

本任务主要介绍如何通过 SQL 语句对数据库中表的数据进行管理,包括基本的 SQL 语句、SQL Server 中的逻辑表达式、运算符以及如何使用 SQL 语句实现表中数据的插入、更新、删除功能。

## 任务4 管理学生成绩管理数据库中的数据

### 4.1 客户需求

要使用“学生成绩管理系统”软件对学生信息进行管理,需要把学生的基本信息录入到系统的数据库中。如果信息录入后,学生的信息发生变化,则需要对数据进行修改,如果有学生中途退学,要及时把相关信息删除。

### 4.2 任务分析

#### 【任务分析】

(1)把学生的基本信息录入到学生成绩管理数据库中,首先要知道学生的哪些信息要录入到系统,数据放入哪个表中,并使用 Insert 语句把数据插入表中。

(2)如果某个学生的信息发生变化,要对该生的信息进行修改。由于每个学生的学号是唯一的,因此首先获取该生的学号,并通过学号定位要修改的记录,使用 Update 语句进行修改。

(3)当某个学生中途退学时,要及时把该生的信息删除,也要首先通过学号定位记录,使用 delete 语句删除该生的信息。

#### 【任务描述】

(1)掌握 SQL(结构化查询语句)的基本语法。

(2)使用 SQL 语句实现对学生成绩管理数据库中数据的增加、修改和删除操作。

## 4.3 制定计划

### 【解决方案】

- (1) 根据本任务进行分组,小组成员讨论并制定具体实施步骤,明确任务分工。
- (2) 实现学生的基本信息录入(8分钟)、学生的信息修改(10分钟)和学生信息删除(8分钟)功能。
- (3) 每组提交一份对学生信息进行录入、修改和删除的 SQL 语句脚本。

## 4.4 信息收集

### 4.4.1 SQL 简介

SQL 全称是“结构化查询语言(Structured Query Language)”,最早是由 IBM 的圣约瑟研究试验室为其关系数据库管理系统 SYSTEM R 开发的一种查询语言,它的前身是 SQUARE 语言。SQL 语言结构简洁、功能强大、简单易学,所以自从 IBM 公司 1981 年推出以来,SQL 语言得到了广泛的应用。

目前 SQL 语言已被确定为关系数据库系统的国际标准,被绝大多数商品化关系数据库系统采用,如 Oracle、Sybase、DB2、Informix、SQL Server,这些数据库管理系统都支持 SQL 语言作为其查询语言。

SQL 是一种介于关系代数与关系演算之间的语言,其功能包括查询、操纵、定义和控制四个方面,是一个通用的功能极强的关系数据库标准语言。

T-SQL 语言是微软公司对 SQL 语言的扩展。

(1) SQL 是非过程化语言,它一次处理一个记录,对数据提供自动导航。

所有 SQL 语句接受集合作为输入,返回集合作为输出。SQL 的集合特性允许一条 SQL 语句的结果作为另一条 SQL 语句的输入。

SQL 不要求用户指定对数据的存放方法,这种特性使用户更易集中精力于要得到的结果。

所有 SQL 语句使用查询优化器,由它决定对指定数据存取的最快速度的手段。查询优化器知道存在什么索引、哪儿使用合适,而用户从不需要知道表是否有索引、表有什么类型的索引。

(2) SQL 可用于所有用户的数据库活动模型,包括系统管理员、数据库管理员、应用程序员、决策支持系统人员及许多其他类型的终端用户。基本的 SQL 命令只需很少时间就能学会,最高级的命令在几天内便可掌握。SQL 为许多任务提供了命令,包括:

- ◇ 查询数据。
- ◇ 在表中插入、修改和删除记录。
- ◇ 建立、修改和删除数据对象。

- ◇ 控制对数据和数据对象的存取。
- ◇ 保证数据库一致性和完整性。

(3)所有主要的关系数据库管理系统都支持 SQL 语言,用户可将使用 SQL 的技能从一个 RDBMS 转到另一个。所有用标准 SQL 编写的程序都是可以移植的。

(4)SQL 包含 4 个部分:

- ◇ 数据定义语言 DDL(Data Definition Language):用来定义数据的结构,如 CREATE、ALTER、DROP 语句。
- ◇ 数据控制语言 DCL(Data Control Language):用来控制数据库组件的存取许可、存取权限,如 GRANT、REVOKE 语句。
- ◇ 数据操纵语言 DML(Data Manipulation Language):用来操纵数据库中的数据,如 INSERT、UPDATE、DELETE 语句。
- ◇ 数据查询语言 DQL(Data Query Language):用来查询数据库中的数据,如 SELECT 语句。

#### 4.4.2 INSERT 语句

INSERT 语句用于向数据库表中插入数据,可向表中添加一行或者多行记录。

INSERT 语句的语法格式如下:

```
INSERT [INTO]表名 [(列 1,列 2,...)] VALUES (值 1,值 2,...)
```

#### 4.4.3 UPDATE 语句

UPDATE 语句可以修改表或视图中的单行、多行或所有行的数据值。

UPDATE 语句的语法如下:

```
UPDATE 表名 SET 列 1=表达式 1[,列 2=表达式 2,...][WHERE 条件]
```

WHERE 子句中有多个条件时,使用逻辑运算符 AND、OR、NOT 或更多谓词(如 LIKE、IN、BETWEEN 等)的组合。

#### 4.4.4 DELETE 语句

DELETE 语句可以删除数据库表或视图中的一个或者多个记录。

DELETE 语句的语法格式如下:

```
DELETE FROM 表名 [WHERE 条件]
```

如果 DELETE 语句中不带条件,则删除表中全部记录。

#### 4.4.5 TRUNCATE 语句

TRUNCATE TABLE 语句提供了一种删除表中所有记录的快速方法,因为 TRUNCATE TABLE 语句不记录日志,只记录整个数据页的释放操作,而 DELETE 语句对每一行的修改都记录日志。

TRUNCATE TABLE 语句的语法格式如下:

```
TRUNCATE TABLE 表名
```

执行的结果是删除表中所有行。

### 提示

TRUNCATE、DELETE、DROP 的区别：

(1) TRUNCATE 语句和 DELETE 语句(不含 WHERE 条件)只删除数据,不删除表的结构。

(2) DROP 语句不仅删除表数据,还将删除表的结构。

## 4.4.6 SQL Server 中的运算符

运算符用来指定在一个或多个表达式中的操作。SQL Server 2008 中使用算术运算符、赋值运算符、位运算符、比较运算符、逻辑运算符、字符串串联运算符和一元运算符。

(1) 算术运算符。

算术运算符用来在两个表达式上执行数学运算,这两个表达式可以是任意两个数值类型的表达式。算术运算符包括+(加)、-(减)、\*(乘)、/(除)、%(模)五个。

(2) 赋值运算符。

Transact-SQL 有一个赋值运算符,即等号(=)。

例如,下面的代码创建了 @MyCounter 变量。然后赋值运算符将 @MyCounter 设置成一个由表达式返回的值。

```
declare @MyCounter int
set @MyCounter=1
```

(3) 位运算符。

位运算符在两个表达式之间执行位操作,这两个表达式可以是任意两个整型数据类型的表达式。位运算符的符号及其定义如表 4-1 所示。

表 4-1 位运算符

| 运算符 | 含义   |
|-----|------|
| &   | 按位与  |
|     | 按位或  |
| ^   | 按位异或 |
| ~   | 按位取反 |

(4) 比较运算符。

比较运算符用来测试两个表达式是否相同。除了 text、ntext 或 image 数据类型的表达式外,比较运算符可以用于所有的表达式。

表 4-2 比较运算符

| 运算符 | 含义   |
|-----|------|
| =   | 等于   |
| >   | 大于   |
| <   | 小于   |
| >=  | 大于等于 |
| <=  | 小于等于 |
| <>  | 不等于  |

比较运算符的结果是布尔数据类型,它有三种值:TRUE、FALSE 和 NULL。那些返回布尔数据类型的表达式被称为布尔表达式。

在 WHERE 子句中使用带有布尔数据类型的表达式,可以筛选出符合搜索条件的行。

#### (5)逻辑运算符。

逻辑运算符用来对某个条件进行测试,返回带有 TRUE 或 FALSE 值的布尔数据类型。

表 4-3 逻辑运算符

| 运算符     | 含义                            |
|---------|-------------------------------|
| ALL     | 如果一系列的比较都为 TRUE,那么就为 TRUE     |
| AND     | 如果两个布尔表达式都为 TRUE,那么就为 TRUE    |
| ANY     | 如果一系列的比较中任何一个为 TRUE,那么就为 TRUE |
| BETWEEN | 如果操作数在某个范围之内,那么就为 TRUE        |
| EXISTS  | 如果子查询包含一些行,那么就为 TRUE          |
| IN      | 如果操作数等于表达式列表中的一个,那么就为 TRUE    |
| LIKE    | 如果操作数与一种模式相匹配,那么就为 TRUE       |
| NOT     | 对任何其他布尔运算符的值取反                |
| OR      | 如果两个布尔表达式中的一个为 TRUE,那么就为 TRUE |
| SOME    | 如果在一系列比较中,有些为 TRUE,那么就为 TRUE  |

#### (6)一元运算符。

一元运算符只对一个表达式执行操作,这个表达式可以是数值类型中的任何一种数据类型。

表 4-4 一元运算符

| 运算符 | 含义      |
|-----|---------|
| +   | 数值为正    |
| -   | 数值为负    |
| ~   | 返回数字的补数 |

#### (7)运算符优先级(Transact-SQL)。

当一个复杂的表达式有多个运算符时,运算符优先级决定执行运算的先后次序,在较低级别的运算符之前先对较高级别的运算符进行求值。

表 4-5 运算符优先级

| 级别 | 运算符(自上而下优先级逐渐降低)                       |
|----|--|
| 1  | ~(位非)                                  |
| 2  | *(乘)、/(除)、%(取模)                        |
| 3  | +(正)、-(负)、+(加)、-(减)、&(位与)、^(位异或)、 (位或) |
| 4  | =、>、<、>=、<=、<>                         |
| 5  | NOT                                    |
| 6  | AND                                    |
| 7  | ALL、ANY、BETWEEN、IN、LIKE、OR、SOME        |
| 8  | =(赋值)                                  |

当一个表达式中的两个运算符有相同的运算符优先级时,将按照它们在表达式中的位置

对其从左到右进行求值。

#### 4.4.7 SQL Server 中的通配符

在搜索数据库中的数据时,SQL 通配符可以替代一个或多个字符。SQL 通配符必须与 LIKE 运算符一起使用。在 SQL 中,可使用表 4-6 列出的通配符。

表 4-6 通配符

| 运算符  | 用法                 | 示例                    |
|------|--------------------|-----------------------|
| _    | 匹配一个字符             | A Like 'C_'           |
| %    | 匹配任意长度的字符串         | B Like 'CO_%'         |
| [ ]  | 匹配括号中所指定范围内的一个字符   | C Like '9W0[1-2]'     |
| [^ ] | 匹配不在括号中所指定范围内的一个字符 | D Like '%[A-D][^1-2]' |

#### 提示

通配符通常与 LIKE 关键字一起来使用。

### 4.5 任务实施

(1) 向学生成绩管理系统中插入姓名为“郭富城”的学生基本信息。

```
insert into Students (SNumber,SName,SAddress,SGrade,SEmail)
values ('100861','郭富城','保定电院',901,'gfc@163.com')
```

单击“执行”按钮或者按【F5】键,执行结果如图 4-1 所示。



图 4-1 插入一行数据后的结果

① 每次插入一行数据,不可能只插入半行或者几列数据,因此,插入的数据是否有效将按照整行的完整性要求来检验。

```
insert into Students(SNumber,SName,SAddress,SGrade,SEmail)
values ('100861')
```

② 每个数据值的数据类型、精度和小数位数必须与相应的列匹配。

```
insert into Students(SNumber,SName,SAddress,SGrade,SEmail)
values ('100861', '郭富城', '保定电院',rj901, 'gfc@163.com')
```

③ 不能为标识列指定值,因为它的值是自动增长的。

```
insert into Score(scoreID,SNumber,CourseID,Score)
values(1,'100861',1,80)
```

④如果在设计表的时候就指定了某列不允许为空,则必须插入数据。

```
insert into Students(SNumber,SAddress,SGrade,SEmail)
values ('100861','保定电院',901,'gfc@163.com')
```

⑤插入的数据项,要求符合检查约束的要求。

```
insert into Students(SNumber,SName,SAddress,SGrade,SEmail)
values('100861','郭富城','保定电院',901,'gfc')
```

⑥具有缺省值的列,可以使用 DEFAULT(缺省)关键字来代替插入的数值。

```
insert into Students(SNumber,SName,SAddress,SGrade,SEmail)
values('100861','郭富城', DEFAULT,901,'gfc@163.com')
```

(2)把学生成绩管理系统中学生的地址“保定电院”改为“保定电力职业技术学院”。

```
update Students set SAddress='保定电力职业技术学院'
where SAddress='保定电院'
```

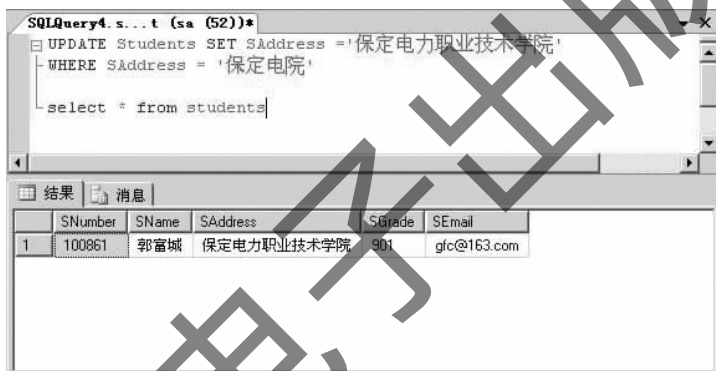


图 4-2 修改数据后的结果

(3)把学生成绩管理系统中分数小于 95 分的学生都加上 5 分。

```
update Scores set Scores=Scores + 5 where Scores <=95
```

(4)把学生成绩管理系统中学号为“100862”的学生删除。

```
delete from Students where SNumber='100862'
```

(5)把学生成绩管理系统中姓“张”的学生删除。

```
delete from Students where Sname like '张%'
```

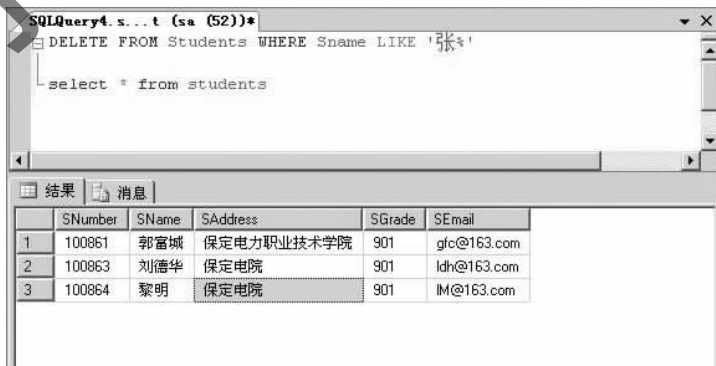


图 4-3 删除数据后的结果

## 4.6 检查与评估

- (1) 学生提交信息录入、修改和删除的 SQL 语句。
- (2) 教师检查录入、修改和删除的 SQL 语句的语法是否准确。
- (3) 教师检查学生信息表的结果是否准确。
- (4) 教师检查语句是否有必要的注释。
- (5) 教师点评。

## 4.7 拓展训练

- (1) 向学生成绩管理系统中添加下表中的数据。

表 4-7 学生信息表

| 学号     | 姓名  | 家庭住址 | 班级   | 电子邮件        |
|--------|-----|------|------|-------------|
| 100861 | 刘德华 | 保定电院 | 0901 | ldh@163.com |
| 100862 | 郭富城 | 保定电院 | 0902 | gfc@163.com |
| 100863 | 黎明  | 保定电院 | 0902 | lm@163.com  |
| 100864 | 张学友 | 保定电院 | 0901 | zxy@163.com |
| 100865 | 赵雅芝 | 保定电院 | 0902 | zyz@163.com |
| 100866 | 张柏芝 | 保定电院 | 0901 | zbz@163.com |

- (2) 向学生成绩管理系统中添加下表中的数据。

表 4-8 成绩信息表

| 学号     | 课程编号 | 成绩 |
|--------|------|----|
| 100861 | 1    | 80 |
| 100862 | 2    | 70 |
| 100863 | 3    | 66 |
| 100864 | 4    | 50 |
| 100865 | 5    | 30 |
| 100866 | 6    | 40 |

- (3) 向学生成绩管理系统中添加下表中的数据。



表 4-9

课程信息表

| 课程编号 | 课程名     |
|------|---------|
| 1    | 英语      |
| 2    | 数学      |
| 3    | 计算机应用基础 |
| 4    | C 语言    |
| 5    | 物理      |
| 6    | 化学      |

(4)把学生成绩管理系统中电子邮件为“lm@163.com”的学生班级改为“0901”。

(5)把学生成绩管理系统中成绩等于 60 的学生全部信息删除。

## 4.8 小结

本任务主要讲述了数据库表中数据管理的相关知识,主要包括:

(1)使用 insert 语句插入数据,在定义了一些设置(如 NULL 值和默认值)和约束的列上插入数据时要注意是否违反约束。

(2)使用 update 语句更新表中的一行或者多行数据数据。

(3)使用 delete 和 truncate table 语句删除数据。