

8.2 数据操作

主要考点

- 1、SELECT基本结构
- 2、简单查询
- 3、连接查询
- 4、子查询
- 5、聚集函数
- 6、分组查询
- 7、字符串操作
- 8、集合操作
- 9、外连接
- 10、INSERT INTO语句
- 11、DELETE语句
- 12、UPDATE语句

SQL的数据操作功能包括SELECT（查询）、INSERT（插入）、DELETE（删除）和UPDATE（修改）4条语句。

学生表S

Sno	Sname	Sex	SD	Age
3001	王平	女	计算机	18
3002	张勇	男	计算机	19
4003	黎明	女	机械	18
4004	刘明远	男	机械	19
1041	赵国庆	男	通信	20
1042	樊建玺	男	通信	20

课程表C

Cno	Cname	Pcno	Credit
C1	数据库	C3	3
C2	数学		4
C3	操作系统	C4	4
C4	数据结构	C7	3
C5	数字通信	C6	3
C6	信息系统	C1	4
C7	程序设计	C2	2

选课表SC

Sno	Cno	Grade
3001	C1	93
3001	C2	84
3001	C3	84
3002	C2	83
3002	C3	93
1042	C1	84
1042	C2	82

教师表T

Tno	Tname	Tage	Sex
C001	王芳	52	男
C002	刘平	34	女

Select 基本结构

语句格式： SELECT [ALL | DISTINCT]<目标列表达式>[, <目标列表达式>]...
FROM <表名或视图名>[, <表名或视图名>]
[WHERE <条件表达式>]
[GROUP BY <列名1>[HAVING <条件表达式>]]
[ORDER BY <列名2>[ASC | DESC]...]

运算符		含义	运算符		含义
集合成员运算符	IN NOT IN	在集合中 不在集合中	算术运算符	> >= < <= = <>	大于 大于等于 小于 小于等于 等于 不等于
字符串匹配运算符	LIKE	与_、%进行 单个、多个字 符匹配			
空值比较运算符	IS NULL IS NOT NULL	为空 不为空	逻辑运算符	AND OR NOT	与 或非

表： WHERE子句的条件表达式中可以使用的运算符

简单查询

最基本的查询语句：

```
Select Sno, Sname  
from S  
where sex='男';
```

//等价于： $\pi_{Sno, Sname}(\sigma_{sex='男'}(S))$

更名操作：

使用AS可以给列和表重命名。

```
Select 学生表.sno as 学号, Sname as 姓名, Grade 成绩  
from S 学生表, SC 课程表  
where 学生表.Sno=课程表.Sno and 学生表.SD='计算机';
```

//AS是可以省略的

//表也可以重命名

//表重命名后，引用时一定要用
新名称，这里不能再S和SC

连接查询

如果涉及两个以上的表，则称为连接查询。

例：

(1) 检索选修了课程号为'C1'的学生号和学生姓名：

```
Select S. Sno, Sname
from S, SC
where S. Sno=SC. Sno and SC. Cno='C1' ;
```

// $\pi_{S. Sno, Sname} (\sigma_{S. Sno=SC. Sno \wedge SC. Cno='C1'} (S \times SC))$

(2) 检索选修课程名为“操作系统”的学生号和学生姓名：

```
Select S. Sno, Sname
from S, SC, C
where S. Sno=SC. Sno and SC. Cno=C. Cno and C. Cname='操作系统' ;
```

(3) 检索至少选修了课程号为“C1”和“C3”的学生号：

```
Select Sno
from SC SC_X, SC SC_Y
where SC_X. Sno=SC_Y. Sno and SC_X. Cno='C1' and SC_Y. Cno='C3' ;
```

学生表s

Sno	Sname	Sex	SD	Age
3001	王平	女	计算机	18
3002	张勇	男	计算机	19
4003	黎明	女	机械	18
4004	刘明远	男	机械	19
1041	赵国庆	男	通信	20
1042	樊建玺	男	通信	20

选课表SC

Sno	Cno	Grade
3001	C1	93
3001	C2	84
3001	C3	84
3002	C2	83
3002	C3	93
1042	C1	84
1042	C2	82

子查询

- 子查询也称嵌套查询。嵌套查询是指一个SELECT-FROM-WHERE查询块中可以嵌入另一个查询块之中。在SQL中允许多重嵌套。

- IN (在……集合中) | NOT IN (不在……集合中)

例：（1）检索选修了课程号为'C1'的学生号和学生姓名：

```
Select Sno, Sname
from S
where Sno IN (Select Sno
              from SC
              Where Cno='C1');
```

（2）检索选修课程名为“操作系统”的学生号和学生姓名：

```
Select Sno, Sname
from S
where Sno IN (Select Sno
              from SC
              Where Cno IN (Select Cno
                           From C
                           Where Cname='操作系统'));
```


子查询

- EXISTS（存在） | NOT EXISTS（不存在）
- 例：查询没有被任何同学选修的课程号和课程名
- `Select Cno, Cname from C`
- `where NOT EXISTS (select * from SC where SC.Cno=C.Cno)`

课程表C

Cno	Cname	Pcno	Credit
C1	数据库	C3	3
C2	数学		4
C3	操作系统	C4	4
C4	数据结构	C7	3
C5	数字通信	C6	3
C6	信息系统	C1	4
C7	程序设计	C2	2

选课表SC

Sno	Cno	Grade
3001	C1	93
3001	C2	84
3001	C3	84
3002	C2	83
3002	C3	93
1042	C1	84
1042	C2	82

Cno	Cname
C4	数据结构
C5	数字通信
C6	信息系统
C7	程序设计

聚集函数

- 聚集函数是一个值的集合为输入，返回单个值的函数。SQL提供了5个预定义集函数：
平均值AVG()、最小值MIN()、最大值MAX()、求和SUM()、计数COUNT()。

集函数名	功能
COUNT ([DISTINCT ALL]*)	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>)	计算一列中值的总和
AVG ([DISTINCT ALL] <列名>)	计算一列中值的平均值
MAX ([DISTINCT ALL] <列名>)	求一列值的最大值
MIN ([DISTINCT ALL] <列名>)	求一列值的最小值

表：集函数的功能

聚集函数

- 例：（1）查询课程C1的最高分和最低分以及高低分之间的差距

```
Select MAX(Grade), MIN(Grade), MAX(Grade) - MIN(Grade)
From SC
Where Cno='C1';
```

- （2）查询其他系比计算机系所有学生年龄都要小的学生姓名和年龄。

```
Select Sname, Age
From S
Where Age < ALL (Select Age
                  From S
                  Where SD='计算机')
AND SD <> '计算机';
```

```
Select Sname, Age
From S
Where Age < (SELECT MIN(Age)
              From S
              Where SD='计算机')
AND SD <> '计算机';
```

学生表S

Sno	Sname	Sex	SD	Age
3001	王平	女	计算机	18
3002	张勇	男	计算机	19
4003	黎明	女	机械	18
4004	刘明远	男	机械	19
1041	赵国庆	男	通信	17
1042	樊建玺	男	通信	20

选课表SC

Sno	Cno	Grade
3001	C1	93
3001	C2	84
3001	C3	84
3002	C2	83
3002	C3	93
1042	C1	84
1042	C2	82

聚集函数

谓 词	等价的聚集函数	语义
$>/>= ANY$	$>/>= MIN$	大于/大于等于子查询结果中的某个值
$>/>= ALL$	$>/>= MAX$	大于/大于等于子查询结果中的所有值
$</<= ANY$	$</<= MAX$	小于/小于等于子查询结果中的某个值
$</<= ALL$	$</<= MIN$	小于/小于等于子查询结果中的所有值
$<>ANY$	--	不等于子查询结果中的某个值
$<>ALL$	NOT IN	不等于子查询结果中的任何一个值
$=ANY$	IN	等于子查询结果中的某个值
$=ALL$	--	等于子查询结果中的所有值

表：ANY、ALL谓词含义及等价的转换关系

分组查询

- 1、GROUP BY子句
- 对元组进行分组，此时聚集函数只用于每个分组。

(1) `Select AVG(Grade)`
`from SC;`

(2) `Select Sno, AVG(Grade)`
`from SC`
`GROUP BY Sno;`

- 2、HAVING子句：
- 假如元组在分组前按照某种方式加上限制，使得不需要的分组为空。可以在GROUP BY子句后面加上HAVING子句。

其实就是分组后做一个选择，去掉不满足HAVING后面的那个条件的分组。

(3) `Select Sno, AVG(Grade)`
`from SC`
`GROUP BY Sno`
`HAVING AVG(Grade)>85;`

AVG(Grade)
86

(1)

Sno	AVG(Grade)
3001	87
3002	88
1042	83

(2)

Sno	AVG(Grade)
3001	87
3002	88

(3)

选课表SC

Sno	Cno	Grade
3001	C1	93
3001	C2	84
3001	C3	84
3002	C2	83
3002	C3	93
1042	C1	84
1042	C2	82

字符串操作

- 字符串常用LIKE来进行匹配操作，“%”匹配任意长度字符串；“_”匹配任意一个字符。模式是区分大小写的。

- 例：

- (1) 查询住址包含“科技路”的学生姓名：

```
Select Sname  
from S  
Where Addr LIKE '%科技路%';
```

- (2) 查询名字为‘国庆’的学生姓名、年龄和所在系：

```
Select Sname, Sage, SD  
from S  
Where Sname LIKE '__国庆';
```

为了使模式中包含特殊字符，即%和_，允许使用ESCAPE关键词来定义转义符，LIKE语句后面紧跟“ESCAPE '\’”表明转义符'\’后面的特殊字符为普通字符。

```
LIKE 'ab\%cd%' escape '\' //匹配所有以ab%cd开头的字符串  
LIKE 'ab\\cd%' escape '\' //匹配所有以ab\cd开头的字符串
```


集合操作

例：贷款(customer-no, loan-no)

存款(customer-no, account-no)

1、UNION（并）、UNION ALL（不去掉重复行）

查询在银行有贷款或者有存款的所有客户身份证号：

```
select customer-no from 贷款
union
```

```
select customer-no from 存款;
```

2、INTERSECT（交）

查询在银行既有贷款又有存款的客户身份证号：

```
select customer-no from 贷款
intersect
```

```
select customer-no from 存款;
```

3、EXCEPT（差）

查询只在银行有贷款而没有存款的客户身份证号：

```
select customer-no from 贷款
except
```

```
select customer-no from 存款;
```

贷款表

customer-no	loan-no
31038734	L001
31025996	L002

存款表

customer-no	account-no
31038734	A001
30012332	A002

UNION

customer-no
31038734
31025996
30012332

UNION ALL

customer-no
31038734
31025996
31038734
30012332

INTERSECT

customer-no
31038734

EXCEPT

customer-no
31025996

外连接

1、LEFT JOIN(左外连接)

```
SELECT S. Sno, Sname, SC. Cno, Grade
FROM S
LEFT JOIN SC
ON S. Sno=SC. Sno;
```

学生表S

Sno	Sname	Sex	SD	Age
3001	王平	女	计算机	18
3002	张勇	男	计算机	19
4003	黎明	女	机械	18
4004	刘明远	男	机械	19
1041	赵国庆	男	通信	20
1042	樊建玺	男	通信	20

选课表SC

Sno	Cno	Grade
3001	C1	93
3001	C2	84
3001	C3	84
3002	C2	83
3002	C3	93
1042	C1	84
1042	C2	82

S LEFT JOIN SC

Sno	Sname	Cno	Grade
3001	王平	C1	93
3001	王平	C2	84
3001	王平	C3	84
3002	张勇	C2	83
3002	张勇	C3	93
4003	黎明	NULL	NULL
4004	刘明远	NULL	NULL
1041	赵国庆	NULL	NULL
1042	樊建玺	C1	84
1042	樊建玺	C2	82

外连接

2、RIGHT JOIN(右外连接)

```
SELECT S. Sno, Sname, SC. Cno, Grade
FROM S
RIGHT JOIN SC
ON S. Sno=SC. Sno;
```

3、FULL JOIN（全外连接）

```
SELECT S. Sno, Sname, SC. Cno, Grade
FROM S
FULL JOIN SC
ON S. Sno=SC. Sno;
```

S RIGHT JOIN SC

Sno	Sname	Cno	Grade
3001	王平	C1	93
3001	王平	C2	84
3001	王平	C3	84
3002	张勇	C2	83
3002	张勇	C3	93
1042	樊建玺	C1	84
1042	樊建玺	C2	82

INSERT INTO语句

INSERT INTO语句用于向表中插入新的行。

语句格式： **INSERT INTO** 表名 **VALUES** (值1, 值2...)

INSERT INTO 表名(列名1, 列名2...) **VALUES** (值1, 值2...)

例： **INSERT INTO SC**

VALUES ('3001', 'C1', '84');

INSERT INTO SC(SNO, CNO)

VALUES ('3001', 'C2');

选课表SC

Sno	Cno	Grade
1042	C1	84
1042	C3	82

(a)

选课表SC

Sno	Cno	Grade
1042	C1	84
1042	C3	82
3001	C1	84

(b)

选课表SC

Sno	Cno	Grade
1042	C1	84
1042	C3	82
3001	C1	84
3001	C2	NULL

(c)

DELETE语句

- DELETE语句用于删除表中的行。
- 语句格式：

DELETE FROM 表名 WHERE 条件表达式

例：

DELETE FROM SC WHERE Cno='C1' ;

- 也可以删除表中的所有数据：

DELETE FROM 表名

例：

DELETE FROM SC ;

注意：不同于DROP TABLE SC，DELETE只是删除表中的数据，但是关系模式仍然存在，表SC仍然存在于数据库中。

选课表SC

Sno	Cno	Grade
1042	C1	84
1042	C3	82
3001	C1	84
3001	C2	NULL

(a)

选课表SC

Sno	Cno	Grade
1042	C3	82
3001	C2	NULL

(b)

选课表SC

Sno	Cno	Grade
-----	-----	-------

(c)

UPDATE语句

- UPDATE语句用于修改表中的数据。
- 语句格式：

UPDATE 表名 SET 列名=新值 WHERE 条件表达式

- 例：

UPDATE SC

SET Grade = 90

WHERE Sno='3001' AND Cno='C2' ;

选课表SC

Sno	Cno	Grade
1042	C1	84
1042	C3	82
3001	C1	84
3001	C2	NULL

(a)

选课表SC

Sno	Cno	Grade
1042	C1	84
1042	C3	82
3001	C1	84
3001	C2	90

(b)