中输入表 1 中的数据; employee 表中输入表 2 中的数据。

图 15 为数据输入的源代码。

表 1 基本表 dept 的数据

D01	Computer School	South1-405
D02	Communication Dept	South1-304
D03	Management School	kejilou-408

表 2 基本表 employee 的数据

E01	LU	null	8000	D01
E02	YIN	E01	7000	D01
E03	XU	E01	2000	D01
E04	QU	E02	2000	D01
E05	ZAO	null	5000	D02
E06	PANA	E05	4000	D02
E07	PANB	E05	3000	D03

```
use ems;
go
insert into dept values('D01','Computer School','North1-405');
insert into dept values('D02','Communication Dept','Notth1-304');
insert into dept values('D03','Management School','kejilou-408');
insert into employee values('E01','LU',null,8000,'D01');
insert into employee values('E02','YIN','E01',7000,'D01');
insert into employee values('E03','XU','E01',2000,'D01');
insert into employee values('E04','QU','E02',2000,'D01');
insert into employee values('E05','ZAO ',null,5000,'D02');
insert into employee values('E06','PANA ','E05',4000,'D02');
insert into employee values('E07','PANB ','E05',3000,'D03');
```

图 15 在基本表中插入数据的源代码



注意：在执行过程中，如果有一条语句出错误，再次执行时，在出错语句前面的语句就会被拒绝执行。因为，在查询分析器中的命令在执行时按顺序逐条执行，当执行到出错语句时停止执行，此时，在出错语句的前面各条语句的结果已经存入表中，故当重复执行时，它们违反了实体完整性。

例 5：插入可能违反参照完整性的插入命令，验证实体完整性检验规则。

在查询分析器中执行如图 16 所示的命令，系统将拒绝执行。执行结果如图 17 所示。

```
use ems;
go
insert into employee values('E10','WU','E11',8000,'D01');
insert into employee values('E02','YIN','E01',7000,'D05');
```

图 16 可能被拒绝的命令

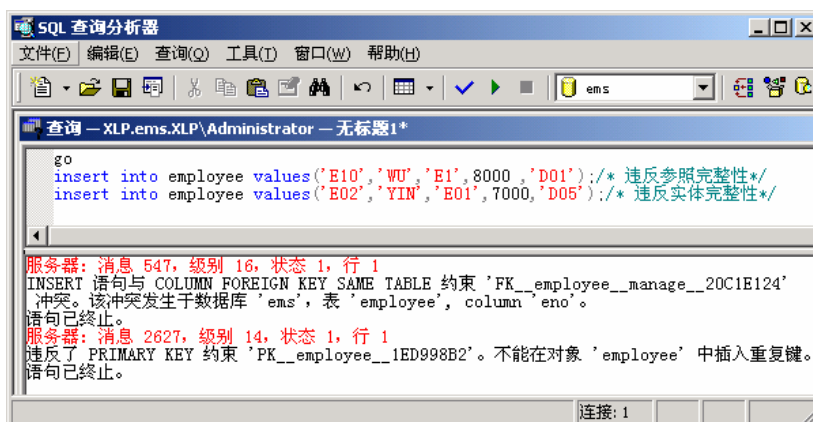


图 17 执行被拒绝命令的结果显示图

## 2 . DELETE 命令

SQL 的删除命令为：DELETE FROM table-name [WHERE condition-expression];

执行 DELETE 命令后，系统会删除满足命令中条件表达式的所有元组。这种删除只是逻辑的。因此，当再次插入一个与被删除的元组具有相同关键字的元组时，被认为违反了实体完整性。

例 6：删除姓名= ' PANB ' 的纪录。在查询分析器中输入并执行如图 18 所示的代码即可完成要求的删除操作。

删除操作可能会引起参照完整性的破坏，对于这些操作系统会根据完整性定义执行或拒绝执行。例如：在没有删除 PANB 纪录前，试图删除他所在的部门的操作就会引起系统的警告，并拒绝执行该操作。

```
use ems;
go
delete from employee
where ename='PANB';
```

图 18 删除记录 PANB 的代码

### 3 . UPDATE 命令

SQL 的修改命令为：

UPDATE table-name

SET column-name = expression

[WHERE condition-expression];

执行 UPDATE 命令后，系统会按照 SET 子句修改满足命令中条件表达式的所有元组。如果使用修改命令更新关键字的值，而该关键字在其他表中作为外码存在时，操作可能违反参照完整性。系统将拒绝执行。

例 7：将表 employee 中所有人的 salary 增加 10%。

```
UPDATE employee
SET salary = salary*1.1;
```

例 8：将 ENO='E01' 的 ENO 更新为'E00'。该操作结果如图 19 所示：

```
UPDATE employee
SET eno = 'E00'
WHERE eno='E01';
```

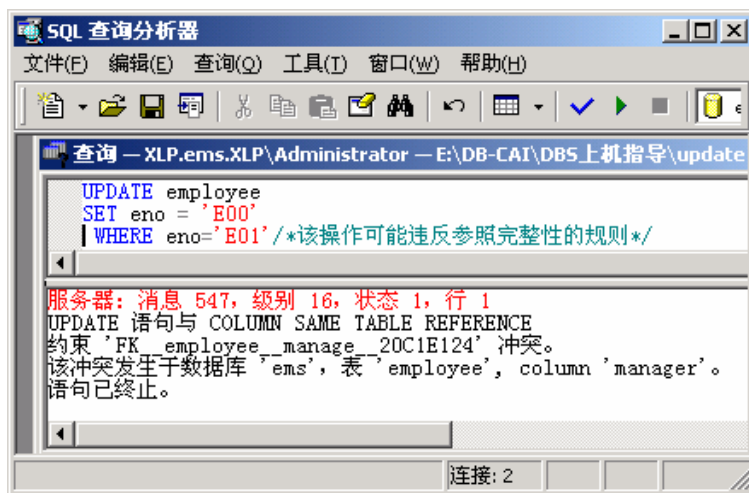


图 19 例 8 的执行结果

## 四 . SELECT 命令

SELECT 命令是 SQL 语言中使用最频繁的命令 ,其变化形式繁多 ,灵活运用 SELECT 命令可以完成任意复杂的查询要求。SELECT 命令的基本语法为:

```
SELECT query_expression|column-list|*  
FROM table_name_list|view_name_list  
[WHERE condition-expression]  
[GROUP BY [HAVING condition-expression]]  
[ORDER BY order_expression| column [ASC|DESC]][,...n];
```

由于 SELECT 命令非常复杂,上面的语法描述还无法完全表达,请参考 Transaction SQL 的帮助。从海量数据库中得到自己所希望看到的信息,是学习 SQL 语言的重要任务。下面只举几个典型例子。

### 1 . 简单查询

例 9 : 列出 ems 中的所有员工的信息。

```
SELECT * FROM employee;
```

### 2 . 使用谓词的查询

谓词有 IN、LIKE、NULL、EXISTS、BETWEEN 等,在查询时 WHERE 子句中可以使用。

例 10 : 列出工资在 2000-3000 之间的所有员工的姓名及他们所在的部门号码,将结果按工资从高到低的顺序排列。在查询分析器中输入并执行图 20 中的代码即可得到所要的结果。

```
SELECT ename,salary,deptno  
FROM employee  
WHERE salary BETWEEN 2000 AND 3000  
ORDER BY salary DESC;
```

图 20 例 10 的检索命令

### 3 . 连接查询

连接查询通过表间的关联字段实现相关查询,连接查询分为等值联接查询、不等值连接查询、外联接查询、自我连接查询等。在书写较为复杂的查询命令时,建议使用别名来提高效率。

例 11 : 查询雇员 PANA 所在的部门及其办公地点 ;

```
SELECT e.deptno,d.location  
FROM employee e,dept d  
WHERE e.deptno = d.deptno AND e.ename = 'PANA';/*This is a Equijoin*/
```

图 21 例 11 的连接查询代码

例 12：列数办公地点在 South1 的所有雇员名单；

```
SELECT eno,ename, e.deptno
FROM employee e, dept d
WHERE e.deptno = d.deptno AND d.location LIKE 'South1%';
```

图 22 例 12 的代码

例 13：列出每个雇员的间接上司（Manager 的 Manager）的姓名。

```
SELECT e1.eno, e1.ename,e3.eno, e3.ename as manager_of_manager
FROM employee e1, employee e2 ,employee e3
WHERE e1.manager = e2.eno and e2.manager = e3.eno
```

图 23 例 13 的代码

#### 4 . 复杂查询

例 14 查询部门 D01 中薪水介于部门 D02 的最高和最低值之间的雇员及他们的薪水。

```
use ems;
SELECT eno,ename,salary
FROM employee
WHERE (salary BETWEEN (SELECT MIN(salary)
                        FROM employee
                        WHERE deptno='D02')
      AND (SELECT MAX(salary)
            FROM employee
            WHERE deptno='D02'))
      AND deptno='D01';
```

图 24 例 14 的代码

## 五 . DCL 的使用方法

### 1 . SQL Server 登录认证

对任何用户来说,数据库的安全性都是至关重要的。由于数据是面向所有合法用户的,因此,对任何 DBMS,良好的用户管理功能是确保数据库安全性的重要保障。SQL Server 2000 的安全性建立在认证和访问许可两种机制上的。认证机制确保登录的用户帐号和密码是正确的,通过认证的用户获得数据库的访问授权后才能对服务器上的数据进行授权许可下的各种操作。在 SQL Server 2000 的安全模式中包括:SQL Server 登录、数据库用户、权限、角色等。SQL Server 2000 有两种安全模式,即 WINDOWS 认证模式和混合模式。WINDOWS 模式只要求用户通过 WINDOWS 操作系统的认证。如果用户登录 SQL Server 2000 时未给出用户的登录名,SQL Server 自动使用该模式,在前面各种操作时均使用该模式。混合认证模式下,可以使用 WINDOWS 认证或 SQL Server 认证。使用 SQL Server 认证时,用户在连接 SQL Server 时必须提供登录名和密码,这些登录信息存放在系统表 syslogins 中,与操作系统无关。

#### 1) SQL Server 认证模式的设置

在对登录进行增加、删除操作前,必须设置 SQL Server 认证模式。设置步骤:启动**企业管理器**→选择要进行认证模式设置的服务器→右击该服务器,在弹出菜单中选择**属性**,弹出如图 25 所示的对话框,点击安全性选择栏,选择身份验证为 *SQL Server 和 Windows (S)* 选项。任选一种**审核级别**,在**启动服务器帐户**选项中选择**系统帐户**。点击**确定**。确认设置。

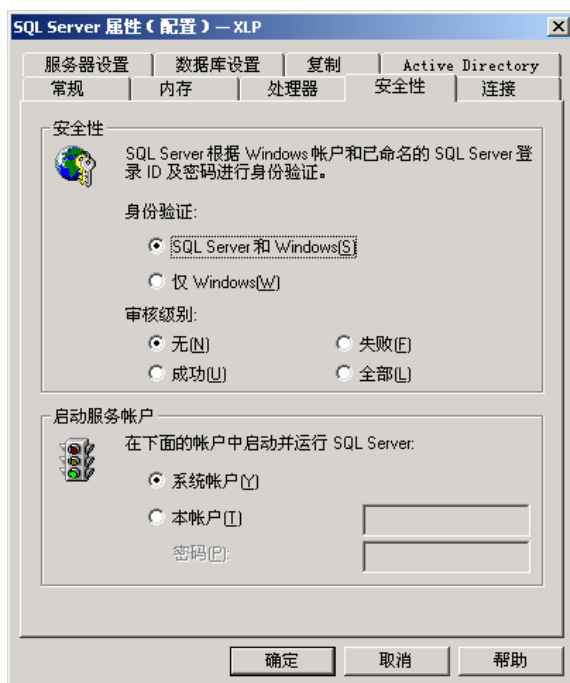


图 25 SQL Server 认证模式的设置

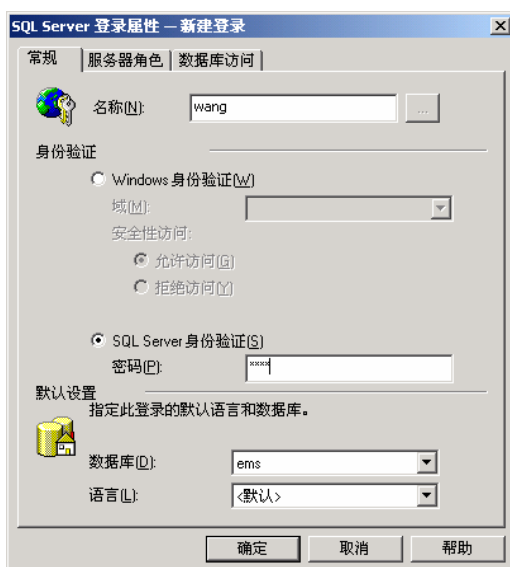


图 26 新建登陆对话框

## 2) 企业管理器管理登录

在 SQL Server 认证模式下可以新建登录和管理登录，其操作步骤如下：启动企业管理器→展开控制台根目录，选择安全性并展开→右击登录，在弹出菜单中选择新建登录，弹出如图 26 所示的对话框，在常规选择栏中，输入登录用户的信息，身份验证选择 SQL Server 验证，并输入密码。选择该登录的数据库，确认后点击确定按钮，新的登录即创建成功。

## 3) Transact\_SQL 存储过程管理登录

在 SQL Server 中，有一些存储过程提供了管理 SQL Server 登录的功能，使用他们可以在查询分析器中，用命令的方式管理登录。包括：sp\_grantlogin sp\_revokelogin sp\_decnylogin sp\_addlogin sp\_droplogin sp\_helplogins。

- 新登录的创建

新的登录者可以是 WINDOWS 的用户或用户组，也可以是使用 SQL Server 认证模式的登录帐号，设定 WINDOWS 的用户为 SQL Server 登录者时使用 sp\_grantlogin。创建新的使用 SQL Server 认证模式的登录帐号时使用 sp\_addlogins。图 27 是使用 sp\_addlogin 创建使用 SQL Server 认证模式的登录‘qbb’的代码，在具有权限的 sa 登录连接的查询分析器中输入以下代码即可创建一个 ems 的登录 qbb 用户，该登录的密码为‘qbb’。

```
use ems
go
EXEC sp_addlogin qbb,qbb
```

图 27 创建登陆 qbb 的代码

图 27 是使用 sp\_addlogin 创建使用 SQL Server 认证模式的登录‘qbb’的代码，在具有权限的 sa 登录连接的查询分析器中输入以下代码即可创建一个 ems 的登录 qbb 用户，该登录的密码为‘qbb’。

- 登录的删除

删除基于使用 SQL Server 认证模式的 SQL Server 的登录可以使用 sp\_droplogin。但是，如果与该登录匹配的数据库用户仍存在 sysuser 表中，则不能删除该登录账号。删除 WINDOWS 的用户或用户组时，使用 sp\_revokelogin，拒绝 WINDOWS 的用户或用户组连接到 SQL Server 时，使用 sp\_decnylogin。图 28 为删除了新建登录 bbb 的代码。

```
use ems
go
EXEC sp_droplogin qbb
```

图 28 删除登陆 qbb 的代码

- 登陆的查看

sp\_helplogins 可以用来显示 SQL Server 的所有登录者的信息。如图 29 是显示数据库 ems 的所有登录者信息的代码。

```
use ems
go
EXEC sp_helplogins
```

图 29 查看登陆的代码

## 2. 用户管理

### 1) 用户的概念

在 SQL Server 中，数据库用户帐号与登录帐号是两个不同的概念，一个合法的登录

帐号，只表明该帐号通过了认证，但不表明其可以对数据库进行操作，所以，一个登录帐号总是和一个或多个数据库用户帐号相对应，这样才可以访问数据库。在系统初始安装时，系统自动与默认的用户相关联，所以我们在前面的操作才能进行。例如，登录帐号 sa 自动与每一个用户 dbo 相关联。这就是为什么我们先前建立的数据库的所有者均为 dbo 的原因。通常数据库帐号总是与某一个登录帐号相关联。但 guest 例外，在安装系统时，guest 用户被加入到 master、pubs、tempdb、和 northwind 中。这使初次使用 SQL Server 的用户能够访问到 SQL Server 自带的数据库，并且对于没有建立安全性防范机制的数据库，均可以使用该用户来访问。

## 2) 企业管理器管理数据库用户

### • 创建新数据库用户

启动企业管理器 → 展开控制台根目录 ; 打开数据库文件夹 → 选择要创建用户的数据库 ( 如本例中的 ems ) , 右击用户, 在弹出菜单中选择 **新建数据库用户**, 弹出如图 30 所示的对话框, 在常规选择栏中, 输入登录名和用户名。在数据库角色成员列表框中选择 public, 确认后点击确定按钮, 新的数据库用户即创建成功。



图 30 新建用户对话框

### • 查看、删除数据库用户

在 SQL Server 企业管理器中, 选择用户图标, 则在右边的窗格中显示当前数据库的所用用户, 选择关心的用户, 点击右键即可查看用户信息或删除该用户。

## 3) Transact\_SQL 存储过程管理数据库用户

SQL Server 利用下列系统过程管理数据库用户：

sp\_adduser sp\_grangdbaccess sp\_dropuser Sp\_dropuser sp\_revokedbaccess sp\_helpuser 。其中 sp\_adduser Sp\_dropuser 是为了保持与以前版本相兼容，因此不主张使用，在 SQL Server 2000 中，建议使用 sp\_grangdbaccess sp\_revokedbaccess 。

### • 创建新数据库用户

所有的数据库用户（除了 guest）均必须与某一登录账号相匹配，因此在使用系统过程创建新数据库用户时，不但要指出新数据库用户的名称，还必须指出一个已经存在的登录账号。系统过程 sp\_grantdbaccess 用于为 SQL Server 登录用户或 NT 用户或用户组建立相匹配的数据库用户帐号。图 31 是为数据库 ems 的登录 qbb 创建了一个名称为 qbb 的数据库用户。

```
use ems
go
EXEC sp_grantdbaccess 'qbb','qbb'
```




图 31 创建新用户的代码

- 删除数据库用户

系统过程 sp\_revokedbaccess 用于将数据库用户从当前数据库中删除，删除后与其匹配的登录即无法再访问该数据库了。图 32 所示的代码删除了数据库 ems 的用户 qbb。

```
use ems
go
EXEC sp_revokedbaccess 'qbb'
```

图 32 删除数据库用户的代码

- 查看数据库用户信息

sp\_helpuser 可用来显示当前数据库的指定用户信息。图 33 为显示数据库 ems 用户的代码。在查询分析器中输入并执行即可列出 ems 的所有用户信息。

```
use ems
go
EXEC sp_helpuser
```

图 33 显示数据库用户的代码

### 3. 用户授权管理

对于登录到 SQL Server 的合法数据库用户，必须获得对数据库操作的授权。在 SQL Server 中包括两种类型的权限，即对象权限和语句权限。

#### 1) 对象权限

对象权限总是针对表、视图、存储过程而言，它决定了能对这些对象执行哪些操作（如 UPDATE DELETE INSERT SELECT EXECUTE）。不同类型的对象支持不同类型的操作，表 3 为各种对象的可能操作列举。



表 3 :对象权限总结表

对 象	操 作
表	SELECT UPDATE DELETE INSERT REFERENCE
视图	SELECT UPDATE DELETE INSERT
存储过程	EXECUTE
列	UPDATE SELECT

例 15 : 为 ems 的用户 ‘ qbb ’ 授予表 employee 查询权限。

以系统分析员身份进入查询分析器，输入并执行如图 34 的代码。然后，在打开查询分析器中文件下拉菜单，点击连接选项，在弹出菜单(如图 35)输入 qbb 的信息，新打开的窗口即为 qbb 用户的窗口，在该窗口中输入并执行 SELECT \* FROM employee 即可得到查询结果，此时若输入 SELECT \* FROM dept,可以发现语句被拒绝执行。因为‘qbb’没有得 dept 表得操作权限。

同样，可以使用 REVOKE SELECT ON employee FRPM qbb 命令可以收回 qbb 对 employee 的操作权限。大家可以参考教材中命令，对自己的数据库的安全性进行全面管理。

## 2) 语句权限

语句权限指数据库用户执行某种语句的操作权，如创建数据库、表、存储过程等。这些语句虽然(如 CREATE 命令)包含有操作对象，但这些对象在操作前并不存在于数据库中。表 4 为所有权语句清单。

表 4 语句权限总结表

语句	含义
CREATE DTATBASE	创建数据库
CREATE TABLE	创建表
CREATE VIEW	创建视图
CREATE RULE	创建规则
CREATE DEFAULT	创建缺省
CREATE PROCEDURE	创建存储过程
BACKUP DATABASE	备份数据库
BACKUP LOG	创建事务日志

```
Use ems
Go
Grant select on employee to qbb
```

图 34 为用户授权的代码

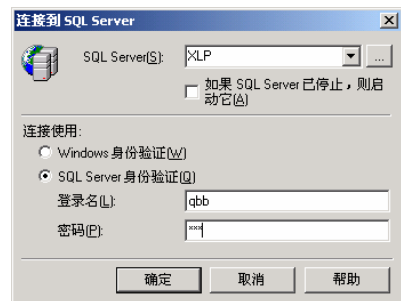


图 35 连接到服务器对话框

例 16 使用语句权限设置命令，授予用户 qbb 在数据库 ems 上建立新表的权限。完成该功能的代码如图 36。

```
Use ems
Go
GRANT CREATE TABLE TO qbb
```

图 36 为用户 qbb 授予创建基表的权限

在 SQL Server 2000 中有角色的概念，角色的设置可以降低权限管理的繁杂性，在大的系统中，虽然有很多数据库用户，但可以将这些用户进行分类，每类定义一种角色，只要对角色进行权限设置，就可以简单管理用户了。角色的设置在此不赘述了。

## 六． 游标的使用

SQL 语言有两种使用方式，即交互式 and 嵌入式。前面我们在查询分析器中执行的命令均使用交互式方式。但是在应用程序开发时往往需要在其他程序开发语言中使用嵌入式的 SQL 命令处理数据库中的数据，这时程序开发语言称为主语言，SQL 语言称为子语言。由于主语言与 SQL 语言处理数据的方式不同，因为关系数据库的处理结果为集合，而大多数程序开发语言使用记录或线性处理。所以必须在两者之间实现沟通，使集合数据也可以一次一条记录的处理。解决的办法是使用游标。游标是一个很重要的概念，它提供了一种对从表中检索结果进行操作手段。能从包括多条记录的结果集中每次提取一条记录。

### 1． 游标的定义

定义游标的命令为：

```
DECLARE cursor_name [INSENSITIVE] [SCROLL] CURSOR
FOR select_statement
[FOR {READ ONLY|UPDATE[OF column_name [...n]]}]
```

可选项 INSENSITIVE 表明游标的操作对应一个内部临时表，在游标的生命周期内，对基本表的修改操作不影响游标的内容，即游标是静态的。否则游标的内容将动态地与基表保持一致。可选项 SCROLL 表示游标可以使用所有的提取操作，如 FIRST、LAST、PRIOR、RELATIVE、ABSOLUTE 等，否则只能使用 NEXT 提取操作。READ ONLY 表示游标内的数据不允许更新。UPDATE [OF column\_name [...n]] 定义在游标中可以修改的列。

### 2． 游标的操作

- 打开游标

游标在定义以后，存储在系统数据库中，如果从游标中读取数据，则必须打开游标，

打开游标的命令为：OPEN [GLOBAL]cursor\_name

由于游标的作用域仅在其所在的批处理中,当建立游标的批处理文件执行结束后,游标会被自动释放。因此在此打开操作前,要重新定义一次,因前面定义的游标已经被释放。如果要保留游标的作用域,可以在定义时指定其作用域为 GLOBAL。否则为 LOCAL。

- 读取游标中的数据

当游标成功打开后,就可以从游标中逐行读取数据。使用 FETCH 命令。语法为：  
FETCH [[NEXT|PRIOR|FIRST|LAST] FROM cursor\_name [INTO @v\_name[,...n]]  
FETCH 命令执行后,变量 @@FETCH\_STATUS 返回被 FETCH 语句执行后游标的状态。如表 5 所示。

表 5 Fetch 命令执行后 @@FETCH\_STATUS 变量的值与状态

返回值	描述
0	FETCH 语句成功。
-1	FETCH 语句失败或此行不在结果集中。
-2	被提取的行不存在。

- 游标的关闭与释放

在处理完游标中的数据后必须关闭游标释放数据结果。使用 CLOSE 命令关闭游标,但不释放定义的数据机构,释放数据机构应使用 DEALLOCATE 命令。语法分别为：

CLOSE cursor\_name

DEALLOCATE cursor\_name

例 17 定义一个游标,逐行显示 employee 表中的信息。对游标进行操作,最后关闭游标。

在查询分析器中输入并执行图 37 所示的代码即可完成要求。

```

use ems;
go
declare cursor_emp cursor global
for select * from employee
for read only;      /* 定义游标*/
declare @v_eno varchar(40),@v_ename varchar(10),
        @v_manager  varchar(40),@v_salary int,
        @v_deptno varchar(30);      /* 定义变量*/
open cursor_emp;    /* 打开游标*/
print 'S*****S';
fetch next from cursor_emp
        into @v_eno,@v_ename,
        @v_manager,@v_salary,@v_deptno; /*从游标中读取数据*/
while @@fetch_status = 0 /*循环处理从游标中读取的数据*/
begin
/*select @v_eno,@v_ename,@v_manager,@v_salary,@v_deptno; */
print @v_eno + @v_ename + @v_manager;
fetch next from cursor_emp
        into @v_eno,@v_ename,
        @v_manager,@v_salary,@v_deptno;
end
print 'E*****E';
close global cursor_emp;
deallocate global cursor_emp;

```

图 37 例 17 的代码

## 七 . 数据库的备份和恢复

备份是指对 SQL Server 数据库或事务日志进行拷贝，如果数据库因意外而损坏，备份文件可以用来恢复数据库。SQL Server 2000 中有四种备份类型，分别是：数据库备份、事务日志备份、差异备份、文件和文件组备份。在使用是可以根据实际情况选择一种或几种的混合应用。

### 1 . 备份设备

在进行备份以前必须创建备份设备。备份设备是用来存储数据库、事务日志、文件和文件组备份的存储介质。可以是磁盘、磁带或管道。

#### 1) 用企业管理器管理备份设备

- 创建备份设备

启动**企业管理器**，登录到要增加备份设备的服务器，打开**管理文件夹**，右击**备份**，在弹出菜单中选择**新建备份设备**选项，弹出**备份设备属性 - 新设备**对话框，如图 38 所示。在名称栏中输入设备的名称，选择设备类型，可以

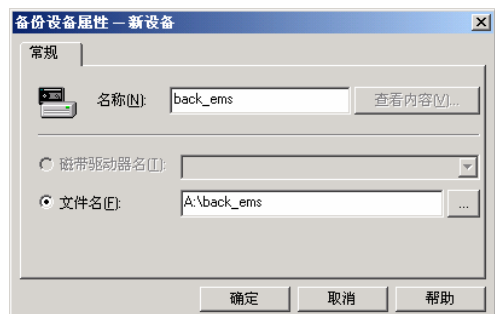


图 38 创建备份设备对话框

选择磁盘、磁带或管道，如果**选择文件名**，则表示使用磁盘做备份，输入文件名后，点击**确定**按钮，新的备份设备即可创建。由于软盘的容量小不适合作为备份设备，通常都选择硬盘，但在本此实习中，由于数据库 ems 的数据很少，也可以将 A：盘作为备份设备。

- 删除备份设备

在创建备份设备的第二步，选中**备份**图标后，在右窗格对话框中右击要删除的备份设备，在弹出菜单中选择**删除**选项，即可删除该备份设备。

## 2) 用系统存储过程管理备份设备

- sp\_addumpdevice (创建备份设备)。

例 :为数据库 ems 创建一个逻辑设备名为 emss, 设备的实际名称为 c:\backupdev\ems.bak 的备份设备。代码如图 39 所示。

```
Use ems
EXEC sp_addumpdevice 'disk', 'emss', 'd:\backupdev\ems.bak'
```

图 39 创建备份设备 emss

- sp\_dropdevice(删除备份设备)

例：将数据库 ems 的逻辑设备 emss, 删去。代码如图 40 所示

```
Use ems
EXEC sp_dropdevice 'emss'
```

图 40 删除备份设备 emss

## 2.数据库备份

SQL Server 2000 提供了两种创建备份的途径，即可视化方式和命令方式。

### 1) 用企业管理器管理备份

启动**企业管理器**，登录到要增加备份设备的服务器，打开**数据库**文件夹，右击要进行备份的数据库图标，在弹出菜单中选择**所有任务**选项，再选择**备份数据库**，弹出 SQL Server 备份对话框，如图 41 所示。在**常规**页中的备份选项栏中选择备份类型，单击**添加**按钮选择备份设备，弹出如图 42 所示对话框，在此可以选择**文件名**或**备份设备**。在**重写**选项栏，若选择**追加到媒体**，则将备份内容添加到当前备份之后，若选择**重写现有设备**，则将原备份覆盖。（在**调度**复选框，可对备份的时间表进行设置）。S 所有设置完成后，点击**确定**按钮即可开始备份。

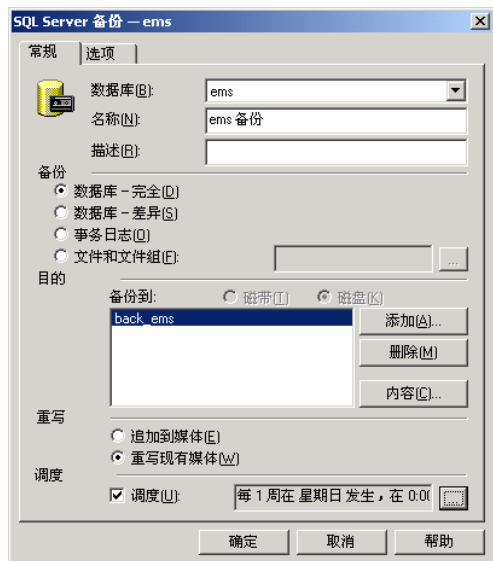


图 41 创建 SQL Server 备份对话框

## 2) 用命令备份数据库

使用 BACKUP 命令进行备份操作。

- 备份完整数据库

例 18 将数据库 ems 完全备份到设备 backupdevice\_emms 上。在驱动器 D 上建立 backupdev 子目录,并执行如图 43 所示的代码,即可将数据库 ems 备份到指定的设备上,图中的第二行是创建备份设备的命令,如果该设备已经存在,则将其注释掉。

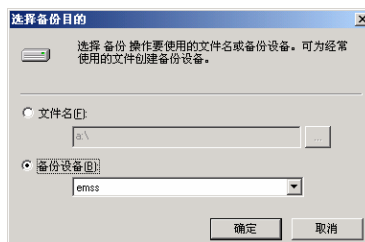


图 42 选择设备对话框

```
use ems
/* EXEC sp_addumpdevice 'DISK','backupdevice_emms', 'd:\backupdev\ems.bak'*/
BACKUP DATABASE ems TO backupdevice_emms
```

图 43 创建数据库备份

- 备份事务日志

例：将数据库 ems 的日志备份到设备 backupdevice\_emms 上。执行如图 44 所示的代码,即可将数据库 ems 备份到指定的设备上,备份的日志与其他备份组织成一个文件,所以,你在资源管理器中看到在 backupdev 下只有一个文件。

```
use ems
BACKUP LOG ems TO backupdevice_emms
```

图 44 创建数据库日志备份

## 3. 恢复数据库

建立备份的目的是在数据库失败时能够尽快恢复到原来状态,并要求保持数据库的一致性。当数据库的备份存在时,SQL Server 提供了将备份数据库恢复到服务器的工具。

### 1) 使用企业管理器恢复数据库

启动**企业管理器**,单击要登录的数据库服务器,从主菜单选择**工具**。在菜单中选择**还原数据库**,弹出**还原数据库**对话框,如图 45 所示。在**常规**页中的**还原为数据库**选项栏中选择要恢复的数据库,在

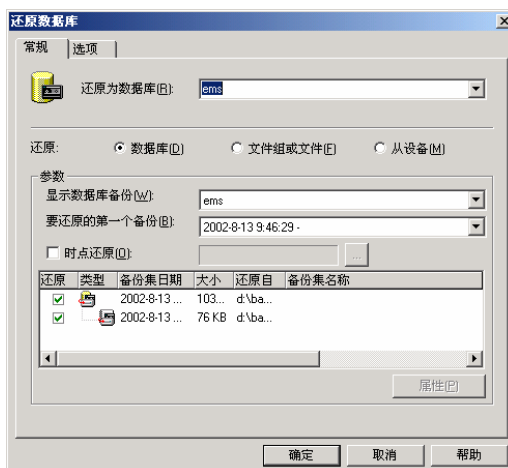


图 45 数据库恢复对话框

还原组中选择相应的备份类型（恢复与备份时的类型可以不同），在参数栏中，选择要恢复的数据库（已经备份的数据库在此均有选项）和使用的备份，缺省情况下使用最近的一次备份。单击选项页设置其他选项。点击确定按钮即可开始恢复数据库。

## 2) 使用命令恢复数据库

数据库恢复命令为：RESTORE DATABASE/LOG。例如：先删除数据库 ems，然后将数据库 ems 在设备 backupdevice\_ems 上的备份恢复到原来状态。执行下面命令即可完成恢复。

```
restore database ems from backupdevice_ems
```

RESTORE 命令中有许多选项，用户可以根据自己的备份恢复策略进行限制。RESTORE 命令还提供了部分数据库恢复能力。在实际应用环境中，如果因误操作而导致部分数据库损坏，如用户错误地删除了一个基表。这时可以使用部分恢复功能。

## 4. 备份和恢复系统数据库

系统数据库保存了 SQL Server 的重要信息，这些数据库的丢失会给系统带来严重后果。对系统数据库的备份是数据库管理员的一项重要任务。在 SQL Server 中重要的数据库有 master、msdb、model。Tempdb 虽然也是系统数据库，但由于 SQL Server 每次启动时，都会重新创建该数据库，而当停止运行时 tempdb 中的所有数据都会被自动清除，所以不必备份 tempdb 数据库。备份系统数据库与备份用户数据库一样，只要数据库状态发生变化，就要通过备份来保存这些变化以防止一旦系统失败而导致数据丢失。在发生以下情况时，应立即对 master 数据库进行备份。

- 增加或删除用户数据库；
- 创建新的登录或执行与登录有关的操作。但增加数据库用户并不影响 master 数据库；
- 创建或删除备份设备；
- 对数据库进行与远程调用有关的配置。

对 master 数据库应进行完全备份。而恢复可以采用两种途径。或使用当前备份，或执行 Rebuild Master Utility 来重建 master 数据库。如果数据库遭到破坏，但 SQL Server 还能启动，这时使用前面介绍的方法用 master 的当前备份来执行恢复；如果系统遭到严重破坏，SQL Server 无法启动，或 master 数据库当前备份也不能使用，则必须使用 Rebuild Master Utility 来重建 master 数据库（这是创建的 master 数据库不包含丢失的数据）然后再使用备份进行恢复。使用 Rebuild Master Utility 创建 master 的步骤为：关闭 SQL Server → 运行位于 program files\Microsoft SQL Server\80\tools\bin\ 下的

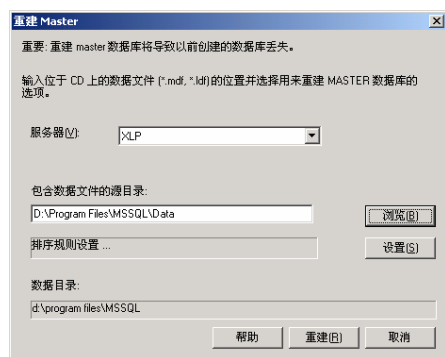


图 46 重建 Master 数据库

**Rebuildm.exe**，出现重建 master 对话框,如图 46 所示单击浏览按钮找到包含 Data 文件的源目录，单击重建按钮重建 master 数据库。重建 master 数据库只是整个恢复工作的开始，然后要装入 master 数据库的备份。接着恢复 msdb、model 及用户数据库。

同时，在 SQL Server 中，可以使用系统过程 `sp_detach_db` 拆开数据库，然后将其重新连接到另外的服务器或同一服务器。拆开数据库就是将数据库从 SQL Server 中删除该数据库，但保持数据和事务日志文件的完整性。然后事务日志和数据文件可在任何运行 SQL Server 的服务器上被重新连接成该数据库。使用系统过程 `sp_attach_db` 可以将以拆开的数据库重新连接到服务器上。

例如：将 ems 数据库从服务器上拆下。执行下列语句。

```
exec sp_detach_db 'smg1','true'
```

如果执行成功，则进入企业管理器，刷新数据库，可以看到 ems 已经从服务器上删除。

例如：将拆开的数据库在重新连接到服务器上。执行下列语句。在此数据库存储文件的位置应由用户指定，通常与 SQL Server 的安装目录有关。

```
exec sp_attach_db smg1,'d:\program files\mssql\data\smg1.mdf'
```

如果执行成功，则进入企业管理器，刷新数据库后，可以看到 ems 又重新在服务器上了。



## 八． 实验练习

### 实验 1 基本表的创建、数据插入

(1) 建立教学管理中的三个基本表：

Students (SNO, SNAME, AGE, SEX) 学生 (学号, 姓名, 年龄, 性别)

Courses (CNO, CNAME, SCORE, PC#) 课程 (课程号, 课程名, 学分, 先行课号)

SC (SNO, CNO, GRADE) 选修 (学号, 课程号, 成绩)

(2) 用 INSERT 命令输入数据。

表 6 基本表 Students 的数据：

S1	LU	20	M
S2	YIN	19	M
S3	XU	18	F
S4	QU	18	F
S6	PAN	14	M
S8	DONG	24	M

表 7 基本表 Courses 的数据

C1	数学	4	M
C2	英语	8	M
C3	数据结构	4	F
C4	数据库	3.5	F
C5	网络	4	M

表 8 基本表 SC 的数据 (空格为未选修, NULL 为选修但还未有成绩):

SNO \ CNO	S1	S2	S3	S4	S6	S8
C1	85	90	89	84	88	87
C2	73	NULL	86	82	75	85
C3	88	80			90	NULL
C4	89	85		NULL	92	88
C5	73	NULL				87

### 实验 2：数据查询

- (1) 列出选修课程号为 C2 的学生学号与姓名。
- (2) 检索选修课程名为“数学”的学生学号与姓名。
- (3) 检索没有选修 C2 课程的学生姓名与年龄。
- (4) 检索选修全部课程的学生姓名。

### 实验 3：数据修改、删除

- (1) 把 C2 课程的非空成绩提高 10%。
- (2) 在 SC 表中删除课程名为“物理”的成绩所对应的元组。
- (3) 在 S 和 SC 表中删除学号为 S8 的所有数据。

### 实验 4：视图的操作

- (1) 建立男生学生的视图, 属性包括学号, 姓名, 选修课程名和成绩。
- (2) 在男生视图中查询平均成绩大于 80 分的学生学号和姓名。

### **实验 5：库函数，授权控制**

- (1) 计算每个学生选修课程的门数、平均成绩。
- (2) 建立一个合法的用户，将 SC 表的查询权限授予该用户。
- (3) 使用 GRANT 语句，把对基本表 students、Courses、SC 的使用权限授予其他用户。

### **实验 6：数据库的备份、恢复**

- (1) 使用完全备份将你的实验数据库备份到软盘。
- (2) 删除你所建立的数据库。
- (3) 恢复你的数据库。
- (4) 在恢复后的数据库上撤销你建立的基本表和视图。

## 九． 数据库课程设计基本要求

### 1． 设计目标

- 1) 运用数据库设计理论设计一个较完善的有实际意义的数据库；
- 2) 掌握目前流行数据库管理系统 SQL Server 2000 的使用与应用开发技术；
- 3) 为数据库开发相应应用程序，构成完整的数据库应用系统；
- 4) 将设计在 SQL Server 2000 上实现。

### 2． 基本要求

本课程设计，按照数据库原理课程中有关数据库应用系统设计章节的内容，主要从以下几个方面要求设计者必须提交相应的设计文档。

#### 1) 问题定义

在设计的第一阶段按软件工程要求给出系统定义，进行需求分析，设计出系统的信息模型即 E-R 图。

#### 2) 库文件结构（关系模式）

选定关系模型作为系统的数据模型，在信息模型的基础上设计合理的数据库文件结构，主要考虑规范化和实际应用需要，一般要求达到三范式，如果需要降低范式时应应对冗余数据及适当的反规范化设计进行说明。

#### 3) 完整性考虑

关系模型的三类完整性约束条件在设计的过程中是必须考虑的，数据之间的关联应详细说明，要求使用 DBMS 对联系进行适当定义和编辑。对有些统计数据可使用触发器（请参考有关资料）。

#### 4) 库文件分类

应用系统中的库文件通常分为：

- 主文件：系统的核心数据，包含有需要永久保存的数据，是数据库系统的共享资源。
- 事务文件：记录事务处理的轨迹，保存数据更新的必要信息，供复核和数据恢复用，属于面对应用的局部数据，为了让设计者体会数据库应用程序的设计技巧，在此要求设计者将此类文件的结构设计出来。
  - 工作文件：应用程序在工作时对应的库文件。
  - 临时文件：存放应用程序执行过程的某些轨迹，供系统缓冲或恢复之用。

为了主文件的安全和提高并发度，通常情况下，将应用程序接收的数据写入临时文件和工作文件（请思考如果将这些数据用变量表示，会用什么不同）。因此，在你的系统中建议区分各类文件。

#### 5) 并发控制

数据库系统中的数据是全局共享的，因此在应用程序开发的过程中，应考虑多用户并

发执行的情况，建议在开发前对工作区进行分配，每个文件尽量在同一工作区打开。建议遵循两段锁协议和共同开发中数据库文件的打开顺序。

#### 6) 安全性考虑

数据库的安全性是至关重要的，建议为系统开发密钥功能，对关键数据应采用隐码存放。

#### 7) 系统体系结构

系统可以使用自含式的 DBMS 开发，也可以使用嵌入式 DBMS，可以设计为单机版或网络版。

#### 8) 用户接口设计

用户接口是系统最终提交给用户的操作界面，可使用菜单式也可使用按钮式。但应使应用程序和数据库相互隔离，禁止将数据库直接暴露给用户。

#### 9) 应用程序功能设计

应用系统的基本功能应根据实际目标来设定，通常有增、删、改、查、备份、恢复、密钥等功能。

### 3. 实验系统参考题目

选题说明：2 或 3 人一组，自由组合，从下列题目中任选一个，在课程设计期间按要求完成设计任务，并提交一份完整的设计报告和已调通的应用系统模型。

- |             |           |
|-------------|-----------|
| 1) 学籍管理系统   | 6) 网上销售系统 |
| 2) 教学管理系统   | 7) 仓储管理系统 |
| 3) 学生管理系统   | 8) 图书管理系统 |
| 4) 财务管理系统   | 9) 超市收银系统 |
| 5) 银行储蓄管理系统 | 10) 自选题目  |

### 4. 文档内容

在课程设计提交的设计报告中，应包括以下内容：

- 1) 数据字典
- 2) 系统设计
- 3) 源程序代码

文档请按照软件工程的要求与格式书写。