

# 数据库技术

主讲教师： 沈蒙波

 : 13832396662

# SQL数据操作语句和视图

## 本讲要点

- SQL的数据插入
- SQL的数据更新
- SQL的数据删除
- 视图的概念
- 定义视图
- 修改和删除视图
- 本讲小结和作业

# 数据操作示例数据库

**学生（xs）数据库：**

- **学生表：student (sno, sname, ssex, sage, sdept)**
- **课程表：course (cno, cname, credit, semester)**
- **学生选课表：sc (sno, cno, grade)**

# 数据操作功能

## 插入

```
INSERT [INTO] TableName [(fname1 [, fname2, ...])]
VALUES (eExpression1 [, eExpression2, ...])
```

## 更新

```
UPDATE <TableName>
SET Column_Name1 = eExpression1
[, Column_Name2 = eExpression2 ...]
[ WHERE FilterCondition1 [AND | OR FilterCondition2 ...] ]
```

## 删除

```
DELETE [ FROM ] <TableName>
[WHERE FilterCondition1 [AND | OR FilterCondition2 ...]]
```

# INSERT命令

SQL插入命令的格式:

**INSERT [into] 表名 VALUES(表中各字段值)**

**INSERT [into] 表名(字段名列表) VALUES(字段名列表中各字段值)**

使用插入命令的注意事项:

- 如果表名后没有指定列名, 则插入记录的各字段值必须按表中列的顺序——对应。
- 如果指定了列名, 则values后的字段值要与指定的列名顺序——对应。



# INSERT命令示例

(1) 利用SQL的INSERT命令在学生表中插入一个新的元组（1531032，陈晨，女，20，计算机系）。

```
INSERT INTO student VALUES ( '1531032','陈晨','女',20,'计算机系')
```

(2) 利用SQL的INSERT命令在course表中插入一个新的元组，课程号为“c232”，课程名称是软件工程。

```
INSERT INTO course(cno,cname) VALUES ('c232', '软件工程')
```

# UPDATE命令

SQL更新命令的格式:

**Update <表名> set 列名1=表达式1 [ ,列名2=表达式2 ] .....  
[ Where <条件> ]**

各参数说明:

- (1)Set子句: 指定要修改的列, 表达式给出修改后的新值。
- (2)Where子句指定只修改表中满足条件的记录。 Where中允许出现子查询。省略Where子句, 将无条件修改表中全部记录。

# Update命令示例

例1：将所有学生的年龄加1。

例2：将学号为“1512101”学生的年龄改为21岁。

例3：将“计算机系”全体学生的成绩加5分。

例4：将选修了“数据库基础”课程的学生成绩修改为： $\text{成绩} \times 85\% + 15$ 。



# Update命令示例

例5：在学生表中增加 “奖学金等级,c (4) ” 字段。用SQL的UPDATE命令填充 “奖学金等级” 字段内容，若选修课数 $\geq 3$ ，选修课成绩平均值 $\geq 85$ ，荣获 “一等” 奖学金，若选修课数 $\geq 3$ ，选修课成绩平均值 $\geq 80$ ，小于85，则荣获 “二等” 奖学金。

# DELETE命令

删除记录的命令格式：

DELETE [ FROM ] <表名> [ where 条件表达式 ]

注意：SQL的Delete命令只是删除表中的数据，并不删除表。

删除表的命令：**DROP TABLE** <表名>

# 删除命令举例

例1：删除学生表中所有女生的记录。

例2：删除所有不及格的学生的选课记录。

例3：删除计算机系不及格的学生的选课记录。

# 第5章 视图

- 什么是视图？
  - **视图**是一个虚拟表，**视图**中的数据是通过对**基本表**进行数据查询产生出来的。
  - 数据库中只存放**视图**的定义，不存放**视图**包含的数据。
  - **视图**是数据库中的一个对象，有自己的**视图**名。在数据库的三级模式中外模式对应到数据库中的概念就是**视图**。

# 定义视图

- 使用SQL的Create View语句定义视图。

语句格式：

```
CREATE VIEW <ViewName> [(视图列名表)] AS  
SQLSELECTStatement
```

说明：

*省略了视图列名表，则视图的列名与查询语句的列名相同。*

# 视图举例

例1：建立信息系学生的视图。

例2：建立信息系选了“c001”号课程的学生视图，列出学生的学号、姓名、年龄和成绩。

例3：建立信息系选修了“c001”号课程且成绩90分以上的学生视图。

例4：建立一个学生学号、姓名和出生年份的视图。



# 定义多表连接的视图

**例5：定义一个名称为STU1的视图，该视图的SELECT语句完成查询：**  
**全体学生的学号、姓名、性别、年龄、所在院系、选修课程名称、**  
**选课成绩。**

# 删除视图

- 删除视图的命令格式：

Drop View <视图名>

例：删除stu1视图

Drop View stu1

# 修改视图

- SQL修改视图的Alter View语句。

语句格式：

Alter View 视图名 [(视图列名表)] as 查询语句

说明：

Alter View语句与Create View语句除了关键字不同，其格式和使用方法完全相同。

# 视图的作用

- 简化数据查询语句
- 使用户能从多角度看待同一数据
- 提高了数据的安全性
- 提供了数据库中数据的逻辑独立性

# 使用视图

- 创建了视图后，可以将视图如基本表一样使用。
  - 可以在视图上又创建视图；
  - 可以根据视图建立查询；

例4：根据创建的stu1视图，查询土木学院男、女生人数。

例5：根据stu1视图，创建没选修课的学生基本情况视图。

例6：删除没有选修课的学生视图。

# 小 结

## ✓ SQL数据查询

- 嵌套查询

## ✓ 视图

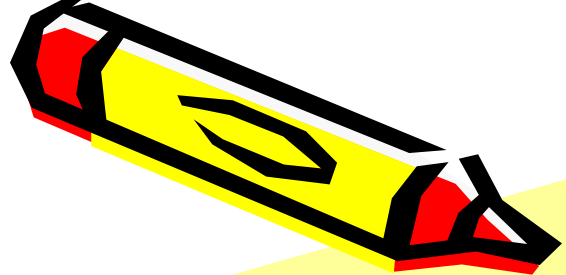
- 视图概念

- 创建视图

- 删除视图

- 视图的作用





# 作业

书面完成以下作业

📖 第3版P65,第4版P86: 19~26;

📖 第3版P72,第4版P93: 3,4,5,6;

📖 认真复习第4章

📖 复习第5章视图



谢谢各位  
下次见!

