

查询结果显示



引言

一个完整的数据查询语句的格式

```
SELECT [ ALL|DISTINCT ] <目标列表达式1> [, <目标列表达2> , ... ]  
FROM <表名或视图名1> [, <表名或视图名2> , ... ]  
[ WHERE <元组选择条件表达式> ]  
[ GROUP BY <属性列名1> [, <属性列名2> , ... ] [ HAVING <组选择  
    条件表达式> ] ]  
[ ORDER BY <目标列名1> [ASC|DESC] [, <目标列名2>  
    [ASC|DESC] , ... ] ]
```



讲授内容

- 1 投影属性列
- 2 去除重复元组
- 3 查询结果的计算
- 4 目标列命名
- 5 查询结果的排序



实验数据库

学生选课数据库

S (SNO, SN, SD, SB, SEX)

C (CNO, CN, PC)

SC (SNO, CNO, GRADE)



投影属性列

- SELECT [ALL|DISTINCT] <目标列表达式1> [,<目标列表达式2>, ...]
FROM <表名> 属性名

► 查询显示所有学生的信息

SELECT SNO,SN,SD,SB, SEX
FROM S

```
SELECT SNO, SN, SD, SB, SEX  
FROM S;
```

100 %

结果 消息

	SNO	SN	SD	SB	SEX
1	s01	王玲	计算机	2000-06-30	女
2	s02	李渊	计算机	1995-03-23	男
3	s03	罗军	计算机	1995-08-12	男
4	s04	赵泽	计算机	1997-09-12	女
5	s05	许若	自动化	2000-06-27	男
6	s06	王仙华	自动化	1996-05-20	男
7	s07	朱祝	自动化	1998-07-10	女
8	s08	王明	数学	1998-10-03	男
9	s09	王学之	物理	1996-01-01	男
10	s10	吴谦	自动化	1996-03-25	女



投影属性列

- SELECT [ALL|DISTINCT] <目标列表达式1> [, <目标列表达式2> , ...]
FROM <表名> 属性名

▶ 查询显示所有学生的信息

SELECT * FROM S

显示基本表中所有属性的信息

SELECT *
FROM S;

100 %

结果 消息

	SNO	SN	SD	SB	SEX
1	s01	王玲	计算机	2000-06-30	女
2	s02	李渊	计算机	1995-03-23	男
3	s03	罗军	计算机	1995-08-12	男
4	s04	赵泽	计算机	1997-09-12	女
5	s05	许若	自动化	2000-06-27	男
6	s06	王仙华	自动化	1996-05-20	男
7	s07	朱祝	自动化	1998-07-10	女
8	s08	王明	数学	1998-10-03	男
9	s09	王学之	物理	1996-01-01	男
10	s10	吴谦	自动化	1996-03-25	女



去除重复元组

- SELECT [ALL|DISTINCT] <目标列表达式1> [, <目标列表达式2> , ...]
FROM <表名>

去掉结果表中的
重复行

▶ 查询选修了课程的学生学号

SELECT SNO FROM SC

SELECT **DISTINCT** SNO
FROM SC

	SNO
1	s01
2	s02
3	s03

	SNO
1	s01
2	s01
3	s01
4	s01
5	s01
6	s01
7	s02
8	s02
9	s03
10	s03



查询结果的计算

- SELECT [ALL|DISTINCT] <目标列表达式1> [, <目标列表达式2> , ...]
FROM <表名>

聚集函数

COUNT(*)

统计元组个数

COUNT([DISTINCT|ALL] <列名>)

统计一列中值的个数

SUM([DISTINCT|ALL] <列名>)

计算一列值的总和 (此列必须是数值型)

AVG([DISTINCT|ALL] <列名>)

计算一列值的平均值 (此列必须是数值型)

MAX([DISTINCT|ALL] <列名>)

求一列值中的最大值

MIN([DISTINCT|ALL] <列名>)

求一列值中的最小值



查询结果的计算

▶ 统计学生的总人数

```
SELECT COUNT(*)  
FROM S
```

SELECT COUNT(*)
FROM S;

100 %

结果 消息

(无列名)	
1	21

▶ 查询选修了课程的学生人数

```
SELECT COUNT(DISTINCT SNO)  
FROM SC
```

SELECT COUNT(DISTINCT SNO)
FROM SC;

100 %

结果 消息

(无列名)	
1	19



查询结果的计算

- SELECT [ALL|DISTINCT] <目标列表达式1> [, <目标列表达式2> , ...]
FROM <表名>

► 查询所有学生的学号和年龄

```
SELECT SNO, datediff(Year, SB, Getdate()+1)  
FROM S;
```

获取系统当前的日期

datediff(datepart, startdate, enddate)

```
SELECT SNO,datediff(Year,SB,Getdate()+1)  
FROM S;
```

100 %

结果 消息

	SNO (无列名)	
1	s01	20
2	s02	25
3	s03	25
4	s04	23
5	s05	21
6	s06	24
7	s07	22
8	s08	22



目标列命名

► 查询学生学号和学生的年龄。

```
SELECT SNO AS 学号, 年龄=datediff(Year, SB, Getdate()+1)  
FROM S;
```

```
SELECT SNO 学号, 年龄=datediff(Year, SB, Getdate()+1)  
FROM S;
```

SQL Query: `SELECT SNO AS 学号, 年龄=datediff(Year, SB, Getdate()+1) FROM S;`

	学号	年龄
1	s01	20
2	s02	25
3	s03	25
4	s04	23
5	s05	21
6	s06	24
7	s07	22
8	s08	22

SQL Query: `SELECT SNO 学号, 年龄=datediff(Year, SB, Getdate()+1) FROM S;`

	学号	年龄
1	s01	20
2	s02	25
3	s03	25
4	s04	23
5	s05	21
6	s06	24
7	s07	22
8	s08	22



查询结果的排序

- SELECT [ALL|DISTINCT]<目标列表达式1>[,<目标列表达式2>, ...]
FROM <表名>
ORDER BY <目标列1> [ASC|DESC][, <目标列2 >[ASC|DESC], ...]

先按第一列排序

升序

降序

第一列相同，再按第二列排序



查询结果的排序

► 查询所有选修信息，按课程号升序排列

```
SELECT * FROM SC
```

The screenshot shows a database query window with the SQL statement `SELECT * FROM SC;` entered. Below the statement, there are tabs for '结果' (Results) and '消息' (Messages). The '结果' tab is active, displaying a table with 12 rows of data. The columns are SNO, CNO, and GRADE. The data is sorted by CNO in ascending order. The first row has SNO 's01', CNO 'c01', and GRADE '93.0'. The last row has SNO 's03', CNO 'C05', and GRADE 'NULL'.

	SNO	CNO	GRADE
1	s01	c01	93.0
2	s01	c02	98.0
3	s01	c03	85.0
4	s01	c04	78.0
5	s01	C05	NULL
6	s01	c07	89.0
7	s02	C05	NULL
8	s02	c11	80.0
9	s03	c01	100.0
10	s03	c02	80.0
11	s03	c04	85.0
12	s03	C05	NULL



查询结果的排序

► 查询所有选修信息，按课程号升序排列

```
SELECT * FROM SC
ORDER BY CNO
```

SELECT *
FROM SC
ORDER BY CNO;

100 %

结果 消息

	SNO	CNO	GRADE
1	s01	c01	93.0
2	s03	c01	100.0
3	s04	c01	80.0
4	s08	c01	90.0
5	s11	c01	90.0
6	s15	c01	80.0
7	s16	c01	80.0
8	s13	c01	86.5
9	s15	c02	79.0
10	s08	c02	95.0
11	s04	c02	NULL

SELECT *
FROM SC;

100 %

结果 消息

	SNO	CNO	GRADE
1	s01	c01	93.0
2	s01	c02	98.0
3	s01	c03	85.0
4	s01	c04	78.0
5	s01	C05	NULL
6	s01	c07	89.0
7	s02	C05	NULL
8	s02	c11	80.0
9	s03	c01	100.0
10	s03	c02	80.0
11	s03	c04	85.0
12	s03	C05	NULL



查询结果的排序

- 查询所有选修信息，按课程号升序排列，同一课程内的学生成绩降序排列。

```
SELECT *  
FROM SC  
ORDER BY CNO, GRADE DESC
```

```
SELECT *  
FROM SC  
ORDER BY 2, 3 DESC
```

SELECT *
FROM SC
ORDER BY CNO, GRADE DESC;

	SNO	CNO	GRADE
1	s03	c01	100.0
2	s01	c01	93.0
3	s08	c01	90.0
4	s11	c01	90.0
5	s13	c01	86.5
6	s15	c01	80.0
7	s16	c01	80.0
8	s04	c01	80.0
9	s01	c02	98.0
10	s08	c02	95.0
11	s03	c02	80.0
12	s15	c02	79.0
13	s04	c02	NULL

SELECT *
FROM SC
ORDER BY 2, 3 DESC;


	SNO	CNO	GRADE
1	s03	c01	100.0
2	s01	c01	93.0
3	s08	c01	90.0
4	s11	c01	90.0
5	s13	c01	86.5
6	s15	c01	80.0
7	s16	c01	80.0
8	s04	c01	80.0
9	s01	c02	98.0
10	s08	c02	95.0
11	s03	c02	80.0
12	s15	c02	79.0
13	s04	c02	NULL



查询结果的排序

- 查询所有学生的学号和年龄，并按年龄升序显示。

```
SELECT SNO, datediff(Year, SB, Getdate())+1  
FROM S  
ORDER BY 2
```



The screenshot shows a SQL query editor with the following query:

```
SELECT SNO,datediff(Year,SB,Getdate())+1  
FROM S  
ORDER BY 2
```

Below the query, there are tabs for "结果" (Results) and "消息" (Messages). The "结果" tab is active, displaying a table with 8 rows of data. The table has three columns: "SNO", "(无列名)" (unnamed), and an unnamed column. The data is sorted by the unnamed column in ascending order.

	SNO	(无列名)
1	s01	20
2	s16	20
3	s21	20
4	s51	20
5	s20	21
6	s15	21
7	s11	21
8	s05	21



小结

SELECT [ALL|DISTINCT] <目标列表达式1> [, <目标列表达式2> , ...]

FROM <表名>

ORDER BY <目标列1> [ASC | DESC] [, <目标列2> [ASC | DESC], ...]