

表的基本操作

实验 3

表(Table)是包含数据库中所有数据的数据库对象。数据在表中的组织方式与在电子表格中相似,都是按行和列的格式组织的。每一行代表一条记录,每一列代表记录中的一个字段。

【知识要点】

1. 创建表的命令格式

```
CREATE TABLE
    [ database_name . [ schema_name ] . | schema_name . ] table_name
    ( { <column_definition> | <computed_column_definition>
      | <column_set_definition> }
      [ <table_constraint> ] [ ,...n ] )
    [ ON { partition_scheme_name ( partition_column_name ) | filegroup
      | "default" } ]
    [ { TEXTIMAGE_ON { filegroup | "default" } } ]
    [ FILESTREAM_ON { partition_scheme_name | filegroup
      | "default" } ]
    [ WITH ( <table_option> [ ,...n ] ) ]
[ ; ]
<column_definition> ::=
column_name <data_type>
    [ FILESTREAM ]
    [ COLLATE collation_name ]
    [ NULL | NOT NULL ]
    [
        [ CONSTRAINT constraint_name ] DEFAULT constant_expression ]
    [ IDENTITY [ ( seed , increment ) ] [ NOT FOR REPLICATION ]
    ]
    [ ROWGUIDCOL ] [ <column_constraint> [ ...n ] ]
    [ SPARSE ]
<data type> ::=
[ type_schema_name . ] type_name
    [ ( precision [ , scale ] | max |
      [ { CONTENT | DOCUMENT } ] xml_schema_collection ) ]
```

```

<column_constraint> ::=

[ CONSTRAINT constraint_name ]
{ { PRIMARY KEY | UNIQUE }
  [ CLUSTERED | NONCLUSTERED ]
  [
    WITH FILLFACTOR = fillfactor
    | WITH ( < index_option > [ , ...n ] )
  ]
  [ ON { partition_scheme_name ( partition_column_name )
    | filegroup | "default" } ]
| [ FOREIGN KEY ]
  REFERENCES [ schema_name . ] referenced_table_name [ ( ref_column ) ]
  [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
  [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
  [ NOT FOR REPLICATION ]
| CHECK [ NOT FOR REPLICATION ] ( logical_expression )
}

```

2. 数据表复制的命令格式

```

SELECT select_list [ INTO new_table ]
[ FROM table_source ] [ WHERE search_condition ]
[ GROUP BY group_by_expression ]
[ HAVING search_condition ]
[ ORDER BY order_expression [ ASC | DESC ] ]

```

3. 修改表结构的命令格式

```

ALTER TABLE [ database_name . [ schema_name ] . | schema_name . ] table_name
{
  ALTER COLUMN column_name
  {
    [ type_schema_name . ] type_name [ ( ( { precision [ , scale ]
      | max | xml_schema_collection } ) ) ]
    [ COLLATE collation_name ]
    [ NULL | NOT NULL ]
  | {ADD | DROP }
    { ROWGUIDCOL | PERSISTED | NOT FOR REPLICATION | SPARSE }
  }
  | [ WITH { CHECK | NOCHECK } ]
  | ADD
  {
    <column_definition>
    | <computed_column_definition>
    | <table_constraint>
    | <column_set_definition>
  } [ , ...n ]
  | DROP
  {
    [ CONSTRAINT ] constraint_name

```

```

    [ WITH ( <drop_clustered_constraint_option> [ ,...n ] ) ]
    | COLUMN column_name

} [ ,...n ]
| [ WITH { CHECK | NOCHECK } ] { CHECK | NOCHECK } CONSTRAINT
    { ALL | constraint_name [ ,...n ] }
| { ENABLE | DISABLE } TRIGGER
    { ALL | trigger_name [ ,...n ] }
| { ENABLE | DISABLE } CHANGE_TRACKING
    [ WITH ( TRACK_COLUMNS_UPDATED = { ON | OFF } ) ]
| SWITCH [ PARTITION source_partition_number_expression ]
    TO target_table
    [ PARTITION target_partition_number_expression ]
| SET ( FILESTREAM_ON = { partition_scheme_name | filegroup |
    "default" | "NULL" } )
| REBUILD
    [ [ PARTITION = ALL ]
    [ WITH ( <rebuild_option> [ ,...n ] ) ]
    | [ PARTITION = partition_number
    [ WITH ( <single_partition_rebuild_option> [ ,...n ] ) ]
    ]
    ]
| (<table_option>)
}
[ ; ]

```

4. 删除表的命令格式

```

DROP TABLE [ database_name . [ schema_name ] . | schema_name . ]
    table_name [ ,...n ] [ ; ]

```

【实验目的】

- (1) 掌握表的创建和删除方法。
- (2) 掌握表结构的修改方法。
- (3) 掌握复制表结构的方法。

实验 3.1 创建表

【实验目的】

- (1) 掌握使用 SQL Server Management Studio 的图形界面创建表。
- (2) 掌握使用 T-SQL 创建表。
- (3) 掌握使用 T-SQL 复制表结构。

【实验内容】

- (1) 使用 SQL Server Management Studio 的图形界面在 Sales 数据库中创建

customers 表和 products 表。

(2) 使用 T-SQL 在 Sales 数据库中创建 agents 表和 order 表。

(3) 使用 T-SQL 语句复制 Sales 数据库中的 orders 表结构创建数据表 test1, 复制 agents 表结构创建数据表 test2。

说明: 数据库 Sales 包含 4 个关系表, 表的结构如附录 A 的图 A.1~图 A.4 所示。

【实验步骤】

1. 使用 SQL Server Management Studio 的图形界面在 Sales 数据库中创建 customers 表和 products 表

(1) 启动 SQL Server Management Studio。执行“开始”→“所有程序”→Microsoft SQL Server 2008→SQL Server Management Studio 命令。显示“连接服务器”对话框, 单击“连接”按钮, 按照默认方式连接服务器。

(2) 在“对象资源管理器”窗格中, 依次展开结点“数据库”→Sales, 右击“表”, 在快捷菜单中选择“新建表”选项, 在查询窗口中显示如图 3.1 所示。

(3) 依次在“列名”栏输入 customers 表的列, 在“数据类型”下拉列表中选择数据类型, 设置是否允许 Null 值, 如图 3.2 所示。



图 3.1 “新建表”对话框

列名	数据类型	允许 Null 值
cid	nvarchar(255)	<input type="checkbox"/>
cname	nvarchar(255)	<input checked="" type="checkbox"/>
city	nvarchar(255)	<input checked="" type="checkbox"/>
discnt	float	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

图 3.2 customers 表的列


(4) 单击工具栏中的  按钮, 显示“选择名称”对话框, 在“输入表名称”栏输入 customers, 单击“确定”按钮, 如图 3.3 所示。表创建完毕之后, 可以在“对象资源管理器”窗格中查看。



图 3.3 “选择名称”对话框

(5) 按照步骤(2)~(4)的顺序创建 products 表。


2. 使用 T-SQL 在 Sales 数据库中创建 agents 表和 order 表

(1) 单击左上角  新建查询(N) 按钮, 打开查询生成器。

(2) 在查询窗口中输入下面的 SQL 语句:

```
use Sales
create table agents (
    aid nvarchar (255) not null,
    aname nvarchar (255),
    city nvarchar (255),
    [percent] float
    primary key (aid))
```

(3) 单击工具栏上“分析”按钮,检查是否有语法错误。

(4) 单击工具栏中的  按钮,执行窗口中的 SQL 语句,结果如图 3.4 所示表示已成功创建 agents 表。可以在“对象资源管理器”窗格中刷新后查看。

(5) 按照步骤(1)~(4)的顺序创建 order 表,其不同的是在步骤(2)中的查询窗口输入下面的 SQL 语句:

```
use Sales
create table orders (
    ordno nvarchar (255) not null,
    month nvarchar (255),
    cid nvarchar (255) not null,
    aid nvarchar (255) not null,
    pid nvarchar (255) not null,
    qty float,
    [dollars] float
    primary key (ordno))
```

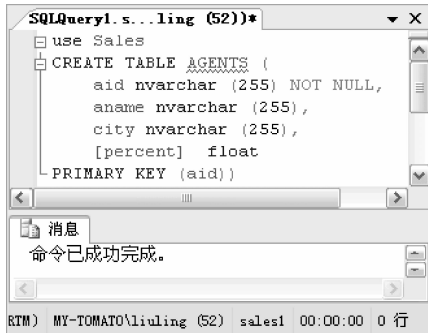


图 3.4 执行结果


3. 使用 T-SQL 语句复制 Sales 数据库中的 orders 表结构创建数据表 test1,复制 agents 表结构创建数据表 test2

(1) 单击左上角  按钮,打开查询生成器。

(2) 在查询窗口中输入下面的 SQL 语句:

```
use Sales
select * into test1 from orders where 1 = 2
```

(3) 单击工具栏上“分析”按钮,检查是否有语法错误。


(4) 单击工具栏中的  按钮,执行窗口中的 SQL 语句,结果显示成功后,在“对象资源管理器”窗格中刷新后查看。比较 test1 和 orders 的表结构是否相同。

(5) 单击左上角  按钮,打开查询生成器。

(6) 在查询窗口中输入下面的 SQL 语句:

```
use Sales
select * into test2 from agents where 1 = 2
```

(7) 单击工具栏上“分析”按钮,检查是否有语法错误。

(8) 单击工具栏中的  按钮,执行窗口中的 SQL 语句,结果显示成功后,在“对象资源管理器”窗格中刷新后查看。比较 test2 和 agents 的表结构是否相同。

实验 3.2 修改表结构

【实验目的】

- (1) 掌握使用 SQL Server Management Studio 的图形界面修改表结构。
- (2) 掌握使用 T-SQL 修改表结构。

【实验内容】

(1) 使用 SQL Server Management Studio 的图形界面向已有数据表 test2 增加电子信箱 E-mail 列：列名,email；数据类型,char；长度,40；允许空否,NOT NULL。

(2) 使用 SQL Server Management Studio 的图形界面修改 test2 中电子信箱 E-mail 列。把 E-mail 列修改成：列名,email；数据类型,char；长度,20；允许空否,NULL。

(3) 使用 SQL Server Management Studio 的图形界面删除 test2 中电子信箱 E-mail 列。

(4) 使用 T-SQL 向已有数据表 test2 增加电子信箱 E-mail 列：列名,email；数据类型,Char；长度,40；允许空否,NOT NULL。

(5) 使用 T-SQL 修改 test2 中电子信箱 E-mail 列。把 E-mail 列修改成：列名,email；数据类型,Char；长度,20；允许空否,NULL。

(6) 使用 T-SQL 删除 test2 中电子信箱 E-mail 列。

【实验步骤】

1. 使用 SQL Server Management Studio 的图形界面向已有数据表 test2 增加电子信箱 E-mail 列：列名,email；数据类型,Char；长度,40；允许空否,NOT NULL

(1) 在“对象资源管理器”窗格中,依次展开结点“数据库”→Sales→“表”,右击 dbo. test2,在快捷菜单中选择“设计”选项,在查询窗口中显示表的结构。

列名	数据类型	允许 Null 值
aid	nvarchar(255)	<input type="checkbox"/>
aname	nvarchar(255)	<input checked="" type="checkbox"/>
city	nvarchar(255)	<input checked="" type="checkbox"/>
[percent]	float	<input checked="" type="checkbox"/>
email	char(40)	<input type="checkbox"/>

图 3.5 对表增加列定义

(2) 在查询窗口的最后一行,添加新列 E-mail 的定义,其中列名,email；数据类型,char；长度,40；不允许 Null 值,如图 3.5 所示。

(3) 单击工具栏中的 按钮保存修改后的 test2 表,如果系统提示如图 3.6 所示,进行步骤(4),否则保存成功,在“对象资源管理器”窗格中刷新后查看 test2 表的列是否已增加 E-mail。



图 3.6 表保存后提示信息

(4) 选择菜单“工具”→“选项”选项,显示“选项”对话框,展开左边结点 Designers,选择“表设计器和数据库设计器”选项,在右边的“表选项”中取消选中“阻止保存要求重新创建表的更改”复选框,如图 3.7 所示。单击“确定”按钮退出,然后重新执行步骤(3)。

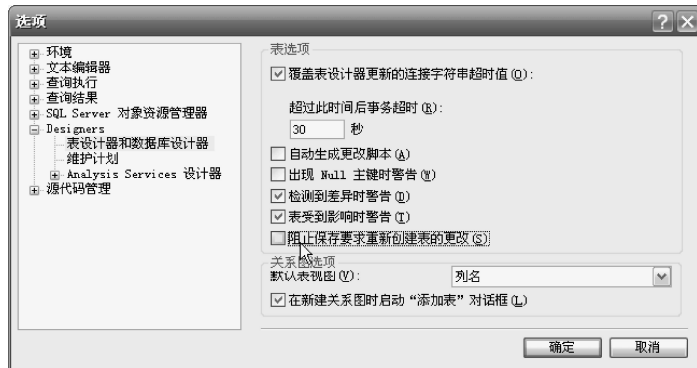



图 3.7 “选项”对话框

2. 使用 SQL Server Management Studio 的图形界面修改 test2 中电子信箱 E-mail 列。把 E-mail 列修改成: 列名, email; 数据类型, char; 长度, 20; 允许空否, NULL

(1) 在“对象资源管理器”窗格中,依次展开结点“数据库”→Sales→“表”,右击 dbo.test2,在快捷菜单中选择“设计”选项,在查询窗口中显示表的结构。

(2) 在查询窗口中,修改列 E-mail 的定义,其中列名, email; 数据类型, char; 长度, 20; 允许空否, NULL,如图 3.8 所示。

(3) 单击工具栏中的  按钮保存修改后的 test2 表,保存成功后,在“对象资源管理器”中刷新后查看 test2 表的 E-mail 列的定义是否已修改。

3. 使用 SQL Server Management Studio 的图形界面删除 test2 中电子信箱 E-mail 列

(1) 在“对象资源管理器”窗格中,依次展开结点“数据库”→Sales→“表”,右击 dbo.test2,在快捷菜单中选择“设计”选项,在查询窗口中显示表的结构。

(2) 在查询窗口中,鼠标指向 email 列,右击,在弹出的快捷菜单中,选择“删除列”选项,如图 3.9 所示。

列名	数据类型	允许 Null 值
aid	nvarchar(255)	<input type="checkbox"/>
aname	nvarchar(255)	<input checked="" type="checkbox"/>
city	nvarchar(255)	<input checked="" type="checkbox"/>
[percent]	float	<input checked="" type="checkbox"/>
email	char(20)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

图 3.8 对表修改列定义

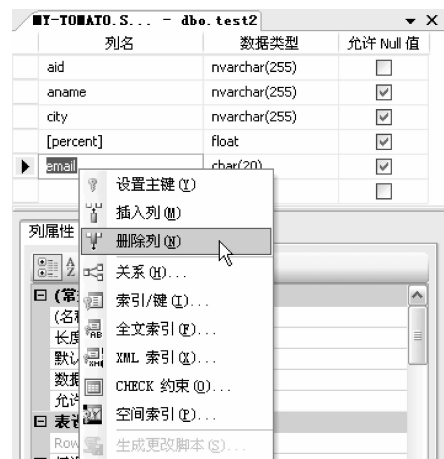



图 3.9 “删除列”菜单

(3) 单击工具栏中的  按钮保存修改后的 test2 表,保存成功后,在“对象资源管理器”中刷新后查看 test2 表的 E-mail 列是否已删除。


4. 使用 T-SQL 向已有数据表 test2 增加电子信箱 E-mail 列: 列名,email; 数据类型,char; 长度,40; 允许空否,NOT NULL

(1) 单击左上角  新建查询 (N) 按钮,打开查询生成器。

(2) 在查询窗口中输入下面的 SQL 语句:

```
use Sales
alter table test2
add email char(40) not null
```

(3) 单击工具栏上“分析”按钮,检查是否有语法错误。

(4) 单击工具栏中的  执行 (E) 按钮,执行窗口中的 SQL 语句,结果显示成功后,在“对象资源管理器”中刷新后查看 test2 表的列是否已增加 E-mail。


5. 使用 T-SQL 修改 test2 中电子信箱 E-mail 列。把 E-mail 列修改成: 列名,email; 数据类型,char; 长度,20; 允许空否,NULL

(1) 单击左上角  新建查询 (N) 按钮,打开查询生成器。

(2) 在查询窗口中输入下面的 SQL 语句:

```
use Sales
alter table test2
alter column email char(20) null
```

(3) 单击工具栏上“分析”按钮,检查是否有语法错误。

(4) 单击工具栏中的  执行 (E) 按钮,执行窗口中的 SQL 语句,结果显示成功后,在“对象资源管理器”中刷新后查看 test2 表的 E-mail 列的定义是否已修改。


6. 使用 T-SQL 删除 test2 中电子信箱 E-mail 列

(1) 单击左上角  新建查询 (N) 按钮,打开查询生成器。

(2) 在查询窗口中输入下面的 SQL 语句:

```
use Sales
alter table test2
drop column email
```

(3) 单击工具栏上“分析”按钮,检查是否有语法错误。

(4) 单击工具栏中的  执行 (E) 按钮,执行窗口中的 SQL 语句,结果显示成功后,在“对象资源管理器”中刷新后查看 test2 表的 E-mail 列是否已删除。

实验 3.3 删除表

【实验目的】

- (1) 掌握使用 SQL Server Management Studio 的图形界面删除表。
- (2) 掌握使用 T-SQL 删除表。

【实验内容】

- (1) 使用 SQL Server Management Studio 的图形界面删除表 test1。
- (2) 使用 T-SQL 删除表 test2。

【实验步骤】

1. 使用 SQL Server Management Studio 的图形界面删除表 test1

(1) 在“对象资源管理器”窗格中,依次展开结点“数据库”→Sales→“表”,右击要删除的表名 test1,在快捷菜单中选择“删除”选项,显示“删除对象”窗口,如图 3.10 所示。

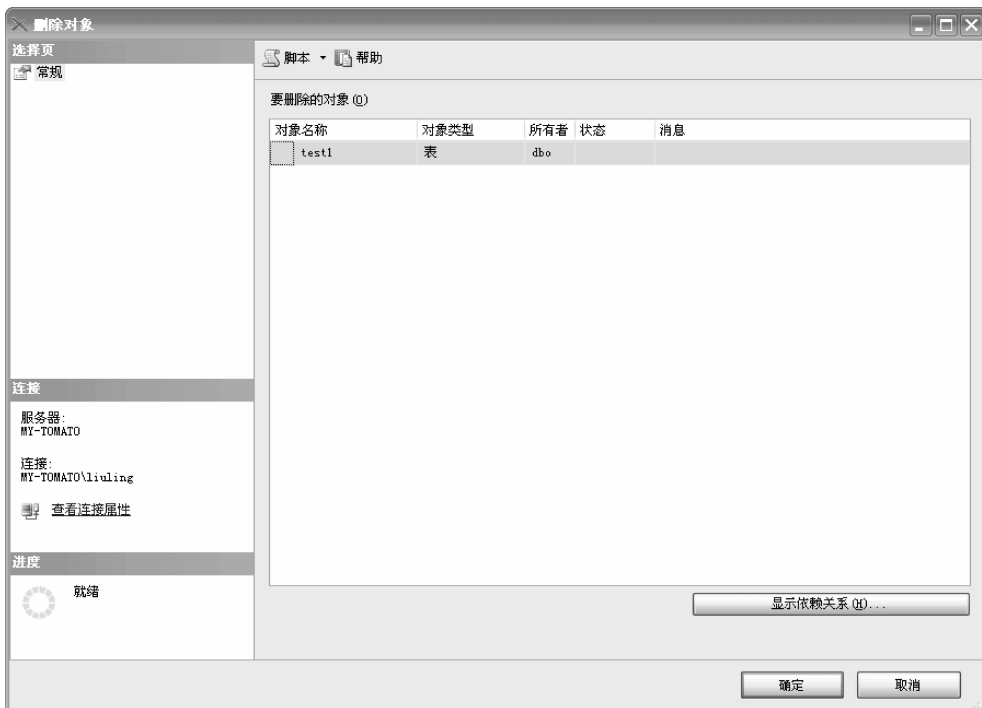


图 3.10 “删除对象”窗口

(2) 在“删除对象”窗口中,选择要删除的对象,单击“确定”按钮,即删除了该表,可以在“对象资源管理器”中查看 test1 表是否存在。

2. 使用 T-SQL 删除表 test2

- (1) 单击左上角 新建查询 (N) 按钮,打开查询生成器。
- (2) 在查询窗口中输入下面的 SQL 语句:

```
use Sales
go
drop table test2
```

- (3) 单击工具栏上“分析”按钮,检查是否有语法错误。
- (4) 单击工具栏中的 执行 (E) 按钮,执行窗口中的 SQL 语句,结果显示成功后,可以在“对象资源管理器”中刷新后查看 test2 是否存在。

习题

针对数据库 Library(表结构和内容如附录 A)进行下面的实验:

1. 使用 SQL Server Management Studio 的图形界面创建读者关系 Reader。
2. 使用 T-SQL 创建图书关系 Book 和借书关系 Borrow。
3. 使用 T-SQL 语句复制 Reader 表结构创建数据表 Readcopy。
4. 使用 T-SQL 向已有读者关系 Reader 增加电子信箱 E-mail 列: 列名, E-mail; 数据类型, char; 长度, 40; 允许空否, NOT NULL。
5. 使用 T-SQL 修改读者关系 Reader 中电子信箱 E-mail 列, 把 E-mail 列修改成下列定义: 列名, E-mail; 数据类型, char; 长度, 20; 允许空否, NULL。
6. 使用 T-SQL 删除读者关系 Reader 中电子信箱 E-mail 列。
7. 使用 T-SQL 删除数据表 Readcopy。