

# 第5章

## 数据管理

本章要点：

- (1) 插入数据
- (2) 更新数据
- (3) 删除数据
- (4) 数据的导入和导出



### 5.1 使用 SSMS 图形界面管理数据

SQL Server 中对数据的管理包括插入、更新和删除。本章以 studentdb 数据库为例来进行讲解。首先介绍使用 SSMS 图形界面管理数据,然后介绍使用 T-SQL 语句管理数据。

#### 1. 使用 SSMS 图形界面向表中添加数据

**任务一：**向部门表 (Department) 中插入一条记录。部门编号为 jsj,部门名称为“计算机应用技术”。

**【步骤 1】** 右击 studentdb 数据库中的 Department 表,如图 5-1 所示。

**【步骤 2】** 选择【编辑前 200 行(E)】,输入 dep\_id 字段的值为 jsj,dep\_name 字段的值为“计算机应用技术”,如图 5-2 所示。

**【步骤 3】** 关闭表编辑窗口,即可实现向 Department 数据表中添加一条记录的任务。

#### 2. 使用 SSMS 图形界面修改表中数据

**任务二：**修改部门表 (Department),将表中部门名称为“计算机应用技术”的部门编号改为 jsj01。

**【步骤 1】** 打开表编辑界面,将 dep\_name 为“计算机应用技术”所在行的部门编号改为 jsj01,如图 5-3 所示。



图 5-1 向部门表 (Department) 中插入一条记录界面 1

JIAXIANGSU-ZJFF...dbo.Department x		
	dep_id	dep_name
⚙	jsj	计算机应用技术
*	NULL	NULL

图 5-2 向部门表(Department)中插入一条记录界面 2

JIAXIANGSU-ZJFF...dbo.Department x		
	dep_id	dep_name
⚙	jsj01	计算机应用技术
*	NULL	NULL

图 5-3 修改部门表中部门名称为“计算机应用技术”的记录界面

**【步骤 2】** 关闭表编辑窗口,即可实现修改数据表中记录的任务。

### 3. 使用 SSMS 图形界面删除表中数据

任务三: 删除部门表(Department)中部门编号为 jsj01 的记录。

**【步骤 1】** 选中部门编号为 jsj01 的一行记录,右击选中的记录,如图 5-4 所示。



图 5-4 删除部门表(Department)中部门编号为 jsj01 的记录

**【步骤 2】** 单击【删除】选项,弹出删除确认对话框,如图 5-5 所示。



图 5-5 删除确认对话框

**【步骤 3】** 单击【是】按钮,即可删除该行记录。

说明: 删除记录也可选中记录,直接按 Delete 键进行删除。

## 5.2 使用 T-SQL 语句插入数据

可以使用 T-SQL 的 INSERT 语句向已经创建好的数据表中添加记录,也可以将现有表中的数据添加到新创建的表中。向已经创建好的数据表中插入记录,可以一次插入一条记录,也可以一次插入多条记录。插入记录时需要注意:插入记录中的值必须符合各字段的数据类型,且插入的字段与值要一一对应。

### 5.2.1 插入单行数据

使用 INSERT 语句插入单行数据的语法格式如下:

```
INSERT [INTO]<表名> [列名列表] VALUES <值列表>
```

上述语法格式中,“[]”表示可选部分。各参数含义说明如下。

- (1) INSERT: 插入数据的关键字。
- (2) INTO: 可选部分,可以省略,加上时可以增强语句的可读性。
- (3) 表名: 指定要向哪个表中插入数据。
- (4) 列名列表: 可选部分。如果列名中有多列,则各列名之间用逗号分隔,而且列名的书写顺序可以由用户自己来定,不一定按照表定义的顺序。如果省略列名,则按照数据表定义的顺序依次插入。
- (5) VALUES: 该关键字后面跟着值列表,指定要插入的数据列表值。
- (6) 值列表: 指定插入各列对应的数值,各值之间用逗号分隔。值列表中的数据要跟列名中的列相对应。

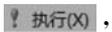
**说明:**

- (1) 插入语句中存在列名列表时,值列表中的数据个数、顺序和数据类型必须与列名列表中的个数、顺序和数据类型一一对应。如果某列暂时无值而此列允许取空值,则可以在列值的相应位置添加 NULL,但是不能省略。
- (2) 若字段不允许为空,且未设置默认值,则必须给该字段设置数据值。
- (3) 不要向标识列中插入数据值。

**任务一:** 向 Department 表中插入一条记录(部门编号为 jsj01,部门名称为“计算机应用技术教研室”)。

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件,在查询编辑器窗口中输入如下 T-SQL 语句:

```
INSERT INTO Department(dep_id,dep_name)
VALUES('jsj01','计算机应用技术教研室')
```

**【步骤 2】** 单击 , 执行语法检查,语法检查通过后,单击 , 执行 T-SQL 命令,如图 5-6 所示。

**【步骤 3】** 打开 Department 表编辑界面,如图 5-7 所示。

**练习:** 向 Department 表中插入两条记录。



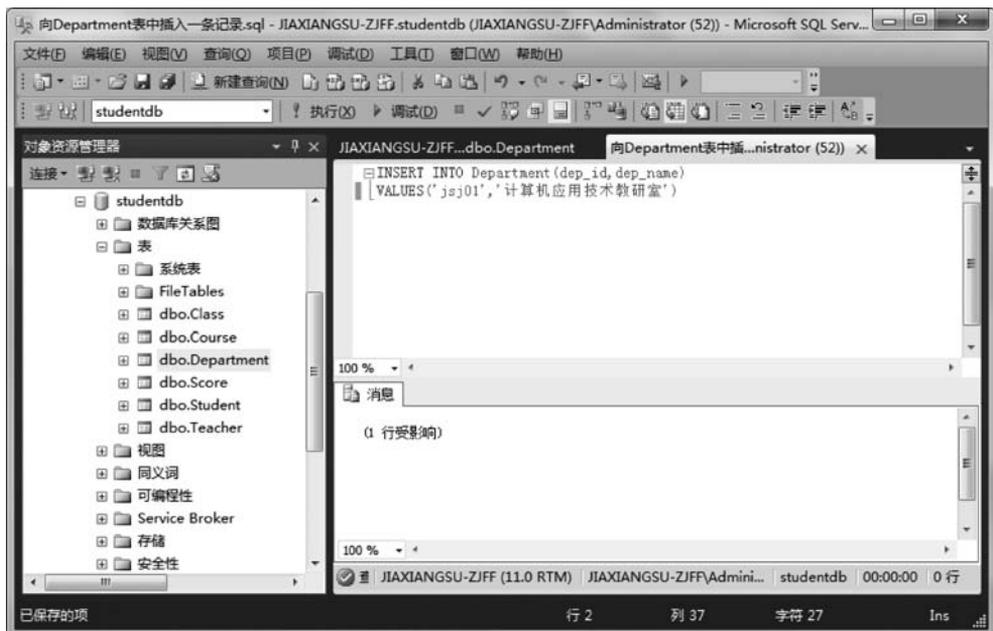


图 5-6 在查询编辑器窗口中执行向 Department 表插入语句的界面

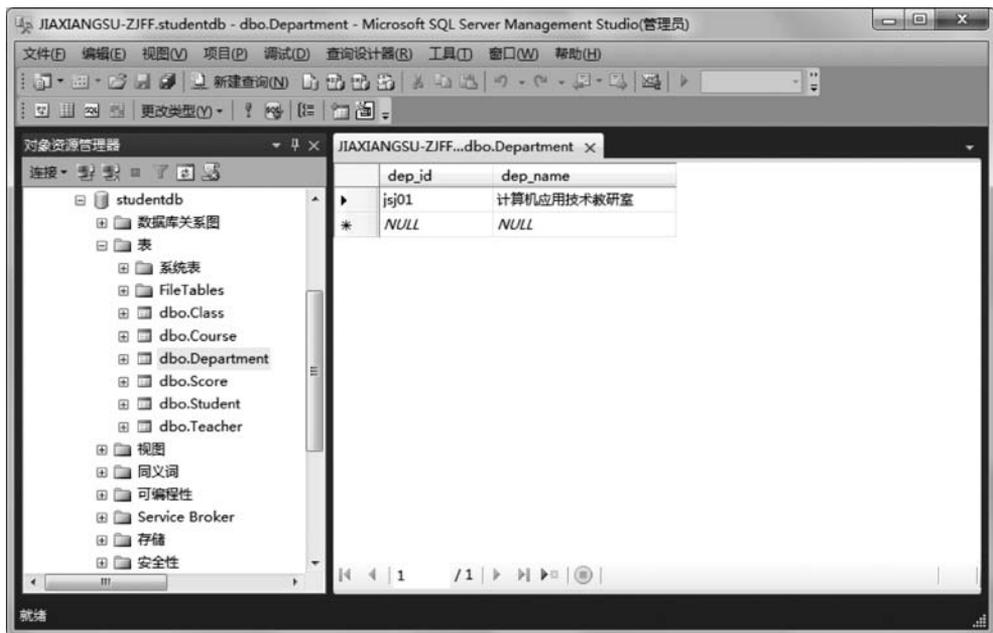


图 5-7 查看 Department 表编辑界面中新插入的一条记录

- (1) 部门编号为 jsj02, 部门名称为“计算机网络技术教研室”。
- (2) 部门编号为 jsj03, 部门名称为“计算机信息管理教研室”。

提示: 可以将两条插入语句放在一个查询编辑器文件中一起执行, 执行成功之后, Department 表编辑界面如图 5-8 所示。

**任务二：**向 Class 表中插入一条记录(班级编号为 2013yy,班级名称为 13 应用,所属专业为计算机应用技术)。

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
INSERT INTO Class(cla_id, cla_name, cla_specialty)
VALUES('2013yy', '13 应用', '计算机应用技术')
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 , 执行 T-SQL 命令, 如图 5-9 所示。

dep_id	dep_name
jsj01	计算机应用技术教研室
jsj02	计算机应用技术教研室
jsj03	计算机信息管理教研室
* NULL	NULL

图 5-8 再次向 Department 表中插入两条记录后的表编辑界面

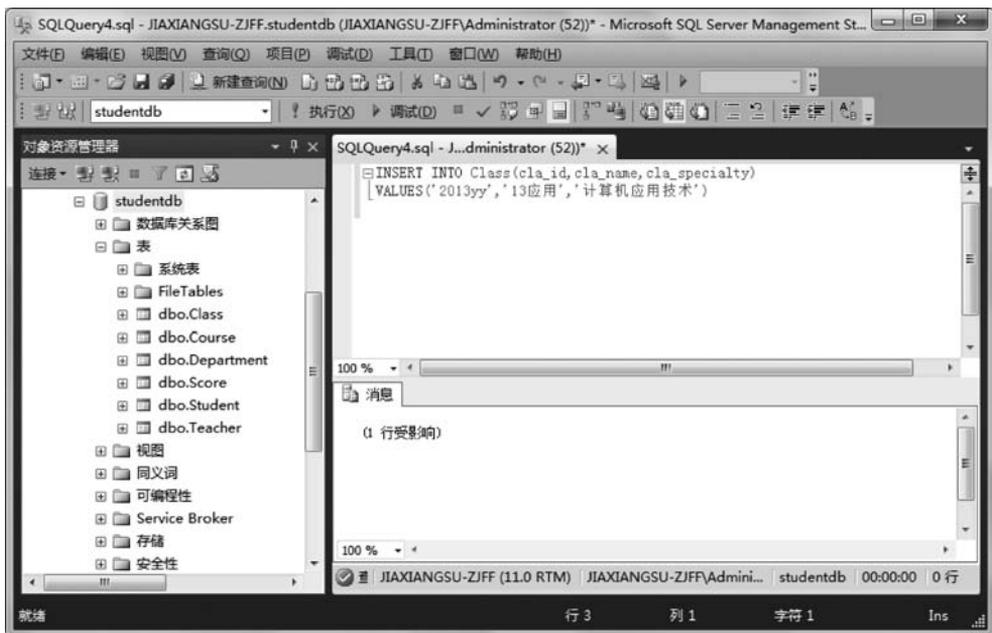


图 5-9 在查询编辑器窗口中执行向 Class 表插入记录语句界面

**【步骤 3】** 打开 Class 表编辑界面, 如图 5-10 所示。

**练习：**向 Class 表中插入两条记录。

(1) 班级编号为 2013wl, 班级名称为 13 网络, 所属专业为计算机网络技术。

(2) 班级编号为 2013xg, 班级名称为 13 信管, 所属专业为计算机信息管理。

此时, Class 表编辑界面如图 5-11 所示。

**任务三：**向 Teacher 表中插入一条记录(教师编号为 200601, 教师姓名为贾祥素, 部门编号为 jsj03)。

T-SQL 语句如下:

```
INSERT INTO Teacher(tea_no, tea_name, tea_departmentid)
VALUES('200601', '贾祥素', 'jsj03')
```

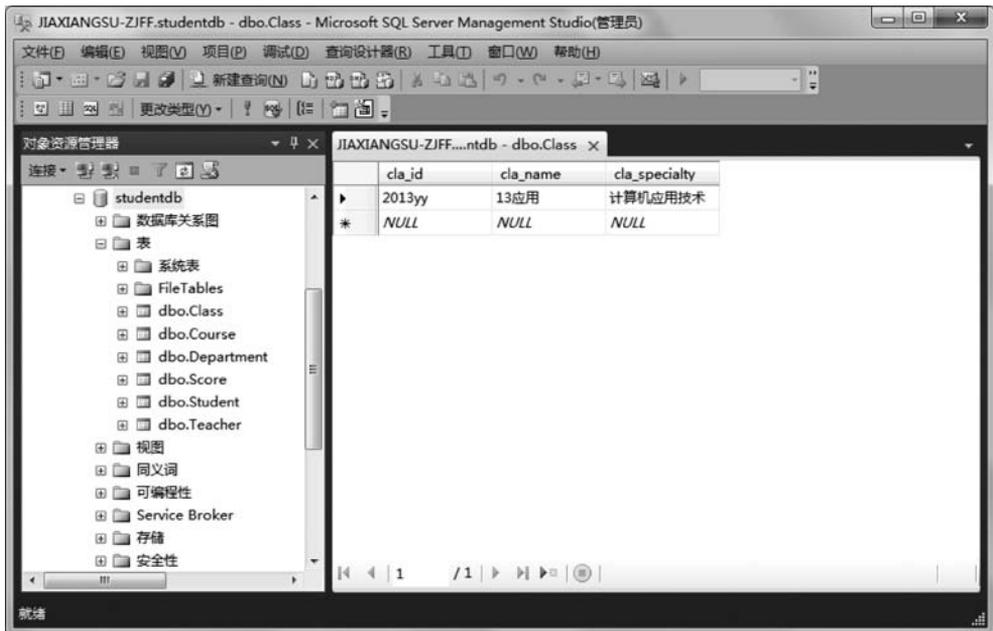


图 5-10 查看 Class 表编辑界面中新插入的一条记录

**练习：**向 Teacher 表中插入两条记录。

- (1) 教师编号为 200602, 教师姓名为李明, 部门编号为 jsj02。
- (2) 教师编号为 200603, 教师姓名为王军, 部门编号为 jsj01。

此时, Teacher 表编辑界面如图 5-12 所示。

cla_id	cla_name	cla_specialty
2013wl	13网络	计算机网络技术
2013xg	13信管	计算机信息管理
2013yy	13应用	计算机应用技术
NULL	NULL	NULL

图 5-11 再次向 Class 表中插入两条记录后的表编辑界面

tea_no	tea_name	tea_departm...
200601	贾祥素	jsj03
200602	李明	jsj02
200603	王军	jsj01
NULL	NULL	NULL

图 5-12 再次向 Teacher 表中插入两条记录后的表编辑界面

**任务四：**向 Course 表中插入一条记录(课程编号为 888001, 课程名称为 SQL Server 管理和查询, 学分为 3, 任课教师编号为 200601)。

T-SQL 语句如下：

```
INSERT INTO Course(cou_id, cou_name, cou_credit, cou_teano)
VALUES('888001', 'SQL Server 管理和查询', 3, '200601')
```

**练习：**向 Course 表中插入两条记录。

- (1) 课程编号为 888002, 课程名称为计算机专业英语, 学分为 2, 任课教师编号为 200601。
- (2) 课程编号为 888003, 课程名称为网页设计, 学分为 2, 任课教师编号为 200601。

此时, Course 表编辑界面如图 5-13 所示。



cou_id	cou_name	cou_credit	cou_teano
888001	SQL Server管理和查询	3	200601
888002	计算机专业英语	2	200601
888003	网页设计	2	200601
* NULL	NULL	NULL	NULL

图 5-13 再次向 Course 表中插入两条记录后的表编辑界面

**任务五：**向 Student 表中插入一条记录(学号为 20130101, 学生姓名为王伟, 性别为男, 籍贯为浙江杭州, 电子邮箱为 wangwei@163.com, 手机号码为 13277776666, 班级编号为 2013yy)。

T-SQL 语句如下：

```
INSERT INTO
Student(stu_no, stu_name, stu_sex, stu_native, stu_email, stu_phone, stu_classid)
VALUES('20130101', '王伟', '男', '浙江杭州', 'wangwei@163.com', '13277776666', '2013yy')
```

此时, Student 表编辑界面如图 5-14 所示。



stu_no	stu_name	stu_sex	stu_native	stu_email	stu_phone	stu_classid
20130101	王伟	男	浙江杭州	wangwei@16...	13277776666	2013yy
* NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 5-14 向 Student 表中插入第 1 条记录后的表编辑界面

**任务五扩展 1：**向 Student 表中插入一条记录(学号为 20130102, 学生姓名为张静, 性别为女, 籍贯为浙江宁波, 电子邮箱为 zhangjing@163.com, 手机号码为 13616715925, 班级编号为 2013yy)。

要求：在进行插入时省略列名列表。

T-SQL 语句如下：

```
INSERT INTO Student
VALUES('20130102', '张静', '女', '浙江宁波', 'zhangjing@163.com', '13616715925', '2013yy')
```

此时, Student 表编辑界面如图 5-15 所示。



stu_no	stu_name	stu_sex	stu_native	stu_email	stu_phone	stu_classid
20130101	王伟	男	浙江杭州	wangwei@16...	13277776666	2013yy
20130102	张静	女	浙江宁波	zhangjing@16...	13616715925	2013yy
* NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 5-15 向 Student 表中插入第 2 条记录后的表编辑界面

**分析：**

在向数据表中插入记录时可以省略列名列表, 但是一定要注意值列表中要按照数据表定义的顺序依次插入。

**任务五扩展 2：**向 Student 表中插入一条记录(学号为 20130201, 学生姓名为李超, 班级

编号为 2013w1)。

要求：在进行插入时只给部分字段赋值。

T-SQL 语句如下：

```
INSERT INTO Student(stu_no,stu_name,stu_classid)
VALUES('20130201','李超','2013w1')
```

此时,Student 表编辑界面如图 5-16 所示。

	stu_no	stu_name	stu_sex	stu_native	stu_email	stu_phone	stu_classid
	20130101	王伟	男	浙江杭州	wangwei@16...	13277776666	2013yy
	20130102	张静	女	浙江宁波	zhangjing@16...	13616715925	2013yy
▶	20130201	李超	男	NULL	NULL	NULL	2013w1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 5-16 向 Student 表中插入第 3 条记录后的表编辑界面

分析：

在向数据表中插入记录时可以只给部分列赋值,但是非空字段一定要赋值。任务五扩展 2 中没有给性别赋值,但是在表编辑界面中性别一列对应值为“男”,是因为 stu\_sex 字段添加了默认约束。

任务五扩展 2 的 T-SQL 语句等价于如下：

```
INSERT INTO Student(stu_no,stu_name,stu_classid,stu_sex)
VALUES('20130201','李超','2013w1',DEFAULT)
```

使用关键字 DEFAULT 代替插入的数值,这样就可以给具有默认值的列插入数据。

**任务六：**向 Score 表中插入一条记录(学生学号为 20130101,课程编号为 888001,平时成绩为 80,期末成绩为 90,总评成绩为 85)。

T-SQL 语句如下：

```
INSERT INTO Score(sco_stuno,sco_courseid,sco_usual,sco_final,sco_overall)
VALUES('20130101','888001',80,90,85)
```

此时,Score 表编辑界面如图 5-17 所示。

	sco_id	sco_stuno	sco_courseid	sco_usual	sco_final	sco_overall
▶	1	20130101	888001	80	90	85
*	NULL	NULL	NULL	NULL	NULL	NULL

图 5-17 向 Score 表中插入第 1 条记录后的表编辑界面

分析：

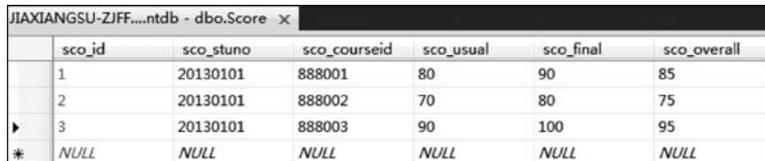
不要给标识列赋值。任务六中没有给成绩编号(sco\_id)赋值,但是在表编辑界面中该列却是有值存在的。成绩编号(sco\_id)的值是系统自动给赋值的。

**练习：**向 Score 表中插入两条记录。

(1) 学生学号为 20130101,课程编号为 888002,平时成绩为 70,期末成绩为 80,总评成绩为 75。

(2) 学生学号为 20130101,课程编号为 888003,平时成绩为 90,期末成绩为 100,总评成绩为 95。

此时,Score 表编辑界面如图 5-18 所示。



	sco_id	sco_stuno	sco_courseid	sco_usual	sco_final	sco_overall
	1	20130101	888001	80	90	85
	2	20130101	888002	70	80	75
▶	3	20130101	888003	90	100	95
*	NULL	NULL	NULL	NULL	NULL	NULL

图 5-18 再次向 Score 表中插入两条记录后的表编辑界面

常见错误汇总:

(1) 向 Department 表中插入一条记录(部门编号为 s01,部门名称为国际贸易教研室)。输入如下 T-SQL 语句,运行该语句并查看结果,运行结果如图 5-19 所示。

```
INSERT INTO Department(dep_name, dep_id)
VALUES('s01', '国际贸易教研室')
```



图 5-19 常见错误(1)

出错的原因是列列表和值列表没有对应好。

正确的 T-SQL 语句如下:

```
INSERT INTO Department(dep_id, dep_name)
VALUES('s01', '国际贸易教研室')
```

(2) 向 Class 表中插入一条记录(班级编号为 2013gm,班级名称为 13 国贸,所属专业为国际贸易)。

输入如下 T-SQL 语句,运行该语句并查看结果,运行结果如图 5-20 所示。

```
INSERT INTO Class(cla_id,cla_specialty)
VALUES('2013gm','国际贸易')
```



图 5-20 常见错误(2)

出错的原因是没有给 cla\_name 列赋值,但是该列在设计时是不允许为空的。

正确 T-SQL 语句如下:

```
INSERT INTO Class(cla_id,cla_name,cla_specialty)
VALUES('2013gm','13 国贸','国际贸易')
```

(3) 向 Teacher 表中插入一条记录(教师编号为 200604,教师姓名为胡晓,部门编号为 jsj01)。

输入如下 T-SQL 语句,运行该语句并查看结果,运行结果如图 5-21 所示。

```
INSERT INTO Teacher(tea_no,tea_name,tea_departmentid)
VALUES('200604',胡晓,'jsj01')
```

出错的原因是 tea\_name 列的数据类型为文本类型,一般字符类型的列在插入数据时最好用单引号引起来。

正确 T-SQL 语句如下:

```
INSERT INTO Teacher(tea_no,tea_name,tea_departmentid)
VALUES('200604','胡晓','jsj01')
```

(4) 向 Student 表中插入一条记录(学号为 20130202,学生姓名为朱伟,性别为男,籍贯为北京,电子邮箱为 zhuwei@163.com,手机号码为 13388885555,班级编号为 2013wl)。

输入如下 T-SQL 语句,运行该语句并查看结果,运行结果如图 5-22 所示。

```
INSERT INTO
Student(stu_no,stu_name,stu_sex,stu_native,stu_email,stu_phone,stu_classid)
```

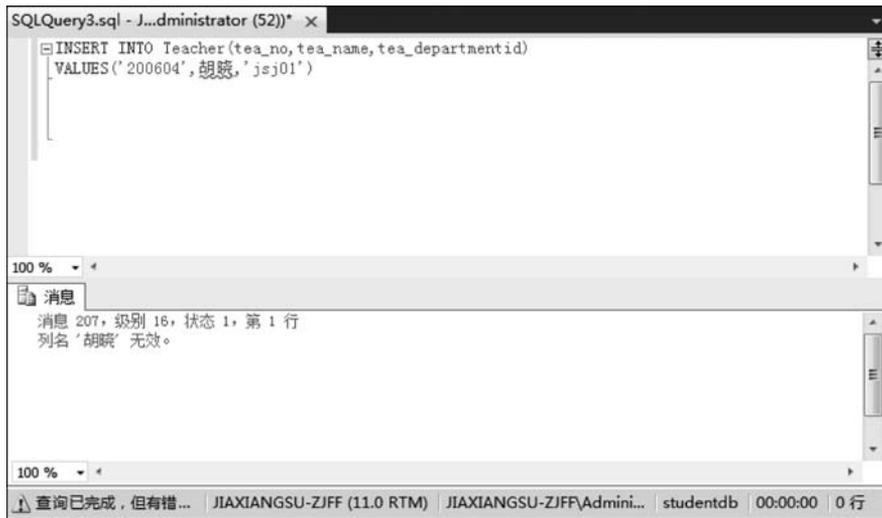


图 5-21 常见错误(3)

```
VALUES('20130202', '朱伟', '男', '北京', 'zhuwei@163com', '13388885555', '2013w1')
```



图 5-22 常见错误(4)

出错的原因是违反了检查约束,在 Student 表中为字段 stu\_email 设置了 CHECK 约束,要求电子邮件的格式包含@符号和点号,而且@符号在点号之前,而常见错误(4)中的电子邮件值没有包含点号。

正确 T-SQL 语句如下:

```
INSERT INTO
Student(stu_no, stu_name, stu_sex, stu_native, stu_email, stu_phone, stu_classid)
VALUES('20130202', '朱伟', '男', '北京', 'zhuwei@163.com', '13388885555', '2013w1')
```

(5) 向 Score 表中插入一条记录(学生学号为 20130102,课程编号为 888001,平时成绩

为 82 分,期末成绩为 83 分,总评成绩为 82.5 分)。

输入如下 T-SQL 语句,运行该语句并查看结果,运行结果如图 5-23 所示。

```
INSERT INTO Score(sco_id,sco_stuno,sco_courseid,sco_usual,sco_final,sco_overall)
VALUES(4,'20130102','888001',82,83,82.5)
```



图 5-23 常见错误(5)

出错的原因是不能给标识列赋值,Score 表中的 sco\_id 列为标识列,而常见错误(5)中的 T-SQL 语句为 sco\_id 赋值为 4。

正确 T-SQL 语句如下:

```
INSERT INTO Score(sco_stuno,sco_courseid,sco_usual,sco_final,sco_overall)
VALUES('20130102','888001',82,83,82.5)
```

此时,studentdb 数据库中的 6 张表中的数据分别如图 5-24~图 5-29 所示。

dep_id	dep_name
jsj01	计算机应用技术教研室
jsj02	计算机网络技术教研室
jsj03	计算机信息管理教研室
s01	国际贸易教研室
NULL	NULL

图 5-24 Department 表中的数据

cla_id	cla_name	cla_specialty
2013gm	13国贸	国际贸易
2013wl	13网络	计算机网络技术
2013xg	13信管	计算机信息管理
2013yy	13应用	计算机应用技术
NULL	NULL	NULL

图 5-25 Class 表中的数据

tea_no	tea_name	tea_departmentid
200601	贾祥素	jsj03
200602	李明	jsj02
200603	王军	jsj01
200604	胡晓	jsj01
NULL	NULL	NULL

图 5-26 Teacher 表中的数据

cou_id	cou_name	cou_credit	cou_teano
888001	SQL Server管理和查询	3	200601
888002	计算机专业英语	2	200601
888003	网页设计	2	200601
888004	平面设计	2	200602
NULL	NULL	NULL	NULL

图 5-27 Course 表中的数据

	stu_no	stu_name	stu_sex	stu_native	stu_email	stu_phone	stu_classid
▶	20130101	王伟	男	浙江杭州	wangwei@163.com	13277776666	2013yy
	20130102	张静	女	浙江宁波	zhangjing@163.com	13616715925	2013yy
	20130201	李超	男	NULL	NULL	NULL	2013wl
	20130202	朱伟	男	北京	zhuwei@163.com	13388885555	2013wl
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 5-28 Student 表中的数据

	sco_id	sco_stuno	sco_courseid	sco_usual	sco_final	sco_overall
▶	1	20130101	888001	80	90	85
	2	20130101	888002	70	80	75
	3	20130101	888003	90	100	95
	4	20130102	888001	82	83	82.5
*	NULL	NULL	NULL	NULL	NULL	NULL

图 5-29 Score 表中的数据

## 5.2.2 插入多行数据

使用 INSERT 语句插入多行数据的语法格式如下：

```
INSERT [INTO]<表名> [列名列表] <子查询>
```

上述语法格式中，“[]”表示可选部分。

列名的数量和数据类型必须和后面子查询的个数和类型一一对应。

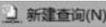
在插入多行数据这一节中会创建一个联系方式数据表 Contact，之后的任务都是向 Contact 表中插入多行数据。

### 1. 从一个表中查询部分信息，将这些信息插入另一个表中

**任务一：**新建联系方式数据表 Contact，有 3 个字段，分别为联系人姓名(con\_name)、联系人邮箱(con\_email)、手机号码(con\_phone)。3 个字段都为文本类型。

建好数据表之后的【表设计】界面如图 5-30 所示。

**任务二：**将 Student 表中已经存在的学生姓名、电子邮箱和手机号码信息插入新建的 Contact 表中。

**【步骤 1】** 单击工具栏中的 ，打开一个空白的 .sql 文件，在查询编辑器窗口输入如下 T-SQL 语句：

```
INSERT INTO Contact(con_name, con_email, con_phone)
SELECT stu_name, stu_email, stu_phone
FROM Student
```

**【步骤 2】** 单击 ，执行语法检查，语法检查通过后，单击 ，执行 T-SQL 命令，如图 5-31 所示。

列名	数据类型	允许 Null 值
con_name	varchar(50)	<input type="checkbox"/>
con_email	varchar(50)	<input checked="" type="checkbox"/>
con_phone	varchar(50)	<input checked="" type="checkbox"/>

图 5-30 Contact【表设计】界面



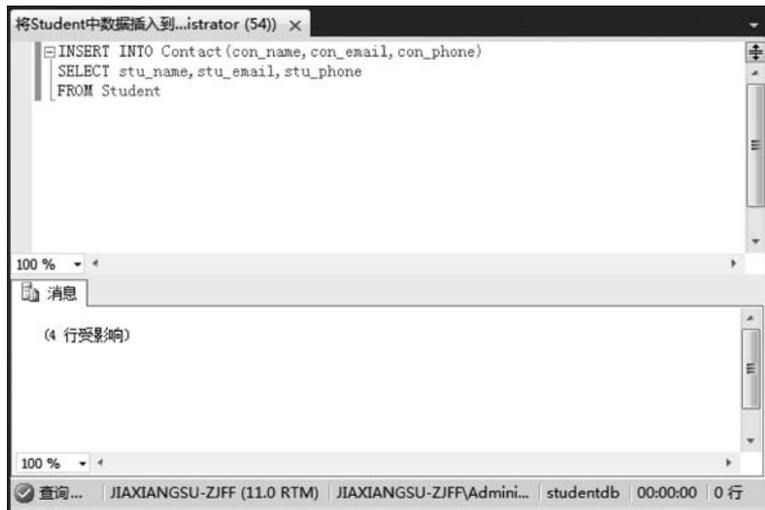


图 5-31 将 Student 表中数据插入 Contact 表中界面 1

**【步骤 3】** 查看 Contact 表中数据,如图 5-32 所示。

con_name	con_email	con_phone
王伟	wangwei@163.com	1327776666
张静	zhangjing@163.com	13616715925
李超	NULL	NULL
朱伟	zhuwei@163.com	13388885555
NULL	NULL	NULL

图 5-32 将 Student 表中数据插入 Contact 表中界面 2

**说明:**

可以通过 INSERT...SELECT 语句将一张表中的数据添加到另一张表中。但是,需要注意的是:查询得到的数据个数、顺序、数据类型必须与插入的项保持一致。

## 2. 通过 Union 关键字合并数据进行插入

**任务三:** 通过 Union 关键字合并数据,向 Contact 表中插入 3 条记录。

**【步骤 1】** 单击工具栏中的 新建查询(N), 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
INSERT INTO Contact(con_name, con_email, con_phone)
SELECT '黄飞', 'huangfei@163.com', '13322226666' UNION
SELECT '徐强', 'xuqiang@163.com', '13644445555' UNION
SELECT '段悦', 'duanyue@163.com', '13566668888'
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 执行(X), 执行 T-SQL 命令, 如图 5-33 所示。

**【步骤 3】** 查看 Contact 表中数据,如图 5-34 所示。

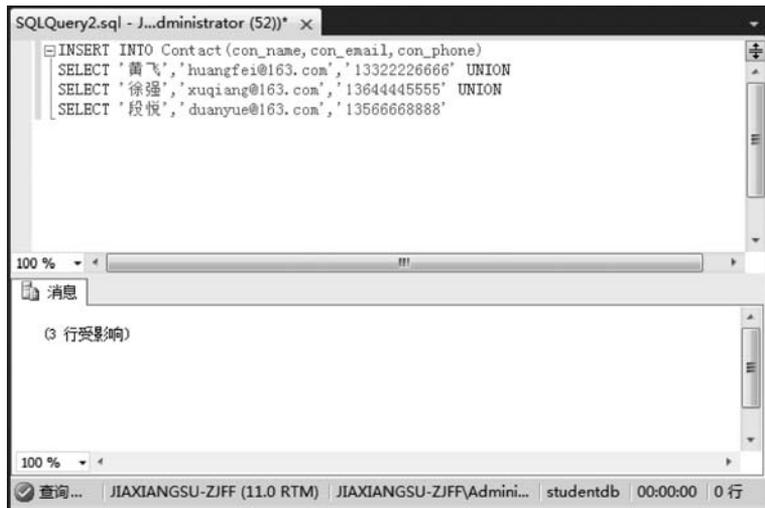


图 5-33 通过 UNION 关键字向 Contact 表中插入多行数据

The screenshot shows the Contact table with the following data:

con_name	con_email	con_phone
王伟	wangwei@163.com	13277776666
张静	zhangjing@163.com	13616715925
李超	NULL	NULL
朱伟	zhuwei@163.com	13388885555
段悦	duanyue@163.com	13566668888
黄飞	huangfei@163.com	13322226666
徐强	xuqiang@163.com	13644445555
* NULL	NULL	NULL

图 5-34 通过 UNION 关键字插入数据后 Contact 表中数据

#### 说明：

可以通过 INSERT...SELECT...UNION 语句向数据表中插入多行数据，其中的 UNION 关键字用于将两个不同的数据或查询结果组合成一个新的结果集。需要注意的是：查询得到的数据个数、顺序、数据类型必须与插入的项保持一致。

### 3. 通过 INSERT...VALUES 语句一次插入多条记录

**任务四：**通过 INSERT...VALUES 语句一次向 Contact 表中插入 3 条记录。

**【步骤 1】** 单击工具栏中的 新建查询(N)，打开一个空白的 .sql 文件，在查询编辑器窗口中输入如下 T-SQL 语句：

```
INSERT INTO Contact(con_name, con_email, con_phone)
VALUES('胡康', 'hukang@163.com', '13211116666'),
      ('周星', 'zhouxing@163.com', '13322227777'),
      ('林杰', 'linjie@163.com', '13399996666')
```

**【步骤 2】** 单击 ，执行语法检查，语法检查通过后，单击 执行(F)，执行 T-SQL 命令，如图 5-35 所示。



图 5-35 通过 INSERT...VALUES 语句向 Contact 表中插入多行数据

**【步骤 3】** 查看 Contact 表中数据,如图 5-36 所示。

con_name	con_email	con_phone
王伟	wangwei@163.com	1327776666
张静	zhangjing@163.com	13616715925
李超	NULL	NULL
朱伟	zhuwei@163.com	1338885555
段悦	duanyue@163.com	1356668888
黄飞	huangfei@163.com	1332226666
徐强	xuqiang@163.com	1364445555
胡康	hukang@163.com	13211116666
周星	zhouxing@163.com	13322227777
林杰	linjie@163.com	13399996666
NULL	NULL	NULL

图 5-36 通过 INSERT...VALUES 语句插入多行数据后 Contact 表中的数据

说明:

可以通过 INSERT...VALUES 语句向数据表中插入多行数据,插入时指定多个值列表,每个值列表之间用逗号分隔。

## 5.3 使用 T-SQL 语句更新数据



数据表中插入数据之后,有时会对数据进行更新。例如,数据表中有一个字段存放网站的访问量,那么这个访问量会随时增加,这就要用到数据的更新。T-SQL 语句中使用 UPDATE 语句更新表中记录,每次可以更新部分或全部记录,更新时可以指定更新条件,从而更新一条或多条记录(若没有满足条件的记录,则记录都不会被更新)。如果没有指定更新条件,则更新全部记录。

使用 UPDATE 语句更新数据的语法格式如下:

```
UPDATE <表名> SET <列名 = 更新值> [WHERE <条件>]
```

(1) SET 关键字后面指定要修改的列名和该列对应的值,可以对多个数据列进行更新,多个“列名=更新值”之间用逗号分隔。注意:最后一个“列名=更新值”后面不用加逗号。

(2) WHERE 子句是可选的。主要指定对哪条记录或者哪些记录进行更新操作。如果省略 WHERE 子句,则对表中所有的记录进行更新。

### 5.3.1 更新单行数据

**任务一:** 为 Contact 表添加两个字段,分别为班级名称(con\_class)和编号(con\_id)。其中,con\_class 的类型为 varchar(50); con\_id 的类型为 int。然后,向 Contact 表中存在数据的行中添加这两个字段的值。

**【步骤 1】** 修改后的 Contact【表设计】界面如图 5-37 所示。

列名	数据类型	允许 Null 值
con_name	varchar(50)	<input type="checkbox"/>
con_email	varchar(50)	<input checked="" type="checkbox"/>
con_phone	varchar(50)	<input checked="" type="checkbox"/>
con_class	varchar(50)	<input checked="" type="checkbox"/>
con_id	int	<input checked="" type="checkbox"/>

图 5-37 修改后的 Contact【表设计】界面

**【步骤 2】** 补充 Contact 表中存在数据行的两个新建字段信息,补充好之后表中数据如图 5-38 所示。

	con_name	con_email	con_phone	con_class	con_id
▶	王伟	wangwei@163.com	13277776666	13应用	1
	张静	zhangjing@163.com	13616715925	13应用	2
	李超	NULL	NULL	13应用	3
	朱伟	zhuwei@163.com	13388885555	13应用	4
	段悦	duanyue@163.com	13566668888	13网络	5
	黄飞	huangfei@163.com	13322226666	13网络	6
	徐强	xuqiang@163.com	13644445555	13网络	7
	胡康	hukang@163.com	13211116666	13信管	8
	周星	zhouxing@163.com	13322227777	13信管	9
	林杰	linjie@163.com	13399996666	13信管	10
*	NULL	NULL	NULL	NULL	NULL

图 5-38 补充新建字段后 Contact 表中的数据

**任务二:** 更新 Contact 表中编号为 1 的学生手机号码为 15788889999。

**【步骤 1】** 单击工具栏中的 新建查询(N), 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
UPDATE Contact SET con_phone = '15788889999'
WHERE con_id = 1
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 执行(F), 执行 T-SQL 命令。

**【步骤 3】** 查看 Contact 表中数据,编号为 1 的手机号码修改成功,如图 5-39 所示。

	con_name	con_email	con_phone	con_class	con_id
▶	王伟	wangwei@163.com	15788889999	13应用	1
	张静	zhangjing@163.com	13616715925	13应用	2
	李超	NULL	NULL	13应用	3
	朱伟	zhuwei@163.com	13388885555	13应用	4
	段悦	duanyue@163.com	13566668888	13网络	5
	黄飞	huangfei@163.com	13322226666	13网络	6
	徐强	xuqiang@163.com	13644445555	13网络	7
	胡康	hukang@163.com	13211116666	13信管	8
	周星	zhouxing@163.com	13322227777	13信管	9
	林杰	linjie@163.com	13399996666	13信管	10
*	NULL	NULL	NULL	NULL	NULL

图 5-39 修改编号为 1 的学生手机号码之后 Contact 表中数据

**任务三:** 更新 Contact 表姓名为“林杰”的学生姓名为“林小杰”。

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
UPDATE Contact SET con_name = '林小杰'
WHERE con_name = '林杰'
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 , 执行 T-SQL 命令。

**【步骤 3】** 查看 Contact 表中数据, 已成功将“林杰”的名字改为“林小杰”, 如图 5-40 所示。

	con_name	con_email	con_phone	con_class	con_id
	王伟	wangwei@163.com	15788889999	13应用	1
	张静	zhangjing@163.com	13616715925	13应用	2
	李超	NULL	NULL	13应用	3
	朱伟	zhuwei@163.com	13388885555	13应用	4
	段悦	duanyue@163.com	13566668888	13网络	5
	黄飞	huangfei@163.com	13322226666	13网络	6
	徐强	xuqiang@163.com	13644445555	13网络	7
	胡康	hukang@163.com	13211116666	13信管	8
	周星	zhouxing@163.com	13322227777	13信管	9
▶	林小杰	linjie@163.com	13399996666	13信管	10
*	NULL	NULL	NULL	NULL	NULL

图 5-40 将“林杰”改为“林小杰”之后 Contact 表中数据

**任务四:** 更新 Contact 表编号为 2 的学生邮箱为 zhj@163.com, 手机号码为 15766667777。

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
UPDATE Contact SET con_email = 'zhj@163.com', con_phone = '15766667777'
WHERE con_id = 2
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 , 执行 T-SQL 命令。

**【步骤 3】** 查看 Contact 表中数据, 编号为 2 的学生邮箱为 zhj@163.com, 手机号码为

15766667777,修改成功,如图 5-41 所示。

JIAXIANGSU-ZJFF....db - dbo.Contact x					
	con_name	con_email	con_phone	con_class	con_id
	王伟	wangwei@163.com	15788889999	13应用	1
▶	张静	zhj@163.com	15766667777	13应用	2
	李超	NULL	NULL	13应用	3
	朱伟	zhuwei@163.com	13388885555	13应用	4
	段悦	duanyue@163.com	13566668888	13网络	5
	黄飞	huangfei@163.com	13322226666	13网络	6
	徐强	xuqiang@163.com	13644445555	13网络	7
	胡康	hukang@163.com	13211116666	13信管	8
	周星	zhouxing@163.com	13322227777	13信管	9
	林小杰	linjie@163.com	13399996666	13信管	10
*	NULL	NULL	NULL	NULL	NULL

图 5-41 修改编号为 2 的学生邮箱和手机号码之后 Contact 表中数据

说明:

- (1) 文本类型的数据放在单引号中,数值类型的数据不用加单引号,直接写数据值即可。
- (2) SET 后面可以设置多个数据列的更新值,之间用逗号隔开。

### 5.3.2 更新多行数据

任务一:更新 Contact 表 con\_email 字段为空的行的电子邮箱为 zjff@163.com。

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件,在查询编辑器窗口中输入如下 T-SQL 语句:

```
UPDATE Contact SET con_email = 'zjff@163.com'
WHERE con_email IS NULL
```

**【步骤 2】** 单击 , 执行语法检查,语法检查通过后,单击 , 执行 T-SQL 命令。

**【步骤 3】** 查看 Contact 表中数据,成功给电子邮箱为空的行添加 zjff@163.com 邮箱,如图 5-42 所示。

JIAXIANGSU-ZJFF....db - dbo.Contact x					
	con_name	con_email	con_phone	con_class	con_id
	王伟	wangwei@163.com	15788889999	13应用	1
	张静	zhj@163.com	15766667777	13应用	2
▶	李超	zjff@163.com	NULL	13应用	3
	朱伟	zhuwei@163.com	13388885555	13应用	4
	段悦	duanyue@163.com	13566668888	13网络	5
	黄飞	huangfei@163.com	13322226666	13网络	6
	徐强	xuqiang@163.com	13644445555	13网络	7
	胡康	hukang@163.com	13211116666	13信管	8
	周星	zhouxing@163.com	13322227777	13信管	9
	林小杰	linjie@163.com	13399996666	13信管	10
*	NULL	NULL	NULL	NULL	NULL

图 5-42 给电子邮箱为空的行添加 zjff@163.com 邮箱之后 Contact 表中的数据

说明:

因为本例中电子邮箱为空的只有一条记录,所以更新了一条记录。如果有多条记录的电子邮箱为空,则会更新多条记录。

**任务二:**更新 Contact 表,设置 13 信管班所有学生的电子邮箱为 xg@163.com。

**【步骤 1】**单击工具栏中的 ,打开一个空白的.sql文件,在查询编辑器窗口中输入如下 T-SQL 语句:

```
UPDATE Contact SET con_email = 'xg@163.com'
WHERE con_class = '13 信管'
```

**【步骤 2】**单击 ,执行语法检查,语法检查通过后,单击 ,执行 T-SQL 命令。

**【步骤 3】**查看 Contact 表中数据,成功将 13 信管班的邮箱都改为 xg@163.com,如图 5-43 所示。

	con_name	con_email	con_phone	con_class	con_id
	王伟	wangwei@163.com	15788889999	13应用	1
	张静	zhj@163.com	15766667777	13应用	2
	李超	zjff@163.com	NULL	13应用	3
	朱伟	zhuwei@163.com	13388885555	13应用	4
	段悦	duanyue@163.com	13566668888	13网络	5
	黄飞	huangfei@163.com	13322226666	13网络	6
	徐强	xuqiang@163.com	13644445555	13网络	7
	胡康	xg@163.com	13211116666	13信管	8
	周星	xg@163.com	13322227777	13信管	9
	林小杰	xg@163.com	13399996666	13信管	10
*	NULL	NULL	NULL	NULL	NULL

图 5-43 将 13 信管班的邮箱都改为 xg@163.com 之后 Contact 表中数据

### 5.3.3 更新所有数据

**任务一:**更新 Contact 表中所有人的手机号码为 13388888888。

**【步骤 1】**单击工具栏中的 ,打开一个空白的.sql文件,在查询编辑器窗口中输入如下 T-SQL 语句:

```
UPDATE Contact SET con_phone = '13388888888'
```

**【步骤 2】**单击 ,执行语法检查,语法检查通过后,单击 ,执行 T-SQL 命令。

**【步骤 3】**查看 Contact 表中数据,成功将表中的手机号码都改为 13388888888,如图 5-44 所示。

**任务二:**更新 Contact 表中所有人的电子邮箱为 zjff@163.com。

**【步骤 1】**单击工具栏中的 ,打开一个空白的.sql文件,在查询编辑器窗口中输入如下 T-SQL 语句:

```
UPDATE Contact SET con_email = 'zjff@163.com'
```

**【步骤 2】**单击 ,执行语法检查,语法检查通过后,单击 ,执行 T-SQL 命令。

con_name	con_email	con_phone	con_class	con_id
王伟	wangwei@163.com	13388888888	13应用	1
张静	zhj@163.com	13388888888	13应用	2
李超	zjff@163.com	13388888888	13应用	3
朱伟	zhuwei@163.com	13388888888	13应用	4
段悦	duanyue@163.com	13388888888	13网络	5
黄飞	huangfei@163.com	13388888888	13网络	6
徐强	xuqiang@163.com	13388888888	13网络	7
胡康	xg@163.com	13388888888	13信管	8
周星	xg@163.com	13388888888	13信管	9
林小杰	xg@163.com	13388888888	13信管	10
NULL	NULL	NULL	NULL	NULL

图 5-44 将手机号码都改为 13388888888 之后 Contact 表中数据

**【步骤 3】** 查看 Contact 表中数据,成功将表中的电子邮箱都改为 zjff@163.com,如图 5-45 所示。

con_name	con_email	con_phone	con_class	con_id
王伟	zjff@163.com	13388888888	13应用	1
张静	zjff@163.com	13388888888	13应用	2
李超	zjff@163.com	13388888888	13应用	3
朱伟	zjff@163.com	13388888888	13应用	4
段悦	zjff@163.com	13388888888	13网络	5
黄飞	zjff@163.com	13388888888	13网络	6
徐强	zjff@163.com	13388888888	13网络	7
胡康	zjff@163.com	13388888888	13信管	8
周星	zjff@163.com	13388888888	13信管	9
林小杰	zjff@163.com	13388888888	13信管	10
NULL	NULL	NULL	NULL	NULL

图 5-45 将电子邮箱都改为 zjff@163.com 之后 Contact 表中数据

总结:

使用 UPDATE 语句更新数据,可能更新一行数据,可能更新多行数据,可能更新所有数据,也可能不更新任何数据(如果不满足 WHERE 子句的条件则不会更新任何数据)。

## 5.4 使用 T-SQL 语句删除数据

数据库中数据会经常变化,有些数据不再需要了就要删除。例如,学生表中有个学生退学了,就可以将该学生的记录从学生表中删除。T-SQL 语句中使用 DELETE 语句删除表中记录,每次可以删除部分记录或全部记录,删除时可以指定删除条件从而删除一条或多条记录(若没有满足条件的记录则一个都不会被删除),如果没有指定删除条件则删除全部记录。

使用 DELETE 语句删除数据的语法格式如下:

```
DELETE FROM <表名> [WHERE <条件>]
```



### 5.4.1 删除单行数据

**任务一：**删除 Contact 表中编号为 10 的学生记录。

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
DELETE FROM Contact WHERE con_id = 10
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 , 执行 T-SQL 命令。

**【步骤 3】** 查看 Contact 表中数据, 成功将编号为 10 的学生删除, 如图 5-46 所示。

con_name	con_email	con_phone	con_class	con_id
王伟	zjff@163.com	13388888888	13应用	1
张静	zjff@163.com	13388888888	13应用	2
李超	zjff@163.com	13388888888	13应用	3
朱伟	zjff@163.com	13388888888	13应用	4
段悦	zjff@163.com	13388888888	13网络	5
黄飞	zjff@163.com	13388888888	13网络	6
徐强	zjff@163.com	13388888888	13网络	7
胡康	zjff@163.com	13388888888	13信管	8
周星	zjff@163.com	13388888888	13信管	9
NULL	NULL	NULL	NULL	NULL

图 5-46 删除编号为 10 的学生记录之后 Contact 表中数据

**任务二：**删除 Contact 表中姓名为“王伟”的学生记录。

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
DELETE FROM Contact WHERE con_name = '王伟'
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 , 执行 T-SQL 命令。

**【步骤 3】** 查看 Contact 表中数据, 成功将姓名为“王伟”的记录删除, 如图 5-47 所示。

con_name	con_email	con_phone	con_class	con_id
张静	zjff@163.com	13388888888	13应用	2
李超	zjff@163.com	13388888888	13应用	3
朱伟	zjff@163.com	13388888888	13应用	4
段悦	zjff@163.com	13388888888	13网络	5
黄飞	zjff@163.com	13388888888	13网络	6
徐强	zjff@163.com	13388888888	13网络	7
胡康	zjff@163.com	13388888888	13信管	8
周星	zjff@163.com	13388888888	13信管	9
NULL	NULL	NULL	NULL	NULL

图 5-47 删除姓名为“王伟”的学生记录之后 Contact 表中数据

**任务三：**删除 Department 表中部门编号为 s01 的记录。

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
DELETE FROM Department WHERE dep_id = 's01'
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 , 执行 T-SQL 命令。

**【步骤 3】** 查看 Department 表中数据, 成功将部门编号为 s01 的记录删除, 如图 5-48 所示。

**任务四: 删除 Department 表中部门编号为 jsj01 的记录。**

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
DELETE FROM Department WHERE dep_id = 'jsj01'
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 , 执行 T-SQL 命令。执行该命令时出现错误, 如图 5-49 所示。



dep_id	dep_name
jsj01	计算机应用技术教研室
jsj02	计算机网络技术教研室
jsj03	计算机信息管理教研室
NULL	NULL

图 5-48 删除部门编号为 s01 的记录之后 Department 表中数据



图 5-49 删除 Department 表中部门编号为 jsj01 的记录时出错

**分析:** 为什么会出现如图 5-49 所示的删除错误信息?

**出错原因:** 这是删除信息时常见的错误类型, 在删除主表中的某条数据信息时, 如果该信息在子表中以外键形式存在, 则禁止删除主表中的该信息数据。本节任务四中, 主表为 Department, 子表为 Teacher, 如果删除主表中的部门信息(jsj01), 而这个部门是 Teacher 表中两位老师所在部门, 则无法删除该部门。而本节中任务三可以删除部门编号为 s01 的记录, 是因为 Teacher 表中不存在部门编号为该部门的教师, 这种情况是允许删除主表中信息的。Teacher 表中数据如图 5-50 所示。



tea_no	tea_name	tea_department
200601	贾祥素	jsj03
200602	李明	jsj02
200603	王军	jsj01
200604	胡晓	jsj01
NULL	NULL	NULL

图 5-50 Teacher 表中数据

**解决办法:** 必须将要删除的部门信息先从子表 Teacher 表中删除, 即删除该部门在教师表中的全部信息后, 才允许删除部门表中的部门信息。但是, 采用这种方法删除信息非常烦琐, 通常采取

的手段是通过删除触发器的办法进行主表和子表数据信息的连带删除。

### 5.4.2 删除多行数据

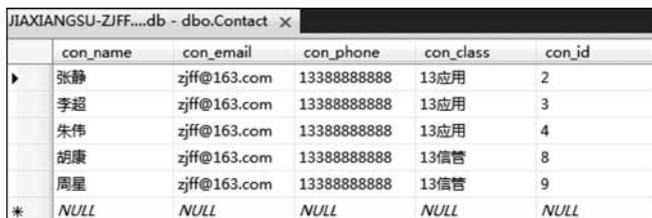
**任务：**删除 Contact 表中班级为“13 网络”的学生记录。

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
DELETE FROM Contact WHERE con_class = '13 网络'
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 , 执行 T-SQL 命令。

**【步骤 3】** 查看 Contact 表中数据, 成功将班级为“13 网络”的记录删除, 如图 5-51 所示。

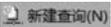


	con_name	con_email	con_phone	con_class	con_id
▶	张静	zjff@163.com	13388888888	13应用	2
	李超	zjff@163.com	13388888888	13应用	3
	朱伟	zjff@163.com	13388888888	13应用	4
	胡康	zjff@163.com	13388888888	13信管	8
	周星	zjff@163.com	13388888888	13信管	9
*	NULL	NULL	NULL	NULL	NULL

图 5-51 删除班级为“13 网络”的记录之后 Contact 表中数据

### 5.4.3 删除所有数据

**任务：**删除 Contact 表中所有记录。

**【步骤 1】** 单击工具栏中的 , 打开一个空白的 .sql 文件, 在查询编辑器窗口中输入如下 T-SQL 语句:

```
DELETE FROM Contact
```

**【步骤 2】** 单击 , 执行语法检查, 语法检查通过后, 单击 , 执行 T-SQL 命令。

**【步骤 3】** 查看 Contact 表中数据, 成功将表中所有记录删除, 如图 5-52 所示。



	con_name	con_email	con_phone	con_class	con_id
*	NULL	NULL	NULL	NULL	NULL

图 5-52 删除表中所有记录之后 Contact 表中数据

**扩展：**使用 TRUNCATE TABLE 删除数据。

**语法：**

```
TRUNCATE TABLE <表名>
```

TRUNCATE TABLE 用来删除表中的所有行, 功能上类似于没有 WHERE 子句的 DELETE 语句, 但是它比 DELETE 语句执行速度快, 而且使用的系统资源和事务日志资源更少。





图 5-54 【SQL Server 导入和导出向导】窗口

**【步骤 2】** 单击**【下一步】**按钮,打开**【SQL Server 导入和导出向导-选择数据源】**窗口,此处选择的数据库是 studentdb,如图 5-55 所示。



图 5-55 选择数据源

**【步骤 3】** 单击【下一步】按钮,打开【选择目标】窗口,确定数据导出的格式及导出文件存放路径。选择【目标】右侧的下拉框,选择【平面文件目标】,以保存文本文件;在文件名处选择文件路径及要保存的文件名称,如图 5-56 所示。



图 5-56 选择目标(平面文件目标)

**【步骤 4】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-指定表复制或查询】窗口,此处选择默认的【复制一个或多个表或视图的数据】,如图 5-57 所示。



图 5-57 指定表复制或查询

**【步骤 5】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-配置平面文件目标】窗口,在【源表或源视图】右侧的下拉列表中选择数据表[dbo].[Department],如图 5-58 所示。



图 5-58 配置平面文件目标

**【步骤 6】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-保存并运行包】窗口,按照默认选中【立即运行】复选框,如图 5-59 所示。



图 5-59 保存并运行包

**【步骤 7】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-完成该向导】窗口,如图 5-60 所示。



图 5-60 完成该向导

**【步骤 8】** 单击【完成】按钮,即开始执行导出操作,导出成功后打开【SQL Server 导入和导出向导-执行成功】窗口,如图 5-61 所示。



图 5-61 执行成功(导出 Department 表)

**【步骤 9】** 单击【关闭】按钮,查看导出文件,如图 5-62 所示。

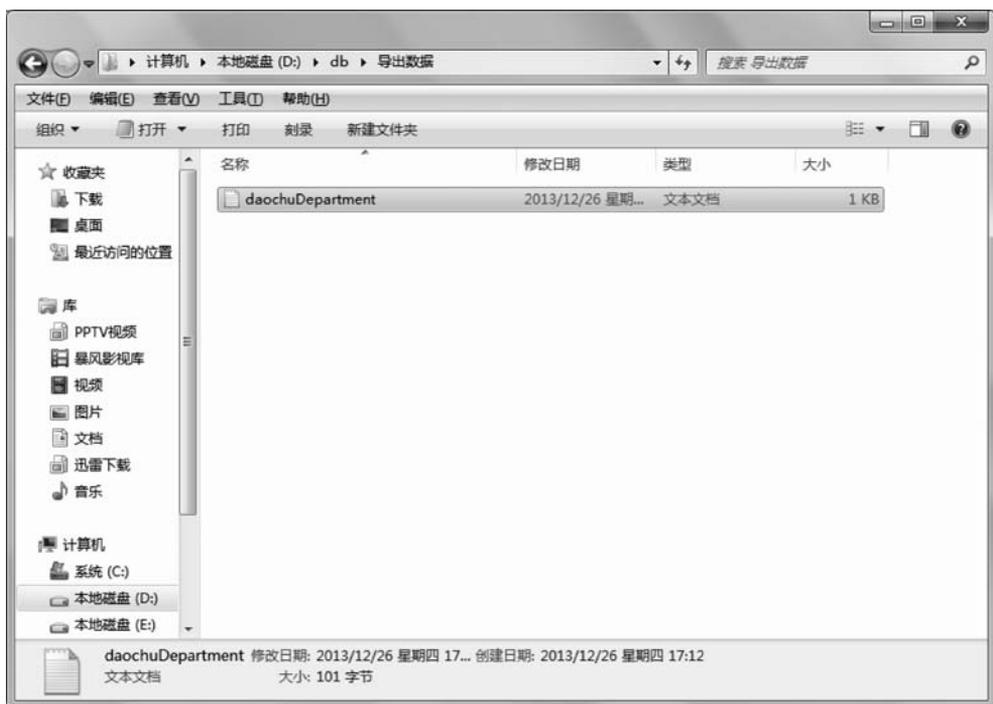


图 5-62 查看导出的文本文件(daochuDepartment)

**【步骤 10】** 双击 daochuDepartment 文件,查看文本文件内容,如图 5-63 所示。

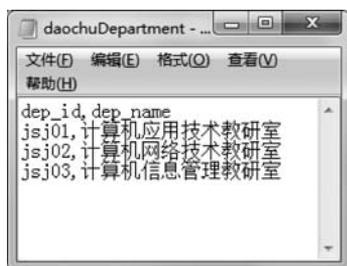


图 5-63 查看文本文件内容

## 2. 将数据库中数据导出为 Excel 文件

**任务二:** 将 studentdb 数据库中 Student 数据表的数据导出,保存为 Excel 文件。

**【步骤 1】** 右击 studentdb 数据库,依次选择【任务】→【导出数据】,单击【导出数据】选项,打开【SQL Server 导入和导出向导】窗口。

**【步骤 2】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-选择数据源】窗口,此处选择的数据库是 studentdb。

库是 studentdb。

**【步骤 3】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-选择目标】窗口,确定数据导出的格式及导出文件存放路径。选择【目标】右侧的下拉列表中的 Microsoft Excel 选项,以保存为 Excel 文件;在 Excel 连接设置处选择文件路径及要保存的文件名称,如图 5-64 所示。

**【步骤 4】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-指定表复制或查询】窗口,此处选择默认的【复制一个或多个表或视图的数据】。

**【步骤 5】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-选择源表和源视图】窗口,选中数据表[dbo].[Student],如图 5-65 所示。



图 5-64 选择目标 (Microsoft Excel)



图 5-65 选择源表和源视图

**【步骤 6】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-查看数据类型映射】窗口,如图 5-66 所示。



图 5-66 查看数据类型映射

**【步骤 7】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-保存并运行包】窗口。

**【步骤 8】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-完成该向导】窗口,如图 5-67 所示。

**【步骤 9】** 单击【完成】按钮,即开始执行导出操作,导出成功后打开【SQL Server 导入和导出向导-执行成功】窗口,如图 5-68 所示。

**【步骤 10】** 单击【关闭】按钮,查看导出文件,如图 5-69 所示。

**【步骤 11】** 双击 daochuStudent 文件,查看文件内容,如图 5-70 所示。

## 5.5.2 导入数据

### 1. 将文本文件导入数据库表中

**任务一:** 将名为 daoruContact 的文本文件导入 studentdb 数据库的 Contact 数据表中。首先查看 daoruContact 文本文件中的内容,如图 5-71 所示。



图 5-67 完成该向导(导出为 Excel 文件)



图 5-68 执行成功(导出 Student 表)

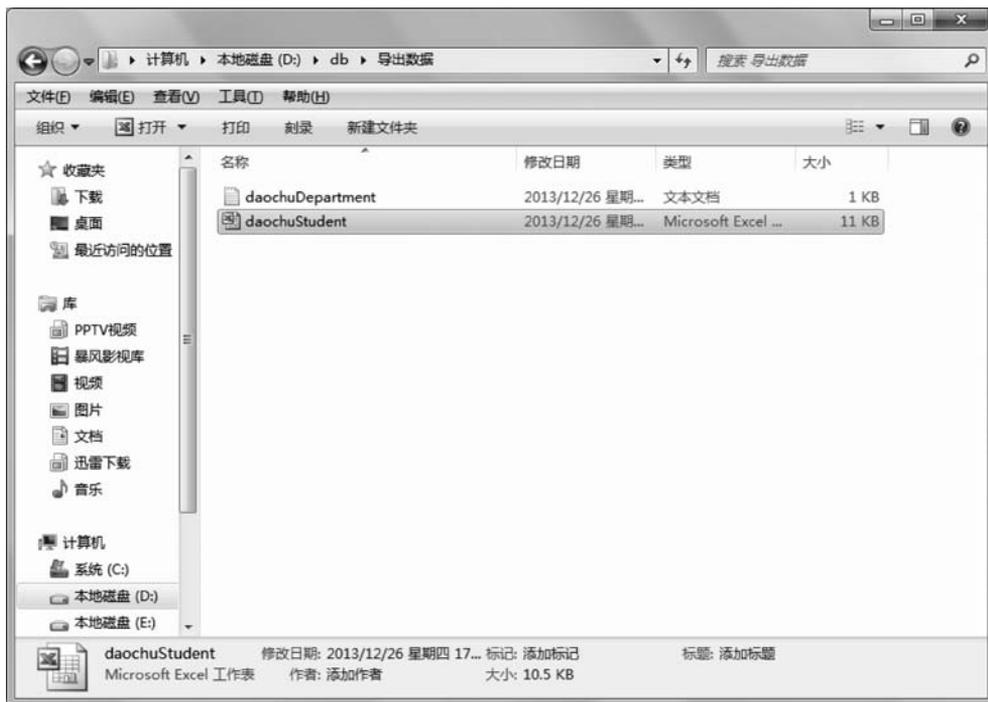


图 5-69 查看导出的 Excel 文件(daochuStudent)

A	B	C	D	E	F	G
stu_no	stu_name	stu_sex	stu_native	stu_email	stu_phone	stu_classid
20130101	王伟	男	浙江杭州	wangwei@163.com	1327776666	2013yy
20130102	张静	女	浙江宁波	zhangjing@163.com	13616715925	2013yy
20130201	李超	男				2013w1
20130202	朱伟	男	北京	zhuwei@163.com	13388885555	2013w1

图 5-70 查看 Excel 文件内容



图 5-71 daoruContact 文本文件中的内容

**【步骤 1】** 右击 studentdb 数据库,依次选择【任务】→【导入数据】,如图 5-72 所示。打开【SQL Server 导入和导出向导】窗口,如图 5-73 所示。

**【步骤 2】** 单击【下一步】按钮,打开【SQL Server 导入和导出向导-选择数据源】窗口,在【数据源】右边的下拉列表中选择【平面文件源】,单击【浏览】按钮选择要从哪个文件导入数据,取消选中【在第一个数据行中显示列名称】复选框,如图 5-74 所示。

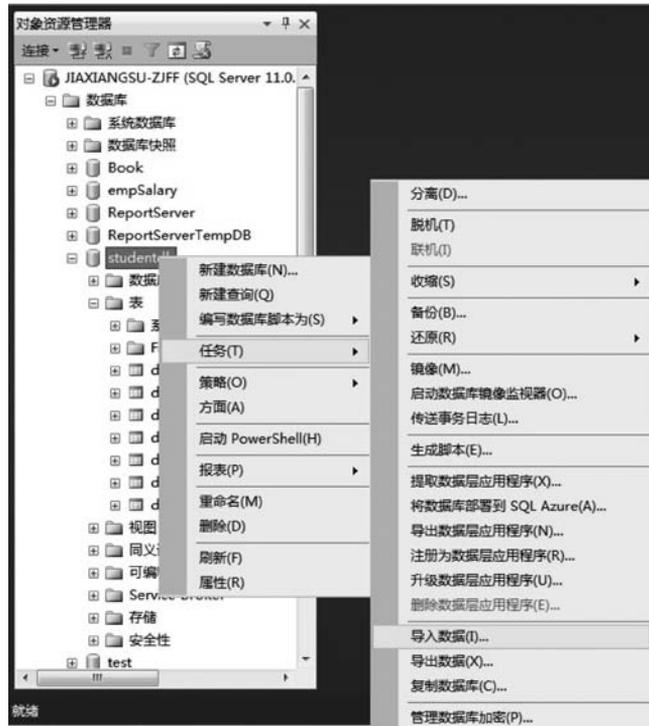


图 5-72 将文本文件导入 Contact 表界面 1



图 5-73 将文本文件导入 Contact 表界面 2



图 5-74 将文本文件导入 Contact 表界面 3

**【步骤 3】** 单击【下一步】按钮,如图 5-75 所示。



图 5-75 将文本文件导入 Contact 表界面 4

**【步骤 4】** 单击【下一步】按钮,选择数据库为 studentdb,如图 5-76 所示。



图 5-76 将文本文件导入 Contact 表界面 5

**【步骤 5】** 单击【下一步】按钮,在目标处选择 [dbo].[Contact],如图 5-77 所示。



图 5-77 将文本文件导入 Contact 表界面 6

**【步骤 6】** 单击【下一步】按钮,按照默认值,如图 5-78 所示。



图 5-78 将文本文件导入 Contact 表界面 7

**【步骤 7】** 单击【下一步】按钮,选中【立即运行】复选框,如图 5-79 所示。



图 5-79 将文本文件导入 Contact 表界面 8

【步骤 8】 单击【下一步】按钮,如图 5-80 所示。



图 5-80 将文本文件导入 Contact 表界面 9

【步骤 9】 单击【完成】按钮,执行导入操作,执行成功后的界面如图 5-81 所示。



图 5-81 将文本文件导入 Contact 表界面 10

**【步骤 10】** 查看 Contact 表中数据,将文本文件中的数据成功导入数据库表中,如图 5-82 所示。

	con_name	con_email	con_phone	con_class	con_id
	王伟	wangwei@16...	1327776666	13应用	1
	张静	zhangjing@16...	13616715925	13信管	2
	李超	lichao@163.c...	13277668844	13信管	3
	朱伟	zhuwei@163.c...	13388885555	13网络	4
*	NULL	NULL	NULL	NULL	NULL

图 5-82 将文本文件导入 Contact 表界面 11

## 2. 将 Excel 文件导入数据库表中

**任务二:** 将名为 daoruContact 的 Excel 文件导入 studentdb 数据库的 Contact 数据表中。首先查看名为 daoruContact 的 Excel 文件中的内容,如图 5-83 所示。

王伟伟	wangwei@163.com	1327776666	13应用	1
张静静	zhangjing@163.com	13616715925	13信管	2
李超超	lichao@163.com	13277668844	13信管	3
朱伟伟	zhuwei@163.com	13388885555	13网络	4

图 5-83 名为 daoruContact 的 Excel 文件中的内容

**【步骤 1】** 右击 studentdb 数据库,依次选择**【任务】**→**【导入数据】**。打开**【SQL Server 导入和导出向导】**窗口,如图 5-84 所示。



图 5-84 将 Excel 文件导入 Contact 表界面 1

**【步骤 2】** 单击【下一步】按钮,在【数据源】右侧的下拉列表中选择 Microsoft Excel 选项,单击【浏览】按钮选择要导入的 Excel 文件的路径,取消选中【首行包含列名称】复选框,如图 5-85 所示。



图 5-85 将 Excel 文件导入 Contact 表界面 2

**【步骤 3】** 单击【下一步】按钮,选择数据库为 studentdb,如图 5-86 所示。



图 5-86 将 Excel 文件导入 Contact 表界面 3

**【步骤 4】** 单击【下一步】按钮,选中【复制一个或多个表或视图的数据】单选按钮,如图 5-87 所示。



图 5-87 将 Excel 文件导入 Contact 表界面 4

**【步骤 5】** 单击【下一步】按钮,选中‘Sheet1\$’复选框(因为要导入的数据放在这个数据表中),在【目标】处选择[dbo].[Contact],如图 5-88 所示。



图 5-88 将 Excel 文件导入 Contact 表界面 5

【步骤 6】 单击【下一步】按钮,如图 5-89 所示。



图 5-89 将 Excel 文件导入 Contact 表界面 6

【步骤 7】 单击【下一步】按钮,选中【立即运行】复选框,如图 5-90 所示。



图 5-90 将 Excel 文件导入 Contact 表界面 7

【步骤 8】 单击【下一步】按钮,如图 5-91 所示。



图 5-91 将 Excel 文件导入 Contact 表界面 8

【步骤 9】 单击【完成】按钮,执行导入操作,导入成功后的界面如图 5-92 所示。



图 5-92 将 Excel 文件导入 Contact 表界面 9

【步骤 10】 查看 Contact 表中数据,成功将 Excel 中的数据导入数据库,如图 5-93 所示。

con_name	con_email	con_phone	con_class	con_id
王伟伟	wangwei@16...	1327776666	13应用	1
张静静	zhangjing@16...	13616715925	13信管	2
李超超	lichao@163.c...	13277668844	13信管	3
朱伟伟	zhuwei@163.c...	13388885555	13网络	4
王伟	wangwei@16...	1327776666	13应用	1
张静	zhangjing@16...	13616715925	13信管	2
李超	lichao@163.c...	13277668844	13信管	3
朱伟	zhuwei@163.c...	13388885555	13网络	4
NULL	NULL	NULL	NULL	NULL

图 5-93 将 Excel 文件导入 Contact 表界面 10

## 5.6 本章总结

1. SQL Server 中对数据的管理包括插入、更新和删除。
2. 对数据的管理有两种方式:使用 SSMS 图形界面和使用 T-SQL 语句。
3. 使用 INSERT 语句插入数据。
4. 使用 UPDATE 语句更新数据。
5. 使用 DELETE 语句删除数据。
6. 使用 SSMS 图形界面进行数据导入和导出。

## 习题 5

### 一、选择题

1. 在 T-SQL 语法中,用来插入数据的命令是( )。
 

(A) DELETE	(B) INSERT
(C) UPDATE	(D) SELECT
2. 在 T-SQL 语法中,用来更新数据的命令是( )。
 

(A) DELETE	(B) INSERT
(C) UPDATE	(D) SELECT
3. 在 T-SQL 语法中,用来删除数据的命令是( )。
 

(A) DELETE	(B) INSERT
(C) UPDATE	(D) SELECT
4. 运行如下 T-SQL 语句:

```
TRUNCATE TABLE Student
```

其运行结果是( )。

- (A) 删除 Student 表

- (B) 删除 Student 表中数据,不会删除表结构及表的约束
- (C) 删除 Student 表中数据,表的约束也被删除
- (D) 删除 Student 表中数据,同时删除表结构

5. 假设 Employee 表中,emp\_id 列为主键,并且为自动增长的标识列(其中,标识种子为 2,标识递增量为 1),同时还有 emp\_name,emp\_age 列,其中 emp\_name 为文本类型,另外两个为整型,目前数据表中还没有数据,则执行如下 T-SQL 语句:

```
INSERT INTO Employee(emp_id,emp_name,emp_age) VALUES (1,'李明',30)
```

其运行结果是( )。

- (A) 插入数据成功,emp\_id 的值为 1
- (B) 插入数据成功,emp\_id 的值为 2
- (C) 插入数据成功,emp\_id 的值为 3
- (D) 插入数据失败

6. 假设 Student 表中 stu\_age 列存放学生年龄,stu\_age 列为整型,Student 表中目前有 10 条记录,则执行如下 T-SQL 语句:

```
UPDATE Student SET stu_age = 17
```

其运行结果是( )。

- (A) 将 Student 表中第一个学生的年龄修改为 17
- (B) 将 Student 表中部分学生的年龄修改为 17
- (C) 将 Student 表中 10 个学生的年龄都修改为 17
- (D) 更新数据失败

7. 假设 Student 表中共有 3 个字段,stu\_name,stu\_email,stu\_address,这 3 个字段均为文本类型,都允许为空,其中 stu\_email 的默认值为 zjff@163.com,则执行如下 T-SQL 语句:

```
INSERT Student(stu_name,stu_address) VALUES ('张三','北京')
```

其运行结果是( )。

- (A) INSERT 语法错误
- (B) stu\_email 列的值为 zjff@163.com
- (C) stu\_email 列的值为 NULL
- (D) stu\_email 列的值为“北京”

8. 表 A 中的列 B 是标识列,属于自动增长数据类型,标识种子是 3,标识递增量为 2,首先插入四行记录,然后删除一行记录,再次向表 A 中添加记录时,标识列的值是( )。

- (A) 11
- (B) 9
- (C) 13
- (D) 7

9. 要创建一个 INSERT 语句,插入取自另一个表的值,使用( )语句或子句代替 VALUES 子句从另一个表中提取数据。

- (A) DELETE
- (B) INSERT
- (C) UPDATE
- (D) SELECT

## 二、操作题

使用 T-SQL 语句管理图书出版管理系统数据库(Book)。

图书出版管理系统中有两个表,分别如下:

- (1) 图书表(书号,书名,作者编号,出版社,出版日期)。
- (2) 作者表(作者编号,作者姓名,年龄,地址,作者手机号码)。

每张表详细的字段信息、约束详见习题 4 操作题。

1. 使用 INSERT 插入单行数据。

要求:分别向两张表中插入一条记录。

1) 作者表

作者编号: a01; 作者姓名: 张强; 年龄: 40; 地址: 浙江金华; 作者手机号码: 13222223333。

2) 图书表

书号: b01; 书名: 网页设计; 作者编号: a01; 出版社: 出版社 1; 出版日期: 2013-1-1。

2. 使用 INSERT...SELECT...UNION 语句向数据表中插入多行数据。

要求:分别向两张表中插入五条记录。

1) 作者表

(1) 作者编号: a02; 作者姓名: 张燕; 年龄: 37; 地址: 浙江宁波; 作者手机号码: 15755556666。

(2) 作者编号: a03; 作者姓名: 周静; 年龄: 39; 地址: 浙江杭州; 作者手机号码: 13899990000。

(3) 作者编号: a04; 作者姓名: 杨丽; 年龄: 49; 地址: 北京; 作者手机号码: 13755557777。

(4) 作者编号: a05; 作者姓名: 胡星; 年龄: 52; 地址: 上海; 作者手机号码: 13688886666。

(5) 作者编号: a06; 作者姓名: 李明; 年龄: 31; 地址: 上海; 作者手机号码: 13233335555。

2) 图书表

(1) 书号: b02; 书名: SQL Server 教程; 作者编号: a01; 出版社: 出版社 1; 出版日期: 2014-1-1。

(2) 书号: b03; 书名: 大学语文; 作者编号: a02; 出版社: 出版社 2; 出版日期: 2013-10-11。

(3) 书号: b04; 书名: 大学英语; 作者编号: a05; 出版社: 出版社 3; 出版日期: 2013-9-21。

(4) 书号: b05; 书名: 计算机网络教程; 作者编号: a03; 出版社: 出版社 1; 出版日期: 2013-8-15。

(5) 书号: b06; 书名: 高等数学; 作者编号: a04; 出版社: 出版社 1; 出版日期: 2014-1-1。

3. 使用 UPDATE 语句更新数据。

要求:分别更新两张表中部分记录。

1) 作者表-修改作者姓名

将作者编号为 a01 的作者姓名修改为“张小强”。

2) 作者表-年龄加 1

将作者表中所有作者的年龄加 1。

3) 图书表

将书号为 b01 的出版时间修改为 2013-2-1。

4. 使用 DELETE 语句删除数据。

要求：分别删除两张表中部分记录。

1) 作者表

删除作者姓名为“李明”的记录。

2) 图书表

删除书名为“大学语文”的记录。

## 上机 5

本次上机任务：

(1) 使用 INSERT 插入数据。

(2) 使用 UPDATE 更新数据。

(3) 使用 DELETE 删除数据。

要求：本章上机用到的数据库为员工工资数据库(empSalary)，该数据库中有三张表格。

分别如下：

(1) 员工信息表(员工编号, 员工姓名, 性别, 年龄, 所属部门编号, 毕业院校, 健康情况, 手机号码)。

(2) 部门表(部门编号, 部门名称)。

(3) 工资信息表(工资编号, 员工编号, 应发工资, 实发工资)。

每张表详细的字段信息、约束详见上机 4。

任务 1：使用 INSERT 插入单行数据。

要求：分别向三张表中插入一条记录。

1) 部门表

部门编号：caiwu01；部门名称：财务部。

2) 员工信息表

员工编号：201301；员工姓名：段杰；性别：男；年龄：25；所属部门编号：caiwu01；毕业院校：北京大学；健康情况：良好；手机号码：13377778888。

3) 工资表

工资编号：gz01；员工编号：201301；应发工资：3900；实发工资：3100。

任务 2：使用 INSERT...SELECT...UNION 语句向数据表中插入多行数据。

要求：分别向三张表中插入四条记录。

### 1) 部门表

- (1) 部门编号: renli01; 部门名称: 人力部。
- (2) 部门编号: shichang01; 部门名称: 市场部。
- (3) 部门编号: xinxi01; 部门名称: 信息部。
- (4) 部门编号: zonghe01; 部门名称: 综合部。

### 2) 员工信息表

(1) 员工编号: 200001; 员工姓名: 李文; 性别: 男; 年龄: 32; 所属部门编号: caiwu01; 毕业院校: 浙江大学; 健康情况: 良好。

(2) 员工编号: 200002; 员工姓名: 江晓丽; 性别: 女; 年龄: 35; 所属部门编号: renli01; 毕业院校: 宁波大学; 健康情况: 良好。

(3) 员工编号: 200003; 员工姓名: 李悦; 性别: 女; 年龄: 37; 所属部门编号: renli 01; 毕业院校: 北京大学; 健康情况: 良好。

(4) 员工编号: 200004; 员工姓名: 李强; 性别: 男; 年龄: 61; 所属部门编号: renli 01; 毕业院校: 西安交大; 健康情况: 一般。

### 3) 工资表

(1) 工资编号: gz02; 员工编号: 200001; 应发工资: 4600; 实发工资: 3850。

(2) 工资编号: gz03; 员工编号: 200002; 应发工资: 4800; 实发工资: 4000。

(3) 工资编号: gz04; 员工编号: 200003; 应发工资: 5100; 实发工资: 4200。

(4) 工资编号: gz05; 员工编号: 200004; 应发工资: 4700; 实发工资: 3900。

任务 3: 新建一个表 EmpInfo2(员工姓名 e\_name, 员工年龄 e\_age)使用 INSERT... SELECT 语句将 EmpInfo 表中的姓名和年龄信息添加到 EmpInfo2 中。

要求: EmpInfo2 表中的员工姓名为文本类型, 员工年龄为整型。

任务 4: 使用 UPDATE 语句更新数据。

要求: 分别更新三张表中部分记录。

### 1) 部门表

将“信息部”修改为“信息中心”。

### 2) 员工信息表

将员工姓名为“李悦”的部门编号修改为 shichang01。

### 3) 工资表

将实发工资小于 4000 元的实发工资增加 50 元。

任务 5: 使用 DELETE 语句删除数据。

要求: 分别删除三张表中部分记录。

### 1) 部门表

删除部门名称为“综合部”的记录。

### 2) 工资表

删除员工编号为 200004 的记录。

### 3) 员工信息表

删除年龄大于 60 的记录。