



# 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

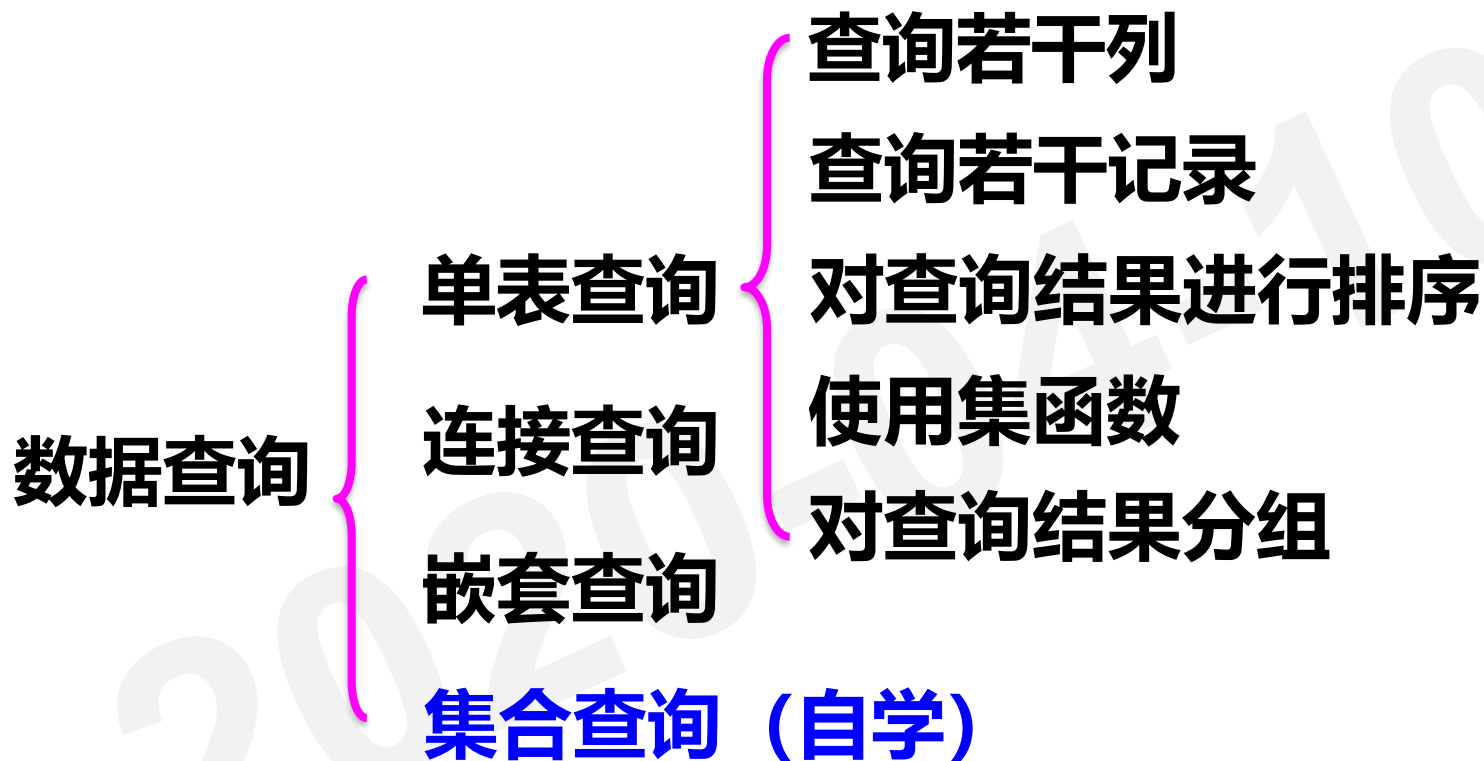
## 3.4 数据查询

## 3.5 数据更新

## 3.6 视图

## 3.4 数据查询

---



### 3.4.1 单表查询

## 5.对查询结果进行分组 ( GROUP BY )

**分组方法：**按指定的一系列或多列值分组，值相等的为一组

- 未对查询结果**分组**，集函数作用于整个查询结果，即**整个查询结果只有一个函数值**。
- 对查询结果**分组后**，集函数分别作用于每一个组，即**每一组都有一个函数值**。

**例21 统计每门课程的选课人数。**

```
SELECT Cno, COUNT(*) FROM SC GROUP BY Cno
```

### 3.4.1 单表查询

- 使用分组子句，**SELECT**后的列名列表中只能出现**分组属性**和**集函数**
- 如果**分组后**还要求**按一定的条件对这些组进行筛选**，最终只输出满足指定条件的组，则可以使用**HAVING**短语指定筛选条件。

例22 统计选课人数**小于20人**的课程号及选课人数。

```
SELECT Cno, COUNT(*) FROM SC GROUP BY Cno  
HAVING COUNT(*)<20
```

### 3.4.2 连接查询

□ 若一个查询同时涉及两个以上的表，则称为**连接查询**

#### 1. 等值连接

$[\langle \text{表名1} \rangle . \langle \text{列名1} \rangle = \langle \text{表名2} \rangle . \langle \text{列名2} \rangle]$

例1 查询每个学生及其选课情况。

```
SELECT Student.*, Cno, Grade FROM Student, SC
```

```
WHERE Student.Sno=SC.Sno;
```

公共属性相等

#### 2. 自身连接

- 连接操作不仅可以在两个表之间进行，也可以是一个表与其自身进行连接，这种连接称为表的自身连接。

## 3.4.2 连接查询

TOP-E15VACJN...a - dbo.Student					SQLC		
Sno	Sna...	Ssex	Sage	Sdept	TOP-E15VACJN.stuData - dbo.SC		
200215121	李勇	男	20	CS	Sno	Cno	Grade
200215122	刘晨	女	19	CS	200215121	1	80
200215123	王敏	女	20	MA	200215121	2	90
200215124	张立	男	19	IS	200215122	1	78
					200215122	3	90
					200215123	2	55

	Sno	Sname	Ssex	Sage	Sdept	Cno	Grade
1	200215121	李勇	男	20	CS	1	80
2	200215121	李勇	男	20	CS	2	90
3	200215122	刘晨	女	19	CS	1	78
4	200215122	刘晨	女	19	CS	3	90
5	200215123	王敏	女	20	MA	2	55

### 3.4.2 连接查询

例2 查询每一门课的间接先行课（即先行课的先行课）。

Course1				Course2			
Cno	Cname	Cpno	Ccredit	Cno	Cname	Cpno	Ccredit
1	数据库	5	4	1	数据库	5	4
2	数学	2	2	2	数学		2
3	信息系统	1	4	3	信息系统	1	4
4	操作系统	6	3	4	操作系统	6	3
5	数据结构	7	4	5	数据结构	7	4

```
SELECT Course1.Cno, Course2.Cpno
FROM Course Course1, Course Course2
WHERE Course1.Cpno=Course2.Cno
```

自身连接的表及属性  
需要别名以示区别



## 3.4.2 连接查询

---

### 3. 外连接

- **内连接**：普通连接，只输出满足连接条件的元组。
- **外连接**：将指定主体表中不满足连接条件的元组一并输出。
  - **左外连接**：列出**左边**关系中**所有**的元组
  - **右外连接**：列出**右边**关系中**所有**的元组

例3 查询学生情况（包括所选课程的成绩）。（与例1对比）

```
SELECT Student.*,Cno,Grade FROM Student  
LEFT OUTER JOIN SC ON Student.Sno=SC.Sno
```

## 3.4.2 连接查询

### 外连接结果

	Sno	Sname	Ssex	Sage	Sdept	Cno	Grade
1	200215121	李勇	男	20	CS	1	80
2	200215121	李勇	男	20	CS	2	90
3	200215122	刘晨	女	19	CS	1	78
4	200215122	刘晨	女	19	CS	3	90
5	200215123	王敏	女	20	MA	2	55
6	200215124	张立	男	19	IS	NULL	NULL

## 4.复合条件连接

例4 查询2号课程成绩在90分以上的学生学号、课程名称和成绩。

```
SELECT Sno,Cname,Grade FROM Course,SC
```

```
WHERE SC.Cno=2 AND Grade>=90 AND Course.Cno=SC.Cno
```

### 3.4.3 嵌套查询

□ 将一个查询块 (SELECT-FROM-WHERE) 嵌套在另一个查询块的WHERE子句或HAVING短语的条件中。

```
SELECT Sname FROM Student WHERE Sno IN
```

父查询  
(外层查询)

```
(SELECT Sno FROM SC WHERE Cno= ' 2 ')
```

子查询(内层查询)

□ 不相关子查询：子查询的查询条件不依赖于父查询

- 由里向外逐层处理。即每个子查询在上一级查询处理之前求解，子查询的结果用于建立其父查询的查找条件。

### 3.4.3 嵌套查询

---

**□相关子查询：**子查询的查询条件**依赖**于父查询

- 先取外层查询中表的第一个元组，根据它与内层查询相关的属性值处理内层查询，若WHERE子句返回真，则取此元组放入结果表；
- 然后再取外层表的下一个元组；
- 重复这一过程，直至外层表全部检查完为止。

#### 1.带有IN谓词的子查询（不相关子查询）

**判断某个属性列是否在子查询的结果中。**

### 3.4.3 嵌套查询

---

**例1 查询与“刘晨”在同一个系学习的学生。**

**① 确定“刘晨”所在系名。**

```
SELECT Sdept FROM Student WHERE Sname= '刘晨'
```

**查询得到系名为IS。**

**② 查找所有在IS系学习的学生。**

```
SELECT * FROM Student WHERE Sdept= 'IS'
```

**构造嵌套查询：**

```
SELECT * FROM Student WHERE Sdept IN
```

```
(SELECT Sdept FROM Student WHERE Sname= '刘晨')
```

### 3.4.3 嵌套查询

{ Student(Sno, Sname, Sex, Sage, Sdept)  
Course(Cno, Cname, Cpname, Credit)  
SC(Sno, Cno, Grade)

例2 查询选修了DB\_Design课程的学号和姓名。

嵌套表达：

```
SELECT Sno,Sname FROM Student WHERE Sno IN  
(SELECT Sno FROM SC WHERE Cno IN  
(SELECT Cno FROM Course WHERE Cname='DB_Design'))
```

上例也可以用连接查询来实现

```
SELECT Student.Sno,Sname FROM Student,SC,Course  
WHERE Cname='DB_Design'  
AND Student.Sno = SC.Sno AND SC.Cno = Course.Cno
```

### 3.4.3 嵌套查询

#### 2.带有比较运算符的子查询（子查询返回单值时）

例3：[例1]可以用 = 代替IN：

Select Sno , Sname, Sdept From Student

Where Sdept =(Select Sdept From Student Where Sname='刘晨')

#### 3. 带有ANY或ALL谓词的子查询（子查询返回多值时）

谓词	注释	谓词	
>ANY	大于子查询结果中的某个值	<=ANY	小于等于子查询结果中的某个值
>ALL	大于子查询结果中的所有值	<=ALL	小于等于子查询结果中的所有值
<ANY	小于子查询结果中的某个值	=ANY	等于子查询结果中的某个值
<ALL	小于子查询结果中的所有值	=ALL	等于子查询结果中的所有值
>=ANY	大于等于子查询结果中的某个值	<>ANY	不等于子查询结果中的某个值
>=ALL	大于等于子查询结果中的所有值	<>ALL	不等于子查询结果中的任何一个值

### 3.4.3 嵌套查询

例4 查询其他系中比信息系某一学生年龄小的学生信息。

Select \* From Student Where Sage < Any

(Select Sage From Student Where Sdept='IS') And Sdept <> 'IS'

□ ANY和ALL谓词有时可以用集函数实现

■ 用集函数实现[例4]：

Select Sname,Sage From Student Where Sage <

(Select Max(Sage) From Student Where Sdept= 'IS') And Sdept <> 'IS'

#### ANY与ALL与集函数的对应关系

	=	<>或!=	<	<=	>	>=
ANY	IN	--	<MAX	<=MAX	>MIN	>= MIN
ALL	--	NOT IN	<MIN	<= MIN	>MAX	>= MAX



## 4. 带有EXISTS谓词的子查询

□ 带有EXISTS谓词的子查询不返回任何实际数据，它只产生逻辑真值 “true”或逻辑假值 “false”。

- 若内层查询结果非空，则返回真值
- 若内层查询结果为空，则返回假值

EXISTS谓词的子查询，其目标列表达式通常用\*，因为带EXISTS的子查询只返回真值或假值，给出列名无实际意义

### 3.4.3 嵌套查询

例5 查询所有选修了1号课程的学生姓名。

**方法一：使用IN**

```
SELECT Sname FROM Student WHERE Sno IN  
(SELECT Sno FROM SC WHERE Cno='1')
```

**方法二：使用EXISTS**

- ① 在Student中依次取每个元组的Sno值，用此值去检查SC关系。
- ② 若SC中存在这样的元组，其Sno值等于此Student.Sno值，并且其Cno= '1'，则取此Student.Sname送入结果关系。

```
SELECT Sname FROM Student WHERE EXISTS
```

**相关子查询**

```
(SELECT * FROM SC WHERE Cno='1' AND Sno=Student.Sno)
```

### 3.4.3 嵌套查询

#### 方法三：使用连接运算

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

一些带(NOT) EXISTS谓词的子查询不能被其他形式的子查询等价替换

所有带IN谓词、比较运算符、ANY(ALL)谓词的子查询都能用带有EXISTS谓词的子查询替换。

EXISTS

带IN谓词、比较运算符、ANY和ALL谓词

### 3.4.3 嵌套查询

---

Student(Sno, Sname, Sex, Sage, Sdept)

SC(Sno, Cno, Grade)

例6 查询未选修1号课程的学生姓名。

**方法一：用NOT IN谓词：**

```
SELECT Sname FROM Student WHERE Sno  
NOT IN (SELECT Sno FROM SC WHERE Cno='1')
```

**方法二：用NOT EXISTS谓词：**

```
SELECT Sname FROM Student WHERE NOT EXISTS  
(SELECT * FROM SC WHERE Student.Sno=SC.Sno AND Cno='1');
```

**此例用连接运算难于实现**