

第3章 结构化查询语言



- 了解 SQL 语言的功能。
- 理解 SQL 的表定义功能。
- 掌握 SQL 的基本查询以及多表联接查询和嵌套查询的相关功能。

在前面章节中学习了如何利用 Visual FoxPro 的菜单以及命令交互方式进行数据库和表的相关操作。这种操作方式方便了用户，但是隐藏了具体的实现过程。要想实现更复杂的数据库操作，设计多样化、高效的数据查询，则有必要了解其内部的具体实现，即 SQL (Structured Query Language，结构化查询语言)。

SQL 语言是用户操作关系数据库的通用语言，几乎所有的关系型数据库管理系统都支持 SQL 语言，Visual FoxPro 也不例外，SQL 已然成为业界的标准。

SQL 语言主要包括数据定义、数据操纵和数据查询、数据控制等功能，本章将从数据定义、数据操纵和数据查询三个方面介绍 Visual FoxPro 所支持的 SQL 语言。

3.1 SQL 概述

SQL 最早是由美国国家标准协会 (American National Standards Institute, ANSI) 在 1986 年 10 月公布的。国际标准化组织 (International Organization for Standardization, ISO) 于 1987 年 6 月正式采纳它为国际标准，并在此基础上进行补充，于 1989 年 4 月提出了具有完整性特征的 SQL-89。随后又出现了 SQL2 (SQL-92) 和 SQL3 (SQL-99)。

SQL 语言主要具有以下几个特点：

(1) 综合统一。

SQL 语言风格统一，可以独立完成数据库生命周期中的全部活动，包括建立数据库、定义表结构、插入数据、查询、更新、维护、数据库重构、数据库安全性控制等一系列操作，这就为数据库应用系统的开发提供了良好的环境。

(2) 高度非过程化。

SQL 是一种非过程化语言，用其进行数据操作，只需要提出“做什么”，而无须指明“怎么做”，因此无须了解存取路径，只要指明所需的数据即可，具体的 SQL 语句操作过程由系统自动完成。这不但大大减轻了用户负担，降低了出错率，而且有利于提高数据的独立性。

(3) 面向集合的操作方式。

SQL 是一种面向集合的语言，与面向记录的方式截然不同，每个命令的操作对象是一个或多个表，结果也是一个表。这种采用集合的操作方式不仅操作对象、查找结果可以是记录的集合，而且一次添加、删除、更新操作的对象也可以是记录的集合。

(4) 同一种语法结构，两种使用方式。

SQL 语言既是自含式语言，又是嵌入式语言，可独立使用也可嵌入到宿主语言中。作为自含式语言它能够独立地用于联机交互方式，即在终端机上进行数据库操作，这适用于终端用户、应用程序员和 DBA。作为嵌入式语言，SQL 语句能够嵌入到高级语言（如 C、COBOL、FORTRAN、PL/1）程序中，供开发使用。在这两种方式下，SQL 语法结构基本一致，为用户提供了极大的灵活性和方便性。

(5) 语言简洁易用。

SQL 语言设计巧妙，以简洁的语句实现强大的功能，9 个动词即可完成核心功能，如表 3.1 所示。另外，语言本身接近英语自然语言，易学易懂。

表 3.1 SQL 语言的动词

SQL 功能	动词
数据查询	SELECT
数据定义	CREATE、DROP、ALTER
数据操纵	INSERT、UPDATE、DELETE
数据控制	GRANT、REVOKE



Visual FoxPro 所支持的 SQL 语言与标准 SQL 语言的区别：用于 Visual FoxPro 的 SQL 语言与标准 SQL 语言的命令格式基本一致，但是 Visual FoxPro 的 SQL 语言只支持数据查询、数据操纵和数据定义功能，没有数据控制功能。

3.2 数据定义功能

完整的 SQL 语言数据定义功能包括数据库定义、表定义、视图定义、存储过程定义、触发器定义和规则定义等。本节主要介绍 Visual FoxPro 所支持的 SQL 语言表定义功能：创建表、修改表结构和撤消表的操作。

3.2.1 创建表——CREATE TABLE

在 Visual FoxPro 中，除了可以采用“表设计器”创建表之外，用户还可以通过 SQL 语言的 CREATE TABLE 命令建立表的结构。

命令格式：CREATE TABLE|DBF<表名>

(<字段名 1><类型>[<字段宽度>[,<小数位数>]])[,<字段名 2><类型>
[(<字段宽度>[,<小数位数>])]....)

说明：其中表名是所要建立的基本表名，该表可以由一个或多个字段（属性）组成。

功能：建立表结构。

【例 3.1】用 SQL 语句创建学生表 XS.DBF。

```
CREATE TABLE XS (学号 C(10),姓名 C(8),性别 C(2),出生日期 D,籍贯 C(12),民族 C(20),专业 C(20),班级 C(20),院系代码 C(10),是否党员 L,个人简历 M,照片 G)
```

【例 3.2】用 SQL 语句创建课程表 KC.DBF。

```
CREATE TABLE KC (课程号 C(4),课程名 C(20),学时数 N(3,0))
```

3.2.2 修改表——ALTER TABLE

修改表结构，主要有如下 3 种命令格式：

命令格式 1: ALTER TABLE <表名> ADD |ALTER [COLUMN]<字段名><类型>
[($\langle$ 宽度 $\rangle$ [, $\langle$ 小数位 $\rangle$]])

功能：用于添加或修改字段，ADD 用于添加，ALTER 用于修改字段。

【例 3.3】用 SQL 语句向学生表 XS.DBF 中添加家庭住址字段。

```
ALTER TABLE XS ADD 家庭住址 C(30)
```

说明：该命令也可以写成“ALTER TABLE XS ADD COLUMN 家庭住址 C(30)”，关键字 COLUMN 是可选项，格式 2 和格式 3 请参照处理。

命令格式 2: ALTER TABLE<表名>DROP[COLUMN]<字段名>

功能：用于删除字段。

【例 3.4】用 SQL 语句删除 XS.DBF 表中的“家庭住址”字段。

```
ALTER TABLE XS DROP 家庭住址
```

命令格式 3: ALTER TABLE<表名>RENAME [COLUMN]<字段名 1>TO<字段名 2>

功能：用于更改字段名称。

【例 3.5】用 SQL 语句将 XS.DBF 表中的“班级”字段改名为 CLASS。

```
ALTER TABLE XS RENAME 班级 to CLASS
```

反之，将 CLASS 字段改回原名“班级”，相应的语句为：

```
ALTER TABLE XS RENAME CLASS TO 班级
```

3.2.3 撤消表——DROP TABLE

撤消表是从磁盘上删除指定的表。

命令格式: DROP TABLE <表名>

【例 3.6】撤消院系表 YX.DBF。

```
DROP TABLE YX
```

3.3 数据操纵功能

SQL 的数据操纵功能主要包含表（或视图）中数据的添加（INSERT）、更新（UPDATE）和删除（DELETE）功能，下面逐一进行介绍。

3.3.1 数据添加——INSERT

INSERT 命令的主要功能是，向表中添加记录，实现该功能的 SQL 语句命令有两种。

命令格式 1: INSERT INTO <表名> [($\langle$ 字段名 1 $\rangle$ [, $\langle$ 字段名 2 $\rangle$...])]
VALUES($\langle$ 表达式 1 $\rangle$ [, $\langle$ 表达式 2 $\rangle$...])

功能：在指定表的末尾添加一条新记录。新记录字段 1 的值为表达式 1，字段 2 的值为表

达式 2,……。各表达式的值必须与对应字段的数据类型一致。如果所添加的新记录全部字段均赋值,则可以省略字段名列表。

命令格式 2: INSERT INTO <表名> FROM ARRAY 数组名|FROM MEMVAR

功能: 由指定数组或内存变量的值在指定表的尾部添加一条新记录。

【例 3.7】向成绩表 CJ.DBF 中添加一条记录。

```
INSERT INTO CJ VALUES("09714046","1006",76)
```

【例 3.8】向课程表 KC.DBF 中添加一条记录。

```
INSERT INTO KC (课程号,课程名) VALUES ("1008","汇编语言",72)
```

3.3.2 数据更新——UPDATE

UPDATE 命令主要是更新表中的记录数据。

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>]

功能: 更新满足条件的记录中指定字段的属性值。其中, WHERE 子句指定要更新的记录, 缺省时表示针对全部记录。

【例 3.9】将课程表 KC.DBF 中“汇编语言”课程的课时数修改为 81 课时。

```
UPDATE KC SET 学时数=81 WHERE 课程名="汇编语言"
```

【例 3.10】将学生表 XS.DBF 中牛芳菲的出生日期改为 1990 年 2 月 3 日, 专业改为中医。

```
UPDATE XS SET 出生日期={^1990-02-03},专业="中医" WHERE 姓名="牛芳菲"
```

3.3.3 数据删除——DELETE

DELETE 语句是对表中的记录进行删除。

格式: DELETE FROM<表名>[WHERE<条件>]

功能: 对指定表中符合条件的记录进行逻辑删除, 缺省 WHERE 子句时表示删除所有记录。

【例 3.11】逻辑删除 CJ.DBF 表中 1002 号课程的成绩记录。

```
DELETE FROM CJ WHERE 课程号="1002"
```

【例 3.12】逻辑删除 XS.DBF 表中的所有男生记录。

```
DELETE FROM XS WHERE 性别="男"
```

3.4 数据查询功能

数据查询是数据库的核心操作。SQL 语言具有强大的数据查询功能, 能够进行单表查询、多表联接查询和嵌套查询。数据查询使用 SELECT-SQL 查询命令。

3.4.1 SELECT-SQL 语句格式

命令格式: SELECT [ALL | DISTINCT] [<表别名>.<表达式 1>[AS<列标题 1>]
[,<表别名>.<表达式 2>[AS<列标题 2>]...]
FROM [<数据库名 1>!<表名 1>|<视图 1> [<本地别名 1>]
[[INNER | LEFT[OUTER] | RIGHT[OUTER] | FULL[OUTER]

```

JOIN  [, <数据库名 2>!<表名 2>| <视图 2> [<本地别名 2>]
ON<联接条件>...]
[[INTO<目标>]
[TO FILE<文件名>[ADDITIVE] | TO PRINTER [PROMPT] | TO SCREEN]]
[WHERE<条件表达式>]
[GROUP BY <分组表达式>[, <分组表达式>...]]
[HAVING<筛选条件>]
[UNION [ALL]<SELECT 命令>]
[ORDER BY <关键字表达式>[ASC | DESC][, <关键字表达式>[ASC | DESC]...]]
[TOP<数值表达式>[PERCENT]]

```

整个 SELECT-SQL 语句的含义是, 根据 WHERE 子句的条件表达式从 FROM 子句指定的表或视图找出满足条件的元组, 再按照 SELECT 子句中的目标列表达式选出元组中的对应字段列表, 形成输出结果。整个语句命令格式看起来非常复杂, 不容易理解, 但只要分解命令格式, 理解其中各个子句的功能和用法, 再综合应用就很容易掌握了。

说明:

(1) SQL 查询的基本结构是 SELECT...FROM...WHERE, 它包含输出字段...数据来源...查询条件。在各种查询语句中, SELECT 子句和 FROM 子句是必选项。

(2) SELECT 子句: 用来确定要在查询结果中显示的字段或表达式, ALL 表示选出的记录中包括重复记录, 这是系统的默认值; DISTINCT 则表示选出的记录中若有重复记录, 则只保留一条。

(3) FROM 子句及其选项: 用于指定要查询的表或视图(视图也是表, 本章凡是对“表”的操作均适用于“视图”)与联接类型。对于非当前数据库中的表可用“数据库名!表名”加以说明。未给表取别名时, 表别名为原表名。在执行查询时, 要查询的表不必先打开。

- JOIN 关键字: 用于联接左右两个<表名>所指定的表或视图。
- INNER | LEFT[OUTER] | RIGHT[OUTER] | FULL[OUTER] 选项: 指定两表联接时的联接类型是内联接、左联接、右联接还是全联接。
- ON<联接条件>子句: 指定联接条件。

(4) INTO 与 TO 子句: 用于指定查询结果的输出去向。系统默认的查询结果显示在“浏览”窗口中。

(5) WHERE 子句: 指定筛选条件或联接条件, 表示在要查询的表中按指定条件筛选符合条件的记录。若已用 JOIN ON 子句指定了联接条件, 则在 WHERE 子句中只能指定筛选条件; 若省略 JOIN ON 子句, 则表示筛选条件和联接条件。

(6) GROUP BY 子句: 对表中的记录按<分组表达式>值分组, 常用于分组统计查询。

(7) HAVING 子句: 当含有 GROUP BY 子句时, HAVING 子句可用作分组记录查询的限制条件; 无 GROUP BY 子句时, HAVING 子句的作用同 WHERE 子句。

(8) UNION 子句: 用于求两个查询结果的并集, 通常是在一个 SELECT-SQL 命令中用 UNION 子句嵌入另一个 SELECT-SQL 命令, 要求两个查询的输出字段类型和宽度必须一致。

(9) ORDER BY 子句: 指定查询结果中记录按<关键字表达式>排序, ASC 是按升序排列, DESC 是按降序排列, 系统默认的是升序 ASC。

(10) TOP 子句: 取查询结果集中的前几条或前百分之几的记录。



关于查询命令书写的说明：在 Visual FoxPro 中，一条查询语句结束时按回车键，系统会自动执行查询。在整个查询语句结束前要连续写，若要换行写，需要使用“;”作为连接符，不能回车，否则系统自动运行未写完的查询语句，可能导致系统报错。

SELECT-SQL 命令可以实现各种各样的查询，下面将通过大量的实例来介绍 SELECT-SQL 命令的使用，示例将用到 XS.DBF、CJ.DBF 和 KC.DBF 三张表，请参见 2.1 节。

3.4.2 简单查询

1. 字段的选取

【例 3.13】查询 XS.DBF 表中所有学生的学号、姓名、籍贯和专业。

```
SELECT 学号,姓名,籍贯,专业 FROM XS
```

查询结果如图 3-1 所示。

学号	姓名	籍贯	专业
10811023	万安宁	南昌	对外汉语
09311047	王维胜	合肥	药学
09713070	李利华	南京	医药软件
09714046	杨康元	宿州	计算机科学与技术
10111001	曹云	贵阳	中医
10111182	张光磊	黄山	中医
10212040	许贝雷	安庆	骨伤
10313057	朱亮	贵阳	制药工程
10412023	金柱	合肥	全科医学
09613032	牛芳菲	芜湖	中医
10431075	童丽祥	金寨	中西医结合
10512001	鲍州	黄山	医药贸易
09312016	李楠	南宁	药学
10512002	曹承	桂林	医药贸易
10611003	仇朵朵	巢湖	护理
09321053	殷晴净	上海	药物分析
10611004	崔振冬	合肥	护理

图 3-1 学生基本信息查询结果

【例 3.14】查询 KC.DBF 表中所有课程的信息。

```
SELECT 课程号,课程名,学时数 FROM KC
```

或者

```
SELECT * FROM KC
```

当查询表中的所有字段时，可以用“*”代表全体字段。查询结果同表 KC.DBF。

2. 使用选择谓词 ALL/DISTINCT 设定是否允许记录重复

如果在 SELECT 中没有使用任何一个选择谓词或使用了选择谓词 ALL，查询将返回符合条件的全部记录，而且允许在查询结果中包含重复记录。而在字段列表使用 DISTINCT，查询结果中若有重复记录则只保留一条。

【例 3.15】查询 CJ.DBF 表中成绩不低于 80 分的课程的课程号。

```
SELECT ALL 课程号 FROM CJ WHERE 成绩>=80
```

查询结果如图 3-2 所示。

【例 3.16】查询 CJ.DBF 表中成绩不低于 80 分的课程的课程号，并过滤重复记录。

```
SELECT DISTINCT 课程号 FROM CJ WHERE 成绩>=80
```

查询结果如图 3-3 所示。

3. 使用 WHERE 子句查询满足条件的元组

查询满足指定条件的元组可以通过 WHERE 子句实现，WHERE 子句一般引导条件表达式。WHERE 子句常用的查询条件如表 3.2 所示。

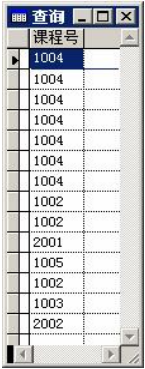


图 3-2 例 3.15 查询结果

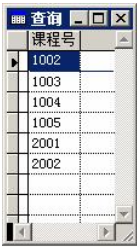


图 3-3 例 3.16 查询结果

表 3.2 WHERE 查询条件谓词一览表

查询条件	谓词
比较	=、>、<、>=、<=、!=、<>、!>、!<、NOT+前述比较运算符
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	AND、OR

【例 3.17】查询 CJ.DBF 表中 2001 号课程高于 80 分的成绩记录。

SELECT * FROM CJ WHERE 课程号="2001" AND 成绩>80

查询结果如图 3-4 所示。

【例 3.18】查询 CJ.DBF 表中 1003 号课程成绩在 60~80 分范围内的记录。

SELECT * FROM CJ WHERE 课程号="1003" AND 成绩 BETWEEN 60 AND 80

查询结果如图 3-5 所示。



图 3-4 例 3.17 查询结果



图 3-5 例 3.18 查询结果

BETWEEN...AND...的意思是“在……和……之间”，指定数据的值域范围。本例也可以写成“SELECT * FROM CJ WHERE 课程号="1003" AND 成绩>=60 AND 成绩<=80”。如果提出与本例相反要求的查询，如查询成绩在 60 分以下或 80 分以上的记录，则可以使用 NOT BETWEEN 进行查询。

另外, 本例涉及多重条件检索, 多重条件之间由谓词 AND 连接。

【例 3.19】查询 XS.DBF 表中籍贯为合肥、南京或南昌的学生的学号、姓名、性别、出生日期、籍贯和专业。

```
SELECT 学号,姓名,性别,出生日期,籍贯,专业 FROM XS WHERE 籍贯 IN ('合肥','南京','南昌')
```

查询结果如图 3-6 所示。



学号	姓名	性别	出生日期	籍贯	专业
10811023	万安宁	女	09/18/89	南昌	对外汉语
09311047	王维胜	男	07/23/89	合肥	药学
09713070	李利华	男	09/16/89	南京	医药软件
10412023	金柱	男	03/05/89	合肥	全科医学
10611004	崔振冬	女	12/30/89	合肥	护理

图 3-6 例 3.19 查询结果

<表达式> IN(<数值表集合>), 测试表达式的值是否在数值表集合中。

同理, 如果提出与本例相反的查询要求, 可以使用 NOT IN 来进行查询。如查询籍贯不是合肥、南京或南昌的学生的相关信息, 命令可以写成 “SELECT 学号,姓名,性别,出生日期,籍贯,专业 FROM XS WHERE 籍贯 NOT IN ('合肥','南京','南昌')”。

【例 3.20】查询 KC.DBF 表中课程名称包含“程序设计”的课程的相关记录。

```
SELECT * FROM KC WHERE 课程名 LIKE "%程序设计%"
```

查询结果如图 3-7 所示。

LIKE <字符串>, 一般进行字符匹配测试, 可以用通配符 “%” 代表 0 个或多个任意字符, 用通配符 “_” 代表一个任意字符, LIKE 一般用于模糊查询, 查询指定列值与匹配串相匹配的元组。LIKE 后使用的字符串通常用单引号、双引号或方括号括起来。

4. 使用 ORDER BY 排序

【例 3.21】查询 CJ.DBF 表中 1002 号课程的考试成绩, 并按成绩降序排列显示结果。

```
SELECT * FROM CJ WHERE 课程号="1002" ORDER BY 成绩 DESC
```

查询结果如图 3-8 所示。



课程号	课程名	学时数
1004	VFP程序设计	72
1005	VB程序设计	72

图 3-7 例 3.20 查询结果



学号	课程号	成绩
09311047	1002	92
10111001	1002	82
10111182	1002	80
09714046	1002	77
10212040	1002	72
09713070	1002	63
09321053	1002	61

图 3-8 例 3.21 查询结果



注意: 选项 ASC|DESC 指明对排序字段的排序规则, 系统默认的是按升序 ASC 排列。

另外, 本例查询命令也可以写成 “SELECT * FROM CJ WHERE 课程号="1002" ORDER BY 3 DESC”, 其中 ORDER BY 3 指的是依据检索结果中的第 3 列 (即 “成绩”) 进行排序。也就是说, ORDER BY 子句后面的排序依据可以是查询结果中的字段名称, 也可以是字段的序列号。但排序依据不能是含有函数的表达式。

【例 3.22】查询 CJ.DBF 表中 1002 号课程成绩位于前三名的学生成绩记录。

```
SELECT * FROM CJ WHERE 课程号="1002" ORDER BY 成绩 DESC TOP 3
```

查询结果如图 3-9 所示。



学号	课程号	成绩
09311047	1002	92
10111001	1002	82
10111182	1002	80

图 3-9 例 3.22 查询结果

如果在 ORDER BY 子句后面使用 TOP N, 则输出检索结果的前 N 条记录; 如果在 ORDER BY 子句后面使用 TOP N PERCENT, 则输出检索结果的前 N% 的记录。并且, TOP 子句只配合 ORDER BY 子句使用, 没有 ORDER BY 子句时, 不可使用 TOP 子句。

另外, TOP 子句也可以写在 SELECT 子句的字段列表前, 因此本例检索也可以写成“SELECT TOP 3 * FROM CJ WHERE 课程号="1002" ORDER BY 成绩 DESC”。

3.4.3 联接查询

基于两个或两个以上的表进行的查询, 称为联接查询。常见的联接主要包括内联接、外联接和全联接。联接查询的语法格式与简单查询大体相似, 主要是 FROM 子句的使用有一些差异。

1. 内联接

内联接, 只有满足联接条件的记录才出现在查询结果中。

【例 3.23】查询医药贸易专业学生的选课情况, 要求列出学生的学号、姓名、专业和课程号、成绩。

```
SELECT XS.学号,姓名,专业,课程号,成绩 FROM XS JOIN CJ ON XS.学号=CJ.学号;  
WHERE 专业="医药贸易"
```

查询结果如图 3-10 所示。



学号	姓名	专业	课程号	成绩
10512001	鲍洲	医药贸易	1004	63
10512002	曹承	医药贸易	1004	69

图 3-10 例 3.23 查询结果

使用多表联接查询时, 要注意以下两点:

(1) 两表的联接条件一定要写明, 如本例的“XS.学号=CJ.学号”, 这是联接的依据, 至关重要。

(2) 若所联接的两表有重名字段, 在查询结果字段列表中需要对重名字段加表名前缀 (如本例的 XS.学号), 以指明是哪个表中的字段。

2. 外联接

外联接查询, 常见的有左联接和右联接两种。

【例 3.24】左联接, 即除满足联接条件的记录出现在查询结果中外, 第一个表中不满足联接条件的记录也出现在查询结果中。如查询所有课程的选课情况, 包括已被选课程和未被选课程, 列出课程的课程号、课程名、学生的学号和成绩。

```
SELECT KC.课程号,课程名,学号,成绩 FROM KC LEFT JOIN CJ;
ON KC.课程号=CJ.课程号
```

查询结果如图 3-11 所示。

【例 3.25】右联接，即除满足联接条件的记录出现在查询结果中外，第二个表中不满足联接条件的记录也出现在查询结果中。如上例查询所有课程的选课情况也可以由右联接实现，命令如下：

```
SELECT KC.课程号,课程名,学号,成绩 FROM CJ RIGHT JOIN KC ON CJ.课程号=KC.课程号
```

查询结果与左联接完全一致。

3. 全联接

除满足联接条件的记录出现在查询结果中外，两个表中不满足联接条件的记录也出现在查询结果中，为全联接。

为了看到全联接查询的效果，在 CJ 表中添加一条新记录(0001,k0002,89)。

【例 3.26】对 CJ 表与 KC 表进行全联接查询。

```
SELECT KC.课程号,课程名,学号,成绩 FROM CJ FULL JOIN KC ON CJ.课程号=KC.课程号
```

查询结果如图 3-12 所示。

课程号	课程名	学号	成绩
2002	中医基础	09714046	84
2001	大学语文	10111182	90
2001	大学语文	10111001	65
1004	VFP程序设计	10412023	86
1004	VFP程序设计	10431075	87
1004	VFP程序设计	10512001	83
1004	VFP程序设计	10512002	69
1004	VFP程序设计	10611003	91
1004	VFP程序设计	10611004	83
1004	VFP程序设计	10212040	85
1004	VFP程序设计	10811023	68
1004	VFP程序设计	10111182	85
1004	VFP程序设计	10111001	72
1005	VB程序设计	09311047	89
1006	高等数学	09311047	79
1006	高等数学	09321053	53
1007	Access数据库	10313057	61
1008	汇编语言	NULL	NULL

图 3-11 例 3.24 查询结果

课程号	课程名	学号	成绩
1004	VFP程序设计	10431075	87
1004	VFP程序设计	10512001	83
1004	VFP程序设计	10512002	69
1004	VFP程序设计	10611003	91
1004	VFP程序设计	10611004	83
1004	VFP程序设计	10212040	85
1004	VFP程序设计	10811023	68
1004	VFP程序设计	10111182	85
1004	VFP程序设计	10111001	72
1005	VB程序设计	09311047	89
1006	高等数学	09311047	79
1006	高等数学	09321053	53
1007	Access数据库	10313057	61
1008	汇编语言	NULL	NULL
0001	VFP程序设计	0001	89

图 3-12 例 3.26 查询结果

3.4.4 分组与统计查询

在日常的数据查询工作中，经常要涉及到一些统计数据的查询。SQL 提供了统计查询功能，主要是通过 SQL 内置的许多聚合函数来实现的。聚合函数也称为集合函数或统计函数、聚集函数，其作用是对一组值进行统计运算并返回一个单值。SQL 提供的聚合函数主要有：

- COUNT([DISTINCT] *): 统计表中元组的个数。
- COUNT([DISTINCT]<字段名>): 统计字段值个数。
- SUM([DISTINCT]<字段名>): 计算字段值总和（必须是数值型字段）。
- AVG([DISTINCT]<字段名>): 计算字段平均值（必须是数值型字段）。
- MAX([DISTINCT]<字段名>): 求字段最大值。
- MIN([DISTINCT]<字段名>): 求字段最小值。



注意: DISTINCT 选项是指去掉字段重复值后再进行统计, 在一条统计命令中只能出现一次 DISTINCT。

1. 简单计算查询

【例 3.27】查询所有学生的学号、姓名与年龄。

```
SELECT 学号,姓名,2011-YEAR(出生日期) AS 年龄 FROM XS
```

查询结果如图 3-13 所示。

本例中, XS.DBF 表中只有学生的“出生日期”字段, 没有所要查询的年龄信息, 但是可以通过对“出生日期”字段进行运算得到每个学生的年龄。这种通过对原表中已存在的字段进行运算得到的新字段, 称为计算字段(或计算列、虚字段、逻辑字段)。

这种计算字段的运算表达式, 如“2011-YEAR(出生日期)”, 运行查询时, 系统所显示的列名为 EXP_n (n 为字段的列标), 可以在字段表达式后用 AS 给该字段取别名, 以便增强查询结果的可读性, 如本例中的“2011-YEAR(出生日期) AS 年龄”, 其中 AS 也可以省略不写。另外, 其他非计算字段也都可以在查询时取别名, 这里不再赘述。

学号	姓名	年龄
10811023	万安宁	22
09311047	王维胜	22
09713070	李利华	22
09714046	杨康元	23
10111001	曹云	21
10111182	张光磊	21
10212040	许贝雷	23
10313057	朱亮	22
10412023	金柱	22
09613032	牛芳菲	21
10431075	童丽祥	21
10512001	鲍洲	22
09312016	李楠	21
10512002	曹承	21
10811003	仇朵朵	23
09321053	殷晴净	23
10811004	崔振冬	22

图 3-13 例 3.27 查询结果

2. 使用聚合函数查询

使用聚合函数且没有使用分组短语 GROUP BY 的查询, 为简单统计查询。

【例 3.28】统计 XS.DBF 表中的学生总人数。

```
SELECT COUNT(*) FROM XS
```

查询结果如图 3-14 所示。

【例 3.29】统计选修了课程的学生人数。

```
SELECT COUNT(DISTINCT 学号) FROM CJ
```

在本例中, 由于一个学生可以选修多门课程, 学号可以出现多次, 所以统计学生人数应该把重复的学号去除, 因而用到了谓词 DISTINCT。查询结果如图 3-15 所示。

Cnt
17

图 3-14 例 3.28 查询结果

Dcnt_学号
17

图 3-15 例 3.29 查询结果

【例 3.30】查询学号为 10811023 的学生的考试总成绩和平均成绩。

```
SELECT 学号,SUM(成绩),AVG(成绩) FROM CJ WHERE 学号="10811023"
```

查询结果如图 3-16 所示。

学号	总分	平均分
10811023	68	68.00

图 3-16 例 3.30 查询结果

【例 3.31】查询“VFP 程序设计”课程的最高分和最低分。

```
SELECT 课程名,MAX(成绩),MIN(成绩) FROM CJ JOIN KC ON CJ.课程号=KC.课程号;
WHERE 课程名="VFP 程序设计"
```

查询结果如图 3-17 所示。

课程名	Max_成绩	Min_成绩
VFP程序设计	91	68

图 3-17 例 3.31 查询结果

3. 分组统计查询

在实际工作中，经常会遇到需要对表中数据进行分组统计的情况，例如要求查询每个学生的总分和平均分，这就需要在 CJ 表中按照学生的学号先分成若干组，每一组代表一个学生的几门课程考试信息，再对各组进行组内成绩统计，计算总分和平均分，这就是分组统计查询的主要功能。分组统计查询的命令格式是：

```
GROUP BY <分组依据列 1>[,<分组依据列 2>...] [HAVING <组提取条件>]
```

可以按一列或多列分组，还可以用 HAVING 进一步限定分组的条件。下面是几个分组查询的例子。

(1) GROUP BY 子句。

【例 3.32】查询每个学生的考试总分和平均分。

```
SELECT 学号, SUM(成绩), AVG(成绩) FROM CJ GROUP BY 学号
```

查询结果如图 3-18 所示。

本例先对表内记录按学生的学号分组，将学号相同的记录作为一组，再在每一组内进行 SUM 函数和 AVG 函数运算，求得每个学生的总分和平均分。

【例 3.33】统计每门课程的选课人数和平均成绩。

```
SELECT 课程号,COUNT(学号) AS 选课人数, AVG(成绩) FROM CJ GROUP BY 课程号
```

查询结果如图 3-19 所示。

学号	Sum_成绩	Avg_成绩
09311047	260	86.67
09312016	75	75.00
09321053	182	60.67
09613032	56	56.00
09713070	200	66.67
09714046	247	82.33
10111001	219	73.00
10111182	255	85.00
10212040	157	78.50
10313057	61	61.00
10412023	86	86.00
10431075	87	87.00
10512001	83	83.00
10512002	69	69.00
10611003	91	91.00
10611004	83	83.00
10811023	68	68.00

图 3-18 例 3.32 查询结果

课程号	选课人数	Avg_成绩
1001	1	68.00
1002	7	75.29
1003	4	71.75
1004	10	80.90
1005	1	89.00
1006	2	66.00
1007	1	61.00
2001	2	77.50
2002	2	75.50

图 3-19 例 3.33 查询结果



注意：GROUP BY 后面所带的<分组依据列>可以写列名，也可以写列的序号。使用 GROUP BY 子句进行分组统计时，SELECT 子句中的输出目标列只能是分组依据列或者聚合函数，不能是其他列。

本例的命令也可以写成“SELECT 课程号,COUNT(学号) AS 选课人数,AVG(成绩) FROM CJ GROUP BY 1”。

(2) HAVING 子句。HAVING 子句用于对分组后的结果再按条件筛选。

【例 3.34】查询选课人数超过 5 人的课程的课程号和选课人数。

```
SELECT 课程号,COUNT(学号) AS 选课人数 FROM CJ;
GROUP BY 课程号 HAVING COUNT(学号)>5
```

查询结果如图 3-20 所示。

HAVING 子句与 WHERE 子句的主要区别：当与 GROUP BY 子句一起使用时，HAVING 子句用于对分组后的记录进行筛选，没有 GROUP BY 子句时，HAVING 子句的作用等同于 WHERE 子句；而 WHERE 子句只能进行分组前的记录筛选，不可以用于 GROUP BY 子句的记录筛选。

课程号	选课人数
1002	7
1004	10

图 3-20 例 3.34 查询结果

3.4.5 嵌套查询

在 SQL 语言中，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 短语的条件表达式中的查询称为嵌套查询。被包含的查询块称为子查询或内层查询，而包含子查询的语句称为主查询或外层查询，为了与外层查询有所区别，总是把子查询写在圆括号内。

嵌套查询常见的命令格式是：

```
SELECT <表达式> FROM <表名> WHERE ... (<子查询>)
```

【例 3.35】查询成绩高于 85 分的学生的学号和姓名。

```
SELECT 学号,姓名 FROM XS WHERE 学号 IN;
(SELECT 学号 FROM CJ WHERE 成绩>85)
```

命令执行时，先执行内层查询语句，从 CJ.DBF 表中查询成绩高于 85 分的学生学号，然后再执行外层查询，根据学生的学号在 XS.DBF 表中找到这些学生对应的记录。单层查询 WHERE 子句中可以出现的条件谓词在嵌套查询中均可用，本例中用到了谓词 IN。此查询也可以用多表联接实现：

```
SELECT XS.学号,姓名 FROM XS,CJ WHERE XS.学号=
CJ.学号 AND 成绩>85
```

查询结果如图 3-21 所示。

学号	姓名
09311047	王维胜
09714046	杨康元
10111182	张光磊
10412023	金柱
10431075	董丽祥
10611003	仇朵朵

图 3-21 例 3.35 查询结果



注意：Visual FoxPro 只支持两层查询，即内层查询块和外层查询块，不支持 SQL 的多层嵌套查询。

【例 3.36】查询与王维胜同籍贯的学生的记录。

```
SELECT 学号,姓名,籍贯 FROM XS WHERE 籍贯=
( SELECT 籍贯 FROM XS WHERE 姓名="王维胜") AND 姓名!="王维胜"
```

查询结果如图 3-22 所示。

【例 3.37】查询 1004 号课程成绩高于该课程平均分的记录。

```
SELECT * FROM CJ WHERE 课程号="1004" AND;
成绩>(SELECT AVG(成绩) FROM CJ WHERE 课程号="1004")
```

查询结果如图 3-23 所示。



学号	姓名	籍贯
10412023	金柱	合肥
10611004	崔振冬	合肥

图 3-22 例 3.36 查询结果



学号	课程号	成绩
10412023	1004	86
10431075	1004	87
10512001	1004	83
10611003	1004	91
10611004	1004	83
10212040	1004	85
10111182	1004	85

图 3-23 例 3.37 查询结果

3.4.6 查询结果的设置

前面对 SQL 查询命令的介绍，默认的查询结果均是在“浏览”窗口中显示的。实际上，可以通过命令语句设置查询结果的多样输出。

将结果送至屏幕、打印机或文本文件：

- TO SCREEN
- TO PRINTER [PROMPT]
- TO FILE <文本文件名> [ADDITIVE]



注意：用 PROMPT 选项，可以设定打印前出现关于打印设置的对话框。用 ADDITIVE 选项，可以将本次查询结果追加到原文件的尾部，否则将覆盖原文件。

将结果放到临时只读表或永久表或数组：

- INTO CURSOR <临时表表名>
- INTO TABLE | DBF <永久表表名>
- INTO ARRAY <数组名>

【例 3.38】查询成绩高于 80 分的学生的学号、姓名、课程号和成绩，并将结果保存到新的名为“高分成绩信息”的表中。

```
SELECT XS.学号,姓名,课程号,成绩 FROM XS,CJ;
WHERE XS.学号=CJ.学号 AND 成绩>80 INTO
TABLE 高分成绩信息
USE 高分成绩信息
BROW
```

命令执行结果如图 3-24 所示。



学号	姓名	课程号	成绩
10412023	金柱	1004	86
10431075	董丽祥	1004	87
10512001	鲍州	1004	83
10611003	仇朵朵	1004	91
10611004	崔振冬	1004	83
10212040	许贝雷	1004	85
10111182	张光磊	1004	85
10111001	曹云	1002	82
10111182	张光磊	2001	90
09311047	王维胜	1005	89
09311047	王维胜	1002	92
09714046	杨康元	1003	86
09714046	杨康元	2002	84

图 3-24 例 3.38 运行结果

**本章小结**

- SQL 语言的数据定义功能、数据操作功能、数据查询功能。其中，数据查询功能是本章的核心内容。
- SQL 语言的数据定义功能，主要涉及 CREATE、ALTER 和 DROP 命令，分别完成表的定义、修改和撤消操作。
- SQL 语言的数据操纵功能，主要涉及 INSERT、UPDATE 和 DELETE 命令，分别完成表中数据的添加、修改和删除操作。
- SQL 语言的数据查询功能强大，主要包括简单查询、联接查询、嵌套查询和统计查询等。在完成查询的同时，还可以实现排序、删除重复记录、设定查询去向、取记录前几条等功能。

**习题三****一、选择题**

1. SQL 中修改表结构的命令是 ()。
A. CREATE TABLE B. ALTER TABLE C. DROP TABLE D. MODIFY TABLE
2. 下列说法中正确的是 ()。
A. INSERT 语句是在表的末尾插入记录
B. INSERT 语句是在表的开头插入记录
C. INSERT 语句是在表中指定位置插入记录
D. INSERT 语句可在表中任意位置插入记录
3. 修改表中数据相关的 SQL 命令是 ()。
A. ALTER B. UPDATE C. DELETE D. SELECT
4. 数据查询的 SQL 命令中，引导条件短语实现对元组选择的关键字是 ()。
A. FROM B. SELECT C. WHERE D. ORDER BY
5. SQL 查询语句中，ORDER BY 子句对查询结果进行降序排列的关键字是 ()。
A. ASC B. DESC C. HAVING D. CHANGE
6. DELETE 命令的作用是 ()。
A. 逻辑删除表中记录 B. 物理删除表中记录
C. 将表中记录全部删除 D. 修改表中数据
7. 标准 SQL 基本查询模块的结构是 ()。
A. SELECT...FROM...ORDER BY B. SELECT...WHERE...GROUP BY
C. SELECT...WHERE...HAVING D. SELECT...FROM...WHERE
8. SQL 查询 GROUP BY 子句的作用是 ()。
A. 对表中记录分组，以便进行分组统计 B. 对查询结果进行排序
C. 查询满足条件的元组 D. 选取所要查询的列
9. 下面不是查询语句输出结果的是 ()。

- A. 表单 B. 屏幕 C. 表 D. “浏览”窗口
10. 在 SQL 语句中,与表达式“年龄 BETWEEN 19 AND 21”功能相同的是 ()。
- A. 年龄>19 AND 年龄<21 B. 年龄>=19 AND 年龄<=21
C. 年龄<19 AND 年龄>21 D. 年龄<19 OR 年龄>21
11. 已知数据表 XSXX.DBF 中包含:学号 (C)、姓名 (C)、专业 (C) 和成绩 (N) 等字段,检索其中成绩大于 80 分的学号,下列语句中正确的是 ()。
- A. SELECT 学号 FROM XSXX WHERE 成绩>80
B. SELECT 学号 FROM XSXX WHERE 成绩 BETWEEN 0 AND 80
C. SELECT 成绩 FROM XSXX WHERE 成绩>80
D. SELECT 成绩 FROM XSXX WHERE 成绩 BETWEEN 0 AND 80
12. 对于数据表 XSXX.DBF,语句“SELECT * FROM XS INTO CURSOR AA”的输出结果将存放于 () 中。
- A. 临时表 AA B. 数据库表 AA C. 自由表 AA D. 数组 A
13. 已知数据表 XSCJ.DBF 包含学号 (C)、英语 (N)、高数 (N)、VFP (N) 4 个字段,想要查询每位学生的平均成绩,正确的 SQL-SELECT 命令是 ()。
- A. SELECT *,AVG(*) AS 平均成绩 FROM XSCJ
B. SELECT *,AVG (英语+高数+VFP) AS 平均成绩 FROM XSCJ
C. SELECT 学号,(英语+高数+VFP)/3 AS 平均成绩 FROM XSCJ
D. SELECT 学号,AVG (英语,高数,VFP) AS 平均成绩 FROM XSCJ
14. 已知选课数据表 XK.DBF 中包含:学号 (C)、课程号 (C)、学分 (N) 等字段,若需要统计其中已经选修了课程的学生人数,下列语句中正确的是 ()。
- A. SELECT COUNT(*) FROM XK
B. SELECT COUNT(学号) FROM XK
C. SELECT COUNT(DISTINCT 学号) FROM XK
D. SELECT COUNT(DISTINCT *) FROM XK

二、填空题

- Visual FoxPro 所支持的 SQL 语言主要包括数据定义、_____和_____三大功能。
- 修改表结构时,_____可以实现向表中添加新字段,_____可以修改字段名称。
- 撤消(删除)表的命令是_____。
- SQL 查询语句中,通常与字符串匹配运算符 LIKE 一起使用的通配符有_____和_____。
- SQL 查询语句中,表达式“籍贯 IN("山东","河南")”可以改写为_____。
- “UPDATE CJ SET 成绩=成绩+5”命令的作用是_____。
- 使用 GROUP BY 子句进行分组统计查询时,如果要对分组后的记录再进行筛选,需要使用_____短语。
- SQL 查询语句中,要筛除重复记录,可以使用_____短语。
- SQL 查询语句中,取查询结果的前几条记录的短语是_____。
- 常见的多表联接查询包含_____、_____和全联接。
- 统计查询中,统计最大值的是_____函数,统计最小值的是_____函数。

三、简答题

1. SQL 语言具有哪些主要功能和特点? Visual FoxPro 中使用的 SQL 语言的命令动词有哪些?
2. SQL 语言的数据定义功能主要包含哪些?
3. 与分组统计查询配合使用的 HAVING 短语与 WHERE 短语有什么区别?
4. 内联接查询与外联接查询的区别是什么?
5. 查询结果有哪些输出方式?

四、操作题

1. 利用 XS.DBF、KC.DBF、CJ.DBF 三张表实现如下查询:
 - (1) 查询课程表 KC.DBF 中的全部数据。
 - (2) 查询护理系学生的全部信息。
 - (3) 查询不及格的成绩信息。
 - (4) 查询学时数在 70~90 之间的课程的信息。
 - (5) 查询所有籍贯为合肥的男生的信息。
 - (6) 统计 1004 号课程的最高分和最低分。
 - (7) 统计每门课程的考试人数以及平均分。
 - (8) 统计每个学生的考试门数、总分和平均分。
 - (9) 查询选修了“大学语文”的学生的学号、姓名和班级。
 - (10) 查询成绩在 80 分以上的学生的学号、姓名、专业、课程名和成绩。
 - (11) 用嵌套查询实现对选修了 1002 号课程的学生的基本信息的查询。
 - (12) 用嵌套查询实现对没有被选修的课程相关信息的查询。
2. 利用 SQL 语句创建如表 3.3 所示结构的新表“员工.DBF”并进行如下操作:

表 3.3 员工表

列名	类型	宽度
员工 ID	C	10
姓名	C	10
出生日期	D	
工龄	I	
家庭住址	C	20

- (1) 向“员工.DBF”表中添加 2 条新记录: (0001,张林,1982-4-5,5,北京市朝阳区东大街)和(0002,李娜,1987-3-4,2,北京市海淀区解放路)。
- (2) 将员工张林的工龄改为 7。
- (3) 删除员工张林的信息。