

第3章 数据表的管理与维护



- 了解：SQL Server 的基本数据类型和数据库完整性的类型。
- 理解：数据表和表数据的概念；表对象的管理和维护；数据库完整性的概念。
- 掌握：对象资源管理器和使用 T-SQL 语句创建表、管理和维护表的基本操作；数据库完整性设置的基本操作。

3.1 数据表的创建和管理

一个数据库可以拥有许多表，每个表都代表一个特定的实体，如学生数据库可能包含学生个人信息、院系信息、课程信息、成绩信息等多个表。对每个实体使用一个单独的表可以消除重复数据，使数据存储更有效，并减少数据输入项错误。

SQL Server 2008 中数据库的主要对象是数据表，创建好数据库后，就可以向数据库中添加数据表。数据通常存储在表中，表存储在数据库文件中，任何有相应权限的用户都可以对其进行操作。在 SQL Server 2008 中数据表的创建可以以图形界面方式完成，也可以使用 T-SQL 完成。

3.1.1 使用对象资源管理器创建数据表

在创建数据表之前应首先确定数据表的结构，即确定数据表的字段个数、字段名、字段类型、字段宽度及小数位数等，然后再输入相应的记录。

使用“对象资源管理器”创建数据库 student_db 中的学生信息表 St_Info，其表结构见 1.4.4 节中表 1.4。操作步骤如下：

(1) 以管理员身份启动 SSMS，在“对象资源管理器”中，选中要添加表的数据库 student_db 并展开，右击“表”对象，如图 3.1 所示。

(2) 在弹出的快捷菜单中单击“新建表”命令，在 SSMS 主窗口的右边弹出“表设计器”窗口，同时在 SSMS 的主菜单栏中出现“表设计器”菜单，可以使用此菜单下的命令对表进行相关操作，当关闭“表设计器”窗口时，此菜单也随之关闭。

(3) 在“表设计器”中，根据设计好的表结构对列名、数据类型（包括长度）、是否允许空进行相应的设置。

注意：在 SQL Server 中，一个汉字占据两个字符的位置，因此计算一个字段长度时，一个汉字的长度是 2B。

(4) 在“St_Id”（学号）列上右击，在弹出的快捷菜单中选择“设置主键”命令，如图 3.2 所示。在“列属性”选项卡中的“说明”项中填写“主键”说明。St_Info 表结构设计后的结果如图 3.3 所示。

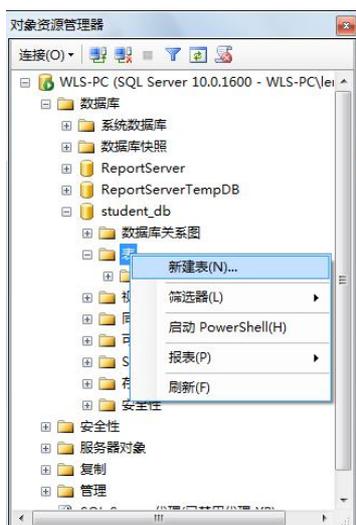


图 3.1 选择“新建表”命令

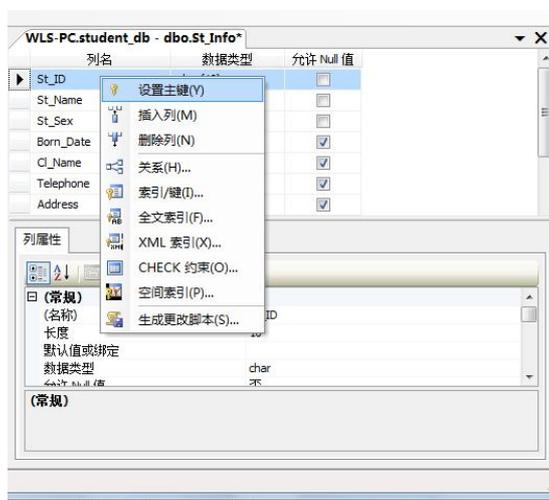


图 3.2 设置 St_Info 表的主键

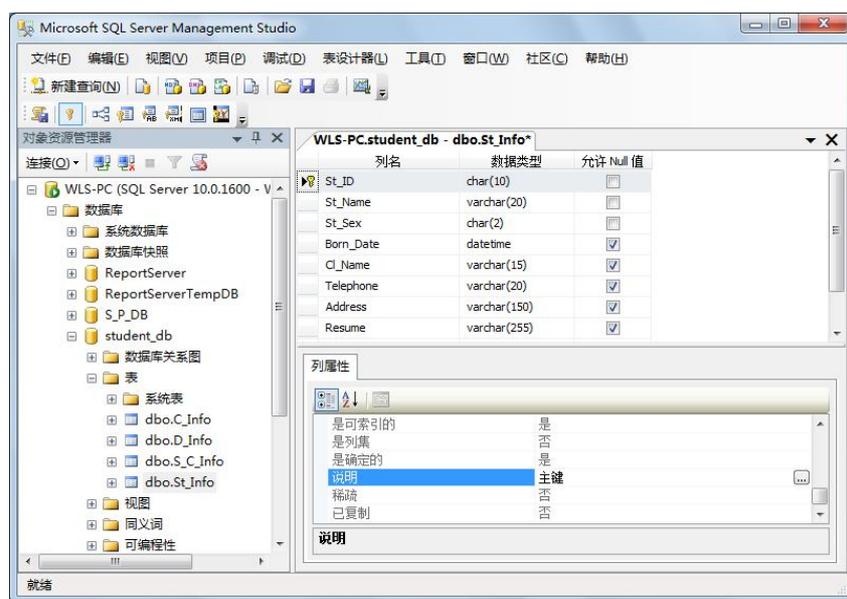


图 3.3 St_Info 表结构编辑完成结果

(5) 设置完成后，单击工具栏上的“保存”按钮或单击“表设计器”右上角的关闭按钮, 在弹出的“选择名称”对话框中输入表名为“St_Info”，如图 3.4 所示。单击“确定”按钮，完成数据表的创建。



图 3.4 “选择名称”对话框

此时可在“对象资源管理器”右窗格的表项目列表中看到新建立的 St_Info 数据表。

3.1.2 使用 T-SQL 创建数据表

使用 T-SQL 创建数据表的命令是 CREATE TABLE, 此语句带有很多参数, 可以完成非常强大的功能。其语法格式如下:

```
CREATE TABLE
[database_name.[owner.][owner.]table_name
({<column_definition>
| column_name AS computed_column_expression
| <table_constraint>:=[CONSTRAINT constraint_name]}
| [{PRIMARY KEY | UNIQUE}])
[ON {filegroup | DEFAULT}]
[TEXTIMAGE_ON {filegroup | DEFAULT}]
<column_definition>:={column_name data_type}
[COLLATE <collation_name>]
[DEFAULT constant_expression]
```

参数说明:

(1) database_name: 表示要在其中创建表的数据库名称。database_name 必须是现有数据库的名称。如果不指定数据库, database_name 默认为当前数据库。

(2) owner: 是新表所有者的用户 ID 名, owner 必须是 database_name 所指定的数据库中的现有用户 ID, owner 默认为与 database_name 所指定的数据库中当前连接相关联的用户 ID。

(3) table_name: 表示新表的名称。表名必须符合标识符规则且命名必须唯一。table_name 最多可包含 128 个字符。

(4) column_name: 表示表中的列名。列名必须符合标识符规则, 并且在表内唯一。

(5) ON {filegroup | DEFAULT}: 指定存储表的文件组。如果指定 filegroup, 则表将存储在指定的文件组中。数据库中必须存在该文件组。如果指定 DEFAULT 或者根本未指定 ON 参数, 则表存储在默认文件组中。

(6) TEXTIMAGE_ON: 表示 text、ntext 和 image 列存储在指定文件组中的关键字。如果表中没有 text、ntext 或 image 列, 则不能使用 TEXTIMAGE_ON。如果没有指定 TEXTIMAGE_ON, 则 text、ntext 和 image 列将与表存储在同一个文件组中。

【例 3.1】在数据库 student_db 中创建 Student 学生信息表, 要求包含 S_NO (学号)、NAME (姓名)、AGE (年龄)、SEX (性别) 信息, 其中学号不能为空。

操作步骤如下:

(1) 在 SSMS 主窗口中, 单击“新建查询”按钮。

(2) 在“查询设计器”窗口中输入创建表的语句。

语句如下:

```
Use student_db
Go
CREATE TABLE Student
(S_NO CHAR(7) NOT NULL,
NAME CHAR(10),
AGE SMALLINT,
SEX CHAR(1))
```

注意：第一条语句“Use student_db”表示打开数据库。在对数据库进行操作之前，必须打开数据库；Go为批处理语句结束标志。

(3)单击工具栏上的“分析”按钮,若结果窗口无错误信息,再单击“执行”按钮,系统返回“命令已成功完成”信息,如图3.5所示。完成 Student(学生信息)表的创建。

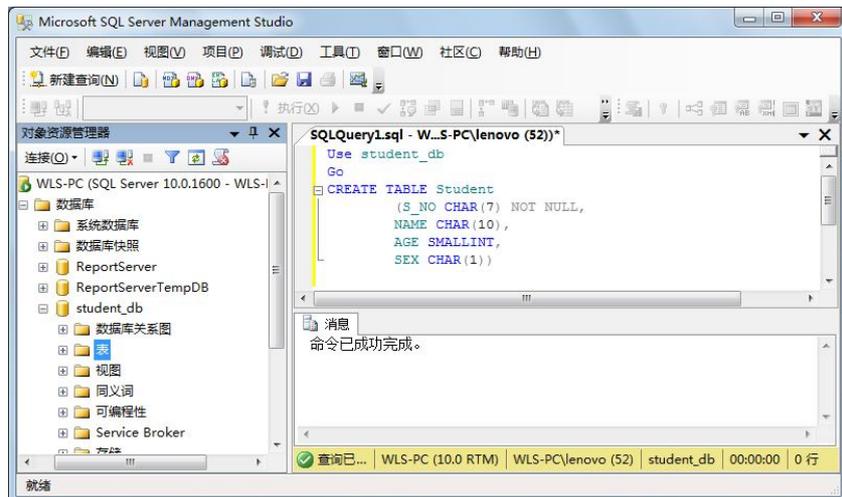


图 3.5 创建 Student 表

【例 3.2】在数据库 student_db 中创建 Student_1 学生信息表,要求包含 S_NO(学号)、NAME(姓名)、AGE(年龄)信息,并限制年龄在 15~25 周岁之间,指定学号为主关键字。语句如下:

```
Use student_db
Go
CREATE TABLE Student_1
    (S_NO CHAR(7) NOT NULL,          /*用 NOT NULL 说明非空*/
    NAME CHAR(10),
    AGE SMALLINT,
    PRIMARY KEY(S_NO),              /*用 PRIMARY KEY 关键字设置主键*/
    CHECK(AGE BETWEEN 15 AND 25))  /*用 CHECK 关键字设置年龄限制*/
```

有时需要临时创建一个中间表,完成一些临时存储数据的功能,在完成临时功能之后,再删除这些临时表。在 T-SQL 中可用 CREATE TABLE 来创建临时表,只要在表名前加“#”或“##”符号。其中“#”表示本地临时表,在当前数据库内使用;“##”表示全局临时表,可在所有数据库内使用。这些表存储在系统数据库 tempdb 中,它们在与服务器的交互结束时自动删除,而且如果与服务器的交互异常而结束,这些表仍会被删除。

【例 3.3】创建临时表 temp_student。

语句如下:

```
CREATE TABLE #temp_student          /*用#说明 temp_student 为本地临时表*/
(学号 SMALLINT NOT NULL,
 姓名 VARCHAR(30) NOT NULL,
 年龄 INT NOT NULL,
  PRIMARY KEY(学号)
)
```

【例 3.4】在学生数据库 student_db 中创建课程信息表 C_Info, 表结构和要求参见 1.4.4 节的表 1.4。

语句如下:

```
Use student_db
Go
CREATE TABLE C_Info
    (C_No CHAR(10) NOT NULL,
    C_Name VARCHAR(30) NOT NULL,
    C_Type CHAR(4),
    C_Credit smallint NOT NULL,
    C_Des VARCHAR(255)
    PRIMARY KEY(C_NO))
```

读者可以参照此方法创建学生数据库 student_db 中的选课信息表 S_C_Info 和学院信息表 D_Info, 这两个表的结构和要求参见 1.4.4 节中的表 1.5 和表 1.6。

【例 3.5】建立一个供货商和货物的数据库 S_P_DB, 此数据库存在以下关系:

(1) 供货商 S (S_NO, S_NAME, STATUS, CITY)。分别表示供货商代码、名称、身份、所在的城市。

(2) 货物 P (P_NO, P_NAME, WEIGHT, CITY)。分别表示货物的编号、名称、重量和产地。

要求如下:

(1) 供货商代码不能为空, 且值是唯一的, 供货商名称也是唯一的。

(2) 货物编号不能为空, 且值是唯一的, 货物的名称也不能为空。

使用以下 T-SQL 语句创建数据库 S_P_DB, 创建关系 S 和关系 P 为表 S 和 P:

```
/* 创建数据库 S_P_DB */
CREATE DATABASE S_P_DB
/* 创建供货商信息表 S */
CREATE TABLE S
    (S_NO CHAR(9) NOT NULL UNIQUE,
    S_NAME CHAR(20) UNIQUE,
    STATUS CHAR(9),
    CITY CHAR(10),
    PRIMARY KEY(S_NO))
/* 创建货物信息表 P */
CREATE TABLE P
    (P_NO CHAR(9) NOT NULL UNIQUE,
    P_NAME CHAR(20) UNIQUE,
    WEIGHT CHAR(9),
    CITY CHAR(10),
    PRIMARY KEY(P_NO))
```

因为表 S 和表 P 之间存在一个多对多的联系, 但在 SQL Server 中, 数据库不能直接管理这种联系, 所以必须创建一个新表 SP (其原因可参见第 1 章的相关内容), 其主键由 S 表和 P 表的主键构成。表 SP 的列分别为 S_NO、P_NO 和 QTY, 其中 S_NO 和 P_NO 为外部关键字, QTY 为进货数量。创建该数据表的 T-SQL 语句如下:

```
/* 创建进货信息表 SP */
CREATE TABLE SP
```

```

(S_NO CHAR(9),
P_NO CHAR(9),
QTY CHAR(9),
PRIMARY KEY(S_NO ,P_NO),
FOREIGN KEY(S_NO) REFERENCES S(S_NO),
FOREIGN KEY(P_NO) REFERENCES P(P_NO))

```

注意：SQL Server 的每个数据库最多可存储 20 亿个表，每个表可以有 1024 列，每行最多可以存储 8060 字节。表和列的命名必须是唯一的，但同一数据库中的不同表可使用相同的列名。必须为每列指定数据类型。

3.1.3 使用对象资源管理器对数据表进行管理

创建数据表后，可以使用对象资源管理器或 T-SQL 对数据表进行管理，这些管理主要有更改表名、增加列、删除列、修改已有列的属性（列名、数据类型、是否为空值等）。

1. 使用对象资源管理器更改数据表名称

SQL Server 2008 中允许改变一个表的名字，但当表名改变后，与此相关的某种对象（如视图），以及通过表名与表相关的存储过程将全部无效。因此，建议一般不要更改一个已有的表名，特别是在其上定义了视图或建立了相关的表时。

使用对象资源管理器更改数据表名称的操作步骤如下：

(1) 在“对象资源管理器”中选择要更改的数据表并右击，在弹出的快捷菜单中单击“重命名”命令，如图 3.6 所示。



图 3.6 选择“重命名”命令

(2) 待重命名的表显示高亮的蓝色状态后，重新输入数据表的名称，然后按回车键，完成选定数据表重命名的操作。

2. 使用对象资源管理器删除数据表

删除数据表时，表的定义、表中的所有数据以及表的索引、触发器、约束等均被删除。

在 SQL Server 2008 中不能删除系统表和外键约束所参照的表。

操作步骤如下：

(1) 在“对象资源管理器”中选择要删除的数据表并右击，在弹出的快捷菜单中单击“删除”命令，参看图 3.6。

(2) 在弹出的“删除对象”窗口（图 3.7）中，显示待删除的表的名字，单击“确定”按钮，进行删除表的操作。

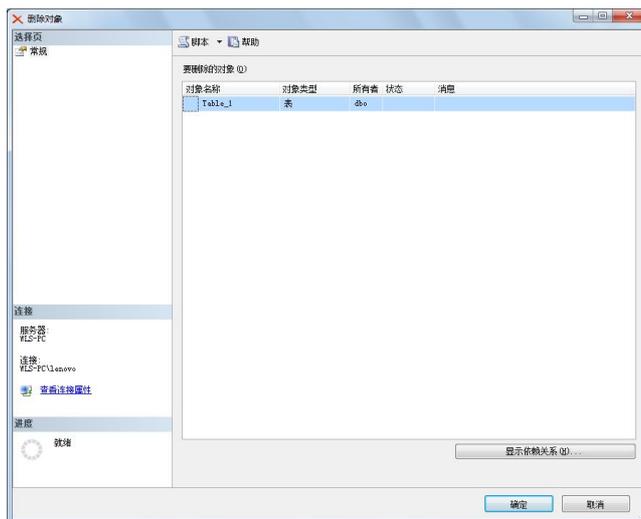


图 3.7 “删除对象”窗口

3. 使用对象资源管理器修改数据表结构

使用对象资源管理器修改数据表结构的操作步骤如下：

(1) 在“对象资源管理器”中选择要修改表结构的数据表并右击，在弹出的快捷菜单中单击“设计”命令，如图 3.8 所示。

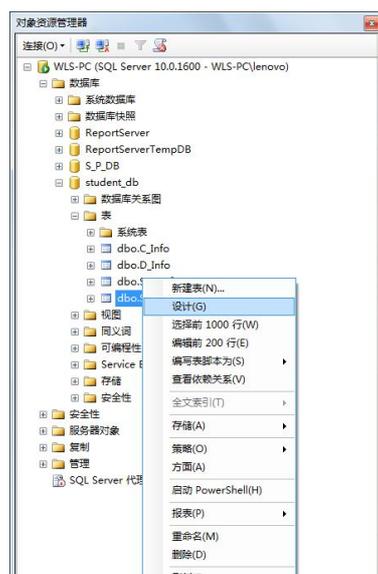


图 3.8 选择“设计”命令

(2) 弹出如图 3.3 所示的“表设计器”界面，可以在此界面中修改数据表的相关选项。修改完成后，若单击“表设计器”右上角的关闭按钮，系统弹出如图 3.9 所示的对话框，单击“是”按钮，修改有效；单击“否”按钮，修改无效；单击“取消”按钮，系统返回“表设计器”界面。

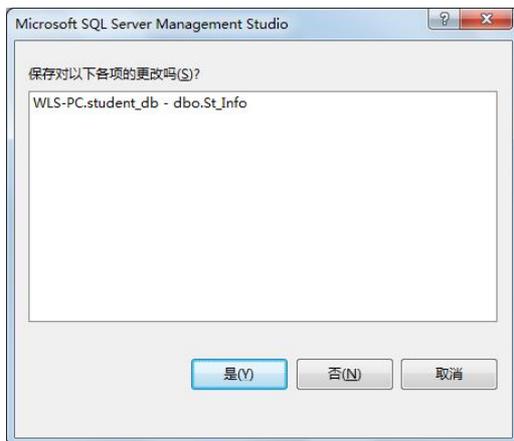


图 3.9 保存数据表结构修改确认对话框

3.1.4 使用 T-SQL 对数据表进行管理

在 SQL Server 2008 中可以使用 T-SQL 对数据表进行修改和删除。

1. 修改数据表

在 T-SQL 中，修改数据表可使用 ALTER TABLE 语句，其语法格式如下：

```
ALTER TABLE table
{ [ ALTER COLUMN column_name
  { new_data_type [ ( precision [ , scale ] ) ]
    [ COLLATE < collation_name > ]
    [ NULL | NOT NULL ]
    | { ADD | DROP } ROWGUIDCOL } ]
  | ADD { [ < column_definition > ] | column_name AS computed_column_expression } [ ,...n ]
  | [ WITH CHECK | WITH NOCHECK ] ADD { < table_constraint > } [ ,...n ]
  | DROP { [ CONSTRAINT ] constraint_name | COLUMN column } [ ,...n ]
  | { CHECK | NOCHECK } CONSTRAINT { ALL | constraint_name [ ,...n ] }
  | { ENABLE | DISABLE } TRIGGER { ALL | trigger_name [ ,...n ] }
}
```

参数说明：

- (1) table: 要更改的数据表的名称。
- (2) ALTER COLUMN: 指定要更改的列。
- (3) column_name: 要更改、添加的列的名称。
- (4) new_data_type [(precision [, scale])]: 如果要修改表中已经存在的列，必须指定与该列相兼容的新数据类型。其中 precision 用于指定数据类型的精度；scale 用于指定数据类型的小数位数。
- (5) COLLATE < collation_name >: 为更改列指定新的排序规则。

(6) NULL|NOT NULL: 指定该列是否可以接受空值, 默认定义是允许空值。

(7) {ADD | DROP } ROWGUIDCOL: 在指定列上添加或去除 ROWGUIDCOL 属性。

(8) ADD { [< column_definition >] | column_name AS computed_column_expression } [,...n]: 指定要添加一个或多个列定义、计算列定义或者表约束。

(9) [WITH CHECK | WITH NOCHECK] ADD { < table_constraint > } [,...n]: 指定表中的数据是否用新添加的或重新启用的 FOREIGN KEY 或 CHECK 约束进行验证。如果没有指定, 将新约束假定为 WITH CHECK, 将重新启用的约束假定为 WITH NOCHECK。

(10) DROP { [CONSTRAINT] constraint_name | COLUMN column } [,...n]: 指定删除某个列。

(11) { CHECK | NOCHECK } CONSTRAINT { ALL | constraint_name [,...n] }: 指定启用或禁用 constraint_name。如果禁用, 将来插入或更新该列时将不用该约束条件进行验证。此选项只能与 FOREIGN KEY 和 CHECK 约束一起使用。其中 ALL 表示指定使用 NOCHECK 选项禁用所有约束, 或者使用 CHECK 选项启用所有约束。

(12) { ENABLE | DISABLE } TRIGGER { ALL | trigger_name [,...n] }: 指定启用或禁用 trigger_name。当一个触发器被禁用时, 它对表的定义依然存在; 然而, 当在表上执行 INSERT、UPDATE 或 DELETE 语句时, 触发器中的操作将不执行, 除非重新启用该触发器。其中 ALL 表示指定启用或禁用表中所有的触发器。

【例 3.6】在例 3.1 的 Student 学生信息 (st_info) 表中增加 zzmm (政治面貌) 字段。增加该字段的语句如下:

```
ALTER TABLE Student ADD zzmm char(4)
```

【例 3.7】将例 3.6 中增加的字段 zzmm 的宽度由 4 修改为 8。修改该字段的语句如下:

```
ALTER TABLE Student ALTER COLUMN zzmm char(8)
```

注意: 在新增加字段时, 不管原来的表中是否有数据, 新增加的字段值一律为空。

【例 3.8】删除例 3.6 中增加的 zzmm 字段。删除该字段的语句如下:

```
ALTER TABLE Student DROP COLUMN zzmm
```

2. 删除数据表

在 T-SQL 中, 删除数据表可使用 DROP TABLE 语句, 其语法格式如下:

```
DROP TABLE table_name
```

参数说明:

table_name: 表示要删除的表名。

注意: ① DROP TABLE 不能用于删除由 FOREIGN KEY 约束引用的表, 必须先删除引用的 FOREIGN KEY 约束或引用的表。②在系统表上不能使用 DROP TABLE 语句。

【例 3.9】删除当前数据库中的 Student_1 学生信息表。

删除该表的语句如下:

```
Drop TABLE Student_1
```

【例 3.10】在同一个语句中指定多个表并对它们进行删除。假设有一个 stu 数据库, 此数据库中存在 book 表和 temp2 表。

同时删除这两张表的语句如下:

```
DROP TABLE book,temp2
```

【例 3.11】删除指定数据库中的表。假设 stu 数据库内有 temp1 表，可以在任何数据库内执行以下语句，完成删除 temp1 表的操作。

删除该表的语句如下：

```
DROP TABLE stu.dbo.temp1
```

注意：删除的表不在当前数据库下时，必须加前缀，即加数据库名和所有者，并用点运算符连接。

3.2 表数据的管理

表数据的管理主要是指对表进行添加或插入新数据、更改或更新现有数据、删除现有数据、检索（或查询）现有数据的操作。这些操作可使用对象资源管理器或 T-SQL 语句完成。

3.2.1 使用对象资源管理器管理表数据

使用对象资源管理器，可以在“表设计器”界面添加、修改和删除数据，还可以创建数据库关系图，实现对表数据的管理。

1. 在表设计器中管理数据

【例 3.12】使用对象资源管理器，完成 student_db 数据库中 St_Info 表的数据输入。St_Info 表的数据如图 3.10 所示。

St_ID	St_Na...	St_Sex	Born_Date	Cl_Name	Telephone	Address	Resume
0603060108	徐文文	男	1987-12-10 00:00:00...	材料科学0601	0731_20223388	湖南省长沙市韶山北路	NULL
0603060109	黄正刚	男	1987-12-26 00:00:00...	材料科学0601	NULL	贵州省平坝县夏云中学	NULL
0603060110	张红飞	男	1988-03-29 00:00:00...	材料科学0601	NULL	河南省焦作市西环路26号	NULL
0603060111	曾莉娟	女	1987-05-13 00:00:00...	材料科学0601	NULL	湖北省天门市多宝镇公益村六组	NULL
2001050105	邓红艳	女	1986-07-03 00:00:00...	法学0501	NULL	广西桂林市兴安县溶江镇司门街	NULL
2001050106	金萍	女	1984-11-06 00:00:00...	法学0502	NULL	广西桂林市社坡福和11队	NULL
2001050107	吴中华	男	1985-04-10 00:00:00...	法学0503	NULL	河北省邯郸市东街37号	NULL
2001050108	王铭	男	1987-09-09 00:00:00...	法学0504	NULL	河南省上蔡县大路李乡涧沟王村	2003年获县级三好学生
2001060103	郑远月	男	1986-06-18 00:00:00...	法学0601	0731_88837342 ...	湖南省邵阳市一中	2003年获市级三好学生
2001060104	张力明	男	1987-08-29 00:00:00...	法学0602	8834123	安徽省太湖县北中镇桐山村	NULL
2001060105	张好然	女	1988-04-19 00:00:00...	法学0603	010_86634234 ...	北京市西城区新街口外大街34号	NULL
2001060106	李娜	女	1988-10-21 00:00:00...	法学0604	13518473581	重庆市黔江中学	NULL
2602060105	杨平娟	女	1988-05-20 00:00:00...	口腔(七)0601	NULL	北京市西城区复兴门内大街97号	NULL
2602060106	王小维	男	1987-12-11 00:00:00...	口腔(七)0601	NULL	泉州泉秀花园西区十二幢	NULL
2602060107	刘小玲	女	1988-05-20 00:00:00...	口腔(七)0601	NULL	厦门市前埔二里42号0306室	NULL
2602060108	何邵阳	男	1987-06-01 00:00:00...	口腔(七)0601	NULL	广东省韶关市广东北江中学	NULL
0603060201	张红飞	男	1988-03-29 00:00:00...	材料科学0602	NULL	河南省焦作市西环路26号	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

图 3.10 学生信息表 St_Info 的数据

操作步骤如下：

(1) 在“对象资源管理器”中，展开要修改的数据表所在的数据库，选择要修改表结构的 St_Info 数据表并右击，在弹出的快捷菜单中单击“编辑前 200 行”命令，如图 3.11 所示。此时在 SSMS 主窗口的主菜单栏中，出现“查询设计器”菜单，并提供相关的命令操作数据表。

(2) 再打开与图 3.10 相似（因刚建好数据结构的表中，此时还无数据）的表数据窗口，可以对数据表中的数据进行添加、修改和删除等操作。

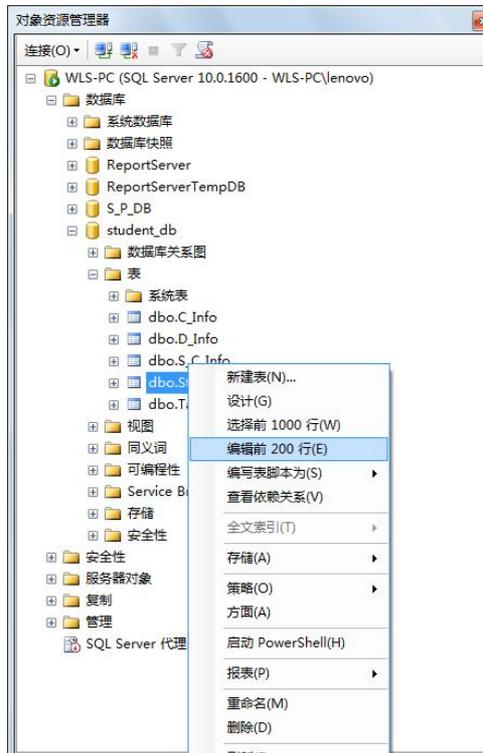


图 3.11 选择“编辑前 200 行”命令

1) 在表数据窗口添加记录的操作方法是：在表数据输入框内输入一条记录，单击第二行，继续输入记录，重复此操作，直到输入完全部记录。单击工具栏上的“执行 SQL”按钮, 保存表数据，然后关闭表数据窗口，此时 St_Info 数据表完全建好。

2) 若要修改数据表中的数据，在表数据窗口先定位要修改的记录字段，然后对该字段值进行修改，修改之后将光标移到下一行即可保存修改的内容。

3) 若要删除数据表中某些记录，在数据窗口先定位要删除的记录行，单击该行最前面的黑色箭头选择全行并右击，在弹出的快捷菜单中选择“删除”命令，如图 3.12 所示。在弹出的删除确认对话框（图 3.13）中，单击“是”按钮将删除所选择的记录，单击“否”按钮将不删除选择的记录。

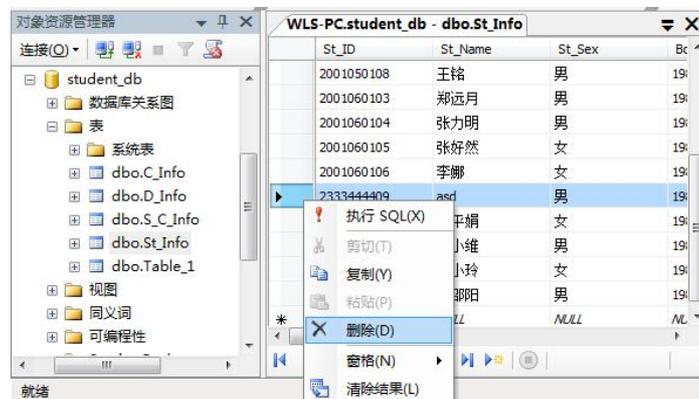


图 3.12 选择“删除”命令



图 3.13 删除记录确认对话框

2. 在数据库关系图中管理数据

数据库关系图是以图形方式显示部分或全部数据库结构的关系图。关系图可用来创建和修改表、列、关系、键、索引、约束。为使数据库可视化，可创建一个或多个关系图，以显示数据库中的部分或全部的表、列、键、关系。操作步骤如下：

(1) 在“对象资源管理器”中展开要操作的数据库，右击“数据库关系图”对象，在弹出的快捷菜单中单击“新建数据库关系图”命令，如图 3.14 所示。

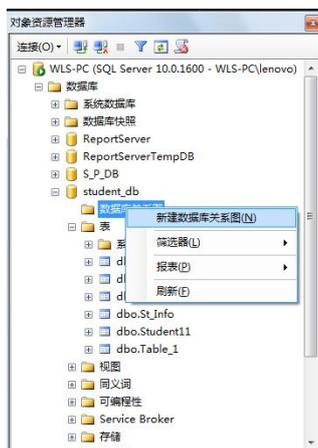


图 3.14 选择“新建数据库关系图”命令

(2) 弹出“添加表”对话框，如图 3.15 所示。在此对话框中选择要建立关系的所有表添加到关系图的列表框中，然后按提示操作完成关系图的创建。这里创建的是学生信息 St_Info 表、课程信息 C_Info 表和选课信息 S_C_Info 表的关系图，如图 3.16 所示。由此图可看到 SSMS 主菜单上出现了“表设计器”和“数据库关系图”两个菜单，可以使用这两个菜单下的相关命令对数据表进行管理。

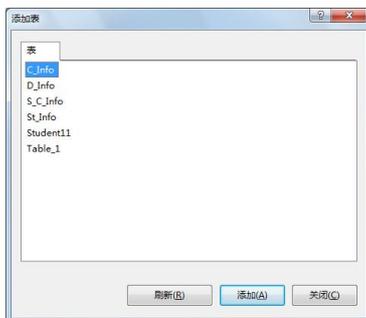


图 3.15 “添加表”对话框

(3) 图 3.16 所示的是 student_db 数据库中一个简单而典型的关系图。在此关系图中, 可以看到表 St_Info 与表 S_C_Info 由一条连接线建立了联系, 表 C_Info 与表 S_C_Info 也由一条连接线建立了联系, 这就是这 3 个表之间的关系, 当鼠标移到该连线上时, 会弹出提示框显示该关系的名称信息。

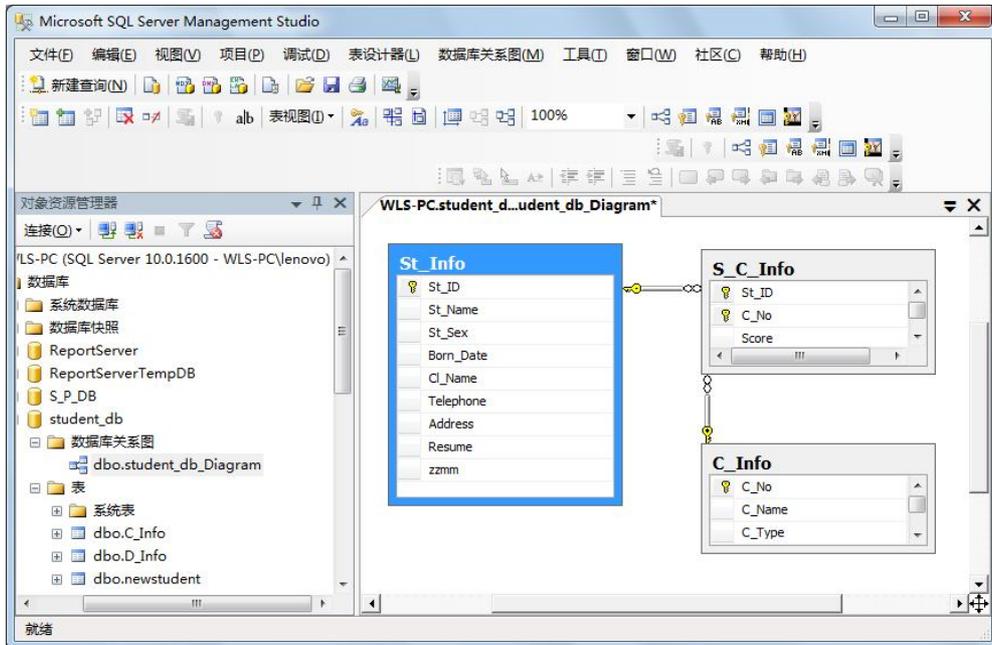


图 3.16 关系图

(4) 当关闭关系图窗格时, 弹出如图 3.17 所示的对话框, 单击“是”按钮, 保存对所列项的更改; 单击“否”按钮, 不保存对所列项的更改; 单击“取消”按钮, 返回关系图的列表框。这里单击“是”按钮, 弹出如图 3.18 所示的对话框。



图 3.17 保存更改对话框

(5) 在“选择名称”对话框的文本框中, 输入数据库关系图的名称“student_db_Diagram”(默认名称是 Diagram_0)。单击“确定”按钮, 完成数据库关系图的创建。



图 3.18 “选择名称”对话框

3.2.2 使用 T-SQL 管理表数据

在“查询设计器”中，可以使用 T-SQL 对表数据进行添加、修改和删除。

1. 表数据的添加

在 T-SQL 中，可以使用 INSERT INTO 语句对数据表进行数据添加，其语法格式如下：

```
INSERT [INTO] table_or_view [(column_list)] data_values
```

参数说明：

(1) [INTO]：一个可选的关键字，可以将它用在 INSERT 和目标表之间。

(2) table_or_view：要输入数据的表名或视图名。

(3) [(column_list)]：要在其中插入数据的一列或多列的列表。必须用圆括号将 column_list 括起来，并且用逗号进行分隔。如果 column_list 没有为表或视图中的所有列命名，将在列表中没有命名的任何列中插入一个 NULL 值（或者在默认情况下为这些列定义的默认值）。在列的列表中没有指定的所有列都必须允许 NULL 值或指定的默认值。

(4) data_values：作为一行或多行插入已命名的表或视图中。

注意：①使用 INSERT 语句一次只能为表插入一行数据。②如果 INSERT 语句违反约束或规则，或者出现与列的数据类型不兼容的值，那么该语句就会失败，并且 SQL Server 将显示错误信息。

【例 3.13】假设数据库 student_db 中已经创建好了课程信息数据表 C_Info（参考 1.4.4 节表结构）。试将新课程记录（9720044，网络技术与应用，选修，3）添加到课程信息数据表 C_Info 中，如图 3.19 所示。

	C_No	C_Name	C_Type	C_Credit	C_Des
	19010122	艺术设计史	选修	5	NULL
	20010051	民法学	必修	6	NULL
	29000011	体育	必修	5	NULL
	9710011	大学计算机基础	必修	2	NULL
	9710021	VB程序设计基础	必修	3	NULL
	9710031	数据库应用基础	必修	3	NULL
	9710041	C++程序设计基础	必修	3	NULL
	9720013	大学计算机基础实践	实践	1	NULL
	9720033	数据库应用基础实践	实践	2	本实践是在学完...
Ⓜ	9720043	C++程序课程设计	实践	2	NULL
	9720044	网络技术与应用	选修	3	NULL
*	NULL	NULL	NULL	NULL	NULL

图 3.19 课程信息数据表

添加新课程记录的语句如下：

```
INSERT INTO C_Info VALUES ('9720044','网络技术与应用','选修',3,null)
```

注意: 由于课程号、课程名和课程类别均为字符型数据, 所以需加上单引号, 而课程说明无值, 所以用空值填上。

【例 3.14】 在 C_Info 表中添加一条新记录, 课程类别和课程说明暂缺。

根据题意知, 要求添加 3 个字段的数据, 使用以下语句:

```
INSERT INTO C_Info(C_NO,C_Name,C_Credit)
VALUES ('9720045','Web 开发技术',2)
```

注意: ①此时必须列出列名 (因为提供的值的个数与表中列的个数不一致)。②C_Info 中的课程类别和课程说明列必须允许为 NULL, 因为系统实际插入的数据为('9720045','Web 开发技术',null,2,null)。

2. 表数据的修改

在 T-SQL 中, 修改数据用到的是 UPDATE 语句, 其语法格式如下:

```
UPDATE
    { table_name | view_name }
SET
    { column_name = { expression | DEFAULT | NULL }
    | @variable = expression
    | @variable = column = expression } [ ,...n ]
[ FROM { < table_source > } [ ,...n ] ]
```

参数说明:

(1) table_name | view_name: 要修改数据的表名或视图名。

(2) SET 子句: 引出后面的赋值表达式。

(3) { column_name = { expression | DEFAULT | NULL } }: 指定要更改数据的列的名称或变量名称和它们的新值, 也可指定使用对列定义的默认值替换列中的现有值。如果该列没有默认值并且定义为允许空值, 也可用来将列更改为 NULL。

(4) @variable = expression: 已声明的变量, 该变量将设置为 expression 所返回的值。

(5) @variable = column = expression } [,...n]: 将变量设置为与列相同的值。

(6) [FROM { < table_source > } [,...n]]: 指定修改的数据将来自一个或多个表或视图。

【例 3.15】 将数据表 C_Info 中的所有学分加 1。

这是无条件修改数据, 使用语句如下:

```
UPDATE C_Info SET C_Credit=C_Credit+1
```

【例 3.16】 将数据表 C_Info 中课程号为“9710011”的学分减 1。

这是有条件修改数据, 使用语句如下:

```
UPDATE C_Info SET C_Credit=C_Credit-1
WHERE C_NO='9710011'
```

3. 表数据的删除

当确定数据表中有些记录不需要时, 就可以将其删除。在 SQL Server 2008 中删除记录的 T-SQL 语句有 DELETE 或 TRUNCATE TABLE。

(1) DELETE 语句

其语法格式如下:

```
DELETE [ FROM ]
    { table_name | view_name
    }
[ FROM { < table_source > } [ ,...n ] ]
```

参数说明:

1) [FROM]: 是可选的关键字, 可用在 DELETE 关键字与目标 table_name、view_name 之间。

2) table_name | view_name: 要删除的行的表名或视图名。

3) [FROM { < table_source > } [,...n]]: 指定删除时用到的额外的表或视图及连接的条件。

(2) TRUNCATE TABLE 语句

其语法格式如下:

```
TRUNCATE TABLE
    [ { database_name.[ schema_name ]. | schema_name . } ]
    table_name [ ; ]
```

参数说明:

1) database_name: 数据库的名称。

2) schema_name: 表所属架构的名称。

3) table_name: 要截断的表的名称, 或要删除其全部行的表的名称。

TRUNCATE TABLE 语句在功能上与不带 WHERE 子句的 DELETE 语句相同: 二者均删除表中的全部行。TRUNCATE TABLE 比 DELETE 速度快, 且使用的系统和事务日志资源少。DELETE 语句每次删除一行, 并在事务日志中为所删除的每行记录一项。TRUNCATE TABLE 通过释放存储表数据所用的数据页来删除数据, 并且只在事务日志中记录页的释放。

TRUNCATE TABLE 删除表中的所有行, 但表结构及其列、约束、索引等保持不变。新行标识所用的计数值重置为该列的种子。如果想保留标识计数值, 则应使用 DELETE 语句。若要删除表定义及其数据, 应使用 DROP TABLE 语句。

【例 3.17】设学生数据库 student_db 中存在 Table_1 表, 并且有若干记录, 要求删除 Table_1 表中全部记录, 但保留数据表结构。

这是无条件全部删除记录, 使用语句如下:

```
TRUNCATE TABLE student_db.dbo.Table_1
```

【例 3.18】在 C_Info 表中删除课程号为“9720045”的记录。

这是有条件删除记录, 使用语句如下:

```
DELETE FROM C_Info
    WHERE C_NO='9720045'
```

3.3 数据库完整性管理

数据库是一种共享资源。因此, 在数据库的使用过程中保证数据的安全、可靠、正确、可用就成为非常重要的问题。数据库的完整性保护可以保证数据的正确性和一致性。读者可以通过学习约束、规则和触发器等技术来充分认识到保证数据库完整性的重要性。

3.3.1 数据库完整性概述

数据的完整性是指数据库中数据的正确性、有效性和一致性。

- (1) 正确性: 指数据的合法性, 如数值型数据只能包含数字, 不能包含字母。
- (2) 有效性: 指数据是否处在定义域的有效范围之内。
- (3) 一致性: 指同一事实的两个数据应该一致, 不一致即是不相容的。

通俗地讲,就是限制数据库中的数据表可输入的数据,防止数据库中存在不符合语义规定的数据和因错误信息的输入输出造成的无效操作或错误信息。使用数据库完整性可确保数据库中的数据质量。例如,如果在学生信息表中规定学号为主关键字,那么在数据表中若已经有了 2001060106 的学号,则该数据库不应允许其他学生再使用该学号;如果计划将 ST_ID 列的范围设定为从 2001060100~2001060199,则数据库不应接受此区间之外的值,如 2001060200 就是一个不符合要求的数据。

对数据库中的数据设置某些约束机制,这些添加在数据上的语义约束条件称为数据库完整性约束条件,完整性约束条件是完整性控制机制的核心。完整性约束保证授权用户对数据库进行修改时不会破坏数据的一致性,从而保护数据库不受意外的破坏。

数据库系统是对现实系统的模拟,现实系统中存在各种各样的规章制度,这些规章制度可以看成是对数据的某种约束。它们定义关于列中允许值的规则,是强制完整性的标准机制。

完整性约束条件作用的对象可以是关系、元组和列 3 种,其状态可以是静态的,也可以是动态的。静态约束是指数据库每一个确定状态时的数据对象所应满足的约束条件,是反映数据库状态合理性的约束,是最重要的一类完整性约束;动态约束是指数据库从一种状态转变为另一种状态时,新、旧值之间所应该满足的约束条件,是反映数据库状态变迁的约束。常说的实体完整性约束、参照完整性约束和域完整性约束均属于静态约束。

数据库系统必须提供一种机制来检验数据库中的数据完整性,看其是否满足语义规定的条件,这种机制称为完整性检查。为此,数据库管理系统的完整性控制机制应该具有以下 3 个方面的功能,防止用户在使用数据库时,输入不合法或不符合语义规则的数据。

(1) 定义功能。提供定义完整性约束条件的机制。

(2) 验证功能。检查用户发出的操作请求是否违背了完整性约束条件。

(3) 处理功能。如果发现是用户的操作请求的数据违背了完整性约束条件,则采取一定的行动来保证数据的完整性。

3.3.2 数据库完整性的类型

数据库完整性有 4 种类型:实体完整性、域完整性、引用完整性和用户定义完整性。

1. 实体完整性

实体完整性是指一个关系中所有主属性不能取空值。“空值”就是“不知道”或“无意义”的值。如主属性取空值,就说明存在某个不可标识的实体,这与现实世界的应用环境相矛盾,因此这个实体一定不是一个完整的实体。

在关系数据库中,空值实际上是一个占位符,它表示“该属性的值是未知的,可能是值域中的任意值”。例如,某个学生的某科成绩为 0 和某科成绩为 NULL 是不同的含义。成绩为 0 表示该学生的该科成绩已经有了,是 0 分;而为 NULL 则表明该成绩还没有被填入。这是两个不同的概念。

实体完整性可以通过标识列、主键约束、唯一性约束以及建立唯一性索引等措施来实现。

(1) 标识列 (IDENTITY)。每个表都可以有一个标识列。每次向表中插入一条记录时,SQL Server 都会根据 IDENTITY 的参数 (初始值、步长值) 自动生成唯一的值作为标识列的值。

(2) 主键约束 (PRIMARY KEY)。主键约束指定表的一列或几列的组合能唯一地标识一行记录。在规范化的表中,每行中的所有数据值都完全依赖于主键,在创建或修改表时可通过

定义 PRIMARY KEY 约束来创建。每个表中只能有一个主键。IMAGE 和 TEXT 数据类型的列不能被指定为主键，也不允许指定的主键列有 NULL 属性。

(3) 唯一性约束 (UNIQUE)。唯一性约束指定一个或多个列的组合的值具有唯一性，以防止在列中输入重复的数据。

(4) 唯一性索引 (UNIQUE INDEX)。数据库中的数据在使用过程中有些原本不相同的数据有可能变成相同数据，这种情况可能会产生错误，可以通过建立唯一性索引来实现数据的实体完整性。

2. 域完整性

域完整性也称列完整性，用于限制用户向列中输入的内容，即保证表的某一列的任何值是该列域（即合法的数据集合）的成员。强制域有效性的方法有限制类型（数据类型、精度、范围、格式和长度等）、使用约束（CHECK 约束、DEFAULT 约束、NOT NULL 约束）和创建规则、默认值等数据库对象来实施。

(1) 限制类型数据库中存储的数据多种多样，为每一列指定一个准确的数据类型是设计表的第一步，列的数据类型规定了列上允许的数据值。当添加或修改数据时，其类型必须要符合建表时所指定的数据类型。这种方式为数据库完整性提供了最基本的保障。

(2) 使用约束是 SQL Server 提供的自动保持数据库完整性的一种方法，是独立于表结构的。约束的方式有以下几种：

1) CHECK 约束（检查约束）。通过约束条件表达式来限制列上可以接受的数据值和格式。

2) DEFAULT 约束（默认约束）。数据库中每一行记录的每一列都应该有一个值。当然这个值也可以是空值，当向表中插入数据时，如果用户没有明确给出某一列的值，SQL Server 自动为该列添加空值，这样可以减少数据输入的工作量。

3) NOT NULL 约束。空值 (NULL) 意味着数据尚未输入。它与 0 或长度为零的字符串 ("") 的含义不同。如果某一列必须有值才能使记录有意义，那么可以指明该列不允许取空值。

(3) 规则 (rule) 就是创建一套准则，可以绑定到一列或多列上，也可以绑定到用户自定义数据类型上。规则和检查约束在使用上的区别是：检查约束可以对一列或多列定义多个约束，而列或用户自定义数据类型只能绑定一个规则。列可以同时绑定一个规则和多个约束；表 CHECK 约束不能直接作用于用户自定义数据类型，它们是相互独立的，但表或用户自定义对象的删除修改不会对与之相联的规则产生影响。

(4) 默认值 (default) 是一种数据库对象，如果在插入行时没有指定列的值，那么由默认值指定列中所使用的值。默认值可以是任何取值为常量的对象，可以绑定到一列或多列上，也可以绑定到用户自定义数据类型上，其作用类似于默认约束，默认约束是在 CREATE TABLE 或 ALTER TABLE 语句中定义的，删除表的时候默认约束也随之被删除了。默认值作为一种单独的数据库对象是独立于表的，删除表不能删除默认值约束。

3. 引用完整性

引用完整性也称为参照完整性，是用来维护相关数据表之间数据一致性的手段。通过实现引用完整性，可以避免因一个数据表的记录改变而使另一个数据表内的数据变成无效的值。引用完整性约束是指引用关系中外码的取值是空值（外码的每个属性值均为空值）或是被引用关系中某个元组的主码值。

4. 用户定义完整性

用户定义完整性使用户得以定义不属于其他任何完整性分类的特定业务规则。所有的完整性类型都支持用户定义完整性。

3.3.3 使用对象资源管理器实现数据库完整性的设置

在 SQL Server 环境下,既可以通过对象资源管理器设置数据库完整性,也可以使用 T-SQL 描述数据库完整性。下面介绍通过对象资源管理器设置数据库完整性的操作方法。

1. 主键约束操作

【例 3.19】使用对象资源管理器,在选课信息表 S_C_Info 中,设置 St_Id 和 C_No 为选课信息表的主关键字,并尝试输入某个不存在的学生学号,验证数据库系统如何实现对实体完整性的保护。

操作步骤如下:

(1) 在“对象资源管理器”中选择要修改的 S_C_Info 数据表并右击,在弹出的快捷菜单中单击“设计”命令。

(2) 在打开的“表设计器”中,选中需设置成主键的第 1 个字段,再按住 Shift (或 Ctrl) 键选择第 2 个字段(因此表的主键是由 St_Id 和 C_No 两个字段构成的)并右击,在弹出的快捷菜单中选择“设置主键”命令,此时可看到设置为主键的字段前面带有  标识,如图 3.20 所示。

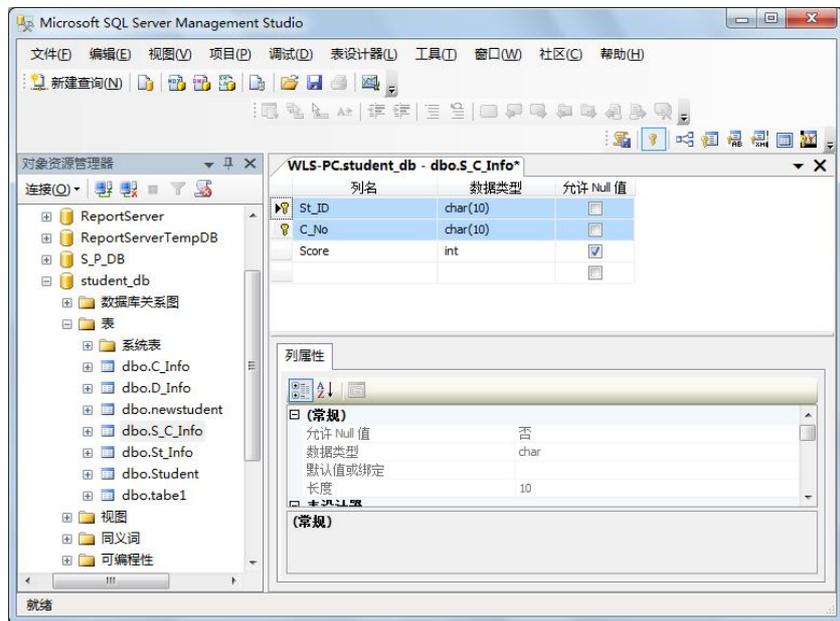


图 3.20 “st_id 和 c_no”设置为主键

注意:也可以单击工具栏中的“设置主键”按钮 ,对选择的字段设置主键。

(3) 关闭“表设计器”。

(4) 在表窗格中右击课程表 S_C_Info,在弹出的快捷菜单中单击“编辑前 200 行”命令。

(5) 在打开的课程表 S_C_Info 中输入某个不存在的学生学号(3602060108),课程编号(9710031)和成绩(99)新记录,如图 3.21 所示。

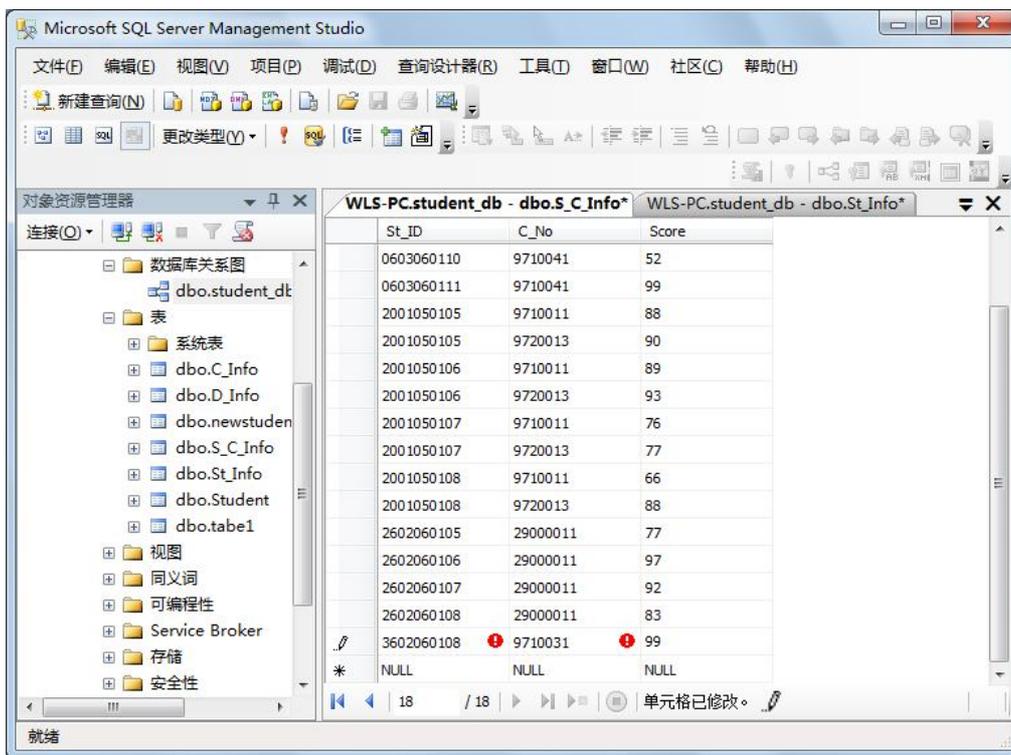


图 3.21 输入不存在的学生学号

(6) 输入该条新记录后单击其他任意一条记录，此时系统提示出错，DBMS 对数据的实体完整性管理得以体现，如图 3.22 所示。



图 3.22 系统提示出错

2. CHECK 约束操作

【例 3.20】 使用对象资源管理器设置 CHECK 约束，将 C_Info 课程信息表中 C_Credit (学分) 的取值范围设置在 1~8 之间，并输入取值范围之外的值来验证数据库系统对域完整性的保护。

操作步骤如下：

(1) 创建 CHECK 约束。在“对象资源管理器”中选择要创建 CHECK 约束的数据表

C_Info 并右击，在弹出的快捷菜单中单击“设计”命令。

(2) 在打开的“表设计器”中指向 C_Credit 字段并右击，在弹出的快捷菜单中单击“CHECK 约束”命令，打开“CHECK 约束”对话框，如图 3.23 所示。

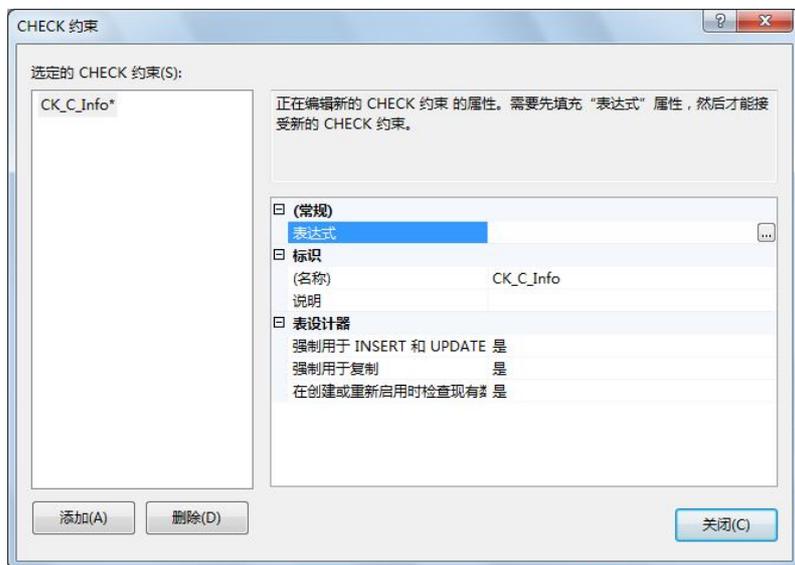


图 3.23 “CHECK 约束”对话框

(3) 在打开的“CHECK 约束”对话框中，单击“添加”按钮，在“选定的 CHECK 约束”栏下，系统自动添加可用的 CHECK 约束“CK_C_Info*”，然后单击“表达式”右边的  按钮，在弹出的“CHECK 约束表达式”对话框中输入约束表达式“C_Credit>= 1 and C_Credit<= 8”，如图 3.24 所示，也可以直接在“表达式”右边的文本框中输入约束表达式。单击“确定”按钮，返回“CHECK 约束”对话框，再单击“关闭”按钮，返回至“表设计器”。

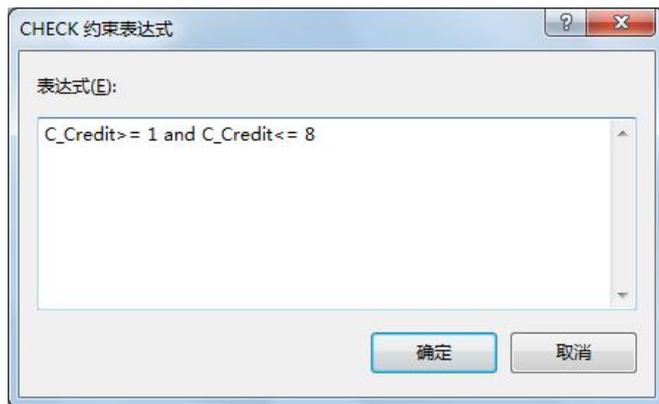


图 3.24 “CHECK 约束表达式”对话框

(4) 单击“表设计器”的关闭按钮，弹出“保存对以下各项的更改吗？”对话框，如图 3.25 所示。单击“是”按钮，关闭“表设计器”。

(5) 验证 CHECK 约束。打开课程信息表 C_Info，输入“C 语言程序设计基础”课程的学分为 9，系统弹出错误信息框，如图 3.26 所示。



图 3.25 “保存对以下各项的更改吗？”对话框



图 3.26 系统弹出的错误信息框

3. 默认值约束操作

【例 3.21】使用对象资源管理器，设置 St_Info 学生信息表中 St_Sex（性别）字段的默认值为“男”。

操作步骤如下：

(1) 在“对象资源管理器”中指向 St_Info 表并右击，在弹出的快捷菜单中单击“设计”命令，打开“表设计器”。

(2) 在打开的“表设计器”中选中 St_Sex（性别）字段，在“表设计器”下面的默认值栏中输入表达式'男'，如图 3.27 所示，关闭表设计器，完成默认值设置。

【例 3.22】使用“对象资源管理器”设置 St_Info 学生信息表中的性别约束为“男或女”。

操作步骤（可参看例 3.20 的操作）如下：

(1) 在“对象资源管理器”中指向 St_Info 表并右击，在弹出的快捷菜单中单击“设计”命令，打开“表设计器”。

(2) 在打开的“表设计器”中，指向 St_Sex 字段并右击，在弹出的快捷菜单中单击“CHECK 约束”命令，打开“CHECK 约束”对话框。

(3) 在打开的“CHECK 约束”对话框中，单击“添加”按钮，在“选定的 CHECK 约束”栏下，系统自动添加可用的 CHECK 约束“CK_St_Info_1*”，然后在“表达式”右边的文本框中输入约束表达式“st_sex like '[男女]’”，如图 3.28 所示。单击“确定”按钮，返回“CHECK 约束”对话框，再单击“关闭”按钮，返回“表设计器”界面。

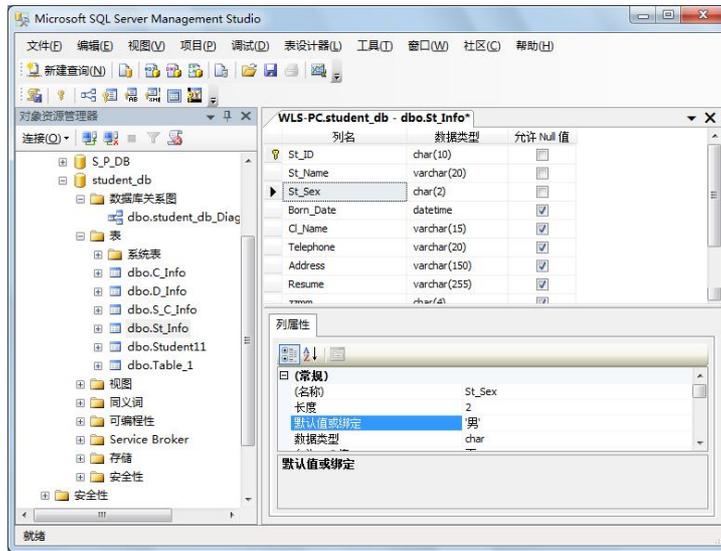


图 3.27 对 St_Sex 字段设置默认值为'男'

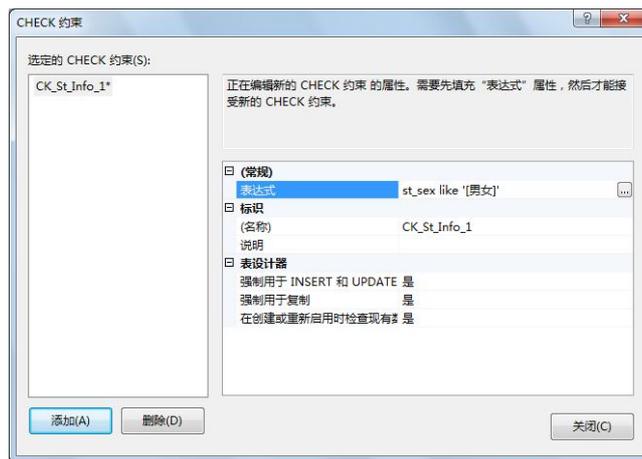


图 3.28 设置约束名为 CK_St_Info_1 的对话框

(4) 单击“表设计器”的关闭按钮，在弹出的“保存对以下各项的更改吗？”对话框中，单击“是”按钮，关闭“表设计器”，完成设置性别约束为“男或女”的操作。

习题 3

一、思考题

- (1) 数据通常存储在什么对象中？表对象存储在什么文件中？什么用户可以对表对象进行操作？
- (2) 什么是数据库完整性？数据库完整性包括哪些内容？为什么要使用数据库完整性？
- (3) 假定利用 CREATE TABLE 命令建立下面的 BOOK 表：

```
CREATE TABLE BOOK
( 总编号 char(6),
```

分类号 char(6),
 书名 char(6),
 单价 numeric(10,2))

则“单价”列的数据类型是什么？列宽度是多少？是否有小数位？

(4) 在 SQL Server 中删除数据表和删除表数据是一个问题吗？为什么？若要删除表的定义及其数据，应使用什么语句？

(5) 什么是实体完整性？实体完整性可通过什么措施实现？主键约束和唯一性约束有什么区别？

二、选择题

- (1) “表设计器”的“允许空”单元格用于设置该字段是否可以输入空值，实际上就是创建该字段的 () 约束。
- A. 主键 B. 外键 C. NULL D. CHECK
- (2) 创建一个数据表时，可以指定的约束类型中不包含 ()。
- A. 主键约束 B. 唯一性约束 C. 共享性 D. 外键约束
- (3) 下列关于主关键字的叙述正确的是 ()。
- A. 一个表可以没有主关键字
 B. 只能将一个字段定义为主关键字
 C. 如果一个表只有一个记录，则主关键字字段可以为空值
 D. 都正确
- (4) 下列语句用来删除表对象或表数据，其中不正确的语句是 ()。
- A. truncate table book
 B. Delete * from book
 C. drop table book
 D. delete from book
- (5) CREATE TABLE 语句 ()。
- A. 必须在数据表名称中指定表所属的数据库
 B. 必须指明数据表的所有者
 C. 指定的所有者和表名称组合起来在数据库中必须唯一
 D. 省略数据表名称时，则自动创建一个本地临时表
- (6) 删除数据表的语句是 ()。
- A. DROP B. ALTER C. UPDATE D. DELETE
- (7) 数据库完整性不包括 ()。
- A. 实体完整性 B. 程序完整性 C. 域完整性 D. 用户自定义完整性
- (8) 下面关于 INSERT 语句的说法，正确的是 ()。
- A. INSERT 一次只能插入一行的元组 B. INSERT 只能插入，不能修改
 C. INSERT 可以指定要插入到哪行 D. INSERT 可以加 WHERE 条件
- (9) 表数据的删除语句是 ()。
- A. DELETE B. INSERT C. UPDATE D. ALTER
- (10) SQL 数据定义语言中，表示外键约束的关键字是 ()。
- A. CHECK B. FOREIGN KEY C. PRIMARY KEY D. UNIQUE