数据库系统概论
An Introduction to Database System

第三章 关系数据库标准语言 SQL（续 2）

中国人民大学信息学院

第三章　关系数据库标准语言 SQL

3.1 SQL 概述

3.2 学生 - 课程数据库

3.3 数据定义

3.4 数据查询

3.5 数据更新

3.6 视图

3.7 小结

3.5 数 据 更 新

3.5.1 插入数据

3.5.2 修改数据

3.5.3 删除数据

3.5.1 插入数据

- 两种插入数据方式
  1. 插入元组
  2. 插入子查询结果
     - 可以一次插入多个元组

一、插入元组

- 语句格式

  INSERT

  INTO < 表名 > [(< 属性列 1>[， < 属性列 2 >…)]

  VALUES (< 常量 1> [， < 常量 2>] …)

- 功能
  - 将新元组插入指定表中

插入元组（续）

- INTO 子句
  - 属性列的顺序可与表定义中的顺序不一致
  - 没有指定属性列
  - 指定部分属性列
- VALUES 子句
  - 提供的值必须与 INTO 子句匹配
    - 值的个数
    - 值的类型

插入元组（续）

［例 1］ 将一个新学生元组（学号： 200215128； 姓名：陈冬；性别：男；所在系： IS； 年龄： 18 岁）插入到 Student 表中。

```
INSERT
INTO  Student
(Sno ， Sname ， Ssex ， Sdept ， Sage)
VALUES ('200215128' ， ' 陈冬 ' ， ' 男 ' ， 'IS' ， 18) ;
```

插入元组（续）

［例２］　将学生张成民的信息插入到 Student 表中。

```
INSERT
INTO  Student
VALUES (‘200215126’，　‘张成民’，　‘男’，　18，　'CS');
```

插入元组（续）

［例 3 ］ 插入一条选课记录 ('200215128' ， '1 ') 。

```
INSERT
INTO SC(Sno ， Cno)
VALUES (‘ 200215128 ’ ，  ‘ 1 ’) ;
```

RDBMS 将在新插入记录的 Grade 列上自动地赋空值。

或者:

```
INSERT
INTO SC
VALUES (' 200215128 ' ， ' 1 ' ， NULL) ;
```

二、插入子查询结果

- 语句格式

  INSERT

  INTO < 表名 > [(< 属性列 1> [, < 属性列 2>…)]

  子查询;

- 功能

  将子查询结果插入指定表中

插入子查询结果（续）

- INTO 子句 (与插入元组类似)

- 子查询
  - SELECT 子句目标列必须与 INTO 子句匹配
    - 值的个数
    - 值的类型

插入子查询结果（续）

[例 4] 对每一个系，求学生的平均年龄，并把结果存入数据库。

第一步：建表

```
CREATE  TABLE  Dept_age
   (Sdept  CHAR(15)              /*  系名 */
    Avg_age SMALLINT) ;          /* 学生平均年龄 */
```

插入子查询结果（续）

第二步：插入数据

```
INSERT
INTO  Dept_age(Sdept ,  Avg_age)
    SELECT  Sdept ,  AVG(Sage)
    FROM  Student
    GROUP BY Sdept ;
```

插入子查询结果（续）

RDBMS 在执行插入语句时会检查所插元组是否破坏表上已定义的完整性规则

- 实体完整性
- 参照完整性
- 用户定义的完整性
  - NOT NULL 约束
  - UNIQUE 约束
  - 值域约束

3.5 数 据 更 新

3.5.1 插入数据

3.5.2 修改数据

3.5.3 删除数据

3.4.2 修改数据

- 语句格式

  UPDATE < 表名 >
  SET < 列名 >=< 表达式 >[, < 列名 >=< 表达式 >]…
  [WHERE < 条件 >] ;

- 功能
  - 修改指定表中满足 WHERE 子句条件的元组

修改数据（续）

- SET 子句
  - 指定修改方式
  - 要修改的列
  - 修改后取值
- WHERE 子句
  - 指定要修改的元组
  - 缺省表示要修改表中的所有元组

修改数据（续）

- 三种修改方式
  1. 修改某一个元组的值
  2. 修改多个元组的值
  3. 带子查询的修改语句

1. 修改某一个元组的值

[例 5] 将学生 200215121 的年龄改为 22 岁

```
UPDATE  Student
SET Sage=22
WHERE  Sno=' 200215121 ' ;
```

2. 修改多个元组的值

[例 6] 将所有学生的年龄增加 1 岁

```
UPDATE Student
SET Sage= Sage+1 ;
```

3. 带子查询的修改语句

[例 7] 将计算机科学系全体学生的成绩置零。

```
UPDATE SC
SET  Grade=0
WHERE  'CS'=
            (SELETE Sdept
             FROM  Student
             WHERE  Student.Sno = SC.Sno) ;
```

修改数据（续）

RDBMS 在执行修改语句时会检查修改操作是否破坏表上已定义的完整性规则

- 实体完整性
- 主码不允许修改
- 用户定义的完整性
  - NOT NULL 约束
  - UNIQUE 约束
  - 值域约束

3.5 数 据 更 新

3.5.1 插入数据

3.5.2 修改数据

3.5.3 删除数据

3.5.3 删除数据

- 语句格式

  DELETE

  FROM　　< 表名 >

  [WHERE < 条件 >] ;

- 功能
  - 删除指定表中满足 WHERE 子句条件的元组
- WHERE 子句
  - 指定要删除的元组
  - 缺省表示要删除表中的全部元组，表的定义仍在字典中

删除数据（续）

- 三种删除方式

1. 删除某一个元组的值
2. 删除多个元组的值
3. 带子查询的删除语句

1. 删除某一个元组的值

[例 8] 删除学号为 200215128 的学生记录。

```
DELETE
FROM Student
WHERE Sno= 200215128 ' ;
```

2. 删除多个元组的值

[例 9] 删除所有的学生选课记录。

```
DELETE
FROM SC ;
```

3. 带子查询的删除语句

[例 10] 删除计算机科学系所有学生的选课记录。

```
DELETE
FROM SC
WHERE  'CS'=
            (SELETE Sdept
             FROM Student
             WHERE Student.Sno=SC.Sno) ;
```

下课了。。。

休息一会儿。。。