第5章 大数据下数据库基础

大数据下数据库技术是通过研究数据库的结构、存储、设计、管理以及应用的基本理论和实现方法，并利用这些理论来实现对数据库中的数据进行处理、分析和理解的技术。即：数据库技术是研究、管理和应用数据库的一门软件科学。

数据库技术已经成为计算机应用中必须掌握的重要技术之一。数据库技术研究和管理的对象是数据，所以数据库技术所涉及的具体内容主要包括：通过对数据的统一组织和管理，按照指定的结构建立相应的数据库和数据仓库；利用数据库管理系统和数据挖掘系统设计出能够实现对数据库中的数据进行添加、修改、删除、处理、分析、理解、报表和打印等多种功能的数据管理和数据挖掘应用系统；并利用应用管理系统最终实现对数据的处理、分析和理解。

5.1 数据库技术基础

5.1.1 数据库技术产生背景及其发展

数据库技术产生于20世纪60年代末70年代初，其主要目的是有效地管理和存取大量的数据资源。数据库技术主要研究如何存储、使用和管理数据。数据库技术和计算机网络技术的发展相互渗透、相互促进，已成为当今计算机领域发展迅速、应用广泛的两大领域。

数据库最初是在大公司或大机构中用作大规模事务处理的基础，后来随着个人计算机的普及，数据库技术被移植到PC机上，供单用户个人数据库应用。如今，数据库正在Internet和内联网中广泛使用。终端用户开始使用互联网技术将独立的计算机连接成网络，终端之间共享数据库，形成了一种新型的多用户数据处理，称为客户机/服务器数据库结构。如今，数据库技术正在被用来同Internet技术相结合，以便在机构内联网、部门局域网甚至WWW上发布数据库数据。

5.1.2 基本概念

数据库技术涉及到许多基本概念，主要包括：信息、数据、数据处理、数据库、数据库管理系统以及数据库系统等。

(1) 数据(Data)：是用于描述现实世界中各种具体事物或抽象概念的，可存储并具有明确意义的符号，包括数字、文字、图形和声音等。数据处理是指对各种形式的数据进行收

集、存储、加工和传播的一系列活动的总和。其目的之一是从大量的，原始的数据中抽取、推导出对人们有价值的信息以作为行动和决策的依据；目的之二是为了借助计算机技术科学地保存和管理复杂的，大量的数据，以便人们能够方便而充分地利用这些宝贵的信息资源。

(2) 数据库(DataBase，DB)：是存储在计算机辅助存储器中的、有组织的、可共享的相关数据集合。数据库具有如下特性：

① 数据库是具有逻辑关系和确定意义的数据集合。

② 数据库是针对明确的应用目标而设计、建立和加载的。每个数据库都具有一组用户，并为这些用户的应用需求服务。

③ 一个数据库反映了客观事物的某些方面，而且需要与客观事物的状态始终保持一致。

(3) 数据库管理系统(DataBase Management System，DBMS)：是对数据库进行管理的系统软件，它的职能是有效地组织和存储数据，获取和管理数据，接受和完成用户提出的各种数据访问请求。能够支持关系型数据模型的数据库管理系统，称为关系型数据库管理系统(Relational DataBase Management System，RDBMS)。

RDBMS 的基本功能包括以下 4 个方面：

① 数据定义功能：RDBMS 提供了数据定义语言(Data Definition Language，DDL)，利用 DDL 可以方便地对数据库中的相关内容进行定义。例如，对数据库、表、字段和索引进行定义，创建和修改。

② 数据操作功能：RDBMS 提供了数据操作语言(Data Manipulation Language，DML)，利用 DML 可以实现在数据库中插入、修改和删除数据等基本操作。

③ 数据查询功能：RDBMS 提供了数据查询语言(Data Query Language，DQL)，利用 DQL 可以实现对数据库的数据查询操作。

④ 数据控制功能：RDBMS 提供了数据控制语言(Data Control Language，DCL)，利用 DCL 可以完成数据库运行控制功能，包括并发控制(即处理多个用户同时使用某些数据时可能产生的问题)，安全性检查，完整性约束条件的检查和执行，数据库的内部维护(例如索引的自动维护)等。RDBMS 的上述许多功能都可以通过结构化查询语言(Structured Query Language，SQL)来实现的，SQL 是关系数据库中的一种标准语言，在不同的 RDBMS 产品中，SQL 中的基本语法是相同的。此外，DDL、DML、DQL 和 DCL 也都属于 SQL。

(4) 数据库系统的组成：采用了数据库技术的完整的计算机系统就是数据库系统。

主要包括：

① 计算机硬件系统：主机、键盘、显示器、硬盘、光驱、鼠标、打印机等。

② 计算机软件系统：操作系统、数据库管理系统及数据库应用系统等。

③ 数据库：按一定法则存储在计算机外存储器中的大批数据。它不仅包括描述事物的数据本身，而且还包括相关事物之间的联系。

④ 用户：包括最终用户、数据库应用系统开发人员和数据库管理员 3 类。最终用户指通过应用系统的用户界面使用数据库的人员，他们一般对数据库知识了解不多。数据库应用系统开发人员包括系统分析员、系统设计员和程序员。系统分析员负责应用系统的分析，他们和用户、数据库管理员相配合，参与系统分析；系统设计员负责应用系统设计和数据库设计；程序员则根据设计要求进行编码。数据库管理员是数据管理机构的一组人员，他们

负责对整个数据库系统进行总体控制和维护，以保证数据库系统的正常运行。

（5）数据库系统的特点：数据库系统是指引进数据库后的计算机系统，实现有组织地、动态地存储大量相关数据，提供数据处理和信息资源共享的便利手段。

一个数据库系统的主要特点如下：

① 实现数据共享，减少冗余。在数据库系统中，对数据的定义和描述已经从应用程序中分离出来，通过数据库系统来同意管理。数据的最小上访问单位是字段，即可以按字段的名称存取某一个或者一组字段，也可以存取一条记录或一组记录。

② 采用特定的数据模型。数据库中的数据是有结构的，这种结构由数据库管理系统所支持的数据模型表现出来。数据库系统不仅可以表示事物内部各数据项之间的联系，而且可以表示事物与事物之间的联系，从而反映出实现世界事物之间的联系。因此，任何数据库管理系统都支持一种抽象的数据模型。

③ 具有较高的数据独立性。在数据库系统中，数据库管理系统提供映像功能，实现了应用程序对数据的总体逻辑结构、物理存储结构之间较高的独立性。用户只以简单的逻辑结构来操作数据，无需考虑数据在存储器上的物理位置与结构。

④ 有统一的数据控制功能。数据库可以被多个用户或应用程序共享，数据的存取往往是并发的，即多个用户同时使用同一个数据库。数据库管理系统必须提供必要的保护措施，包括并发访问控制功能、数据的安全控制功能和实际的完整性控制功能。

5.1.3 数据库管理技术的发展阶段

数据库技术是现代信息科学与技术的重要组成部分，是计算机数据处理与信息管理系统的核心。数据库技术研究和解决了计算机信息处理过程中大量数据有效地组织和存储的问题，在数据库系统中减少数据存储冗余、实现数据共享、保障数据安全以及高效地检索数据和处理数据。数据库技术的根本目标是要解决数据的共享问题。发展数据管理技术是对数据进行分类、组织、编码、输入、存储、检索、维护和输出的技术。数据管理技术的发展大致经过了以下三个阶段：人工管理阶段；文件系统阶段；数据库系统阶段。

（1）人工管理阶段：20 世纪 50 年代以前，计算机主要用于数值计算，从当时的硬件看，外存只有纸带、卡片、磁带，没有直接存取设备；从软件看（实际上，当时还未形成软件的整体概念），没有操作系统以及管理数据的软件；从数据看，数据量小，数据无结构，由用户直接管理，且数据间缺乏逻辑组织，数据依赖于特定的应用程序，缺乏独立性。

（2）文件系统阶段：50 年代后期到 60 年代中期，出现了磁鼓、磁盘等数据存储设备。新的数据处理系统迅速发展起来，这种数据处理系统是把计算机中的数据组织成相互独立的数据文件，系统可以按照文件的名称对其进行访问，对文件中的记录进行存取，并可以实现对文件的修改、插入和删除，这就是文件系统。文件系统实现了记录内的结构化，即给出了记录内各种数据间的关系。但是，文件从整体来看却是无结构的。其数据面向特定的应用程序，因此数据共享性、独立性差，且冗余度大，管理和维护的代价也很大。

（3）数据库系统阶段：60 年代后期，出现了数据库这样的数据管理技术。数据库的特点是数据不再只针对某一特定应用，而是面向全组织，具有整体的结构性，共享性高，冗余度小，具有一定的程序与数据间的独立性，并且实现了对数据的统一控制。

5.1.4　数据模型

数据模型是现实世界在数据库中的抽象，也是数据库系统的核心和基础。数据模型通常包括3个要素：

(1) 数据结构：数据结构主要用于描述数据的静态特征，包括数据的结构和数据间的联系。

(2) 数据操作：数据操作是指在数据库中能够进行的查询、修改、删除现有数据或增加新数据的各种数据访问方式，并且包括数据访问相关的规则。

(3) 数据完整性约束：数据完整性约束由一组完整性规则组成。

数据库理论领域中最常见的数据模型主要有层次模型、网状模型和关系模型3种。

(1) 层次模型(Hierarchical Model)：层次模型使用树形结构来表示数据以及数据之间的联系。

(2) 网状模型(Network Model)：网状模型使用网状结构表示数据以及数据之间的联系。

(3) 关系模型(Relational Model)：关系模型是一种理论最成熟，应用最广泛的数据模型。在关系模型中，数据存放在一种称为二维表的逻辑单元中，整个数据库又是由若干个相互关联的二维表组成的。

关系数据库管理系统有很多，如：Sybase、Oracle和SQL Server，而不同关系数据库使用不同查询语言，就会带来很多问题，唯一的解决方法就是标准的语言SQL。

SQL全称是结构化查询语言(Structured Query Language)，是对数据库中的数据进行组织、管理和检索的工具。

5.1.5　SQL概述

在20世纪80年代初，ANSI开始着手制定SQL标准。目前，各主流数据库产品采用的SQL标准是1992年制定的SQL92，由于它功能丰富、语言简洁而备受计算机界欢迎。

按照ANSI的规定，SQL被作为关系数据库的标准语言。SQL语句可以用来执行各种各样的操作。SQL语言由三部分组成，它们是：

数据定义语言DDL(Data Definition Language)；

数据操作语言DML(Data Manipulation Language)；

数据控制语言DCL(Data Control Language)。

SQL语言具有如下特点。

1. 高度集成化

SQL语言集数据定义、数据查询、数据操纵和数据控制功能于一体，可以独立完成数据库操作和管理的全部操作，为数据库应用系统的开发提供了良好的手段。

2. 非过程化

SQL语言是一种高度非过程化的语言。它不必告诉计算机怎么做，只要提出做什么，

SQL 语言就可以将要求交给系统，自动完成全部工作从而大大减轻了用户的负担，还有利于提高数据独立性。

3. 简洁易学

SQL 语言功能很强，但却非常简洁，它只有为数不多的 9 条命令，如表 5-1 所示。另外 SQL 的语法也非常简单，它很接近英语自然语言，因此容易学习和掌握。

表 5-1 SQL 命令动词

SQL 功能	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

4. 用法灵活

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。需要注意的是，SQL 虽然在各种数据库产品中得到了广泛的支持，但迄今为止，它只是一种建议标准，各种数据库产品中所实现的 SQL 语法虽然基本是一致的，但还是略有差异。

5.2 SQL 的数据定义功能

标准 SQL 的数据定义功能非常广泛，包括数据库、表、视图、存储过程、规则及索引的定义等。数据定义语言由 CREATE（创建），DROP（删除），ALTER（修改）三个命令组成。这三个命令针对不同的数据对象分别有 3 条命令，如操作数据表时可使用 CREATE、DROP 和 ALTER 命令，操作视图也可以使用这 3 条命令。

5.2.1 建立表结构

1. 命令格式

```
CREATE TABLE|DBF <表名 1> [NAME <长表名>][FREE]
(<字段名 1> <类型>(<宽度>[,<小数位数>])[NULL|NOT NULL]
[CHECK <条件表达式 1>[ERROR <出错显示信息>]] [DEFAULT <表达式 1>]
[PRIMARY KEY | UNIQUE]REFERENCES <表名 2>[TAG <标识 1>]
[<字段名 2><类型>(<宽度>[,<小数位数>])[NULL|NOT NULL]
[CHECK <条件表达式 2>[ERROR <出错显示信息>]] [DEFAULT <表达式 2>]
[PRIMARY KEY | UNIQUE]REFERENCES <表名 3>[TAG <标识 2>]
…… )|FROM ARRAY <数组名>
```

2. 命令说明

（1）CREATE TABLE 或 CREATE DBF 功能等价，都是建立表。

（2）FREE：指明所创建的表为自由表。默认在数据库未打开时创建的表是自由表，在

数据库打开时创建的表为数据库表。

(3) 字段名1、字段名2…：所要建立的新表的字段名，在语法格式中，两个字段名之间的语法成分都是对一个字段的属性说明，包括：

① 类型——说明字段类型，可选项的字段类型见表5-2。

表5-2　数据类型说明

字段类型	字段宽度	小数位	说　明
C	N	—	字符型字段的宽度位N
D	—	—	日期型(Date)
T	—	—	日期时间型(Datetime)
N	N	D	数值字段类型(Numeric)，宽度位N，小数位D
F	N	D	浮点数值字段类型(Float)，宽度位N，小数位D
I	—	—	整数类型(Integer)
B	—	D	双精度类型(Double)
Y	—	—	货币型(Currency)
L	—	—	逻辑型(Logic)
M	—	—	备注型(Memo)
G	—	—	通用型(General)

② 宽度及小数位数——字段宽度及小数位数见表5-2说明。

③ NULL、NOT NULL——该字段是否允许“空值”，其默认值为NULL，即允许“空”值。

④ CHECK <条件表达式>——用来检测字段的值是否有效，这是实行数据库的一种完整性检查。

⑤ ERROR <出错显示信息>——当完整性检查有错误，即条件表达式的值为假时的提示信息。

⑥ DEFAULT <表达式>——为一个字段指定的默认值。

⑦ PRIMARY KEY——指定该字段为关键字段，它能保证关键字段的唯一性和非空性，非数据库表不能使用该参数。

⑧ UNIQUE——指定该字段为一个候选关键字段。注意，指定为关键字或候选关键字的字段都不允许出现重复值，这称为对字段值的唯一性约束。

⑨ REFERENCES <表名>——这里指定的表作为新建表的永久性父表，新建表作为子表。

⑩ TAG <标识>——父表中的关联字段，若缺省该参数，则默认父表的主索引字段作为关联字段。

(4) FROM ARRAY <数组名>：根据指定数组的内容建立表，数组元素依次是字段名、类型等。

从以上命令格式可以看出，除了建立表的基本功能外，它还包括满足实体完整性的主关键字(主索引)PRIMARY KEY、定义域完整性的CHECK约束及出错提示信息ERROR、定义默认值的DEFAULT等。另外还有描述表之间联系的FOREIGN KEY和REFERENCES等。

【例 5-1】 使用 SQL 命令建立学生管理数据库，其中包含 3 个表：学生表、选课表和课程表。操作步骤如下：

（1）用 CREATE 命令建立数据库。

```
CREATE DATABASE  D:\学生管理
```

（2）用 CREATE 命令建立学生表。

```
CREATE TABLE 学生(学号 C(6)  PRIMARY KEY,姓名 C(8),性别 C(2),出生日期 D,;
少数民族否 L,籍贯 C(10),入学成绩  N(3,0)  CHECK(入学成绩>0) ERROR "成绩应该大于 0!",简
历  M,照片 G NULL)
```

其中指定学号是主关键字，设置入学成绩字段有效性规则。

（3）建立课程表。

```
CREATE TABLE 课程(课程号 C(6)  PRIMARY KEY,课程名 C(10),学分 N(1))
```

其中指定课程号是主关键字。

（4）建立选课表。

```
CREATE TABLE 选课(学号 C(6),课程号 C(6);
成绩 N(3,0)  CHECK(成绩>=0 AND 成绩<=100);
ERROR "成绩值的范围 0～100!" DEFAULT 60;
FOREIGN KEY 学号 TAG 学号 REFERENCES 学生;
FOREIGN KEY 课程号 TAG 课程号 REFERENCES 课程)
```

注意：

用 SQL CREATE 命令新建的表自动在最小可用工作区打开，并可以通过别名引用，新表的打开方式为独占方式，忽略 SET EXCLUSIVE 的当前设置。

如果建立自由表（当前没有打开的数据库或使用了 FREE），则很多选项在命令中不能使用，如 NAME、CHECK、DEFAULT、FOREIGN KEY、PRIMARY KEY 和 REFERENCES 等。

步骤（4）命令有两个 FOREIGN KEY…REFERENCES…短语，分别说明了学生表与选课表、课程表与选课表之间的联系。

以上所有建立表的命令执行完后，可以在数据库设计器中看到各个表以及它们之间的联系，如图 5-1 所示，然后可以用其他的方法来编辑参照完整性，进一步完善数据库的设计。

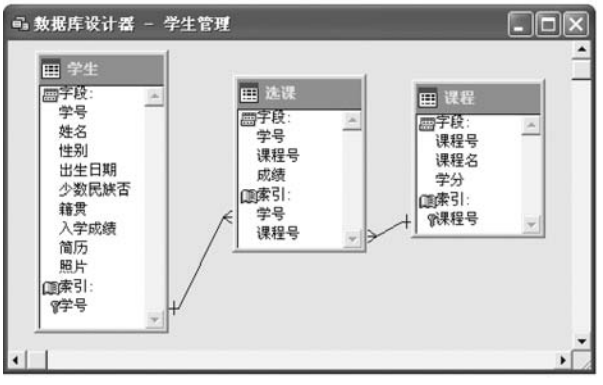

图 5-1　数据库设计器中各表与表间的联系

5.2.2　删除表

当某个表不再需要时，可以使用 DROP TABLE 语句删除它。

基本表定义一旦删除，表中的数据、此表上建立的索引和视图都将自动被删除掉。因此执行删除基本表的操作一定要格外小心。

删除表的 SQL 命令格式是：

```
DROP TABLE <表名>
```

DROP TABLE 命令直接从磁盘上删除所指定的表文件。如果指定的表文件是数据库中的表并且相应的数据库是当前数据库，则可从数据库中删除表。否则虽然从磁盘上删除了表文件，但是记录在数据库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，最好应使数据库是当前打开的数据库，在数据库中进行操作。

例如：删除“学生管理”数据库的“课程”表：

```
OPEN DATABASE 学生管理
DROP TABLE 课程
```

5.2.3　修改表结构

如果需要修改已建立好的表结构，SQL 语言提供了 ALTER TABLE 语句，该命令有 3 种格式。

1. 格式 1

```
ALTER TABLE <表名 1 >
ADD|ALTER [COLUMN] <字段名><字段类型>[(<宽度>[,<小数位数>])]
[NULL | NOT NULL][CHECK <逻辑表达式> [ERROR <出错显示信息>]]
[DEFAULT <表达式>][PRIMARY KEY|UNIQUE]
[REFERENCES <表名 2 >[TAG <标识名>]]
```

该格式可以添加字段、修改字段的类型、宽度、有效性规则、错误信息、默认值，定义主关键字和联系等。

【例 5-2】 为选课表增加一个字段：平时成绩 N(5,1)。

```
ALTER TABLE 选课 ADD 平时成绩 N(5,1)
```

【例 5-3】 将课程表的课程名字段的宽度由原来的 10 改为 20。

```
ALTER TABLE 课程 ALTER 课程名 C(20)
```

2. 格式 2

```
ALTER TABLE <表名>
ALTER [COLUMN] <字段名> [NULL|NOT NULL]
[SET DEFAULT <表达式>[SET CHECK <逻辑表达式> [ERROR <出错显示信息>]]
[DROP DEFAULT][DROP CHECK]
```

该格式命令主要用于定义、修改和删除有效性规则以及默认值定义。命令说明：

（1）SET DEFAULT <表达式> 用来设置默认值；SET CHECK <逻辑表达式> [ERROR <出错显示信息>]短语用来设置约束条件。

（2）DROP DEFAULT 短语用来删除默认值；DROP CHECK 短语用来删除约束条件。

（3）本命令仅仅适合数据库表。

【例 5-4】 为学生表的入学成绩字段添加有效性规则。

```
ALTER TABLE 学生 ALTER 入学成绩 SET CHECK(入学成绩>=0) ;
ERROR "入学语成绩应大于 0!"
```

【例 5-5】 删除平时成绩字段的有效性规则并设置字段默认值为 80。

```
ALTER TABLE 选课 ALTER 平时成绩 DROP CHECK
ALTER TABLE 选课 ALTER 平时成绩 SET DEFAULT 80
```

3. 格式 3

```
ALTER TABLE <表名> [DROP [COLUMN] <字段名>]
[SET CHECK <逻辑表达式>[ERROR <出错显示信息>]]
[DROP CHECK]
[ADD PRIMARY KEY <表达式> TAG <索引标识> [FOR <逻辑表达式>]]
[DROP PRIMARY KEY]
[ADD UNIQUE <表达式> [TAG <索引标识> [FOR <逻辑表达式>]
[DROP UNIQUE TAG <索引标识>
[ADD FOREIGN KEY <表达式> TAG <索引标识> [FOR <逻辑表达式>]]
REFERENCES <表名 2>[TAG <索引标识>]]
[DROP FOREIGN KEY TAG <索引标识>[SAVE]]
[RENAME COLUMN <原字段名> TO <目标字段名>]
```

该格式的命令可以删除指定字段（DROP [COLUMN]）、修改字段名（RENAME COLUMN）、修改指定表的完整性规则，包括主索引、外关键字、候选索引及表的合法值限定的添加与删除。

【例 5-6】 将选课表中平时成绩字段改为平时分。

```
ALTER TABLE 选课 RENAME COLUMN 平时成绩 TO 平时分
```

【例 5-7】 删除选课表的平时分字段。

```
ALTER TABLE 课程 DROP COLUMN 平时分
```

【例 5-8】 在学生表中定义学号和姓名为候选索引。

```
ALTER TABLE 学生 ADD UNIQUE 学号 + 姓名 TAG RAN
```

【例 5-9】 删除学生表的候选索引 RAN。

```
ALTER TABLE 学生 DROP UNIQUE TAG RAN
```

说明：如被删除的字段建立了索引，则必须先将索引删除，然后才能删除该字段。

5.3　SQL 的数据修改功能

SQL 语言的数据修改功能主要有：记录的插入、删除和数据更新等功能，其命令主要有：INSERT、DELETE、UPDATE 命令。

5.3.1　插入记录

格式 1：

```
INSERT INTO <表名>[(字段名 1[<字段名 2>[,…]])] VALUES(<表达式 1>[,<表达式 2>[,…]])
```

该命令在指定的表尾添加一条新记录，其值为 VALUES 后面表达式的值。

当需要插入表中所有字段的数据时，表名后面的字段名可以缺省，但插入数据的格式及顺序必须与表的结构完全吻合；若只需要插入表中某些字段的数据，就需要列出插入数据的字段名，当然相应表达式的数据位置应与之对应。

【例 5-10】 向学生表中添加记录。

```
INSERT INTO 学生 VALUES("231002","杨阳","男",{^1984-07-07},.T.,"北京",680,"",NULL)
INSERT INTO 学生(学号,姓名)  VALUES("231109","李兵")
```

格式 2：

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

该命令在指定的表尾添加一条新记录，其值来自于数组或对应的同名内存变量。

【例 5-11】 已经定义了数组 A(5)，A 中各元素的值分别是：A(1)＝"231013"，A(2)＝"张阳"，A(3)＝"女"，A(4)＝{^1985-01-02}，A(5)＝.F.。利用该数组向学生表中添加记录。

```
INSERT INTO 学生 FROM ARRAY A
```

5.3.2　删除记录

DELETE 可以为指定的数据表中的记录添加删除标记。命令格式是：

```
DELETE  FROM [<数据库名>!] <表名> [WHERE <条件表达式>
```

该命令从指定表中，根据指定的条件逻辑删除记录。如果要真正物理删除记录，在该命令后还必须用 PACK 命令，也可以使用命令 RECALL 恢复逻辑删除的记录。

【例 5-12】 将“学生”表所有男生的记录逻辑删除。

```
DELETE FROM 学生 WHERE 性别 = "男"
```

5.3.3　更新记录

更新记录时对存储在表中的记录进行修改，命令是 UPDATE，也可以对用 SELECT 语

句选择出的记录进行数据更新。命令格式是：

```
UPDATE [<数据库名>!]<表名>
SET <字段名 1>=<表达式 1>[,<字段名 2>=<表达式 2>…] [WHERE <逻辑表达式>]
```

该命令用指定的新值更新记录。

【例 5-13】 将“学生”表中姓名为杨阳的学生的入学成绩改为 600。

```
UPDATE 学生 SET 入学成绩=600 WHERE 姓名="杨阳"
```

【例 5-14】 所有男生的入学成绩加 20 分。

```
UPDATE 选课 SET 入学成绩=入学成绩+20;
    WHERE 学号 IN (SELECT 学号 FROM 学生 WHERE 性别="男")
```

以上命令中，用到了 WHERE 条件运算符 IN 和对用 SELECT 语句选择出的记录进行数据更新。注意 UPDATE 一次只能在单一的表中更新记录。

5.4 SQL 的数据查询

SQL 核心是查询。SQL 的查询命令也称作 SELECT，它的基本形式由 SELECT-FROM-WHERE 查询块组成，多个查询块可以嵌套执行。通过使用 SQL-SELECT 命令，可以对数据源进行各种组合，有效地筛选记录、管理数据、对结果排序、指定输出去向等，无论查询多么复杂，其内容只有一条 SELECT 语句。语法格式如下：

```
SELECT [ALL|DISTINCT] [TOP N [PERCENT]]
[<别名>.]<选项>[AS <显示列名>][,[<别名>.]<选项>[AS <显示列名>]…]
FROM [<数据库名!]<表名>[[AS] <本地别名>]
[[INNER | LEFT [OUTER] | RIGHT[OUTER]|FULL [OUTER]
JOIN <数据库名>!]<表名>[[AS]<本地别名>][ON <联接条件>…]
[[INTO <目标>|[TO FILE <文件名>][ADDITIVE]
|TO PRINTER [PROMPT]|TO SCREEN]]
[PREFERENCE <参照名>][NOCONSOLE][PLAIN][NOWAIT]
[WHERE <联接条件 1>[AND <联接条件 2>…]
[AND|OR <过滤条件 1>[AND|OR <过滤条件 2>…]]]
[GROUP BY <分组列名 1>[,<分组列名 2>…]][HAVING <过滤条件>]
[UNION[ALL]SELECT 命令]
[ORDER BY <排序选项 1>[ASC|DESC][,<排序选项 2>[ASC|DESC]…]]
```

命令功能：根据指定条件从一个或者多个表中检索输出数据。

命令说明：

(1) SELECT 短语指明要在查询结果中输出的字段内容。其中 DISTINCT 用来指定消除输出结果中重复的行，TOP <数值表达式>[PERCENT]用来指定输出的行数或百分比，默认为 ALL。使用短语 TOP 必须要排序，即使用 ORDER BY 短语。

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

(3) WHERE 说明查询条件，即选择元组的条件。

(4) GROUP BY 短语用语对查询结果进行分组，可以利用它进行分组汇总；其中

HAVING 短语用来限定分组必须满足的条件。

(5) ORDER BY 短语用来对查询的结果进行排序。默认为升序,降序必须使用 DESC。

(6) INTO <目标>短语指明查询结果的输出目的地。INTO ARRAY 表示输出到数组,INTO CURSOR 表示输出到临时表,INTO DBF 或者 INTO TABLE 表示输出到数据表中。默认为浏览窗口。

以上短语是学习和理解 SQL SELECT 命令必须要掌握的,还有一些短语是 Visual FoxPro 特有的。

SELECT 查询命令的使用非常灵活,用它可以构造各种各样的查询。本节将通过大量实例来介绍 SELECT 命令的使用,在例子中再具体解释各个短语的含义,为方便说明,首先给出学生、选课、课程三个表的内容:

学生表的内容如表 5-3 所示。

表 5-3　学生表

学号	姓名	性别	出生日期	少数民族否	籍贯	入学成绩	简历	照片
610221	王大为	男	02/05/85	F	江苏	568.0	memo	gen
610204	彭　斌	男	12/31/83	T	北京	547.0	memo	gen
240111	李远明	女	11/12/85	F	重庆	621.0	memo	gen
240105	冯珊珊	女	02/04/87	F	重庆	470.0	memo	gen
250205	张大力	男	02/04/86	F	四川成都	250.0	memo	gen
810213	陈雪花	女	05/05/86	F	广州	368.0	memo	gen
820106	汤　莉	男	06/21/70	F	重庆	456.0	memo	gen
510204	查亚平	女	04/07/71	F	重庆	666.0	memo	gen
860307	杨武胜	男	04/05/78	T	湖南	568.0	memo	gen
520204	钱广花	女	02/07/80	T	湖北	589.0	memo	gen
231002	杨　阳	男	07/28/12	T	北京	680.0	memo	gen

选课表的内容如表 5-4 所示。

表 5-4　选课表

学号	课程号	成绩
610221	01101	85.0
610204	01102	95.0
240111	12100	95.0
240105	15105	65.0
250205	01102	85.0
820106	01103	68.0
510204	01101	88.0
860307	01101	98.0
520204	01102	78.0

课程表的内容如表 5-5 所示。

表 5-5 课程表

课程号	课 程 名	学分
01101	数据库原理	3.0
01102	软件工程	2.0
01103	VFP 程序设计	4.0
12100	计算机网络	2.0
15104	英语口语	3.0

5.4.1 基本查询

所谓简单查询是指基于一个表，可以有简单的查询条件或者没有条件，基本上由SELECT、FROM、WHERE 构成简单查询。

【例 5-15】 列出所有学生名单。

```
SELECT  *  FROM 学生
```

命令中的 * 表示输出所有字段，数据来源是学生表，所有内容以浏览方式显示。

【例 5-16】 在学生表中查询所有男生的学号，姓名和出生日期。

```
SELECT  学号,姓名,出生日期 FROM 学生 WHERE  性别 = "男"
```

【例 5-17】 列出所有学生姓名，去掉重名。

```
SELECT DISTINCT 姓名 AS  学生名单  FROM 学生
```

5.4.2 带特殊运算符的条件查询

WHERE 是条件语句关键字，是可选项，其格式是：

```
WHERE <条件表达式>
```

其中条件表达式可以是单表的条件表达式，也可以是多表之间的条件表达式，表达式用的比较符为：=（等于）、<>、!=（不等于）、==（精确等于）、>（大于）、>=（大于等于）、<（小于）、<=（小于等于）。

在 SELECT 命令中还可以使用 BETWEEN、IN、LIKE 等特殊运算符，这些运算符的使用，可以方便灵活使用 SQL 语言。表 5-6 列出了可用于条件表达式中几个特殊运算符的意义和使用方法。

表 5-6 WHERE 子句中的特殊运算符

运算符	说 明
BETWEEN	字段值在指定范围内，用法：<字段> BETWEEN <范围始值> AND <范围终值>
IN	字段值是结果集合的内容<字段> [NOT] IN <结果集合>
LIKE	对字符型数据进行字符串比较，提供两种通配符，即下画线_（代表 1 个字符）和百分号%（代表 0 或多个字符） 用法：<字段> LIKE <字符表达式>

如查询入学成绩在 600 分到 650 分之间的学生，可以使用如下的方法：

```
SELECT * FROM 学生 WHERE 入学成绩 BETWEEN 600 AND 650
```

这里的数学 BETWEEN 600 AND 650 与入学成绩>=600 AND 入学成绩<=650 是等效的。

【例 5-18】 列出学生的学号尾数为 2 的所有学生，注意学号字段的类型为字符型数据。

```
SELECT *  FROM 学生 WHERE 学号 LIKE "%2"
```

查询结果如图 5-2 所示。

图 5-2　带特殊运算符的查询

这里的 LIKE 是字符串匹配运算符，通配符"%"表示 0 个或者多个字符。通配符"_"表示 1 个字符。如：

```
SELECT * FROM 学生 WHERE 学号 LIKE "_5%"
```

表示学号第二个字符为 5 的所有学生。

【例 5-19】 列出数学成绩不在 80 到 95 之间的学生。

```
SELECT  *  FROM 学生 WHERE 数学  NOT BETWEEN  80 AND  95
```

【例 5-20】 列出所有的姓赵的学生名单。

```
SELECT 学号,姓名 FROM 学生 WHERE 姓名 LIKE "赵%"
```

以上命令的功能等同于：

```
SELECT 学号,姓名,专业 FROM 学生 WHERE 姓名 = "赵"
```

【例 5-21】 列出重庆和成都的学生信息。

```
SELECT * FROM 学生 WHERE 籍贯 IN ("重庆","成都")
```

该命令的查询条件等同于：

```
WHERE 籍贯 = "重庆" or 籍贯 = "成都"
```

【例 5-22】 列出所有非重庆籍的学生的学号、姓名和出生日期。

```
SELECT  学号,姓名,出生日期 FROM 学生 WHERE 籍贯!= "重庆"
```

在 SQL 中，"不等于"可以用"!="，"#"或"<>"表示。另外还可以用否定运算符 NOT 表示取反(非)操作，例如，上述查询条件也可以写为

```
WHERE NOT(籍贯 = "重庆")
```

5.4.3　简单的计算查询

SELECT 命令中的选项，不仅可以是字段名，还可以是表达式，也可以是一些函数。

表 5-7 列出了 SELECT 命令可操纵的常用聚合函数。

表 5-7 SELECT 命令常用聚合函数

函 数	功 能	函 数	功 能
AVG(字段名)	求字段的平均值	SUM(字段名)	求字段的和
MAX(字段名)	求字段的最大值	MIN(字段名)	求字段的最小值
COUNT(*)	求满足条件的数值		

【例 5-23】 将所有的学生入学成绩四舍五入，只显示学号、姓名和数学成绩。

```
SELECT 学号,姓名,ROUND(入学成绩,0) AS "总成绩"  FROM  学生
```

注意：这个结果不影响数据库表中的结果，只是在输出时通过函数计算输出。

【例 5-24】 求出所有学生的入学成绩平均分、最高分、最低分。

```
SELECT AVG(入学成绩) AS  "入学成绩平均分",MAX(入学成绩) AS  "入学成绩最高分",MIN(入学成绩) AS  "入学成绩最低分"  FROM 学生
```

查询结果如图 5-3 所示。

图 5-3 带函数的 SELECT 查询结果

5.4.4 分组统计查询(GROUP)与筛选(HAVING)

查询结果可以分组，其格式是：

```
GROUP BY <分组选项 1>[,<分组选项 2>…]
```

其中<分组选项>可以是字段名，SQL 函数表达式，也可以是列序号(最左边为 1)。

筛选条件格式是：

```
HAVING <筛选条件表达式>
```

HAVING 子句与 WHERE 功能一样，只不过是与 GROUP BY 子句连用，用来指定每一分组内应满足的条件。

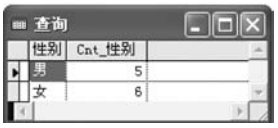

图 5-4 分组查询

【例 5-25】 分别统计男女人数。

```
SELECT 性别,COUNT(性别) FROM 学生 GROUP BY 性别
```

查询结果如图 5-4 所示。

【例 5-26】 分别统计男女中入学成绩大于 600 分的学生人数。

```
SELECT 性别,COUNT(性别) AS 人数 FROM 学生 GROUP BY 性别 WHERE 入学成绩> 600
```

如果把命令写成如下形式，统计的结果就是错误的。

```
SELECT 性别,COUNT(性别) AS 人数 FROM 学生 GROUP BY 性别 HAVING 入学成绩> 85
```

【例 5-27】 统计每门课程的平均成绩。

```
SELECT 课程号,AVG(成绩) FROM 选课 GROUP BY 课程号
```

【例 5-28】 列出平均成绩大于 80 分的课程号。

```
SELECT 课程号,AVG(成绩) 平均成绩 FROM 选课 GROUP BY 课程号 HAVING AVG(成绩)> 80
```

5.4.5　排序查询

SELECT 的查询结果是按查询过程中的自然顺序给出的，因此查询结果通常无序，如果希望查询结果有序输出，需要下面的子句配合：

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

其中排序选项可以是字段名，也可以是数字。字段名必须是主 SELECT 子句的选项，当然是 FROM <表>中的字段。数字是表的列序号，第 1 列为 1。ASC 指定的排序项按升序排列，DESC 指定的排序项按降序排列。

【例 5-29】 按性别升序列出学生的学号、姓名、性别及入学成绩，性别相同的再按入学成绩排序由高到低排序。

```
SELECT 学号,姓名,性别,入学成绩　FROM 学生 ORDER BY　性别,入学成绩　DESC
```

查询结果如图 5-5 所示。


查询

学号	姓名	性别	入学成绩
231002	杨阳	男	680
610221	王大为	男	568
860307	杨武胜	男	568
610204	彭斌	男	547
250205	张大力	男	250
510204	查亚平	女	666
240111	李远明	女	621
520204	钱广花	女	489
240105	冯姗姗	女	470
820106	汤莉	女	456
810213	陈雪花	女	368

图 5-5　多关键字排序查询

【例 5-30】 对学生表，请输出入学成绩最高的前五名学生的信息。

```
SELECT　*　TOP　5　FROM 学生 ORDER BY 入学成绩 DESC
```

输出的结果可能超过五条记录，如果入学成绩有并列的则都要输出。

5.4.6　查询结果输出

在用 SELECT 语句进行查询时，默认的输出结果都在屏幕上，需要改变输出结果可以使用 INTO 是可选项，其格式如下：

```
[INTO <目标>] | [TO FILE <文件名>[ADDITIVE] | TO PRINTER]
```

其中：

<目标>有如下 3 种形式：

ARRAY <数组名>：将查询结果存到指定数组名的内存变量数组中。

CURSOR <临时表>：将输出结果存到一个临时表(游标)，这个表的操作与其他表一

样，不同的是，一旦被关闭就被删除。

DBF <表>|TABLE <表>：将结果存到一个表，如果该表已经打开，则系统自动关闭它。如果 SET SAFETY OFF，则重新打开它不提示。如果没有指定后缀，则默认为.dbf。在 SELECT 命令执行完后，该表为打开状态。

"TO FILE <文件名>[ADDITIVE]"将结果输出到指定文本文件，ADDITIVE 表示将结果添加到文件后面。在输出的文件中，系统可以自动处理重名的问题。如不同文件同字段名用文件名来区分，表达式用 EXP-A、EXP-B 等来自动命名，SELECT 函数用函数名来辅助命名。

"TO PRINTER"将结果送打印机输出。

【例 5-31】 输出学生表中学号、性别、入学成绩，按照性别升序，入学成绩降序，将查询的结果保存到 test1.txt 文本文件中。

```
SELECT 学号,姓名,性别,入学成绩  FROM 学生 ORDER BY  性别,入学成绩  DESC
TO FILE test1
```

【例 5-32】 将例 5-29 的查询结果保存到存入 testtable 表中。

```
SELECT 学号,姓名,性别,入学成绩  FROM 学生 ORDER BY  性别,入学成绩  DESC INTO
TABLE testtable
```

5.4.7 多表查询

在一个表中进行查询，一般说来是比较简单的，连接查询是基于多个表的查询，表之间的联系是通过字段值来体现的，而这种字段通常称为连接字段。连接操作的目的就是通过加在连接字段的条件将多个表连接起来，达到从多个表中获取数据的目的。

用来连接两个表的条件称为连接条件或连接谓词，其一般格式为：

```
[<表名 1>.]<列名 1> <比较运算符> [<表名 2>.]<列名 2>
```

其中比较运算符主要有：=、>、<、>=、<=、!=。

此外连接谓词还可以使用下面形式：

```
[<表名 1>.]<列名 1> BETWEEN [<表名 2>.]<列名 2> AND [<表名 2>.]<列名 3>
```

当连接运算符为“=”时，称为等值连接，使用其他运算符称为非等值连接。

1. 等值连接

【例 5-33】 查询所有学生的成绩单，要求给出学号、姓名、课程号、课程名和成绩。

```
SELECT a.学号,姓名,b.课程号,课程名,成绩 FROM 学生 a,选课 b,课程 c;
WHERE a.学号 = b.学号 AND b.课程号 = c.课程号
```

注意：在短语 FROM 学生 A，表示选择学生表，并将学生表的表名为 A，其他表类似。在短语 SELECT A.学号 表示取学生表的学号字段。

学生情况存放在学生表中，学生选课情况存放在选课表中，课程的信息存放在课程表中，所以本查询实际上同时涉及学生、选课、课程三个表中的数据。这三个表之间的联系是

分别通过字段学号和课程号实现的。要查询学生及其选修课程的情况，就必须分别将表中学号相同的元组以及课程号相同的元组连接起来。这是一个等值连接。

【例 5-34】 查询男生的选课情况，要求列出学号、姓名、课程号、课程名和学分数。

```
SELECT a.学号,姓名 AS 学生姓名,b.课程号,课程名, c.学分;
FROM 学生 a,选课 b,课程 c,;
WHERE a.学号 = b.学号 AND b.课程号 = c.课程号 AND  a.性别 = "男"
```

2. 非等值连接查询

【例 5-35】 列出选修 01102 课的学生中，成绩大于学号为 250205 的学生该门课成绩的那些的学号及其成绩。

```
SELECT a.学号,a.成绩 FROM 选课 a,选课 b;
WHERE a.成绩> b.成绩 AND a.课程号 = b.课程号 AND b.课程号 = "01102"AND b.学号 = "250205"
```

在命令中，将成绩表看作 a 和 b 两张独立的表，表 b 中选出学号为 250205 同学的 01102 课的成绩，a 表中选出的是选修 01102 课学生的成绩，“a.成绩> b.成绩”反映的是不等值联接，查询结果如图 5-6 所示。

图 5-6　自连接查询

5.4.8　嵌套查询

有时候一个 SELECT 命令无法完成查询任务，需要一个子 SELECT 的结果作为条件语句的条件，即需要在一个 SELECT 命令的 WHERE 子句中出现另一个 SELECT 命令，这种查询称为嵌套查询。通常把仅嵌入一层子查询的 SELECT 命令称为单层嵌套查询，把嵌入子查询多于一层的查询称为多层嵌套查询。Visual FoxPro 只支持单层嵌套查询。

1. 返回单值的子查询

【例 5-36】 列出选修“数据库原理”的所有学生的学号。

```
SELECT 学号 FROM 选课 WHERE 课程号 = ;
(SELECT 课程号 FROM 课程 WHERE 课程名 = "数据库原理" )
```

上述 SQL 语句执行的是两个过程，首先在课程表中找出“数据库原理”的课程号（比如 01001），然后再在选课表中找出课程号等于 01101 的记录，列出这些记录的学号，查询结果如图 5-7 所示。

2. 返回一组值的子查询

若某个子查询返回值不止一个，则必须指明在 WHERE 子句中应怎样使用这些返回值。通常使用条件 ANY（或 SOME）、ALL 和 IN。表 5-8 列出了这些运算符的意义和使用方法。

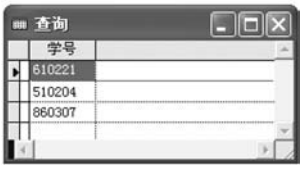

图 5-7　返回单值的子查询

表 5-8 WHERE 子句中的特殊运算符

运算符	说　明
ALL	满足子查询中所有值的记录，用法：<字段><比较符> ALL(<子查询>)
ANY	满足子查询中任意一个值的记录 用法：<字段><比较符> ANY(<子查询>)
EXISTS	测试子查询中查询结果是否为空，若为空，则返回. F. 用法：[NOT] EXISTS(<子查询>)
IN	字段值是子查询中的内容，<字段> [NOT] IN(<子查询>)
SOME	满足集合中某个值，功能与用法等同于 ANY 用法：<字段><比较符> SOME(<子查询>)

1）ANY 运算符的用法

【例 5-37】 列出选修 01101 课的学生中成绩比选修 01102 的最低成绩高的学生的学号和成绩。

```
SELECT 学号,成绩 FROM 选课 WHERE 课程号 = "01101"  AND 成绩> ANY;
(SELECT 成绩 FROM 选课 WHERE 课程号 = "01102")
```

图 5-8　返回一组值的子查询

该查询必须做两件事：先找出选修 01102 课的所有学生的期末成绩（比如说结果为 92 和 51），然后在选修 01101 课的学生中选出其成绩高于选修 01102 课的任何一个学生的成绩（即高于 72 分）的那些学生，查询结果如图 5-8 所示。

当然你也可以先找出选修 01102 的最低成绩，然后再查询。所以你也可以写成：

```
SELECT 学号,成绩 FROM 选课 WHERE 课程号 = "01101" AND 成绩>;
(SELECT MIN(成绩) FROM 选课 WHERE 课程号 = "01102")
```

2）ALL 运算符的用法

【例 5-38】 列出选修 01101 课的学生，这些学生的成绩比选修 01102 课的最高成绩还要高的学生的学号和成绩。

```
SELECT 学号,成绩 FROM 选课 WHERE 课程号 = "01101"  AND 成绩> ALL;
  (SELECT 成绩 FROM 选课 WHERE 课程号 = "01102")
```

图 5-9　含 ALL 运算符的查询

该查询的含义是：先找出选修 01102 课的所有学生的成绩，然后再在选修 01101 课的学生中选出其成绩中高于选修 01102 课的所有成绩的那些学生，查询结果如图 5-9 所示。

当然你也可以先找出选修 01102 的最高成绩，然后再查询。所以你也可以写成：

```
SELECT 学号,成绩 FROM 选课 WHERE 课程号 = "01101" AND 成绩 = ;
(SELECT MAX(成绩) FROM 选课 WHERE 课程号 = "01102")
```

3）IN 运算符的用法

【例 5-39】 列出选修"数据库原理"或"软件工程"的所有学生的学号。

```
SELECT 学号 FROM 选课 WHERE 课程号 IN;
(SELECT 课程号 FROM 课程 WHERE 课程名 = "数据库原理"OR 课程名 = "软件工程")
```

IN 是属于的意思，等价于"＝ANY"，即等于子查询中任何一个值。

5.4.9　输出合并(UNION)

输出合并是指将两个查询结果进行集合并操作，其子句格式是：

```
[UNION [ALL] < SELECT 命令>]
```

其中 ALL 表示结果全部合并。若没有 ALL，则重复的记录将被自动取掉。合并的规则是：

(1) 不能合并子查询的结果。

(2) 两个 SELECT 命令必须输出同样的列数。

(3) 两个表各相应列出的数据类型必须相同，数字和字符不能合并。

(4) 仅最后一个< SELECT 命令>中可以用 ORDER BY 子句，且排序选项必须用数字说明。

【例 5-40】 列出选修 01101 或 01102 课程的所有学生的学号。

```
SELECT 学号 FROM 选课 WHERE 课程号 = "01101" UNION SELECT 学号 FROM;
选课 WHERE 课程号 = "01102"
```

习题

一、选择题

(1) SQL 语句中条件语句的关键字是(　　)。

A. IF　　B. FOR　　C. WHILE　　D. WHERE

(2) 从数据库中删除表的命令是(　　)。

A. DROP TABLE　　B. ALTER TABLE

C. DELETE TABLE　　D. CREATE TABLE

(3) 建立表结构的 SQL 命令是(　　)。

A. CREATE CURSOR　　B. CREATE TABLE

C. CREATE INDEX　　D. CREATE VIEW

(4) 有如下 SQL 语句：DELETE FROM SS WHERE 年龄> 60，其功能是(　　)。

A. 从 SS 表中彻底删除年龄> 60 岁的记录

B. 在 SS 表中将年龄> 60 岁的记录加上删除标记

C. 删除 SS 表

D. 删除 SS 表的“年龄”字段

(5) SQL 语句中修改表结构的命令是(　　)。

A. UPDATE STRUCTURE　　B. MODIFY STRUCTURE

C. ALTER TABLE　　D. ALTER STRUCTURE

（6）SQL-SELECT 语句是（　　）。

A. 选择工作区语句　　B. 数据查询语句

C. 选择标准语句　　D. 数据修改语句

（7）在 Visual Foxpro 中，关于查询的说法正确的是（　　）。

A. “联接”选项卡与 SQL 语句的 GROUP BY 对应

B. “筛选”选项卡与 SQL 语句的 HAVING 对应

C. “排序依据”选项卡与 SQL 语句的 ORDER BY 对应

D. “分组依据”选项卡与 SQL 语句的 JOIN ON 对应

二、填空题

（1）SQL 语言集_____、_____、_____、_____功能于一体。

（2）在 VFP 6.0 支持的 SQL 语句中，_____命令可以修改表中数据，_____命令可以修改表结构。

（3）在 SQL-SELECT 语句中，允许在_____子句中给表定义别名，以便于在查询的其他部分使用。

（4）在 SQL 语句中，_____命令可以从表中删除记录，_____命令可以从数据库中删除表。

（5）在 SQL-SELECT 语句中，带_____子句可以消除查询结果中重复的记录。

（6）在 SQL-SELECT 语句中，分组用_____子句，排序用_____子句。

（7）在 ORDER BY 子句的选项中，DESC 代表_____输出，省略 DESC 时，代表_____输出。

（8）HAVING 子句不能单独使用，必须在_____短语之后使用。