数据库原理及应用

Principle and Application of Database

华东交通大学
喻佳



3.4

关系数据库标准语言SQL

------ 单表查询

SQL语言的数据查询语句：SELECT

一般格式：

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]];

指定要显示的属性列

SQL语言的数据查询语句：SELECT

一般格式：

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]];

指定查询对象(表或视图)

SQL语言的数据查询语句：SELECT

一般格式：

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]];

指定查询条件

SQL语言的数据查询语句：SELECT

一般格式：

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]];

筛选出满足条件的组

对查询结果按指定列的值分组，列值相等的元组为一个组

SQL语言的数据查询语句：SELECT

一般格式：

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]];

对查询结果按指定列值进行排序

ASC升序；DESC降序

当查询仅涉及一个表，则称为**单表查询**。

一、查询表中的若干列

例1：查询全体学生的姓名、学号、所在系。

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

Sname	Sno	Sdept
李勇	201815121	CS
刘晨	201815122	CS
王敏	201815123	MA
张立	201815125	IS

例2：查询全体学生的详细记录。

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

或

```
SELECT  *
FROM  Student;
```

Sno	Sname	Ssex	Sage	Sdept
201815121	李勇	男	20	CS
201815122	刘晨	女	19	CS
201815123	王敏	女	18	MA
201815125	张立	男	19	IS

一、查询表中的若干列

例3：查询选修了课程的学生学号。

```
SELECT  Sno
FROM  SC;
```

Sno
201815121
201815121
201815121
201815122
201815122

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

Sno
201815121
201815122

消除查询结果中重复的元组

二、查询满足条件的元组

通过**WHERE**子句实现

查询条件	运算符
比较	=，>，<，<=，>=，!=
多重条件	AND，OR，NOT
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

二、查询满足条件的元组

1、比较大小

例4：查询所有年龄超过20岁的学生姓名及其年龄。

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

例5：查询考试成绩有不及格的学生的学号。

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

二、查询满足条件的元组

1、比较大小

例4：查询所有年龄超过20岁的学生姓名及其年龄。

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

例5：查询考试成绩有不及格的学生的学号。

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

二、查询满足条件的元组

2、多重条件

例6：查询选修了1号或2号课程并且成绩大于等于90分的学生的学号、课程号和成绩。

学号 Sno	课程号 Cno	成绩 Grade
201815121	1	92
201815121	2	85
201815121	3	88
201815122	2	90
201815122	3	80

```
SELECT  Sno, Cno, Grade
FROM    SC
WHERE  (Cno = '1'  OR  Cno = '2')  AND  Grade >= 90;
```

逻辑运算符的优先级由高到低为：NOT、AND、OR，

用户可以使用括号改变优先级。

二、查询满足条件的元组

3、确定范围

例7：查询年龄在20~23岁（包括20岁和23岁）之间的学生的姓名、所在系和年龄。

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

等价于：Sage >= 20 AND Sage <= 23

二、查询满足条件的元组

4、确定集合

例8：查询选修了1号或2号课程的学生的学号、课程号和成绩。

```
SELECT  Sno, Cno, Grade
FROM    SC
WHERE  Cno  IN  ('1', '2');
```

等价于：Cno = '1' OR Cno = '2'

二、查询满足条件的元组

5、字符匹配(模糊查询)

<列名> [NOT] LIKE ‘<匹配字符串>’ [ESCAPE ‘<换码字符>’]

- 匹配字符串：固定字符串或含通配符（% 或 _ ）的字符串。
- % ：代表任意长度(长度可以为0)的字符串。
- _ ：代表任意的单个字符。

如：对于匹配字符串a%b和a_b来说，

acb，addgb，ab都满足a%b，但只有acb满足a_b

二、查询满足条件的元组

5、字符匹配(模糊查询)

<列名> [NOT] LIKE ‘<匹配字符串>’ [ESCAPE ‘<换码字符>’]

- 匹配字符串：固定字符串或含通配符（% 或 _ ）的字符串。
- % ：代表任意长度(长度可以为0)的字符串。
- _ ：代表任意的单个字符。
- ESCAPE短语：当用户查询的字符串含有通配符% 或 _ 时，需要使用ESCAPE ‘<换码字符>’短语将通配符转为本义。

二、查询满足条件的元组

5、字符匹配(模糊查询)

例9：查询所有姓“刘”的学生的姓名和性别。

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

例10：查询所有不姓“刘”的学生的姓名和性别。

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

二、查询满足条件的元组

5、字符匹配(模糊查询)

例11：查询课程名为“DB_Design”的课程号和学分。

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

例12：查询以DB_开头，且倒数第3个字符为i的课程信息。

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

二、查询满足条件的元组

6、空值查询

通常一个列没有输入值时，该列的值就是空值。

空值不同于零和空格，它不占任何存储空间。

例13：查询没有考试成绩的学生的学号和相应的课程号。

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

切记不可写成： Grade = NULL

谢　　谢

THANK YOU FOR YOUR WATCHING

