

数据操纵

数据操纵包括三种操作：向表中添加若干行数据、修改表中的数据和删除表中的若干数据。在SQL中分别对应INSERT、UPDATE和DELETE三类语句。

数据插入

当表结构定义好之后，就能够往表里插入数据。SQL的数据插入语句INSERT通常有两种形式，一种是插入一个元组，另一种是插入子查询结果。后者可以一次插入多个元组，实现批量插入数据。本小节主要介绍插入一个元组的INSERT语句，其基本格式如下：

```
INSERT
INTO <表名> [ (<属性列1> [, <属性列2>] ...) ]
VALUES (<常量1> [, <常量2>]...);
```

INTO子句中指定要插入数据的基础表以及要插入的属性列名，VALUES子句指定插入元组的各属性值，其中常量1为新元组属性列1的值，常量2为新元组属性列2的值，……。此外，INTO子句中可以不指定任何属性列名，那么，VALUES子句中新插入的元组必须在每个属性列上均有值，且要按照表定义中属性列的顺序。

[例5.25] 向Student表中插入一个新学生元组（学号：20220013，姓名：宇轩，性别：男，年龄：20）。

```
INSERT
INTO Student
VALUES('20220013','宇轩','男',20);
```

另一种SQL表示

```
INSERT
INTO Student(Sno, Sname, Age, Gender)
VALUES('20220013','宇轩',20,'男');
```

第一种SQL表示中，INTO子句中未指定任何属性列名时，VALUES子句中的新插入的属性列值要与表定义的属性列一致，否则将出错。第二种SQL表示中，INTO子句中指定了属性列名，属性的顺序可以和表定义的顺序不一致，但是VALUES中插入的属性值要与INTO中指定的属性列名对应。另外，VALUES子句中字符串常数要用单引号（英文符号）括起来。

[例5.26] 向SC表中插入一条选课记录（学号：2022013，课程号：1，成绩：NULL）。

```
INSERT
INTO SC(Sno, Cno)
VALUES('2022013','1');
```

另一种SQL表示

```
INSERT
INTO SC
VALUES('2022013','1',NULL);
```

第一种SQL表示中，INTO子句没有指定SC表中的Grade属性列，关系数据库管理系统将在新插入元组的Grade列上自动地赋空值。需要注意的是，表定义时说明了NOT NULL的属性列不能取空值，否则会出错。第二种SQL表

示中，INTO子句没有指定SC表的任何属性列，那么VALUES子句中新元组每个属性列上均要有值，因此Grade列上要明确给出空值。

数据修改

数据的修改操作又称为数据更新。SQL语句的一般格式为：

```
UPDATE <表名>
SET <列名1>=<表达式1> [, <列名2>=<表达式2>...]
[WHERE <条件表达式>]
```

其功能是修改指定表中满足WHERE子句条件的元组。SET子句中<表达式>的值用于取代相应的属性列值。如果省略WHERE子句，则表示修改表中的所有元组。

[例5.27] 将学生2022013的年龄改为19岁。

```
UPDATE Student
SET Age = 19
WHERE Sno = '2022013';
```

[例5.28] 将所有男生的年龄增加1岁。

```
UPDATE Student
SET Age = Age + 1
WHERE Gender = '男';
```

数据删除

删除语句的一般格式为：

```
DELETE
FROM <表名>
[WHERE <条件表达式>]
```

该功能是从指定表中删除满足WHERE子句条件的所有元组。如果省略WHERE子句则表示删除表中的全部元组，但表的定义仍在字典中。也就是说，DELETE语句删除的是表中的数据，而不是关于表的定义。

[例5.29] 删除学号为2022013的学生信息。

```
DELETE
FROM Student
WHERE Sno = '2022013';
```

[例5.30] 删除所有学生的选课记录。

```
DELETE
FROM SC;
```