

主要内容

- 3.1 SQL概述
- 3.2 数据定义
- 3.3 数据查询
- 3.4 数据更新
- 3.5 视图

3.3 数据查询

- 3.3.1 单表查询
- 3.3.2 连接查询
- 3.3.3 嵌套查询
- 3.3.4 集合查询
- 3.3.5 基于派生表的查询

```
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> ]...|(SELECT 语句)  
[AS]<别名>  
[ WHERE <条件表达式> ]  
[ GROUP BY <列名1> [ HAVING <  
条件表达式> ] ]  
[ ORDER BY <列名2>  
[ ASC|DESC ] ];
```

3.3 数据查询

- 3.3.1 单表查询
 - 选择表中的若干列
 - 选择表中的若干元组
 - ORDER BY子句
 - 聚集函数
 - GROUP BY子句

3.3 数据查询

• 3.3.1 单表查询

- 选择表中的若干列

* 别名

--单表查询（查询指定列）

```
SELECT Sno, Sname  
FROM Student;
```

	Sno	Sname
1	201215121	李勇
2	201215122	刘晨
3	201215123	王敏
4	201215125	张立

--单表查询（查询所有列）

```
SELECT *  
FROM Student;
```

	Sno	Sname	Ssex	Sage	Sdept
1	201215121	李勇	男	20	CS
2	201215122	刘晨	女	20	CS
3	201215123	王敏	女	20	MA
4	201215125	张立	男	20	IS

	Sname	(无列名)	Birthday	(无列名)
1	李勇	Year of Birth	1998	cs
2	刘晨	Year of Birth	1999	cs
3	王敏	Year of Birth	2000	ma
4	张立	Year of Birth	1999	is

--单表查询（查询经过计算的值）

```
SELECT Sname, 'Year of Birth', 2018-Sage  
Birthday, LOWER(Sdept)  
FROM Student;
```


3.3 数据查询

• 3.3.1 单表查询

- 选择表中的若干元组

DISTINCT WHERE

—单表查询（消除重复行）

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

	Sno
1	201215121
2	201215122

查 询 条 件	谓 词
比 较	=, >, <, >=, <=, !=, <>, !>, !<; NOT +上述比较运算符
确定范围	BETWEEN AND, NOT BETWEEN AND
确定集合	IN, NOT IN
字符匹配	LIKE, NOT LIKE
空 值	IS NULL, IS NOT NULL
多重条件（逻辑运算）	AND, OR, NOT

3.3 数据查询

• 3.3.1 单表查询

- 选择表中的若干元组

—查询计算机系全体同学姓名

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

	Sname
1	李勇
2	刘晨

—查询计算机系、数学系、信息系学生的姓名和性别

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

DISTINCT **WHERE**

—查询年龄在20-23岁的学生姓名、系别、年龄

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

	Sname	Sdept	Sage
1	李勇	CS	20

	Sname	Ssex
1	李勇	男
2	刘晨	女
3	王敏	女
4	张立	男

3.3 数据查询

• 3.3.1 单表查询

- 选择表中的若干元组

- %

- _

--查询刘姓同学的姓名、学号和性别

```
SELECT Sname, Sno, Ssex
```

```
FROM Student
```

```
WHERE Sname LIKE '刘%';
```

	Sname	Sno	Ssex
1	刘晨	201215122	女

--查询DB_Design课程的课程号和学分。

```
SELECT Cno, Ccredit
```

```
FROM Course
```

```
WHERE Cname LIKE 'DB\_Design' ESCAPE '\';
```

DISTINCT **WHERE**

--查询不姓刘同学且名为单字的姓名、学号和性别

```
SELECT Sname, Sno, Ssex
```

```
FROM Student
```

```
WHERE Sname NOT LIKE '刘_';
```

	Sname	Sno	Ssex
1	李勇	201215121	男
2	王敏	201215123	女
3	张立	201215125	男

'DB_%i__' ESCAPE '\ ' ?

3.3 数据查询

• 3.3.1 单表查询

- 选择表中的若干元组

DISTINCT **WHERE**

--查询先行课为空的课程号和课程名

```
SELECT Cno, Cname
```

```
FROM Course
```

```
WHERE Cpno IS NULL;
```

	Cno	Cname
1	2	数学
2	6	数据处理

--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;
```

	Sname
1	刘晨

--查询计算机系、数学系、信息系学生的姓名和性别

```
SELECT Sname, Ssex
```

```
FROM Student
```

```
WHERE Sdept= 'CS' OR Sdept= 'MA' OR Sdept= 'IS';
```


3.3 数据查询

• 3.3.1 单表查询

- ORDER BY子句 **DESC,ASC(默认)**

—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
1	201215121	李勇	男	20	CS
2	201215122	刘晨	女	19	CS
3	201215125	张立	男	19	IS
4	201215123	王敏	女	18	MA

3.3 数据查询

• 3.3.1 单表查询

• 聚集函数 **COUNT** **SUM** **AVG** **MAX/MIN**

—查询学生总人数

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

(无列名)	
1	4

—计算1号课程的学生平均成绩

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

(无列名)	
1	92

3.3 数据查询

• 3.3.1 单表查询

• GROUP BY子句

—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	1
2	2	2
3	3	2

—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
1	201215121

聚集函数只能用于SELECT语句和HAVING子句

WHERE子句和HAVING子句的区别：作用对象不同

基本表和
视图

元组

3.3 数据查询

• 3.3.2 连接查询

- 等值连接
- 自身连接
- 外连接
- 多表连接

3.3 数据查询

• 3.3.2 连接查询

• 等值连接

--查询每个学生及其选修课程的情况

```
SELECT  Student.*, SC.*  
FROM    Student, SC  
WHERE   Student.Sno = SC.Sno
```

	Sno	Sname	Ssex	Sage	Sdept	Sno	Cno	Grade
1	201215121	李勇	男	20	CS	201215121	1	92
2	201215121	李勇	男	20	CS	201215121	2	85
3	201215121	李勇	男	20	CS	201215121	3	88
4	201215122	刘晨	女	19	CS	201215122	2	90
5	201215122	刘晨	女	19	CS	201215122	3	80

--查询每个学生及其选修课程的情况的自然连接

```
SELECT  Student.Sno, Sname, Ssex, Sage, Sdept, Cno, Grade  
FROM    Student, SC  
WHERE   Student.Sno = SC.Sno;
```

3.3 数据查询

- 3.3.2 连接查询
 - 等值连接

--查询选修2号课程且成绩在90分以上的所有学生的学号和姓名

```
SELECT Student.Sno, Sname
FROM Student, SC
WHERE Student.Sno=SC.Sno AND SC.Cno='1' AND SC.Grade>90;
```

	Sno	Sname
1	201215121	李勇

3.3 数据查询

• 3.3.2 连接查询

• 自身连接

— 查询每一门课的间接先修课（即先修课的先修课）

```
SELECT  FIRST.Cno, SECOND.Cpno  
FROM    Course FIRST, Course SECOND  
WHERE   FIRST.Cpno = SECOND.Cno;
```

	Cno	Cpno
1	1	7
2	3	5
3	4	NULL
4	5	6
5	7	NULL

3.3 数据查询

• 3.3.2 连接查询

• 外连接

—查询每个学生及其选修课程的情况的左外连接

```
SELECT Student.Sno, Sname, Ssex, Sage, Sdept, Cno, Grade  
FROM Student LEFT JOIN SC ON  
(Student.Sno=SC.Sno);
```

左外连接：LEFT (OUTER) JOIN

右外连接：RIGHT (OUTER) JOIN

全外连接：FULL (OUTER) JOIN

	Sno	Sname	Ssex	Sage	Sdept	Cno	Grade
1	201215121	李勇	男	20	CS	1	92
2	201215121	李勇	男	20	CS	2	85
3	201215121	李勇	男	20	CS	3	88
4	201215122	刘晨	女	19	CS	2	90
5	201215122	刘晨	女	19	CS	3	80
6	201215123	王敏	女	18	MA	NULL	NULL
7	201215125	张立	男	19	IS	NULL	NULL

3.3 数据查询

• 3.3.2 连接查询

• 多表连接

—查询每个学生的学号、姓名、选修的课程名及成绩

```
SELECT Student.Sno, Sname, Cname, Grade
FROM Student, SC, Course
WHERE Student.Sno = SC.Sno AND SC.Cno = Course.Cno;
```

	Sno	Sname	Cname	Grade
1	201215121	李勇	数据库	92
2	201215121	李勇	数学	85
3	201215121	李勇	信息系统	88
4	201215122	刘晨	数学	90
5	201215122	刘晨	信息系统	80

3.3 数据查询

- 3.3.3 嵌套查询
 - 带有IN谓词的子查询
 - 带有比较运算符的子查询
 - 带有ANY或ALL谓词的子查询
 - 带有EXISTS谓词的子查询

SQL语言允许多层嵌套查询

子查询不能使用ORDER BY子句

不相关子查询：子查询的查询条件不依赖于父查询

相关子查询：子查询的查询条件依赖于父查询

3.3 数据查询

• 3.3.3 嵌套查询

• 带有IN谓词的子查询

—查询与刘晨同系的学生学号, 姓名, 系别

```
SELECT Sno, Sname, Sdept
FROM Student
WHERE Sdept IN
    (SELECT Sdept
     FROM Student
     WHERE Sname= '刘晨');
```

	Sno	Sname	Sdept
1	201215121	李勇	CS
2	201215122	刘晨	CS

—查询与刘晨同系的学生学号, 姓名, 系别(自身连接)

```
SELECT S1.Sno, S1.Sname, S1.Sdept
FROM Student S1, Student S2
WHERE S1.Sdept = S2.Sdept AND S2.Sname = '刘晨';
```

3.3 数据查询

• 3.3.3 嵌套查询

• 带有比较运算符的子查询

——找出每个学生超过他选修课程平均成绩的课程号

```
SELECT Sno, Cno
FROM SC x
WHERE Grade >= (SELECT AVG(Grade)
                 FROM SC y
                 WHERE y.Sno=x.Sno);
```

	Sno	Cno
1	201215121	1
2	201215121	3
3	201215122	2

3.3 数据查询

- 3.3.3 嵌套查询
 - 带有ANY或ALL谓词的子查询

一查询非计算机科学系中比计算机科学系任意一个学生年龄小的学生姓名和年龄

```
SELECT Sname, Sage
FROM Student
WHERE Sage < ANY (SELECT Sage
FROM Student
WHERE Sdept= 'CS')
AND Sdept <> 'CS';
```

/*父查询块中的条件 */

	Sname	Sage
1	王敏	18
2	张立	19

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

3.3 数据查询

• 3.3.3 嵌套查询

• 带有EXISTS谓词的子查询

— 查询所有选修了1号课程的学生姓名。

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

	Sname
1	李勇

3.3 数据查询

- 3.3.4 集合查询
 - UNION
 - INTERSECT
 - EXCEPT

3.3 数据查询

• 3.3.5 基于派生表的查询

——找出每个学生超过他自己选修课程平均成绩的课程号

```
SELECT Sno, Cno
FROM SC, (SELECT Sno, Avg(Grade)
          FROM SC
          GROUP BY Sno)
          AS Avg_sc(avg_sno, avg_grade)
WHERE SC.Sno = Avg_sc.avg_sno and SC.Grade >=Avg_sc.avg_grade;
```

	Sno	Cno
1	201215121	1
2	201215121	3
3	201215122	2

数据查询小结

SELECT [ALL|DISTINCT]

<目标列表达式> [别名] [,<目标列表达式> [别名]] ...

FROM <表名或视图名> [别名]

[,<表名或视图名> [别名]] ...

|(<SELECT语句>)[AS]<别名>

[**WHERE** <条件表达式>]

[**GROUP BY** <列名1>[**HAVING**<条件表达式>]]

[**ORDER BY** <列名2> [ASC|DESC]];

- 目标列表达式格式

(1) *

(2) <表名>.*

(3) COUNT([DISTINCT|ALL]*)

(4) [<表名>.]<属性列名表达式>[,<表名>.]<属性列名表达式>]...

其中<属性列名表达式>可以由属性列、作用于属性列的聚集函数和常量的任意算术运算 (+, -, *, /) 组成的运算公式

- 聚集函数的一般格式

COUNT
SUM
AVG
MAX
MIN

([DISTINCT|ALL] <列名>)

• WHERE子句的条件表达式

$\langle \text{属性列名} \rangle \theta \left\{ \begin{array}{l} \langle \text{属性列名} \rangle \\ \langle \text{常量} \rangle \\ [\text{ANY}|\text{ALL}] \text{ (SELECT语句)} \end{array} \right\}$

$\langle \text{属性列名} \rangle [\text{NOT}] \text{BETWEEN} \left\{ \begin{array}{l} \langle \text{属性列名} \rangle \\ \langle \text{常量} \rangle \\ (\text{SELECT语句}) \end{array} \right\} \text{AND} \left\{ \begin{array}{l} \langle \text{属性列名} \rangle \\ \langle \text{常量} \rangle \\ (\text{SELECT语句}) \end{array} \right\}$

• WHERE子句的条件表达式

<属性列名> [NOT] IN { (<值1>[, <值2>] ...) }
(SELECT语句)

<属性列名> [NOT] LIKE <匹配串>

<属性列名> IS [NOT] NULL

[NOT] EXISTS (SELECT语句)