

实验四 索引、数据完整性与安全性

一、实验目的

- (1) 掌握利用 SQL Server Management Studio 和 SQL 语言建立、删除索引的方法;
- (2) 掌握利用 SQL Server Management Studio 和 SQL 语言实现数据完整性的方法;
- (3) 掌握在 SQL Server Management Studio 中实现数据安全性管理的方法。

二、实验原理

1.索引

在关系型数据库中，索引是一种可以加快数据检索的数据库结构。SQL Server 系统中主要有两种类型的索引，即聚集索引、非聚集索引。

(1) 聚集索引

聚集索引定义了数据在表中存储的物理顺序。一个表只能定义一个聚集索引。

(2) 非聚集索引

非聚集索引并不存储表数据本身。相反，非聚集索引只存储指向表数据的指针，该指针作为索引键的一部分，因此，在一个表中同时可以存在多个非聚集索引。

(3) 利用 SQL 命令建立索引

简化语法格式：

```
CREATE [UNIQUE] [CLUSTERED|NONCLUSTERED]  
INDEX index_name ON {table|view}(column|ASC|DESC)[,...n])
```

其中：UNIQUE。可选。该选项用于通知 SQL Server 索引中列出的列的值是每行唯一的。如果试图插入重复的行，则该选项会强制 SQL Server 返回一个错误信息。

CLUSTERED 或 NONCLUSTERED。可选。如果这两个选项都没有被明确列出，则默认将索引创建为 NONCLUSTERED（非聚集索引）。

(4) 通过 SQL 命令删除索引

语法格式：DROP INDEX 'table.index|view.index'[,...n]

2.表主键和 UNIQUE 约束

表主键通过表数据中一个列或者多个列组合的数据来唯一标识表中的每一行数据。即表主键就是用来约束数据表中不能存在相同的两行数据。在 SQL Server 系统中，定义表的主键可以在创建表的同时定义，也可以给已有的表添加主键。如果表中指定的主键不再需要时，也可以通过 Transaction-SQL 语句将其删除。

(1) 创建表的同时创建 PRIMARY KEY 约束或 UNIQUE 约束

语法格式：

```
CREATE TABLE table_name
```

```
(column_name datatype [CONSTRAINT constraint_name]
PRIMARY KEY|UNIQUE [...n])
```

(2) 通过修改表创建 PRIMARY KEY 约束或 UNIQUE 约束

语法格式:

```
ALTER TABLE table_name
ADD[CONSTRAINT constrain_name] PRIMARY KEY |UNIQUE (column[,...n])
```

(3)删除 PRIMARY KEY 约束或 UNIQUE 约束

语法格式:

```
ALTER TABLE table_name
DROP CONSTRAINT constrain_name [...n]
```

删除其他约束的方法与此相同，只是约束名不同，下面其他约束就不再列举此项了。

3.外键

外键也是由表的一个列或多个列组成的，它是和在相关表中事先定义的具有唯一性的列一起使用的。

外键约束用来建立和强调两个表之间的关联。

(1)创建表的同时定义外码约束

语法格式:

```
CREATE TABLE table_name
(column_name datatype [FOREIGN KEY] REFERENCES ref_table(ref_column)
[,...n])
```

(2) 通过修改表定义外码约束

语法格式:

```
ALTER TABLE table_name
ADD [CONSTRAINT constraint_name]
FOREIGN KEY(column [...n]) REFERENCES ref_table(ref_column) [...n])
```

4.CHECK 约束

检查约束使用逻辑表达式来限制列上可以接受的数据。可以在一列上使用多个检查约束，也可以在表上建立一个可以在多个列上使用的检查约束。

(1) 利用 SQL 语句在创建表时创建 CHECK 约束

语法格式:

```
CREATE TABLE table_name
(column_name datatype not null|null
[DEFAULT constraint_expression]
[[check_name]CHECK(logical_expression)][,...n])
```

(2) 利用 SQL 语句在修改表时创建 CHECK 约束

语法格式:

```
ALTER TABLE table_name
```

ADD CONSTRAINT check_name CHECK (logical_expression)

5.身份验证模式

身份验证模式是 Microsoft SQL Server 系统验证客户端和服务端之间连接的方式。SQL Server 提供了两种身份验证模式：Windows 身份验证模式和混合模式。在 Windows 身份验证模式中，用户通过 Microsoft Windows 用户账户连接时，SQL Server 使用 Windows 操作系统中的信息验证账户名和密码。在混合模式中，既可采取 Windows 身份验证，也可采取 SQL Server 身份验证。在服务器节点上右击选择“属性”，打开服务器属性对话框（图 4-1），在该对话框中可以设置修改 SQL Server 的身份验证模式。

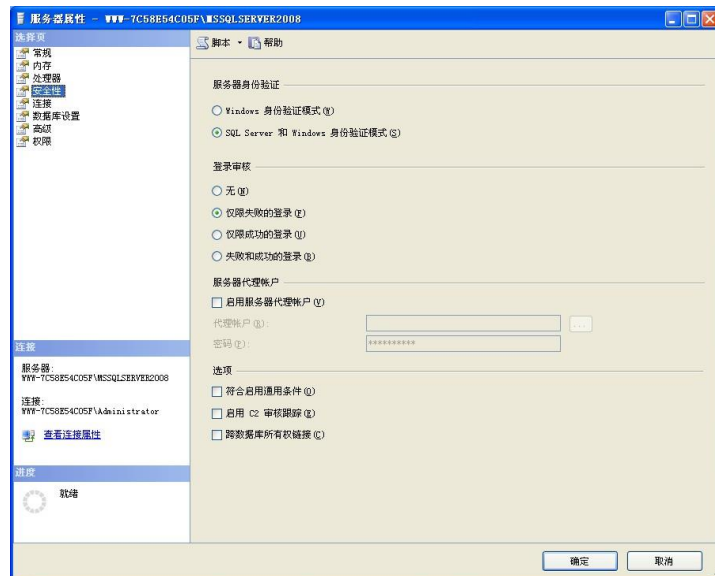


图 4-1 服务器属性对话框

SQL Server 默认的身份验证类型为 Windows 身份验证，只有获得 Windows 账户的客户才能建立与 SQL Server 的信任连接（即 SQL Server 委托 Windows 验证用户的密码）。如果其他客户需要建立信任连接，则必须为它们创建 SQL Server 账户登录。

6.固定服务器角色

为便于管理服务器权限，SQL Server 提供了若干角色。服务器角色也称为固定服务器角色，因为不能创建新的服务器级角色。服务器级角色的权限作用域为服务器范围，可以向服务器角色中添加 SQL Server 登录名、Windows 账户和 Windows 组。SQL Server 包括 8 个服务器角色，用户可以指派给 8 个服务器角色中的任一角色。这些角色如图 4-2 所示。

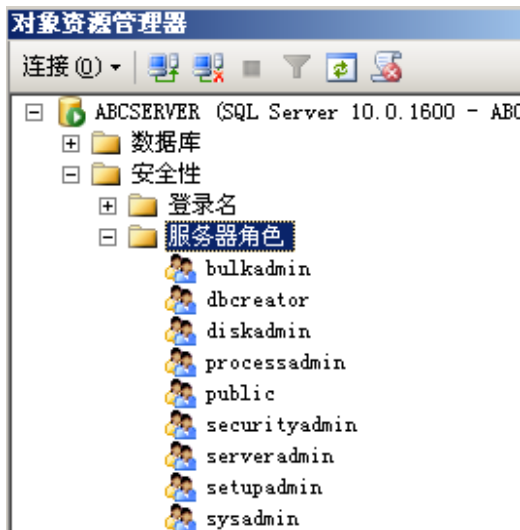


图 4-2 固定服务器角色

7.固定数据库角色

固定数据库角色存在于每个数据库中，在数据库级别提供管理特权分组。管理员可将任何有效的数据库用户添加为固定数据库角色成员，每个成员都将获得固定数据库角色的权限。用户不能增加、修改和删除固定数据库角色。

SQL Server 在数据库级设置了固定数据库角色来提供最基本的数据库权限的综合管理。数据库创建时，系统默认创建了 10 个固定数据库角色。

使用系统存储过程 `sp_helpdbfixedrole` 可以返回固定数据库角色的列表，如图 4-3 所示。

	DbFixedRole	Description
1	db_owner	DB Owners
2	db_accessadmin	DB Access Administrators
3	db_securityadmin	DB Security Administrators
4	db_ddladmin	DB DDL Administrators
5	db_backupoper...	DB Backup Operator
6	db_datareader	DB Data Reader
7	db_datawriter	DB Data Writer
8	db_denydatare...	DB Deny Data Reader
9	db_denydatawri...	DB Deny Data Writer

图 4-3 查看固定数据库角色

三、实验示例

1.使用 SQL Server Management Studio 创建索引

SQL Server Management Studio—数据库—GZGL（数据库名）—XS（需建立索引的表名），右击索引节点，在弹出的快捷菜单中选择新建索引命令。在新建索引对话框的常规页面配置索引的名称，选择索引的类型，是否是唯一索引等，如图 4-4 所示。

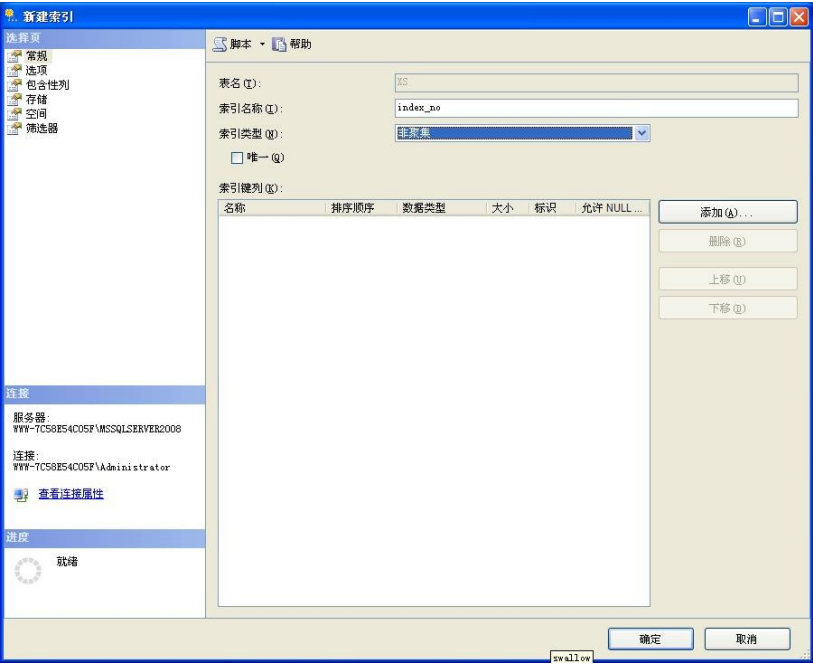


图 4-4 新建索引对话框

单击添加按钮，打开从“dbo.XS”中选择列对话框，在对话框中的表列列表中启用学号复选框，如图 4-5 所示。



图 4-5 选择索引列



图 4-6 索引创建成功

单击确定按钮，返回新建索引对话框，然后单击新建索引对话框的确定按钮，索引节点下便生成了一个名“index_no”的聚集索引，说明该索引创建成功，如图 4-6 所示。

2.使用 Transact-SQL 语句创建索引

通过图形化工具创建的名称为“index_no”的聚集索引也可以通过如下代码进行创建：

```
create clustered index index_no on XS (学号)
```

3.数据完整性

【例 1】在 GZGL 库中创建主表 XS1 和 KC1，其中 XS1.学号和 KC1.课程号为主键，然后定义从表 XS_KC1.XS_KC1.学号为外码，与 XS1 表的主键对应，当对主表进行更新和删除操作时，对从表采用级联操作，XS_KC1.课程号与 KC1 的主键对应，当对主表进行更新和删除时，对从表采用 NO ACTION 方式。

```
create table xs1
(学号 char(12) primary key,
 姓名 char(8) null,
 性别 bit null default(0),
 总学分 tinyint check(总学分>=0 and 总学分<170) null)
```

```
create table kc1
(课程号 char(6) primary key,
 课程名 char(20) null,
 学分 integer null)
```

```
create table xs_kc1
(学号 char(12) foreign key references xs1(学号)
on delete cascade on update cascade,
 课程号 char(6) not null references kc1(课程号)
on delete no action on update no action,
 成绩 integer null)
```

【例 2】删除 GZGL 数据库的 XS 表中学号字段的 CHECK 约束。

```
alter table XS drop constraint xhcheck
```

4.数据安全性的实现。

(1) 创建 Windows 账户登录

① 创建 Windows 用户

我的电脑—控制面板—管理工具—计算机管理—本地用户和组--右击用户—新用户，打开新用户对话框，输入相应的信息，如用户名、描述和密码等；

创建用户完成后，右击组节点，从快捷菜单中选择“新建组”命令，打开新建组对话框，输入组名，单击添加按钮，进入选择用户对话框，把前面创建的用户添加到这个组中。

② 将 Windows 账户加入到 SQL Server 中。

SSMS—安全性—右击登录名—新建登录名，打开新建登录名对话框，如图 4-7 所示。

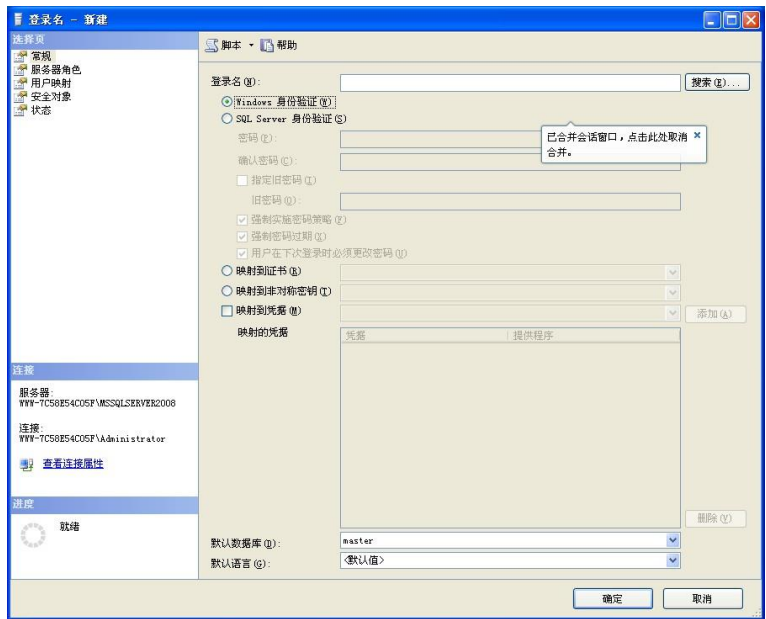


图 4-7 新建 Windows 登录对话框

在上面的对话框中单击搜索按钮，打开选择用户或组对话框（图 4-8），单击对象类型按钮，进入对象类型对话框，启用组复选框，如图 4-9 所示。



图 4-8 选择用户或组



图 4-9 设置对象类型

单击确定按钮返回。在选择用户或组对话框中，可以将创建的组添加进去。在登录名-新建对话框中，选择 Windows 身份验证，并且选择需使用的数据库为默认数据库，单击确定按钮完成创建。

（2）创建 SQL Server 账户登录

SSMS—安全性—右击登录名—新建登录名，打开新建登录名对话框，输入登录名 zhou、密码，选择“SQL Server 身份验证”，点击“确定”按钮完成 SQL Server 登录账户的创建（图 4-10）。

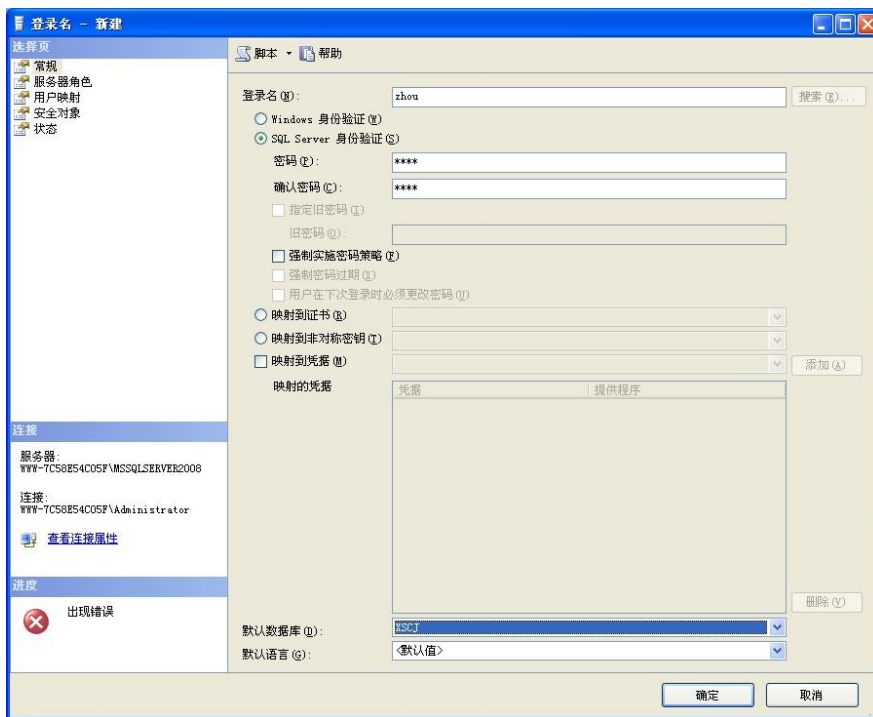


图 4-10 确定 SQL Server 登录名对话框

(3) 创建数据库用户账户

SSMS—服务器—数据库—GZGL—安全性，右击安全性下的用户节点，选择新建用户命令，打开数据库用户-新建对话框，单击登录名文本框旁边的按钮，打开选择登录名对话框，单击浏览，打开查找对象对话框，选择上例创建的登录账户 zhou，如图 4-11 所示。

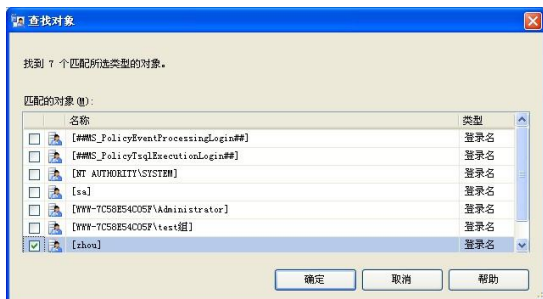


图 4-11 选择登录账户

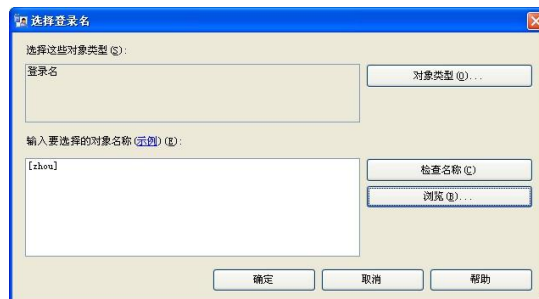


图 4-12 选择登录名对话框

单击确定按钮返回，在选择登录名对话框中就可以看到选择的登录名对象（图 4-12）。

单击确定按钮返回，设置用户名为 LD，选择构架为 dbo，并设置用户的角色为 db_owner，具体设置如图 4-13 所示。

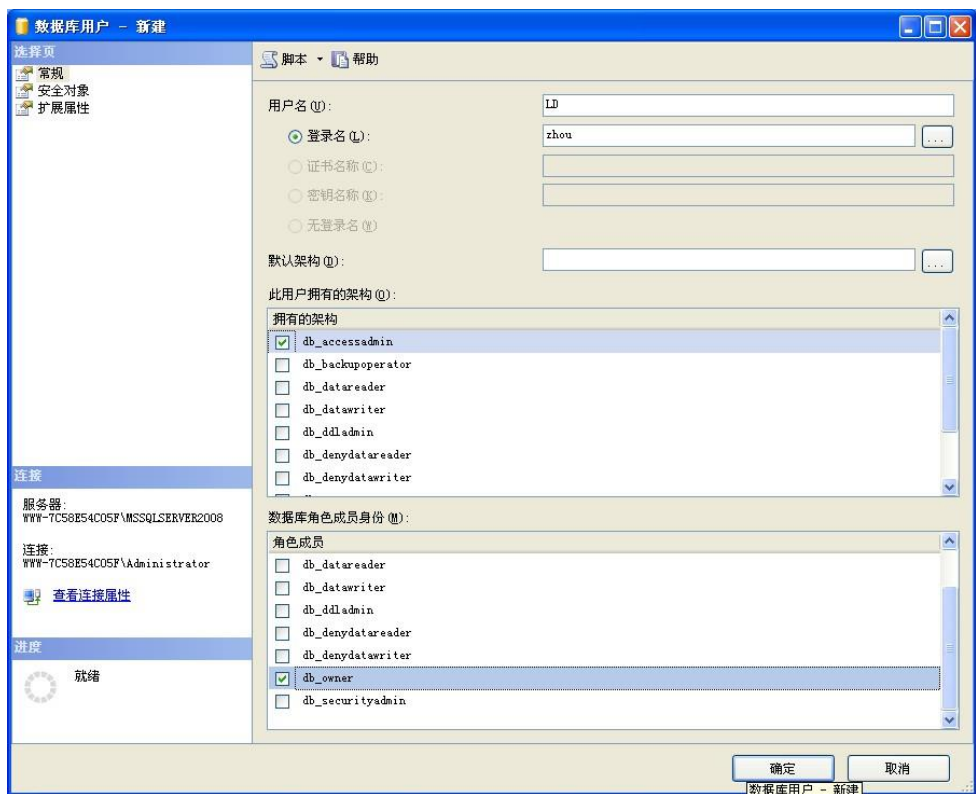


图 4-13 新建数据库用户

(4) 删除登录账户和数据库用户

SSMS—安全性—右击登录名，在弹出的快捷菜单中选择删除命令，打开删除对象对话框，单击确定按钮即可删除登录账户。

SSMS—服务器—数据库—GZGL—安全性，右击安全性下的用户节点，从快捷菜单中选择删除命令，打开删除对象对话框，单击确定按钮即可删除数据库用户。

(5) 分配服务器角色

SSMS—安全性—服务器角色，在展开的服务器角色节点上双击 sysadmin 节点，打开服务器角色属性—sysadmin 对话框，单击添加按钮，打开选择登录名对话框。单击浏览按钮，打开查找对象对话框，启用 zhou 选项旁边的复选框（图 4-14）。

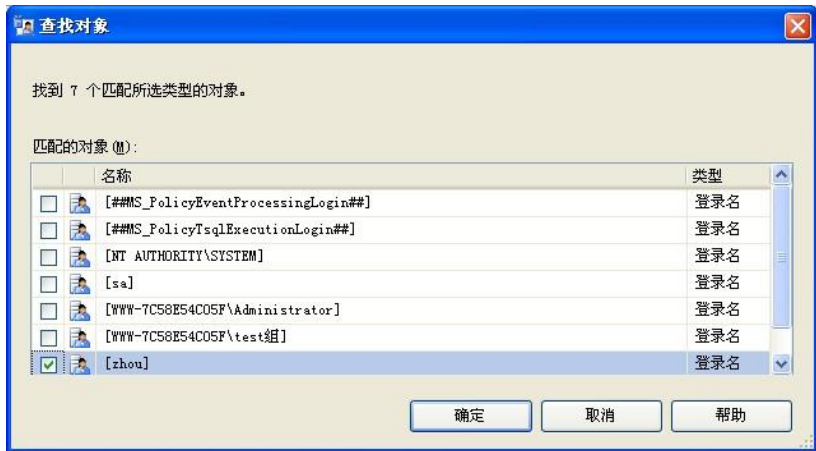


图 4-14 添加登录名

单击确定按钮返回到选择登录名对话框，可以看到刚添加的登录账户 zhou（图 4-15）。



图 4-15 选择登录名对话框

单击确定按钮返回服务器角色属性对话框，在角色成员列表中可以看到服务器角色 sysadmin 的所有成员，其中包括刚添加的 zhou，如图 4-16 所示。

用户可通过添加按钮添加新的登录账户，也可通过删除按钮删除某些不需要的登录账户。

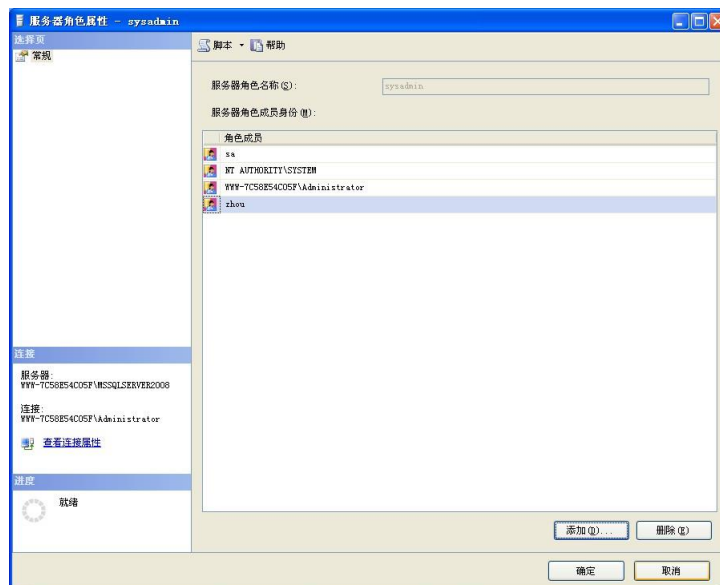


图 4-16 服务器角色属性对话框

(6) 分配数据库角色及授权

SSMS—数据库—GZGL—安全性—角色—数据库角色，双击 db_owner，打开数据库角色对话框（图 4-17），可见前面已添加的 LD 用户。单击添加，打开数据库用户或角色对话框，然后单击浏览按钮打开查找对象对话框，选中其他数据库用户复选框，返回数据库角色属性对话框即可看到当前该角色拥有的架构以及该角色所有的成员，包括刚添加的数据库用户。

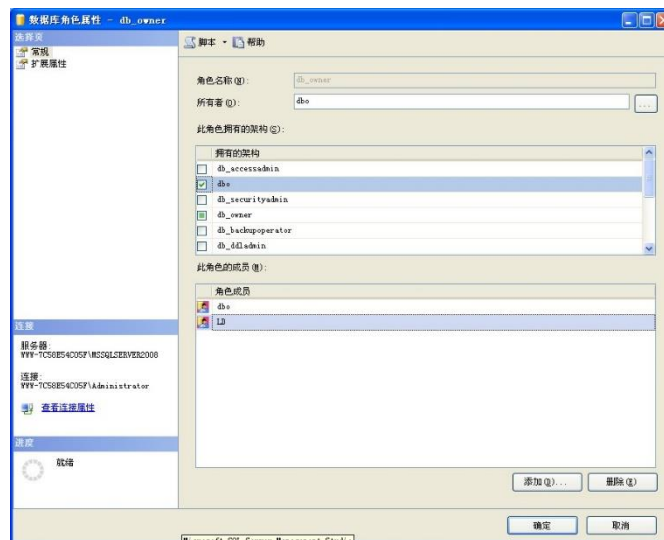


图 4-17 数据库角色属性对话框

四、实验内容

1. 使用修改表命令将 members 表的 m_address 设置为非空，添加如下记录，体会 NOT NULL

约束的使用。

‘wenh’，‘翁红’，‘男’，NULL，5500，‘888’

首先，输入更新数据表 members 的 SQL 代码：

```
ALTER TABLE members ALTER COLUMN m_address VARCHAR(50) NOT NULL
```

运行更新数据表的 SQL 代码，然后输入插入记录到数据表 members 的 SQL 代码：

```
INSERT INTO members VALUES('wenh','翁红','男',NULL,5500,'888')
```

执行此段 SQL 插入记录代码后，将在结果框中显示以下消息，表示执行插入操作失败：

服务器: 消息 515，级别 16，状态 2，行 1

无法将 NULL 值插入列 'm_address,表 'eshop.dbo.members'；该列不允许空值。INSERT 失败。

语句已终止。

2. 在 members 表中，使用 SQL 语言将 m_account 设置为主键，添加如下记录，体会 PRIMARY

KEY 约束的作用。并观察 SSMS 中键是否增加，系统是否为主键创建了索引。

‘liuzc’，‘翁红’，‘男’，‘湖南株洲’，5500，‘123’

3. 在 members 表中，使用 SQL 语言为 m_name 增加 UNIQUE 约束，添加如下记录，体会

UNIQUE 约束的作用。并观察系统是否为唯一性约束创建了索引。

‘jjyw’，‘津津’，‘男’，‘湖南株洲’，5500，‘jjyw’

4. 在 members 表中，使用 SQL 语言为会员增加性别只能为男或女的 CHECK 约束，添加如下记

录，体会 CHECK 约束的作用。并观察 SSMS 中约束是否增加。

‘liuzh’，‘刘忠怀’，‘工’，‘湖南株洲’，3000，‘liuzh’

8. 将服务器的登录模式设置为“SQL Server 和 Windows 身份验证模式”。

9. 新建 SQL Server 登录账户。

在安全性→登录名中，新建SQL Server登录账户，输入登录名NewLogin，选择SQL Server身份验证并输入密码，默认数据库设置为eshop，并在用户映射中选择eshop。

10.查看 NewLogin 用户的权限,并以 sa 用户身份和 NewLogin 用户身份分别登录 SQL SERVER,比较 sa 与 NewLogin 用户权限的区别。(sa 默认为 sysadmin 角色，普通用户仅默认为 public 角色)

11. 用 SQL 语句为 NewLogin 用户赋予 members 中 m_account 和 m_address 列的查询权限。

12. 关闭所有窗口，登录为 NewLogin。

展开数据库-esshop-表，可见表节点下面只显示了拥有查看权限的一张表（图 4-21），由于在列权限窗口设置该角色的权限为不允许查看 members 表的后面 2 列，故在输入下列查询语句时将出现错误。



图 4-21 使用 SELECT 语句验证权限

五、注意事项

1. Windows 验证模式和混合验证模式。
2. 登录账户和数据库用户的关系。