数据库原理及应用

Principle and Application of Database



3.5

关系数据库标准语言SQL

------ 数据更新

SQL语言的数据更新语句：

INSERT、 UPDATE、 DELETE

插入

修改

删除

一、插入数据：INSERT

1、插入单个元组数据

2、插入子查询结果

一、插入数据：INSERT

1、插入单个元组

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

一、插入数据：INSERT

1、插入单个元组

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

- 指定插入元组的表名及属性列
- 属性列的顺序可与表定义中的顺序不一致
- 指定部分属性列：插入的元组在其余属性列上取空值
- 没有指定属性列：表示要插入的是一条完整的元组，且属性列属性与表定义中的顺序一致

一、插入数据：INSERT

1、插入单个元组

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

提供的值必须与INTO子句匹配

- 值的个数
- 值的类型

一、插入数据：INSERT

例1：在Student表插入一条新的学生记录。

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
201815128	陈冬	男	18	IS

一、插入数据：INSERT

例1：在Student表插入一条新的学生记录。

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
201815128	陈冬	男	18	IS

```
INSERT
INTO    Student
VALUES  ('201815128' , '陈冬' , '男' , 18, 'IS' );
```

一、插入数据：INSERT

例2：在SC表插入一条选修记录。

学号 Sno	课程号 Cno
201815122	1

一、插入数据：INSERT

例2：在SC表插入一条选修记录。

学号 Sno	课程号 Cno
201815122	1

```
INSERT
INTO    SC(Sno, Cno)
VALUES  (‘201815122’ , ‘1’ );
```

一、插入数据：INSERT

2、插入子查询结果

INSERT

INTO <表名> [(<属性列1>[，<属性列2 > …)]

子查询

SELECT子句的目标属性列必须与INTO子句匹配

- 值的个数
- 值的类型

一、插入数据：INSERT

例3：将选课学生各自的选课总学分，存入新表。

STcredit

学号 Sno	学分总数 Tcredit

```
CREATE TABLE STcredit
( Sno CHAR(9) PRIMARY KEY,
  Tcredit SMALLINT,
  FOREIGN KEY (Sno) REFERENCE SC(Sno));
```

一、插入数据：INSERT

例3：将选课学生各自的选课总学分，存入新表。

STcredit

学号 Sno	学分总数 Tcredit

```
SELECT  Sno, SUM (Ccredit)
FROM    SC, Course
WHERE   SC.Cno = Course.Cno
GROUP  BY  Sno;
```

选修表SC

学号 Sno	课程号 Cno	成绩 Grade
201815121	1	92
201815121	2	85
201815121	3	88
201815122	2	90
201815122	3	80

课程表Course

课程号 Cno	课程名 Cname	先行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	C语言	6	4

一、插入数据：INSERT

例3：将选课学生各自的选课总学分，存入新表。

STcredit

学号 Sno	学分总数 Tcredit

```
INSERT
INTO    STcredit
SELECT  Sno, SUM (Ccredit)
FROM    SC, Course
WHERE   SC.Cno = Course.Cno
GROUP   BY   Sno;
```

选修表SC

学号 Sno	课程号 Cno	成绩 Grade
201815121	1	92
201815121	2	85
201815121	3	88
201815122	2	90
201815122	3	80

课程表Course

课程号 Cno	课程名 Cname	先行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	C语言	6	4

一、插入数据：INSERT

例3：将选课学生各自的选课总学分，存入新表。

STcredit

学号 Sno	学分总数 Tcredit
201815121	10
201815122	6

```
INSERT
INTO    STcredit
SELECT  Sno, SUM (Ccredit)
FROM    SC, Course
WHERE   SC.Cno = Course.Cno
GROUP   BY   Sno;
```

选修表SC

学号 Sno	课程号 Cno	成绩 Grade
201815121	1	92
201815121	2	85
201815121	3	88
201815122	2	90
201815122	3	80

课程表Course

课程号 Cno	课程名 Cname	先行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	C语言	6	4

二、修改数据：UPDATE

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

二、修改数据：UPDATE

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

修改的属性列

修改后的取值

二、修改数据：UPDATE

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

指定修改的元组需要满足的条件

缺省，表示要修改表中的所有元组

可以带有子查询块

二、修改数据：UPDATE

例4：将学号为“201815121”的学生年龄改为22岁。

```
UPDATE   Student
SET   Sage = 22
WHERE  Sno = ‘201815121’ ;
```

二、修改数据：UPDATE

例4：将学号为“201815121”的学生年龄改为22岁。

```
UPDATE  Student
SET  Sage = 22
WHERE  Sno = '201815121' ;
```

例5：将信息系（IS）所有学生的年龄增加1岁。

```
UPDATE  Student
SET  Sage = Sage +1
WHERE  Sdept = 'IS' ;
```

二、修改数据：UPDATE

例6：将计算机科学系（CS）全体学生的成绩置零。

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

三、删除数据：DELETE

```
DELETE
FROM    <表名>
[ WHERE <条件> ]
```

三、删除数据：DELETE

```
DELETE
FROM    <表名>
[ WHERE <条件> ]
```

指定删除的元组需要满足的条件

缺省，表示要删除表中的所有元组

可以带有子查询块

三、删除数据：DELETE

例7：删除2号课程的所有选课记录。

```
DELETE
FROM   SC
WHERE  Cno = '2' ;
```

三、删除数据：DELETE

例8：删除计算机科学系（CS）全体学生的选修记录。

```
DELETE
FROM    SC
WHERE  Sno  IN
        ( SELECT   Sno
          FROM   Student
          WHERE   Sdept = 'CS');
```

更新数据的完整性检查

- 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
  - ✓ CHECK约束
  - ✓ ……

谢　　谢

THANK YOU FOR YOUR WATCHING

