

第 5 章 视图

教学目标

本章将要学习 SQL Server 2005 中视图操作的相关知识，包括视图概述、视图的创建、视图的查看、视图的修改、视图的删除以及视图中数据的查询和修改等。

重点难点

- 视图的概念
- 视图的创建、查看、修改、删除
- 通过视图插入、更新、删除基本表中的数据
- 使用视图加强数据的安全性

5.1 视图概述

视图是关系数据库系统提供给用户以多种角度观察数据库中数据的重要机制。使用视图可以简化用户对数据的操作，还可以保证数据的安全性。

5.1.1 视图的概念

在网上书店数据库中有这样两张表：图书信息表（BookInfo 表）和图书类别表（BookType 表）。如果想查询所有图书对应的图书类别名称，则需要同时对这两张表进行检索，对应的 SELECT 语句如下：

```
SELECT B_Name,BT_Name,B_Author,B_Publisher
FROM BookInfo BI,BookType BT
WHERE BI.BT_ID=BT.BT_ID
```

这种检索可能在应用系统中使用的比较广泛，并且比较固定，这时就可以通过创建视图来代替这种检索。

视图是一种数据库对象，是从一个或多个表或视图中导出的虚表。视图所对应的数据并不真正地存储在视图中（索引视图除外），而是存储在视图所引用的基本表中。视图的结构和数据是对基

本表进行查询的结果。

视图一经定义便存储在数据库中，与其相对应的数据并没有在数据库中另外存储一份，通过视图看到的数据只是存放在基本表中的数据。对视图的操作与对表的操作一样，可以对其进行查询、修改（有一定的限制）和删除。当通过视图修改数据时，实际上是在修改基本表中的数据；同样地，如果基本表中的数据发生变化，则这种变化也会自动地反映到视图中。

图 5-1 显示了在这两个表上建立的典型视图。从图 5-1 中可以看出，视图与表非常相似，也包含一系列被定义的数据列和多个数据行。



图 5-1 典型视图

5.1.2 视图的优点

使用视图有很多优点，它使用户对数据库中数据的操作更灵活、更安全。使用视图的主要优点如下：

(1) 简化操作。视图大大简化了用户对数据的操作。因为在定义视图时，如果视图本身是一个复杂查询的结果集，这样在每一次执行相同的查询时就不必重新写这些复杂的查询语句，只要一条简单的查询视图语句即可。可见，视图向用户隐藏了表与表之间复杂的连接操作。

(2) 分割数据。在某些情况下，由于表中的数据量太大，因此需要对表中的数据进行水平或

者垂直分割。如果直接分割数据表，可能会引起应用程序的执行错误。这时可以使用视图对数据表中的数据进行分块显示，从而使原有的应用程序仍可以通过视图来重载数据。

(3) 安全性。视图作为一种安全机制，可以通过限制用户使用数据来实现。通过视图，用户只能查看和修改他们所能看到的数据，其他数据库或表中数据既不可见也不可访问。如果某一用户想要访问视图的结果集，则必须授予其访问权限。视图所引用表的访问权限与视图权限的设置互不影响。视图可以将用户限定在表中的特定行与特定列上。

5.1.3 视图的分类

在 SQL Server 2005 中，视图可分为 3 种类型，即标准视图、索引视图和分区视图。

(1) 标准视图。标准视图是从一个或几个基本表中导出的表，是一种虚拟表，不占用物理存储空间，只保存了一个 SQL 查询。通常情况下的视图都是标准视图，它是一种逻辑对象。

(2) 索引视图。相对于标准视图而言，对于那些需要对大量数据进行复杂处理的视图，为每个引用视图的查询动态生成结果集的开销就很大。如果在查询中需要频繁地引用这类视图，那么就可以通过对视图创建唯一聚集索引来提高性能。对视图创建唯一聚集索引后，结果集将存储在数据库中，就像为表创建了聚集索引一样。

索引视图是被具体化的视图，即它是通过计算并存储的视图。索引视图可以提高某些类型查询的性能，尤其适用于聚合许多行的查询，但不太适用经常更新的基本数据集。

(3) 分区视图。通过使用分区视图，可以连接一台或多台服务器的成员表中的分区数据，使得这些数据看起来就像来自同一个表中一样。分区视图又分为本地分区视图和分布式分区视图。本地分区视图是指视图中所有参与的基本表或视图存在于同一个 SQL Server 实例中；而分布式分区视图表示视图中所有参与的基本表或视图中至少有一个存在于不同的服务器 SQL Server 实例中。

5.2 创建视图

在 SQL Server 2005 中创建视图有两种方法：一是使用 SSMS 创建视图，二是使用 T-SQL 语句中的 CREATE VIEW 命令创建视图。建立视图时应注意以下几点：

(1) 只能在当前数据库中创建视图，但是视图所引用的表或视图可以来自其他数据库，甚至是其他服务器上的表或视图。

(2) 建立视图时必须遵循标识符命名规则，在数据库范围内视图名称要具有唯一性，不能与用户所拥有的其他数据库对象名称相同。

(3) 一个视图最多可以引用 1024 个字段，这些字段可以来自一个表或视图，也可以来自多个表或视图。

(4) 视图可以建立在其他视图之上。SQL Server 允许视图最多嵌套 32 层。

(5) 即使删除了一个视图所依赖的表或视图，这个视图的定义仍然保留在数据库中。

(6) 不能在视图上定义全文索引。

(7) 不能在视图上绑定规则、默认值和触发器。

(8) 不能建立临时视图，也不能在一个临时表上建立视图。

5.2.1 使用 SSMS 创建视图

【例 5-1】在网上书店数据库 BookStore 中创建 SQL Server 类别图书基本信息的视图 vw_BookInfo。

(1) 启动 SQL Server Management Studio, 在“对象资源管理器”中展开“数据库”节点, 然后展开 BookStore 数据库节点。

(2) 右击“视图”节点, 选择“新建视图”, 如图 5-2 所示。

(3) 打开“添加表”对话框, 单击要添加到新视图中的表或视图, 按 CTRL 键可以同时选择多个表。这里选择 BookInfo 表和 BookType 表, 然后单击“添加”按钮, 所选表格就会出现在视图窗口中, 并按照表的结构, 自动建立两个表的连接。单击“关闭”按钮, 完成表的添加, 如图 5-3 所示。



图 5-2 选择“新建视图”命令



图 5-3 “添加表”对话框


(4) 视图窗口如图 5-4 所示。视图窗口共有 4 个子窗格, 单击“查询设计器”工具栏上相应的按钮可以控制这些子窗格是否可见。


关系图窗格用于选择基本表和列, 还可以添加多个表, 设置各表和列之间的关系。如果图中两张表之间没有建立连接, 可以用左键按住 BookInfo 表中的 BT_ID 字段, 然后拖向 BookType 表中的 BT_ID 字段, 这时两张表之间就会建立起连接。在这里选择 BookInfo 表中的 B_Name、B_Author、B_Publisher 字段和 BookType 表中的 BT_Name 字段。

条件窗格用于设置选择列的属性及查询条件。在这里分别设置各列的别名, 并在 BT_Name 字段所对应的筛选器位置设置查询条件“=SQL Server”。

在 SQL 窗格中会显示创建视图所需要执行的 SQL 语句。

以上三个窗格的设置是联动的, 即改变其中一个窗格中的内容, 另两个窗格中的内容会随之变化。

(5) 单击“查询设计器”工具栏上的  按钮, 或者在视图窗口中右击, 弹出如图 5-5 所示的快捷菜单, 选择“执行 SQL”命令, 就可以看到视图对应的结果集, 效果见图 5-4 中显示结果窗格。

(6) 单击“标准”工具栏上的  按钮, 或者右击“视图”选项卡, 选择“保存视图”命令, 如图 5-6 所示。

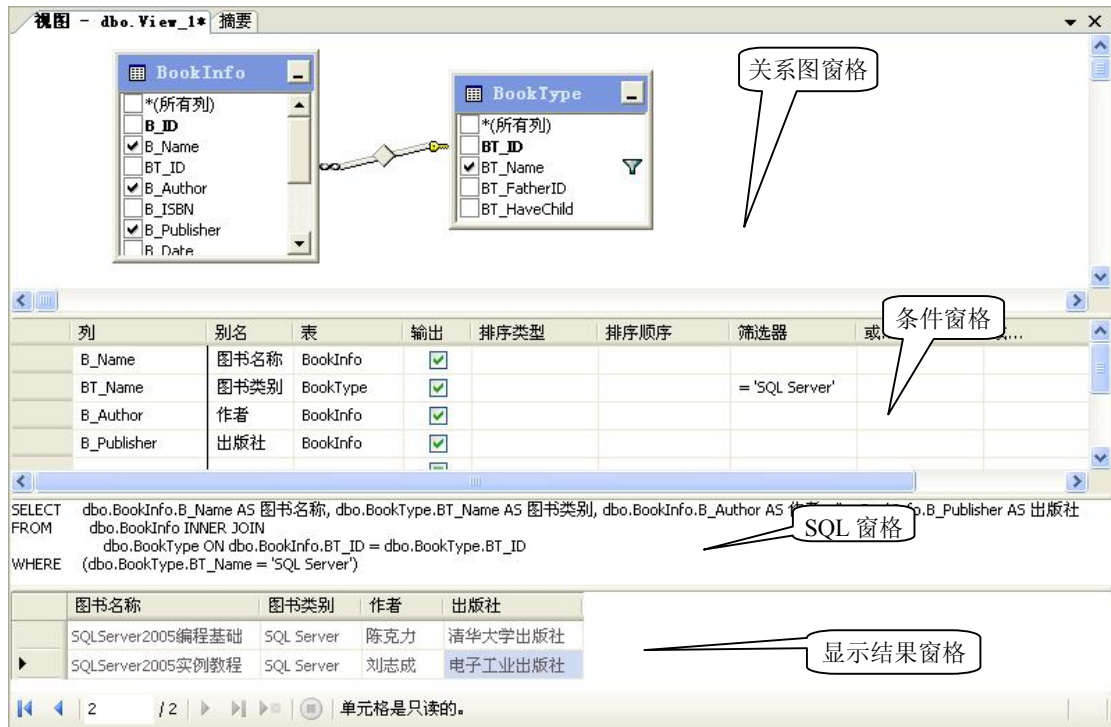


图 5-4 视图窗口



图 5-5 选择“执行 SQL”命令

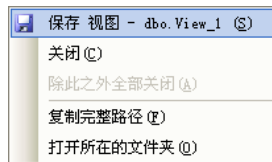


图 5-6 选择“保存视图”命令

(7) 打开“选择名称”对话框，如图 5-7 所示，输入视图的名称 vw_BookInfo，单击“确定”按钮，这样就完成了视图的创建。



图 5-7 输入视图名称

5.2.2 使用 T-SQL 语句创建视图

使用 T-SQL 命令 CREATE VIEW 可以创建视图，其基本语法格式如下：

```
CREATE VIEW <视图名>
```

```
[ WITH <视图属性> ]
AS
<SELECT 语句>
[ WITH CHECK OPTION ]
```

其中，各参数说明如下：

- 视图名：必须符合标识符规则。
- 视图属性：包括 ENCRYPTION（文本加密）、SCHEMABINDING（将视图绑定到基本表的架构）和 VIEW_METADATA（指定引用视图的元数据）。
- SELECT 语句：用于创建视图的 SELECT 语句，利用 SELECT 命令可以从多个表或视图中选择列来构成新视图的列，也可以使用 Union 关键字联合起来多个 SELECT 语句。
- WITH CHECK OPTION：强制针对视图执行的所有数据修改语句都必须符合在 SELECT 语句中设置的条件。通过视图修改行时，WITH CHECK OPTION 可确保提交修改后，仍可通过视图看到修改后的数据。

【例 5-2】从图书信息表（BookInfo 表）和订单详情表（OrderDetails 表）中创建每张订单中图书详情的视图 vw_OrderBook，并查询该视图中的数据。

```
USE BookStore
GO
CREATE VIEW vw_OrderBook
AS
SELECT OD.O_ID, BI.B_Name, BI.B_Author, BI.B_ISBN, BI.B_Publisher
FROM BookInfo BI, OrderDetails OD
WHERE BI.B_ID = OD.B_ID
GO
SELECT * FROM vw_OrderBook
```


在查询窗口中输入上述 SQL 代码，并单击“SQL 编辑器”工具栏上的  执行按钮，执行查询操作，运行结果如图 5-8 所示。



图 5-8 查询视图结果

5.3 查看视图

创建视图后，可以查询由视图定义的数据和视图的信息。

5.3.1 使用 SSMS 查看视图

【例 5-3】查看视图 vw_BookInfo 定义的数据。

(1) 启动 SQL Server Management Studio，在“对象资源管理器”中展开“数据库”节点，然后展开 BookStore 数据库节点，再展开“视图”节点。

(2) 右击 vw_BookInfo 视图，选择“打开视图”命令，如图 5-9 所示。这时在文档窗口中可以查看视图定义的数据了，如图 5-10 所示。

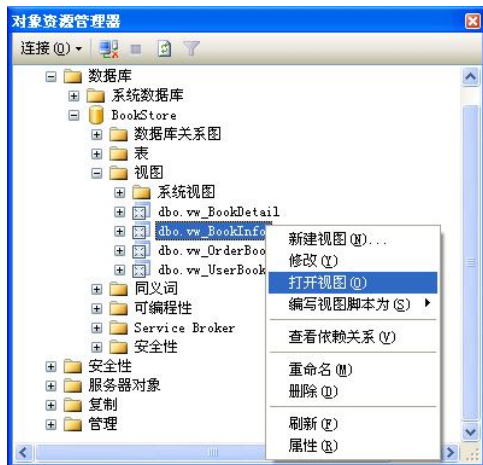


图 5-9 选择“打开视图”命令

| 视图 - dbo.vw_BookInfo 摘要 | | | |
|-------------------------|------------|------|---------|
| 图书名称 | 图书类别 | 作者 | 出版社 |
| SQLServer2005编程基础 | SQL Server | 陈克力 | 清华大学出版社 |
| SQLServer2005实例教程 | SQL Server | 刘志成 | 电子工业出版社 |
| ▶* | NULL | NULL | NULL |

图 5-10 查看视图数据

【例 5-4】查看视图 vw_BookInfo 的基本信息。

(1) 启动 SQL Server Management Studio，在“对象资源管理器”中展开“数据库”节点，然后展开 BookStore 数据库节点，再展开“视图”节点。

(2) 右击 vw_BookInfo 视图，选择“属性”，如图 5-11 所示。

(3) 打开“视图属性”对话框，在该对话框中可以查看视图的“常规”、“权限”和“扩展属性”，如图 5-12 所示。

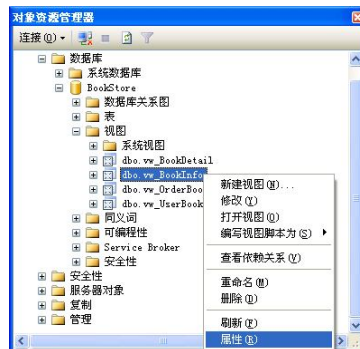


图 5-11 选择“属性”命令

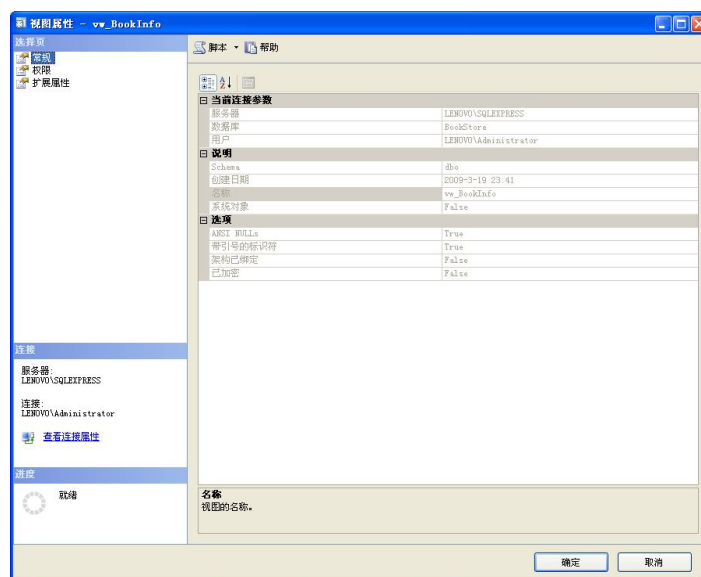


图 5-12 查看视图属性

【例 5-5】查看视图 vw_BookInfo 的依赖关系。

(1) 启动 SQL Server Management Studio, 在“对象资源管理器”中展开“数据库”节点, 然后展开 BookStore 数据库节点, 再展开“视图”节点。

(2) 右击 vw_BookInfo 视图, 选择“查看依赖关系”, 如图 5-13 所示。

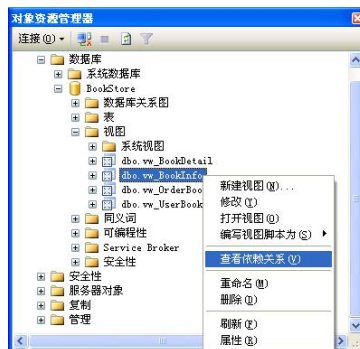


图 5-13 选择“查看依赖关系”命令

(3) 打开“对象依赖关系”对话框，在该对话框中可以查看视图 vw_BookInfo 所依赖的对象，也可以查看依赖于视图 vw_BookInfo 的对象，如图 5-14 所示。

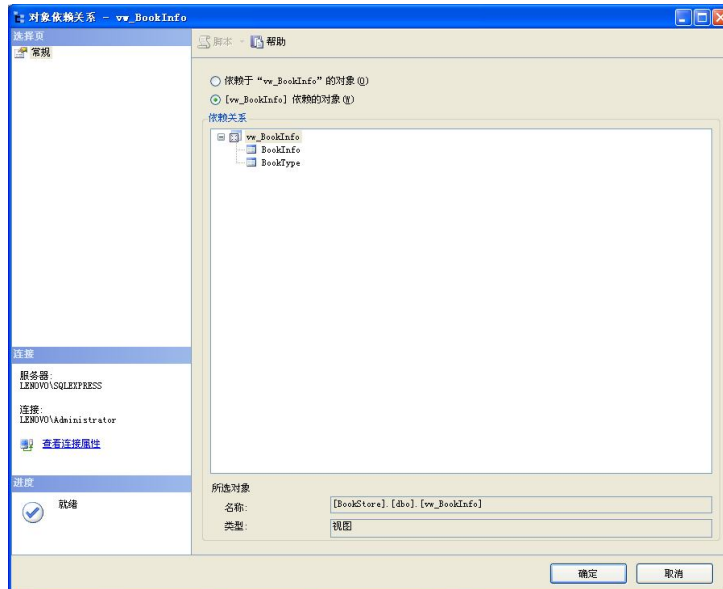


图 5-14 查看视图依赖关系

5.3.2 使用 T-SQL 语句查看视图

视图与表具有相似的结构，当创建视图以后，用户就可以像对基本表一样对视图进行查询操作。

【例 5-6】 查询清华大学出版社出版的 SQL Server 类图书详情。

```
USE BookStore
GO
SELECT *
FROM vw_BookInfo
WHERE 出版社='清华大学出版社'
```

运行结果如图 5-15 所示。

| | 图书名称 | 图书类别 | 作者 | 出版社 |
|---|-------------------|------------|-----|---------|
| 1 | SQLServer2005编程基础 | SQL Server | 陈克力 | 清华大学出版社 |

图 5-15 查询视图 vw_BookInfo

注意：视图中使用的列名取决于创建视图时指定的名称，而不是基本表中的列名。如果在创建视图时没有指定列的别名，则视图中使用的列名与基本表中的列名一致。

5.3.3 使用系统存储过程查看视图

1. 查看视图定义

使用系统存储过程 sp_help 可以查看视图的定义，其基本语法格式如下：

```
sp_help <视图名>
```

【例 5-7】查看视图 vw_BookInfo 的定义。

```
sp_help vw_BookInfo
```

运行结果如图 5-16 所示。

| Name | Owner | Type | Created_datetime | |
|------|-------------|------|------------------|-------------------------|
| 1 | vw_BookInfo | dbo | view | 2009-03-19 23:41:18.327 |

| Column_name | Type | Computed | Length | Prec | Scale | Nullable | TrimTrailingBlanks | FixedLenNullInSource | Collation |
|-------------|------|----------|--------|------|-------|----------|--------------------|----------------------|-------------------|
| 1 | 图书名称 | varchar | no | 50 | | no | no | no | Chinese_PRC_CI_AS |
| 2 | 图书类别 | varchar | no | 20 | | no | no | no | Chinese_PRC_CI_AS |
| 3 | 作者 | varchar | no | 20 | | no | no | no | Chinese_PRC_CI_AS |
| 4 | 出版社 | varchar | no | 30 | | no | no | no | Chinese_PRC_CI_AS |

| Identity | Seed | Increment | Not For Replication | |
|----------|-----------------------------|-----------|---------------------|------|
| 1 | No identity column defined. | NULL | NULL | NULL |

| RowGuidCol | |
|------------|-------------------------------|
| 1 | No rowguidcol column defined. |

图 5-16 查看视图 vw_BookInfo 的定义

2. 查看视图的定义文本

使用系统存储过程 sp_helptext 可以查看视图的定义文本，其基本语法格式如下：

```
sp_helptext <视图名>
```

【例 5-8】查看视图 vw_BookInfo 的定义文本。

```
sp_helptext vw_BookInfo
```

运行结果如图 5-17 所示。

| | Text |
|---|---|
| 1 | CREATE VIEW dbo.vw_BookInfo |
| 2 | AS |
| 3 | SELECT dbo.BookInfo.B_Name AS 图书名称, dbo.Book... |
| 4 | FROM dbo.BookInfo INNER JOIN |
| 5 | dbo.BookType ON dbo.BookInfo.BT_ID = dbo.... |
| 6 | WHERE (dbo.BookType.BT_Name = 'SQL Server') |

图 5-17 查看视图 vw_BookInfo 的定义文本

当出于安全考虑要求视图定义对于用户不可见时，可以在定义视图时使用加密语句。下面将修改视图 vw_BookInfo，加上加密设置，对应语句如下：

```
USE BookStore
GO
ALTER VIEW vw_BookInfo
WITH ENCRYPTION
AS
SELECT BI.B_Name, BT.BT_Name, BI.B_Author, BI.B_Publisher
FROM BookInfo BI,BookType BT
WHERE BI.BT_ID =BT.BT_ID AND BT.BT_Name = 'SQL Server'
```

上述命令成功执行后，再次使用 sp_helptext 命令查看视图 vw_BookInfo 的定义文本时，系统会提示“对象'vw_BookInfo' 的文本已加密”。同时，在“对象资源管理器”中，视图 vw_BookInfo 所对应的图标上增加了锁标记，表示该视图被加密，不能被修改。

3. 查看视图依赖关系

使用系统存储过程 `sp_depends` 可以查看视图的依赖关系，其基本语法格式如下：

```
sp_depends <视图名>
```

【例 5-9】查看视图 `vw_BookInfo` 的依赖关系。

```
sp_depends vw_BookInfo
```

运行结果如图 5-18 所示。

| | name | type | updated | selected | column |
|---|--------------|------------|---------|----------|------------|
| 1 | dbo.BookType | user table | no | yes | BT_ID |
| 2 | dbo.BookType | user table | no | yes | BT_Name |
| 3 | dbo.BookInfo | user table | no | yes | B_Name |
| 4 | dbo.BookInfo | user table | no | yes | BT_ID |
| 5 | dbo.BookInfo | user table | no | yes | B_Author |
| 6 | dbo.BookInfo | user table | no | yes | B_Publi... |

图 5-18 查看视图 `vw_BookInfo` 的依赖关系

5.4 修改视图

创建视图后，可以根据需要修改视图，包括修改视图的定义、修改视图的名称等。

5.4.1 使用 SSMS 修改视图

1. 修改视图定义

【例 5-10】修改数据库中视图 `vw_BookDetail`，使其在原有列的基础上再增加两个列：会员价格 (`B_SalePrice`) 和库存数量 (`B_Quality`)。

(1) 启动 SQL Server Management Studio，在“对象资源管理器”中展开“数据库”节点，然后展开 `BookStore` 数据库节点，再展开“视图”节点。

(2) 右击 `vw_BookDetail` 视图，选择“修改”。

(3) 在视图窗口中增加 `B_SalePrice` 列和 `B_Quality` 列，并分别设置列别名为会员价格和库存数量，如图 5-19 所示。

(4) 修改完成后，保存修改后的视图定义。

提示：修改视图实际上就是修改对应的 `SELECT` 语句。

2. 重命名视图

【例 5-11】将数据库中视图 `vw_OrderBook` 的名称修改为 `vw_BookOrder`。

(1) 启动 SQL Server Management Studio，在“对象资源管理器”中展开“数据库”节点，然后展开 `BookStore` 数据库节点，再展开“视图”节点。

(2) 右击 `vw_OrderBook` 视图，选择“重命名”命令，或者在选定的视图名称上单击，进入视图名称编辑状态。

(3) 进入编辑状态后，输入新的视图名，然后在空白处单击鼠标，即可完成视图名称的修改。

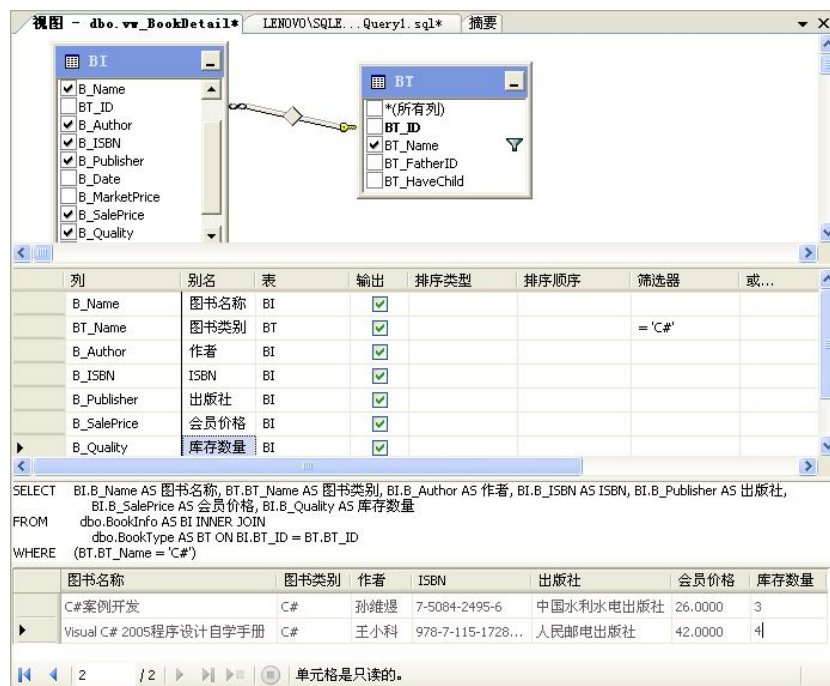


图 5-19 修改视图

5.4.2 使用 T-SQL 语句修改视图

1. 修改视图定义

使用 T-SQL 命令 ALTER VIEW 可以修改视图，ALTER VIEW 的语法格式与 CREATE VIEW 的语法格式类似，其基本语法格式如下：

```

ALTER VIEW <视图名>
[ WITH <视图属性> ]
AS
<SELECT 语句>
[ WITH CHECK OPTION ]

```

【例 5-12】修改视图 vw_BookDetail，使其只包含 B_Name，BT_Name，B_Author， B_ISBN 和 B_Publisher 这 5 列。

```

ALTER VIEW vw_BookDetail
AS
SELECT BI.B_Name AS 图书名称, BT.BT_Name AS 图书类别, BI.B_Author AS 作者,
       BI.B_ISBN AS ISBN, BI.B_Publisher AS 出版社
FROM   BookInfo AS BI, BookType AS BT
WHERE  BI.BT_ID = BT.BT_ID AND BT.BT_Name = 'C#'

```

2. 重命名视图

使用系统存储过程 sp_rename 可以重命名视图，其基本语法格式如下：

```

sp_rename <旧的视图名>, <新的视图名>

```

【例 5-13】将数据库中视图 vw_BookOrder 的名称修改为 vw_OrderBook。

```
sp_rename vw_BookOrder,vw_OrderBook
```

5.5 删除视图

删除视图只是删除了视图的定义和指派给它的所有权限，并不会对基本表造成影响。如果删除的视图是另一个视图的基视图，则当删除该视图时，系统会给出错误提示。

5.5.1 使用 SSMS 删除视图

【例 5-14】删除视图 vw_UserBook。

(1) 启动 SQL Server Management Studio，在“对象资源管理器”中展开“数据库”节点，然后展开 BookStore 数据库节点，再展开“视图”节点。

(2) 右击 vw_UserBook 视图，选择“删除”。

(3) 打开“删除对象”对话框，单击“确定”按钮，删除视图，如图 5-20 所示。

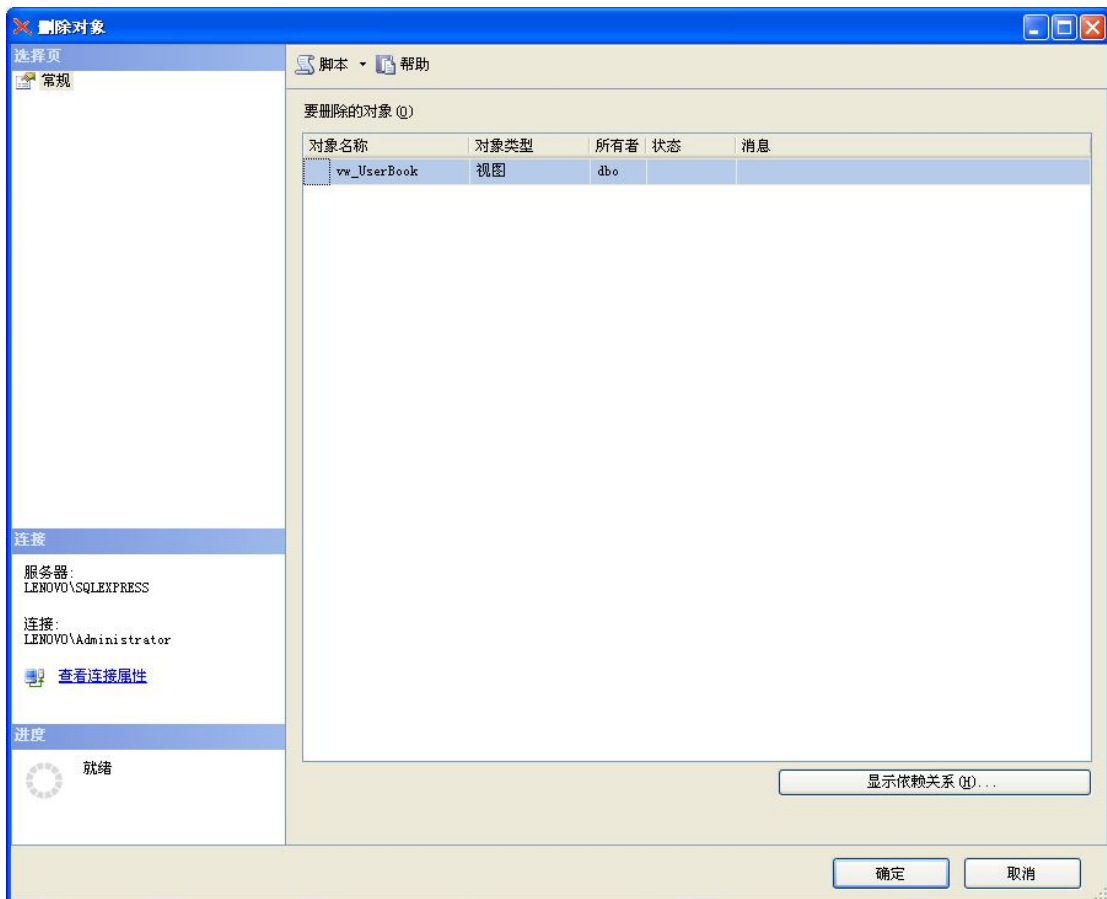


图 5-20 “删除对象”对话框

5.5.2 使用 T-SQL 语句删除视图

删除视图使用 **DROP VIEW** 命令，可以同时删除多个视图，只需要将被删除的视图名之间以逗号隔开。**DROP VIEW** 的基本语法格式如下：

```
DROP VIEW <视图名>
```

【例 5-15】删除视图 vw_BookInfo。

```
DROP VIEW vw_BookInfo
```

5.6 通过视图修改基本表数据

在 SQL Server 中不仅可以通过视图查看基本表中的数据，还可以通过视图向基本表中插入、更新及删除数据。通过视图修改基本表数据时，需要注意以下几点：

- (1) 修改视图中的数据时，一次不能修改多个视图基本表。
- (2) 对视图中的所有列的修改必须遵守视图基本表中所定义的各种数据完整性约束，要符合列的空值属性、约束、标识属性、默认值等条件限制。
- (3) 不允许对视图中的计算列进行修改，也不运行对视图定义中所包含有统计函数或 **GROUP BY** 子句的视图进行插入或修改操作。
- (4) 如果视图引用多个表时，无法用 **DELETE** 命令删除数据。

通过视图修改基本表中数据有两种方法：一是使用 SQL Server Management Studio 来修改视图中的数据，操作方法同表中数据操作类似；二是利用 T-SQL 语句实现视图中数据的修改，这是本节的所要讲述的内容。

5.6.1 通过视图向基本表中插入数据

【例 5-16】通过视图 vw_Users 向会员信息表 (Users 表) 中添加一个会员。

- (1) 创建视图 vw_Users，使其包含 U_ID、U_Name、U_RealName 和 U_Pwd 这 4 个列。

```
USE BookStore
GO
CREATE VIEW vw_Users
AS
SELECT U_ID,U_Name,U_RealName,U_Pwd
FROM Users
```

- (2) 通过视图 vw_Users 添加记录。

```
INSERT INTO vw_Users
VALUES (9, 'lele', '乐乐', '123')
GO
SELECT * FROM Users
```

上述语句成功执行后，将会看到在 Users 表中新增了一个名为 lele 的会员，如图 5-21 所示。

| | U_ID | U_Name | U_RealName | U_Pwd | U_Sex | U_Phone | U_Email | U_Address | U_PostCode |
|---|------|---------|------------|--------|-------|--------------|------------------|-----------|------------|
| 1 | 1 | linli | 林丽 | 123456 | 女 | 13064276022 | linli@163.com | 和平大街12号 | 110001 |
| 2 | 2 | lijuan | 李娟 | 123456 | 女 | 13940403838 | lijuan@sina.com | 陵东街67号 | 110032 |
| 3 | 3 | wangxy | 王晓云 | 123456 | 女 | 024-25456368 | wangxy@126.com | 崇山西路12号 | 110032 |
| 4 | 4 | yanglw | 杨利伟 | 123456 | 男 | 024-83630204 | yanglw@163.com | 青年大街3号 | 110001 |
| 5 | 5 | zhangyp | 张一平 | 123456 | 男 | 13848359090 | zhangyp@sohu.com | 贵和街52号 | 110023 |
| 6 | 6 | zhaohe | 赵贺 | 123456 | 男 | 13345588666 | zhaohe@sina.com | 黄河北大街1号 | 110034 |
| 7 | 7 | zhangh | 张红 | 123456 | 女 | NULL | zhangh@163.com | 黄河北大街3号 | 110034 |
| 8 | 8 | xiayu | 夏雨 | 123456 | 男 | 15734345688 | xiayu@126.com | 贵和街5号 | 110023 |
| 9 | 9 | lele | 乐乐 | 123 | NULL | NULL | NULL | NULL | NULL |

图 5-21 通过视图向基本表中插入数据

5.6.2 通过视图更新基本表中的数据

【例 5-17】通过视图 vw_Users 修改会员信息表 (Users 表) 中会员名为 lele 的会员信息, 将其密码修改为 456。

```
UPDATE vw_Users
SET U_Pwd='456'
WHERE U_Name='lele'
GO
SELECT * FROM Users
```

上述语句成功执行后, 将会看到 Users 表中会员 lele 的密码已经改为 456, 如图 5-22 所示。

| | U_ID | U_Name | U_RealName | U_Pwd | U_Sex | U_Phone | U_Email | U_Address | U_PostCode |
|---|------|---------|------------|--------|-------|--------------|------------------|-----------|------------|
| 1 | 1 | linli | 林丽 | 123456 | 女 | 13064276022 | linli@163.com | 和平大街12号 | 110001 |
| 2 | 2 | lijuan | 李娟 | 123456 | 女 | 13940403838 | lijuan@sina.com | 陵东街67号 | 110032 |
| 3 | 3 | wangxy | 王晓云 | 123456 | 女 | 024-25456368 | wangxy@126.com | 崇山西路12号 | 110032 |
| 4 | 4 | yanglw | 杨利伟 | 123456 | 男 | 024-83630204 | yanglw@163.com | 青年大街3号 | 110001 |
| 5 | 5 | zhangyp | 张一平 | 123456 | 男 | 13848359090 | zhangyp@sohu.com | 贵和街52号 | 110023 |
| 6 | 6 | zhaohe | 赵贺 | 123456 | 男 | 13345588666 | zhaohe@sina.com | 黄河北大街1号 | 110034 |
| 7 | 7 | zhangh | 张红 | 123456 | 女 | NULL | zhangh@163.com | 黄河北大街3号 | 110034 |
| 8 | 8 | xiayu | 夏雨 | 123456 | 男 | 15734345688 | xiayu@126.com | 贵和街5号 | 110023 |
| 9 | 9 | lele | 乐乐 | 456 | NULL | NULL | NULL | NULL | NULL |

图 5-22 通过视图更新基本表中的数据

【例 5-18】通过视图 vw_OrderStatus 将订单编号为 4 的会员的真实姓名修改为“李红”, 同时修改订单状态为 1。

- (1) 创建视图 vw_OrderStatus, 使其包含 O_ID, U_RealName 和 O_Status 这 3 列。

```
CREATE VIEW vw_OrderStatus
AS
SELECT O.O_ID,U.U_RealName,O.O_Status
FROM Users AS U,Orders AS O
WHERE U.U_ID=O.U_ID
```

- (2) 通过视图 vw_OrderStatus 修改数据。

```
UPDATE vw_OrderStatus1
SET U_RealName='李红',O_Status=1
WHERE O_ID=4
```

执行上述 SQL 语句, 系统报告错误: “视图或函数'vw_OrderStatus' 不可更新, 因为修改会影响多个基表”。

修改上述 SQL 语句如下:

```
UPDATE vw_OrderStatus
SET U_RealName='李红'
WHERE O_ID=4
GO
UPDATE vw_OrderStatus
SET O_Status=1
WHERE O_ID=4
```

此时 SQL 语句可以正确执行。

5.6.3 通过视图删除基本表中的数据

【例 5-19】通过视图 vw_Users 删除会员信息表 (Users 表) 中会员名为 lele 的会员信息。

```
DELETE vw_Users
WHERE U_Name='lele'
```

本章小结

视图是一个由 SELECT 语句指定,用以检索数据库表中某些行或列数据的语句存储定义。从本质上说,视图其实是一种 SQL 查询。但对于数据库用户来说,视图似乎是一个真实的表。当 SQL Server 处理视图的查询操作时,它会在数据库中找到该视图的定义,然后把对视图的查询转化为对视图基本表的等价查询,并且执行这个等价查询。通过这种方法进行数据查询,既可以保持基本表的完整性,又可以保持视图的方便性。

本章介绍了视图的相关知识,首先介绍了视图的概念、优点和分类,然后通过典型实例介绍了如何创建、查看、修改和删除视图,最后介绍了如何通过视图插入、更新、删除基本表中的数据

习题五

一、填空题

1. 视图是从一个或多个表或视图中导出的_____。
2. 在 SQL Server 2005 中,可以把视图分成 3 中类型,即_____、索引视图和_____。
3. 查看视图的基本信息可以使用系统存储过程_____,查看视图的定义文本可以使用系统存储过程_____。

二、选择题

1. 以下关于视图的描述,错误的是()。
A. 视图是从一个或多个基表或视图中导出的虚表
B. 视图并不实际存储数据,只在数据库中保存其逻辑定义

- C. 视图里面的任何数据不可以进行修改
 D. SQL 中的 SELECT 语句可以像查询基表一样对视图进行查询
2. 如果需要加密视图的定义文本, 可以使用下面的子句 ()。
- A. WITH CHECK OPTION B. WITH ENCRYPTION
 C. WITH NOCHECK D. WITH SCHEMABINDING
3. 下列 () 系统存储过程可以用于显示数据库对象所依赖的对象。
- A. sp_depends B. sp_help
 C. sp_helptext D. sp_depends

三、操作题

1. 从会员信息表 (Users 表)、图书信息表 (BookInfo 表)、订单信息表 (Orders 表) 和订单详情表 (OrderDetails 表) 中创建“李娟”订购图书的视图 vw_UserBook, 显示结果如图 5-23 所示。



| 订单编号 | 会员名 | 图书名 |
|------|-----|-------------------------|
| 6 | 李娟 | ASP.NET2.0完全自学手册 |
| 7 | 李娟 | ASP.NET+SQLServer典型网站建设 |

图 5-23 显示结果

2. 将视图 vw_UserBook 中订单编号为 7 的图书名改为“ASP.NET2.0+SQLSERVER2005 典型网站设计”。

3. 从图书信息表 (BookInfo 表) 和图书类别表 (BookType 表) 中创建 C#类图书详情的视图 vw_BookDetail, 并查询该视图中的数据, 显示结果如图 5-24 所示。

| | 图书名称 | 图书类别 | 作者 | ISBN | 出版社 |
|---|------------------------|------|-----|-------------------|-----------|
| 1 | C#案例开发 | C# | 孙维煜 | 7-5084-2495-6 | 中国水利水电出版社 |
| 2 | Visual C# 2005程序设计自学手册 | C# | 王小科 | 978-7-115-17289-1 | 人民邮电出版社 |

图 5-24 显示结果

4. 在视图 vw_BookDetail 中查询“中国水利水电出版社”出版的图书信息。