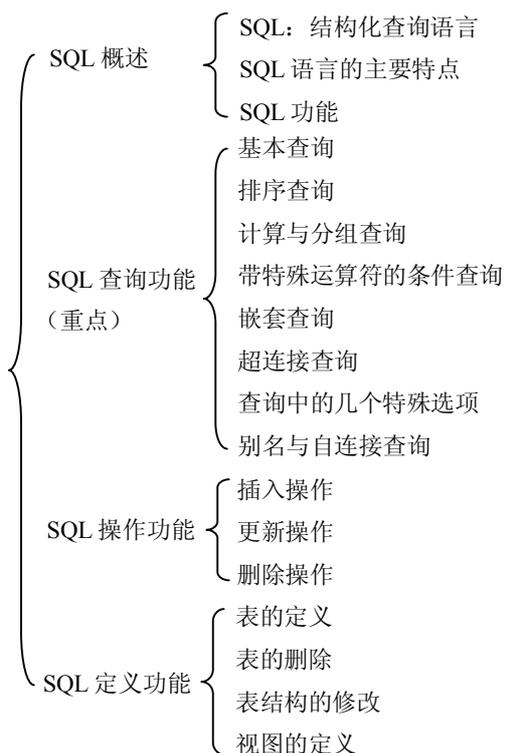


第 4 章 结构化查询语言 SQL

知识结构图



SQL (Structured Query Language) 是结构化查询语言。“查询”是 SQL 语言的重要组成部分，但不是全部内容，SQL 语言还包含数据定义、数据操作、数据控制功能。

SQL 是由美国国家标准协会 (American National Standards Institute, ANSI) 于 1986 年 10 月公布的，随后 SQL 就被当作关系数据库的工业标准语言。1987 年，这个标准被国际标准化组织 (International Standards Organization, ISO) 批准为国际标准，并在此基础上进行了补充。

SQL 语言成为国际标准后，各数据库公司纷纷推出了自己符合 SQL 标准的数据库管理系统 (DBMS)，如 Oracle、Microsoft SQL Server、Informix、Sybase 等。SQL 标准可使不同关系数据库之间实现一致性和可移植性。Visual FoxPro 作为一种可视化的关系数据语言，也使用 SQL 语言作为数据库定义、数据操作和数据检索的工具。

SQL 语言的主要特点如下：

(1) SQL 是一种一体化的语言。它包括了数据定义、数据查询、数据操作和数据控制等方面的功能，它可以完成数据库活动中的全部工作。以前的非关系模型是数据语言，一般包括存储模式描述语言、概念模式描述语言、外部模式描述语言等，这种模型的数据语言，一是内容多，二是掌握和使用起来不像 SQL 那样简单、实用。

(2) SQL 语言是一种高度非过程化的语言。它没有必要一步步地告诉计算机“如何”去做，而只需要描述清楚用户要“做什么”，SQL 语言可将要求交给系统，自动完成全部工作。

(3) SQL 语言非常简洁。虽然 SQL 语言功能很强，但它只有为数不多的几条命令，分别用于完成数据定义、数据操作和数据查询等功能，如表 4-1 所示。另外，SQL 的语法也非常简单，它很接近英语自然语言，因此容易学习、掌握。

表 4-1 SQL 语言的 9 个命令动词

SQL 功能		命令动词
数据查询		Select
数据操作	插入记录	Insert
	更新记录	Update
	删除记录	Delete
数据定义	定义基本表或索引	Create
	删除基本表或索引	Drop
	修改表结构	Alter
数据控制	授权	Grant
	收回权限	Revoke

(4) SQL 语言可以直接以命令方式交互使用，也可以嵌入到程序设计语言中以程序方式使用。现在很多数据库应用开发工具都将 SQL 语言直接融入到自身的语言之中，使用起来更方便，Visual FoxPro 就是如此。这些使用方式为用户提供了灵活的选择余地。此外，尽管 SQL 的使用方式不同，但 SQL 语言的语法基本是一致的。

SQL 语言的功能很强大，但由于 Visual FoxPro 自身安全控制方面的缺陷，所以它没有提供数据控制功能。在 Visual FoxPro 中只支持数据定义、数据查询和数据操作功能。

4.1 数据查询

SQL 语言的核心是查询。SQL 的查询命令的基本形式由 Select-From-Where 查询块组成，多个查询块可以嵌套执行。查询是根据需要，以可读的方式从数据库中提取数据。

SQL SELECT 查询命令的格式如下：

```
SELECT [ALL | DISTINCT][TOP <表达式> [PERCENT]][<别名>.<列表表达式>[AS <别名>];
[<别名>.<列表表达式>[AS <别名>]...];
FROM [<数据库名!><表名>[,<数据库名!><表名>...];
[INNER | LEFT | RIGHT | FULL JOIN [<数据库名!><表名>];
[ON <连接条件>...];
[[INTO TABLE <新表名>] | [TO FILE <文件名> | TO PRINTER | TO SCREEN]];
[WHERE <连接条件>[AND <连接条件>...];
[AND | OR <查询条件> [AND | OR<查询条件>...]];
[GROUP BY <字段名>[,<字段名>...][HAVING <分组条件>];
[ORDER BY <字段名>[ASC | DESC][,<字段名>[ASC | DESC]...]]
```

SQL SELECT 命令看上去非常复杂，但常用的只有 6 个子句：SELECT、FROM、WHERE、GROUP BY、HAVING、ORDER BY。

(1) SELECT 短语：用于说明要查询的数据。

(2) FROM 短语：用于说明要查询的数据来自哪个或哪些表，可对单个表或多个表进行查询。

(3) WHERE 短语：用于说明查询条件；如果是多表查询还可通过该子句指明表与表之间的连接条件进行连接。

(4) GROUP BY 短语：用于对查询进行分组，可利用它进行分组汇总。

(5) HAVING 短语：跟随在 GROUP BY 之后使用，它是用来限定分组，要求只有满足分组条件的记录才能进行分组查询。

(6) ORDER BY 短语：用于对查询的结果进行排序。

(7) AS 短语：用于指定查询结果中字段的名称，AS 短语前的表达式可以是一个字段名、表达式、函数等。

(8) DISTINCT 短语：用于说明在查询结果中去掉重复值。

(9) INTO TABLE 短语：用于指明查询结果保存在哪个表中。

在以后的学习中，常用的 SQL SELECT 语句的具体格式如下：

SELECT [ALL | DISTINCT]<检索项>;

FROM <表名列表>;

[WHERE <查询条件>[AND<连接条件>]];

[GROUP BY <字段列表>[HAVING<分组条件>]];

[ORDER BY <字段名>[ASC | DESC]]

[INTO TABLE <新表名>] [TO FILE <文件名>]

说明：

1) 在 SQL SELECT 查询语句中，SELECT...FROM...两个命令动词是最基本的并且是不可缺少的，其他命令动词可以根据需要选用。

2) SQL SELCET 查询语句的功能是将满足要求的数据查询出来，它不会更改数据库中的数据。

SELECT 查询命令的使用非常灵活，用它可以构造出各种各样的查询。本节将通过大量的实例来介绍 SELECT 查询命令的使用，通过实例具体学习各个短语的含义。

为了方便读者对照和验证操作结果，下面给出各个表的具体记录值，如图 4-1 至图 4-4 所示。

学号	姓名	性别	出生日期	民族	政治面貌	所在院系	专业名称	语种	导师姓名	家庭住址	学籍变动	照片
42005001	韩春杰	男	10/15/83	汉族	T	文学院	汉语言文学	英语	连秀君	黑龙江省齐齐哈尔市克山县	memo	Gen
42005002	李作宇	女	05/14/80	汉族	F	文学院	中国古代文学	英语	张建	黑龙江省牡丹江市宁安市	memo	Gen
42005003	张华	男	02/11/84	满族	F	文学院	中国古代文学	俄语	崔明材	哈尔滨市	memo	Gen
42005004	张莉	女	06/09/82	汉族	T	文学院	中国现当代文学	英语	孙学成	山东省聊城市	memo	gen
42005005	王春秀	女	10/13/82	汉族	F	文学院	中国现当代文学	英语	战爱成	湖南省长沙市	memo	gen
42005006	宋志新	男	02/15/82	达斡尔族	F	文学院	中国现当代文学	日语	陈志伟	黑龙江省佳木斯市	memo	gen
42005007	陈敏	女	01/08/77	汉族	F	文学院	中国现当代文学	英语	战爱成	黑龙江省尚志市	memo	gen
42005008	张辉	女	08/30/82	汉族	T	英语学院	英语语言文学	二外俄语	周世明	黑龙江省盘锦市	memo	gen
42005009	国美阳	女	06/15/75	回族	F	英语学院	英语语言文学	二外俄语	李爽	山东省菏泽市	memo	gen
42005010	李云鹤	男	07/15/82	汉族	F	英语学院	英语语言文学	二外俄语	周世明	黑龙江省佳木斯市	memo	gen
42005011	周娜	女	05/27/81	汉族	F	英语学院	英语语言文学	二外法语	程雅平	哈尔滨市	memo	gen
42005012	赵妍名	女	09/24/82	汉族	F	艺术学院	音乐学	俄语	苗辰光	黑龙江省鸡西市	memo	gen
42005013	刘斌男	男	09/05/82	满族	T	艺术学院	音乐学	英语	王军	黑龙江省鸡西市	memo	gen
42005014	陈庆海	男	08/10/81	汉族	F	艺术学院	美术学	英语	白强	辽宁省鞍山市	memo	gen
42005015	王春宇	男	08/08/78	汉族	F	艺术学院	美术学	英语	白强	哈尔滨市	memo	gen
42005016	李梅	女	02/13/81	汉族	F	数学学院	基础数学	英语	张文斌	云南省曲靖市	memo	gen
42005017	李帅	男	05/14/79	汉族	F	数学学院	基础数学	英语	赵正	吉林省长春市	memo	gen
42005018	卢莹	女	08/23/82	汉族	T	数学学院	应用数学	俄语	王锦	吉林省长春市	memo	gen
42005019	孙楠	女	07/14/82	汉族	F	计算机学院	计算机应用技术	英语	刘绍梅	黑龙江省大庆市	memo	gen
42005020	迟丽凤	女	06/01/81	汉族	F	计算机学院	计算机应用技术	英语	刘绍梅	黑龙江省五常市	memo	gen

图 4-1 学生 (student) 表

课程编号	课程名称	开课学院	课程类别	学时	学分	开课学期	教师姓名	教师编号	考核方式
10002101	外国语	NULL	学位公共课	120	4	1	NULL	NULL	闭卷
10002102	马克思主义理论	NULL	学位公共课	60	2	1	NULL	NULL	开卷
10002103	文艺理论基础问题	文学院	学位基础课	56	2	1	陈志伟	10230006	论文
10002104	文艺美学	文学院	专业课	56	2	2	张建	10230002	论文
10002105	文化研究	文学院	专业课	56	2	3	连秀君	10230001	论文
10002106	中国古代美学经典文献	文学院	专业课	56	2	2	战斐成	10230005	闭卷
10002107	大众文化研究	文学院	选修课	36	1	4	崔明材	10230003	闭卷
10002108	周易美学研究	文学院	选修课	36	1	5	崔明材	10230003	开卷
10002109	英语语言学	英语学院	学位基础课	52	4	1	李爽	10230008	闭卷
10002110	英语语言史	英语学院	学位基础课	52	4	1	周世明	10230007	闭卷
10002111	美国小说	英语学院	专业课	52	3	2	周世明	10230007	闭卷
10002112	英美文论选读	英语学院	专业课	52	3	3	程雅平	10230009	论文
10002113	英汉对比语言学	英语学院	选修课	36	2	4	王宏旭	10230017	开卷
10002114	翻译学	英语学院	选修课	36	2	4	成双	10230018	论文
10002115	音乐学概论	艺术学院	学位基础课	60	2	2	王军	10230011	闭卷
10002116	美术学概论	艺术学院	学位基础课	60	2	2	白强	10230012	闭卷
10002117	作品分析	艺术学院	专业课	60	2	6	苗辰光	10230010	闭卷
10002118	西方现代美术史	艺术学院	专业课	60	2	5	林然	10230019	闭卷
10002119	代数学	数学学院	学位基础课	72	3	1	张文斌	10230013	闭卷
10002120	泛函分析	数学学院	学位基础课	72	3	1	赵正	10230014	闭卷
10002121	最优化方法	数学学院	专业课	72	3	4	王锦	10230015	闭卷
10002122	非线性控制	数学学院	专业课	72	3	6	宋军	10230020	开卷
10002123	线性偏微分方程	数学学院	选修课	60	2	5	宋军	10230020	论文
10002124	面向对象软件工程	计算机学院	学位基础课	48	3	1	刘绍梅	10230016	闭卷
10002125	数据通信与计算机网络	计算机学院	学位基础课	48	3	1	王立军	10230022	闭卷
10002126	软件体系结构	计算机学院	专业课	36	2	2	王立军	10230022	闭卷

图 4-2 课程 (course) 表

课程编号	课程名称	学号	姓名	成绩
10002101	外国语	42005004	张莉	0
10002101	外国语	42005005	王春秀	45
10002101	外国语	42005006	宋志新	90
10002101	外国语	42005007	陈敏	77
10002101	外国语	42005012	赵研名	89
10002101	外国语	42005013	刘晴男	50
10002101	外国语	42005014	陈庆海	0
10002101	外国语	42005015	王春宇	68
10002101	外国语	42005016	李梅	84
10002101	外国语	42005017	李帅	42
10002101	外国语	42005018	卢莹	96
10002101	外国语	42005019	孙楠	75
10002101	外国语	42005020	迟丽凤	91
10002103	文艺理论基础研究	42005001	韩春杰	78
10002103	文艺理论基础研究	42005002	李作宇	65
10002103	文艺理论基础研究	42005003	张华	96
10002103	文艺理论基础研究	42005004	张莉	0
10002103	文艺理论基础研究	42005005	王春秀	89
10002103	文艺理论基础研究	42005006	宋志新	56
10002103	文艺理论基础研究	42005007	陈敏	78
10002107	大众文化研究	42005001	韩春杰	66
10002107	大众文化研究	42005003	张华	75
10002107	大众文化研究	42005007	陈敏	70
10002109	英语语言学	42005008	张辉	74
10002109	英语语言学	42005009	国贵阳	86
10002109	英语语言学	42005010	李云鹤	68
10002109	英语语言学	42005011	商娜	78
10002113	英汉对比语言学	42005007	陈敏	83
10002113	英汉对比语言学	42005009	国贵阳	88
10002115	音乐学概论	42005012	赵研名	75
10002115	音乐学概论	42005013	刘晴男	78
10002120	泛函分析	42005016	李梅	65
10002120	泛函分析	42005017	李帅	95
10002120	泛函分析	42005018	卢莹	84
10002125	面向对象软件工程	42005019	孙楠	73
10002125	面向对象软件工程	42005020	迟丽凤	66

图 4-3 成绩 (score) 表

教师编号	教师姓名	性别	出生日期	所在院系	研究方向	职称	学位	学历
10230001	连秀君	女	05/25/71	文学院	汉语语法学	教授	博士	研究生
10230002	张建	男	10/01/68	文学院	中国古代美学	副教授	硕士	研究生
10230003	崔明材	男	01/05/75	文学院	唐宋文学	副教授	硕士	研究生
10230004	孙学成	男	04/30/72	文学院	中国现代文学	教授	硕士	研究生
10230005	战斐成	男	11/12/62	文学院	语言应用于写作	副教授	硕士	研究生
10230006	陈志伟	男	11/02/71	文学院	中国现代文学	副教授	硕士	研究生
10230007	周世明	男	06/05/73	英语学院	美国文学	教授	硕士	研究生
10230008	李爽	女	11/06/74	英语学院	美国文化	教授	硕士	研究生
10230009	程雅平	女	08/07/68	英语学院	语用学	副教授	博士	研究生
10230010	苗辰光	男	12/24/67	艺术学院	现代声乐的演唱与研究	副教授	硕士	研究生
10230011	王军	男	02/28/52	艺术学院	声乐表演艺术	教授	学士	本科
10230012	白强	男	01/15/72	艺术学院	西方当代美术研究	副教授	硕士	研究生
10230013	张文斌	男	12/07/72	数学学院	泛函分析	副教授	硕士	研究生
10230014	赵正	男	07/28/57	数学学院	再生核空间理解	教授	学士	本科
10230015	王锦	女	05/20/63	数学学院	泛函分析	副教授	博士	研究生
10230016	刘绍梅	女	07/14/64	计算机学院	数据库	教授	硕士	研究生
10230017	王宏旭	男	04/10/77	英语学院	英国文学	讲师	硕士	研究生
10230018	成双	女	09/08/79	英语学院	美国文学	助教	博士	研究生
10230019	林然	女	09/11/75	艺术学院	媒体艺术设计	助教	博士	研究生
10230020	宋军	男	10/16/66	数学学院	逼近论	讲师	博士	研究生
10230021	王立军	男	03/26/78	计算机学院	计算机网络	讲师	硕士	研究生

图 4-4 教师 (teacher) 表

4.1.1 基本查询

学习 SQL SELECT 查询语句, 首先从几个最简单的查询开始, 这些查询可以基于单个表或者基于多个表, 可以有简单的查询条件和连接条件。

1. 单表查询

单表查询是基于一个表的所有字段的查询, 可以由 SELECT...FROM...[WHERE...]子句构成。

(1) 单表无条件查询。

格式: SELECT [DISTINCT]<检索项>;

FROM <表名>

例 4.1 从 student 表中检索出学生的学号、姓名、性别、出生日期、所在院系。

```
SELECT 学号,姓名,性别,出生日期,所在院系 FROM student
```

查询结果如图 4-5 所示。

例 4.2 查询 teacher 表中的所有信息。

方法一:

```
SELECT 教师编号,教师姓名,性别,出生日期,所在院系,研究方向,职称,学位,学历;
FROM teacher
```

方法二:

```
SELECT * FROM teacher
```

两种方法查询的结果一致,如图 4-6 所示。

学号	姓名	性别	出生日期	所在院系
42005001	韩春杰	男	10/15/83	文学院
42005002	李作宁	女	05/14/80	文学院
42005003	张华	男	02/11/84	文学院
42005004	张新	女	06/09/82	文学院
42005005	王春秀	女	10/13/82	文学院
42005006	宋志新	男	02/15/82	文学院
42005007	陈敏	女	01/08/77	文学院
42005008	张辉	女	08/30/82	英语学院
42005009	国美阳	女	06/15/75	英语学院
42005010	李云鹤	男	07/15/82	英语学院
42005011	商娜	女	05/27/81	英语学院
42005012	赵妍名	女	09/24/82	艺术学院
42005013	刘婧男	女	09/05/82	艺术学院
42005014	陈庆海	男	08/10/81	艺术学院
42005015	王春宇	男	08/08/78	艺术学院
42005016	李梅	女	02/13/81	数学学院
42005017	李帅	男	05/14/79	数学学院
42005018	卢莹	女	08/23/82	数学学院
42005019	孙楠	女	07/14/82	计算机学院
42005020	迟丽凤	女	06/01/81	计算机学院

图 4-5 例 4.1 的查询结果

教师编号	教师姓名	性别	出生日期	所在院系	研究方向	职称	学位	学历
10230001	连秀君	女	05/25/71	文学院	汉语语法学	教授	博士	研究生
10230002	张建	男	10/01/68	文学院	中国古代美学	副教授	硕士	研究生
10230003	崔明材	男	01/05/75	文学院	唐宋文学	副教授	硕士	研究生
10230004	孙学成	男	04/30/72	文学院	中国现代文学	教授	硕士	研究生
10230005	战爱成	男	11/12/62	文学院	语言应用于写作	副教授	硕士	研究生
10230006	陈志伟	男	11/02/71	文学院	中国现代文学	副教授	硕士	研究生
10230007	周世明	男	06/05/73	英语学院	美国文学	教授	硕士	研究生
10230008	李爽	女	11/06/74	英语学院	美国文化	教授	硕士	研究生
10230009	程维平	女	06/07/68	英语学院	语言学	博士	研究生	研究生
10230010	崔晓光	男	12/24/67	艺术学院	现代声乐的演唱与研究	副教授	硕士	研究生
10230011	王军	男	02/28/52	艺术学院	声乐表演艺术	教授	学士	本科
10230012	白强	男	01/15/72	艺术学院	西方当代美术研究	副教授	硕士	研究生
10230013	张文斌	男	12/07/72	数学学院	泛函分析	副教授	硕士	研究生
10230014	赵正	男	07/28/57	数学学院	再生核空间理解	教授	学士	本科
10230015	王锦	女	05/20/63	数学学院	泛函分析	副教授	博士	研究生
10230016	刘淑梅	女	07/14/64	计算机学院	数据库	教授	硕士	研究生
10230017	王宏旭	男	04/10/77	英语学院	英国文学	讲师	硕士	研究生
10230018	成双	女	09/08/79	英语学院	美国文学	助教	博士	研究生
10230019	林然	女	09/11/75	艺术学院	媒体艺术设计	助教	博士	研究生
10230020	宋军	男	10/16/66	数学学院	逼近论	讲师	博士	研究生
10230021	王立军	男	03/26/78	计算机学院	计算机网络	讲师	硕士	研究生

图 4-6 例 4.2 的查询结果

说明:

- 1) SELECT 短语后面接检索项,即具体要查询的内容。
- 2) FROM 短语指明检索项来自的表,对于 SELECT 查询语句来说, FROM 子句是必需的,是不可以省略的。
- 3) 当查询内容为表中所有信息时,可以用通配符“*”,表示输出表中所有字段的值。

例 4.3 将 student 表中的专业名称信息检索出来。

```
SELECT 专业名称 FROM student
```

查询结果如图 4-7 所示。

例 4.4 将 student 表中的专业名称信息检索出来(去掉重复值)。

```
SELECT DISTINCT 专业名称 FROM student
```

查询结果如图 4-8 所示。

专业名称
汉语言文学
中国古代文学
中国古代文学
中国现代文学
中国现代文学
中国现代文学
英语语言文学
英语语言文学
英语语言文学
英语语言文学
音乐学
音乐学
美术学
美术学
基础数学
基础数学
应用数学
计算机应用技术
计算机应用技术

图 4-7 例 4.3 的查询结果

专业名称
汉语言文学
基础数学
计算机应用技术
美术学
音乐学
应用数学
英语语言文学
中国古代文学
中国现代文学

图 4-8 例 4.4 的查询结果

说明: Distinct 短语的作用是去掉查询结果中的重复值。注意 Distinct 短语的位置位于需要去掉重复值的字段名前面。

(2) 单表有条件查询。

格式: SELECT [DISTINCT] <检索项> FROM <表名> WHERE <查询条件>

例 4.5 查询女教师的教师姓名、性别、出生日期、所在院系、研究方向、职称等信息。

```
SELECT 教师姓名, 性别, 出生日期, 所在院系, 研究方向, 职称;
FROM teacher WHERE 性别="女"
```

查询结果如图 4-9 所示。

说明: 当题目中有限定条件时, 需要用 WHERE 短语来指定查询条件。WHERE 短语位置应位于 FROM 短语后面。

例 4.6 从 score 表中检索出选修了课程编号为“10002101”的, 并且成绩不及格的课程编号, 课程名称, 学号, 姓名和成绩等信息。

```
方法一: SELECT 课程编号, 课程名称, 学号, 姓名, 成绩 FROM score;
        WHERE 课程编号="10002101" AND 成绩<60
```

```
方法二: SELECT * FROM score;
        WHERE 课程编号="10002101" AND 成绩<60
```

查询结果如图 4-10 所示。

教师姓名	性别	出生日期	所在院系	研究方向	职称
汪奕君	女	05/25/71	文学院	汉语语法学	教授
李爽	女	11/06/74	英语学院	美国文化	教授
程雅平	女	08/07/68	英语学院	语用学	副教授
王婧	女	05/20/63	数学学院	泛函分析	副教授
刘铭梅	女	07/14/64	计算机学院	数据库	教授
成双	女	09/08/79	英语学院	美国文学	助教
林然	女	09/11/75	艺术学院	媒体艺术设计	助教

图 4-9 例 4.5 的查询结果

课程编号	课程名称	学号	姓名	成绩
10002101	外国语	42005004	张莉	0
10002101	外国语	42005005	王春秀	45
10002101	外国语	42005013	刘耀男	50
10002101	外国语	42005014	陈庆海	0
10002101	外国语	42005017	李帅	42

图 4-10 例 4.6 的查询结果

例 4.7 检索出选修了课程编号为“10002103”或“10002115”的, 并且成绩大于 70 分的课程编号、课程名称、姓名、成绩等信息。

```
SELECT 课程编号, 课程名称, 姓名, 成绩 FROM score;
WHERE (课程编号="10002103" OR 课程编号="10002115")
AND 成绩>70
```

查询结果如图 4-11 所示。

说明:

1) 在查询中查询条件为两个或两个以上时, 应用逻辑运算符 AND 或 OR 将多个条件连接起来。

- AND 表示相与的关系, 当 AND 左右两边的条件都成立时, 表达式结果才为真。
- OR 表示相或的关系, 当 OR 左右两边的条件有一个成立时, 表达式的结果就为真。

2) 在 Visual FoxPro 的命令窗口中, 一行是一条命令语句, 如果一条命令语句太长放在一行上不方便, 可以分成多行显示, 但是需要在分行处加上续行符“;”用来表示续行。

例 4.8 从 student 表中检索出在 1980 年 12 月 31 日之后出生的所有男同学的学号、姓名、性别、年龄、所在院系、专业名称、家庭住址等信息。

```
SELECT 学号, 姓名, 性别, YEAR( DATE() ) - YEAR( 出生日期 ) AS 年龄, 所在院系;
专业名称, 家庭住址 FROM student;
```

```
WHERE 出生日期 > { ^1980-12-31 } AND 性别="男"
```

查询结果如图 4-12 所示。

课程编号	课程名称	姓名	成绩
10002103	文艺理论基础研究	韩春杰	78
10002103	文艺理论基础研究	张华	96
10002103	文艺理论基础研究	王春秀	88
10002103	文艺理论基础研究	陈敏	78
10002115	音乐学概论	赵翔名	75
10002115	音乐学概论	刘耀男	78

图 4-11 例 4.7 的查询结果

说明:

- 1) YEAR(DATE())-YEAR(出生日期) 表达式: 用于计算学生的年龄。
- 2) 在表达式后可以用 AS 短语为表达式命别名。

例如, YEAR(DATE())-YEAR(出生日期) AS 年龄, 则在查询结果中会出现一个新的字段名“年龄”。

3) 对于“出生日期”字段是日期型的数据, 所以在书写表达式时要注意类型匹配与类型格式, 如出生日期<{^1980-01-01}>。

学号	姓名	性别	年龄	所在院系	专业名称	家庭住址
42005001	韩春杰	男	26	文学院	汉语言文学	黑龙江省齐齐哈尔市克山县
42005003	张华	男	25	文学院	中国古代文学	哈尔滨市
42005006	宋志新	男	27	文学院	中国现当代文学	黑龙江省佳木斯市
42005010	李云鹤	男	27	英语学院	英语语言文学	黑龙江省佳木斯市
42005014	陈庆海	男	28	艺术学院	美术学	辽宁省鞍山市

图 4-12 例 4.8 的查询结果

2. 多表连接查询

连接是关系的基本操作之一, 当查询涉及多个表时, 就要用到多表的连接查询。连接查询基于多个关系的查询。下面通过几个简单的连接查询实例来学习多表连接查询。

(1) 多表无条件的连接查询。

格式: SELECT <检索项> FROM <表名> WHERE <连接条件>

例 4.9 检索出学生的学号、姓名、性别、选修课程名称、所在院系和成绩等信息。

```
SELECT student.学号, student.姓名, 性别, 课程名称, 所在院系, 成绩;
FROM student, score;
WHERE student.学号=score.学号
```

查询结果如图 4-13 所示。

学号	姓名	性别	课程名称	所在院系	成绩
42005004	张莉	女	外国语	文学院	0
42005005	王春秀	女	外国语	文学院	45
42005006	宋志新	男	外国语	文学院	90
42005007	陈敏	女	外国语	文学院	77
42005012	赵妍名	女	外国语	艺术学院	89
42005013	刘婧	女	外国语	艺术学院	50
42005014	陈庆海	男	外国语	艺术学院	0
42005015	王春宇	男	外国语	艺术学院	68
42005016	李梅	女	外国语	数学学院	84
42005017	李帅	男	外国语	数学学院	42
42005018	卢莹	女	外国语	数学学院	96

图 4-13 例 4.9 的部分查询结果

说明:

1) FROM 短语后面接多个表名。当查询涉及多个表时, 将所有的表名都放在 FROM 短语后面, 并且两个表之间用逗号隔开。

2) 这里“student.学号=score.学号”是连接条件。两个表能够进行连接查询的前提是两个表一定要有相同的字段名。当 FROM 短语之后的多个表中含有相同的字段名时, 必须用表名(关系名)前缀直接指明字段名, 格式如下:

表名 1.字段名=表名 2.字段名

3) SELECT 短语中的检索项如果不唯一, 需要指明该检索项来自哪个表。例如, 在例 4.9 中检索项学号在 student 表和 score 表中都出现了, 所以要指明该学号到底来自哪个表。

例 4.10 检索选修课的课程编号、课程名称、学分和考核方式。

```
SELECT DISTINCT score.课程编号, score.课程名称, 学分, 考核方式;
FROM score, course;
WHERE course.课程编号=score.课程编号
```

查询结果如图 4-14 所示。

(2) 多表有条件连接查询。

格式: SELECT <检索项> FROM <表名> WHERE <连接条件> AND <查询条件>

例 4.11 检索出所有男同学的并且成绩大于 70 的学号、姓名、性别、课程名称和成绩信息。

```
SELECT student.学号, student.姓名, 性别, 课程名称, 成绩;
FROM student, score;
WHERE student.学号=score.学号 AND (性别="男" AND 成绩>70)
```

查询结果如图 4-15 所示。



课程编号	课程名称	学分	考核方式
10002101	外语	4	闭卷
10002103	文艺理论基础研究	2	论文
10002107	大众文化研究	1	闭卷
10002109	英语文体学	4	闭卷
10002113	英汉对比语言学	2	开卷
10002115	音乐学概论	2	闭卷
10002120	泛函分析	3	闭卷
10002125	面向对象软件工程	3	闭卷

图 4-14 例 4.10 的查询结果



学号	姓名	性别	课程名称	成绩
42005006	宋志新	男	外语	90
42005001	韩春杰	男	文艺理论基础研究	78
42005003	张华	男	文艺理论基础研究	96
42005003	张华	男	大众文化研究	75
42005017	李坤	男	泛函分析	95

图 4-15 例 4.11 的查询结果

说明:

1) 连接条件与查询条件同时放在 WHERE 短语中, “性别=“男” and 成绩>70”为查询条件, “student.学号=score.学号”为连接条件, 两个条件用 AND 连接。

2) 连接条件与查询条件无先后顺序。

例 4.12 检索所有女同学的学号、姓名、课程名称、所在院系、导师姓名、研究方向和职称等信息。

```
SELECT student.学号, student.姓名, score.课程名称, student.所在院系;
导师姓名, 研究方向, 职称;
FROM student, course, score, teacher;
WHERE student.学号=score.学号 AND;
course.课程编号=score.课程编号 AND;
course.教师编号=teacher.教师编号 AND;
student.性别="女"
```

查询结果如图 4-16 所示。



学号	姓名	课程名称	所在院系	导师姓名	研究方向	职称
42005002	李作宇	文艺理论基础研究	文学院	张建	中国现代文学	副教授
42005004	张莉	文艺理论基础研究	文学院	孙学成	中国现代文学	副教授
42005005	王春秀	文艺理论基础研究	文学院	战竞成	中国现代文学	副教授
42005007	陈敏	英汉对比语言学	文学院	战竞成	英国文学	讲师
42005007	陈敏	大众文化研究	文学院	战竞成	唐宋文学	副教授
42005007	陈敏	文艺理论基础研究	文学院	战竞成	中国现代文学	副教授
42005008	张辉	英语文体学	英语学院	周世明	美国文化	教授
42005009	国美阳	英语文体学	英语学院	李爽	美国文化	教授
42005009	国美阳	英汉对比语言学	英语学院	李爽	英国文学	讲师
42005011	商娜	英语文体学	英语学院	程雅平	美国文化	教授
42005012	赵研名	音乐学概论	艺术学院	苗辰光	声乐表演艺术	教授
42005013	刘辉男	音乐学概论	艺术学院	王军	声乐表演艺术	教授
42005016	李梅	泛函分析	数学学院	张文斌	再生核空间理解	教授
42005018	卢莹	泛函分析	数学学院	王锦	再生核空间理解	教授

图 4-16 例 4.12 的查询结果

4.1.2 排序查询

利用 SQL SELECT 查询语句进行查询时可以将查询的结果进行排序, 排序的短语为 ORDER BY。

格式: SELECT <检索项> FROM <表名> [WHERE <连接条件> AND <查询条件>];
ORDER BY <字段名> [ASC | DESC] [, <字段名> [ASC | DESC]...]

例 4.13 检索出所有成绩大于 80 (含 80) 分的学生成绩信息, 查询结果按成绩降序排列。

```
SELECT * FROM score WHERE 成绩>=80 ORDER BY 成绩 DESC
```

查询结果如图 4-17 所示。

说明: ASC 表示升序, DESC 表示降序, 默认的情况表示升序。

例 4.14 检索课程表中的课程编号、课程名称、课程类别、学时、学分, 按学分的升序排序, 如果学分相同则按学时降序进行排序。

```
SELECT 课程编号, 课程名称, 课程类别, 学时, 学分 FROM course;
```

```
ORDER BY 学分 ASC, 学时 DESC
```

查询结果如图 4-18 所示。



课程编号	课程名称	学号	姓名	成绩
10002101	外语	42005018	卢莹	96
10002103	文艺理论基础研究	42005003	张华	96
10002120	泛函分析	42005017	李帅	95
10002101	外语	42005020	迟丽凤	91
10002101	外语	42005006	宋志新	90
10002101	外语	42005012	赵朋名	89
10002103	文艺理论基础研究	42005005	王春秀	88
10002113	英汉对比语言学	42005009	国美阳	88
10002109	英语文体学	42005009	国美阳	86
10002101	外语	42005002	李作宁	85
10002101	外语	42005016	李煜	84
10002120	泛函分析	42005018	卢莹	84
10002113	英汉对比语言学	42005007	陈敬	83

图 4-17 例 4.13 的查询结果



课程编号	课程名称	课程类别	学时	学分
10002107	大众文化研究	选修课	36	1
10002108	周易美学研究	选修课	36	1
10002102	马克思主义理论	学位公共课	60	2
10002115	音乐学概论	学位基础课	60	2
10002116	美术学概论	学位基础课	60	2
10002117	作品分析	专业课	60	2
10002118	西方现代美术史	专业课	60	2
10002123	线性偏微分方程	选修课	60	2
10002103	文艺理论基础问题	学位基础课	56	2
10002104	文艺美学	专业课	56	2
10002105	文化研究	专业课	56	2
10002106	中国古代美学经典	专业课	56	2
10002113	英汉对比语言学	选修课	36	2
10002114	翻译学	选修课	36	2
10002126	软件体系结构	专业课	36	2
10002119	代数学	学位基础课	72	3
10002120	泛函分析	学位基础课	72	3
10002121	最优化方法	专业课	72	3
10002122	非线性控制	专业课	72	3
10002111	美国小说	专业课	52	3
10002112	英美文论选读	专业课	52	3
10002124	面向对象软件工程	学位基础课	48	3
10002125	数据通信与计算机网	学位基础课	48	3
10002101	外语	学位公共课	120	4
10002109	英语文体学	学位基础课	52	4
10002110	英语语言学	学位基础课	52	4

图 4-18 例 4.14 的查询结果

例 4.15 检索所有女同学的学号, 姓名, 出生日期, 课程编号、课程名称和成绩, 并按出生日期的升序排序, 再按成绩的降序排序。

```
SELECT student.学号, student.姓名, 出生日期;  
score.课程编号, score.课程名称, 成绩;  
FROM student, score  
WHERE student.学号=score.学号 AND 性别="女";  
ORDER BY 出生日期 ASC, 6 DESC
```

查询结果如图 4-19 所示。

说明:

- 1) 在一个查询语句中, 可以按一列或多列进行排序。需要注意排序关键字之间用逗号隔开。
- 2) ORDER BY 短语中的排序依据可以用列序号表示, 如按成绩升序排序, 可以表示为“ORDER BY 4”, 其中 4 为第四列。

例 4.16 检索所有女同学的学号、姓名、年龄并按年龄升序排序。

```
SELECT 学号,姓名, YEAR( DATE ( ) ) - YEAR(出生日期) AS 年龄;
FROM student WHERE 性别="女" ORDER BY 年龄
```

学号	姓名	出生日期	课程编号	课程名称	成绩
42005009	国美阳	06/15/75	10002113	英汉对比语言学	86
42005009	国美阳	06/15/75	10002109	英语文体学	86
42005007	陈敏	01/08/77	10002113	英汉对比语言学	83
42005007	陈敏	01/08/77	10002103	文艺理论基础研究	78
42005007	陈敏	01/08/77	10002101	外国语	77
42005007	陈敏	01/08/77	10002107	大众文化研究	70
42005002	李作宇	05/14/80	10002101	外国语	85
42005002	李作宇	05/14/80	10002103	文艺理论基础研究	65
42005016	李梅	02/13/81	10002101	外国语	84
42005016	李梅	02/13/81	10002120	泛函分析	65
42005011	商娜	05/27/81	10002109	英语文体学	78
42005020	迟丽凤	06/01/81	10002101	外国语	91
42005020	迟丽凤	06/01/81	10002125	面向对象软件工程	66
42005004	张莉	06/09/82	10002101	外国语	0
42005004	张莉	06/09/82	10002103	文艺理论基础研究	0
42005019	孙楠	07/14/82	10002101	外国语	75
42005019	孙楠	07/14/82	10002125	面向对象软件工程	73
42005018	卢莹	08/23/82	10002101	外国语	96
42005018	卢莹	08/23/82	10002120	泛函分析	84
42005008	张辉	08/30/82	10002109	英语文体学	74
42005013	刘婧男	09/05/82	10002115	音乐学概论	78
42005013	刘婧男	09/05/82	10002101	外国语	50
42005012	赵妍名	09/24/82	10002101	外国语	89
42005012	赵妍名	09/24/82	10002115	音乐学概论	75
42005005	王春秀	10/13/82	10002103	文艺理论基础研究	88
42005005	王春秀	10/13/82	10002101	外国语	45

图 4-19 例 4.15 的查询结果

学号	姓名	年龄
42005004	张莉	27
42005005	王春秀	27
42005008	张辉	27
42005012	赵妍名	27
42005013	刘婧男	27
42005018	卢莹	27
42005019	孙楠	27
42005011	商娜	28
42005016	李梅	28
42005020	迟丽凤	28
42005002	李作宇	29
42005007	陈敏	32
42005009	国美阳	34

图 4-20 例 4.16 的查询结果

4.1.3 计算与分组查询

1. 简单的计算查询

事实上, SQL 不仅具有一般的检索能力, 而且还具有计算方式的检索能力。下面介绍常用于计算检索的函数, 如表 4-2 所示。

表 4-2 常用计算函数及其功能

函数名称	格式	功能
AVG	AVG(字段名)	按列计算平均值
SUM	SUM(字段名)	按列计算值的总和
COUNT	COUNT(字段名)	按列值统计个数
MAX	MAX(字段名)	求一列中的最大值
MIN	MIN(字段名)	求一列中的最小值

这些函数可以用在 SELECT 短语中对查询结果进行计算。

例 4.17 统计 student 表中有多少个学生记录。

```
SELECT COUNT(*) AS 人数 FROM student
```

查询结果如图 4-21 所示。

说明:

- 1) 如果要统计一个表中有多少元组(行)数, 可以用 COUNT(*)格式。
- 2) 表达式“COUNT(*)AS 人数”不仅统计出表中的记录个数, 而且为函数表达式命名。

例 4.18 在 score 表中统计选修了课程编号为“10002101”的记录个数。

```
SELECT COUNT(*) AS 人数 FROM score WHERE 课程编号="10002101"
```

查询结果如图 4-22 所示。

例 4.19 统计有多少个学生选修了“文艺理论基础研究”这门课程。

```
SELECT COUNT(*) AS 人数 FROM score WHERE 课程名称="文艺理论基础研究"
```

查询结果如图 4-23 所示。



人数
20

图 4-21 例 4.17 的查询结果



As人数
13

图 4-22 例 4.18 的查询结果



人数
7

图 4-23 例 4.19 的查询结果

例 4.20 统计成绩表中学生选修了多少门课程。

```
SELECT COUNT(DISTINCT 课程编号) AS 人数 FROM score
```

查询结果如图 4-24 所示。

说明：在 SQL 查询语句中，除非对表中的元组个数进行统计，一般情况下 COUNT 函数应该使用 DISTINCT 短语。

例 4.21 计算“外国语”这门课程的平均成绩。

```
SELECT AVG(成绩)AS 平均分 FROM score WHERE 课程名称="外国语"
```

查询结果如图 4-25 所示。

例 4.22 检索 score 表中选修了“文艺理论基础研究”这门课程的最高分和最低分。

```
SELECT 课程名称,MAX(成绩) AS 最高分,MIN(成绩) AS 最低分 FROM score;
WHERE 课程名称="文艺理论基础研究"
```

查询结果如图 4-26 所示。



人数
8

图 4-24 例 4.20 的查询结果



平均分
62.06

图 4-25 例 4.21 的查询结果



课程名称	最高分	最低分
文艺理论基础研究	96	0

图 4-26 例 4.22 的查询结果

例 4.23 检索 course 表中所有学位公共课的学时总和。

```
SELECT SUM(学时) AS 学时总和 FROM course WHERE 课程类别="学位公共课"
```

查询结果如图 4-27 所示。

2. 分组与计算查询

GROUP BY 短语用来对查询结果进行分组，把某一列的值相同的记录分在一组，通过计算每组产生一个结果。

格式：SELECT...FROM...[WHERE 连接条件 AND 查询条件]GROUP BY <字段名>[HAVING <分组条件表达式>]

说明：

- 1) GROUP BY 短语后面可以有多个字段名，分组时把在这些字段上值相同的记录分在一组。
- 2) HAVING 子句总是跟在 GROUP BY 子句之后，用来限定分组必须满足的条件，不可以单独使用。
- 3) 与连接条件表达式并不矛盾，当一个查询命令中既有 WHERE 短语同时又有 HAVING 短语时，是先用 WHERE 子句限定元组，然后进行分组，最后再用 HAVING 子句限定分组。

例 4.24 检索成绩表中每个学生的平均成绩。

```
SELECT 学号,AVG(成绩)AS 平均分 FROM score GROUP BY 学号
```

查询结果如图 4-28 所示。



学时总和
180

图 4-27 例 4.23 的查询结果

说明：在这个查询中，首先按学号进行分组，将学号相同的学生记录放在一起，然后再计算每个学生所选修的所有课程的平均分。

例 4.25 检索出每个学生的学号、姓名、总分、平均分、最高分和最低分，并将结果按总分降序排列。

```
SELECT score.学号,姓名, SUM(成绩) AS 总分, AVG(成绩) AS 平均分;
      MAX(成绩) AS 最高分,MIN(成绩) AS 最低分;
FROM student,score;
WHERE student.学号=score.学号;
GROUP BY score.学号;
ORDER BY 总分 DESC
```

查询结果如图 4-29 所示。

学号	平均分
42005001	72.00
42005002	65.00
42005003	85.50
42005004	0.00
42005005	66.50
42005006	73.00
42005007	77.00
42005008	74.00
42005009	87.00
42005010	68.00
42005011	78.00
42005012	82.00
42005013	64.00
42005014	0.00
42005015	68.00
42005016	74.50
42005017	68.50
42005018	90.00
42005019	74.00
42005020	78.50

图 4-28 例 4.24 的查询结果

学号	姓名	总分	平均分	最高分	最低分
42005007	陈敏	308	77.00	83	70
42005018	卢莹	180	90.00	96	84
42005009	国美阳	174	87.00	88	86
42005003	张华	171	85.50	96	75
42005012	赵妍名	164	82.00	89	75
42005020	迟丽凤	157	78.50	91	66
42005016	李梅	149	74.50	84	65
42005019	孙楠	148	74.00	75	73
42005006	宋志新	146	73.00	90	56
42005001	韩春杰	144	72.00	78	66
42005017	李帅	137	68.50	95	42
42005005	王春秀	133	66.50	88	45
42005013	刘婧男	128	64.00	78	50
42005011	商娜	78	78.00	78	78
42005008	张辉	74	74.00	74	74
42005010	李云鹤	68	68.00	68	68
42005015	王春宇	68	68.00	68	68
42005002	李作宁	65	65.00	65	65
42005004	张莉	0	0.00	0	0
42005014	陈庆梅	0	0.00	0	0

图 4-29 例 4.25 的查询结果

例 4.26 检索出平均分在 70 分以上的学生的学号、姓名和平均分，并将结果按平均分升序排列。

```
SELECT score.学号,姓名,AVG(成绩) AS 平均分;
FROM student,score;
WHERE student.学号=score.学号;
GROUP BY score.学号 HAVING AVG(成绩)>70;
ORDER BY 平均分
```

查询结果如图 4-30 所示。

例 4.27 检索至少有两名学生选修的课程的平均分，检索项包含课程编号、课程名称和平均分。

```
SELECT 课程编号,课程名称,AVG(成绩) AS 平均分 FROM score;
GROUP BY 课程编号;
HAVING COUNT(*)>=2
```

查询结果如图 4-31 所示。

说明：

- 1) ORDER BY 短语后面只能用字段名或别名，不可以用表达式。
- 2) “HAVING COUNT(*)>=2”表示：每一组的元组个数大于等于 2 时满足条件。

例 4.28 检索选修了两门或两门以上课程的学生的平均成绩，检索项包含学号、姓名、平均分。

```
SELECT 学号,姓名,AVG(成绩) AS 平均分 FROM score;
GROUP BY 学号 HAVING COUNT(*)>=2
```

查询结果如图 4-32 所示。

学号	姓名	平均分
42005001	韩春杰	72.00
42005006	宋志新	73.00
42005008	张辉	74.00
42005019	孙楠	74.00
42005016	李梅	74.50
42005007	陈敏	77.00
42005011	商娜	78.00
42005020	迟丽凤	78.50
42005012	赵妍名	82.00
42005003	张华	85.50
42005009	国美阳	87.00
42005018	卢莹	90.00

图 4-30 例 4.26 的查询结果

课程编号	课程名称	平均分
10002101	外国语	82.08
10002103	文艺理论基础研究	85.86
10002107	大众文化研究	70.33
10002109	英语文体学	76.50
10002113	英汉对比语言学	85.50
10002115	音乐学概论	76.50
10002120	泛函分析	81.33
10002125	面向对象软件工程	69.50

图 4-31 例 4.27 的查询结果

学号	姓名	平均分
42005001	韩春杰	72.00
42005003	张华	85.50
42005004	张莉	0.00
42005005	王春秀	66.50
42005006	宋志新	73.00
42005007	陈敏	77.00
42005009	国美阳	87.00
42005012	赵妍名	82.00
42005013	刘耀男	64.00
42005016	李梅	74.50
42005017	李帅	68.50
42005018	卢莹	90.00
42005019	孙楠	74.00
42005020	迟丽凤	78.50

图 4-32 例 4.28 的查询结果

4.1.4 带特殊运算符的条件查询

在进行一些更复杂、涉及更多关系的检索时，往往会用到一些特殊运算符，在这里将介绍一下可以在 SQL SELECT 查询中使用的几个特殊运算符。

1. 确定范围

格式：BETWEEN...AND...

例 4.29 查询出生日期在 1980 年 1 月 1 日至 1984 年 12 月 31 日之间的学生的学号、姓名、性别、出生日期。

```
SELECT 学号,姓名,性别,出生日期 FROM student;
WHERE 出生日期 BETWEEN {^1980-1-1} AND {^1984-12-31}
```

这个查询等价于：

```
SELECT * FROM student;
WHERE 出生日期>= {^1980-1-1} AND 出生日期<={^1984-12-31}
```

查询结果如图 4-33 所示。

说明：

1) 表示“……和……之间”，显然使用它会使表达条件更清晰、更简洁。

2) BETWEEN...AND...查询的结果包括两边的界值。

例如：BETWEEN {^1980-1-1} AND {^1984-12-31} 等价于

出生日期>= {^1980-1-1} AND 出生日期<={^1984-12-31}

2. 确定集合

利用 IN 操作可以查询字段值属于指定集合的记录。

格式：SELECT...FROM...;

WHERE 字段名 [NOT]IN (集合) [AND 连接条件]

例 4.30 查询选修了课程编号为“10002101”或“10002107”或“10002120”，并且成绩在 70 分以上的学生的学号、课程编号和成绩，并按成绩降序排序。

```
SELECT 学号,课程编号,成绩 FROM score;
WHERE 课程编号 IN ("10002101","10002107","10002120") AND 成绩>70;
ORDER BY 成绩 DESC
```

查询结果如图 4-34 所示。

例 4.31 查询没有选修课程编号为“10002101”或“10002107”或“10002120”，并且成绩在 80 分以上的学生的学号、课程编号和成绩，并按成绩降序排序。

```
SELECT 学号,课程编号,成绩 FROM score ;
WHERE 课程编号 NOT IN ("10002101","10002107","10002120") AND 成绩>70;
ORDER BY 成绩 DESC
```

查询结果如图 4-35 所示。

学号	姓名	性别	出生日期
42005001	韩春杰	男	10/15/83
42005002	李作宇	女	05/14/80
42005003	张华	男	02/11/84
42005004	张莉	女	06/09/82
42005005	王春秀	女	10/13/82
42005006	宋志新	男	02/15/82
42005008	张辉	女	08/30/82
42005010	李云鹤	男	07/15/82
42005011	陈娜	女	05/27/81
42005012	赵研名	女	09/24/82
42005013	刘耀男	女	09/05/82
42005014	陈庆海	男	08/10/81
42005016	李梅	女	02/13/81
42005018	卢莹	女	08/23/82
42005019	孙楠	女	07/14/82
42005020	迟丽凤	女	06/01/81

图 4-33 例 4.29 的查询结果

学号	课程编号	成绩
42005018	10002101	96
42005017	10002120	95
42005020	10002101	91
42005006	10002101	90
42005012	10002101	89
42005015	10002101	84
42005018	10002120	84
42005007	10002101	77
42005019	10002101	75
42005003	10002107	75

图 4-34 例 4.30 的查询结果

学号	课程编号	成绩
42005003	10002103	96
42005005	10002103	88
42005009	10002113	88
42005009	10002109	86
42005007	10002113	83
42005001	10002103	78
42005007	10002103	78
42005011	10002109	78
42005013	10002115	78
42005012	10002115	75
42005008	10002109	74
42005019	10002125	73

图 4-35 例 4.31 的查询结果

说明：IN 表示在一定的范围，表示不在一定范围则在前面加 NOT，即 NOT IN。

3. 部分匹配查询

以上例子均属于完全匹配查询，当不知道完全精确的值时，可以使用 LIKE 或 NOT LIKE 进行部分匹配查询（也称模糊查询）。在进行匹配运算时经常用到通配符“%”和“_”。“%”表示 0 个或多个字符；“_”表示 1 个字符。

LIKE 的基本格式如下：SELECT...FROM...;

WHERE <字段名> LIKE <字符串常量> [AND 连接条件]

例 4.32 查询所有姓“张”的学生的学号、姓名、性别、所在院系、专业名称。

```
SELECT 学号,姓名,性别,所在院系,专业名称;
FROM student WHERE 姓名 LIKE "张%"
```

查询结果如图 4-36 所示。

学号	姓名	性别	所在院系	专业名称
42005003	张华	男	文学院	中国古代文学
42005004	张莉	女	文学院	中国现当代文学
42005008	张辉	女	英语学院	英语语言文学

图 4-36 例 4.32 的查询结果

学号	姓名	性别	所在院系	专业名称
42005001	韩春杰	男	文学院	汉语言文字学
42005005	王春秀	女	文学院	中国现当代文学
42005015	王春宇	男	艺术学院	美术学

图 4-37 例 4.33 的查询结果

例 4.33 查询第二个汉字是“春”的学生的学号、姓名、性别、所在院系、专业名称。

```
SELECT 学号,姓名,性别,所在院系,专业名称;
FROM student WHERE 姓名 LIKE "_春%"
```

查询结果如图 4-37 所示。

4. 不等于 (!=)

例 4.34 查询家庭住址属于黑龙江省的并且语种不属于俄语的学生的学号、姓名、性别、家庭住址。

```
SELECT 学号,姓名,性别,家庭住址 FROM student;
WHERE 家庭住址 LIKE "黑龙江%" AND 语种!="俄语"
```

查询结果如图 4-38 所示。

学号	姓名	性别	家庭住址
42005001	韩春杰	男	黑龙江省齐齐哈尔市克山县
42005002	李作宇	女	黑龙江省牡丹江市宁安市
42005006	宋志新	男	黑龙江省佳木斯市
42005007	陈娜	女	黑龙江省尚志市
42005008	张辉	女	黑龙江省富锦市
42005010	李云鹤	男	黑龙江省佳木斯市
42005013	刘耀男	女	黑龙江省鸡西市
42005019	孙楠	女	黑龙江省大庆市
42005020	迟丽凤	女	黑龙江省五常市

图 4-38 例 4.34 的查询结果

4.1.5 利用空值查询

空值就是缺值或还没有确定值。比如表示价格的一个字段值，空值表示没有定价，而数值 0 可能表示免费。空值与空（或空白）字符串、数值 0 等具有不同的含义。假设在 score 表中有些学生某门课程还没有考试，则可以将成绩设为 NULL，表示成绩尚未确定，并不表示 0 分。在 SQL SELECT 查询中，空值的具体格式如下：

```
SELECT... FROM ... ;
WHERE <字段名> IS [NOT]NULL [AND 连接条件]
```

说明：

- 1) <字段名>IS NULL（为空值）。
- 2) <字段名>IS NOT NULL（不为空值）。
- 3) 在表达式中不能写成“=NULL”或“!=NULL”

例 4.35 检索出还没有确定授课教师的课程信息。

方法一：SELECT * FROM course WHERE 教师编号 IS NULL

方法二：SELECT * FROM course WHERE 教师姓名 IS NULL

查询结果如图 4-39 所示。



课程编号	课程名称	开课学院	课程类别	学时	学分	开课学期	教师姓名	教师编号	考核方式
10002101	外语	NULL	学位公共课	120	4	1	NULL	NULL	闭卷
10002102	马克思主义理论	NULL	学位公共课	60	2	1	NULL	NULL	开卷

图 4-39 例 4.35 的查询结果

例 4.36 检索有哪些学院已经确定了开设哪些课程，检索内容包括课程编号、课程名称、开课学院、课程类别。

```
SELECT 课程编号,课程名称,开课学院,课程类别;
```

```
FROM course;
```

```
WHERE 开课学院 IS NOT NULL
```

查询结果如图 4-40 所示。



课程编号	课程名称	开课学院	课程类别
10002103	文艺理论基础问题	文学院	学位基础课
10002104	文艺美学	文学院	专业课
10002105	文化研究	文学院	专业课
10002106	中国古代美学经典文献	文学院	专业课
10002107	大众文化研究	文学院	选修课
10002108	周易美学研究	文学院	选修课
10002109	英语文体学	英语学院	学位基础课
10002110	英语语言史	英语学院	学位基础课
10002111	美国小说	英语学院	专业课
10002112	英美文论选读	英语学院	专业课
10002113	英汉对比语言学	英语学院	选修课
10002114	翻译学	英语学院	选修课
10002115	音乐学概论	艺术学院	学位基础课
10002116	美术学概论	艺术学院	学位基础课
10002117	作品分析	艺术学院	专业课
10002118	西方现代美术史	艺术学院	专业课
10002119	代数学	数学学院	学位基础课
10002120	泛函分析	数学学院	学位基础课
10002121	最优化方法	数学学院	专业课
10002122	非线性控制	数学学院	专业课
10002123	线性偏微分方程	数学学院	选修课
10002124	面向对象软件工程	计算机学院	学位基础课
10002125	数据通信与计算机网络	计算机学院	学位基础课
10002126	软件体系结构	计算机学院	专业课

图 4-40 例 4.36 的查询结果

4.1.6 嵌套查询

接下来讨论另一类基于多个关系的查询，这类查询所要求的结果出自一个关系，但相关的条件却涉及多个关系。

格式：SELECT...FROM...WHERE...(SELECT...FROM...WHERE...)

说明：

1) 在嵌套查询中有两层 SELECT...FROM...[WHERE...]查询块，即内层查询块和外层查询块，内层查询（括号里的查询）也叫子查询，外层查询也叫主查询。

2) 在进行查询时，先执行子查询，子查询形成一个结果作为主查询的条件，然后再进行外层的主查询。

在嵌套查询中子查询的常用谓词和量词有以下几种。

1. 带有 IN 或 NOT IN 运算符的嵌套查询

由于在嵌套查询中，子查询的结果往往是一个集合，所以运算符 IN 和 NOT IN 是嵌套查询中最经常使用的谓词。带有 IN 或 NOT IN 运算符的嵌套查询是指主查询与子查询之间用 IN 或 NOT IN 连接，用来判断某个属性列值是否在子查询的结果中。

格式：SELECT...FROM...WHERE...IN(SELECT...FROM...WHERE...)

说明：

1) 在子查询中 SELECT 后的字段名必须与主查询中 IN 短语前面的字段名相同。

2) 子查询中不能有 ORDER BY 排序短语。

例 4.37 将所有选修了“英语文体学”这门课程并且成绩大于 70 分的学生的学号、姓名、性别、所在院系、专业名称等信息检索出来。

```
SELECT 学号,姓名,性别,所在院系,专业名称;
FROM student;
WHERE 学号 IN (SELECT 学号 FROM score;
                WHERE 课程名称="英语文体学" AND 成绩>70)
```

等价于：

```
SELECT student.学号, student.姓名,性别,所在院系,专业名称;
FROM student,score;
WHERE student.学号=score.学号 AND 成绩>70 AND 课程名称="英语文体学"
```

查询结果如图 4-41 所示。

例 4.38 检索没有被选修的课程的课程名称、课程类别、学时、学分、考核方式等信息。

```
SELECT 课程名称,课程类别,学时,学分,考核方式 FROM course;
WHERE 课程编号 NOT IN (SELECT 课程编号 FROM score)
```

查询结果如图 4-42 所示。



学号	姓名	性别	所在院系	专业名称
42005008	张辉	女	英语学院	英语语言文学
42005009	团美阳	女	英语学院	英语语言文学
42005011	商娜	女	英语学院	英语语言文学

图 4-41 例 4.37 的查询结果



课程名称	课程类别	学时	学分	考核方式
马克思主义理论	学位公共课	60	2	开卷
文艺美学	专业课	56	2	论文
文化研究	专业课	56	2	论文
中国古代美学经典文献	专业课	56	2	闭卷
周易美学研究	选修课	36	1	开卷
英语语言史	学位基础课	52	4	闭卷
美国小说	专业课	52	3	闭卷
英美文论选读	专业课	52	3	论文
翻译学	选修课	36	2	论文
美学概论	学位基础课	60	2	闭卷
作品分析	专业课	60	2	闭卷
西方现代艺术史	专业课	60	2	闭卷
代数学	学位基础课	72	3	闭卷
最优化方法	专业课	72	3	闭卷
非线性控制	专业课	72	3	开卷
线性偏微分方程	选修课	60	2	论文
面向对象软件工程	学位基础课	48	3	闭卷
软件体系结构	专业课	36	2	论文

图 4-42 例 4.38 的查询结果

例 4.39 检索没有确定授课的教师信息，查询内容包括教师姓名、出生日期、所在院系、研究方向、职称、学位。

```
SELECT 教师姓名,出生日期,所在院系,研究方向,职称,学位 FROM teacher;
WHERE 教师编号 NOT IN (SELECT 教师编号 FROM course)
```

查询结果如图 4-43 所示。

2. 带有比较运算符的嵌套查询

带有比较运算符的子查询是指主查询与子查询之间用比较运算符进行连接。当能确切知道内层查询返回的是单值时，可以用 <、<=、>、>=、<>、!= 等运算符。

格式：SELECT...FROM...WHERE...比较运算符(SELECT...FROM...WHERE...)

例 4.40 检索成绩高于平均分的学生的学号、姓名、课程名称、成绩，并按成绩降序排序。

```
SELECT 学号,姓名,课程名称,成绩 FROM score;
WHERE 成绩> (SELECT AVG(成绩) FROM score) ORDER BY 成绩 DESC
```

查询结果如图 4-44 所示。



教师姓名	出生日期	所在院系	研究方向	职称	学位
孙学成	04/30/72	文学院	中国现代文学	教授	硕士
王立军	03/26/78	计算机学院	计算机网络	讲师	硕士

图 4-43 例 4.39 的查询结果



学号	姓名	课程名称	成绩
42005018	卢莹	外国语	96
42005003	张华	文艺理论基础研究	96
42005017	李帅	泛函分析	95
42005020	迟丽凤	外国语	91
42005006	宋志新	外国语	90
42005012	赵妍名	外国语	89
42005005	王春秀	文艺理论基础研究	88
42005009	国美阳	英汉对比语言学	88
42005009	国美阳	英语文体学	86
42005002	李作宁	外国语	85
42005016	李梅	外国语	84
42005018	卢莹	泛函分析	84
42005007	陈敏	英汉对比语言学	83
42005001	韩春杰	外国语	78
42005001	韩春杰	文艺理论基础研究	78
42005007	陈敏	文艺理论基础研究	78
42005011	商娜	英语文体学	78
42005013	刘婧男	音乐学概论	78
42005007	陈敏	外国语	77
42005019	孙楠	外国语	75
42005003	张华	大众文化研究	75
42005012	赵妍名	音乐学概论	75
42005008	张辉	英语文体学	74
42005019	孙楠	面向对象软件工程	73
42005007	陈敏	大众文化研究	70

图 4-44 例 4.40 的查询结果

3. 带有 ANY、ALL、SOME 量词的嵌套查询

ANY、ALL 和 SOME 为量词，ANY 和 SOME 是同义词，在进行比较运算时只要子查询中有一个记录能使结果为真，则结果为真；ALL 则要求子查询中的所有记录都使结果为真时，结果才为真。

格式：SELECT...FROM...;

WHERE<表达式><比较运算符>[ANY|ALL|SOME](SELECT...FROM... [WHERE...])

例 4.41 查询学分最高的课程信息。

方法一：SELECT * FROM course WHERE 学分>=ALL (SELECT 学分 FROM course)

方法二：SELECT * FROM course WHERE NOT 学分< ANY (SELECT 学分 FROM course)

方法三：SELECT * FROM course WHERE NOT 学分< SOME (SELECT 学分 FROM course)

方法四：SELECT * FROM course WHERE 学分= (SELECT MAX(学分) FROM course)

查询结果如图 4-45 所示。

课程编号	课程名称	开课学院	课程类别	学时	学分	开课学期	教师姓名	教师编号	考核方式
10002101	外国语	NULL	学位公共课	120	4	1	NULL	NULL	闭卷
10002109	英语文体学	英语学院	学位基础课	52	4	1	李爽	10230008	闭卷
10002110	英语语言史	英语学院	学位基础课	52	4	1	周世明	10230007	闭卷

图 4-45 例 4.41 的查询结果

例 4.42 查询学分不是最高的课程信息。

方法一: `SELECT * FROM course;`
`WHERE 学分<ANY (SELECT 学分 FROM course)`

方法二: `SELECT * FROM course;`
`WHERE 学分<SOME (SELECT 学分 FROM course)`

4. 带有 EXISTS (NOT EXISTS) 谓词的子查询

EXISTS 为谓词, EXISTS 和 NOT EXISTS 是用来检查在子查询中是否有结果返回, 也就是存在元组 (记录) 或不存在元组 (记录)。

格式: `SELECT...FROM...;`

`WHERE [NOT] EXISTS(SELECT...FROM...[WHERE...])`

例 4.43 查询职称为“教授”的教师授课情况, 检索项包含课程名称、课程类别、学时、授课教师姓名。

`SELECT 课程名称, 课程类别, 学时, 教师姓名;`
`FROM course;`
`WHERE EXISTS ;`

(`SELECT * FROM teacher WHERE teacher.教师编号=course.教师编号 AND 职称="教授"`)

注意: 本例中内层查询引用了外层查询的表, 只有这样使用谓词 EXISTS 或 NOT EXISTS 才有意义。

查询结果如图 4-46 所示。

例 4.44 检索选修的每门课程的成绩都高于或等于 75 分的学生的学号、姓名和性别。

`SELECT 学号, 姓名, 性别 FROM student WHERE NOT EXISTS;`
(`SELECT * FROM score WHERE student.学号=score.学号 AND 成绩<75`)

查询结果如图 4-47 所示。

课程名称	课程类别	学时	教师姓名
文艺理论基础问题	学位基础课	56	陈志伟
文艺美学	专业课	56	张建
文化研究	专业课	56	陆秀君
中国古代美学经典文献	专业课	56	战爱成
大众文化研究	选修课	36	崔明村
周易美学研究	选修课	36	崔明村
英语文体学	学位基础课	52	李爽
英语语言史	学位基础课	52	周世明
美国小说	专业课	52	周世明
英美文化选读	专业课	52	程雅平
英汉对比语言学	选修课	36	王宏旭
翻译学	选修课	36	成双
音乐学概论	学位基础课	60	王军
美术学概论	学位基础课	60	白强
作品分析	专业课	60	苗辰光
西方现代美术史	专业课	60	林然
代数学	学位基础课	72	张文斌
泛函分析	学位基础课	72	赵正
最优方法	专业课	72	王锦
非线性控制	专业课	72	宋军
线性偏微分方程	选修课	60	宋军
面向对象软件工程	学位基础课	48	刘绍福

图 4-46 例 4.43 的查询结果

学号	姓名	性别
42005009	国美阳	女
42005011	商娜	女
42005012	赵妍名	女
42005018	卢莹	女

图 4-47 例 4.44 的查询结果

4.1.7 别名与自连接查询

在连接操作中, 要使用关系名作前缀, 有时为简单起见, SQL 允许在 FROM 短语中为关

系名定义别名。

格式: <关系名> <别名>

例 4.45 查询学生的姓名, 性别和所选修课程的课程名称、学分、学时和成绩信息, 要求成绩必须大于 80 分, 且按成绩降序排列。

```
SELECT student.姓名, 性别, score.课程名称, 学分, 学时, 成绩;
FROM student, course, score ;
WHERE student.学号=score.学号 AND ;
course.课程编号=score.课程编号 AND 成绩>80 ;
ORDER BY 成绩 DESC
```

在上面的查询中, 如果使用别名会简单一些。下面给出利用别名的查询语句, 它与上面查询是等价的。

```
SELECT S.姓名, 性别, SC.课程名称, 学分, 学时, 成绩;
FROM student S, course C, score SC;
WHERE S.学号=SC.学号 AND C.课程编号=SC.课程编号 AND 成绩>80;
ORDER BY 成绩 DESC
```

查询结果如图 4-48 所示。

姓名	性别	课程名称	学分	学时	成绩
张华	男	文艺理论基础研究	2	56	96
卢莹	女	外国语	4	120	96
李冲	男	泛函分析	3	72	95
迟丽凤	女	外国语	4	120	91
宋志新	男	外国语	4	120	90
赵妍名	女	外国语	4	120	89
王春秀	女	文艺理论基础研究	2	56	88
国美阳	女	英汉对比语言学	2	36	88
国美阳	女	英语文体学	4	52	86
李作宁	女	外国语	4	120	85
李梅	女	外国语	4	120	84
卢莹	女	泛函分析	3	72	84
陈敏	女	英汉对比语言学	2	36	83

图 4-48 例 4.45 的查询结果

说明: 在上面的例子中, 别名并不是必需的, 但是在关系的自连接操作中, 别名则是必不可少的。SQL 不仅可以对多个关系实行连接操作, 也可将同一关系与其自身进行连接, 这种连接就称为自连接。在这种自连接操作关系上, 本质上存在着一种特殊的递归联系, 也就是关系中的一些元组, 根据出自同一值域的两个不同的属性, 可以与另外一些元组有一种对应关系 (一对多的联系)。注: 元组即记录。

下面以“先修课”表为例介绍自连接, 在“先修课”关系中, “课程编号”与“先修课”出自一个值域, 一门课可以是多门课程的先修课。

例 4.46 试查询某门课程的先修课的课程名称。

分析: 需要利用自连接, 首先将“先修课”表另起两个别名, 分别为 XS1 和 XS2, 代表两个表 (如图 4-49 所示), 然后可以类似多表连接查询的操作进行查询即可。

课程编号	课程名称	先修课
08	C语言	
02	程序设计	08
04	数据结构	08
06	操作系统	02
05	离散数学	01
07	组成原理	06
01	高等数学	

图 4-49 “先修课”表

```
SELECT XS1.课程编号, XS2.课程名称;
FROM 先修课 XS1, 先修课 XS2;
WHERE XS1.先修课=XS2.课程编号
```

查询结果如图 4-50 所示。

课程编号	课程名称
02	C语言
04	C语言
06	程序设计
07	操作系统
05	高等数学

图 4-50 例 4.46 查询结果

4.1.8 超连接查询

在 SQL 标准中还提供了一种新的连接运算：超连接运算。它与原来所了解的等值连接和自然连接不同。原来的连接是只有满足连接条件，相应的结果才会出现在结果表中；而超连接运算是，首先保证一个表中满足条件的元组都在结果中，然后将满足连接条件的元组与另一个表中的元组进行连接，不满足连接条件的则应将来自另一表的属性置为空值。

在一般 SQL 中的超连接运算符是 “* =” 和 “= *”。

“* =” 也叫左连接：含义是在结果表中包含第一个表中满足条件的所有记录；如果有在连接条件上匹配的元组，则第二个表返回相应值，否则第二个表返回空值。

“= *” 也叫右连接：含义是在结果表中包含第二个表中满足条件的所有记录；如果有在连接条件上匹配的元组，则第一个表返回相应值，否则第一个表返回空值。

在 Visual FoxPro 中不支持 “* =” 和 “= *” 这个超连接运算符，而对于 Visual FoxPro 有专门的连接运算格式，它们支持超连接查询。

格式：SELECT <检索项>;

FROM <表名 1>INNER|LEFT|RIGHT|FULL JOIN <表名 2>;

ON <连接条件>;

WHERE <查询条件>

说明：

- 1) INNER JOIN 短语：等价于 JOIN 为普通连接。在 Visual FoxPro 中称为内部连接。
- 2) LEFT JOIN 短语：为左连接。
- 3) RIGHT JOIN 短语：为右连接。
- 4) FULL JOIN 短语：为全连接。即两个表中的记录不管是否满足连接条件将都在目标表或查询结果中出现，不满足连接条件的记录对应部分为 NULL。
- 5) ON 短语：用于指定连接条件。
- 6) 在超连接查询中，连接条件与查询条件要分开并且放在不同的命令短语后面。
- 7) 4 种超连接类型：在查询过程中，超连接类型在 FROM 短语中给出。

下面通过例题来观察超连接运算。用到 score 表和 student 表，如图 4-51 和图 4-52 所示。

学号	课程编号	成绩
200731001	02	91.00
200731001	05	88.00
200731004	05	95.00
200731005	01	94.00
200731006	01	83.00
200731006	04	90.00
200731007	02	80.00
200731009	04	82.00
200731011	03	96.00
200731012	02	70.00
200731013	03	79.00

图 4-51 score 表

学号	姓名	性别	出生日期	专业	家庭住址
200731003	李梅	女	08/06/88	英语	长沙
200731004	王东	男	11/04/87	商务英语	哈尔滨
200731005	孙丽娜	女	07/28/89	英语	呼和浩特
200731006	郭光	男	03/16/88	计算机科学与技术	哈尔滨
200731007	李爽	女	09/25/89	商务英语	上海
200731008	刘德强	男	10/30/88	应用数学	北京
200731009	张昊天	男	07/29/90	对外汉语	合肥
200731010	孙明娇	男	NULL	应用数学	北京

图 4-52 student 表

例 4.47 内部连接。即只有满足连接条件的记录才出现在查询结果中。

```
SELECT student.姓名, 成绩;
FROM student INNER JOIN score;
ON student.学号 = score.学号
```

等价于

```
SELECT student.姓名, 成绩;
FROM student, score;
WHERE student.学号 = score.学号
```

查询结果如图 4-53 所示。

例 4.48 左连接。即除满足连接条件的记录出现在查询结果中外，第一个表中不满足连接条件的记录也出现在查询结果中。

```
SELECT student.姓名, 成绩;
FROM student LEFT JOIN score;
ON student.学号 = score.学号
```

查询结果如图 4-54 所示。

例 4.49 右连接。即除满足连接条件的记录出现在查询结果中外，第二个表中不满足连接条件的记录也出现在查询结果中。

```
SELECT student.姓名, 成绩;
FROM student RIGHT JOIN score;
ON student.学号 = score.学号
```

查询结果如图 4-55 所示。

姓名	成绩
王海东	91.00
王海东	88.00
王东	95.00
孙丽娜	94.00
郭光	83.00
郭光	90.00
李爽	80.00
张昊天	82.00

图 4-53 例 4.47 的查询结果

姓名	成绩
王海东	91.00
王海东	88.00
黄晓棠	NULL
李梅	NULL
王东	95.00
孙丽娜	94.00
郭光	83.00
郭光	90.00
李爽	80.00
刘国强	NULL
张昊天	82.00
孙明姝	NULL

图 4-54 例 4.48 的查询结果

姓名	成绩
王海东	91.00
王海东	88.00
王东	95.00
孙丽娜	94.00
郭光	83.00
郭光	90.00
李爽	80.00
张昊天	82.00
NULL	96.00
NULL	70.00
NULL	79.00

图 4-55 例 4.49 的查询结果

例 4.50 全连接。即除满足连接条件的记录出现在查询结果中外，两个表中不满足连接条件的记录也出现在查询结果中

```
SELECT student.姓名, 成绩;
FROM student FULL JOIN score;
ON student.学号 = score.学号
```

查询结果如图 4-56 所示。

例4.51 用SQL语句性别为“男”且考试成绩在85分以上（含85分）的学生的全部信息，并将结果按学号升序存入自由表nine.dbf文件中（结构同student，并在其后加入成绩字段）。

```
SELECT student.*,score.成绩;
FROM course INNER JOIN score;
INNER JOIN student;
ON score.学号 = student.学号;
ON course.课程编号 = score.课程编号;
WHERE score.成绩 >= 85;
AND 性别="男";
```

```
ORDER BY student.学号;
INTO TABLE NINE.DBF
查询结果如图 4-57 所示。
```



姓名	成绩
王海东	91.00
王海东	88.00
黄晓荣	NULL
李梅	NULL
王东	95.00
孙丽娜	94.00
郭光	83.00
郭光	90.00
李爽	80.00
刘德强	NULL
张昊天	82.00
孙明娟	NULL
NULL	96.00
NULL	70.00
NULL	79.00

图 4-56 例 4.50 的查询结果



学号	姓名	性别	出生日期	专业	家庭住址	成绩
200731001	王海东	男	02/22/88	国际经济与贸易	武汉	91.00
200731001	王海东	男	02/22/88	国际经济与贸易	武汉	88.00
200731004	王东	男	11/04/87	商务英语	哈尔滨	95.00
200731006	郭光	男	03/16/88	计算机科学与技术	哈尔滨	90.00

图 4-57 例 4.51 的查询结果

4.1.9 集合的并运算

在 Visual FoxPro 中，将两个 SELECT 语句的查询结果通过并运算合并成一个查询结果。为进行并运算，要求两个查询结果具有相同的字段个数，并且对应字段的值要出自同一域，也就是具有相同的数据类型和取值范围。

格式：SELECT...FROM...WHERE...;

UNION;

SELECT...FROM...WHERE...

例 4.52 查询课程编号为“10002107”与“10002109”的选课信息。

```
SELECT * FROM score WHERE 课程编号="10002107";
```

```
UNION;
```

```
SELECT * FROM score WHERE 课程编号=" 10002109"
```

查询结果如图 4-58 所示。



课程编号	课程名称	学号	姓名	成绩
10002107	大众文化研究	42005001	韩春杰	66
10002107	大众文化研究	42005003	张华	75
10002107	大众文化研究	42005007	陈敏	70
10002109	英语文体学	42005008	张梅	74
10002109	英语文体学	42005009	国美阳	86
10002109	英语文体学	42005010	李云鹤	68
10002109	英语文体学	42005011	商娜	78

图 4-58 例 4.52 的查询结果

4.1.10 查询中的几个特殊选项

1. 显示部分结果

格式：TOP n [PERCENT]

功能：只需要满足条件的前几个记录。

说明：

- 1) n 是 1~32767 之间的整数，说明显示前几个记录。
- 2) 当使用 PERCENT 参数时，说明显示结果中前百分之几的记录。
- 3) 通常与 ORDER BY 子句连用。

例 4.53 显示成绩最高的 3 名学生选课信息。

```
SELECT * TOP 3 FROM score ORDER BY 成绩 DESC
```

查询结果如图 4-59 所示。

例 4.54 显示成绩最低的那 10% 的学生选课信息。

```
SELECT * TOP 10 PERCENT FROM score ORDER BY 成绩 ASC
```

查询结果如图 4-60 所示。



课程编号	课程名称	学号	姓名	成绩
10002101	外语	42005018	卢莹	96
10002103	文艺理论基础研究	42005003	张华	96
10002120	泛函分析	42005017	李帅	95

图 4-59 例 4.53 的查询结果



课程编号	课程名称	学号	姓名	成绩
10002101	外语	42005004	张莉	0
10002101	外语	42005014	陈庆海	0
10002103	文艺理论基础研究	42005004	张莉	0
10002101	外语	42005017	李帅	42

图 4-60 例 4.54 的查询结果

2. 将查询结果存放在数组中

格式: INTO ARRAY 数组名

例 4.55 将 student 表信息存放在数组 temp 中。

```
SELECT * FROM student INTO ARRAY TEMP
```

说明:

1) 一般将存放查询结果的数据作为二维数组来使用, 每一行一条记录, 每列对应于查询结果的一列。查询结果存放在数组中, 可以非常方便地在程序中使用。

2) 运行该命令后, 数组元素 temp(1,1) 中存放 student 表中第一条记录的第一个值“42005001”, temp(1,4) 中存放 student 表中第一条记录的第四个值“学位公共课”。

3. 将结果存放在临时文件中

格式: INTO CURSOR 临时表名

例 4.56 将 score 表中的信息保存在临时文件“学生成绩”中, 并浏览“学生成绩”表。

```
SELECT * FROM score INTO CURSOR 学生成绩
```

```
BROWSE
```

说明:

1) 临时表是一个只读的 DBF 文件, 当查询结束后该临时文件是当前文件, 可像一般的 DBF 文件一样使用, 当关闭文件时该文件将自动删除。

2) 一般利用 INTO CURSOR 短语存放一些临时结果, 比如一些复杂的汇总可能需要分阶段完成, 需要根据几个中间结果再汇总等, 这时利用该短语存放中间结果就非常合适, 当使用完后这些临时文件会自动删除。

4. 将结果存放在永久表中

格式: INTO TABLE|DBF TABLENAME

例 4.57 查询所有“学位基础课”信息, 并将查询结果按学时降序保存在永久表“课程”中。

```
SELECT * FROM course INTO TABLE 课程;
WHERE 课程类别="学位基础课" ORDER BY 学时 DESC
```

另一种格式:

```
SELECT * FROM course;
WHERE 课程类别="学位基础课" ORDER BY 学时 DESC INTO TABLE 课程
```

说明:

1) INTO TABLE <表名> 的位置比较灵活, 可以放在 FROM 短语后, 也可位于 SQL SELECT 查询语句的最后面。

2) 对于查询的结果被直接保存在永久表中, 如果需要查看结果可以使用下面的命令操作。

USE 表名

BROWSE

3) 通过该子句可实现表的复制。

例 4.58 将 student 表中的全部信息复制到新表“学生”中。

```
SELECT * FROM student INTO TABLE 学生
```

例4.59 利用SQL语句查询选修了职称为“副教授”的课程的学生全部信息, 并将结果按成绩降序存放在 netp.dbf文件中(表的结构同student, 并在其后加入课程名称和成绩字段)。

```
SELECT student.*, course.课程名称,score.成绩;
FROM student ,score, course ,teacher;
WHERE score.课程编号 = course.课程编号 AND;
student.学号 = score.学号 AND teacher.教师编号=course.教师编号;
AND 职称="副教授";
ORDER BY 成绩 DESC;
INTO TABLE NETP.DBF
```

查询结果如图 4-61 所示。

学号	姓名	性别	出生日期	民族	政治面貌	所在院系	专业名称	语种	导师姓名	家庭住址	学籍变动	照片	课程名称	成绩
42005003	张华	男	02/11/84	满族	F	文学院	中国古代文学	俄语	崔明材	哈尔滨市	memo	gen	文艺理论基础问题	96
42005005	王春秀	女	10/13/82	汉族	F	文学院	中国现代文学	英语	战爱成	湖南省长沙市	memo	gen	文艺理论基础问题	88
42005001	韩春杰	男	10/15/83	汉族	T	文学院	汉语言文学	英语	连秀君	黑龙江省齐齐哈尔市克山县	memo	gen	文艺理论基础问题	78
42005007	陈敏	女	01/08/77	汉族	F	文学院	中国现代文学	英语	战爱成	黑龙江省尚志市	memo	gen	文艺理论基础问题	78
42005003	张华	男	02/11/84	满族	F	文学院	中国古代文学	俄语	崔明材	哈尔滨市	memo	gen	大众文化研究	75
42005007	陈敏	女	01/08/77	汉族	F	文学院	中国现代文学	英语	战爱成	黑龙江省尚志市	memo	gen	大众文化研究	70
42005001	韩春杰	男	10/15/83	汉族	T	文学院	汉语言文学	英语	连秀君	黑龙江省齐齐哈尔市克山县	memo	gen	大众文化研究	66
42005002	李作宇	女	05/14/80	汉族	F	文学院	中国古代文学	英语	张建	黑龙江省牡丹江市宁安市	memo	gen	文艺理论基础问题	65
42005006	宋志新	男	02/15/82	达斡尔族	F	文学院	中国现代文学	日语	陈志伟	黑龙江省佳木斯市	memo	gen	文艺理论基础问题	56
42005004	张莉	女	06/09/82	汉族	T	文学院	中国现代文学	英语	孙学斌	山东省聊城市	memo	gen	文艺理论基础问题	0

图 4-61 例 4.59 的查询结果

5. 将结果存放到文本文件中

格式: TO FILE FILENAME [ADDITIVE]

说明:

1) FILENAME 给出文本文件的名称。

2) 如果使用 ADDITIVE 短语, 则将结果追加到原文件的尾部, 否则将覆盖原有文件。

3) TO FILE 短语可以放置在 FROM 短语中, 也可以放置在查询语句的最后面。

例 4.60 将所有成绩大于 85 分的成绩信息按成绩降序保存到文本文件 temp1.txt 中。

```
SELECT * FROM score WHERE 成绩>85 TO FILE TEMP1 ORDER BY 成绩 DESC
```

查询结果如图 4-62 所示。

例 4.61 将选课在 2 门课程以上(包括 2 门)的学生的学号、姓名、平均分和选课门数按平均分降序排序, 并将结果存放于文本文件 temp2.txt 中(字段名包含学号、姓名、平均分和选课门数)。

```
SELECT student.学号, student.姓名, AVG(score.成绩) AS 平均分,
COUNT(score.课程编号) AS 选课门数;
FROM student, score, course;
WHERE student.学号 = score.学号 AND;
score.课程编号 = course.课程编号;
GROUP BY student.学号;
HAVING COUNT(score.课程编号)>=2;
ORDER BY 3 DESC;
TO FILE TEMP2.TXT
```

查询结果如图 4-63 所示。

课程编号	课程名称	学号	姓名	成绩
10002101	外语	42005018	卢宝	96
10002103	文艺理论基础研究	42005003	张华	96
10002120	名号分析	42005017	李梅	95
10002101	外语	42005020	迟丽凤	91
10002101	外语	42005006	宋志新	90
10002101	外语	42005012	赵妍名	89
10002103	文艺理论基础研究	42005005	王春秀	88
10002113	英语对比语言学	42005009	国美阳	88
10002109	英语文律学	42005009	国美阳	86

图 4-62 例 4.60 的查询结果

学号	姓名	平均分	选课门数
42005018	卢宝	90.00	2
42005009	国美阳	87.00	2
42005012	赵妍名	82.00	2
42005003	张华	78.67	3
42005020	迟丽凤	78.50	2
42005007	陈敏	77.00	4
42005002	李作宁	75.00	2
42005016	李梅	74.50	2
42005001	韩森杰	74.00	3
42005019	孙楠	74.00	2
42005006	宋志新	73.00	2
42005017	李刚	68.50	2
42005005	王春秀	66.50	2
42005013	刘婧男	64.00	2
42005004	张莉	0.00	2

图 4-63 例 4.61 的查询结果

6. 将结果直接输出到打印机

格式: TO PRINTER [PROMPT]

说明: 使用 TO PRINTER[PROMPT]命令可以直接将查询的结果输出到打印机, 如果使用了 PROMPT 选项, 在开始打印之前会打开打印机设置对话框。

4.2 数据操作

SQL 语言的操作功能是指对数据库中数据的操作功能, 主要包括数据的插入、更新和删除 3 个方面的操作, 下面通过实例具体介绍。

4.2.1 插入操作

Visual FoxPro 支持两种 SQL 插入命令的格式。

第一种格式: (标准格式)

INSERT INTO <表名>[(字段名 1[,字段名 2...]);

VALUSE(表达式 1[,表达式 2...])

第二种格式: (Visual FoxPro 的特殊格式)

INSERT INTO <表名> FROM ARRAY <数组名>|MEMVAR

功能: 用于将数据插入到指定的表中。

说明:

- 1) INSERT INTO <表名>: 说明向哪个表中插入记录。
- 2) [(字段名 1[,字段名 2...]): 如果被插入的记录是一条完整的记录, 则可以省略字段名序列; 如果被插入的记录不是一条完整的记录, 则将值所对应的字段名列出。
- 3) VALUSE(表达式 1[,表达式 2...]): 要插入的具体的记录值。
- 4) FROM ARRAY <数组名>: 表示数据来源于指定的数组。
- 5) FROM MEMVAR: 将同名的内存变量的值插入到表中, 如果不存在同名的内存变量, 那么相应的字段为默认值或空值。

例 4.62 向 score 表中插入一条记录("10002103","大众文化研究","42005013","刘婧男",98)。

```
INSERT INTO score (课程编号,课程名称,学号,姓名,成绩);
```

```
VALUES ("10002103","大众文化研究"," 42005013","刘婧男",98)
```

等价于

```
INSERT INTO score;
VALUES ("10002103", "大众文化研究", "42005013", "刘婧男", 98)
```

例 4.63 向 teacher 表中插入一条记录("10230025", "朱杰", "讲师")。

```
INSERT INTO teacher (教师编号, 教师姓名, 职称);
VALUES ("10230025", "朱杰", "讲师")
```

说明：例 4.62 插入的是一条完整的记录，所以在 INSERT INTO score 后面可以不加字段名序列；而例 4.63 插入的记录不是一条完整的记录，所以在表名的后面必须要加上具体值所对应的字段名。

4.2.2 删除操作

在 Visual FoxPro 中 SQL-DELETE 语句可以为指定的数据表中的记录添加删除标记。

格式：DELETE FROM <表名>[WHERE<条件表达式>]

说明：

- 1) FROM 短语：表示指定从哪个表中删除数据。
- 2) WHERE 短语：表示指定被删除的记录所满足的条件，如果省略 WHERE 子句，则逻辑删除该表中的所有记录。
- 3) 在 Visual FoxPro 中 SQL-DELETE 命令是逻辑删除记录，如果要物理删除记录需要继续使用 PACK 命令。
- 4) 类似于 Visual FoxPro 中的 DELETE...FOR...命令。

例 4.64 使用 SQL 语句将 teacher 表中职称为副教授的教师信息删除。

```
DELETE FROM teacher WHERE 职称="副教授"
```

删除结果如图 4-64 所示。

与之等价的命令如下：

```
USE teacher
DELETE FOR 职称="副教授"
```

删除结果同上。

教师编号	教师姓名	性别	出生日期	所在院系	研究方向	职称	学位	学历
10230001	连秀君	女	05/25/71	文学院	汉语语法学	教授	博士	研究生
10230002	张建	男	10/01/68	文学院	中国古代美学	副教授	硕士	研究生
10230003	崔明材	男	01/05/75	文学院	唐宋文学	副教授	硕士	研究生
10230004	孙学成	男	04/30/72	文学院	中国现代文学	教授	硕士	研究生
10230005	战爱成	男	11/12/62	文学院	语言应用于写作	副教授	硕士	研究生
10230006	陈志伟	男	11/02/71	文学院	中国现代文学	副教授	硕士	研究生
10230007	周世明	男	06/05/73	英语学院	美国文学	教授	硕士	研究生
10230008	李爽	女	11/06/74	英语学院	美国文化	教授	硕士	研究生
10230009	程雅平	女	08/07/68	英语学院	语用学	副教授	博士	研究生
10230010	苗辰光	男	12/24/67	艺术学院	现代声乐的演唱与研究	副教授	硕士	研究生
10230011	王军	男	02/28/52	艺术学院	声乐表演艺术	教授	学士	本科
10230012	白强	男	01/15/72	艺术学院	西方当代美术研究	副教授	硕士	研究生
10230013	张文斌	男	12/07/72	数学学院	泛函分析	副教授	硕士	研究生
10230014	赵正	男	07/28/57	数学学院	再生核空间理解	教授	学士	本科
10230015	王锦	女	05/20/63	数学学院	泛函分析	副教授	博士	研究生
10230016	刘昭梅	女	07/14/64	计算机学院	数据库	教授	硕士	研究生
10230017	王宏旭	男	04/10/77	英语学院	英国文学	讲师	硕士	研究生
10230018	戚翌	女	09/08/79	英语学院	美国文学	助教	博士	研究生
10230019	林然	女	09/11/75	艺术学院	媒体艺术设计	助教	博士	研究生
10230020	宋军	男	10/18/66	数学学院	逼近论	讲师	博士	研究生
10230021	王立军	男	03/28/78	计算机学院	计算机网络	讲师	硕士	研究生

图 4-64 例 4.64 的删除结果

例 4.65 使用 SQL 语句将所授课程的学时小于 60 且考核方式为闭卷的教师信息逻辑删除。

```
DELETE FROM teacher;
WHERE 教师编号 IN;
(SELECT 教师编号 FROM course WHERE 学时<60 AND 考核方式="闭卷")
```

4.2.3 更新操作

在 Visual FoxPro 中，SQL 语言的更新操作是对表中的数据进行修改。

格式：UPDATE <表名>;

SET <字段名 1>=<表达式 1>[,<字段名 2>=<表达式 2>...];

[WHERE <条件表达式>]

说明：

- 1) 一般使用 WHERE 子句指定修改条件，如果省略 WHERE 子句，则更新表中的所有记录。
- 2) SQL 更新命令类似于 Visual FoxPro 中的 REPLACE...WITH...命令。

例 4.66 将 course 表中课程编号为“10002107”的课程的学分加 2 分。

```
UPDATE course SET 学分=学分+2 WHERE 课程编号="10002107"
```

与之等价的命令如下：

```
USE course
```

```
REPLACE ALL 学分 WITH 学分+2 FOR 课程编号=" 10002107"
```

例 4.67 将学分为 3 的所有课程的成绩提高 10%。

```
UPDATE score;
```

```
SET 成绩=成绩*1.1;
```

```
WHERE 课程编号 IN (SELECT 课程编号 FROM course WHERE 学分=3)
```

4.3 数据定义

标准 SQL 语言的数据定义功能非常广泛，一般包括数据库的定义、表的定义、视图的定义、规则的定义和索引的定义等若干部分。在本节将主要介绍 Visual FoxPro 支持的表定义功能和视图定义功能。

4.3.1 定义表

利用表设计器建立表的方法在第 3 章已经介绍了，在 Visual FoxPro 中不仅可以通过表设计器来建立表，也可以通过 SQL 语言中的 CREATE TABLE 命令建立表结构。具体格式如下：

```
CREATE TABLE|DBF <表名 1>[NAME <长表名>][FREE];
```

```
( <字段名 1> <字段类型> [(字段宽度)][NULL|NOT NULL];
```

```
[CHECK<逻辑表达式 1>[ERROR <提示信息>]];
```

```
[DEFAULT <表达式 1>];
```

```
[PRIMARY KEY | UNIQUE];
```

```
[REFERENCES <表名 2> [TAG<索引名 1>]];
```

```
[,<字段名 2> <字段类型> [(字段宽度)][NULL|NOT NULL].....];
```

```
[,CHECK <逻辑表达式 2>[ERROR <提示信息 2>]]
```

```
[,PRIMARY KEY <表达式 2> TAG <索引名 2>
```

```
],UNIQUE <表达式 3> TAG <索引名 3>]
```

```
[,FOREIGN KEY <表达式 4> TAG <索引名 4> REFERENCES <表名 3>[TAG <索引名 5>]]
```

```
)
```

说明：

1) 从以上建立表的基本格式可以看出，用 CREATE TABLE 命令建立表可以完成用第 3 章介绍的表设计器能够完成的所有功能。

2) 具体命令短语功能如表 4-3 所示。

表 4-3 命令短语功能

命令短语	功能
NAME 长表名	为建立的表指定一个长名
FREE	建立的表不添加到当前数据库中，即建立一个自由表
NULL 或 NOT NULL	说明字段允许或不允许为空值
CHECK	用来定义域完整性
ERROR	用来定义出错信息
DEFAULT	用来定义默认值
PRIMARY KEY	用来定义主关键字（主索引）
UNIQUE	用来定义候选索引
FOREIGN KEY ...REFERENCES...	用来定义表之间的联系

3) 在 CREATE TABLE 命令中可以使用的字段类型及说明如表 4-4 所示。

表 4-4 表中常用的字段类型

字段类型	字段宽度	小数位	说明
C	n	-	字符型字段的宽度为 n
D	默认 (8)	-	日期类型 (Date)
T	默认 (8)	-	日期时间类型 (DateTime)
N	n	D	数值字段类型，宽度为 n，小数位为 d (Numeric)
F	n	d	浮点数值字段类型，宽度为 n，小数位为 d (Float)
I	默认 (4)	-	整数类型 (Integer)
B	默认 (8)	d	双精度类型 (Double)
Y	默认 (8)	-	货币类型 (Currency)
L	1	-	逻辑类型 (Logical)
M	4	-	备注类型 (Memo)
G	4	-	通用类型 (General)

本节通过创建“学生数据库”及其中的表(如图 4-65 所示)来学习使用 SQL CREATE TABLE 命令建立表。表结构如下：

学生表(学号 C(4), 姓名 C(8), 性别 L, 年龄 I, 入学时间 D)

成绩表(学号 C(4), 课程编号 C(2), 成绩 N(6, 2))

课程表(课程编号 C(2), 课程名称 C(20), 学分 I, 教师编号 C(3))

教师表(教师编号 C(3), 教师姓名 C(8), 职称 C(8), 工资 I)

例 4.68 用命令创建“学生数据库”，并用 SQL CREATE 命令建立“学生表”。

```
CREATE DATABASE 学生数据库
```

```
CREATE TABLE 学生表(;
```

```
学号 C(4) PRIMARY KEY ,;
```

```
姓名 C(8),;
```

```
性别 L ,;
```

```
年龄 I,;
```

```
入学时间 D)
```

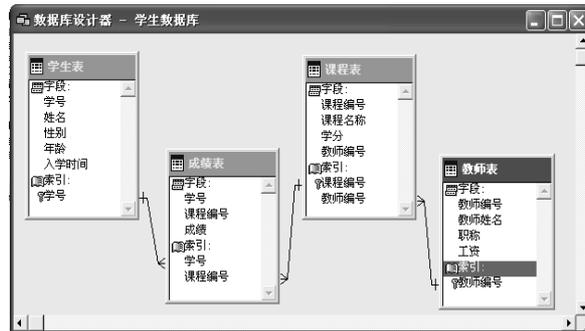


图 4-65 学生数据库

例 4.69 用命令创建“教师表”。

```
CREATE TABLE 教师表(
  教师编号 C(3) PRIMARY KEY,
  教师姓名 C(8),
  职称 C(8),
  工资 I CHECK(工资>1000 AND 工资<=6000);
  ERROR "工资的范围在 1000~6000!";
  DEFAULT 1500)
```

例 4.70 用命令创建“课程表”。

```
CREATE TABLE 课程表(
  课程编号 C(2) PRIMARY KEY,
  课程名称 C(20),
  学分 I,
  教师编号 C(3),
  FOREIGN KEY 教师编号 TAG 教师编号 REFERENCES 教师表)
```

例 4.71 用命令创建“成绩表”。

```
CREATE TABLE 成绩表(
  学号 C(4),
  课程编号 C(2),
  成绩 N(6,2) CHECK(成绩>=0 AND 成绩<=100);
  ERROR "成绩应该在 0 至 100 之间";
  DEFAULT 0,
  FOREIGN KEY 学号 TAG 学号 REFERENCES 学生表,
  FOREIGN KEY 课程号 TAG 课程号 REFERENCES 课程表)
```

说明:

- 1) 在学生表中用 PRIMARY KEY 命令说明“学号”为主关键字。
- 2) 在教师表中用 CHECK 说明了“工资”字段的有效性规则为“工资>1000 AND 工资<=6000”。

用 ERROR 说明了当输入出错时的提示信息为“工资的范围在 1000~6000!”;

用 DEFAULT 说明了当没有输入教师工资时默认显示为 1500。

- 3) 在成绩表中用 CHECK 说明了“成绩”字段的有效性规则;用 ERROR 说明了出错信息;还用 default 命令说明了“成绩”字段的默认值为 0。

4) 利用短语“FOREIGN KEY 学号”和“FOREIGN KEY 课程号”在成绩表中建立两个普通索引,作为连接字段,并通过引用学生表的主索引“学号”(TAG 学号 REFERENCES 学生表)建立了成绩表与学生表之间的联系;通过引用课程表的主索引“课程号”(TAG 课程

号 REFERENCES 课程表)建立了成绩表与课程表之间的联系。

4.3.2 删除表

删除表的 SQL 命令如下。

格式: DROP TABLE <表名>

功能: 直接从磁盘上删除指定的表所对应的 DBF 文件。

说明: 如果要删除的表是数据库中的表并且相应的数据库是当前数据库, 则从数据库中删除了表; 否则虽然从磁盘上删除了 DBF 文件, 但是记录在数据库 DBC 文件中的信息却没有删除, 此后会出现错误提示。所以要删除数据库中的表时, 最好使用的数据是当前打开的数据库, 在数据库中进行操作。

例 4.72 删除上例中所建的表“课程表”。

```
DROP TABLE 课程表
```

4.3.3 修改表结构

修改表结构的命令是 ALTER TABLE, 它包括添加 (ADD)、修改 (ALTER)、删除 (DROP) 表中的字段、字段类型、字段宽度、字段有效性规则及索引等。修改表结构的命令有 3 种格式。

格式 1: ALTER TABLE <表名 1> ADD|ALTER ;

[COLUMN] <字段名 1> <字段类型>[(字段宽度)][NULL|NOT NULL]

[CHECK <逻辑表达式>][ERROR<提示信息>][DEFAULT<表达式>]

[PRIMARY KEY|UNIQUE]

[REFERENCES <表名 2>[TAG <索引名>]]

说明: 此格式可添加 (ADD) 新的字段或修改 (ALTER) 已有的字段, 它的句法基本与 CREATE TABLE 的句法相对应。

例 4.73 为 score 表增加一个长度为 2 的整型的学分字段, 要求学分有效性规则为大于等于 0, 默认值为 2, 提示信息为“学分应该大于 0”。

```
ALTER TABLE score ADD 学分 I CHECK 学分>=0 ERROR "
学分应该大于 0" DEFAULT 2
```

结果如图 4-66 所示。

例 4.74 将 student 表中的家庭住址的长度由 8 改为 20。

```
ALTER TABLE student ALTER 家庭住址 C(20)
```

说明:

1) 从以上两个例题可以看出, 该格式可以修改字段的类型、宽度、有效性规则、错误信息、默认值。另外, 还可以定义主关键字和联系等。

2) 但是用“ALTER TABLE ... ADD|ALTER ...”命令格式不能修改字段名, 不能删除字段, 也不能删除已经定义的有效性规则。

格式 2: ALTER TABLE <表名> ALTER [COLUMN] <字段名 1> [NULL|NOT NULL]

[SET DEFAULT]

[SET CHECK <逻辑表达式>][ERROR<提示信息>]

[DROP DEFAULT]

[DROP CHECK]



学号	课程编号	成绩	学分
200731001	02	91.00	2
200731001	05	88.00	2
200731001	03	93.50	2
200731004	05	95.00	2
200731005	01	94.00	2
200731005	03	83.60	2
200731006	01	83.00	2
200731006	04	99.00	2
200731007	02	80.00	2
200731008	03	104.50	2
200731009	04	90.20	2

图 4-66 例 4.73 的添加结果

说明:

1) SET DEFAULT: 用于设置默认值。

2) SET CHECK <逻辑表达式>[ERROR<提示信息>]: 用于设置有效性规则和错误提示信息。

3) DROP DEFAULT: 用于删除已经存在的默认值。

4) DROP CHECK: 用于删除已经存在的有效性规则。

例 4.75 使用 SQL 命令为 score 表中“成绩”字段增加有效性规则: 成绩在 0~100 之间且默认值为 60。

```
ALTER TABLE score ALTER 成绩 ;
SET DEFAULT 60 ;
SET CHECK 成绩>=0 AND 成绩<=100 ERROR"成绩应在 0~100 之间!"
```

例 4.76 使用 SQL 命令删除 score 表中成绩字段的有效性规则。

```
ALTER TABLE score ALTER 成绩 DROP DEFAULT DROP CHECK
```

以上两种格式都不能删除字段, 也不能更改字段名, 所有修改是在字段一级。为了补充这个功能, 引出第三种格式。

格式 3: ALTER TABLE <表名 1> [DROP [COLUMN]<字段名>]
 [SET CHECK <表达式 >ERROR<提示信息>]
 [DROP CHECK]
 [ADD PRIMARY KEY <表达式 >TAG <索引名>[FOR<表达式>]]
 [DROP PRIMARY KEY]
 [ADD UNIQUE <表达式>[TAG <索引名>[FOR<表达式>]]]
 [DROP UNIQUE <表达式>]
 [ADD FOREIGN KEY <表达式> TAG <索引名>[FOR<表达式>]]
 REFERENCES <表名 2>[TAG <索引名>]
 [DROP FOREIGN KEY TAG <索引名>[SAVE]]
 [RENAME [COLUMN] <字段名> TO <新字段名>]

说明:

1) 此格式可删除字段, 可修改字段名; 可定义、修改和删除表一级的有效性规则等。

2) DROP [COLUMN]字段名: 删除指定的字段。

3) SET CHECK <表达式> ERROR<提示信息>: 定义表一级的有效性规则。

4) DROP CHECK: 删除表一级的有效性规则。

5) ADD PRIMARY KEY <表达式>TAG <索引名>[FOR<表达式>]: 定义一个主索引项。

6) DROP PRIMARY KEY: 删除主索引。

7) ADD NUIQUE <表达式>[TAG <索引名> [FOR<表达式>]]: 定义一个候选索引。

8) DROP UNIQUE <表达式>: 删除候选索引。

9) ADD FOREIGN KEY <表达式 >TAG <索引名>[FOR<表达式>]: 建立表之间的关系。

10) RENAME [COLUMN] <字段名> TO <新字段名>: 修改字段的名称。

例 4.77 使用 SQL 命令将 course 表中的“教师编号”字段名改为“授课教师”。

```
ALTER TABLE course RENAME 教师编号 TO 授课教师
```

例 4.78 使用 SQL 命令删除 score 表中的“学分”字段。

```
ALTER TABLE score DROP 学分
```

例 4.79 使用 SQL 命令将 course 表的“课程名称”和“教师编号”定义为候选索引(候

选关键字), 索引名是 `htt_sp`。

```
ALTER TABLE score ADD NUIQUE 课程名称+教师编号 TAG htt_sp
```

例 4.80 使用 SQL 命令将 `teacher` 表的主索引删除。

```
ALTER TABLE teacher DROP PRIMARY KEY
```

4.3.4 视图

1. 视图的概念及其定义

Visual FoxPro 中视图是一个虚拟的表, 可以把它看作是从表中派生出来的虚表。视图可引用一个或多个表, 或者引用其他视图。视图依赖于表, 不独立存在。在关系数据库中, 视图也称为窗口, 即视图是操作表的窗口。

使用 SQL 命令定义视图的基本格式如下:

```
CREATE VIEW <视图名>[(<字段名 1>[,<字段名 2>]...)];
```

```
AS SELECT 查询语句
```

说明:

1) SELECT 查询语句: 可以是任意的 SELECT 查询语句, 由它说明该视图包含哪些数据。

2)[(<字段名 1>[,<字段名 2>]...)]: 没有为视图指定字段名时, 视图的字段名将与 SELECT 查询中所指定的字段名同名。

3) 视图是根据表定义或派生出来的, 所以在涉及视图时, 常把表称为基本表。

4) 视图通常保存在数据库中, 在磁盘上找不到视图文件。

例 4.81 使用 SQL 命令创建一个学生视图, 视图名为 `student_view`。

```
CREATE VIEW student_VIEW AS SELECT * FROM student
```

说明:

1) `student_view` 是视图的名称, 视图中存放的内容为 AS 后面的 SELECT 查询语句的查询结果。

2) 视图一经定义, 就可以和基本表一样进行各种查询, 也可以进行一些修改操作。对于最终用户来说, 有时并不需要知道操作的是基本表还是视图。如下例所示:

```
SELECT * FROM student_VIEW
```

等价于

```
SELECT * FROM student
```

例 4.82 使用 SQL 命令建立一个视图 `xuesheng_view`, 要求该视图包含 1980 年 1 月 1 日之后出生的学生的学号、姓名、平均成绩。

```
SELECT student.学号, student.姓名,AVG(成绩) AS 平均成绩;
```

```
FROM student,score;
```

```
WHERE student.学号 = score.学号 AND 出生日期>{^1980-01-01};
```

```
GROUP BY score.学号
```

2. 视图的删除

视图由于是从表派生而来的, 所以不存在修改结构的问题, 但是视图可以删除。

格式: `DROP VIEW <视图名>`

例 4.83 使用 SQL 命令删除视图 `xuesheng_view`。

```
DROP VIEW xuesheng_view
```

注意: 在关系数据库中, 视图始终不真正含有数据, 它总是原有表的一个窗口。所以, 虽然对视图可以像表一样进行各种查询, 但是插入、更新和删除操作在视图上却有一定限制。

在一般情况下,当一个视图是由单个表导出时可以进行插入和更新操作,但不能进行删除操作;当视图是从多个表导出时,插入、更新和删除操作都不允许进行。

习题四

一、选择题

- 在 SQL 的 ALTER TABLE 语句中,为了增加一个新的字段应该使用短语 ()。
 - CREATE
 - APPEND
 - COLUMN
 - ADD
- 查询 1978 年 10 月 27 日出生的职工信息的正确命令是 ()。
 - SELECT * FROM 职工 WHERE 出生日期={^1978-10-27}
 - SELECT * FROM 职工 WHERE 出生日期=1978-10-27
 - SELECT * FROM 职工 WHERE 出生日期="^1978-10-27"
 - SELECT * FROM 职工 WHERE 出生日期=("1978-10-27")
- 有以下 SQL SELECT 语句


```
SELECT * FROM stock;
WHERE 单价 BETWEEN 15.76 AND 19.20
```

 与该语句等价的是 ()。
 - SELECT*FROM stock WHERE 单价<=19.20.AND.单价>=15.76
 - SELECT*FROM stock WHERE 单价<19.20.AND.单价>15.76
 - SELECT*FROM stock WHERE 单价>=19.20.AND.单价<=15.76
 - SELECT*FROM stock WHERE 单价>19.20.AND.单价<15.76
- 设有关系 SC(SNO,CNO,GRADE),其中 SNO、CNO 分别表示学号、课程号(两者均为字符型),GRADE 表示成绩(数值型),若要把学号为“S101”的同学,选修课程号为“C11”,成绩为 98 分的记录插到表 SC 中,正确的语句是 ()。
 - INSERT INTO SC(SNO,CNO,GRADE)VALUES('S101 ','C11','98')
 - INSERT INTO SC(SNO,CNO,GRADE)VALUES(S101,C11,98)
 - INSERT ('S101','C11','98') INTO SC
 - INSERT INTO SC VALUES ('S101','C11',98)
- SQL 的 SELECT 查询语句中的特殊运算符不包括 ()。
 - BETWEEN
 - IN
 - OR
 - LIKE
- 下列关于 SQL 的并运算,说法不正确的是 ()。
 - 集合的差运算,即 UNION,是指将两个 SELECT 语句的查询结果通过合并运算合并成一个查询结果
 - 集合的并运算,即 UNION,是指将三个以上的 SELECT 语句的查询结果通过合并运算合并成一个查询结果
 - 进行并运算要求两上查询结果具有相同的字段的数据,并且对应的字段的值要出自同一个域
 - 两个查询结果要具有相同的数据类型和取值范围
- 使用 SQL 语句从 student 表中查询所有姓孙的学生信息,正确的命令是 ()。
 - SELECT * FROM student WHERE STR(姓名,2)="孙"

- B. SELECT * FROM student WHERE TRIM(姓名,2)="孙"
 C. SELECT * FROM student WHERE RIGHT(姓名,2)="孙"
 D. SELECT * FROM student WHERE LEFT(姓名,2)="孙"

8. 关于 SQL 嵌套查询的说法正确的是 ()。

- A. 能对外层查询排序, 又能对内层查询排序
 B. 能对外层查询排序, 不能对内层查询排序
 C. 不能对外层查询排序, 只能对内层查询排序
 D. 既不能对外层查询排序, 也不能对内层查询排序

9. 在 Visual FoxPro 中, 假设教师表 T(教师号, 姓名, 性别, 职称, 研究生导师)中, 性别是 C 型字段, 研究生导师是 L 型字段。若要查询“是研究生导师的女老师”信息, 那么 SQL 语句“SELECT * FROM T WHERE <逻辑表达式>”中的<逻辑表达式>应是 ()。

- A. 研究生导师 AND 性别="女" B. 研究生导师 OR 性别="女"
 C. 性别="女" AND 研究生导师=F. D. 研究生导师=T. OR 性别=女

第 10 到第 14 题基于学生表 S 和学生选课表 SC 两个数据库表, 他们的结构如下:

S(学号,姓名,性别,年龄), 其中学号、姓名和性别为 C 型字段, 年龄为 N 型字段。

SC(学号,课程号,成绩), 其中学号和课程号为 C 型字段, 成绩为 N 型字段 (初始为空值)。

10. 查询学生选修课程成绩小于 60 分的学号, 正确的 SQL 语句是 ()。

- A. SELECT DISTINCT 学号 FROM SC WHERE "成绩" <60
 B. SELECT DISTINCT 学号 FROM SC WHERE 成绩 < "60"
 C. SELECT DISTINCT 学号 FROM SC WHERE 成绩 <60
 D. SELECT DISTINCT "学号" FROM SC WHERE "成绩" <60

11. 查询学生表 S 的全部记录并存储于临时表文件 one 中的 SQL 命令是 ()。

- A. SELECT * FROM 学生表 INTO CURSOR one
 B. SELECT * FROM 学生表 TO CURSOR one
 C. SELECT * FROM 学生表 INTO CURSOR DBF one
 D. SELECT * FROM 学生表 TO CURSOR DBF one

12. 查询成绩在 70 分至 85 分之间学生的学号、课程号和成绩, 正确的 SQL 语句是 ()。

- A. SELECT 学号,课程号,成绩 FROM sc WHERE 成绩 BETWEEN 70 AND 85
 B. SELECT 学号,课程号,成绩 FROM sc WHERE 成绩 >=70 OR 成绩 <=85
 C. SELECT 学号,课程号,成绩 FROM sc WHERE 成绩 >=70 OR <=85
 D. SELECT 学号,课程号,成绩 FROM sc WHERE 成绩 >=70 AND <=85

13. 查询有选课记录,但没有考试成绩的学生的学号和课程号,正确的 SQL 语句是()。

- A. SELECT 学号,课程号 FROM sc WHERE 成绩 = ""
 B. SELECT 学号,课程号 FROM sc WHERE 成绩 = NULL
 C. SELECT 学号,课程号 FROM sc WHERE 成绩 IS NULL
 D. SELECT 学号,课程号 FROM sc WHERE 成绩

14. 查询选修 C2 课程号的学生姓名, 下列 SQL 语句中错误的是 ()。

- A. SELECT 姓名 FROM S WHERE EXISTS;
 (SELECT * FROM SC WHERE 学号=S.学号 AND 课程号='C2')
 B. SELECT 姓名 FROM S WHERE 学号 IN;
 (SELECT * FROM SC WHERE 课程号='C2')

- C. SELECT 姓名 FROM S JOIN ON S.学号=SC.学号 WHERE 课程号='C2'
- D. SELECT 姓名 FROM S WHERE 学号=;
(SELECT * FROM SC WHERE 课程号='C2')
15. 使用 SQL 语句将“李平”的工资增加 300 元的语句是 ()。
- A. REPLACE 教师 WITH 工资=工资+300 WHERE 姓名="李平"
- B. UPDATE 教师 SET 工资=工资+300 WHEN 姓名="李平"
- C. UPDATE 教师 工资 WITH 工资+300 WHERE 姓名="李平"
- D. UPDATE 教师 SET 工资=工资+300 WHERE 姓名="李平"

二、填空题

1. 使用 SQL 语言的 SELECT 语句进行分组查询时, 如果希望去掉不满足条件的分组, 应当在 GROUP BY 用_____子句。
2. Visual FoxPro 中用于计算检索的函数有 5 个, _____用于计数, _____用于求和, _____用于求平均值, _____用于求最大值, _____用于求最小值。
3. 设有 s(学号, 姓名, 性别)和 sc(学号, 课程号, 成绩)两个表, 下面 SQL 的 SELECT 语句检索选修的每门课程的成绩都高于或等于 85 分的学生的学号、姓名和性别。
SELECT 学号, 姓名, 性别 FROM s;
WHERE _____(SELECT * FROM sc WHERE sc.学号 = s.学号 AND 成绩 < 85)
4. 在 Visual FoxPro 中, 使用 SQL 的 SELECT 语句将查询结果存放到数组中的短语是_____; 将查询结果存放到临时表中的短语是_____; 将查询结果存放到表中的短语是_____。
5. 下面各题使用以下的“教师”表和“学院”表。

“教师”表:

职工号	姓名	职称	年龄	工资	系号
11020001	肖天海	副教授	35	2000.00	01
11020002	王岩盐	教授	40	3000.00	02
11020003	刘星魂	讲师	25	1500.00	01
11020004	张月新	讲师	30	1500.00	03
11020005	李明玉	教授	34	2000.00	01
11020006	孙民山	教授	47	2100.00	02
11020007	钱无名	教授	49	2200.00	03

“学院”表:

系号	系名
01	英语
02	会计
03	工商管理

- (1) 使用 SQL 语句将一条新的记录插入学院表。
INSERT _____学院(系号, 系名) _____("04", "计算机")
- (2) 使用 SQL 语句求“工商管理”系的所有职工的工资总和。
SELECT _____(工资) FROM 教师;
WHERE 系号 IN (SELECT 系号 FROM _____ WHERE 系名="工商管理")
- (3) 使用 SQL 语句完成如下操作 (将所有教授的工资提高 5%)。

_____教师 SET 工资=工资*1.05_____职称="教授"

6. 在 Visual FoxPro 中, 使用 SQL 语言的 ALTER TABLE 命令给学生表 student 增加一个 Email 字段, 长度为 30, 命令是 (关键字必须拼写完整)。

ALTER TABLE student_____Email C(30)

7. 在 SQL SELECT 语句中为了将查询结果存储到临时表中, 应该使用_____短语。

8. 在 Visual FoxPro 中, 使用 SQL 的 CREATE TABLE 语句建立数据库表时, 使用_____子句说明主索引。

9. SELECT * FROM student_____FILE student 命令将查询结果存储在 student.txt 文本文件中。

10. 设有 SC(学号,课程号,成绩)表, 下面 SQL 的 SELECT 语句检索成绩高于或等于平均成绩的学生的学号。

```
SELECT 学号 FROM sc;  
WHERE 成绩>=(SELECT _____FROM sc)
```