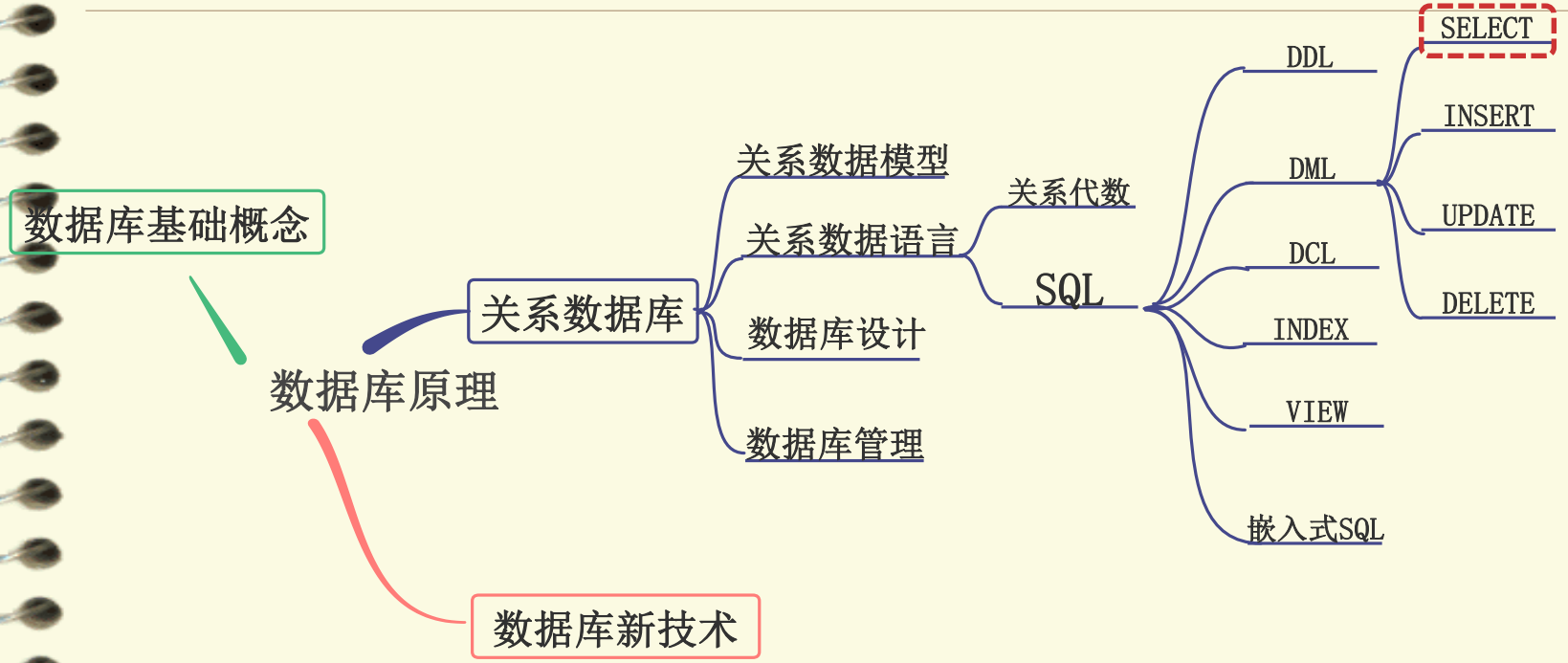
数据库原理

第4章 SQL --SELECT

辽东学院 鲁 琴

本节要点


数据库基础概念
数据库原理
数据库新技术
关系数据库
关系数据模型
关系数据语言
数据库设计
数据库管理
关系代数
SQL
DDL
DML
DCL
INDEX
VIEW
嵌入式SQL
SELECT
INSERT
UPDATE
DELETE

查询语句格式

```
Select A1,A2,…,An
From   R1,R2, …,Rm
Where  condition
Group By columns
Having condition
Order by columns
```

两者不同：

1.SELECT是**等值连接**

在 condition中要有等值连接条件

2.SELECT结果不是集合是包，有重复元组

$$\pi_{A_1,A_2,\ldots,A_n}(\sigma_{condition}(R_1 \infty R_2 \ \ldots \ \infty R_m))$$

示例数据库

teach数据库

- **学生表：**

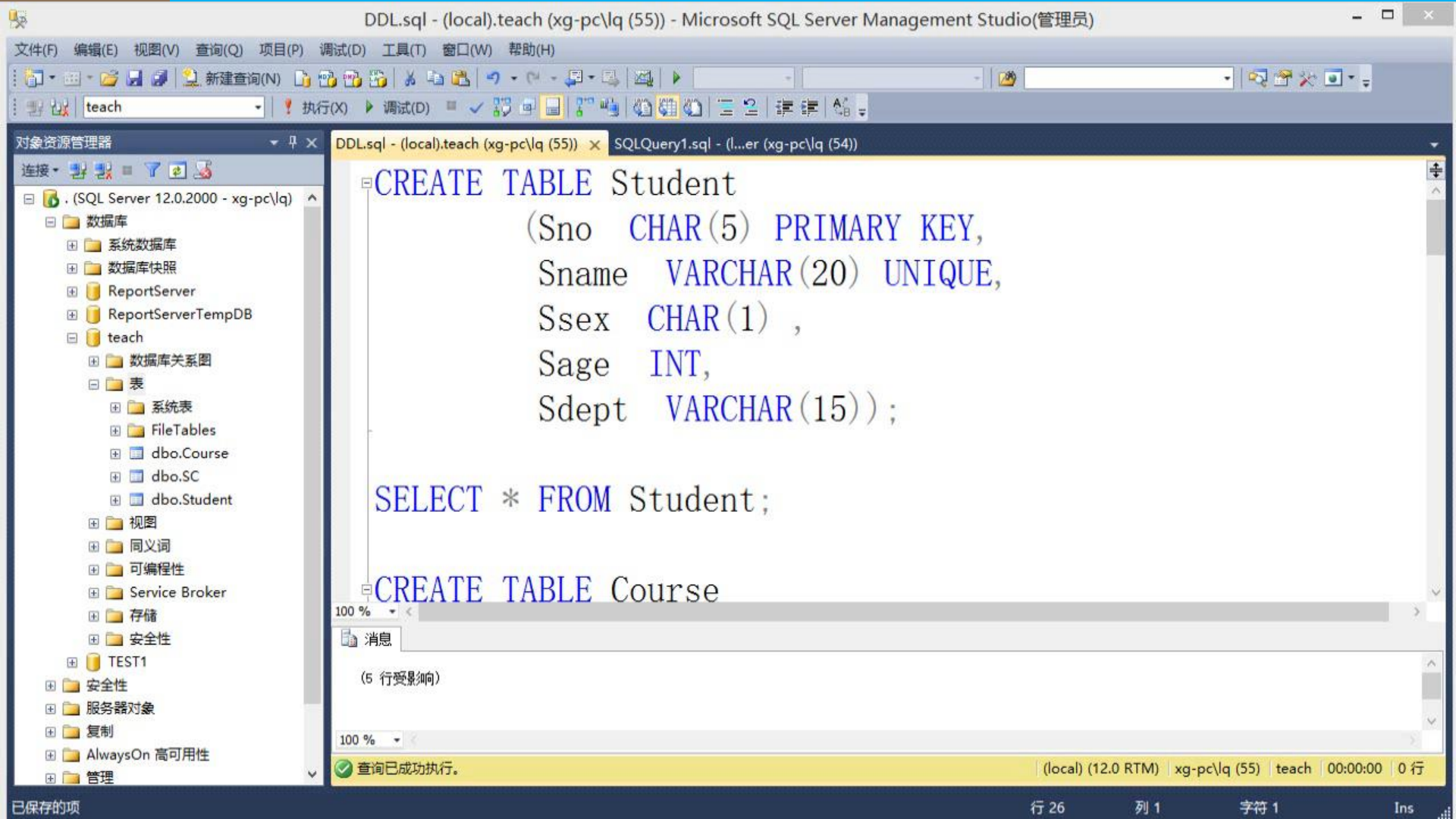
  Student(Sno，Sname，Ssex，Sage，Sdept)

- **课程表：**

  Course(Cno，Cname，Cpno，Ccredit)

- **学生选课表：**

  SC(Sno，Cno，Grade)


DDL.sql - (local).teach (xg-pc\lq (55)) - Microsoft SQL Server Management Studio(管理员)
文件(F) 编辑(E) 视图(V) 查询(Q) 项目(P) 调试(D) 工具(T) 窗口(W) 帮助(H)
新建查询(N)
teach
执行(X)
调试(D)
对象资源管理器
连接
. (SQL Server 12.0.2000 - xg-pc\lq)
数据库
系统数据库
数据库快照
ReportServer
ReportServerTempDB
teach
数据库关系图
表
系统表
FileTables
dbo.Course
dbo.SC
dbo.Student
视图
同义词
可编程性
Service Broker
存储
安全性
TEST1
安全性
服务器对象
复制
AlwaysOn 高可用性
管理
DDL.sql - (local).teach (xg-pc\lq (55))
SQLQuery1.sql - (l...er (xg-pc\lq (54))
CREATE TABLE Student
(Sno CHAR(5) PRIMARY KEY,
Sname VARCHAR(20) UNIQUE,
Ssex CHAR(1) ,
Sage INT,
Sdept VARCHAR(15));
SELECT * FROM Student;
CREATE TABLE Course
100 %
消息
(5 行受影响)
100 %
查询已成功执行。
(local) (12.0 RTM) xg-pc\lq (55) teach 00:00:00 0 行
已保存的项
行 26
列 1
字符 1
Ins

数 据 查 询

1 单表查询

2 连接查询

3 嵌套查询

4 集合查询

1 单表查询

- 查询仅涉及一个表，是一种最简单的查询操作
  - （1）选择表中的若干列 —SELECT子句
  - （2）选择表中的若干元组 —WHERE子句
  - （3）对查询结果排序 — ORDER BY子句
  - （4）使用集函数 —5个集函数
  - （5）对查询结果分组 —GROUP BY 子句
  - （6）对分组之后结果进行筛选 —HAVING子句

（1）选择表中的若干列

- 属投影运算
  - 不消除重复行
- SELECT子句查询指定属性列
  - 查询全部列
  - 查询经过计算的值

查询指定列

在**SELECT子句**中指定要查询的**属性列**

- 各个列的先后顺序可以与表中的逻辑顺序不一致
- 用户可以根据应用的需要改变列的显示顺序

例题

[例1] 查询全体学生的学号与姓名。

SELECT Sno,Sname FROM Student;

等价的关系代数表达式：

$\pi_{Sno,Sname}(Student)$

[例2] 查询全体学生的姓名、学号、所在系。

SELECT Sname,Sno,Sdept FROM Student;

等价的关系代数表达式：

$\pi_{Sname,Sno,Sdept}(Student)$

查询全部列

- 在**SELECT**关键字后面列出所有属性列名
- 当列的显示顺序与其在基表中的顺序相同时，也可以简单地将属性列表指定为 *
- * 代表全部列

例题

[例3] 查询全体学生的详细记录。

```
SELECT  Sno,Sname,Ssex,Sage,Sdept  FROM Student;
```

或

```
SELECT  * FROM Student;
```

查询经过计算的值

SELECT子句的属性列表为表达式

- 算术表达式
- 字符串常量
- 函数
- 列别名
- ……

例题

[例4] 查全体学生的姓名及其出生年份。

```
SELECT Sname,2020-Sage
FROM Student;
```

```
SELECT Sname,year(getdate()) -Sage
FROM Student;
```

输出结果：

Sname	（无列名）
李勇	1999
刘晨	2000
王名	2001
张立	2001

SQL Server函数：

year():返回指定日期的“年”日期部分的整数

getdate():返回当前的系统日期和时间

例题（续）

[例5]　查询全体学生的姓名、出生年份和所在系，要求用小写字母表示所有系名。

```
SELECT Sname,'Year of Birth: ',2020-Sage,LOWER(Sdept)
FROM Student;
```

Sname	(无列名)	(无列名)	(无列名)
李勇	Year of Birth:	1999	cs
刘晨	Year of Birth:	2000	is
王名	Year of Birth:	2001	ma
张立	Year of Birth:	2001	is

例题（续）

[例5.1] 使用列别名改变查询结果的列标题

```
SELECT Sname NAME,'Year of Birth: ' BIRTH,
       2020-Sage BIRTHDAY,
       LOWER(Sdept) DEPARTMENT
FROM Student;
```

NAME	BIRTH	BIRTHDAY	DEPARTMENT
李勇	Year of Birth:	1999	cs
刘晨	Year of Birth:	2000	is
王名	Year of Birth:	2001	ma
张立	Year of Birth:	2001	is

(2) 选择表中的若干元组

- 消除取值重复的行
- 查询满足条件的元组

消除取值重复的行

◆ 在SELECT子句中使用**DISTINCT**短语

[例6] 查询选修了课程的学生学号。

```
SELECT Sno  FROM SC;
```

或

```
SELECT ALL Sno   FROM SC;
```

Sno
95001
95001
95001
95002
95002

```
SELECT DISTINCT Sno
FROM SC;
```

Sno
95001
95002

例题

DISTINCT短语的作用范围是所有目标列

例：查询选修课程的各种成绩

错误的写法

```
SELECT DISTINCT Cno,DISTINCT Grade
FROM SC;
```

正确的写法

```
SELECT DISTINCT Cno,Grade   FROM SC;
```

查询语句格式

SELECT [ALL|DISTINCT]

<目标列表达式> [<别名>]

[，<目标列表达式>[<别名>]] ...

指定要显示的属性列

FROM <表名或视图名>[<别名>]

[， <表名或视图名>[<别名>]] ...

指定查询对象(基本表或视图)

[**WHERE** <条件表达式>]

指定查询条件

[**GROUP BY** <列名> [, <列名>] ...

对查询结果按指定列的值分组

对分组后的结果进行筛选

[**HAVING** <条件表达式>]]

[**ORDER BY** <列名> [, <列名>] ... [ASC|DESC]];

对查询结果表按指定列值的升序或降序排序

查询满足条件的元组

- 选择运算
- 通过 **WHERE Condition** 子句实现

查询条件Condition	谓　词
比较大小	=,>,<,>=,<=,!=,<>,!>,!<; NOT+上述比较运算符
确定范围	BETWEEN…AND…, NOT BETWEEN…AND…
确定集合	IN, NOT IN
字符串匹配	LIKE, NOT LIKE
空值查询	IS NULL, IS NOT NULL
多重条件查询	AND , OR

比较大小

- 在WHERE子句的Condition中使用比较运算符
  - =，>，<，>=，<=，!=或<>，!>，!<，
  - 逻辑运算符NOT + 含上述比较运算符的表达式

[例] 查询计算机系全体学生的名单。

```
SELECT Sname
FROM Student
WHERE Sdept = 'CS';
```

等价的关系代数表达式：

$\pi_{Sname}(\sigma_{Sdept = 'CS'}(Student))$

例题

[例7] 查询所有年龄在20岁以下的学生姓名及其年龄。

```
SELECT Sname,Sage
FROM Student
WHERE Sage < 20;
```

或

```
SELECT Sname,Sage
FROM Student
WHERE NOT Sage >= 20;
```

等价的关系代数表达式：

$$\pi_{Sname,Sage}(\sigma_{Sage<20}(Student))$$

例题

[例8] 查询考试成绩有不及格的学生的学号。

```
SELECT   DISTINCT Sno
FROM     SC
WHERE    Grade < 60;
```

确定范围

- 使用谓词
  - **BETWEEN … AND …**
  - **NOT BETWEEN … AND …**
    - **BETWEEN后：范围的下限（即低值）**
    - **AND后：范围的上限（即高值）**
- 用多重条件查询实现

例题

[例9] 查询年龄在20~23岁(包括20岁和23岁)之间的学生的姓名、系别和年龄

```
SELECT Sname,Sdept,Sage
FROM Student
WHERE Sage BETWEEN 20 AND 23;
```

Sname	Sdept	Sage
李勇	CS	20

例题（续）

[例10] 查询年龄不在20~23岁之间的学生姓名、系别和年龄

```
SELECT Sname,Sdept,Sage
FROM Student
WHERE Sage NOT BETWEEN 20 AND 23;
```

Sname	Sdept	Sage
刘晨	IS	19
王名	MA	18
张立	IS	18

确定集合

- 使用谓词　　IN <值表>

  NOT IN <值表>

  - <值表>：用逗号分隔的一组取值

- 用多重条件查询实现

例题

[例11] 查询信息系(IS)、数学系(MA)和计算机科学系(CS)学生的姓名和性别

```
SELECT Sname,Ssex
FROM Student
WHERE Sdept IN ('IS','MA','CS' );
```

Sname	Ssex
李勇	M
刘晨	F
王名	F
张立	M

例题

[例12]查询既不是信息系、数学系，也不是计算机科学系的学生的姓名和性别

```
SELECT Sname,Ssex
FROM Student
WHERE Sdept NOT IN ( 'IS','MA','CS' );
```

字符串匹配

使用谓词 **LIKE或NOT LIKE**

格式：

[NOT] LIKE '<匹配串>' [ESCAPE '<换码字符>']

- <匹配串>：指定匹配模板
  - 匹配模板：固定字符串或含通配符的字符串
  - 当匹配模板为固定字符串时，可以用=运算符取代LIKE谓词，用 != 或 < >运算符取代NOT LIKE谓词

例题：匹配模板为固定字符串

[例13] 查询学号为95001的学生的详细情况。

```
SELECT *
FROM  Student
WHERE  Sno LIKE '95001';
```

等价于：

```
SELECT  *
FROM  Student
WHERE Sno = '95001';
```

通配符

% (百分号) 代表任意长度（长度可以为0）的字符串

- **例：a%b表示以a开头，以b结尾的任意长度的字符串。如acb，addgb，ab 等都满足该匹配串**

_ (下横线) 代表任意单个字符

- **例：a_b表示以a开头，以b结尾的长度为3的任意字符串。如acb，afb等都满足该匹配串**

例题： 匹配模板为含通配符的字符串

[例14] 查询所有姓刘学生的姓名、学号和性别

```
SELECT Sname,Sno,Ssex
FROM Student
WHERE  Sname LIKE '刘%';
```

[例15] 查询所有不姓刘的学生姓名、学号和性别

```
SELECT Sname,Sno,Ssex
FROM Student
WHERE Sname NOT LIKE '刘%';
```

匹配模板为含通配符的字符串（续）

[例16] 查询姓"欧阳"且全名为三个汉字的学生的姓名

```
SELECT Sname
FROM   Student
WHERE  Sname LIKE '欧阳_ ';
```

[例17] 查询名字中第2个字为"阳"字的学生的姓名和学号

```
SELECT Sname，Sno
FROM Student
WHERE Sname LIKE '_ 阳%';
```

ESCAPE 短语

当用户要查询的字符串本身就含有 % 或 _ 时，

要使用**ESCAPE '<换码字符>'** 短语对通配符进行转义

例题:使用换码字符将通配符转义为普通字符

[例18] 查询DB_Design课程的课程号和学分。

```
SELECT Cno,Ccredit
FROM Course
WHERE Cname LIKE 'DB_Design'
```

?

```
SELECT Cno,Ccredit
FROM Course
WHERE Cname LIKE 'DB\_Design'   ESCAPE '\'
```

使用换码字符将通配符转义为普通字符(续)

[例19] 查询以"DB_"开头，且倒数第3个字符为 i的课程的详细情况

```
SELECT  *
FROM   Course
WHERE  Cname LIKE  'DB\_%i_ _'  ESCAPE ' \ ';
```

涉及空值的查询

- 使用谓词**IS NULL**或**IS NOT NULL**
- **“IS NULL” 不能用 “= NULL” 代替**

例题

[例20] 某些学生选修课程后没有参加考试，所以有选课记录，但没有考试成绩。查询缺少成绩的学生的学号和相应的课程号。

```
SELECT Sno,Cno
FROM SC
WHERE Grade IS NULL;
```

例题(续)

[例21] 查所有有成绩的学生学号和课程号

```
SELECT Sno,Cno
FROM  SC
WHERE  Grade IS NOT NULL;
```

多重条件查询

- 用逻辑运算符AND和OR来联结多个查询条件
  - AND的优先级高于OR
  - 可以用括号改变优先级
- 可用来实现多种其他谓词
  - [NOT] IN
  - [NOT] BETWEEN … AND …

例题

[例22] 查询计算机系年龄在20岁以下的学生姓名

```
SELECT Sname
 FROM  Student
 WHERE Sdept= 'CS' AND Sage<20;
```

例题（续）

[例23] 查询信息系（IS）、数学系（MA）和计算机科学系（CS）学生的姓名和性别

```
SELECT Sname，Ssex
FROM Student
WHERE Sdept IN ( 'IS'，'MA'，'CS' )
```

```
SELECT Sname，Ssex
FROM   Student
WHERE  Sdept= ' IS ' OR Sdept= ' MA' OR Sdept= ' CS ';
```

例题（续）

[例24] 查询年龄在20~23岁（包括20岁和23岁）之间的学生的姓名、系别和年龄

```
SELECT Sname,Sdept,Sage
 FROM Student
 WHERE Sage BETWEEN 20 AND 23;
```

```
SELECT Sname，Sdept，Sage
FROM Student
WHERE  Sage>=20 AND Sage<=23;
```

（3）对查询结果排序

- 使用**ORDER BY**子句
  - 可以按一个或多个属性列排序
  - 升序：**ASC**；降序：**DESC**；缺省值为升序
- 当排序列含空值时
  - **ASC**：排序列为空值的元组最后显示
  - **DESC**：排序列为空值的元组最先显示

例题

[例25] 查询选修了3号课程的学生的学号及其成绩，查询结果按分数降序排列

```
SELECT Sno,Grade
    FROM  SC
    WHERE  Cno= 3
    ORDER BY Grade DESC;
```

Sno	Grade
95001	88.0
95002	80.0

例题

[例26] 查询全体学生情况，查询结果按所在系的系号升序排列，同一系中的学生按年龄降序排列。

```
SELECT *
FROM Student
ORDER BY Sdept,Sage DESC;
```

Sno	Sname	Ssex	Sage	Sdept
95001	李勇	M	20	CS
95002	刘晨	F	19	IS
95004	张立	M	18	IS
95003	王名	F	18	MA

（4）使用集函数

5类主要集函数

- 统计个数
  COUNT（<列名>）
- 计算列总和
  SUM（<列名>）
- 计算列平均值
  AVG（<列名>）
- 求列最大值
  MAX（<列名>）
- 求列最小值
  MIN（<列名>）

统计时不计NULL值

COUNT(*)

统计行数

DISTINCT

在列名前加DISTINCT

则统计时去掉重复行

使用集函数（续）

[例27] 查询学生总人数。

```
SELECT COUNT(*)
FROM  Student;
```

（无列名）
4

[例28] 查询选修了课程的学生人数。

```
SELECT COUNT(DISTINCT Sno)
FROM SC;
```

（无列名）
2

注：用DISTINCT以避免重复计算学生人数

使用集函数 （续）

[例29] 计算1号课程的学生平均成绩

```
SELECT AVG(Grade)
FROM SC
WHERE Cno= 1 ;
```

[例30] 查询选修1号课程的学生最高分数

```
SELECT MAX(Grade)
FROM SC
WHER Cno= 1 ;
```

（5）对查询结果分组

- 用途
  - 细化集函数的作用对象
    - 未对查询结果分组，集函数将作用于整个查询结果
    - 对查询结果分组后，集函数将分别作用于每个组

对查询结果分组 （续）

- 使用**GROUP BY**子句分组
  - 分组方法：按指定的一列或多列值分组，值相等的为一组
  - 使用**GROUP BY**子句后，**SELECT**子句的列名列表中只能出现分组属性和集函数
  - **GROUP BY**子句的作用对象是查询的中间结果表

例题

[例31] 求各个课程号及相应的选课人数。

```
SELECT Cno,COUNT(Sno)
FROM SC
GROUP BY Cno;
```

Cno	(无列名)
1	1
2	2
3	2

例题

[例32] 求各个课程号及相应的课程成绩在90分以上的学生人数

```
SELECT Cno,COUNT(Sno)
FROM SC
WHERE Grade>=90
GROUP BY Cno;
```

Cno	（无列名）
1	1
2	1

对查询结果分组 （续）

- 使用**HAVING**短语筛选最终输出结果
  - 只有满足**HAVING**短语指定条件的组才输出
- **HAVING**短语与**WHERE**子句的区别：作用对象不同
  - **WHERE**子句作用于基表或视图，从中选择满足条件的元组
  - **HAVING**短语作用于组，从中选择满足条件的组

例题

[例33] 查询选修了3门以上课程的学生学号

```
SELECT Sno
FROM SC
GROUP BY Sno
HAVING COUNT(*) >=3;
```

Sno
95001

例题

[例34] 查询有3门以上课程在90分以上的学生的学号及90分以上的课程数

```
SELECT Sno, COUNT(*)
FROM SC
WHERE Grade>=90
GROUP BY Sno
HAVING COUNT(*)>=3;
```

［例35］统计每门课程的最高分

SELECT cno,max(grade)

FROM SC

GROUP BY cno

结果：

cno	（无列名）
1	92.0
2	90.0
3	88.0

查询语句格式

SELECT [ALL|DISTINCT]

<目标列表达式> [<别名>]

[，<目标列表达式>[<别名>]] ...

指定要显示的属性列

FROM <表名或视图名>[<别名>]

[， <表名或视图名>[<别名>]] ...

指定查询对象(基本表或视图)

[**WHERE** <条件表达式>]

指定查询条件

[**GROUP BY** <列名> [, <列名>] ...

对查询结果按指定列的值分组

对分组后的结果进行筛选

[**HAVING** <条件表达式>]]

[**ORDER BY** <列名> [, <列名>] ... [ASC|DESC]];

对查询结果表按指定列值的升序或降序排序