



# 数据库原理与应用

西南交通大学电气工程学院

西南交通大学

# 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

## 3.5 数据更新

---

### 3.5.1 插入数据

#### 1. 插入元组

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

① 属性列列表中包括表中所有列。（可以不列出列名）

例1 将一个完整的学生元组插入Student表中。

```
INSERT INTO Student VALUES ('95020', '陈冬', '男', 18, 'IS');
```

② 属性列列表中不包括表中所有列。（必须列出列名）

例2 插入一个学生元组，只包含部分数据。

```
INSERT INTO Student (Sno,Sname,Sdept) VALUES ('95009','陈波', 'IS');
```

## 3.5.1 插入数据

---

### 2.插入子查询结果

**INSERT INTO <表名> [(<属性列1>[,<属性列2>...)] 子查询**

功能：一次性**批量插入**子查询的结果到指定表中。

例3 求每个系学生的平均年龄，并把结果存入数据库。

① 建立一个新表，用于存放各系学生的平均年龄。

```
CREATE TABLE Dept_age  
(Sdept CHAR(6),Avg_age SMALLINT)
```

② 插入数据。

```
INSERT INTO Dept_age
```

```
SELECT Sdept,AVG(Sage) FROM Student GROUP BY Sdept
```

## 3.5.2 修改数据

**UPDATE** <表名> **SET** <列名>=<表达式>[,<列名>=<表达式>]...

### 1.修改所有元组的值

例4 将所有学生的考试成绩加上6分。

**UPDATE** SC **SET** Grade=6+Grade

### 2.修改某一个元组的值

例5 数学系学生欧阳庆转到信息系，修改其记录。

**UPDATE** Student **SET** Sdept='IS'

**WHERE** Sdept='MS' AND Sname='欧阳庆'

### 3.带子查询的修改语句

例6 将计算机科学系全体学生的成绩置零。

**UPDATE** SC **SET** Grade=0 **WHERE** Sno IN

(**SELECT** Sno **FROM** Student **WHERE** Sdept='CS')

### 3.5.3 删除数据

---

**DELETE FROM** <表名> [WHERE <条件>];

#### 1. 删除某一个元组的值

例7 删除学号为201215128的学生记录。

**DELETE FROM** Student WHERE Sno=' 201215128 '

#### 2. 删除多个元组的值

例8 删除所有学生的选课记录。

**DELETE FROM** SC SC将成为空表

#### 3. 带子查询的删除语句

例9 删除计算机科学系所有学生的选课记录。

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

## 3.6 视图

□ **虚表**，数据库只存储视图的定义，不存储视图对应的数据

### 一、定义视图

```
CREATE VIEW <视图名>[(<列名>[,<列名>]...)]  
AS <子查询> [ WITH CHECK OPTION];
```

□ **组成视图的属性列名：全部省略或全部指定**

■ **全部省略：**

- 由子查询中SELECT 目标列中的诸字段组成

■ **明确指定视图的所有列名：**

- 某个目标列是聚集函数或列表表达式
- 多表连接时选出了几个同名列作为视图的字段
- 需要在视图中为某个列启用新的更合适的名字

### 3.6.1 定义视图

---

**例1 建立学生视图，用于查看学生的学号，姓名和所在的系。**

**CREATE VIEW V\_Student**

**AS SELECT Sno,Sname,Sdept FROM Student**

**例2 建立课程设置视图，用于表示课程名称和课号。**

**CREATE VIEW V\_Course(课程名称,课号)**

**AS SELECT Cname,Cno FROM Course**

**□ 从单个基表导出，只是去掉了基表的某些行和某些列，保留了码，这类视图称为行列子集视图。**

**■ 上述两个视图都是行列子集视图。**




### 3.6.1 定义视图

---

□ 若某目标列是集函数或列表达式，则必须明确指定视图的所有列名。这类视图称为**分组视图**或**带表达式的视图**。

例3 定义一个反映学生出生年份的视图。（带表达式的视图）

```
CREATE VIEW BT_S(Sno,Sname,Sbirth) 
AS SELECT Sno,Sname, year(getdate())-Sage FROM Student
```

例4 建立系平均年龄视图，显示**各系的平均年龄**。（分组视图）

```
CREATE VIEW V_age(系,平均年龄) AS
SELECT Sdept, AVG(Sage) FROM Student GROUP BY Sdept
```

## 3.6.1 定义视图

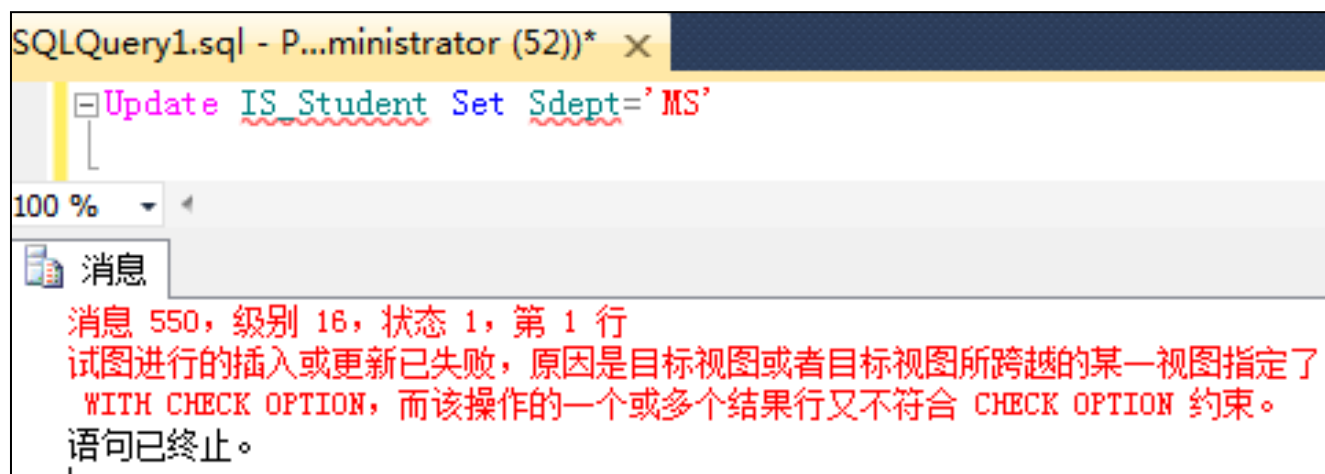
❑ **WITH CHECK OPTION** 透过视图进行增删改操作时，不得破坏视图定义中的谓词条件（即子查询中的条件表达式）

例5 建立信息系学生的视图，并要求透过该视图进行的更新操作只涉及信息系学生。

```
CREATE VIEW IS_Student AS SELECT Sno, Sname, Sage, Sdept  
FROM Student WHERE Sdept= 'IS' WITH CHECK OPTION
```

能否通过该视图  
修改学生的系？

Update IS\_Student  
Set **Sdept='MS'**



### 3.6.1 定义视图

---

□ 视图可以建立在**多个基表**或已经建立的**视图**上。

例6 建立**信息系**选修了**1号课程**的学生视图。

```
CREATE VIEW IS_S1(Sno,Sname,Grade) AS  
SELECT Student.Sno,Sname,Grade FROM Student,SC  
WHERE Sdept='IS' AND Student.Sno= SC.Sno  
AND SC.Cno='1'
```

例7 建立**信息系**选修**1号课程**且**成绩在90分以上**的学生视图。

```
CREATE VIEW IS_S2 AS  
SELECT Sno,Sname,Grade FROM IS_S1 WHERE Grade>=90
```

## 二、删除视图

**DROP VIEW <视图名> [CASCADE]**

- 如果视图上还导出了其他视图，则使用CASCADE级联删除该视图和由它导出的所有视图。
- 删除基表时，由该基表导出的所有视图没有被删除，但均已无法使用。故这些视图的定义都必须显式删除

例8 删除视图IS\_S1。

**DROP VIEW IS\_S1 CASCADE;**

### 3.6.2 查询视图

---

□ 采用**视图消解法**（转换为对基表的查询）实现查询

例9 在信息系学生的视图中找出年龄小于20岁的学生

```
SELECT * FROM IS_Student WHERE Sage<20
```

视图消解转换后的查询语句:

```
SELECT * FROM Student WHERE Sdept= 'IS' AND Sage<20
```

例10 查询信息系选修了1号课程的学生。

```
SELECT Sno, Sname FROM IS_Student, SC  
WHERE IS_Student.Sno =SC.Sno AND SC.Cno= '1'
```

□ 多数DBMS对**行列子集视图的查询**能**正确转换**，其他视图的查询不一定能正确转换，此类查询最好直接对基表进行

### 3.6.3 更新视图

□ 对视图执行的所有操作，最终都是在对**基表**进行操作。

#### 1.插入视图（基表中不允许空值的列都出现在视图中）

例12 向信息系学生视图IS\_Student中插入一个信息系学生记录，其学号为95029，姓名为赵新，年龄为20岁。

```
INSERT INTO IS_Student VALUES('95029','赵新',20,'IS')
```

Student表

Sno	Sname	Ssex	Sage	Sdept
95029	赵新	NULL	20	IS

#### 2.删除视图

例13 删除信息系视图IS\_Student中学号为20131211的记录。

```
DELETE FROM IS_Student WHERE Sno= ' 20131211'
```

该学生是已经退学还是从信息系转到其他系 

### 3.6.3 更新视图

#### 3.修改视图

例14 将IS\_Student中学号121的学生姓名改为“刘辰”。

```
UPDATE IS_Student SET Sname= '刘辰' WHERE Sno= '121'
```

□ 一些视图是**不可更新**的，因为无法转换为基本表的更新。

```
CREATE VIEW S_G (Sno, Gavg) AS
```

```
SELECT Sno,AVG(Grade) FROM SC GROUP BY Sno;
```

```
Update S_G Set Gavg=90 Where Sno= '20121211'
```



- 允许对**行列子集视图**进行更新
- 对其他类型视图的更新不同系统有不同限制

### 3.6.4 视图的作用

---

1. 简化用户的操作

2. 使用户从多种角度看待同一数据

3. 对数据库提供了一定程度的安全性

① 通过视图把保密数据对无权存取这些数据的用户隐藏起来。

② 对不同的视图，赋予不同的权限。对视图权限的设置与对基表权限的设置互相独立。

4. 对重构数据库提供了一定程度的逻辑独立性



### 3.6.4 视图的作用

---

例：数据库逻辑结构发生改变

学生关系Student(Sno, Sname, Ssex, Sage, Sdept)

“垂直”地分成两个基本表：

SX(Sno, Sname, Sage)      SY(Sno, Ssex, Sdept)

通过建立一个视图Student：

```
CREATE VIEW Student(Sno,Sname,Ssex,Sage,Sdept) AS
```

```
SELECT SX.Sno,SX.Sname,SY.Ssex,SX.Sage,SY.Sdept
```

```
FROM SX,SY WHERE SX.Sno=SY.Sno;
```

使外模式保持不变，从而对原Student表的查询程序不必修改

# 作业

---

## 《数据库系统概论》 P130

4题 (第5小题不做)

5题

6题

7题

8题

9题