

专题4

数据库查询

为什么要查询数据？

如何实现查询？

专题4

数据库查询

4.1 基本查询

4.2 条件查询

4.3 查询结果处理

4.4 连接查询

4.5 嵌套查询

4.6 综合查询

数据查询语句SELECT的基本框架：

SELECT - **FROM** - **WHERE**

显示哪些列

从哪些表

根据什么条件

4.1

基本查询

- ➔ 选择列
- ➔ 定义别名
- ➔ 消除结果集的重复行
- ➔ 限制结果集行数
- ➔ 计算列值

01 选择列

基本查询语句的格式：

SELECT 属性列表

FROM 数据表名

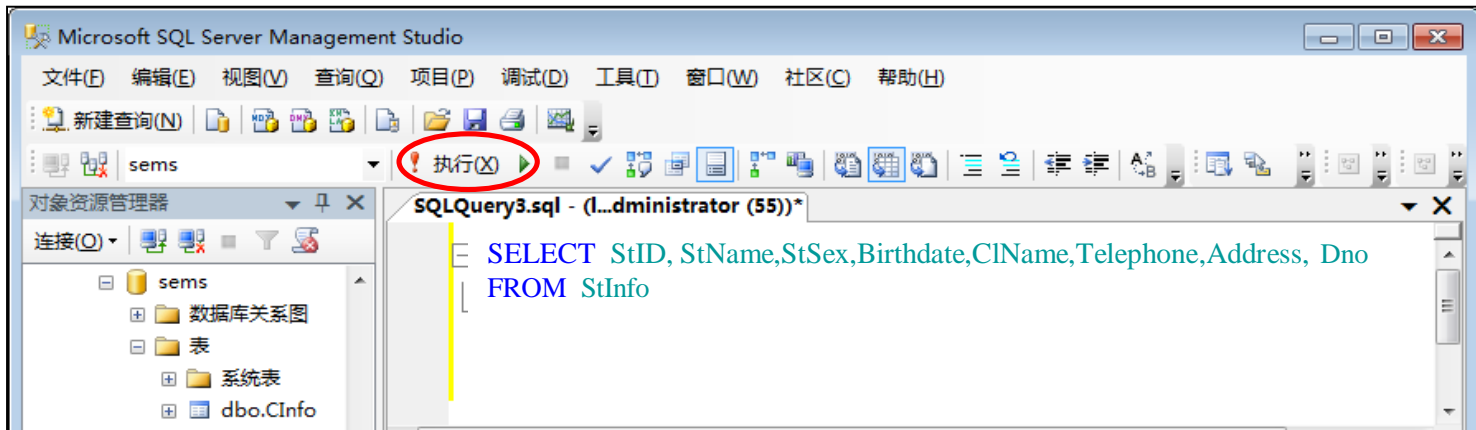
01 选择列

例1 查询StInfo表中所有字段信息。

```
SELECT StID,StName,StSex,Birthdate,CName,Telephone,Address,Dno  
FROM StInfo
```

或者写成:

```
SELECT * FROM StInfo
```



02 定义别名

例2 查询StInfo表中StID、StName、CName字段，分别以学号、姓名和班级作为查询结果的列名。

```
SELECT StID as 学号, StName AS 姓名 , CName 班级 FROM StInfo
```

	学号	姓名	班级
1	0603170108	徐文文	材料科学1701
2	0603170109	黄正刚	材料科学1701
3	0603170110	张红飞	材料科学1701
4	0603170211	曾莉娟	材料科学1702
5	0603170212	张红飞	材料科学1702
6	2001160115	邓红艳	法学1601
7	2001160206	金萍	法学1602

03 消除结果集中的重复行

例3 查询StlInfo表中CName字段信息。

```
SELECT DISTINCT CName FROM StlInfo
```

	CName
1	材料科学1701
2	材料科学1701
3	材料科学1701
4	材料科学1702
5	材料科学1702
6	法学1601
7	法学1602
8	法学1603
9	法学1603

	CName
1	材料科学1701
2	材料科学1702
3	法学1601
4	法学1602
5	法学1603

04 限制结果集的行数

使用TOP n [PERCENT] 限制结果集的行数

- 返回例2查询结果集的前5行

```
SELECT TOP 5 StID 学号, StName 姓名 , CName 班级  
FROM StInfo
```

- 返回例2查询结果前10%行

```
SELECT TOP 10 PERCENT StID as 学号, StName AS 姓名 , CName 班级  
FROM StInfo
```

	学号	姓名	班级
1	0603170108	徐文文	材料科学1701
2	0603170109	黄正刚	材料科学1701
3	0603170110	张红飞	材料科学1701
4	0603170211	曾莉娟	材料科学1702
5	0603170212	张红飞	材料科学1702

	学号	姓名	班级
1	0603170108	徐文文	材料科学1701
2	0603170109	黄正刚	材料科学1701
3	0603170110	张红飞	材料科学1701

05 计算列值

(1) 使用表达式计算列值

例4 按120分计算成绩并显示SCInfo表中前5行学生的成绩情况。

SELECT TOP 5 StID 学号, CNo 课程编号, Score 成绩100, 成绩120=Score*1.2 FROM SCInfo

	学号	课程编号	成绩100	成绩120
1	0603170108	9710021	66	79.2
2	0603170108	9710041	67	80.4
3	0603170109	9710041	78	93.6
4	0603170110	9710041	52	62.4
5	0603170211	9710041	99	118.8

使用表达式
计算列值

05 计算列值

(2) 使用聚合函数计算列值

聚合函数是对一组值执行计算并返回单一值的函数。

	学号	课程编号	成绩
1	0603170108	9710021	56
2	0603170108	9710041	67
3	0603170109	9710041	78
4	0603170110	9710041	52
5	0603170211	9710041	99
6	0603170211	1805012	85

计算平均成绩



平均成绩
82

05 计算列值

(2) 使用聚合函数计算列值

聚合函数是对一组值执行计算并返回单一值的函数

常用聚合函数：

函 数	功 能
AVG(<字段表达式>)	求一列数据的平均值
SUM(<字段表达式>)	求一列数据的和
MIN(<字段表达式>)	求列中的最小值
MAX(<字段表达式>)	求列中的最大值
COUNT(* <字段名>)	统计查询的行数

05 计算列值

(2) 使用聚合函数计算列值

聚合函数是对一组值执行计算并返回单一值的函数

例5 查询所有学生所有课程的平均成绩。

```
SELECT AVG(Score) FROM SCInfo
```

	平均成绩
1	82

05 计算列值

(2) 使用聚合函数计算列值

聚合函数是对一组值执行计算并返回单一值的函数

例6 查询StInfo表的学生总数和所有学生的平均年龄。

- 分析：

年龄=今年的年份-出生年份

GETDATE()函数： 获取当前系统的日期时间

YEAR(日期)函数： 计算日期数据的年份

计算平均年龄： **AVG(YEAR(GETDATE())-YEAR(BirthDate))**

- 查询语句如下：

```
SELECT COUNT(*) AS 总人数 ,  
       AVG(YEAR(GETDATE())-YEAR(BirthDate)) AS 平均年龄  
FROM StInfo
```

	总人数	平均年龄
1	24	20

06 小结

- ◆ 查询语句的基本框架SELECT – FROM - WHERE
- ◆ 基本查询语句针对一个数据表实现查询。
关键字AS定义别名,
DISTINCT消除结果集的重复行;
TOP限制结果集的行数;
列值可以通过表达式计算, 也可能通过聚合函数计算。

专题4

数据库查询

4.1 基本查询

4.2 条件查询

4.3 查询结果处理

4.4 连接查询

4.5 嵌套查询

4.6 综合查询

专题4

数据库查询

SELECT语句的基本格式:

SELECT 属性列表

FROM 数据表名

WHERE 查询条件

4.2

条件查询

- ➡ 比较运算
- ➡ 逻辑运算
- ➡ 字符匹配运算
- ➡ 范围比较运算
- ➡ 空值比较运算

01 比较运算

用于WHERE子句的比较运算符有：

- =(等于)、!=或<>(不等于)、>(大于)、>=(大于等于)、<(小于)、<=(小于等于) 等多个运算符。由比较运算符构成的表达式称为 **关系表达式**。
- 参与比较运算的数据类型可以是字符型、数值型、日期型。
- 运算结果为真或假。

01 比较运算

例1 查询StInfo表中2000年及以后出生的学生情况，要求列出学号、姓名、出生年份和所在班级。

```
SELECT StID, StName, YEAR(Birthdate) AS YearofBirth, CName  
FROM StInfo  
WHERE YEAR(Birthdate) >= 2000
```

	StID	StName	YearofBirth	CName
1	2701170101	孙洋	2000	计算机1701
2	2701180101	李玉霞	2002	计算机1801
3	2701180102	张瑞嘉	2000	计算机1801

02 逻辑运算

逻辑运算符：AND（与）、OR（或）、NOT（非）

运算符	含 义
AND	如果两个表达式都为真，运算结果为真，否则为假
OR	只要有一个表达式为真，运算结果就为真，否则为假
NOT	真值取反为假值，假值取反为真值

02 逻辑运算

例2 查询SCInfo表中选课成绩分数大于等于80分，且小于90分的学生信息。

使用以下语句：

```
SELECT * FROM SCInfo  
WHERE Score>=80 AND Score<90
```

	StID	CNo	Score	SCDate
1	0603170211	1805012	85	2017-09-01
2	2001160115	9710011	88	2016-09-01
3	2001160206	9710011	89	2016-09-01

03 字符匹配运算LIKE

语法格式：

属性名或表达式 [NOT] LIKE '格式字符串'

运算符	描 述	示 例
%	包含零个或多个字符的任意字符串	address LIKE '%公司%' address字段的字符串任意位置包含“公司”
_	下划线，对应任何单个字符	employee_name LIKE '_卫平' 将查找以“卫平”结尾的共3个字符的名字
[]	指定范围(如[a-f])或集合(如[abcdef])中的任何单个字符	employee_name LIKE '[张李王]卫平' 将查找张卫平、李卫平、王卫平的名字
[^]	不属于指定范围或集合的任何单个字符	employee_name LIKE '[^张李]卫平' 将查找不姓张、李的名为卫平的名字

03 字符匹配运算

例3 查询StInfo表中姓张的学生信息。

使用以下语句：

```
SELECT * FROM StInfo  
WHERE StName LIKE '张%'
```

	StID	StName	StSex	Birthdate
1	0603170108	徐文文	男	1998-07-01
2	0603170109	黄正刚	男	1999-12-26
3	0603170110	张红飞	男	1999-03-29
4	0603170211	曾莉娟	女	1998-08-15
5	0603170212	张红飞	男	1988-03-29
6	2001160115	邓红艳	女	1997-07-03
7	2001160206	金萍	女	1998-11-06
8	2001160307	吴中华	男	1997-04-10
9	2001160308	王铭	男	1998-09-09
10	2001170103	郑远月	男	1999-06-18
11	2001170104	张力明	男	1999-08-29
12	2001170205	张好然	女	1999-04-19
13	2001170206	李娜	女	1999-10-21

StInfo表的部分信息

03 字符匹配运算

例4 在StInfo表中查询学号倒数第3个数字为1，倒数第1个数字在1~4之间的学生的学号、姓名、班级信息。

```
SELECT StID, StName, CName  
FROM StInfo  
WHERE StID LIKE '%1_[1234]'
```

	StID	StName	CName
1	2001170103	郑远月	法学1701
2	2001170104	张力明	法学1701
3	2201150101	张炜	临床(五年)1501
4	2201150102	刘永州	临床(五年)1501

04 范围比较运算BETWEEN...AND

语法格式:

表达式 [NOT] **BETWEEN** 开始范围 **AND** 结束范围

例5 在StInfo表中查询2000年出生的学生信息。

```
SELECT * FROM StInfo  
WHERE BirthDate BETWEEN '2000-1-1' AND '2000-12-31'
```

	StID	StName	StSex	Birthdate	CName	Telephone	Address	DNo
1	2701170101	孙洋	女	2000-05-05	计算机1701	0853_88561...	贵州贵阳	27
2	2701180102	张瑞嘉	女	2000-06-07	计算机1801	0731-65731...	湖南长沙	27

04 范围比较运算BETWEEN...AND

例2中，查询SCInfo表中成绩大于等于80小于90的学生信息。

```
SELECT * FROM S_C_Info  
WHERE Score >=80 AND Score<90
```

可以写成：

```
SELECT * FROM SCInfo  
WHERE Score BETWEEN 80 AND 89
```

05 空值比较运算

空值表示值未知。空值不同于空白或零值。

空值比较运算: [NOT] IS NULL

例6 对StInfo表, 查询所有Telephone为空值的信息。

```
SELECT * FROM StInfo  
WHERE Telephone IS NULL
```

是否可以写成:
Telephone = NULL?

	StID	StName	StSex	Birthdate	CName	Telephone	Address	DNo
1	2201150205	赵彦	男	1997-08-05	临床(五年)1502	NULL	湖北省汉阳市多宝镇	22

06 小结

◆ WHERE 条件表达式

◆ 条件运算

- 比较运算 >、>=、<、<=、!=、=
- 逻辑运算 AND OR NOT
- 字符匹配运算 LIKE
- 范围比较运算 BETWEEN...AND
- 空值比较运算 IS NULL

专题4

数据库查询

4.1 查询语句框架

4.2 条件查询

4.3 查询结果处理

4.4 连接查询

4.5 嵌套查询

4.6 综合查询

4.3

查询结果处理


- ➡ 排序输出
- ➡ 重定向输出
- ➡ 分组与筛选
- ➡ 结果的集合运算

01 排序输出

语法格式

ORDER BY 排序表达式 [ASC|DESC] , 排序表达式i[ASC|DESC] [...]

ORDER BY按排序表达式对结果集排序。



列 名
表达式
正整数

01 排序输出

例1 按班级升序列出StInfo表的学生信息，同一个班级的再按年龄由小到大排序。

```
SELECT *  
FROM StInfo  
ORDER BY CName, BirthDate DESC
```

StID	StName	StSex	Birthdate	CName	Telephone	Address	DNo
0603170109	黄正刚	男	1999-12-26	材料科学1701	0851_86541235	贵州省平坝县...	06
0603170110	张红飞	男	1999-03-29	材料科学1701	74586321	河南省焦作市...	06
0603170108	徐文文	男	1998-07-01	材料科学1701	0731_20223388	湖南省长沙市...	06
0603170211	曾莉娟	女	1998-08-15	材料科学1702	0710_65234237	湖北省天门市...	06
0603170212	张红飞	男	1988-03-29	材料科学1702	0370_54586336	河南省焦作市...	06
2001160115	邓红艳	女	1997-07-03	法学1601	35687957	广西桂林市兴...	20
2001160206	金萍	女	1996-11-06	法学1602	0770_55687958	广西桂平市社...	20
2001160308	王铭	男	1998-09-09	法学1603	0371_64586366	河南省上蔡县...	20

02 重定向输出

语法格式

INTO 新数据表

02 重定向输出

例2 查询SCInfo表选修 课程号为 “9710011” (课程名为 “大学计算机基础”) 课程的学生信息，并将结果存入newstudent表中。

```
SELECT StID 学号, CNo 大学计算机基础, Score 成绩  
INTO newstudent  
FROM SCInfo  
WHERE CNo= '9710011'
```



03 分组与筛选

分组子句语法格式

SELECT [列名列表], 汇总表达式
FROM 表名
GROUP BY 列名列表

03 分组与筛选

例3 分别统计SCInfo表中每门课程的课程编号和平均成绩。

```
SELECT CNo 课程编号, AVG(Score) 平均成绩  
FROM SCInfo  
GROUP BY CNo
```

	课程编号	平均成绩
1	1805012	87
2	29000011	87
3	9710011	79
4	9710021	64
5	9710041	74
6	9720013	87
7	9720044	90

03 分组与筛选

筛选子句语法格式

HAVING 筛选条件

03 分组与筛选

查询例3中平均成绩大于85的课程的课程编号和平均成绩。

```
SELECT CNo 课程编号, AVG(Score) 平均成绩
FROM SCInfo
GROUP BY CNo
HAVING AVG(Score)>85
```

	课程编号	平均成绩
1	1805012	87
2	29000011	87
3	9720013	87
4	9720044	90

想一想：

- 对于分组统计查询，SELECT子句的属性列表有什么特点？
- WHERE 与 HAVING两者的区别。

04 结果的集合运算

- UNION并集运算
- EXCEPT差集运算
- INTERSECT交集运算

04 UNION并集运算

语法格式

[UNION [ALL] <SELECT语句>]

UNION运算将两个或多个SELECT查询的结果合并成一个结果集。

注意： ALL关键字表示包含重复的元组。

04 UNION并集运算

例4 列出CInfo表中课程编号为 “9710011”或
“9720033”的课程名称和学分。

```
SELECT CName,CCredit  
FROM CInfo WHERE CNo='9710011'
```

UNION

```
SELECT CName,CCredit  
FROM CInfo WHERE CNo='9720033'
```

	CName	CCredit
1	大学计算机基础	2

	CName	CCredit
1	数据库应用基础实践	3



	CName	CCredit
1	大学计算机基础	2
2	数据库应用基础实践	3

想一想：能否合并不同表的数据？

```
SELECT StId,StName from StInfo  
UNION  
SELECT CNo,CName from CInfo
```

04 EXCEPT差集运算

语法格式

<SELECT语句1>

EXCEPT

<SELECT语句2>

EXCEPT运算从EXCEPT操作数左边的查询中返回右边的查询没有找到的所有非重复值。

例5 对StInfo表和SCInfo表使用EXCEPT进行运算。

```
SELECT StID FROM StInfo
```

```
EXCEPT
```

```
SELECT StID FROM SCInfo
```

	StID
1	2001170103
2	2001170104
3	2001170205
4	2001170206
5	2201150102
6	2201150204
7	2201150205
8	2701170101
9	2701180102

04 INTERSECT交集运算

语法格式

<SELECT语句1>

INTERSECT

<SELECT语句2>

INTERSECT返回INTERSECT操作数左右两个查询均返回的所有非重复值。

04 INTERSECT交集运算

例6 对StInfo表和SCInfo表使用INTERSECT进行运算

```
SELECT StID FROM StInfo
```

```
INTERSECT
```

```
SELECT StID FROM SCInfo
```

注意：使用 UNION、EXCEPT 或 INTERSECT 运算的基本规则：

- 所有查询中的列数和列的顺序必须相同
- 对应列的数据类型必须相同

	StID
1	0603170108
2	0603170109
3	0603170110
4	0603170211
5	0603170212
6	2001160115
7	2001160206
8	2001160307
9	2001160308
10	2201150101
11	2602170105
12	2602170106
13	2602170107
14	2602170108
15	2701180101

07 小结

- ◆ ORDER BY子句用于对查询结果按排序表达式排序输出
- ◆ INTO 子句用于将查询结果保存到一个新建的数据表中
- ◆ GROUP BY子句用于将结果集分组，HAVING通常和GROUP BY子句组合使用，用于筛选分组
- ◆ UNION、EXCEPT、INTERSECT运算分别用于两个及以上结果集的合并、剔除、取重操作，对应关系运算的并、差、交运算

专题4

数据库查询

4.1 查询语句框架

4.2 条件查询

4.3 查询结果处理

4.4 连接查询

4.5 嵌套查询

4.6 综合查询

4.4 连接查询

- ➔ 连接概述
- ➔ 内连接
- ➔ 外连接
- ➔ 交叉连接

01 连接概述

引入

例1 查询学生的选课情况，要求显示学生学号、所修课程的课程编号、课程名称和成绩。

StID	CNo	CName	Score
0603170108	9710021	VB .NET程序设计	66
0603170108	9710041	C++程序设计基础	67
0603170109	9710041	C++程序设计基础	78
0603170110	9710041	C++程序设计基础	52
0603170211	9710041	C++程序设计基础	99
0603170211	1805012	大学英语	85
2001160115	9710011	大学计算机基础	88



StID	CNo	Score
0603170108	9710021	66
0603170108	9710041	67
0603170109	9710041	78
0603170110	9710041	52
0603170211	9710041	99
0603170211	1805012	85
2001160115	9710011	88

Scinfo.Cno=Cinfo.Cno

CNo	CName	CType
1805012	大学英语	必修
19010122	艺术设计史	选修
20010051	民法学	必修
29000011	体育	必修
9710011	大学计算机基础	必修
9710021	VB .NET程序设计	必修
9710031	数据库技术与应用	必修

01 连接概述

连接字段

连接条件中用于两个关系连接的字段称为连接字段。


连接条件的一般格式如下：

表名1.字段名 比较运算符 表名2.字段名。

StID	CNo	Score
0603170108	9710021	66
0603170108	9710041	67
0603170109	9710041	78
0603170110	9710041	62
0603170211	9710041	79
0603170211	1805012	85
2001160115	9710011	88

CNo	CName	CType	CCredit
1805012	大学英语	必修	6
19010122	艺术设计史	选修	2
20010051	民法学	必修	4
29000011	体育	必修	6
9710011	大学计算机基础	必修	2
9710021	VB .NET程序设计	必修	3
9710031	数据库技术与应用	必修	3

Scinfo.Cno=Cinfo.Cno



01 连接概述

如果连接条件在WHERE子句中给出，将返回连接表中符合连接条件的数据行。

例如，查询例1中学生的选课情况。

```
SELECT SCInfo.StID, CInfo.Cno, CInfo.CName, SCInfo.Score  
FROM SCInfo, CInfo  
WHERE SCInfo.CNo = CInfo.CNo
```

```
SELECT x.StID, x.CNo, y.CName, x.Score  
FROM SCInfo x, CInfo y  
WHERE x.CNo = y.CNo
```

x、y分别为SCInfo、
CInfo表的别名

01 连接概述

在FROM子句中，JOIN关键字指定连接表，ON关键字指出连接条件。
格式如下：

FROM 表1 [连接类型] JOIN 表2 ON 连接条件

01 连接概述

◆ 连接类型

根据连接结果处理方式的不同，分为：

- 内连接 INNER
- 外连接 OUTER
- 交叉连接 CROSS

02 内连接

内连接返回连接表中符合连接条件的数据行。

根据连接条件比较操作不同分为：

等值连接 (=) 、 不等值连接 (>、<、>=、<=、<>)

02 内连接

◆ 等值连接

在连接条件中使用等号运算符(=)比较被连接数据表的相应列的列值，若列值相等，则连接被连接数据表中列值相等的记录，包括重复列。

例如，查询例1中学生的选课情况。

```
SELECT SCInfo.StID, CInfo.CName, SCInfo.Score  
FROM SCInfo JOIN CInfo  
ON SCInfo.CNo=CInfo.CNo
```

StID	CNo	CName	Score
0603170108	9710021	VB .NET程序设计	66
0603170108	9710041	C++程序设计基础	67
0603170109	9710041	C++程序设计基础	78
0603170110	9710041	C++程序设计基础	52
0603170211	9710041	C++程序设计基础	99
0603170211	1805012	大学英语	85
2001160115	9710011	大学计算机基础	88

02 内连接

◆ 等值连接

例2 在sems数据库中，查询材料科学1701班学生的选课情况。要求列出学号、姓名、课程号、课程名称和成绩。

StID	StName	StSex	Birthdate
0603170108	徐文文	男	1998-07-01
0603170109	黄正	男	1998-07-26
0603170110	张红飞	男	1999-03-29
0603170211	曾莉娟	女	1998-08-15
0603170212	张红飞	男	1999-03-29
2001160115	邓红艳	女	1998-07-03
2001160206	金萍	女	1998-11-06

a (StInfo表)

StID	CNo	Score
0603170108	9710021	66
0603170108	9710041	67
0603170109	9710041	62
0603170110	9710041	62
0603170211	9710041	79
0603170211	1805012	85
2001160115	9710011	88

b (SCInfo表)

CNo	CName	CType
1805012	大学英语	必修
19010122	艺术设计史	选修
20010051	民法学	必修
29000011	体育	必修
9710011	大学计算机基础	必修
9710021	VB .NET程序设计	必修
9710031	数据库技术与应用	必修

c (CInfo表)

02 内连接

◆ 等值连接

例2 在sems数据库中，查询材料科学1701班学生的选课情况。要求列出学号、姓名、课程名、课程号和成绩。

```
SELECT a.StID,a.StName,b.CNo, c.CName,c.Score  
FROM StInfo a INNER JOIN SCInfo b ON a.StID=b.StID  
JOIN CInfo c ON b.CNo=c.CNo  
WHERE a.CName = '材料科学1701'
```

	StID	StName	CNo	CName	Score
1	0603170108	徐文文	9710021	VB .NET程序设计	66
2	0603170108	徐文文	9710041	C++程序设计基础	67
3	0603170109	黄正刚	9710041	C++程序设计基础	78
4	0603170110	张红飞	9710041	C++程序设计基础	62

02 内连接

◆ 自然连接

是一种特殊的等值连接，要求连接字段必须是相同的属性，并且在结果中去掉重复的字段。

例3 在sems数据库中，查询课程的选课情况。

SELECT CInfo.* , StID, Score, SCDate

FROM CInfo JOIN SCInfo ON CInfo.CNo=SCInfo.CNo

	CNo	CName	CType	CCredit	Hours	StID	CNo	Score	SCDate
1	9710021	VB .NET程序设计	必修	3	48	0603170108	9710021	66	2018-09-01
2	9710041	C++程序设计基础	必修	3	48	0603170108	9710041	67	2018-03-01
3	9710041	C++程序设计基础	必修	3	48	0603170109	9710041	78	2018-03-01
4	9710041	C++程序设计基础	必修	3	48	0603170110	9710041	52	2018-03-01



	CNo	CName	CType	CCredit	Hours	StID	Score	SCDate
1	9710021	VB .NET程序设计	必修	3	48	0603170108	66	2018-09-01
2	9710041	C++程序设计基础	必修	3	48	0603170108	67	2018-03-01
3	9710041	C++程序设计基础	必修	3	48	0603170109	78	2018-03-01
4	9710041	C++程序设计基础	必修	3	48	0603170110	52	2018-03-01

02 内连接

◆ 自连接 是指一个表自己与自己建立连接，是连接的特例。

例4 在sems数据库中，查找不同课程成绩相同的学生的学号、课程号、成绩。

```
SELECT a.StID, a.CNo, b.CNo, a.Score
FROM SCInfo a JOIN SCInfo b
      ON a.StID=b.StID AND a.score=b.Score
      AND a.CNo!=b.CNo
```

	StID	CNo	CNo	Score
1	2001160307	9720013	29000011	77
2	2001160307	29000011	9720013	77

StID	CNo	Score
2001160307	9710011	76
2001160307	9720013	77
2001160308	9710011	66
2001160308	9720013	88
2602170105	29000011	77
2602170106	29000011	97
2602170107	29000011	92
2602170108	29000011	83
0603170212	1805012	92
2201150101	1805012	86
2201150101	29000011	90
2701180101	9720044	90
2001160307	29000011	77

03 外连接

◆ 外连接(OUTER JOIN)

- 左外连接 (LEFT OUTER JOIN)
- 右外连接 (RIGHT OUTER JOIN)
- 全外连接 (FULL OUTER JOIN)

04 外连接

◆ 左外连接

使用**LEFT OUTER JOIN**关键字进行连接。左外连接保留了第一个表(左表)的所有行，但只包含第二个表(右表)与第一个表匹配的行。第二个表相应的空行的字段置NULL值。

04 外连接

例5 StInfo表左外连接SCInfo表，查询学号、姓名、课程号和成绩。

SELECT a.StID, a.StName, b.CNo, b.Score

FROM StInfo a **LEFT OUTER JOIN** SCInfo b **ON** a.StID = b.StID

左表

	StID	StName	CNo	Score
14	2001160308	王铭	9710011	66
15	2001160308	王铭	9720013	88
16	2001170103	郑远月	NULL	NULL
17	2001170104	张力明	NULL	NULL
18	2001170205	张好然	NULL	NULL
19	2001170206	李娜	NULL	NULL
20	2201150101	张炜	1805012	86
21	2201150101	张炜	29000011	90
22	2201150102	刘永州	NULL	NULL

右表

查询结果部分信息

04 外连接

右外连接

右外连接使用**RIGHT OUTER JOIN**关键字进行连接。右外连接包含**右表**的所有行, 左表相应的空行被放入NULL值。

例6 StInfo表右外连接SCInfo表。

```
SELECT a.StID, a.StName, b.CNo, b.Score  
FROM StInfo a RIGHT OUTER JOIN SCInfo b  
      ON a.StID = b.StID
```

04 外连接

◆ 全外连接

使用**FULL OUTER JOIN**关键字连接，它返回两个表的所有行。不管两个表的行是否满足连接条件，均返回查询结果集。对不满足连接条件的记录，另一个表相对应字段用NULL代替。

例7 StInfo表全外连接SCInfo表。

```
SELECT StInfo.StId, StName, CNo, Score  
FROM StInfo FULL OUTER JOIN SCInfo  
ON StInfo.StID =SCInfo. StID
```


04 交叉连接

- 交叉连接没有WHERE子句，它返回连接表中所有数据行的笛卡尔积。笛卡尔积结果集的大小为第一个表的行数乘以第二个表的行数。交叉连接使用关键字CROSS JOIN进行连接。

例8 将StInfo表和SCInfo表进行交叉连接。

```
SELECT StInfo.StID, StInfo.StName, SCInfo.*  
FROM StInfo CROSS JOIN SCInfo
```

	StID	StName	StID	CNo	Score
1	0603170108	徐文文	0603170108	9710021	66
2	0603170109	黄正刚	0603170108	9710021	66
3	0603170110	张红飞	0603170108	9710021	66
4	0603170211	曾莉娟	0603170108	9710021	66
5	0603170212	张红飞	0603170108	9710021	66
6	2001160115	邓红艳	0603170108	9710021	66
7	2001160206	金萍	0603170108	9710021	66
8	2001160307	李中华	0603170108	9710021	66

查询结果部分信息

05 小结

- 连接条件决定了返回结果表中的数据行。
- JOIN关键字指定连接类型：内连接、外连接、交叉连接。
- 常用的内连接是等值连接，使用=作为条件运算符；自连接是一个表与自身进行的连接，是连接的特例。
- 外连接分为左外连接、右外连接、全外连接
- 交叉连接返回连接表中所有数据行的笛卡尔积。

专题4

数据库查询

4.1 查询语句框架

4.2 条件查询

4.3 查询结果处理

4.4 连接查询

4.5 嵌套查询

4.6 综合查询

4.5 嵌套查询

- ➔ 单值嵌套查询
- ➔ 多值嵌套查询

01 嵌套查询概念

◆ 定义

在一个SELECT 语句的 WHERE 子句或 HAVING 子句中嵌套另一个 SELECT语句的查询称为嵌套查询，又称子查询。

◆ 类型

单值嵌套查询

多值嵌套查询

02 单值嵌套

◆ 子查询的返回结果是一个值的嵌套查询称为**单值**嵌套查询。

例1 对sems数据库，查询选修“数据库技术与应用”课程的学生的学号和成绩。

StID	CNo	Score
0603170110	9710041	62
0603170211	9710041	79
0603170211	1805012	85
2001160115	9710011	88
2001160115	9720013	90
0603170212	9710031	80
2602170106	9710031	89
2602170105	9710031	92

CNo	CName	CType
1805012	大学英语	必修
19010122	艺术设计史	选修
20010051	民法学	必修
29000011	体育	必修
9710011	大学计算机基础	必修
9710021	Visual Basic.Net程序设计基础	必修
9710031	数据库技术与应用	必修
9710041	C++程序设计基础	必修
9720013	大学计算机基础实践	实践
9720033	数据库技术与应用实践	实践



02 单值嵌套

◆ 子查询的返回结果是一个值的嵌套查询称为单值嵌套查询。

例1 对sems数据库，查询选修“数据库技术与应用”课程的学生的学号和成绩。

```
SELECT StId, Score
FROM SCInfo
WHERE CNo= ( SELECT CNo
              FROM CInfo
              WHERE CName='数据库技术与应用' )
```

03 多值嵌套

❖ 子查询的返回结果是多个值的嵌套查询称为多值嵌套查询。

❖ 集合条件运算

- ANY
- ALL
- IN
- EXISTS

03 多值嵌套

➤ 使用ANY运算

子查询结果集中任意一个值满足比较条件则返回TURE，否则返回FALSE。

ANY通常组合比较运算符使用。

格式如：>ANY 表示大于任意一个值。

如：>ANY (80, 72, 92) 表示大于72即为TRUE

03 多值嵌套

➤ 使用ANY运算符

子查询结果集中任意一个值满足比较条件就返回TURE， 否则返回FALSE。

例2 在sems数据库中， 查询选修“9710031”课程的学生们的成绩比选修“9710041” 的学生的最低成绩高的学生的学号和成绩。

```
SELECT StId, Score
FROM SCInfo
WHERE CNo='9710031' AND Score >ANY
(SELECT Score FROM SCInfo
WHERE CNo='9710041')
```

	cno	Score
1	9710041	67
2	9710041	78
3	9710041	62
4	9710041	79
5	9710041	90

	StId	CNo	Score
1	0603170212	9710031	80
2	2602170106	9710031	89
3	2602170105	9710031	92

	StId	Score
1	0603170212	80
2	2602170106	89
3	2602170105	92

03 多值嵌套

➤ 使用ALL运算

子查询结果集中每个值都满足比较条件时返回TURE，否则返回FALSE。

ALL通常组合比较运算符使用。

格式如： >ALL 表示大于所有值

如， >ALL (80, 72, 92) 表示大于 92才为TRUE

03 多值嵌套

➤ 使用ALL运算符

子查询结果集中每个值都满足比较条件时返回TURE，否则返回FALSE。

例3 在sems数据库中，查询选修“9710031”课程的学生们的成绩比选修“9710041”的学生的最高成绩还要高的学生的学号和成绩。

```
SELECT StId, Score
FROM SCInfo
WHERE CNo='9710031' AND Score >ALL
(SELECT Score FROM SCInfo
WHERE CNo='9710041')
```

	cno	Score
1	9710041	67
2	9710041	78
3	9710041	62
4	9710041	79
5	9710041	90

	StId	CNo	Score
1	0603170212	9710031	80
2	2602170106	9710031	89
3	2602170105	9710031	92

	StId	Score
1	2602170105	92

03 多值嵌套

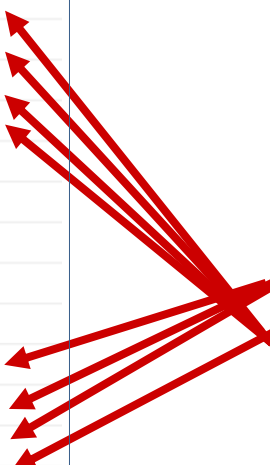
➤ 使用IN运算符

IN是属于的意思，等价于“=ANY”，等于子查询中任何一个值即为TRUE。

例4 对sems数据库，查询选修“数据库技术与应用”或选修“C++程序设计基础”的学生学号和成绩。

StID	CNo	Score
0603170108	9710021	66
0603170108	9710041	67
0603170109	9710041	78
0603170110	9710041	62
0603170211	9710041	79
0603170211	1805012	85
2001160115	9710011	88
2001160115	9720013	90
2001160206	9710011	89
2001160206	9720013	93
0603170212	9710031	80
2602170106	9710031	89
2602170105	9710031	92
2602170105	9710041	90

CNo	CName	CType
1805012	大学英语	必修
19010122	艺术设计史	选修
20010051	民法学	必修
29000011	体育	必修
9710011	大学计算机基础	必修
9710021	Visual Basic.Net程序设...	必修
9710031	数据库技术与应用	必修
9710041	C++程序设计基础	必修



03 多值嵌套

➤ 使用IN运算符

IN是属于的意思，等价于“=ANY”，等于子查询中任何一个值即为TRUE。

例4 对sems数据库，查询选修“数据库技术与应用”或选修“C++程序设计基础”的学生学号和成绩。

```
SELECT StId, Score
FROM SCInfo
WHERE CNo IN
  (SELECT CNo FROM CInfo
   WHERE CName='数据库技术与应用'
    OR CName='C++程序设计基础')
```

	StId	CNo	Score
1	0603170108	9710041	67
2	0603170109	9710041	78
3	0603170110	9710041	62
4	0603170211	9710041	79
5	0603170212	9710031	80
6	2602170106	9710031	89
7	2602170105	9710031	92
8	2602170105	9710041	90

03 多值嵌套

➤ 使用EXISTS运算

如果子查询包含至少一行数据，那么值为TRUE

例5 对sems数据库，列出所有成绩在90分以上的学生的姓名和班级名称。

```
SELECT StName ,CName  
FROM StInfo  
WHERE EXISTS (SELECT * FROM SCInfo  
               WHERE Score >90 AND StInfo.StID = SCInfo.StID )
```

	StName	CName
1	张红飞	材料科学1702
2	金萍	法学1602
3	杨平娟	口腔(七)1701
4	王小维	口腔(七)1701
5	刘小玲	口腔(七)1701

04 小结

◆ 单值嵌套查询

◆ 多值嵌套查询

使用集合条件运算

➤ ANY

➤ ALL

➤ IN

➤ EXISTS

专题4

数据库查询

4.1 查询语句框架

4.2 条件查询

4.3 查询结果处理

4.4 连接查询

4.5 嵌套查询

4.6 综合查询

01 分组统计应用实例

- 在SEMS数据库中，查找出各课程的平均成绩，按课程号分组，只选择修课学生人数超过3人的课程的成绩，并按由小到大顺序显示。

三个方面：

(1) 分组统计。

GROUP BY Cno
AVG(score)

(2) 筛选。

COUNT(*)>3

(3) 排序。

平均成绩
AVG(score)

查询语句：

```
SELECT Cno,AVG(score)平均成绩
FROM SCInfo
GROUP BY Cno
HAVING COUNT(*)>3
ORDER BY 平均成绩
```

	cno	平均成绩
1	1805012	87
2	29000011	86
3	9710011	79
4	9710021	66
5	9710041	74
6	9720013	87
7	9720044	90



	cno	平均成绩
1	29000011	86
2	9710011	79
3	9710041	74
4	9720013	87



	cno	平均成绩
1	9710041	74
2	9710011	79
3	29000011	86
4	9720013	87

02 连接嵌套查询

◆ 查找选修了学号为2001160115的学生所修课程的学生的学号和姓名。

分三步：

(1) 查找2001160115学生所修的课程号。

(2) 使用连接查询获取所有学生的学号、姓名、课程号等信息。

(3) 通过WHERE子句选择以下课程号匹配的学生记录。

9710011, 9720013

查询语句：

```
SELECT DISTINCT a.StID, StName
FROM SCInfo b JOIN StInfo a ON a.StID=b.StID
WHERE b.cno in (
    SELECT CNo FROM SCInfo
    WHERE StID ='2001160115'
)
```

	CNo
1	9710011
2	9720013



	StID	StName
1	2001160115	邓红艳
2	2001160206	金萍
3	2001160307	吴中华
4	2001160308	王铭

03 集合运算应用

查询只有男生选修的课程的课程名称。

分三步：

- (1) 确定男生选修了哪些课。
- (2) 确定女生选修了哪些课。
- (3) 使用EXCEPT运算从男生选修课程中去掉女生选修课程

查询语句：

```
SELECT DISTINCT CName
FROM SCInfo b JOIN stInfo a ON a.StID=b .StID
      JOIN CInfo c ON b.CNo =c .CNo
WHERE StSex ='男'
EXCEPT
SELECT DISTINCT Cname
FROM SCInfo b JOIN stInfo a ON a.StID=b .StID
      JOIN CInfo c ON b.CNo =c .CNo
WHERE StSex ='女'
```

	Cname
1	VB .NET程序设计
2	Web开发技术

=

	Cname
1	C++程序设计基础
2	VB .NET程序设计
3	Web开发技术
4	大学计算机基础
5	大学计算机基础实践
6	大学英语
7	体育

-

	Cname
1	C++程序设计基础
2	大学计算机基础
3	大学计算机基础实践
4	大学英语
5	体育
6	网络技术与应用

04 外连接查询

查找没有被任何学生选修的课程的课程名称。

分二步：

(1) 使用右外连接查询所有课程记录，没有选课记录的以 NULL 值填充。

(2) 使用 IS NULL 可筛选出 Score 值为空的记录，即为没有被选修的课程。

查询语句：

```
SELECT CName  
FROM SCInfo b RIGHT OUTER JOIN CInfo c  
ON b.CNo =c .CNo  
WHERE Score IS NULL
```

	CName	Score
1	艺术设计史	NULL
2	民法学	NULL
3	数据库技术与应用	NULL
4	数据库应用基础实践	NULL
5	C++程序课程设计	NULL
6	C++程序设计基础	52
7	大学计算机基础	66
8	VB.NET程序设计	66
9	C++程序设计基础	67
10	...	



	Cname
1	C++程序课程设计
2	民法学
3	数据库技术与应用
4	数据库应用基础实践
5	艺术设计史

05 小结

- ◆ 如果查询中有统计函数，通常要使用GROUP BY；如果还要筛选，使用HAVING。
- ◆ JOIN连接时，确定是否需要外连接。
- ◆ 在WHERE后可以加子查询扩大条件选择范围。

06 专题总结

- ◆ 数据查询的基本框架包括SELECT、FROM、WHERE三个部分。
- ◆ 查询结果可以用ORDER BY子句排序，INTO保存新数据表，GROUP BY分组，HAVING筛选分组；UNION、EXCEPT、INTERSECT运算进行两个及以上结果集的合并、剔除、取重操作。
- ◆ 多表查询使用连接运算，JOIN关键字指定连接类型，有内连接、外连接、交叉连接。常用的内连接是等值连接，使用=作为条件运算符。
- ◆ 在WHERE后可以加子查询扩大条件选择范围，子查询即嵌套查询，分为单值嵌套查询和多值嵌套查询；多值嵌套查询需要使用基于集合的条件运算ANY、ALL、IN等。