

|                        |  |
|------------------------|--|
| 章节标题                   | 第四章 关系数据库标准语言 SQL  |
| 授课时数                   | 10+6   |
| 教学目标                   | 了解: SQL 的概念、特点, SQL 的定义功能及视图定义<br>理解: SQL 语言的组成, 各种 SQL 语句的语法格式<br>掌握: SQL 的各种查询语句的用法, SQL 操作语句的用法   |
| 主要知识点                  | SQL 的基本概念<br>SQL 的数据查询功能: 简单查询、嵌套查询、联接查询、分组和计算查询、集合的并运算<br>SQL 的数据操作功能: 删除 (DELETE-SQL)、插入 (INSERT-SQL)、更新 (UPDATE-SQL)<br>SQL 的数据定义功能: 表的定义、表的删除、表结构的修改、视图的定义                         |
| 教学重点                   | 简单查询、嵌套查询、联接查询、分组和计算查询、插入、更新   |
| 教学难点                   | 嵌套查询、联接查询、分组和计算查询, 表结构的修改  |
| 教学方式(教学方法, 教学技术手段的运用等) | 本章先对 SQL 的一些基本概念、特点进行阐述, 接着介绍查询语句的组成部分, 然后通过大量的实例来介绍各种不同类型的查询语句, 包括简单查询、嵌套查询、联接查询、分组和计算查询等。然后介绍 SQL 中关于数据插入、更新和删除等操作功能。最后介绍的是表的定义、删除, 表结构的修改和视图的定义这些 SQL 定义功能。使得同学们对 SQL 语言有个较全面深刻的理解。 |
| 必要说明                   |  |

## 第四章 关系数据库标准语言 SQL

SQL 是结构化查询语言 Structured Query Language 的缩写。可以说查询是 SQL 语言的重要组成部分，但不是全部，SQL 还包括数据定义、数据操纵和数据控制功能等部分。SQL 已经成为关系数据库的标准数据语言，所以现在所有的关系数据库管理系统都支持 SQL。本章将较详细地介绍 SQL 语言当中的查询部分的内容以及部分数据操纵和数据控制功能。

| 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         |

表 4.1 SQL 命令动词

### 4.1 SQL 概述

SQL 语言具有如下主要特点：

- SQL 是一种一体化的语言，它包括了数据定义、数据查询、数据操纵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。
- 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 语言就可以将要求交给系统，自动完成全部工作。
- SQL 语言非常简洁。
- 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。VF 是将 SQL 语言直接融入到自身的语言之中，使用起来更方便。

### 4.2 SQL 查询功能

```
SELECT [ALL|DISTINCT][<别名>.<选项>[, [<别名>.<选项>].....]
FROM <表名>[<别名>][,<表名>[<别名>].....]
[WHERE<条件表达式>][AND<条件表达式>.....]
[GROUP BY<分组选项>[,<分组选项>.....]]
[HAVING<组条件表达式>]
[ORDER BY<排序选项>[ASC|DESC][,<排序选项>[ASC|DESC].....]]
```

命令中各参数的含义如下：

SELECT 是该命令的主要关键字。

ALL|DISTINCT 表示 ALL 和 DISTINCT 任选其一，ALL 表示所有的记录，DISTINCT 表示去掉重复记录。

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；WHERE 说明查询条件，即选择元组的条件；

GROUP BY 短语用于对查询结果进行分组，可以利用它进行分组汇总；

HAVING 短语必须跟随 GROUP BY 使用，它用来限定分组必须满足的条件；

ORDER BY 短语用来对查询的结果进行排序

#### 4.2.1 简单查询

#### 4.2.2 简单的联接查询

#### 4.2.3 嵌套查询

SELECT 嵌套查询是一种子查询，子查询的特征是能够将一个查询的结果作为另

一个查询的一部分，子查询是对查询结果的查询。在 SQL 中，有些查询不用子查询无法表达。子查询要加括号，并且与 SELECT 语句的形式类似，也有 FROM 子句，以及可选择的 WHERE、GROUP BY 和 HAVING 子句等。子查询中的子句与 SELECT 语句中的子句格式相同，用于子查询时，它们执行正常的功能，但是子查询和 SELECT 语句还是有如下区别：

(1) 子查询通常必须生成单字段数据作为其查询结果，即必须是一个确定的项。若为一个集合，则需要使用谓词演算查询。

(2) ORDER BY 子句不能用于子查询，子查询结果只是被主查询内部使用，对用户是不可见的，所以对它们的任何排序都是没有意义的。

#### 4.2.4 几个特殊运算符

BETWEEN ... AND...：指定查询的条件是在什么范围内（数值） LIKE (% | \_): 指定查询与通配符相匹配的字符。

#### 4.2.5 排 序

使用 SQL-SELECT 可以将查询结果排序，排的短语是 ORDER BY，具体格式如下：

ORDER BY 字段 1 [ASC | DESC] [, 字段 2] [ASC | DESC...]

由以上格式可看出，可以按升序（ASC）或降序（DESC）排序，允许按一列或多列排序。

注：ORDER BY 是对最终的查询结果进行排序，不可以在子查询中使用该短语。

#### 4.2.6 简单的计算查询

SQL 命令不仅具有一般的检索能力，而且还有计算方式的检索，比如检索职工的平均工资、检索某个仓库中职工的最高工资值等。用于计算检索的函数有：

- (1) COUNT—计数
- (2) SUM—求和
- (3) AVG—计算平均值
- (4) MAX—求最大值
- (5) MIN—求最小值

这些函数可以用在 SELECT 短语中对查询结果进行计算。

#### 4.2.7 分组与计算查询

利用 GROUP BY 子句进行分组计算查询，格式如下：

GROUP BY 字段 1[, 字段 2][HAVING 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。

#### 4.2.8 利用空值查询

在第四章介绍过空值的概念，SQL 支持空值，当然也可以利用空值进行查询。IS NULL | IS NOT NULL 假设在订购单关系中，一名职工正在准备订购单，但尚未选定供应商，这样若把信息存入数据库，则供应商号和订购日期两个属性均为空值。

#### 4.2.9 别名与自联接查询

在联接操作中，经常需要使用关系名作前缀，有时这样显得很麻烦。因此，SQL 允许在 FROM 短语中为关系名定义别名，格式为：

<关系名><别名>

#### 4.2.10 内外层互相关嵌套查询

上面讨论的嵌套查询都是外层查询依赖于内层查询的结果，而内层查询与外层查询无关。事实上，有时也需要内、外层互相关的查询，这里内层查询的条件需要外层查询提供值，而外层查询的条件需要内层查询的结果。

#### 4.2.11 使用量词和谓词的查询

<表达式><比较运算符>[ANY | ALL | SOME] (子查询)

[NOT]EXISTS (子查询)

ANY、ALL 和 SOME 是量词，其中 ANY 和 SOME 是同义词，在进行比较运算时只要子查询中有一行能使结果为真，则结果就为真；而 ALL 则要求子查询中的所有行都使结果为真时，结果才为真。

EXISTS 是谓词，EXISTS 或 NOT EXISTS 是用来检查在子查询中是否有结果返回，即存在元组或不存在元组。

#### § 4.2.12 超联接查询

超联接:首先保证一个表中满足条件的元组都在结果表中；然后将满足联接条件的元组与另一个表的元组进行联接，不满足联接条件的则将应来自另一表的属性值置为空值。共分以下几种超联接

- 内部联接
- 左联接
- 右联接
- 全联接

内部联接：只有满足联接条件的记录才出现在查询结果中。(Inner join)

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。(Left join)

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。(Right join)

全联接：除满足条件的记录出现，两表中不满足联系条件的记录也出现在查询结果中。(Full join)

超联接查询格式如下：

SELECT.....

FROM Table INNER|LEFT|RIGHT|FULL Join Table

ON Join Condition

WHERE.....

注：联接条件在 ON 短语中给出，而不在 WHERE 短语中，联接类型在 FROM 短语中给出  
知识点小结：

1. 基本概念：超联接、内部联接、左联接、右联接、全联接
2. 学会使用 SQL 超联接查询语句。

#### 4.2.13 集合的并运算

SQL 支持集合的并 (UNION) 运算，即将两个 SELECT 语句的查询结果通过并运算合并成一个查询结果。为了进行并运算，要求这样的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个值域，即具有相同的数据类型和取值范围。

#### 4.2.14 特殊查询

##### 1. 显示部分结果

TOP nExpr [PERCENT]

其中 nExpr 是数字表达式，当不使用 PERCENT 时，nExpr 是 1 至 32767 间的整数，说明显示前几个记录；当使用 PERCENT 时，nExpr 是 0.01 至 99.99 间的实数，说明显示结果中前百分之几的记录。注意：TOP 短语要与 ORDER BY 短语同时使用才有效。

##### ● 查询结果去向

##### 1、将查询结果存放到数组中

INTO ARRAY ArrayName

一般存放查询结果的数组作为二维数组来使用，每行一条记录，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、将查询结果存放在临时文件中

INTO CURSOR 临时文件名

该命令产生的临时文件是一个只读的 dbf 文件，当查询结束后该临时文件是当前文件，可以像一般的 dbf 文件一样使用，但仅是只读。当关闭文件时该文件将自动删除。

注：一般利用 INTO CURSOR 短语存放一些临时结果，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，利用该短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。

##### 3、将查询结果存放到永久表中

INTO DBF|TABLE 文件名

##### 4、将查询结果存放到文本文件中

TO FILE 文本文件名 [ADDITIVE]

将查询结果存放到文本文件中，扩展名是.txt，如果使用 ADDITIVE 结果将追加在原文件的尾部，否则将覆盖原有文件。

##### 5、将查询结果直接输出到打印机

TO PRINTER [PROMPT]

可直接将查询结果输出到打印机，若使用了 PROMPT 选项，在开始打印之前会打开打印机设置对话框。

注：若在同一个查询中同时包括了 INTO 和 TO 子句，则 TO 子句不起作用。

### 4.3 SQL 操作功能

SQL 的操作功能：插入、更新、删除

#### 4.3.1 插入（添加数据）

格式一：INSERT INTO <<表名> [(<字段名 1>[,<字段名 2>[,...]])] VALUES (<表达式 1>[,<表达式 2>[,...]])

格式二：INSERT INTO 表名 FROM ARRAY 数组名|FROM MEMVAR

格式一：是向指定的表中插入记录，当插入的不是完整的记录时，可以指定字段。VALUES 用来给出具体的记录值

格式二：FROM ARRAY 说明从指定的数组中插入记录值，FROM MEMVAR 说明根据同名的内存变量来插入记录值，如果同名的变量不存在，那么相应的字段为默认值或空。

注：当一个表定义了主索引或候选索引后，由于相应的字段具有关键字的特性，即不能为空，

所以只能用此命令插入记录。VF 以前的插入命令 (INSER 或 APPEND) 是先插入一条空记录, 然后再输入各字段的值, 由于关键字字段不允许为空, 所以使用以前的方法就不能成功地插入记录。

### 4.3.2 更新

UPDATE <表名>

SET <列名>=表达式……

[WHERE<条件>];

一般使用 WHERE 子句指定条件, 以更新满足条件的一些记录的字段值, 并且一次可以更新多个字段, 如果不使用 WHERE 子句, 则更新全部记录。

### 4.3.3 删除

DELETE FROM <表名>[WHERE <条件>]

FROM 指定从哪个表中删除数据, WHERE 指定被删除的记录所满足的条件, 若不使用 WHERE 子句, 则删除该表中的所有记录。

注: 在 VF 中 SQL DELETE 命令同样是逻辑删除记录, 如果要物理删除记录需要继续使用 PACK 命令。

## 4.4 SQL 定义功能

### 4.4.1 表的定义

定义表名 CREATE TABLE | DBF 表名 [NAME 长表名][FREE]

定义字段 (字段 1 字段类型[(字段宽度),[字段精度]] 是否允许空值 [NULL | NOT NULL] (默认允许空值) 实体完整性 [PRIMARY KEY | UNIQUE]

[REFERENCES 表名 2 [TAG 索引文件名]]

域完整性 [CHECK [表达式]]

出错提示信息 [ERROR [提示信息]]

定义默认值 [DEFAULT [表达式]]

[字段 2 字段类型[(字段宽度), [字段精度]]

同上…

| FROM ARRAY 数组名

从以上句法格式可以看出, 用 CREATE TABLE 命令建立表可以完成用第四章介绍的表设计器完成的所有功能。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 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。

### 4.4.2 表的删除

DROP TABLE <表名>

作用: 可直接从磁盘上删除 .dbf 表文件。若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。

### 4.4.3 表结构的修改

格式一:

```
ALTER TABLE 表名 ADD | ALTER[COLUMN]字段 1 字段类型[(字段宽度), [字段精度]]
    [NULL | NOT NULL] (默认允许空值)
    [CHECK 字段 1[ERROR [提示信息]]
    [DEFAULT [表达式]]
    [PRIMARY KEY | UNIQUE]
    [REFERENCES 表名 2 [TAG 索引文件名 1]]
```

该命令格式可以添加(ADD)新的字段或修改(ALTER)已有的字段，它的句法基本可以与CREATE TABLE的语句相对应。

从命令格式可以看出，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，定义主关键字和联系等；但是不能修改字段名，不能删除字段，也不能删除已经定义的规则等。

格式二：

```
ALTER TABLE 表名 ALTER[COLUMN]字段 1 [NULL | NOT NULL]
[SET DEFAULT [表达式 1]] [SET CHECK 表达式 2[ERROR [提示信息]]
[DROP DEFAULT][DROP CHECK]
```

从命令格式可以看出，该格式主要用于定义、修改和删除有效性规则和默认值定义。

以上两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。第三种格式正是在这些方面对前两种格式的补充。

格式三：

```
ALTER TABLE 表名 1 [DROP [COLUMN]字段 1
[SET DEFAULT [表达式 1]] [ERROR [提示信息]]
[DROP CHECK]
[ADD PRIMARY KEY 表达式 TAG 索引名][FOR 条件]
[DROP PRIMARY KEY]
[ADD UNIQUE 表达式 [TAG 索引名] [FOR 条件]]
[DROP UNIQUE TAG 索引名]
[ADD FOREIGN KEY[表达式] [TAG 索引名] [FOR 条件]]
REFERENCES 表名 2 [TAG 索引名]
DROP FOREIGN KEY TAG 索引名
[RENAME COLUMN 字段名 TO 新字段名]
```

该格式可以删除字段(DROP [COLUMN])、可以修改字段名(RENAME COLUMN)、可以定义、修改和删除表一级的有效性规则等。

#### 4.4.4 视图的定义

在VF中视图是一个定制的虚拟表，可以是本地的、远程的或带参数的。视图可引用一个或多个表，或者引用其他视图。视图是可更新的，它可引用远程表。在关系数据库中，视图也称作窗口，即视图是操作表的窗口，可以把它看作是从表中派生出来的虚表。它依赖于表，但不独立存在，视图是根据对表的查询定义的，其命令格式如下：

```
CREATE VIEW 视图名[(字段名 1[,字段名 2]...)
AS select_statement
```

其中select\_statement可以是任意的SELECT查询语句，它说明和限定了视图中的数据；当没有为视图指定字段名时，视图的字段名将与select\_statement中指定的字段名或表中的字段名同名。视图是根据表定义或派生出来的，所以在涉及到视图的时候，常把表称作基本表。

- 1、从单个表派生出的视图
- 2、从多个表派生出的视图
- 3、视图中的虚字段。

用一个查询来建立一个视图的 **SELECT** 子句可以包含算术表达式或函数，这些表达式或函数与视图的其他字段一样对待，由于它们是计算得来的，并不存储在表内，所以称为虚字段。

#### 4、视图的删除

视图由于是从表中派生出来的，所以不存在修改结构的问题，但是视图可以删除。在 VF 中可以修改视图，我们将在第五章中介绍。删除视图的命令格式是：

```
DROP VIEW <视图名>
```

比如要删除视图 `v_sal`，只要键入命令：

```
DROP VIEW v_sal
```

#### 5、关于视图的说明

在 VF 中视图是可更新的，但是这种更新是否反映在基本表中则取决于视图更新属性的设置。在 VF 中视图有它特殊的概念和用途，我们将在第五章具体讲解。

在关系数据库中，视图始终不真正含有数据，它总是原来表的一个窗口。所以，虽然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。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。这种限制是很有必要的，它可以避免一些潜在问题的发生。

本章小结：本章较全面地介绍了关系数据库标准语言 SQL,尤其 4.2 节中用大量的实例介绍了 SQL SELECT 语句的使用方法，这些实例都可以在 VF 下执行。SQL 已经成为 VF 的基本内容，不掌握 SQL 要用好 VF 基本是不可能的。下一章将介绍的视图和查询，如果没有 SQL 的基础也是很难理解的。所以说 SQL 是学好、用好 VF 的基础。

作业：P154 习题四