

数据库系统课程教案

第8单元

学时：8

教材内容	第3章 SQL语言（3.4.2~3.4.5）
基本知识点	SELECT语句的一般格式，掌握SQL的连接查询、SQL的嵌套查询、SQL的集合查询
教学重点	SQL的连接查询、SQL的嵌套查询、SQL的集合查询
教学难点	SQL的连接查询、SQL的嵌套查询
要求掌握内容	掌握SQL的连接查询、SQL的嵌套查询、SQL的集合查询
教学思路，采用的教学方法和辅助手段，板书设计，重点如何突出，难点如何解决，师生互动等	<p>教学思路：</p> <p>一、复习旧课，巩固上次授课主要内容</p> <ol style="list-style-type: none"> 1、SQL语言的特点是什么？ 2、什么是索引？唯一索引与聚簇索引的区别是什么？ 3、如何利用SQL创建表与索引？SQL的单表查询。 <p>二、导入新课，明确本次授课的目的与要求</p> <ol style="list-style-type: none"> 1、理解SELECT语句的一般格式。 2、掌握SQL的连接查询。 3、掌握SQL的嵌套查询。 4、掌握SQL的集合查询。 <p>三、讲解本次授课的具体内容</p> <p>教学方法：整合教学内容，强化基础训练；努力营造生动活泼的课堂气氛，搭建师生间良好的沟通渠道；采用多媒体教学与传统的板书设计相结合的方式，教学手段灵活多变。</p> <p>辅助手段：通过PPT幻灯片演示结合板书设计和举例加以阐述。</p>
本章思考题和作业	P130 第5(1)~(7)题
主要教材参考资料	<ol style="list-style-type: none"> 1.《数据库系统概论》，萨师煊，王珊，高等教育出版社，2014.9 2.《数据库系统概论学习指导与习题解答》，王珊，张俊，高等教育出版社，2015.7
备 注	

本次授课具体内容

3.4 数据查询

(-)连接查询：同时涉及多个表的查询。用来连接两个表的条件称为连接条件或谓词，一般格式为：
[<表名 1>.]<列名 1> <比较运算符> [<表名 2>.]<列名 2>

其中：连接谓词中的列名称为连接字段，各连接字段类型必须是可比的，但不必是相同的。

1、连接操作的执行过程

(1)嵌套循环法

①首先在表 1 中找到第一个元组，然后从头开始扫描表 2，逐一查找满足连接件的元组，找到后就将表 1 中的第一个元组与该元组拼接起来，形成结果表中一个元组。

②表 2 全部查找完后，再找表 1 中第二个元组，然后再从头开始扫描表 2，逐一查找满足连接条件的元组，找到后就将表 1 中的第二个元组与该元组拼接起来，形成结果表中一个元组。

③重复上述操作，直到表 1 中的全部元组都处理完毕。

(2)排序合并法

①首先按连接属性对表 1 和表 2 排序。

②对表 1 的第一个元组，从头开始扫描表 2，顺序查找满足连接条件的元组，找到后就将表 1 中的第一个元组与该元组拼接起来，形成结果表中一个元组。当遇到表 2 中第一条大于表 1 连接字段值的元组时，对表 2 的查询不再继续。

③找到表 1 的第二条元组，然后从刚才的中断点处继续顺序扫描表 2，查找满足连接条件的元组，找到后就将表 1 中的第一个元组与该元组拼接起来，形成结果表中一个元组。直接遇到表 2 中大于表 1 连接字段值的元组时，对表 2 的查询不再继续。

④重复上述操作，直到表 1 或表 2 中的全部元组都处理完毕为止。

(3)索引连接

①对表 2 按连接字段建立索引。

②对表 1 中的每个元组，依次根据其连接字段值查询表 2 的索引，从中找到满足条件的元组，找到后就将表 1 中的第一个元组与该元组拼接起来，形成结果表中一个元组。

2、SQL 中连接查询的主要类型：广义笛卡尔积查询、等值连接(含自然连接)查询、非等值连接查询、自身连接查询、外连接查询、复合条件连接查询。

(1)广义笛卡尔积(很少使用)：不带连接谓词的连接。

例：SELECT Student.*, SC.* FROM Student, SC;

(2)等值连接：连接运算符为 = 的连接操作。格式：[<表名 1>.]<列名 1> = [<表名 2>.]<列名 2>

说明：任何子句中引用表 1 和表 2 中同名属性时，都必须加表名前缀。引用唯一属性名时可以加也可以省略表名前缀。

[例 32] 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;

(3)自然连接：等值连接的一种特殊情况，把目标列中重复的属性列去掉。

[例 33] 对例 32 用自然连接完成。 SELECT Student.Sno, Sname, Ssex, Sage, Sdept, Cno, Grade
FROM Student, SC WHERE Student.Sno = SC.Sno;

(4)非等值连接查询：连接运算符不是 = 的连接操作。

(5)自身连接：一个表与其自己进行连接。连接需给表起别名以示区别，由于所有属性名都是同名属性，因此须使用别名前缀。

[例 34] 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;
```

(6)外连接：在普通连接操作中只有满足连接条件的元组才能作为结果输出，如例 32 的结果表中没有 95003 和 95004 的学生信息，原因是他们没有选课，在 SC 表中没有相应的元组，但要以 Student 表为连接主体列出每个学生的基本情况及其选课情况，若某个学生没有选课，只输出其基本情况，其选课信息为空，这时就需要使用外连接。在表名后面加外连接操作符(*)或(+)指定非主体表，非主体表有一万能的虚行，该行全部由空值组成，虚行可与主体表中所有不满足连接条件的元组进行连接，由于虚行各列全部是空值，因此与虚行连接的结果中来自非主体表的属性值全部是空值。外连接符出现在连接条件的左边的称为左外连接，外连接符出现在连接条件的右边的称为右外连接。将例 32 改为：

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

(7)复合条件连接：WHERE 子句中含多个连接条件。

[例 35] 查询选修 2 号课程且成绩在 90 分以上所有学生的学号、姓名。

```
SELECT Student.Sno, student.Sname FROM Student, SC
WHERE Student.Sno = SC.Sno AND SC.Cno= '2' AND SC.Grade > 90;
```

(8)多表连接

[例 36] 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;
```

(二)嵌套查询

1、概述：一个 SELECT-FROM-WHERE 语句称为一个查询块，将一个查询块嵌套在另一个查询块的 WHERE 子句或 HAVING 短语的条件中的查询称为嵌套查询。其中外层查询块称为父查询，内层查询块称为子查询，子查询不能使用 ORDER BY 子句。层层嵌套方式反映了 SQL 语言的结构化，有些嵌套查询可以用连接运算来替代。如：SELECT Sname FROM Student 外层查询/父查询

WHERE Sno IN (SELECT Sno FROM SC WHERE Cno= '2'); 内层查询/子查询

2、嵌套查询分类

(1)不相关子查询：子查询的查询条件不依赖于父查询，由里向外逐层处理。即每个子查询在上一级查询处理之前求解，子查询的结果用于建立其父查询的查找条件。

(2)相关子查询：子查询的查询条件依赖于父查询，首先取外层查询中表的第一个元组，根据它与内层查询相关的属性值处理内层查询，若 WHERE 子句返回值为真，则取此元组放入结果表；然后再取外层表的下一个元组。重复这一过程，直至外层表全部检查完为止。

引出子查询的谓词：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 谓词子查询。

(3)带有 IN 谓词的子查询

[例 37]查询与刘晨在同一个系学习的学生。此查询要求可以分以下步骤来完成：

- ①确定刘晨所在系名。SELECT Sdept FROM Student WHERE Sname= '刘晨'；
- ②查找所有 IS 系的学生。SELECT Sno, Sname, Sdept FROM Student WHERE Sdept= 'IS'；
- ③构造嵌套查询：将第一步查询嵌入到第二步查询条件中。

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

此查询为不相关子查询，DBMS 求解该查询时也是分步去做的。

[例 38]查询选修了课程名为信息系统的学生学号和姓名。

```
SELECT Sno, Sname FROM Student WHERE Sno IN (SELECT Sno FROM SC
WHERE Cno IN (SELECT Cno FROM Course WHERE Cname= '信息系统'));
```

③最后在 Student 关系中取出 Sno 和 Sname(4)带有比较运算符的子查询：当能确切知道内层查询返回单值时，可用比较运算符(>, <, =, >=, <=, !=或< >)。

假设一个学生只可能在一个系学习，并且必须属于一个系，则在例 37 可以用 = 代替 IN：

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

子查询一定要跟在比较符之后，一个错误的例子如下：

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

(5)带有 ANY 或 ALL 谓词的子查询：ANY 表示任意一个值，ALL 表示所有值，需配合使用比较运算符。

[例 39]查询其他系中比信息系任意一个(其中某一个)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= 'IS') AND Sdept <> 'IS';
```

[例 40]查询其他系中比 IS 系所有学生年龄都小的学生姓名及年龄。

方法一（用 ALL 谓词）SELECT Sname, Sage FROM Student WHERE Sage < ALL
(SELECT Sage FROM Student WHERE Sdept='IS') AND Sdept<>'IS';

方法二（用集函数）SELECT Sname, Sage FROM Student WHERE Sage<
(SELECT MIN(Sage) FROM Student WHERE Sdept='IS') AND Sdept<>'IS';

(6)带有 EXISTS 谓词的子查询：[NOT]EXISTS 谓词相当于存在量词 \exists 。带有 EXISTS 谓词的子查询不返回任何数据，只产生逻辑 true 或 false，若内层查询结果非空则返回真值，否则返回假值。由 EXISTS 引出的子查询的目标列表达式通常都用*，因为带 EXISTS 的子查询只返回真或假值，给出列名无意义。

[例 41]查询所有选修了 1 号课程的学生姓名。

思路分析：本查询涉及 Student 和 SC 关系，首先在 Student 中依次取每个元组的 Sno 值，用此值去检查 SC 关系，若 SC 中存在这样的元组，其 Sno 值等于此 Student.Sno 值，并且其 Cno= '1'，则取此 Student.Sname 送入结果关系。

用嵌套查询 SELECT Sname FROM Student WHERE EXISTS /*相关子查询*/

(SELECT * FROM SC WHERE Sno=Student.Sno AND Cno= '11');

用连接查询 SELECT Sname FROM Student, SC WHERE Student.Sno=SC.Sno AND SC.Cno= '11';

一些带[NOT]EXISTS谓词的子查询不能被其他形式的子查询等价替换，所有带IN谓词、比较运算符、ANY和ALL谓词的子查询都能用带EXISTS谓词的子查询等价替换。

[例42]查询没有选修1号课程的学生姓名。(用连接运算难于实现)

SELECT Sname FROM Student

WHERE NOT EXISTS (SELECT * FROM SC WHERE Sno = Student.Sno AND Cno='11');

例37 查询与刘晨在同一个系学习的学生，可以用带EXISTS谓词的子查询替换：

SELECT Sno, Sname, Sdept FROM Student S1 WHERE EXISTS

SELECT * FROM Student S2 WHERE S2.Sdept = S1.Sdept AND S2.Sname = '刘晨';

(7)用[NOT]EXISTS实现全称量词(难点)：SQL语言中没有全称量词 \forall ，可以把带有全称量词的谓词转换为等价的带有存在量词的谓词： $(\forall x)P \equiv \neg (\exists x(\neg P))$ 。

[例43]查询选修了全部课程的学生姓名。

SELECT Sname FROM Student WHERE NOT EXISTS

(SELECT * FROM Course WHERE NOT EXISTS

(SELECT * FROM SC WHERE Sno= Student.Sno AND Cno= Course.Cno));

(8)用[NOT]EXISTS实现逻辑蕴涵(难点)：SQL语言中没有蕴涵逻辑运算，可以利用谓词演算将逻辑蕴涵谓词等价转换为： $p \rightarrow q \equiv \neg p \vee q$ 。

[例44]查询至少选修了学生95002选修的全部课程的学生号码。

解题思路：

①逻辑蕴涵表达：查询学号为x的学生，对所有的课程y，只要95002学生选修了课程y，则x也选修了y。

②形式化表示：用P表示谓词“学生95002选修了课程y”，q表示谓词“学生x选修了课程y”，则上述查询为： $(\forall y)p \rightarrow q$ 。

③变换： $(\forall y)p \rightarrow q \equiv \neg (\exists y(\neg (p \rightarrow q))) \equiv \neg (\exists y(\neg (\neg p \vee q))) \equiv \neg \exists y(p \wedge \neg q)$

④变换后语义：不存在这样的课程y，学生95002选修了y，而学生x没有选。

⑤用NOT EXISTS谓词表示：

SELECT DISTINCT Sno FROM SC SCX WHERE NOT EXISTS

(SELECT * FROM SC SCY WHERE SCY.Sno='95002' AND NOT EXISTS

(SELECT * FROM SC SCZ WHERE SCZ.Sno=SCX.Sno AND SCZ.Cno=SCY.Cno));

(二)集合查询：标准SQL直接支持并操作，而一般商用数据库支持并操作、交操作和差操作。

1、并操作(UNION)：参加UNION操作的各结果表的列数必须相同；对应项的数据类型也必须相同。一般格式为：<查询块> UNION <查询块>

[例45]查询计算机系的学生及年龄不大于19岁的学生。

方法一：SELECT * FROM Student WHERE Sdept= 'CS' UNION

SELECT * FROM Student WHERE Sage<=19;

方法二：SELECT DISTINCT * FROM Student WHERE Sdept= 'CS' OR Sage<=19;

[例 46] 查询选修了课程 1 或者选修了课程 2 的学生。

方法一：SELECT Sno FROM SC WHERE Cno='1' UNION

SELECT Sno FROM SC WHERE Cno='2';

方法二：SELECT DISTINCT Sno FROM SC WHERE Cno='1' OR Cno= '2' ;

2、交操作 (INTERSECT)：标准 SQL 中没有提供集合交操作，但可用其他方法间接实现。

[例 47] 查询计算机系且年龄不大于 19 的学生的交集 (即查询计算机系中年龄不大于 19 的学生)。

SELECT * FROM Student WHERE Sdept= 'CS' AND Sage<=19;

[例 48] 查询选修课程 1 与选修课程 2 的学生的交集 (即查询既选修了课程 1 又选修了课程 2 的学生)。

SELECT Sno FROM SC WHERE Cno='1' AND Sno IN (SELECT Sno FROM SC WHERE Cno='2');

3、差操作 (MINUS)：标准 SQL 中没有提供集合差操作，但可用其他方法间接实现。

[例 49] 查询计算机科学系的学生与年龄不大于 19 岁的学生的差集 (实际上就是查询计算机科学系中年龄大于 19 岁的学生)。

SELECT * FROM Student WHERE Sdept= 'CS' AND Sage>19; 对集合操作结果的排序：ORDER BY 子句用数字指定排序属性，只用于对最终查询结果排序，不能对中间结果排序。任何情况下，ORDER BY 子句只能出现在最后。

如：错误写法 SELECT * FROM Student WHERE Sdept= 'CS' ORDER BY Sno UNION
SELECT * FROM Student WHERE Sage<=19 ORDER BY Sno;

正确写法 SELECT * FROM Student WHERE Sdept= 'CS' UNION
SELECT * FROM Student WHERE Sage<=19 ORDER BY 1;

(四) SELECT 语句的一般格式

SELECT [ALL|DISTINCT] <目标列表表达式> [别名] [, <目标列表表达式> [别名]] ... FROM <表名或视图名> [别名] [, <表名或视图名> [别名]] ... [WHERE <条件表达式>]

[GROUP BY <列名 1> [, <列名 2>] ...

[HAVING <条件表达式>]]

[ORDER BY <列名 1> [ASC|DESC] [, <列名 2> [ASC|DESC]]]

本次授课小结

本次授课讲述了 SELECT 语句的一般格式、SQL 的连接查询、SQL 的嵌套查询、SQL 的集合查询。学生课后复习时应着重于其中的第 2、3、4 点内容，为进一步学习后续章节打好基础。

实验

实验 2 表的创建与管理