

第五章 查询、视图与SQL

5.1 创建查询

利用查询可以为其他数据库提供新的数据表，可以从单个表中提取有用的数据，也可以从多个表中提取综合信息。查询文件可以通过查询设计器来创建，也可以利用查询向导来创建，查询文件的扩展名为.qpr。

5.1.1 用查询设计器创建查询

用查询设计器创建查询需要打开查询设计器，打开查询设计器最常用的方法有如下两种：

(1) 用菜单方式打开查询设计器

选择【文件】菜单的【新建】命令，或单击【常用】工具栏的【新建】按钮，打开【新建】对话框，在该对话框中选择【查询】，再单击【新建文件】按钮，即可进入【查询设计器】窗口。

(2) 用命令方式打开查询设计器

格式: CREATE QUERY <查询文件名>

功能: 打开查询设计器创建查询。

说明: <查询文件名>可以不加扩展名, 系统会自动为该文件加上扩展名.qpr。

不管使用哪种方法打开查询设计器创建查询，在打开【查询设计器】窗口的同时都需要选定查询所需的表或视图。如果没有打开数据库，会在打开【查询设计器】窗口的同时出现【打开】对话框供用户在其中选择所需的表或视图。如果已经打开了数据库，则会在打开【查询设计器】窗口的同时出现如下图所示的【添加表或视图】对话框。



在该对话框的【数据库中的表】列表框中显示了当前数据中的所有表以供选择，依次选择要用到的表或视图，然后单击【添加】按钮；如果还要选择自由表，则可以单击【其他...】按钮进行选择。表或视图添加完成后，单击【关闭】按钮正式进入【查设计器】窗口。

如果选择的是多个表或视图，则表之间必须是有联系的。查询设计器会自动根据联系提取联接条件，否则在打开查询设计器之前还会打开一个指定联接条件的对话框，由用户来设计联接条件。

利用查询设计器不仅可以创建单表查询，还可以创建多表查询，其操作方法下面分别举例说明。

【例5.1】 创建一个对单表dab.dbf的查询文件dabcx.qpr，查询男生的学号、姓名、专业班级，并按“学号”字段排序。

操作步骤如下：

①启动查询设计器。

打开【查询设计器】窗口并选择需要使用的数据表
dab. dbf

②选择查询输出字段。

在【字段】选项卡页面的【可用字段】中，选择字段：
学号、姓名、专业班级，单击【添加】按钮将所选字段添加到
【选定字段】列表框中，或双击上述字段，结果如下图所示。

查询设计器 - 查询1

Dab

- *
- 学号
- 姓名
- 性别
- 出生日期
- 专业班级
- 照片

字段 | 联接 | 筛选 | 排序依据 | 分组依据 | 杂项

可用字段 (B):

- Dab. 性别
- Dab. 出生日期
- Dab. 专业班级
- Dab. 照片

添加 (A) >

全部添加 (Q) >>

< 移去 (R)

<< 全部移去 (L)

选定字段 (C):

- Dab. 学号
- Dab. 姓名
- Dab. 专业班级

函数和表达式 (U):

...

③设置查询条件。

选择【筛选】选项卡，单击【字段名】下的组合框，选择筛选的字段“dab. 性别”，在【条件】组合框中选择“=”，在【实例】文本框中输入“男”，如下图所示。



④设置查询输出排序依据。

选择【排序依据】选项卡，双击“学号”字段，设置查询结果按“学号”升序排列，如下图所示。



⑤保存查询文件。

选择【文件】菜单下的【保存】或【另存为】命令，进入【另存为】对话框，输入查询文件名“dabcx”，单击【保存】按钮，得到查询文件dabcx.qpr。

⑥关闭查询设计器窗口。

单击【查询设计器】窗口的关闭按钮。

【例5.2】利用上一章已经建立的“学生学籍管理系统.dbc”数据库创建一个多表查询dbcx1.qpr，查询“陈兴华”同学的各科成绩，要求给出学号、姓名、课程名和成绩，并按照“成绩”进行排序。

操作步骤如下：

①启动查询设计器。

打开“学生学籍管理.dbc”数据库。启动查询设计器的同时依次选择dab.dbf, xscjb.dbf和kcb.dbf, 由于该数据库中的这3个表已建立永久关系，故在查询设计器中这3表已自动建立联接，如下图所示。



②选择查询输出字段。

在【字段】页面中选择输出字段：dab. 学号、dab. 姓名、kcb. 课程名和xkcjb. 成绩。

③设置查询条件。

设定筛选条件为：dab. 姓名="陈兴华"。

④设置查询输出排序依据。

选择【排序依据】选项卡，双击字段xkcjb.成绩，添加到排序条件中。

⑤保存查询文件。

选择【文件】菜单下【保存】或【另存为】命令，进入【另存为】对话框，将查询文件保存为dbcx1.qpr。

⑥关闭查询设计器窗口。

单击【查询设计器】窗口的关闭按钮。

5.1.2 用查询向导创建查询

利用查询向导创建查询，操作方法举例说明如下。

【例5.3】 利用查询向导创建数据表kcb.dbf和xkcjb.dbf的多表查询dbcx2.qpr，查询选修大学计算机基础且成绩在60分及以上的学生学号、课程名、成绩。

操作步骤如下：

①启动查询向导。

选择【文件】菜单下【新建】命令，打开【新建】对话框，选择【查询】，单击【向导】按钮，弹出【向导选取】对话框，如下图所示。



②输出字段的选取。

在【选择要使用的向导：】列表框中选择“查询向导”，然后单击【确定】按钮，即出现【查询向导】的“步骤1-字段选取”对话框，如下图所示。

在【数据库和表】组合框中，选择“学生学籍管理系统”数据库。在列表框中选择查询输出字段，如下图所示：



在xkcjb.dbf中，将“可用字段”栏中的字段学号选入“选定字段”栏中。

在kcb.dbf中，将“可用字段”栏中的字段课程名选入“选定字段”栏中。

在xkcjb.dbf中，将“可用字段”栏中的字段成绩选入“选定字段”栏中。

③建立表间关系。

单击【下一步】按钮，进入【查询向导】“步骤2-为表建立关系”对话框。在左侧的父表字段栏选择“xkcjb.课程号”，在右侧的子表字段栏选“kcb.课程号”，单击【添加】按钮，将两表的关系添加到了关系栏列表框中，结果如下图所示。单击【下一步】按钮，进入【查询向导】“步骤2a-字段选取”对话框，共有4种字段选取方式，这里采用默认项。

查询向导 [X]

步骤 2 - 为表建立关系

您希望如何建立查询所基于的表间关系？

选择匹配字段创建关系，并将每一个关系加入到列表中



E	F	G
1	AAA	XXX
2	BBB	XXX
3	CCC	XXX
4	DDD	XXX

H	F	I
1	AAA	YYY
2	BBB	YYY
3	CCC	YYY
4	DDD	YYY

XKCJB. 课程号 (C) KC3. 课程号 (C)

添加 (A) 移去 (R)

XKCJB. 课程号=KC3. 课程号

帮助 (H) 取消 <上一步 (P) 下一步 (N) > 完成 (F)

④设置查询条件。

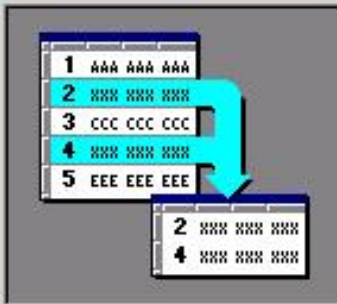
单击【下一步】按钮，进入【查询向导】的“步骤3-筛选记录”对话框。设置查询记录输出的筛选条件如下图所示：

➤在【字段】栏中选取“kcb.课程名”选项，在【操作符】列表框选取“等于”选项，在“值”栏输入“大学计算机基础”。

➤在【字段】栏中选取“xkcjb.成绩”选项，在【操作符】列表框选取“大于或等于”选项，在“值”栏输入60。

查询向导

步骤 3 - 筛选记录



是否仅想显示部分记录？

如果想查询特定记录时，可使用字段框，操作符框和值框来创建表达式。

单击“预览”按钮可以查看结果。

字段 (I):

KCB. 课程名

操作符 (I):

等于

值 (V):

“大学计算机基础”

与 (A) 或 (O)

字段 (E):

XKCJB. 成绩

操作符 (E):

大于或等于

值 (U):

60

预览 (P)...

帮助 (H)

取消

<上一步 (B)

下一步 (N) >

完成 (F)

⑤预览查询效果。

单击【预览】按钮，进入【预览】窗口，结果如下图所示。



The image shows a software window titled "预览" (Preview) with a blue title bar and standard Windows window controls (minimize, maximize, close). The window contains a table with the following data:

学号	课程名	成绩
20050025	大学计算机基础	93.0
20050010	大学计算机基础	72.0

⑥排序查询结果。

关闭【预览】窗口，返回【查询向导】“步骤3-筛选记录”。单击【下一步】按钮，进入【查询向导】的“步骤4-排序记录”对话框。这一步设置输出字段的排序顺序。将【可用字段】栏中的“xkcjb.成绩”以降序添加到“选定字段”栏。单击【下一步】按钮，进入【查询向导】的“步骤4a-限制记录”对话框，该对话框主要用来设置在浏览查询窗口中显示记录的限制。本例选定默认值。

⑦保存查询文件。

单击【下一步】按钮，进入【查询向导】的“步骤5-完成”对话框，如下图所示。选择“保存并运行查询”选项，单击【完成】按钮，进入【另存为】对话框，输入查询文件名dbcx2，单击【保存】按钮，则多表查询文件生成。



5.1.3 查询文件的运行

查询文件设计完成后，可以运行查询文件显示查询结果，如果查询结果不满意或不符合要求时，可以修改查询文件。

运行查询文件通常有3种方法：

（1）在查询设计器中直接运行

在查询设计器中设计查询时，每设计好一部分，都可以运行查询，查看结果。其方法是：在【查询设计器】窗口中选择【查询】菜单中的【运行查询】选项，或单击常用工具栏的【运行】按钮，即可运行查询。

(2) 利用菜单选项运行

当查询文件设计完成并保存后，可以通过选择【程序】菜单中的【运行】命令，打开【运行】对话框。选择要运行的查询文件，再单击【运行】按钮，即可运行。例如，我们选择运行查询文件dabcx.qpr后结果如下图所示。



The screenshot shows a window titled '查询' (Query) with a table containing student information. The table has four columns: '学号' (Student ID), '姓名' (Name), '专业班级' (Major Class), and an empty column. The first row is selected, showing student ID 20043679, name 邬德斋, and major class 机械制造4班. The other rows are: 20050005 张广田 计算机1班; 20050025 陈兴华 英语3班; 20050030 陈宗贵 英语3班; 20050035 张伟 英语3班; 20050040 张光辉 英语3班.

学号	姓名	专业班级	
20043679	邬德斋	机械制造4班	
20050005	张广田	计算机1班	
20050025	陈兴华	英语3班	
20050030	陈宗贵	英语3班	
20050035	张伟	英语3班	
20050040	张光辉	英语3班	

(3) 用命令运行

在命令窗口中执行运行查询文件的命令，也可以运行查询文件。

格式： D O <查询文件名>

功能： 运行查询文件。

说明： <查询文件名>中的扩展名不能缺省。

例如，我们要运行查询文件dbcx1.qpr，则可以在命令窗口中输入命令：D0 dbcx1.qpr，其运行结果如下图所示。



学号	姓名	课程名	成绩
20050025	陈兴华	PC机原理与应用	87.0
20050025	陈兴华	大学计算机基础	93.0

5.1.4 查询文件的修改

修改查询文件的方法通常有2种：

（1）用菜单打开查询文件进行修改

选择【文件】菜单中的【打开】命令，指定文件类型为“查询”，选择相应的查询文件，单击【确定】，打开该查询文件的查询设计器，即可对该查询文件进行修改。

（2）用命令打开查询文件进行修改

格式：MODIFY QUERY<查询文件名>

功能：打开指定查询文件的查询设计器，以便修改该查询文件。

5.1.5 查询结果的输出

在VFP中，根据需要可将查询结果按以下几种方式输出：

浏览：将查询结果输出到浏览窗口（默认）。

临时表：将查询结果保存于一个临时表（只读）中。

表：将查询结果保存于一个数据表中，用户可随意处理这个表。

图形：将查询结果用图形方式输出。

屏幕：将查询结果输出到屏幕上。

报表：将查询结果输出到一个报表文件（.frx）。

标签：将查询结果输出到一个标签文件（.lbx）。

【例5.4】为【例5.3】所创建的查询文件dbcx2.qpr，定制查询结果的输出格式。

操作步骤如下：

①打开查询文件dbcx2.qpr，如下图所示。



②选择【查询】菜单下【查询去向】，进入【查询去向】对话框，如下图所示。



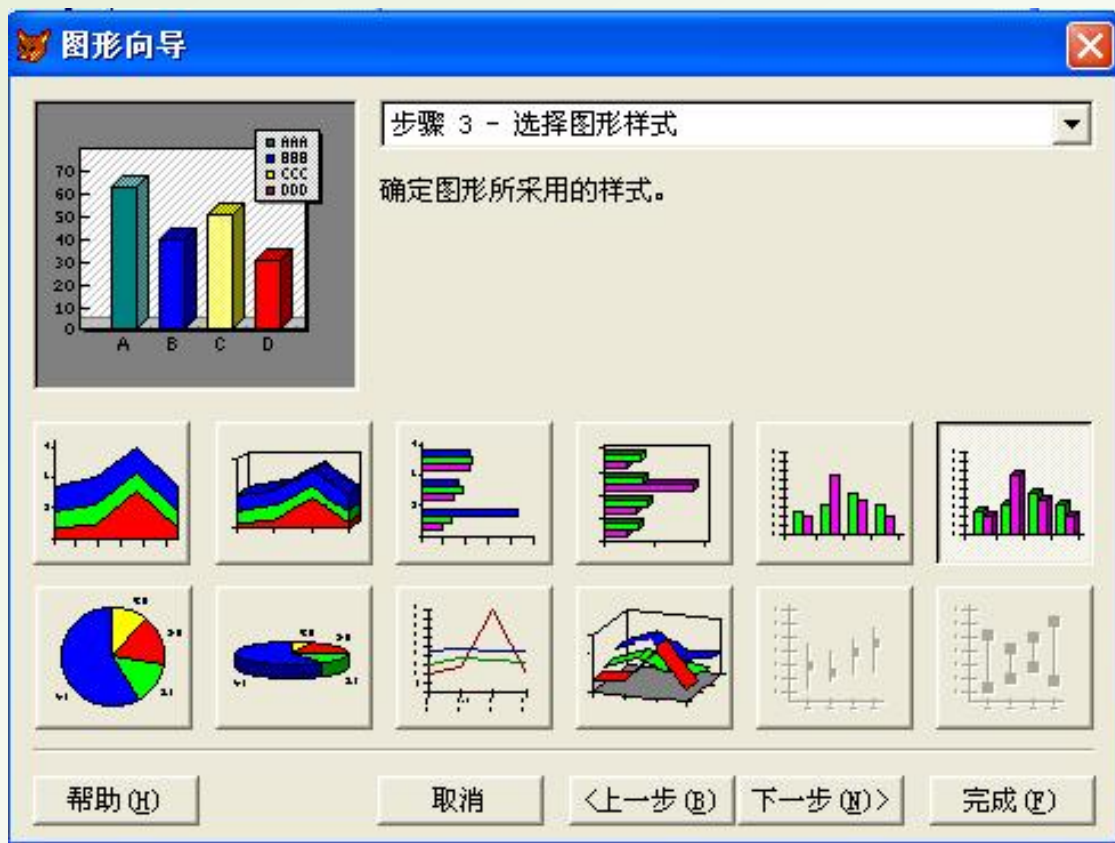
有7种输出去向可供选择，在这里只选择其中常用的4种。

- 若选择【浏览】选项，单击【确定】按钮，则返回Visual FoxPro 6.0主菜单。选择【查询】菜单下【运行查询】命令，可以查看结果。
- 若选择“临时表”选项，单击【确定】按钮，则输出被存入名为dbcx2.dbf的临时文件中，并返回Visual FoxPro 6.0主菜单。
- 若选择“表”选项，单击【确定】按钮，则系统自动生成一个名为dbcx2.dbf的表文件。

• 若选择“图形”选项，单击【确定】按钮，则返回Visual FoxPro 6.0主菜单。选择【查询】菜单下【运行查询】命令，则进入“图形向导”的【步骤2-定义布局】对话框，如图5.16所示。定义图形的布局，即横、纵坐标的数据来源。

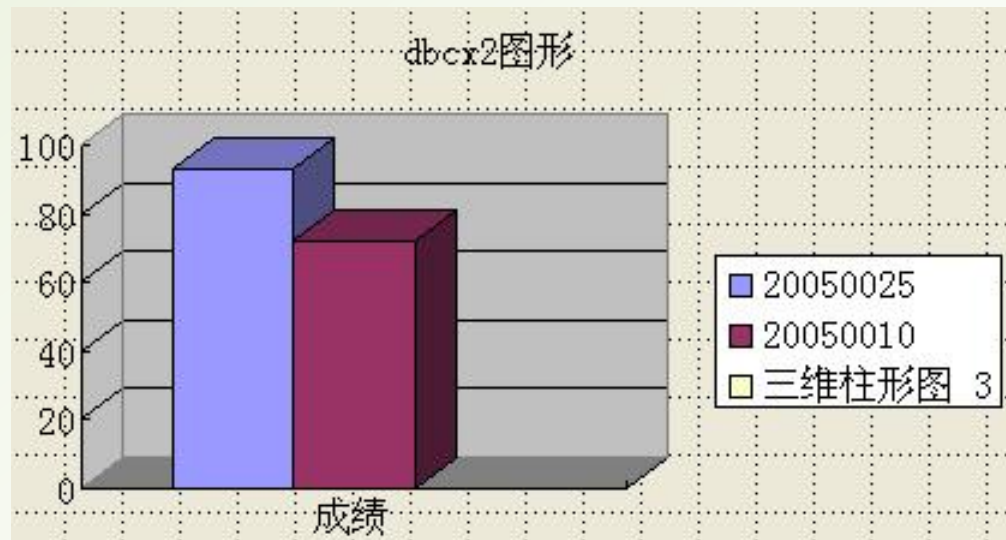


单击【下一步】按钮，则进入【图形向导】“步骤3-选择图形样式”，如下图所示。根据需要选择其中一种图形样式。



单击【下一步】按钮，则进入【图形向导】“步骤4-完成”窗口，输入图形标题，单击【完成】按钮，进入“另存为”窗口。

在“另存为”窗口中，输入图形文件名“dbcx2图形”，单击【保存】按钮，则出现如下图所示结果。



5.2 创建视图

视图与查询相似，它们都可以从一个或多个相关联的表中提取信息，但查询是以只读方式访问表文件，而视图却是以读写方式访问表文件，因此可以通过修改视图中记录的值来实现对源表的数据更新。视图可以看成是从一个或多个数据表中导出的一张“虚拟表”，因为其中并不存放真正的数据，即视图中的数据取自源表，每次打开视图时，都要从源表中重新提取数据。视图是不能单独存在的，它依赖于数据库，是数据库的一部分，只有在打开数据库时才能创建和使用视图。

根据数据库中数据的来源，视图可以分为本地视图和远程视图。本地视图指由本地表而生成的视图。远程视图指通过ODBC从远程数据源所建立的视图。

创建视图的过程与创建查询的过程类似，既可通过视图设计器创建，又可以利用视图向导创建。

5.2.1 利用视图设计器创建视图

视图是数据库的一部分，只有打开或创建包含视图的数据库后，才能创建视图。

【例5.5】为数据库文件“学生学籍管理系统.dbc”创建一个本地视图文件v1.vue，要求显示学号、姓名、课程名称、成绩，并且要求只能更新“成绩”字段。

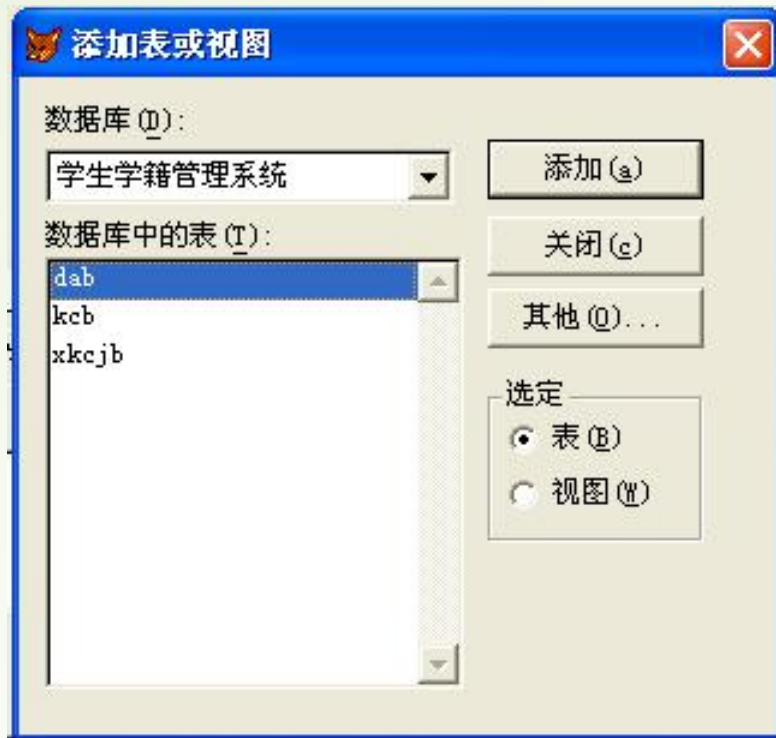
操作步骤如下：

①打开数据库。

打开数据库文件“学生学籍管理系统.dbc”，进入数据库设计窗口。

②启动视图设计器。

选择【文件】菜单下【新建】命令，打开“新建”窗口。选择“视图”，单击【新建文件】按钮，进入【视图设计器】窗口，并同时打开“添加表或视图”对话框，如下图所示。



③添加表或视图。

分别将表dab. dbf、 xkcjb. dbf和kcb. dbf添加到视图设计器中，单击【关闭】按钮，返回到【视图设计器】窗口。

④在【视图设计器】窗口中，进行如下设置：

- 在“字段”选项卡选择字段： dab. 学号、 dab. 姓名、 kcb. 课程名和xkcjb. 成绩。
- 在“更新条件”选项卡中的“表”中选择可更新表： xkcjb，在“字段名”中选择xkcjb的更新关键字和字段： xkcjb. 成绩，并将字段名前面的关键字标识（小钥匙）、字段更新标识（小粉笔）均加上“√”。

- 在“SQL WHERE 子句包括”中勾选“关键字和可更新字段”，在“使用更新”中勾选“SQL UPDATE”，然后勾选“发送SQL更新”，如下图所示。



⑤保存视图。选择【文件】菜单下【保存】命令，输入视图文件名v1。此时在数据库“学生学籍管理系统.dbc”中会添加一个视图文件v1.vue，如下图所示。



⑥浏览视图，打开数据库。在【数据库设计器】窗口中双击视图对象，可以在【浏览】窗口中显示视图内容，如下图所示。



	学号	姓名	课程名	成绩
▶	20050010	蜂雪燕	高等数学 I	38.0
	20050010	蜂雪燕	大学英语	100.0
	20050020	冯艳	高等数学 I	45.0
	20043567	那措央中	PC机原理与应用	54.0
	20043567	那措央中	操作系统基础	88.0
	20050025	陈兴华	大学计算机基础	88.0
	20050010	蜂雪燕	马列、毛泽东思想理论	98.5
	20050010	蜂雪燕	大学计算机基础	67.0
	20050005	张广田	大学英语	58.0
	20050025	陈兴华	PC机原理与应用	82.0
	20045643	崔婷婷	高等数学 I	50.0
	20050035	张伟	马列、毛泽东思想理论	95.5
	20045643	崔婷婷	操作系统基础	76.5

⑦打开视图，选中某个学生记录中的“成绩”字段值，进行更改。关闭视图后，原数据表中数据就被更新了。

5.2.2 利用视图向导创建视图

利用【视图向导】可以创建本地视图，启动【视图向导】的方法常用的有如下2种：

- 打开【数据库设计器】，选择【数据库】菜单中的【新建本地视图】命令，进入【新建本地视图】对话框，然后单击【视图向导】按钮。
- 选择【文件】菜单的【新建】命令，在出现的【新建】对话框中选择【视图】，然后单击【向导】按钮。

视图还可以用命令CREATE VIEW<视图文件名>进行创建，可以用命令DROP VIEW<视图文件名>进行删除。

视图一经建立，就可以像表一样使用，凡是用到表的地方都可以使用视图。

5.3 SQL语言

结构化查询语言SQL (Structured Query Language) 首先由Boyce 和Chamberlin 在1974年提出, 1979由IBM公司首次成功地使用在关系型数据库管理系统System R上。SQL语言简洁易学, 功能丰富, 使用方便, 备受用户欢迎, 现在已成为关系型数据库通用的查询语言, 适用于绝大多数的关系数据库系统。

SQL语言的功能主要包括数据定义、数据控制、数据操作和数据查询等, 其中, 数据定义是对关系模式一级的定义; 数据控制是对数据访问权限的授予或撤销; 数据操作是对关系中的具体数据进行增加、删除和更新等操作; 数据查询是对关系中的数据进行检索。

SQL语言具有以下的特点：

- SQL是一种一体化的语言，它包括了数据定义、数据控制、数据操作和数据查询等方面的功能，可以完成与数据库相关的全部工作。查询是SQL语言最重要的组成部分。
- SQL语言是一种高度非过程化的语言，它没有必要告诉计算机“如何”去做，只需要告诉计算机“做什么”，SQL语言就可以将要求交给系统自动完成。
- SQL语言非常简洁。虽然SQL语言功能较强，但它只有为数不多的几条命令，表 5.1 给出了各类命令动词。另外SQL语言也非常简单，很接近英语自然语言，因此容易学习和掌握。

SQL命令动词

SQL功能	命令动词
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

5.3.2 使用SQL语句定义表

SQL语句可以进行表的定义，即表结构的创建、修改和删除，从而实现对表结构的设计及维护等操作。

1) 创建表

表结构可以通过表设计器建立，也可以通过SQL的CREATE TABLE命令建立。

格式： CREATE TABLE|DBF <表名> [FREE]
(<字段名1> <类型> [(宽度[, 小数位数)])]
[,<字段名2> <类型> [(宽度[, 小数位数)])]
[null][not null][default <表达式>][primary key]
[.....]

功能： 生成一个由表名所标识的表。

下表列出了在CREATE TABLE命令中可以使用的数据类型及说明。

字母类型	字段宽度	小数位	说明
C	N	-	字符型字段的宽度为n
D	-	-	日期类型 (Date)
T	-	-	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	-	-	整数类型 (Integer)
B	-	-	双精度类型 (Double)
Y	-	-	货币类型 (Currency)
L	-	-	逻辑类型 (Logical)
M	-	-	备注类型 (Memo)
G	-	-	通用类型 (General)

【例5.6】建立表dab2.dbf（结构和表dab.dbf类似）。

CREATE TABLE dab2 (学号 C(8) NOT NULL PRIMARY KEY;

,姓名 C(8),性别 C(2),出生日期 D;

,专业班级 C(20),照片 G,简历 M)

用MODIFY STRUCTURE显示，结果如下图所示。



【例5.7】 建立数据库表xkcjb2.dbf（结构和表xkcjb.dbf类似）。

```
CREATE DATABASE XSXJGLXT
```

```
CREATE TABLE xkcjb2 ( 学号 C(8) NOT NULL, 课程号 C(3), ;  
学期 C(2), 成绩 n(5, 1) , 重考成绩 n(5, 1))
```

2) 修改表结构

用SQL语句修改表结构，无须用USE命令打开表。

修改表结构的命令均是以ALTER TABLE开头。

(1) 修改字段属性

格式：

```
ALTER TABLE <表名> ALTER <字段名1> <类型> [(宽度[, 小数位数])] ;  
                [ALTER <字段名2> <类型> [(宽度[, 小数位数])] [...]
```

功能：修改指定表的结构。

【例5.8】 把表dab2.dbf中的学号字段宽度改为10，专业班级字段宽度改为30。

```
ALTER TABLE dab2 ALTER 学号 C(10) ALTER 专业班级 C(30)
```

(2) 增加字段

格式:

```
ALTER TABLE <表名> ADD <字段名1> <类型> [(宽度[, 小数位数])] ;  
        [ADD <字段名2> <类型> [(宽度[, 小数位数])] [...]
```

功能: 为指定表增加部分字段。

【例5.9】 给表dab2.dbf增加字段入学总分C(5, 1)及身高N(4, 2)。

```
ALTER TABLE dab2 ADD 入学总分 N(5, 1)  ADD 身高 N(4, 2)
```

(3) 字段改名

格式:

```
ALTER TABLE 表名 RENAME <字段名1> TO 新字段名1;  
          [RENAME <字段名2> TO 新字段名2] [...]
```

功能: 为指定表中的部分字段改名。

【例5.10】 将表dab2.dbf中“入学总分”字段改名为“入学成绩”。

```
ALTER TABLE dab2 RENAME 入学总分 TO 入学成绩
```

(4) 删除字段

格式:

```
ALTER TABLE 表名 DROP<字段名1> [DROP<字段名2>] [...]
```

功能: 删除指定表中的部分字段。

【例5.11】删除表dab2.dbf中身高字段。

```
ALTER TABLE dab2 DROP身高
```


3) 删除表

格式:

```
DROP TABLE 表名
```

【例5.12】 删除自由表dab2.dbf。

```
DROP TABLE dab2
```

【例5.13】 删除数据库表xkcjb2.dbf。

```
OPEN DATABASE XSXJGLXT
```

```
DROP TABLE xkcjb2
```

注意：DROP TABLE直接从磁盘上删除“表名”所对应的.dbf文件。如果“表名”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，此后在打开时会出现错误提示。所以要删除数据库中的表时，最好应使数据库是当前打开的数据库。

5.3.3 使用SQL语句查询表数据

SQL语句最主要的功能是查询，即把数据表中的数据查找出来，以便浏览和使用。SQL语句中查询表数据使用的是SELECT语句。SELECT语句的基本形式分为3部分：查什么数据、由哪儿查数据、查的条件是什么，即SELECT FROM WHERE。

1) 基本的SELECT语句

格式：

```
SELECT    <字段名1> [, <字段名2>...]
```

```
FROM      [ <数据库名称! > ] <表或视图>
```

功能：从指定表或视图中查询全部记录的指定字段值。

【例5.14】 查询dab中所有学生。

```
SELECT * FROM dab
```

结果如下图所示。其中“*”是通配符，表示所有字段，该命令等价于：

```
SELECT 学号, 姓名, 性别, 出生日期, 专业班级 FROM dab
```



学号	姓名	性别	出生日期	专业班级
20050005	张广田	男	10/11/87	计算机1班
20050010	蜂雪燕	女	07/02/87	计算机1班
20050015	何芳	女	05/06/87	计算机1班
20050020	冯艳	女	09/09/87	计算机1班
20050025	陈兴华	男	02/14/87	英语3班
20050030	陈宗贵	男	07/02/87	英语3班
20050035	张伟	男	03/24/87	英语3班
20050040	张光辉	男	03/07/87	英语3班
20043567	那措央中	女	06/30/86	机械制造4班
20043679	鄂德斋	男	04/26/86	机械制造4班
20045643	崔婷婷	女	06/08/86	机械制造4班
20047634	冯丽娟	女	06/30/86	机械制造4班

【例5.15】 查询dab中“姓名”、“性别”字段。

```
SELECT 姓名, 性别 FROM dab
```

结果将仅显示如图中的第2列、第3列数据。

2) 唯一性查询

唯一性查询指对于要显示的结果，若有重复出现时，则只显示重复行中的一行。这可通过在基本SELECT语句中加入DISTINCT关键字的方法实现。若不加此关键字，则显示全部行。

【例5.16】查询dab中“专业班级”字段，要求数据重复时，仅显示相重复的一条记录。

```
SELECT 专业班级 FROM dab DISTINCT
```

结果如图所示。



专业班级
机械制造4班
计算机1班
英语3班

3) 带条件的查询

在SELECT语句中，可以通过WHERE子句为查询规定条件

WHERE子句的格式：

WHERE <条件表达式1> [AND <条件表达式2>...]

功能：查询指定表或视图中满足查询条件的记录。其中<条件表达式>可以是关系表达式，也可以是逻辑表达式，下表列出了组成<条件表达式>常用的运算符。

运算符类别	运算符	实例
关系运算符	=、>、<、>=、<=、<>	成绩>=60
逻辑运算符	NOT、 AND、 OR	成绩>=60 AND成绩<70
字符串匹配运算符	LIKE	性别 LIKE "男"
范围运算符	BETWEEN AND	成绩BETWEEN 60 AND 70
空值运算符	IS NULL	成绩 IS NULL
集合运算符	IN、 NOT IN	专业班级 IN ("计算机", "英语")

【例5.17】 查询dab中的男生学号、姓名、性别。

```
SELECT 学号, 姓名, 性别 FROM dab WHERE 性别="男"
```

结果如下图所示。



学号	姓名	性别
20050005	张广田	男
20050025	陈兴华	男
20050030	陈宗贵	男
20050035	张伟	男
20050040	张光辉	男
20043679	邬德斋	男

【例5.18】 查询dab中计算机1班的男生姓名。

```
SELECT 姓名 FROM dab WHERE 性别="男" AND 专业班级="计算机1班"
```

结果是： 张广田

【例5.19】 查询xkcjb中成绩在70~90分范围内的学生。

```
SELECT * FROM xkcjb WHERE 成绩 BETWEEN 70 AND 90
```

结果如图所示。等价于：

```
SELECT * FROM xkcjb WHERE 成绩>=70 AND 成绩<=90
```



学号	课程号	学期	成绩	重考成绩
20050010	030	1	72.0	0.0
20050025	034	2	87.0	0.0
20045643	033	2	81.5	0.0

【例5.20】 查询dab中所有姓“张”的学生。

```
SELECT * FROM dab WHERE 姓名 LIKE "张%"
```

结果如下图所示。



学号	姓名	性别	出生日期	专业班级	照片	简历
20050005	张广田	男	10/11/87	计算机1班	gen	memo
20050035	张伟	男	03/24/87	英语3班	gen	memo
20050040	张光辉	男	03/07/87	英语3班	gen	memo

LIKE子句中可以用通配符：下划线“_”，表示任何一个字符；百分号符“%”，表示一串字符。

【例5.21】 查询xkcjb中选修课程号为“001”的学生。

```
SELECT * FROM xkcjb WHERE 课程号 IN ("001")
```

结果有3条记录，学号分别为“20050010”、“20050020”和“20045643”。

4) 排序查询

通过ORDER BY子句可以实现查询结果的排序输出，允许按一列或多列排序。

格式：

```
ORDER BY <排序选项1> [ASC|DESC] [, <排序选项2>
[ASC|DESC] ...]
```

其中，ASC表示升序排序（缺省方式），DESC表示降序排序。

【例5.22】 按课程的学分升序查询出全部课程信息。

```
SELECT * FROM kcb ORDER BY 学分
```

【例5.23】 按学号升序，相同学号再按成绩降序查询出全部学生成绩情况。

```
SELECT * FROM xkcjb ORDER BY 学号,成绩 DESC
```

结果如下图所示。



The screenshot shows a window titled "查询" (Query) displaying the results of a SQL query. The table has six columns: 学号 (Student ID), 课程号 (Course ID), 学期 (Semester), 成绩 (Score), and 重考成绩 (Retake Score). The data is sorted by student ID in ascending order, and for each student, the records are sorted by score in descending order.

学号	课程号	学期	成绩	重考成绩
20043567	033	2	88.0	0.0
20043567	034	2	54.0	62.0
20045643	033	2	76.5	0.0
20045643	001	1	50.0	75.5
20050005	010	1	58.0	70.0
20050010	010	1	100.0	0.0
20050010	023	1	98.5	0.0
20050010	030	1	67.0	0.0
20050010	001	1	38.0	68.0
20050020	001	1	45.0	70.0
20050025	030	1	88.0	0.0
20050025	034	2	82.0	0.0
20050035	023	2	95.5	0.0

5) 带计算函数的查询

在SELECT语句中，使用SQL语言提供的一些查询计算函数，可以增强查询功能。基本的查询计算函数的格式及功能如下表所示。

函数的格式	函数功能
COUNT (*)	计算记录条数
SUM(字段名)	求字段名所指定字段值的总和
AVG(字段名)	求字段名所指定字段的平均值
MAX(字段名)	求字段名所指定字段的最大值
MIN(字段名)	求字段名所指定字段的最小值

【例5.24】 求所有课程的总学分。

```
SELECT SUM(学分) FROM kcb
```

结果是：

Sum_学分

31

【例5.25】 求学生总人数。

```
SELECT COUNT(*) FROM dab
```

结果是：

Cnt

12

【例5.26】 求学生平均年龄。

```
SELECT AVG (YEAR (DATE ()) - YEAR (出生日期)) FROM dab
```

结果是:

```
Avg_exp_1
```

```
19.33
```

【例5.27】 求选修课程的最高成绩和最低成绩。

```
SELECT MAX (成绩) , MIN (成绩) FROM xkcjb
```

结果是:

```
Max_成绩      Min_成绩
```

```
100.0         38.0
```


6) 用别名输出列标题（字段名）

将列名用含义更明确的别名输出，便于理解，可以通过 AS 关键字实现。

格式： AS <列别名>

功能： 将查询输出的列名以列别名输出。其中，列别名若为常字符串，则无须加定界符

【例5.28】 求选修课程的最高成绩和最低成绩。

```
SELECT MAX (成绩) AS 最高成绩, ;  
        MIN(成绩) AS 最低成绩 FROM xkcjb
```

结果是：

最高成绩	最低成绩
------	------

100.0	38.0
-------	------

7) 分组查询

通过GROUP BY子句可以实现分组查询。

格式:

```
GROUP BY <分组字段名1> [, <分组字段名2>...]  
[HAVING <过滤条件>]
```

其中, <分组字段名>可以是表的字段名、字段函数名或标识列的数值型表达式; [HAVING <过滤条件>] 子句进一步限定分组的条件。

注意: 若只有HAVING子句而无GROUP BY子句, 则HAVING子句相当于WHERE子句。

【例5.29】求各门课程的平均成绩。

```
SELECT 课程号, AVG(成绩);
```

```
FROM xkcjb GROUP BY 课程号
```

结果如图所示。



课程号	Avg_成绩
001	44.33
010	79.00
023	97.00
030	77.50
033	82.25
034	68.00

【例5.30】求至少选修了2门课程的学生们的信息。

```
SELECT dab.学号, 姓名, 课程号 FROM dab, xkcjb WHERE
```

```
dab.学号=xkcjb.学号;
```

```
GROUP BY dab.学号 HAVING COUNT(*) >=2
```

结果如图所示。



学号	姓名	课程号
20043567	那措央中	033
20045643	崔婷婷	033
20050010	蜂雪燕	030
20050025	陈兴华	034

8) 多表联接查询

SELECT语句可以实现对Visual FoxPro 6.0的4种联接类型的查询：

- **内部联接 (Inner Join)** : 只有2个表的字段都满足联接条件时，才将此记录选入查询结果中。
- **左联接 (Left Outer Join)** : 联接条件左边表中的记录都包含在查询结果中，而右边表中的记录只有满足联接条件时，才选入查询结果中。
- **右联接 (Right Outer Join)** : 与左联接正好相反。
- **完全联接 (Full Join)** : 无论2个表中的记录是否满足联接条件，都将全部记录选入查询结果中。

通常有3种方法实现多表联接查询，一是在查询命令中显示地指定联接类型；二是在查询命令中使用WHERE子句；三是使用嵌套查询。

(1) 指定联接类型实现多表间的联接查询

格式：SELECT ...

FROM <表或视图> INNER|LEFT|RIGHT|FULL JOIN <表或视图>

ON <联接条件>

WHERE ...

功能：通过指定的联接类型建立多表间的联接

注意：若SELECT后要查询的列名（字段名）在2个表中都有，则必须采用“表名.字段名”，若字段名唯一，则可仅写出字段名。

【例5.31】显示dab和xkcjb中成绩不及格的学生姓名、学号、课程号及成绩。

```
SELECT dab.学号, 姓名, 课程号, 成绩 FROM xkcjb;  
INNER JOIN dab ON dab.学号=xkcjb.学号;  
WHERE 成绩<60
```

结果如图所示。

学号	姓名	课程号	成绩
20050010	蜂雪燕	001	38.0
20050020	冯艳	001	45.0
20043567	那措央中	034	54.0
20050005	张广田	010	58.0
20045643	崔婷婷	001	50.0

(2) 用WHERE子句实现多表间的联接查询：用WHERE子句实现多表间的联接查询时，无须直接指明联接类型（隐含是内部联接），只需把联接条件直接写入WHERE子句即可。

【例5.32】用WHERE子句完成例5.31的功能。

```
SELECT dab. 学号, 姓名, 课程号, 成绩 FROM xkcjb, dab ;  
WHERE dab. 学号=xkcjb. 学号 AND 成绩<60
```

(3) 使用嵌套查询实现多表间的联接查询：在SQL语言中，由SELECT、FROM、WHERE语句组成一个查询块。嵌套查询就是将第2个查询块放在第1个查询块的WHERE条件中，形成外层(第1个)查询包含内层(第2个)查询的嵌套查询。

外层查询也称为主查询、父查询，内层查询也称为下层查询、子查询。系统对嵌套查询的处理过程是先做子查询，在子查询的基础上再做主查询。

在子查询中还可以再包含更下层的查询，从而形成层嵌套查询。

【例5.33】 查询成绩在90分及以上的学生姓名。

```
SELECT 姓名 FROM dab WHERE 学号 IN;
```

```
(SELECT 学号 FROM xkcjb WHERE 成绩>=90)
```

结果是蜂雪燕和张伟。

9) 查询中的几个特殊选项

(1) 使用TOP <数值表达式> [PERCENT] 短语显示部分结果

【例5.34】 显示学分较高的2门课程的信息。

```
SELECT * TOP 2 FROM KCB ORDER BY 学分 DESC
```

注意： TOP短语要与ORDER BY短语同时使用才有效。

【例5.35】 显示学分最低的20%课程的信息。

```
SELECT * TOP 20 PERCENT FROM KCB ORDER BY 学分
```

(2) 使用INTO ARRAY <数组名>将查询结果存放到数组中。

【例5.36】 查询学生档案信息并将结果放入数组TEMP中。

```
SELECT * FROM DAB INTO ARRAY TEMP
```

(3) 使用INTO CURSOR <文件名>将查询结果存放到临时表中。

【例5.37】 查询课程表信息并将结果放入临时表TEMP中。

```
SELECT * FROM KCB INTO CURSOR TEMP
```

(4) 使用INTO DBF|TABLE <文件名>将查询结果存放到永久表中。

【例5.38】 查询课程表信息并将结果放入永久表TEMP中。

```
SELECT * FROM KCB INTO DBF TEMP
```

5.3.4 使用SQL语句更新表

使用SQL语句对表中记录进行操作，无须用USE命令打开表，便可完成有关记录的插入、更新、删除等相关操作。SQL语句与前面介绍的添加记录、修改记录及删除记录命令相比较，在操作方式和方法上有很大的不同。SQL语句在只对一条记录的操作时要简捷得多，但对多条记录操作并不方便。

1) 插入记录

格式：INSERT INTO<表名>(<字段名1> [， <字段名2>...]);
VALUES (<表达式1> [， <表达式2>...])

功能：在指定表末尾添加一条新记录。

【例5.39】向表dab添加一条记录。

```
INSERT INTO dab (学号, 姓名, 性别, 出生日期, 专业班级);  
VALUES ("20060002", "张三", "男", {^1987 10 10}, "计算机1班")
```

结果如图所示



学号	姓名	性别	出生日期	专业班级
20050005	张广田	男	10/11/87	计算机1班
20050010	蜂雪燕	女	07/02/87	计算机1班
20050015	何芳	女	05/06/87	计算机1班
20050020	冯艳	女	09/09/87	计算机1班
20050025	陈兴华	男	02/14/87	英语3班
20050030	陈宗贵	男	07/02/87	英语3班
20050035	张伟	男	03/24/87	英语3班
20050040	张光辉	男	03/07/87	英语3班
20043567	那措央中	女	06/30/86	机械制造4班
20043679	鄂德斋	男	04/26/86	机械制造4班
20045643	崔婷婷	女	06/08/86	机械制造4班
20047634	冯丽娟	女	06/30/86	机械制造4班
20060002	张三	男	10/10/87	计算机1班

2) 更新记录

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>]
```

功能: 更新表中满足条件记录的字段值, 即用<表达式>的值替换对应的<字段名>的值。

【例5.40】将dab表中的姓名“张三”改为“李四”。

```
UPDATE dab SET 姓名="李四" WHERE 姓名="张三"
```

结果如下图所示。



学号	姓名	性别	出生日期	专业班级
20050005	张广田	男	10/11/87	计算机1班
20050010	蜂雪燕	女	07/02/87	计算机1班
20050015	何芳	女	05/06/87	计算机1班
20050020	冯艳	女	09/09/87	计算机1班
20050025	陈兴华	男	02/14/87	英语3班
20050030	陈宗贵	男	07/02/87	英语3班
20050035	张伟	男	03/24/87	英语3班
20050040	张光辉	男	03/07/87	英语3班
20043567	那措央中	女	06/30/86	机械制造4班
20043679	邬德斋	男	04/26/86	机械制造4班
20045643	崔婷婷	女	06/08/86	机械制造4班
20047634	冯丽娟	女	06/30/86	机械制造4班
20060002	李四	男	10/10/87	计算机1班

3) 删除记录

格式:

```
DELETE FROM <表名> [WHERE <条件表达式>]
```

功能: 逻辑删除表中满足条件的记录。

注意: SQL语句的删除记录语句是逻辑删除, 如果要物理删除记录需要继续使用PACK命令。

【例5.41】逻辑删除表dab中姓名为“李四”的记录。

```
DELETE FROM dab WHERE 姓名="李四"
```

结果如图所示



学号	姓名	性别	出生日期	专业班级
20050005	张广田	男	10/11/87	计算机1班
20050010	蜂雪燕	女	07/02/87	计算机1班
20050015	何芳	女	05/06/87	计算机1班
20050020	冯艳	女	09/09/87	计算机1班
20050025	陈兴华	男	02/14/87	英语3班
20050030	陈宗贵	男	07/02/87	英语3班
20050035	张伟	男	03/24/87	英语3班
20050040	张光辉	男	03/07/87	英语3班
20043567	那措央中	女	06/30/86	机械制造4班
20043679	鄂德斋	男	04/26/86	机械制造4班
20045643	崔婷婷	女	06/08/86	机械制造4班
20047634	冯丽娟	女	06/30/86	机械制造4班
20060002	李四	男	10/10/87	计算机1班

本章小结

本章系统地介绍了VFP中关于建立视图与数据查询以及使用嵌入到VFP中的SQL语言实现数据库操作的内容，这些都是VFP的特色所在。其中SQL语言尤其重要，在以后的程序设计过程中可以通过使用SQL语言简化大量的程序代码。学好这一章，是掌握VFP的关键，因此应引起足够重视。

8、多表联接查询

SELECT语句可以实现对Visual FoxPro 6.0的4种联接类型的查询：

- ❖ 内部联接（**Inner Join**）：只有两个表的字段都满足联接条件时，才将此记录选入查询结果中。
- ❖ 左联接（**Left Outer Join**）：联接条件左边表中的记录都包含在查询结果中，而右边表中的记录只有满足联接条件时，才选入查询结果中。
- ❖ 右联接（**Right Outer Join**）：与左联接正好相反。
- ❖ 完全联接（**Full Join**）：无论两个表中的记录是否满足联接条件，都将全部记录选入查询结果中。

通常有三种方法实现多表联接查询，一是在查询命令中显示地指定联接类型；二是在查询命令中使用**WHERE**子句；三是使用嵌套查询。

(1) 指定联接类型实现多表间的联接查询

格式：**SELECT**

FROM <表或视图> **INNER|LEFT|RIGHT|FULL JOIN** <表或视图>

ON <联接条件>

WHERE

(2) 用**WHERE**子句实现多表间的联接查询

用**WHERE**子句实现多表间的联接查询时，无须直接指明联接类型（隐含是内部联接），只须把联接条件直接写入**WHERE**子句即可。

(3) 使用嵌套查询实现多表间的联接查询

在**SQL**语言中，由**SELECT**、**FROM**、**WHERE**语句组成一个查询块。嵌套查询就是将第2个查询块放在第1个查询块的**WHERE**条件中，形成外层(第1个)查询包含内层(第2个查询的嵌套查询。外层查询也称为主查询、父查询，内层查询也称为下层查询、子查询。系统对嵌套查询的处理过程是先做子查询，在子查询的基础上再做主查询。在子查询中还可以再包含更下层的查询，从而形成层嵌套查询。

9. 查询中的几个特殊选项

(1)使用TOP <数值表达式> [PERCENT]短语显示部分结果

【例 5.35】 显示学分较高的两门课程的信息。

```
SELECT * TOP 2 FROM KCB ORDER BY 学分 DESC
```

注意：TOP短语要ORDER BY短语同时使用才有效。

【例 5.36】 显示学分最低的那20%课程的信息。

```
SELECT * TOP 20 PERCENT FROM KCB ORDER BY 学分
```

(2)使用INTO ARRAY <数组名>将查询结果存放到数组中

【例 5.37】 查询学生档案信息并将结果放入数组TEMP中。

```
SELECT * FROM DAB INTO ARRAY TEMP
```

(3)使用INTO CURSOR <文件名>将查询结果存放到临时表中

【例 5.38】 查询课程表信息并将结果放入临时表TEMP中。

```
SELECT * FROM KCB INTO CURSOR TEMP
```

(4)使用INTO DBF|TABLE <文件名>将查询结果存放到永久表中

【例 5.39】 查询课程表信息并将结果放入永久表TEMP中。

```
SELECT * FROM KCB INTO DBF TEMP
```

5.3.4 使用SQL语句更新表

1、插入记录

格式: **INSERT INTO**<表名>(<字段名1>[, <字段名2>.....]);
VALUES(<表达式1>[, <表达式2>.....])

功能: 在指定表末尾添加一条新记录。

2、更新记录

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>]

功能: 更新表中满足条件记录的字段值, 即用<表达式>的值替换对应的<字段名>的值。

3、删除记录

格式：

DELETE FROM <表名> [WHERE <条件表达式>]

功能：逻辑删除表中满足条件的记录。

注意：**SQL**语句的删除记录语句是逻辑删除，如果要物理删除记录需要继续使用**PACK**命令。

本章小结：本系统地介绍了VFP中关于建立视图与数据查询以及使用嵌入到VFP中的SQL语言实现数据库操作的内容，这些都是VFP的特色所在。其中SQL语言尤其重要，在以后的程序设计过程中可以通过使用SQL语言简化大量的程序代码。读者学好这一章，是掌握VFP关键，因此应引起读者的足够重视。