

## 第5章

# 查询的创建与使用

数据查找是数据库管理系统中的一个基本功能。使用查询可以将不同表中的信息结合起来,提供一个相关数据项的统一视图。使用查询还可以选择记录、更新表中数据、向表中添加新记录。

## 5.1 查询简介



### 5.1.1 查询和表的区别

在使用数据库中的数据时,并不是简单地使用某一个表中的数据,而常常是将有“关系”的多张表中的数据关联起来使用,有时还可能要对这些数据进行一定的计算。对于这样的要求,建立“查询”对象可以很轻松地解决,查询就是依据一定的查询条件,对数据库中的数据信息进行查找。查询的字段来自互相之间有“关系”的表,这些字段组合成一个新的数据表视图,但它并不存储任何的数据。当改变表中的数据时,查询得到的数据也随之发生改变。

在运行查询时,查询的结果动态地来源于表对象,是表中数据的一个镜像。所以在查询数据表中无法加入或删除字段,也不能修改查询字段的字段名。

### 5.1.2 常见的查询

使用查询,可以按照不同的方式查看、更改和分析数据。同时,查询也可以作为窗体、报表的数据源。Access 中常见的查询有:选择查询、参数查询、聚合查询、交叉表查询、操作查询、SQL 查询。

#### 1. 选择查询

选择查询是最常见的一种查询,它从一个或多个有“关系”的表中将满足要求的数据提取出来,并把这些数据显示在新的查询数据表中,并能对记录进行分组、总计、计数、求平均值,以及其他类型的计算。选择查询的查询条件在创建查询时给定、除非修改,否则不变。

#### 2. 参数查询

如果用户查询时需要通过在对话框中输入要查询的数据,就要创建参数查询。参数查询可以在运行查询的过程中修改查询的规则,并且执行参数查询时会显示一个输入对话框以提示用户输入信息。

Access 的参数查询是建立在选择查询或交叉表查询的基础之上的,在运行选择查询或交叉表查询之前,为用户提供了一个设置条件的参数对话框,可以很方便地更改查询限制或

对象。当然不仅可以建立单个参数的查询,还可以建立多字段参数查询。例如,可以设计用它提示输入两个日期,然后通过 Access 检索在这两个日期之间的数据。

### 3. 聚合查询

聚合查询有时也称为分组查询,创建聚合查询有助于快速分组和汇总数据。因为使用传统的选择查询只能检索显示在数据源中的记录,而使用聚合查询则可以对检索数据进行诸如总计、平均值、计数等汇总操作。

### 4. 交叉表查询

Access 支持一种特殊类型的总计查询,称为交叉表查询,交叉表查询允许用户精确确定汇总数据如何在屏幕上显示。利用交叉表查询,可以在类似电子表格的格式中查看计算值,也能够计算数据的总计、平均值、计数或其他类型的操作。交叉表查询以传统的行列电子数据表形式显示汇总数据并且与 Excel 数据透视表密切相关。

### 5. 操作查询

使用操作查询只需进行一次操作就可对许多记录进行更改和移动。操作查询有四种:删除查询、更新查询、追加查询、生成表查询。

#### 1) 删除查询

从一个或多个表中删除一组记录。例如,可以使用删除查询来删除已经离校的学生信息。使用删除查询,会删除整个记录。

#### 2) 更新查询

对一个或多个表中的一组记录做全局的更改。使用更新查询,可以更改已有表中的数据。例如,给所有职工的工资增加 200 元。

#### 3) 追加查询

将一个或多个表中的一组记录添加到另一个表的末尾。例如,新入校学生的信息存放在新生表中,可以通过追加查询将其追加到总表中。

#### 4) 生成表查询

生成表查询可以根据一定的准则来新建表,然后再将所生成的表导出到其他数据库中,或者在窗体和报表中加以利用。

### 6. SQL 查询

SQL 查询是用户使用 SQL 语句直接创建的一种查询。实际上,Access 所有的查询都可以认为是一个 SQL 查询,因为 Access 就是以 SQL 语句为基础来实现查询功能的。

#### 1) SQL 查询的分类

SQL 查询可以分为四类:联合查询、传递查询、数据定义查询和子查询。

联合查询:该查询使用 UNION 运算符来合并两个或更多选择查询的结果。

传递查询:SQL 特定查询,可以用于直接向 ODBC 数据库服务器发送命令。通过传递查询,可以直接使用服务器上的表,而不是由 Access 数据库引擎处理后的数据。

数据定义查询:包含数据定义语言(DDL)语句的 SQL 特有查询,这些语句可用来创建或更改数据库中的对象。

子查询:嵌在另一个选择查询或动作查询内的 SQL 语句。

#### 2) SQL 语句的使用场合

SQL 语句可以在 Access 中的很多场合使用,只要这些场合能够输入表、查询或字段的

名称即可。某些情况下,Access 会自动填入 SQL 语句。例如,当使用向导创建窗体或报表以便从多个表中获得数据时,Access 会自动创建一个 SQL 语句,并将该语句用作窗体或报表的“记录源”属性。在通过向导创建列表框或组合框时,Access 会创建一个 SQL 语句,并将该语句用作列表框或组合框的“行来源”属性设置。

如果不使用向导,也可以为“记录源”或“行来源”属性生成一个 SQL 语句,方法是单击这些属性旁的任意一个“生成”按钮,然后在查询“设计”视图中创建查询。

## 5.2 选择查询的创建

### 5.2.1 利用查询设计视图建立查询

直接使用查询设计视图建立查询有利于更好地理解数据库中表之间的关系,这对建立一个优秀的数据库非常有用。

建立一个“学生成绩查询”,通过这个查询显示学生的学习成绩,包括“学号”“姓名”“课程名称”“任课教师”“成绩”等字段,步骤如下。

(1) 首先打开 xsjl 数据库,然后单击“创建”菜单中的“查询/查询设计”项,弹出“显示表”对话框,如图 5.1 所示。

(2) 在“显示表”对话框中,“表”选项卡中列出了所有的表,“查询”选项卡中列出了所有的查询,而“两者都有”可以把数据库中所有“表”和“查询”对象都显示出来,这样有助于从选择的表或查询中选取新建查询的字段。

单击“显示表”对话框上的“两者都有”标签,在列表栏中选择需要的表或查询。然后单击“添加”按钮,这样就可以将表添加到查询窗口中。



图 5.1 “显示表”对话框

(3) 关闭“显示表”对话框,回到“查询设计”窗口,如图 5.2 所示。

若从查询设计器中删除表非常简单,只需要在查询设计器中右击相应的表,然后从快捷菜单中选择“删除表”命令即可。

“查询设计”窗口中有三个主要的视图,通过左上角的视图切换按钮可以在不同的视图之间切换。

设计视图:在该视图可以创建查询。

数据表视图:显示查询结果。

SQL 视图:显示查询对应的 SQL 语句。

查询设计器由表查询窗格和按示例查询(QBE)设计网格两部分构成。这两个窗格通过一个窗格大小调整滚动条水平分隔。可使用该滚动条将设计网格向左或向右移动,使用鼠标单击并向上或向下拖动该滚动条,可以更改上部窗格和下部窗格的相对大小。



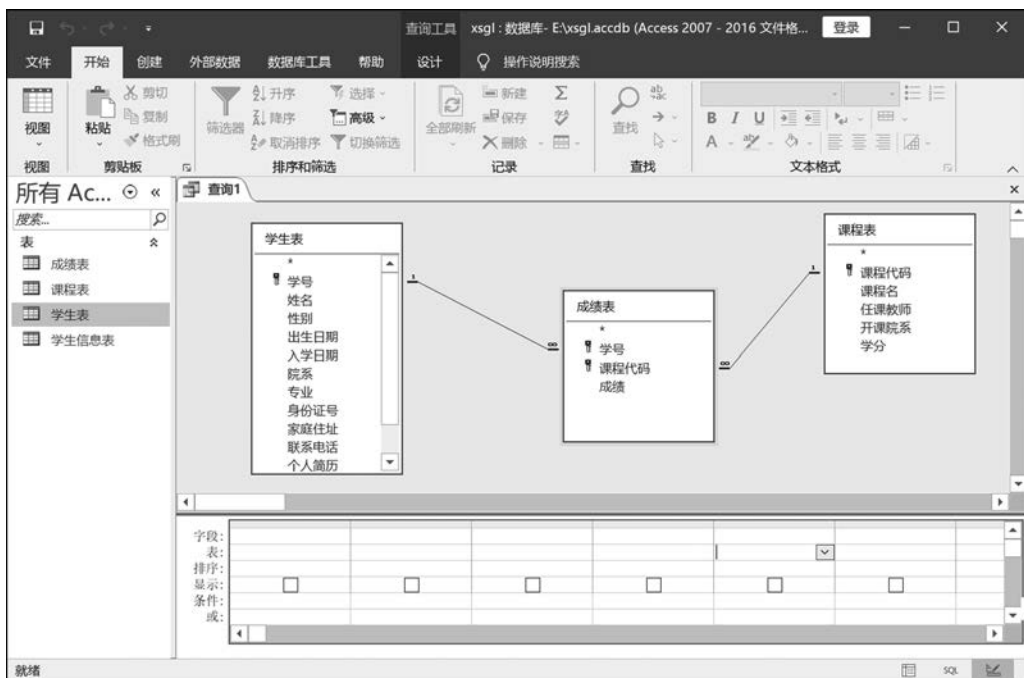


图 5.2 “查询设计”窗口

查询设计器窗口的上部是表查询窗格。

向查询的设计中添加表或查询及其对应的字段列表时,将添加到该位置。每个要添加的对象都有一个对应的字段列表。每个字段列表包含对应表或查询中所有字段的名称。

查询设计器窗口的下部是按示例查询(QBE)设计网格。

QBE 窗格保存查询中涉及的字段名以及用于选择记录的条件。QBE 窗格中的每一列包含上部窗格中表或查询包含的单个字段的信息。将字段从上部窗格拖动到 QBE 窗格可设计查询的构成字段。

QBE 窗格包含以下 6 个行标题。

- ① 字段: 查询中所使用的字段名。
- ② 表: 该字段来自于数据表。
- ③ 排序: 是否按该字段排序。
- ④ 显示: 该字段是否在结果集工作表中显示。
- ⑤ 条件: 查询条件。
- ⑥ 或: 用来提供多个查询条件。

(4) 在查询中添加或删除目标字段。

在查询设计表格中添加的字段称为“目标字段”,添加目标字段有两种方法。

第一种方法: 在“示例查询设计”窗口的表格中选择一个空白的列,单击第一行对应的格子,格子的右边出现一个带下箭头的按钮,单击这个按钮出现下拉框,在下拉框中就可以选择相应的目标字段。

第二种方法: 选中目标字段所在的表,在它的列表框中找到需要添加的字段,将鼠标移

动到列表框中标有这个字段的选项上,按住鼠标左键,这时鼠标光标变成一个长方形,拖动鼠标将长方形拖到下方查询表格中的一个空白列,放开鼠标左键,这样就可以将目标字段添加到查询表格中。

如果要删除一个目标字段,将鼠标移动到要删除的目标字段所在列的选择条上,光标会变成一个向下的箭头,单击鼠标左键将这一列都选中,按 Delete 键,选中的目标字段将被删除。

现在加入“学号”“姓名”“课程名”“成绩”字段,结果如图 5.3 所示。

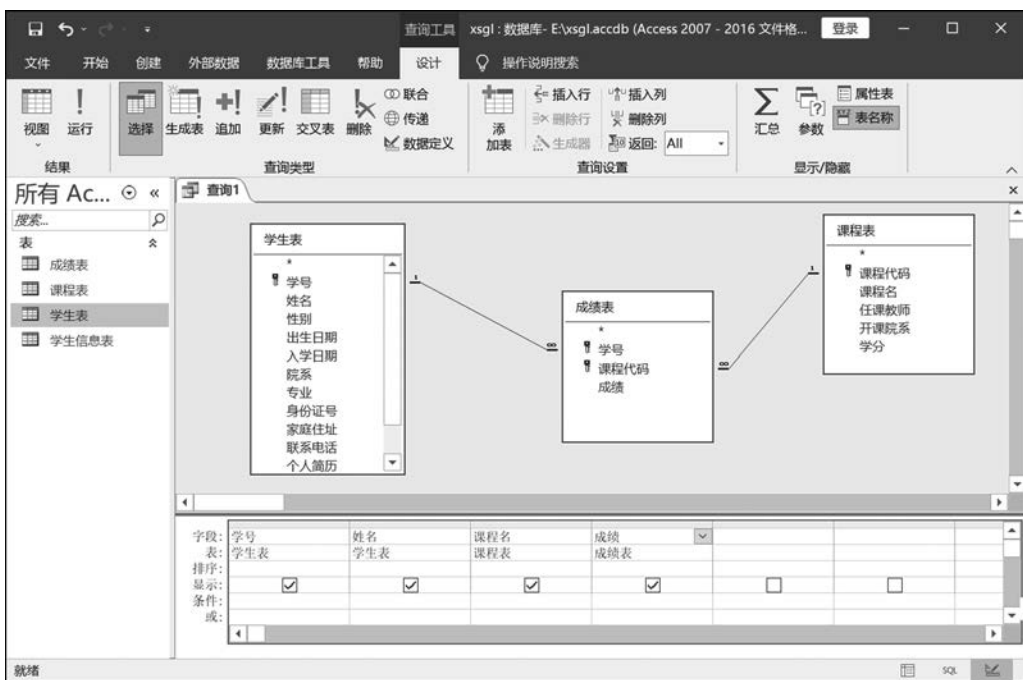


图 5.3 设置结果

(5) 切换到查询的数据表视图查看查询结果。

“查询”可以在“设计”视图和“数据表”视图中切换。在 Access 中,视图之间切换非常简单,在视图切换按钮中可以选择“设计”视图和“数据表”视图。选择“数据表视图”,结果如图 5.4 所示。

(6) 保存查询。

查询已经基本建立成功,现在需要进行查询的保存。单击“文件”菜单,选择“保存”命令,然后输入查询的名称“学生成绩查询”,单击“确定”按钮即可保存查询。

## 5.2.2 管理查询字段

在设计查询时,有时需要对字段进行特定操作。例如,重新排序、插入新字段或者删除现有字段,或者将某个字段添加到 QBE 窗格但不显示它,等等。

### 1. 更改字段顺序

字段在 QBE 窗格中从左到右的显示顺序决定了它们在数据表视图中的显示顺序。移动 QBE 窗格中的字段,可以在查询结果中实现新的字段序列,在选中字段的情况下,只需要

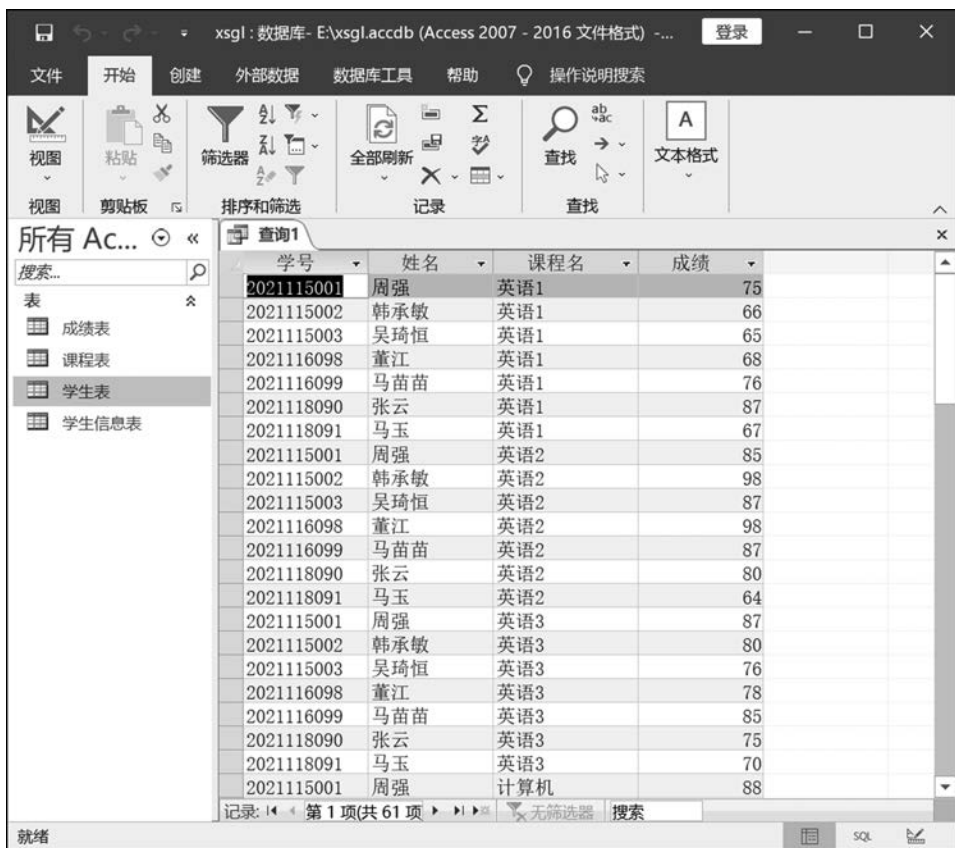


图 5.4 查询运行结果

将字段拖动到新位置,即可在 QBE 设计中移动字段。

更改字段显示顺序的过程如下。

(1) 单击某个字段的选择器栏,所选字段变为黑色,如图 5.5 所示。

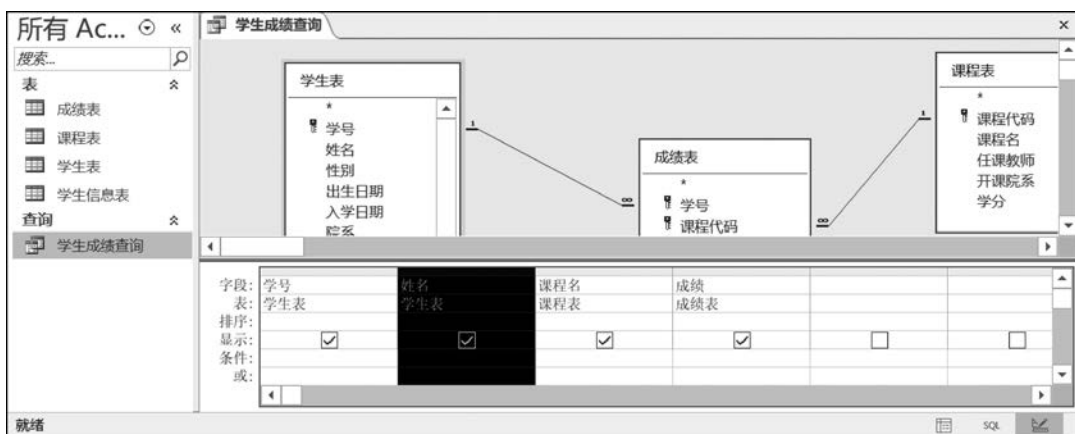


图 5.5 更改字段顺序

(2) 然后在按住鼠标左键的同时,将该字段拖动到 QBE 窗格中的新位置。

注意: 查询中的字段顺序与数据在窗体或报表中的显示方式无关。

## 2. 删除字段

要从 QBE 窗格中删除某个字段,可以选择字段并按 Delete 键。也可以右击某个字段的选择器栏,然后从显示的快捷菜单中选择“剪切”命令即可。

## 3. 插入字段

要在 QBE 窗格中插入新字段,可以将表查询窗格中某个字段列表窗口中的字段拖动到 QBE 窗格的某一列上,新列将插入到该列的左侧。

## 4. 隐藏字段

执行查询时,有时可能希望仅显示 QBE 窗格中的某些字段,这时可以将不需要显示的字段隐藏。一般情况下,在查询中隐藏某个字段的常见原因是该字段用于作为排序或选择条件,但在查询结果中并不需要显示其内容。

例如,显示物理学院所有学生的成绩。这时,物理学院是选择条件,不需要显示。只需要在 QBE 窗格中“院系”栏取消选中该字段的“显示”复选框,如图 5.6 所示。

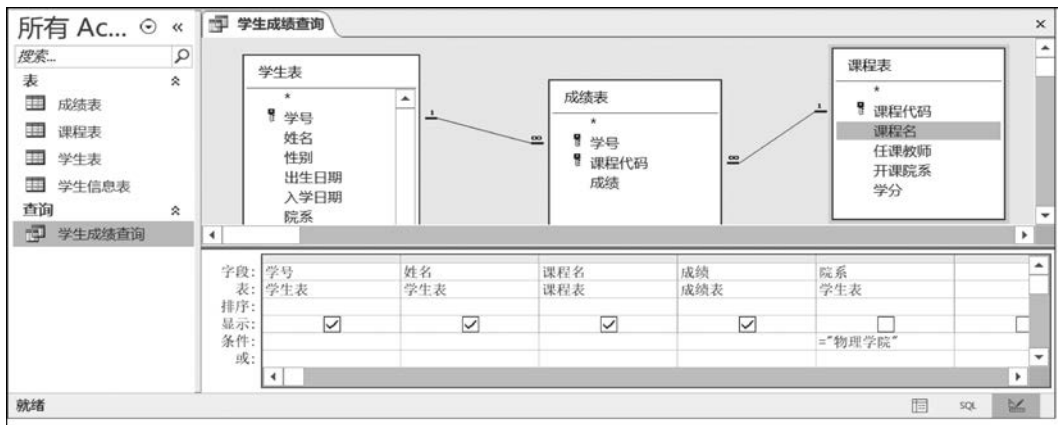


图 5.6 隐藏字段

## 5.2.3 运算符简介

运算符和操作数的组合称为表达式。对于表达式 Access 会对其进行求值。表达式可用于在表的字段属性中指定数据验证规则、在窗体和报表中创建计算字段,以及在查询中指定条件。

运算符根据运算对象不同可分为算术运算符、比较运算符、字符串运算符、逻辑运算符和其他运算符。

### 1. 算术运算符

算术运算符用于执行数值计算。当使用算术运算符时,参与运算的可以是任意数值数据类型,可以是常量、变量或者字段。运算时可以单独使用这些数值,也可以组合使用以创建复杂的表达式。算术运算符有 7 种,如表 5.1 所示。

表 5.1 算术运算符

| 运算符 | 名称 | 功 能                  | 举 例    | 结 果 |
|-----|----|----------------------|--------|-----|
| +   | 加  | 两个数字型字段值相加,两个文本字符串连接 | 100+20 | 120 |
| -   | 减  | 两个数字型字段值相减           | 100-20 | 80  |

续表

| 运算符 | 名称 | 功 能              | 举 例        | 结 果              |
|-----|----|------------------|------------|------------------|
| *   | 乘  | 两个数字型字段值相乘       | 100 * 20   | 2000             |
| /   | 除  | 两个数字型字段值相除       | 100/6      | 16.667           |
| \   | 整除 | 两个数字型字段值相除四舍五入取整 | 100\6      | 16               |
| ^   | 幂  | A 的 B 次幂         | 2^2.1      | 4.28709385014517 |
| Mod | 模  | 取余, A 除以 B 的余数   | 22.52mod 4 | 3                |

## 2. 比较运算符

比较运算符用于比较两个值,基于比较运算符生成的表达式的运算结果为 True(用-1表示)、False(用0表示)或 Null。关系成立结果为 True,关系不成立结果为 False,当无法计算表达式时就返回 Null。一般情况下 Access 在比较大小上不区分大小写。比较运算符有 6 种,如表 5.2 所示。

表 5.2 比较运算符

| 运算符 | 名称    | 功 能                                 | 举 例           | 结 果                       |
|-----|-------|-------------------------------------|---------------|---------------------------|
| =   | 等于    | 两个表达式的值相等返回 True,若一侧为 Null,则返回 Null | 100=20        | False                     |
| <>  | 不等于   | 两个表达式的值不相等返回 True                   | 100<>20       | True                      |
| <   | 小于    | 左侧小于右侧返回 True                       | “man”>“women” | False                     |
| <=  | 小于或等于 | 左侧小于或等于右侧返回 True                    | 10<=10        | True                      |
| >   | 大于    | 左侧大于右侧返回 True                       | [成绩]>60       | 若成绩字段的值大于 60 则结果为 True    |
| >=  | 大于或等于 | 左侧大于或等于右侧返回 True                    | [成绩]>=60      | 若成绩字段的值大于或等于 60 则结果为 True |

## 3. 字符串运算符

与算术运算符和逻辑运算符不同,字符串运算符是专门用于处理字符串数据类型的。字符串运算符有 3 种,如表 5.3 所示。

表 5.3 字符串运算符

| 运算符      | 名称     | 功 能                    | 举 例                | 结 果                  |
|----------|--------|------------------------|--------------------|----------------------|
| &        | 连接     | 将两个字符串连接为一个字符串         | “姓名:”&“张明”         | 姓名:张明                |
| Like     | 操作数类似  | 确定一个字符串是否与另一个字符串的模式相匹配 | [xm] Like “周强”     | 若 xm 字段的值是周强则为 True  |
| Not Like | 操作数不类似 | 确定一个字符串是否与另一个字符串的模式不匹配 | [xm] Not Like “周强” | 若 xm 字段的值不是周强则为 True |

**注意:** Like 运算符及 Not Like 运算符用于比较两个字符串表达式。这些运算符确定一个字符串是否与另一个字符串的模式相匹配,返回值为 True、False 或 Null(如果 Like 运算中有操作数为 Null,则结果为 Null)。Like 和 Not Like 运算符运算时不区分大小写。

Like 运算符使用语法如下：

表达式 Like 模式字符串

其功能是在模式字符串中查找表达式，如果存在返回 True。

Like 和 Not Like 运算符为字符串比较提供了强大且灵活的工具，而通配符则可以增强 Like 运算符的灵活性。表 5.4 显示了 5 个可与 Like 运算符结合使用的通配符。

表 5.4 可与 Like 运算符结合使用的通配符

| 通 配 符    | 功 能                 |
|----------|---------------------|
| ?        | 单个字符(0~9,A(a)~Z(z)) |
| *        | 任意字符串               |
| #        | 任意单个数字              |
| [list]   | 列表中的任意单个字符          |
| [! list] | 不在列表中的任意单个字符        |

[list]和[! list]都可在两个字符之间使用连字符来表示范围。表 5.5 列举了一些通配符示例。

表 5.5 通配符使用示例

| 示 例                        | 含 义                                             |
|----------------------------|-------------------------------------------------|
| [Answer] Like “[A-D]”      | 若 Answer 的值为 A、B、C、D、a、b、c、d，返回 True，否则返回 False |
| [xm] Like “???”            | xm 长度刚好是 3 的返回 True，否则返回 False                  |
| [code] Like “610103 * ”    | code 以 610103 开头的返回 True，否则返回 False             |
| [xm] Not Like “[周,李,张] * ” | xm 不是周姓、李姓、张姓返回 True，否则返回 False                 |

#### 4. 逻辑运算符

逻辑运算符用于在表达式中连接多个条件。与比较运算符类似，这些运算符也是始终返回 True、False 或 Null。常见的逻辑运算符有 5 种，如表 5.6 所示。

表 5.6 逻辑运算符

| 运算符 | 名称   | 示 例             | 运 算 规 则                                |
|-----|------|-----------------|----------------------------------------|
| And | 与    | 表达式 1 And 表达式 2 | 当表达式 1 和表达式 2 都为 True 时返回 True         |
| Or  | 或    | 表达式 1 Or 表达式 2  | 当表达式 1 或表达式 2 中的任何一个为 True 时返回 True    |
| Not | 非    | Not 表达式         | 表达式为 True 时返回 False，否则返回 True          |
| Xor | 异或   | 表达式 1 Xor 表达式 2 | 当表达式 1 和表达式 2 的值相同时，返回 False，否则返回 True |
| Eqv | 逻辑相等 | 表达式 1 Eqv 表达式 2 | 当表达式 1 和表达式 2 的值相同时，返回 True，否则返回 False |

#### 5. 其他运算符

除了以上运算符外，Access 还有 3 个常用的运算符 Between...And、In、Is，其运算规则如表 5.7 所示。

表 5.7 3 个常用运算符的运算规则

| 运 算 符         | 功 能    | 举 例                |
|---------------|--------|--------------------|
| Between...And | 指定范围操作 | Between“A”And“B”   |
| In            | 指定枚举范围 | In (“A, B, C”)     |
| Is            | 判定是    | Expression Is Null |

**注意：**通常情况下 Is 运算符与关键字 Null 结合使用以确定对象的值是否为 Null。在 VBA 环境中 Is 运算符可用于比较各种对象,以确定它们是否代表相同的实体。也就是说,Is 运算符仅适用于对象和对象变量,例如表中的字段。Is 运算符不能用于简单变量(例如字符串或数字)。

#### 6. 运算符优先顺序

在使用具有多个运算符的复杂表达式时,Access 必须确定运算符计算的先后次序。优先级高的先计算,优先级低的后计算。Access 将始终遵循运算符的优先级,除非使用括号改写其默认优先顺序。括号内的运算会先于括号外的运算执行。在括号中,Access 会遵循默认的运算符优先顺序进行计算。

不同类别运算符的计算优先顺序如下:

算术运算符→比较运算符→逻辑运算符

每个类别都包含自己的优先顺序。

(1) 算术运算符的优先级:

指数→取舍→乘法与除法(从左到右)→整除→求模→加法和减法(从左到右)→字符串的连接。

(2) 比较运算符的优先级:

等于→不等于→小于→大于→小于或等于→大于或等于→Like 运算符

(3) 逻辑运算符的优先级:

Not→And→Or→Xor→Eqv

### 5.2.4 为查询添加准则

运算符和表达式的一种基本应用是生成复杂的查询条件。在使用选择查询时,可能需要指定一个或多个条件以限制显示信息的范围。

#### 1. 查询中常用运算符

在查询中所用到的运算符有算术运算符、比较运算符、逻辑运算符和字符串运算符等。表 5.8 显示了用于选择查询的最常见运算符。

表 5.8 查询中常用运算符

| 算术运算符 | 关系运算符 | 逻辑运算符 | 字符串运算符   | 其他运算符         |
|-------|-------|-------|----------|---------------|
| *     | =     | And   | &        | Between...And |
| /     | <>    | Or    | Like     | In            |
| +     | >     | Not   | Not Like | Is Null       |
| -     | <     |       |          | Is Not Null   |

在设置查询条件时,经常会用到 Access 的内置函数,而日期函数是最常用的一类。表 5.9 列出了常用的系统内置日期函数。

表 5.9 常见的日期函数

| 函 数     | 功 能      | 函 数       | 功 能          |
|---------|----------|-----------|--------------|
| Date()  | 返回系统当前日期 | Weekday() | 返回日期中的星期数    |
| Year()  | 返回日期中的年份 | Hour()    | 返回时间中的小时数    |
| Month() | 返回日期中的月份 | Now()     | 返回系统当前的日期与时间 |
| Day()   | 返回日期中的日数 |           |              |

在准则表达式中使用日期/时间时,必须要在日期值两边加上“#”。例如: #Feb 12, 98 #, #2/12/98 #。

## 2. 在查询添加准则时需要考虑的问题

给查询添加选择准则时,有两个问题需要考虑:首先是为哪个字段添加准则,其次就是要为这个字段添加什么样的准则。如果只想看物理学院学生的考试成绩,很明显就是为“院系”字段添加准则,而添加的准则就是“院系”字段的值只能等于“物理学院”。限定了这两个条件,就很容易实现任何一种选择准则。

## 3. 添加准则

下例中,创建查询,选择信息学院和物理学院所有不及格的学生的学号、姓名、性别、院系、课程名、成绩、学分。

具体过程如下。

(1) 通过设计视图创建查询,字段包括学号、姓名、性别、院系、课程名、成绩、学分。结果如图 5.7 所示。

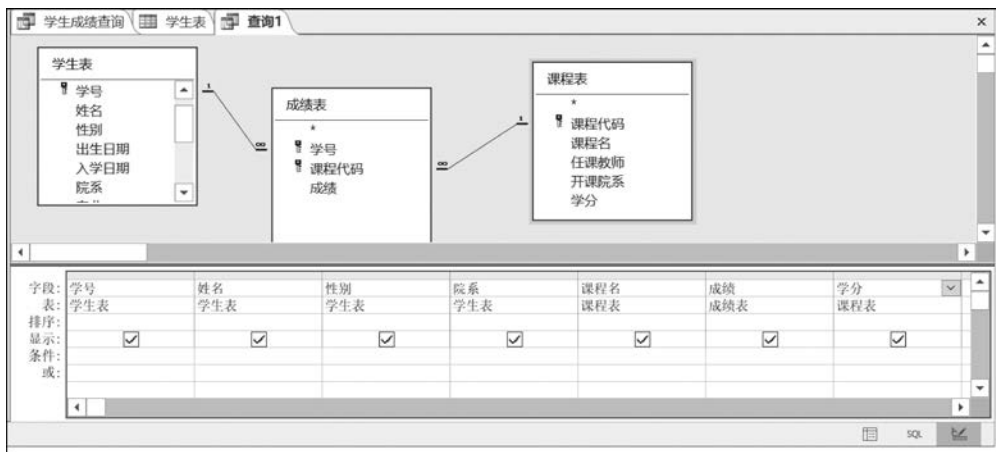


图 5.7 创建选择查询

(2) 在“院系”字段的“条件”属性中写上“=物理学院”,其中的“=”和“”均可以省略,系统会自动在物理学院的两边加上“”;在“院系”字段的“或”属性中写上“信息学院”;在“成绩”字段的“条件”属性中写上“<60”;在“成绩”字段的“或”属性中写上“<60”,取消选中“成绩”字段的“显示”复选框,如图 5.8 所示。

(3) 保存查询,为其命名为“信息-物理学院不及格学生查询,运行查询”,结果如图 5.9 所示。

注意:

(1) 可以在选择查询中使用函数。

修改成绩字段为 Int(Sqr([成绩]) \* 10),如图 5.10 所示。

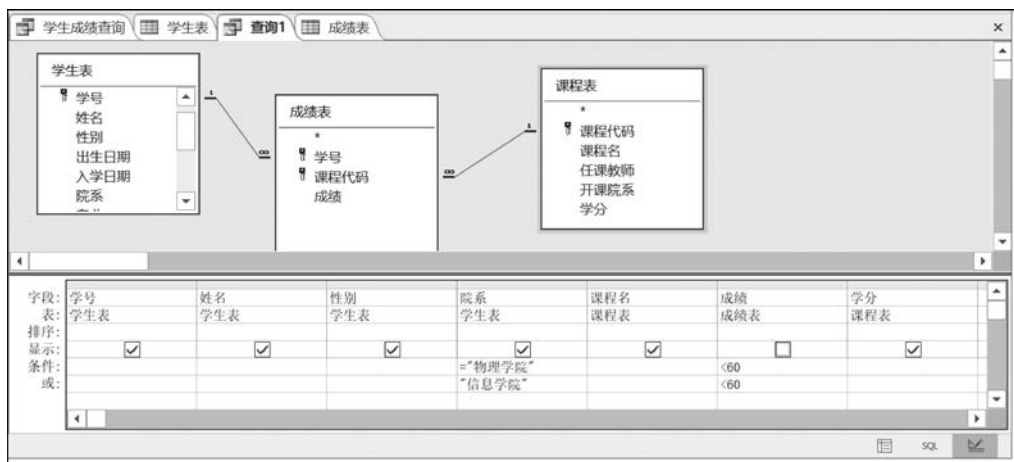


图 5.8 为字段添加条件

| 学号         | 姓名 | 性别 | 院系   | 课程名     | 学分 |
|------------|----|----|------|---------|----|
| 2021118091 | 马玉 | 女  | 物理学院 | 体育      | 1  |
| 2021118091 | 马玉 | 女  | 物理学院 | 程序设计基础1 | 3  |
| 2021115001 | 周强 | 男  | 信息学院 | 体育      | 1  |

图 5.9 查询运行结果

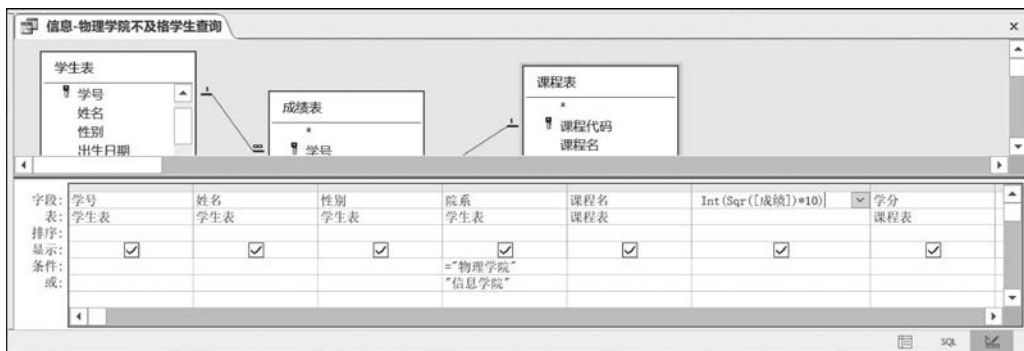


图 5.10 在查询中使用函数

表示对成绩字段的算术平方根乘以 10, 然后取整。在查询中使用表名和字段名时, 最好用 [ ] 括起来。

(2) 可以使用 Like 运算符和通配符。

例如, 查询所有姓张的学生, 可以在“姓名”字段的“条件”属性中书写: Like“张 \*”

(3) 可以使用 Between...And 运算符。

例如, 要查找 2001 年 1 月 1 日到 2001 年 12 月 31 日出生的学生, 先添加出生日期字段, 然后在“出生日期”字段的“条件”属性中书写: Between #2001/1/1# and #2001/12/31#。

## 5.3 参数查询

参数查询可以在运行查询的过程中由用户输入查询规则。用户在执行参数查询时,会显示一个输入对话框,提示用户输入查询条件,这种查询称为参数查询。

Access 的参数查询是建立在选择查询或交叉表查询的基础上的,是在运行选择查询或交叉查询之前,为用户提供了一个设置准则的参数对话框,可以很方便地设置查询条件。

当需要对某个字段创建参数查询时,设置过程如下。

(1) 通过查询向导创建一个能够查找学生成绩及相关信息的选择查询,并为其命名为“按姓名成绩查询”。也可以通过设计视图直接创建该查询。

(2) 打开“按姓名成绩查询”,切换到设计视图,在“姓名”字段的“条件”属性中,在方括号内输入相应的提示文本([请输入姓名:]),如图 5.11 所示。

**注意:** 不能省略方括号。



图 5.11 设置输入参数

输入完毕后,将查询切换到数据表视图,这时在屏幕中就会出现一个对话框。此时,输入查询条件就可以找到满足条件的记录。

不仅可以建立单个字段的参数查询,还可以根据需要同时为多个字段建立参数查询。

## 5.4 聚合查询

使用选择查询检索和显示特定记录是数据分析的基本功能。数据分析的范围宽泛,包括分组和比较数据,更新和删除数据,对数据执行计算以及报告数据等。Access 提供专门的内置函数来实现数据分析与处理。

聚合查询有时也称为分组查询,创建聚合查询有助于快速分组和汇总数据。因为使用传统的选择查询只能检索显示数据源中的数据,而使用聚合查询则可以对检索数据进行诸如总计、平均值、计数等汇总运算。

### 5.4.1 创建聚合查询

例如,创建聚合查询,求每个学生的平均成绩。

过程如下。

(1) 在导航窗格,复制学生成绩查询,并为其命名为“学生成绩汇总查询”。

(2) 打开“学生成绩汇总查询”,切换到设计视图,如图 5.12 所示。

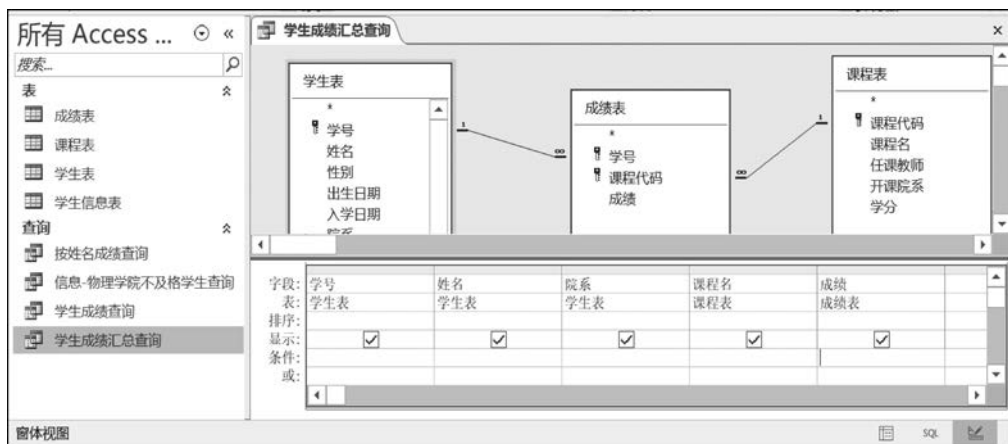


图 5.12 学生成绩汇总查询设计视图

在图 5.12 中,可以发现没有汇总属性。此时,需要在设计视图窗口中激活“汇总”属性。具体做法如下。

先转到功能区,选择“设计”选项卡,然后单击“汇总”按钮,在设计窗口中激活“汇总”属性。此时,在 QBE 窗格中会出现新的一行,名为“总计”。“总计”行会指示 Access 在对指定字段执行聚合时使用什么聚合函数,其默认函数为 Group By,如图 5.13 所示。

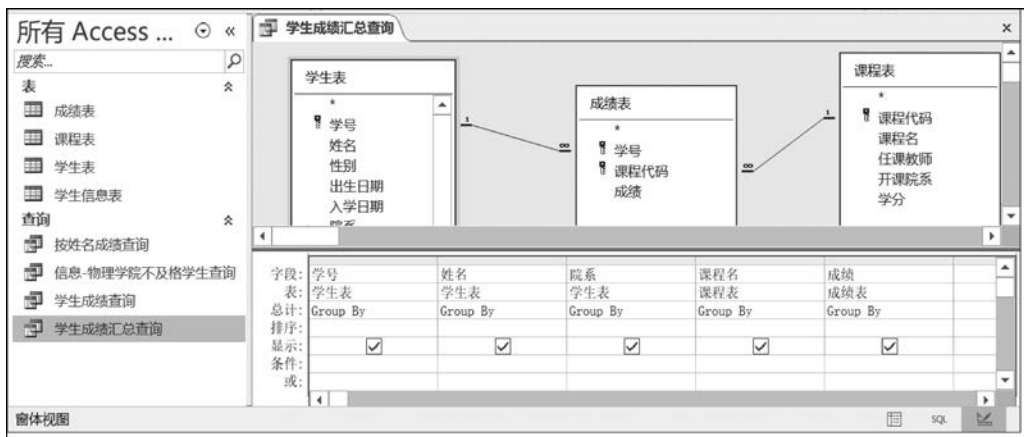


图 5.13 添加汇总属性

在图 5.13 中可以发现,“总计”行在网格的每个字段下都包含 Group By,这意味着将对字段中所有类似的记录进行分组。后面将介绍不同的聚合函数,这里要做的是调整“总计”行中的聚合函数,来获得所需要的数据分析结果。

(3) 删除“课程名”字段,将“成绩”字段的聚合函数从 Group By 更改为“平均值”。目的在于告诉 Access 对“成绩”字段中的内容求平均值,如图 5.14 所示。

(4) 运行查询,结果如图 5.15 所示。

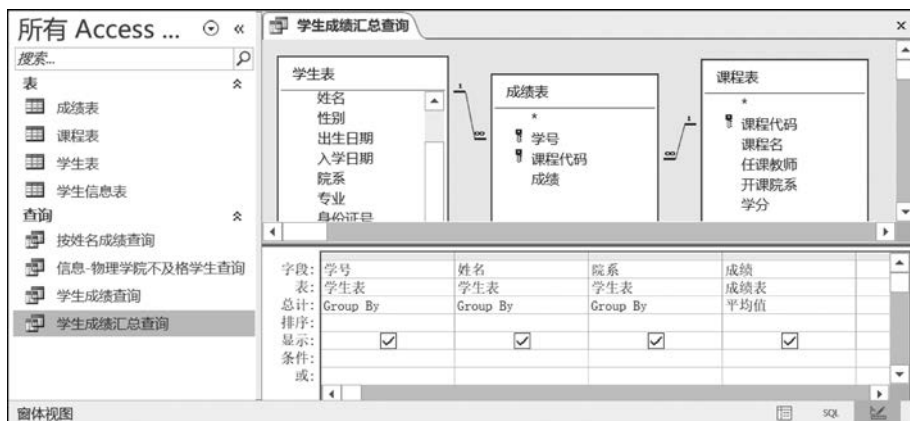


图 5.14 选择聚合函数



图 5.15 成绩求平均值结果

在图 5.15 中可以发现,系统为汇总的列命名为“成绩之平均值”,这是 Access 添加新列时的传统命名方法。为了便于理解,可以为新增加的列创建别名。

(5) 切换到设计视图,在需要添加别名的字段前添加别名,并在别名后跟一个冒号,如图 5.16 所示(例中为“平均成绩:成绩之平均值”,注意:冒号为英文冒号)。

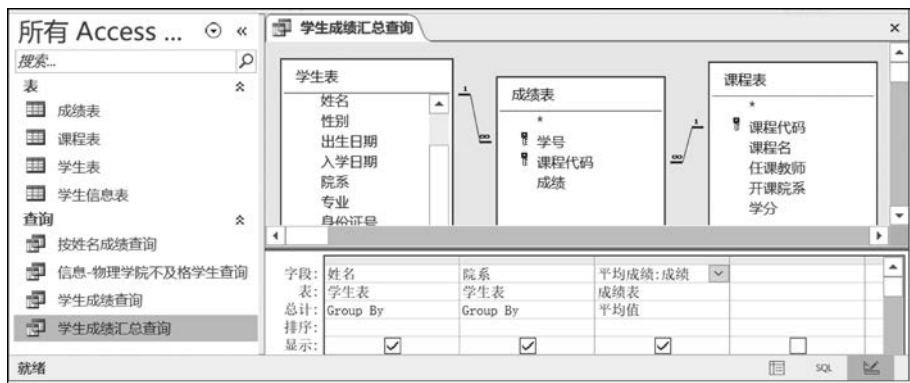


图 5.16 输入别名

## 5.4.2 常见的聚合函数

在 Access 中,提供了 12 种聚合函数,如图 5.17 所示。了解每种聚合函数的功能对于数据分析有不同的意义。

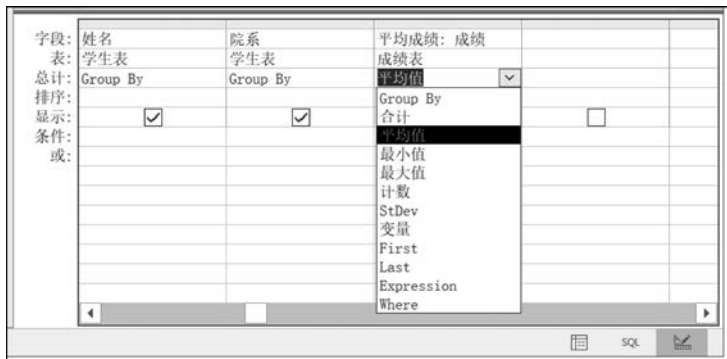


图 5.17 12 种聚合函数

### 1. Group By

Group By 聚合函数按指定字段中对所有记录进行分组。在使用 Group By 聚合函数时,需要注意以下几点。

- (1) Access 会在其他任何聚合函数之前执行 Group By 函数。
- (2) Access 对每个分组依据分组字段进行升序排序。如果查询有多个分组依据字段,将从最左侧的字段开始,对每个字段进行升序排序。
- (3) Access 将多个分组依据字段视为一个唯一项。

### 2. 合计、平均值、计数、StdDev、变量

这类聚合函数都是针对选定字段中的记录执行数学计算。需要注意的是,计算时这些函数将排除值为 Null 的记录。也就是说,这些聚合函数将忽略空单元格。

#### 1) 合计

计算指定字段或分组中所有记录的合计值。该函数只能用于自动编号、货币、日期时间以及数字。

#### 2) 平均值

计算指定字段或分组中所有记录的平均值。该函数只能用于自动编号、货币、日期时间以及数字。

#### 3) 计数

对指定字段或分组中的条目数进行计数。该函数可用于所有数据类型。

#### 4) StDev

计算指定字段或分组中所有记录的标准偏差。该函数只能用于自动编号、货币、日期时间以及数字。

#### 5) 变量

计算指定字段或分组中的所有值与组平均值的差量。该函数只能用于自动编号、货币、日期时间以及数字。

### 3. 最小值、最大值、First、Last

与其他聚合函数的不同之处在于,这4个函数将对指定字段或分组中的所有记录进行求值,并从该组返回单个值。

#### 1) 最小值

返回指定字段或分组中记录的最低值。该函数只能用于自动编号、货币、日期时间、数字以及文本。

#### 2) 最大值

返回指定字段或分组中记录的最高值。该函数只能用于自动编号、货币、日期时间、数字以及文本。

#### 3) First

返回指定字段或分组中第一条记录的值。该函数可用于所有数据类型。

#### 4) Last

返回指定字段或分组中最后一条记录的值。该函数可用于所有数据类型。

### 4. Expression、Where

聚合查询有一条固定的规则就是必须针对每个字段执行聚合,但某些情况下需要使用字段执行计算或应用筛选器。此时,这些字段是获得最终分析的一种方式,而不是最终分析的一部分,这时就需要使用 Expression 函数或 Where 子句。Expression 函数和 Where 子句的独特之处在于它们本身不执行任何分组操作。

#### 1) Expression

Expression 聚合函数一般是在聚合查询中利用自定义计算或其他函数时应用。Expression 指示 Access 针对每个记录或组分别执行指定的自定义计算。

#### 2) Where

Where 子句允许对未包含在聚合查询中的字段应用条件。

图 5.18 描述了 Where 的使用,例中,求信息学院学生的平均成绩。



图 5.18 通过 Where 求信息学院学生的平均成绩

注意: Where 子句的一个要点是“显示”行中对应字段的复选框中不能有复选标记。这是因为通过 Where 子句标记的字段无法在聚合查询中显示。因此,该复选框必须为空。

## 5.5 交叉表查询



### 5.5.1 交叉表查询的概念

交叉表查询以一种独特的概括形式返回一个表内的总计数字,这种概括形式是其他查询无法完成的。交叉表查询为用户提供了非常清楚的汇总数据,便于分析和使用。

交叉表查询将来源于某个表中的字段进行分组,一组列在交叉表左侧,一组列在交叉表上部,并在交叉表行与列交叉处显示表中某个字段的计算值。

在创建交叉表查询时,需要指定 3 种字段:一是放在交叉表最左端的行标题,它将某一字段的相关数据放入指定的行中;二是放在交叉表最上面的列标题,它将某一字段的相关数据放入指定的列中;三是放在交叉表行与列交叉位置上的字段,需要为该字段指定一个总计项,如总计、平均值、计数等。在交叉表查询中,只能指定一个列字段和一个总计类型的字段。

### 5.5.2 使用交叉表查询向导创建交叉表查询

使用交叉表查询向导是创建交叉表查询最快、最简单的方法。向导会为用户完成大部分工作,使用向导创建一个交叉表查询的步骤如下。

(1) 打开 xsgl 数据库,选择“创建”选项卡,单击“查询/查询向导”项,弹出“新建查询”对话框。

(2) 在“新建查询”对话框中选择“交叉表查询向导”项,单击“确定”按钮,弹出“交叉表查询向导”对话框,如图 5.19 所示。



图 5.19 创建交叉表查询-确定数据源

在“视图”列表中选择“表”“查询”和“两者”之一。注意,使用向导创建交叉表查询时,只能选择一个数据源。如果交叉查询中包含多个表中的字段,则先创建一个含有所需全部字段的查询,然后用这个查询创建交叉表查询。

本例中,以学生成绩查询作为数据源。

(3) 单击“下一步”按钮,出现“选择行标题字段”对话框。在创建交叉表查询时,需要指定哪些字段包含行标题,哪些字段包含列标题,以及哪些字段包含要汇总的值。在指定行标题时,最多可使用三个字段。使用的行标题越少,交叉表查询数据表就越容易阅读。

本例中选择“姓名”作为行标题字段。如图 5.20 所示。



交叉表查询向导

请确定用哪些字段的值作为行标题:

最多可以选择 3 个字段。

请按所需的信息排序顺序选择字段。  
例如,可以先按国家再按地区对值进行排序和分组。

可用字段:

- 学号
- 院系
- 课程名
- 成绩

选定字段:

- 姓名

示例:

| 姓名  | 标题1 | 标题2 | 标题3 |
|-----|-----|-----|-----|
| 姓名1 | 总计  |     |     |
| 姓名2 |     |     |     |
| 姓名3 |     |     |     |
| 姓名4 |     |     |     |

取消 < 上一步(B) 下一步(N) > 完成(F)

图 5.20 创建交叉表查询-选择行标题

(4) 单击“下一步”按钮,出现“选择列标题字段”对话框。交叉表中只能有一个列标题,要慎重选择列标题字段。

本例中选择“课程名”作为列标题字段。如图 5.21 所示。



交叉表查询向导

请确定用哪个字段的值作为列标题:

例如,为了用每位雇员的姓名作为列标题,可以选择“雇员姓名”。

可用字段:

- 学号
- 院系
- 课程名
- 成绩

示例:

| 姓名  | 课程名1 | 课程名2 | 课程名3 |
|-----|------|------|------|
| 姓名1 | 总计   |      |      |
| 姓名2 |      |      |      |
| 姓名3 |      |      |      |
| 姓名4 |      |      |      |

取消 < 上一步(B) 下一步(N) > 完成(F)

图 5.21 创建交叉表查询-选择列标题字段

(5) 单击“下一步”按钮,选择在表中的交叉点计算出什么数值,所选字段的数据类型将决定哪些函数可用。

本例中,在“字段”选项中选择“成绩”,“函数”选项中选择“最大”,然后选定“是,包含各行小计”复选框。如果包含行小计,则交叉表查询中有一个附加行标题,该标题与字段值使用相同的字段和函数。包含行小计还会插入一个对其余列进行汇总的附加列,如图 5.22 所示。

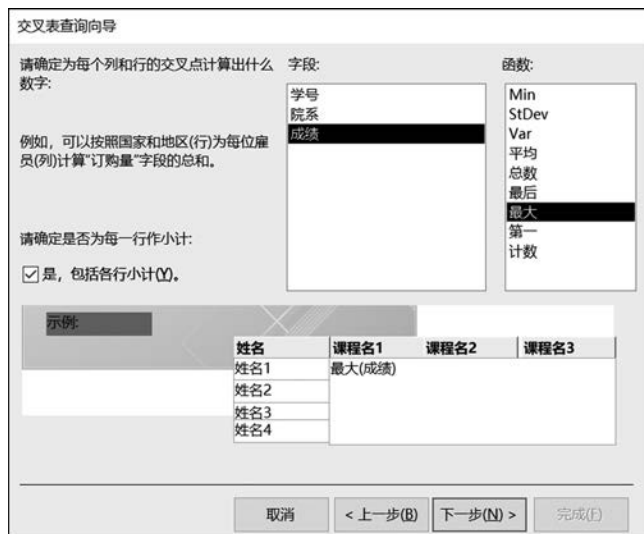


图 5.22 创建交叉表查询-选择计算内容

(6) 单击“下一步”按钮,为新建的查询取名(学生成绩\_交叉表查询),并单击“完成”按钮。这样一个交叉表查询就完成了,运行查询,结果如图 5.23 所示。

| 学生成绩 交叉表查询            |    |    |       |       |     |      |       |    |     |     |     | x   |         |
|-----------------------|----|----|-------|-------|-----|------|-------|----|-----|-----|-----|-----|---------|
| 姓名                    | 总计 | 成绩 | 程序设计基 | 多媒体技术 | 计算机 | 数据结构 | 数据库基础 | 体育 | 英语1 | 英语2 | 英语3 |     |         |
| 董江                    |    | 98 | 85    | 75    | 67  | 66   | 89    | 65 | 68  | 98  | 78  |     |         |
| 韩承敏                   |    | 98 | 68    | 68    | 76  | 75   | 86    | 67 | 66  | 98  | 80  |     |         |
| 马苗苗                   |    | 88 | 86    | 87    | 88  | 77   | 67    | 65 | 76  | 87  | 85  |     |         |
| 马玉                    |    | 76 | 50    | 76    | 76  | 67   | 51    | 67 | 64  | 70  |     |     |         |
| 吴琦恒                   |    | 90 | 76    | 65    | 90  | 75   | 87    | 78 | 65  | 87  | 76  |     |         |
| 张云                    |    | 89 | 89    | 85    | 77  | 76   | 86    | 75 | 87  | 80  | 75  |     |         |
| 周强                    |    | 88 | 88    | 79    | 88  | 65   | 78    | 50 | 75  | 85  | 87  |     |         |
| 记录: ( 第 1 项 (共 7 项) ) |    |    |       |       |     |      |       |    |     |     |     |     | 无筛选器 搜索 |
|                       |    |    |       |       |     |      |       |    |     |     |     | SQL |         |

图 5.23 运行结果

### 5.5.3 通过设计视图创建交叉表查询

尽管通过“交叉表查询向导”只需要几次简单的选择操作便可轻松地创建交叉表,但它存在一些限制会降低数据分析的功效,通过查询向导建立交叉表查询存在以下不足之处。

(1) 只能为交叉表选择一个基础数据源。

这意味着如果需要对保存在多个表中的数据执行交叉表查询,必须通过额外的步骤创建一个临时查询作为数据源。

(2) 在“交叉表查询向导”中无法使用条件对交叉表查询进行筛选或限制。

(3) 最多只能使用三个行标题。

(4) 不能在“交叉表查询向导”中显式定义列标题的顺序。

这些不足均可以通过查询设计视图创建交叉表查询来解决。通过查询设计视图创建交叉表查询的一个优势在于不必只使用一个数据源,可以使用多个数据源定义查询中的字段。

通过设计视图创建交叉表查询的步骤如下。

(1) 首先通过设计视图创建一个聚合查询,如图 5.24 所示。

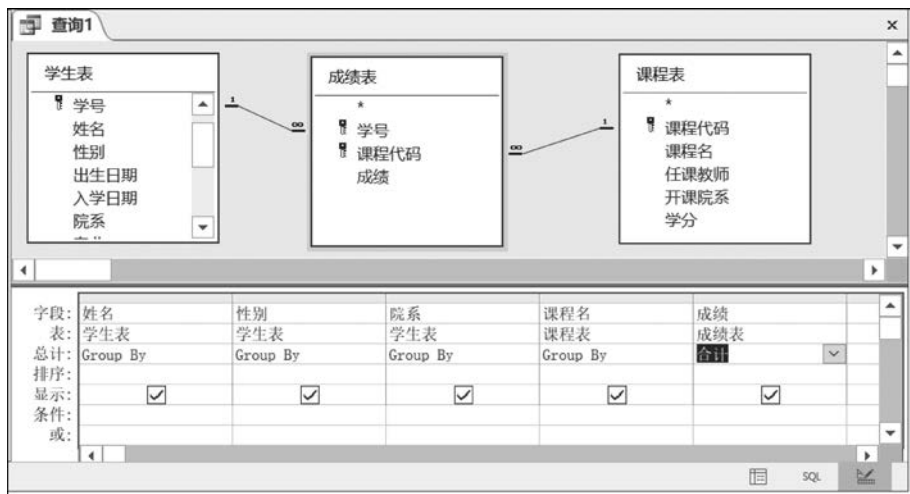


图 5.24 创建聚合查询

(2) 选择功能区的“设计”选项卡,然后单击“交叉表”按钮。此时 QBE 窗格内添加了名为“交叉表”的行,如图 5.25 所示。该行的用途是定义每个字段将在交叉表查询中扮演的角色。

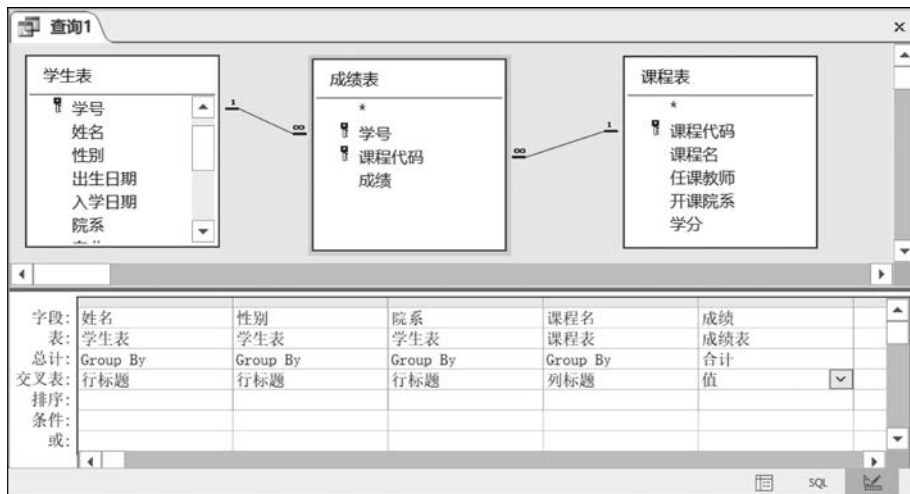


图 5.25 设置各字段的角色

(3) 运行该查询,交叉表的实际效果如图 5.26 所示。

在查询设计视图中构建交叉表查询时,有以下两点需要注意。

- (1) 必须至少有一个行标题、一个列标题和一个值字段。
- (2) 并未限制只能有三个行标题。

| 姓名  | 性别 | 院系   | 程序设计基 | 多媒体技术 | 计算机 | 数据结构 | 数据库基础 | 体育 | 英语1 | 英语2 | 英语3 |
|-----|----|------|-------|-------|-----|------|-------|----|-----|-----|-----|
| 董江  |    | 外语学院 | 85    | 75    | 67  | 66   | 89    | 65 | 68  | 98  | 78  |
| 韩承敏 | 男  | 信息学院 | 68    | 68    | 76  | 75   | 86    | 67 | 66  | 98  | 80  |
| 马苗苗 | 女  | 外语学院 | 86    | 87    | 88  | 77   | 67    | 65 | 76  | 87  | 85  |
| 马玉  | 女  | 物理学院 | 50    |       | 76  |      | 67    | 51 | 67  | 64  | 70  |
| 吴琦恒 | 男  | 信息学院 | 76    | 65    | 90  | 75   | 87    | 78 | 65  | 87  | 76  |
| 张云  | 女  | 物理学院 | 89    | 85    | 77  | 76   | 86    | 75 | 87  | 80  | 75  |
| 周强  | 男  | 信息学院 | 88    | 79    | 88  | 65   | 78    | 50 | 75  | 85  | 87  |

图 5.26 通过查询设计视图建立交叉表查询的运行结果

## 5.6 操作查询

在对数据库进行维护时,常常需要大量地修改数据。例如,删除成绩小于 60 分的记录,将成绩在 90 分以上的记录存储到一个新表中等。这些操作既要检索记录,又要更新记录,操作查询能够实现这样的功能。操作查询包括生成表查询、删除查询、更新查询和追加查询 4 种。



### 5.6.1 生成表查询

生成表查询是利用一个或多个表中的全部或部分数据建立新表。在 Access 中,从表中访问数据要比从查询中访问数据快得多,因此如果经常要从几个表中提取数据,最好的方法是使用生成表查询,将从多个表中提取的数据组合起来生成一个新表。

建立一个生成表查询,将成绩在 80 分以上的学生存放到“80 分以上学生情况”表中。操作步骤如下。

(1) 通过查询设计视图创建选择查询,设置选择条件,如图 5.27 所示。

| 字段: | 学号                                  | 姓名                                  | 院系                                  | 课程名                                 | 成绩                                  |
|-----|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 表:  | 学生表                                 | 学生表                                 | 学生表                                 | 课程表                                 | 成绩表                                 |
| 排序: |                                     |                                     |                                     |                                     |                                     |
| 显示: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 条件: |                                     |                                     |                                     |                                     | >80                                 |
| 或:  |                                     |                                     |                                     |                                     |                                     |

图 5.27 设置选择条件

(2) 选择“设计”选项卡,在“查询类型”中单击“生成表”按钮,弹出“生成表”对话框。在“表名称”文本框中输入“80 分以上学生情况表”,如图 5.28 所示。



图 5.28 “生成表”对话框

(3) 单击“确定”按钮,保存查询,名称为 80 分以上学生情况表,生成查询。

(4) 运行查询,可以发现产生了一个新表,即 80 分以上学生情况表。

### 5.6.2 删除查询

删除查询是所有操作查询中最具破坏性的一个。查询所使用的字段只是用来作为查询的条件,可以从单个表删除记录,也可以通过级联删除相关记录而从相关表中删除记录。

创建删除查询的过程如下。

(1) 通过查询设计视图创建一个选择查询,如图 5.29 所示。



图 5.29 创建删除查询

(2) 选择“设计”选项卡,在“查询类型”工具栏中单击“删除”按钮。

(3) 输入删除条件。例中,在学号字段的条件域输入“2021116099”。

(4) 保存删除查询,名字为“按学号删除查询”。

(5) 运行删除查询。运行时,系统会出现删除提示信息,单击“是”按钮便可完成删除。

### 5.6.3 更新查询

更新查询用于同时更改多个记录中的一个或多个字段值,用户可以通过添加条件来选择所要更新的记录。大部分更新查询可以用表达式来规定更新规则。表 5.10 列举了常见

的更新规则表达式。

表 5.10 更新规则的常用表达式

| 字段类型 | 表达式         | 含义                     |
|------|-------------|------------------------|
| 货币   | [单价] * 1.15 | 把单价增加 15%              |
| 日期   | #9/19/09#   | 把日期更改为 2009 年 9 月 19 日 |
| 文本   | “完成”        | 把数据更改为“完成”             |
| 文本   | “平均”&[单价]   | 把字符“平均”连接到“单价”字段数据     |
| 是/否  | Yes         | 把特定的“否”数据更改为“是”        |

例中选择“80 分以上学生情况表”作为数据源,首先创建一个选择查询。

然后选择“设计”选项卡,在“查询类型”工具栏中单击“更新”按钮。

在“更新为”属性栏中输入更新条件,例中求所有成绩的算术平方根(进行四舍五入),将“更新为”设置为 Round(SQR[成绩]),如图 5.30 所示。

保存查询并运行,可以发现,表中所有记录均被更新。



图 5.30 更新查询

#### 5.6.4 追加查询

追加查询可将一组记录(行)从一个或多个源表(或查询)添加到另一个目标表。追加查询也可以从另一个数据库表中读取数据记录并向当前表内添加记录,由于两个表之间的字段定义可能不同,追加查询只能添加相互匹配的字段内容,而那些不匹配的字段将被忽略。

特别要注意的是,不能使用追加查询来更改现有记录个别字段中的数据。要执行此类任务,应使用更新查询,追加查询只能用来添加数据行。

例如,建立一个追加查询,将不及格的学生信息添加到已建立的“80 分以上学生情况表”中,操作步骤如下。

(1) 首先通过查询设计视图创建一个选择查询,如图 5.31 所示。

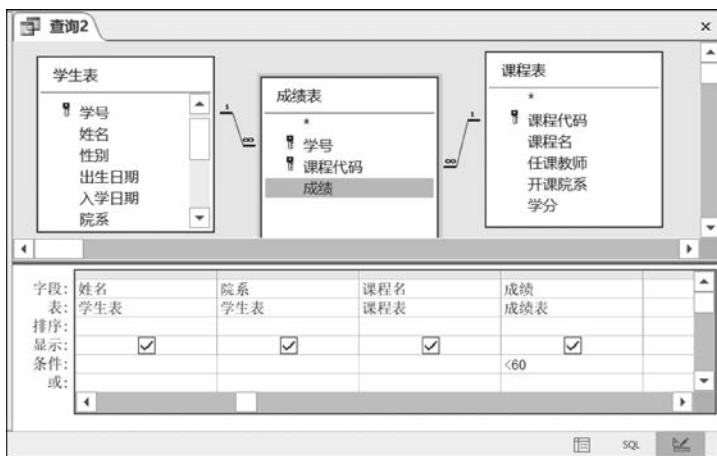


图 5.31 通过设计视图创建选择查询

(2) 打开“设计”选项卡,在“查询类型”工具栏中单击“追加”按钮,弹出“追加”对话框。通过“表名称”文本框中右侧下拉按钮选择“80分以上学生情况表”,表示将查询的记录追加到“80分以上学生情况表”中,同时选中“当前数据库”单选按钮,如图 5.32 所示。



图 5.32 追加对话框

(3) 保存并运行查询,发现不及格的成绩已经追加。

在执行操作查询时,可能更改许多记录,并且在执行操作查询后,不能撤销刚做过的更改。因此,在运行操作查询之前,最好先单击工具栏上的“视图”按钮,预览即将更改的记录,如果预览结果正确,再执行操作查询,这样可防止误操作。另外,在使用操作查询之前,最好备份数据。

## 5.7 Access SQL 查询

SQL(Structured Query Language,结构化查询语言)是关系数据库管理系统的标准语言。SQL 语句可以用于多种数据库操作,例如,更新表中的数据,从表中提取数据等。目前,绝大多数流行的关系数据库管理系统,如 Oracle、Sybase、Microsoft SQL Server、Access 等都支持 SQL。虽然很多 DBMS 都对 SQL 语句进行了再开发和扩展,但是包括 SELECT、INSERT、UPDATE、DELETE、CREATE 及 DROP 在内的标准 SQL 命令仍然可以用来完成几乎所有的数据库操作。

### 5.7.1 SQL 概述

#### 1. SQL 的特点

SQL 之所以能够成为国际标准并为用户所接受,主要原因在于它是一种综合的、通用的、功能强大且简单易学的语言。SQL 集数据查询、数据操纵、数据定义和数据控制功能于一体,充分体现了关系数据语言的优点。其主要包括以下特点。

##### 1) 一体化特性

包括数据定义、数据查询、数据操纵和数据控制等方面的功能,可以完成数据库活动中的全部工作。

##### 2) 高度非过程化

它无须告诉计算机如何去做,而只需要用户描述清楚要做什么,SQL 就可以将要求交给系统,自动完成全部工作。

##### 3) 语言简洁

虽然 SQL 功能很强,但它只有为数不多的几条命令,另外 SQL 的语法也非常简单,它



很接近自然语言,因此容易学习掌握。

#### 4) 支持多种使用方式

SQL 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 直接融入自身的语言之中,使用起来更方便。尽管 SQL 的使用方式不同,但 SQL 的语法基本是一致的。

总之,SQL 功能强大,语言简洁。完成数据定义、数据操纵、数据控制的核心功能只用了 9 条语句: CREATE、DROP、ALTER、SELECT、INSERT、UPDATE、DELETE、GRANT、REVOKE,如表 5.11 所示。

表 5.11 SQL 操作的语句

| SQL 功能 | 语 句                 | SQL 功能 | 语 句                    |
|--------|---------------------|--------|------------------------|
| 数据查询   | SELECT              | 数据操纵   | INSERT, UPDATE, DELETE |
| 数据定义   | CREATE, DROP, ALTER | 数据控制   | GRANT, REVOKE          |

## 2. Access SQL 语句与 SQL 的区别

Access 中提供查询对象,在设计时可以采用设计视图和 Access SQL 视图,非常方便,Access SQL 视图中的 SQL 语句可以在 SQL Server 中使用,但两者之间还有一些微小的差别。表 5.12 总结了 Microsoft Access SQL 与 Microsoft SQL Server 中 SQL 语法的区别。

表 5.12 Access SQL 与 Microsoft SQL Server 中 SQL 语法的区别

| SQL 语法元素     | Microsoft Access SQL                                                 | Microsoft SQL Server                                                                      |
|--------------|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| 标识符          | 限制不超过 64 个字符;<br>允许使用关键字和特殊字符;<br>可以用任何字符开头                          | SQL Server 6.5: 限制不超过 30 个字符;不允许使用关键字和特殊字符;必须用字母字符开头。<br>SQL Server 7.0 的标识符与 Access 完全兼容 |
| 输出字段         | 允许多个输出字段具有相同名称                                                       | 在视图中不支持多个相同输出字段名                                                                          |
| 日期分隔符号       | 英镑符(#)                                                               | 撇号(')                                                                                     |
| Boolean 常量   | True、False; On、Off; Yes、No                                           | 整数: 1(真)、0(假)                                                                             |
| 字符串连接        | 和号(&)                                                                | 加号(+)                                                                                     |
| 通配符          | 星号(*)与零个或更多字符匹配;<br>问号(?)与单个字符匹配;<br>叹号(!)意味着不在列表中;<br>英镑符(#)意味着单个数字 | 百分号(%)与零个或更多字符匹配;<br>下画线(_)与单个字符匹配;<br>上插入符(^)意味着不在列表中;没有与英镑符(#)对应的字符                     |
| TOP          | 如果有一个 ORDER BY 子句,自动包含层次                                             | SQL Server 6.5 不支持;<br>SQL Server 7.0 需要一个明确的 WITH TIES 子句                                |
| CREATE INDEX | 允许创建升序和降序索引;<br>允许声明主键,没有 Null 值,并且忽略 Null 值                         | 不支持                                                                                       |
| DROP INDEX   | 语法是:<br>DROP INDEX <index name> ON<br><table name>                   | 语法是:<br>DROP INDEX <table name>, <index name>                                             |

续表

| SQL 语法元素              | Microsoft Access SQL                  | Microsoft SQL Server |
|-----------------------|---------------------------------------|----------------------|
| DISTINCTROW           | 支持(允许选择单个记录)                          | 不支持                  |
| OWNERACCESS           | 支持(在执行时控制许可权)                         | 不支持                  |
| TABLE in UNION        | 支持(允许使用下列语法指定表:<br>TABLE <tablename>) | 不支持                  |
| ORDER BY in<br>UNIONS | 支持。允许通过联合查询中的子句实现多种排序                 | 支持。允许通过语句末尾的子句实现一种排序 |
| TRANSFORM             | 支持。用于交叉表查询                            | 不支持                  |
| PARAMETERS            | 支持(在 SQL 中记录)                         | 不支持                  |

5.7.2 数据定义



SQL 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、存储过程的定义、规则的定义和索引的定义等多个部分。

SQL 中数据定义的基本语句有如下四个。

(1) 建立新表:

CREATE TABLE ...

(2) 添加字段:

ALTER TABLE ... ADD ...

(3) 删除字段:

ALTER TABLE ... DROP ...

(4) 基本表删除:

DROP TABLE ...

1. 基本数据类型

在 Access 中使用 SQL 语句时,可以使用的基本数据类型及其特点如表 5.13 所示。

表 5.13 基本数据类型及其特点

| 数据类型             | 存储大小   | 说 明                                                      |
|------------------|--------|----------------------------------------------------------|
| BINARY           | 每字符 1B | 任何类型的数据都可存储在这种类型的字段中                                     |
| BIT              | 1B     | Yes/No(True/False, ON/OFF, - 1/0) 只包含两值之一的字段             |
| BYTE             | 1B     | 介于 0~255 的整数                                             |
| MONEY            | 8B     | 介于 - 922,337,203,685,477. 5808~922,337,203,685,477. 5807 |
| DATETIME         | 8 字节   | 介于 100~9999 年的日期或时间数值                                    |
| UNIQUEIDENTIFIER | 128b   | 用于远程过程调用的唯一识别数字                                          |
| REAL             | 4B     | 单精度浮点数                                                   |
| FLOAT            | 8B     | 双精度浮点数                                                   |
| SMALLINT         | 2B     | 介于 - 32,768~32,767 的短整型数                                 |
| INTEGER          | 4B     | 介于 - 2,147,483,648~2,147,483,647 的长整型数                   |
| DECIMAL          | 17B    | 可以定义精度(1~28)和符号(0~定义精度)。默认精度和符号分别是 18 和 0                |

续表

| 数据类型  | 存储大小    | 说明                    |
|-------|---------|-----------------------|
| TEXT  | 每字符 2B  | 从 0 到最大 2GB           |
| IMAGE | 视实际需要而定 | 从 0 到最大 2GB 用于 OLE 对象 |
| CHAR  | 每字符 2B  | 长度从 0 到 255 字符        |

说明:

- (1) BINARY: 二进制型,可以指定长度。
- (2) BIT: 位型,可用格式(Yes/No, True/False, OH/OFF)。
- (3) BYTE: 数字-字节,不要指定长度和精度,否则会报错。
- (4) MONEY,CURRENCY: 货币型,不要指定长度和精度,否则会报错。
- (5) DATETIME: 日期时间型,不要指定长度,否则会报错。
- (6) UNIQUEIDENTIFIER: 用于远程过程调用的唯一识别数字。
- (7) REAL,SINGLE: 数字-单精度型,不要指定长度和精度,否则会报错。
- (8) FLOAT,DOUBLE,NUMBER: 数字-双精度型,不要指定长度和精度,否则会报错。
- (9) SMALLINT,SHORT: 数字-整型,不要指定长度和精度,否则会报错。
- (10) INTEGER,INT,LONG: 数字-长整型,不要指定长度和精度,否则会报错。
- (11) DECIMAL: 数字-小数,可以指定长度和精度,如只指定长度,那么精度默认为 0,如都不指定,那么默认长度为 18,默认精度为 0。
- (12) TEXT: 文本型(指定长度时)。
- (13) IMAGE,OLEOBJECT: OLE OBJECT 型,不要指定长度,否则会报错。
- (14) CHAR,NCHAR,VARCHAR,NVARCHAR: 文本型,可以指定长度,否则默认值为 255。
- (15) MEMO: 备注型,不要指定长度,否则会报错。

## 2. 在 Access 中使用 SQL

在数据库窗口中,选择“创建”选项卡,单击“查询”中的“查询设计”按钮,弹出“显示表”对话框,单击“关闭”按钮,系统将建立一个名为“查询 1”的空查询。单击功能区的 SQL 视图按钮,切换到 SQL 视图,如图 5.33 所示。

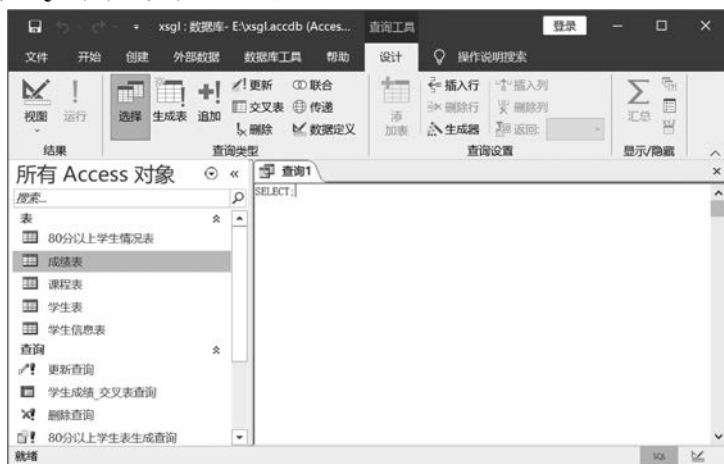


图 5.33 SQL 视图

在图 5.33 中,系统默认为 SELECT 命令。删除该命令,输入完成特定功能的 SQL 命令,然后单击功能区的“运行”按钮,便可运行对应的 SQL 命令。

### 3. 表的建立

SQL 中的 CREATE TABLE 语句用来建立新表。

CREATE TABLE 语句的使用格式如下。

```
CREATE TABLE [数据库名.]表名  
(列名 1 类型 [(宽度,[小数位数])] [NULL/NOT NULL],  
列名 2 类型 [(宽度,[小数位数])] [NULL/NOT NULL],  
...  
...)
```

说明:

[数据库名.]: 用于指明所建立的表隶属于哪个数据库。

表名: 所创建表的名称。

列名: 所建立表的字段名。

类型: 指明对应字段的数据类型。常见的基本类型及其符号表示如表 5.13 所示。

[(宽度,[小数位数])]: 指明对应字段的宽度。如有小数部分,还需指出小数的位数。

[NULL/NOT NULL]: 指明字段是否可以取空值。

**【例 5.1】** 用 SQL 在 xsgl 数据库中建立基本表 JBQK、CJ 和 KC。

(1) 创建基本情况表 JBQK。

通过查询设计视图创建查询,并切换到 SQL 视图,输入如下命令,结果如图 5.34 所示。



图 5.34 SQL 视图

```
CREATE TABLE JBQK  
(SNO CHAR(4) NOT NULL,  
SNAME CHAR(8) NOT NULL,  
AGE BYTE,  
GENDER CHAR(2),  
DEPT CHAR(12))
```

其中:JBQK 为表名,SNO,SNAME,AGE,SEX 为字段名,NOT NULL 用于说明列值不能为空。

单击“运行”按钮,系统运行 SQL 命令,便会生成基本表 JBQK。后面的 SQL 命令都按此法创建并运行。

#### (2) 创建成绩表 CJ。

```
CREATE TABLE CJ
(SNO CHAR(4) NOT NULL,
KNO CHAR(4) NOT NULL,
ACHIEVEMENT SINGLE)
```

#### (3) 创建课程表 KC。

```
CREATE TABLE KC
(KNO CHAR(4),KNAME CHAR(10),GRADE SHORT)
```

#### 注意:

(1) 使用 CREATE TABLE 语句创建基本表时,最初得到的只是一个空的框架(表结构),用户可以使用 INSERT 命令插入内容。

(2) 使用 SQL 语句创建表时,表名以及表中字段名称必须以字母开头,后面可以使用字母、数字或下画线。

### 4. 修改表结构

基本表创建以后,经过一段时间的使用,表的结构可能会无法满足实际的要求,这时就需要对表的结构进行修改,例如,增加新字段或者删除无用字段。

#### (1) 增加新字段。

语句格式:

```
ALTER TABLE 表名 ADD 字段名 类型 [(宽度[,小数位数])]
```

**【例 5.2】** 在基本情况表 JBQK 中增加 ADDRESS 字段。

```
ALTER TABLE JBQK ADD ADDRESS CHAR (40)
```

#### (2) 删除无用字段。

语句格式:

```
ALTER TABLE 表名 DROP 字段名
```

**【例 5.3】** 删除基本情况表 JBQK 中的 ADDRESS 字段。

```
ALTER TABLE JBQK DROP ADDRESS
```

#### (3) 修改字段的类型宽度等。

语句格式:

```
ALTER TABLE 表名 ALTER 字段名 新类型 [(新宽度[,小数位数])]
```

**【例 5.4】** 将成绩表 CJ 中的 ACHIEVEMENT 字段的类型改为双精度型。

```
ALTER TABLE CJ ALTER ACHIEVEMENT FLOAT
```

### 5. 删除基本表

在 SQL 中使用 DROP TABLE 命令删除某个表格及该表格中的所有记录。DROP TABLE 命令的使用格式为：

```
DROP TABLE 表名
```

**【例 5.5】** 删除基本情况表 CJ。

```
DROP TABLE CJ
```

## 5.7.3 数据更新

Access SQL 的数据更新包括插入数据、删除数据、修改数据三种基本操作。

### 1. 插入数据

Access SQL 使用 INSERT 语句向表中插入或添加新的数据行。INSERT 语句的基本格式为：

```
INSERT INTO 基本表名(列名表)  
VALUES(元组值)
```

**【例 5.6】** 给课程表 KC 添加一条记录。

```
INSERT INTO KC  
VALUES ("1-06", "C 语言", 4)
```

简单来说,当向数据库表中添加新记录时,在关键词 INSERT INTO 后面输入所要添加的表名称,然后在括号中列出将要添加新值的列的名称。若省略了列名表,则按表结构中列的顺序输入相应的内容。

### 2. 删除数据

SQL 使用 DELETE 语句删除数据库表格中的行或记录。DELETE 语句的格式为：

```
DELETE FROM 基本表名[WHERE 条件表达式]
```

从基本表中删除满足条件的表达式的元组,该语句每次只能从一个基本表中删除元组。当无条件时,表示删除所有的元组。

**【例 5.7】** 在课程表 KC 中,删除课程编号 KNO 为 1-06 的课程。

```
DELETE FROM KC  
WHERE KNO="1-06"
```

**【例 5.8】** 在成绩表 CJ 中,将课程编码为“0033”的课程成绩小于该科平均成绩的元组从成绩表 CJ 中删除。

```
DELETE *  
FROM CJ  
WHERE KNO="0033" AND
```



```
ACHIEVEMENT <(SELECT AVG(ACHIEVEMENT)
FROM CJ
WHERE KNO="0033")
```

简单来说,当需要删除某一行或某个记录时,在 DELETE FROM 关键词之后输入表格名称,然后在 WHERE 从句中设定删除记录的判断条件。注意,如果用户在使用 DELETE 语句时省略 WHERE 从句,则表格中的所有记录将全部被删除。

### 3. 修改数据

SQL 使用 UPDATE 语句更新或修改满足规定条件的记录。UPDATE 语句的格式为:

```
UPDATE 基本表名
SET 列名=值表达式[,SET 列名=值表达式...]
[WHERE 条件表达式]
```

命令功能:修改基本表中满足条件表达式的那些元组指定列的值,所需修改的列由 SET 子句指出。

**【例 5.9】** 在 JBQK 表中将学号(SNO)为 2014119091 的记录性别(GENDER)改为“女”。

```
UPDATE JBQK SET GENDER = "女"
WHERE SNO=" 2014119091"
```

**【例 5.10】** 将所有男同学的成绩提高 15%。

```
UPDATE CJ SET ACHIEVEMENT= ACHIEVEMENT * 1.15
WHERE SNO IN(SELECT SNO FROM JBQK WHERE GENDER="男")
```

使用 UPDATE 语句时,关键一点就是要设定好用于进行判断的 WHERE 条件从句。



## 5.7.4 数据查询

数据查询是对数据库进行的最基本的操作,查询效率的高低对软件有着重要的影响。在 SQL 中提供了 SELECT 查询语句,其功能强大且内容丰富。

### 1. SELECT 的语法格式

SELECT 的语法格式如下。

```
SELECT 目标表的列名序列
FROM 基本表视图序列
[INTO 目标位置]
[WHERE 行条件表达式]
[GROP BY 列名序列]
[HAVING 组条件表达式]
[ORDER BY 列名[ASC|DESC] ...]
```

[ ]: 表示可选项,用户根据实际需要进行选择。

各子句的含义及功能如下。

- (1) 目标表的列名序列:指明查询结果的字段构成,可以是字段名、表达式、常量等。
- (2) 基本表视图序列:用于指明查询信息的数据来源。
- (3) [INTO 目标位置]:指明查询结果的输出位置,输出位置包括:
  - ARRAY 数组名:存放指定的数组中。

- CURSOR 临时表名：存放到一个临时表中。
- TO FILE 文件名：存放到一个文件中。
- TO PRINTER：打印查询结果。
- TO SCREEN：将结果在屏幕上显示(默认方式)。

(4) [WHERE 行条件表达式]：用于在连接结果中选择满足条件的元组。

(5) [GROUP BY 列名序列]：用于对结果进行分组。分组记录的字段可以有多个,这些字段的顺序决定最高到最低的分组层次。

(6) [HAVING 组条件表达式]：用于选择满足条件的组。

(7) [ORDER BY 列名[ASC|DESC]...]：用于设置查询结果的排序方式。ASC 表示升序,DESC 表示降序。使用的目的是将查询的结果依照指定字段加以排序。若没有 ORDER BY,查询出的数据集将不会排序。

## 2. 语句的执行过程

SELECT 语句的执行过程如下。

- (1) 首先读取 FROM 子句中的基本表和视图,然后对其进行笛卡儿积运算。
- (2) 根据 WHERE 子句,选出满足条件表达式的元组。
- (3) 按照 GROUP 子句中指定字段的值进行分组,并从这些分组中选择满足 HAVING 子句中条件的分组。
- (4) 按照 SELECT 子句给出的字段求值得到目标表。
- (5) 用 ORDER 子句对目标表进行排序。

**【例 5.11】** 在 CJ 表中查询出所有成绩大于或等于 60 分的学生。

```
SELECT SNO,KNO, ACHIEVEMENT
FROM CJ
WHERE ACHIEVEMENT >=60
```

## 3. WHERE 子句中的运算符

SQL 是完备的,也就是说,只要数据是按关系方式存入数据库的,就能构造合适的 SQL 命令把它检索出来。事实上,SQL 不仅具有一般的检索能力,还可以通过在 WHERE 子句中加入运算符进行计算方式的检索。在 WHERE 子句中使用的一些运算符如表 5.14 所示。

表 5.14 WHERE 子句中的运算符

| 运 算 符 号   | 运 算 符                                         |
|-----------|-----------------------------------------------|
| 算术比较运算符   | =(等于)、>(大于)、<(小于)、>=(大于或等于)、<=(小于或等于)、<>(不等于) |
| 逻辑运算符     | AND、OR、NOT                                    |
| 集合运算符     | UNION(并)、INTERSECT(交)、EXCEPT(差)               |
| 集合成员资格运算符 | IN、NOT IN                                     |
| 谓词        | EXISTS(存在)、ALL、SOME、UNIQUE                    |
| 数学函数      | AVG、MIN、MAX、SUM、COUNT、FIRST、LAST              |
| 其他        | LIKE、BETWEEN... AND...                        |

通过使用 LIKE 运算符可以只选择与用户规定格式相同的记录。可以将一字符串与另一特定字符串样式比较,并将符合该字符串样式的记录过滤出来。若要查询出所有姓“李”的人,可以利用下面的语句。

---

```
LIKE "李 *"
```

---

**【例 5.12】** 计算学生成绩高于或等于 60 分的平均分。

---

```
SELECT AVG(ACHIEVEMENT) AS 平均分  
FROM CJ  
WHERE ACHIEVEMENT >= 60
```

---

**【例 5.13】** 统计出成绩表 CJ 中选课程号码 KNO 为“1-02”的学生人数。

---

```
SELECT COUNT(SNO) AS 人数  
FROM CJ  
WHERE KNO="1-02"
```

---

**【例 5.14】** 在基本情况表 JBQK 中找出姓名字段 SNAME 的第一条数据和学号字段 SNO 的最后一条数据。

---

```
SELECT FIRST(SNAME), LAST(SNO)  
FROM JBQK
```

---

**【例 5.15】** 在成绩表 CJ 中计算出总成绩。

---

```
SELECT SUM(ACHIEVEMENT) AS 总成绩  
FROM CJ
```

---

#### 4. 简单查询

所谓简单查询是指查询仅涉及数据库中的一个表。

**【例 5.16】** 显示 JBQK 表中所有字段的数据。

---

```
SELECT *  
FROM JBQK
```

---

选择表中部分字段并指定它们的显示次序,查询结果集合中字段的排列顺序与命令中所指定的字段名排列顺序相同。

**【例 5.17】** 显示 JBQK 表中指定字段 SNAME,AGE 的数据。

---

```
SELECT SNAME, AGE  
FROM JBQK
```

---

在选择列表中,可重新指定列标题。定义格式为:

---

字段名 AS 列标题

---

**【例 5.18】** 使用列标题别名。

---

```
SELECT SNAME AS 姓名, AGE AS 年龄  
FROM JBQK
```

---

**【例 5.19】** TOP *n* 指令示例。

---

```
SELECT TOP 4 SNO, SNAME FROM JBQK  
SELECT TOP 50 PERCENT * FROM JBQK
```

---

使用 TOP  $n$ [PERCENT]选项限制返回的数据行数, TOP  $n$  说明返回  $n$  行, 而 TOP  $n$  PERCENT 时, 说明  $n$  是表示一个百分数, 指定返回的行数等于总行数的百分之几。

**【例 5.20】** 查询名字的第二个汉字为“志”的学生的姓名与学号。

```
SELECT SNAME, SNO
FROM JBQK
WHERE SNAME LIKE "?志 *"
```

**【例 5.21】** 查询姓“马”的学生的姓名与学号。

```
SELECT SNAME, SNO
FROM JBQK
WHERE SNAME LIKE "马 *"
```

说明: “?”代表任意一个字符, “\*”表任意长的字符串。

**【例 5.22】** 查询 JBQK 表中全体学生情况, 查询结果按所在系的系号升序排列, 同一系的学生按年龄的降序排列。

```
SELECT *
FROM JBQK
ORDER BY(DEPT, AGE DESC)
```

**【例 5.23】** 查询 JBQK 表中“信息学院”和“新闻学院”同学的学号、姓名和年龄字段。

```
SELECT SNO, SNAME, AGE
FROM JBQK
WHERE DEPT IN ("信息学院", "新闻学院")
```

## 5. 嵌套查询

嵌套查询也称为子查询, 是指一个 SELECT...FROM...WHERE 查询可以嵌套在另外一个查询中。SQL 允许多层嵌套。每个子查询在上一级查询处理之前求解, 即嵌套查询是由里向外处理的, 这样外查询可以利用内查询的结果。

当查询涉及多个关系时用嵌套查询逐次求解, 层次分明, 易于理解, 易于书写, 具有结构化程序设计的优点。

**【例 5.24】** 在 JBQK 表中查询和李娜在同一系的学生的学号、姓名。

```
(1) 在 JBQK 表中查询李娜所在的系.
SELECT DEPT FROM JBQK WHERE SNAME="李娜"
(2) 查询和张志刚在同一系的学生的学号、姓名.
SELECT SNO, SNAME FROM JBQK
WHERE DEPT=( SELECT DEPT FROM JBQK WHERE SNAME="张志刚")
```

**【例 5.25】** 在成绩表 CJ 中查询选修了 1-01 课或选修了 1-03 课的学生学号。

```
SELECT SNO FROM CJ WHERE KNO="1-01"
UNION
SELECT SNO FROM CJ WHERE KNO="1-03"
```

## 6. 连接查询

连接查询涉及两个以上的表, 连接查询是关系数据库中最主要的查询, 包括等值连接、

自然连接、非等值连接、自身连接、外连接和复合连接查询等。

【例 5.26】 查询每个学生及其选修情况。

```
SELECT JBQK.SNO, SNAME, KNAME, CJ.ACHIEVEMENT  
FROM JBQK, CJ, KC  
WHERE JBQK.SNO=CJ.SNO AND CJ.KNO=KC.KNO
```

【例 5.27】 对 JBQK, CJ, KC 中的数据进行以下 3 种查询操作。

(1) 统计每一年龄组选修课程的学生人数。

```
SELECT AGE, COUNT(*)  
FROM JBQK, CJ  
WHERE JBQK.SNO=CJ.SNO  
GROUP BY AGE
```

(2) 求基本情况表 JBQK 中女同学的每一年龄组(超过 1 人)的人数,要求查询结果按人数的升序排列,人数相同按年龄的降序排列。

```
SELECT AGE, COUNT(*)  
FROM JBQK  
WHERE GENDER="女"  
GROUP BY AGE  
HAVING COUNT(*)>=2  
ORDER BY AGE DESC
```

(3) 检索女同学选修的课程的课程号。

```
SELECT DISTINCT KNO FROM JBQK, CJ  
WHERE JBQK.SNO=CJ.SNO AND SEX="女"
```

## 习 题

### 一、填空题

1. 查询就是依据一定的查询条件,对\_\_\_\_\_中的数据信息进行查找。
2. 聚合查询有时也称为\_\_\_\_\_,创建聚合查询有助于快速分组和汇总数据。
3. 操作查询包括更新查询、追加查询、\_\_\_\_\_和生成表查询等。
4. Access 数据库中,SQL 查询的 GROUP BY 语句用于\_\_\_\_\_。
5. 条件语句“WHERE 工资额>1000”的意思是\_\_\_\_\_。
6. 利用对话框提示用户输入查询条件,这样的查询属于\_\_\_\_\_。
7. SQL 是在数据库系统中应用广泛的数据库查询语言,它包括数据定义、数据查询、\_\_\_\_\_和\_\_\_\_\_4 种功能。
8. 使用查询向导创建交叉表查询的数据源必须来自\_\_\_\_\_个表或查询。

### 二、选择题

1. Access 数据库中,在创建交叉表查询时,用户需要指定三种字段,下面( )不是交叉表查询所需要指定的字段。

A. 格式字段      B. 列标题字段      C. 行标题字段      D. 总计类型字段

2. 将 Access 数据库中 C 语言课程不及格的学生从“学生表”中删除,要用( )。  
A. 追加查询      B. 生成表查询      C. 更新查询      D. 删除查询
3. 在 Access 中,可以把( )作为创建查询的数据源。  
A. 查询      B. 报表      C. 窗体      D. 外部数据表
4. 在 Access 数据库中,对数据表进行删除的是( )。  
A. 汇总查询      B. 操作查询      C. 选择查询      D. SQL 查询
5. 在 Access 数据库中,从数据表找到符合特定准则的数据信息的是( )。  
A. 汇总查询      B. 动作查询      C. 选择查询      D. SQL 查询
6. 若在学生表中查找所有出生日期在“2000-1-1”和“2000-12-31”之间的记录,可以在查询设计视图的准则行中输入( )。  
A. Between # 2000-1-1 # And # 2000-12-31 #  
B. Between "2000-1-1" And "2000-12-31"  
C. "2000-1-1" And "2000-12-31"  
D. # 2000-1-1 # And # 2000-12-31 #
7. 在 SQL 的 SELECT 语句中,用于实现选择运算的子句是( )。  
A. FOR      B. IF      C. WHILE      D. WHERE
8. 条件中“性别="女"AND 工资额>2000”的意思是( )。  
A. 性别为“女”并且工资额大于 2000 的记录  
B. 性别为“女”或者工资额大于 2000 的记录  
C. 性别为“女”并非工资额大于 2000 的记录  
D. 性别为“女”或者工资额大于 2000,且二者择一的记录
9. 条件“NOT 工资额>2000”的意思是( )。  
A. 除了工资额大于 2000 之外的工资额的记录  
B. 工资额大于 2000 的记录  
C. 并非工资额大于 2000 的记录  
D. 字段工资额大于 2000,且二者择一的记录
10. 已知“借阅表”中有“借阅编号”“学号”和“借阅图书编号”等字段,每个学生每借阅一本书生成一条记录,要求按学生学号统计出每个学生的借阅次数,下列 SQL 语句中,正确的是( )。  
A. SELECT 学号 COUNT(学号)FROM 借阅  
B. SELECT 学号 COUNT(学号)FROM 借阅 GROUP BY 学号  
C. SELECT 学号 SUM(学号)FROM 借阅  
D. SELECT 学号 SUM(学号)FROM 借阅 ORDER BY 学号
11. 将信息系 1999 年以前参加工作的教师的职称改为副教授,合适的查询为( )。  
A. 生成表查询      B. 更新查询      C. 删除查询      D. 追加查询
12. 以下不属于操作查询的是( )。  
A. 交叉表查询      B. 生成表查询      C. 更新查询      D. 追加查询
13. 下面对查询功能的叙述中正确的是( )。  
A. 查询和表基本类似,都可用于数据的存储

- B. 在查询中,编辑记录主要包括添加记录、修改记录、删除记录和导入导出记录  
C. 在查询中,查询不仅可以找到满足条件的记录,而且还可以在建立查询的过程中进行各种统计计算  
D. 以上说法均不对
14. 用 SQL 描述“在教师表中查找男教师的全部信息”,以下描述正确的是( )。  
A. SELECT FROM 教师表 IF(性别="男")  
B. SELECT 性别 FROM 教师表 IF(性别="男")  
C. SELECT \* FROM 教师表 WHERE(性别="男")  
D. SELECT \* FROM 性别 WHERE(性别="男")
15. ( )会在执行时弹出对话框,提示用户输入必要的信息,再按照这些信息进行查询。  
A. 选择查询      B. 参数查询      C. 交叉表查询      D. 操作查询
16. 查询能实现的功能有( )。  
A. 选择字段,选择记录,编辑记录,实现计算,建立新表,建立数据库  
B. 选择字段,选择记录,编辑记录,实现计算,建立新表,更新表  
C. 选择字段,选择记录,编辑记录,实现计算,建立新表,设置格式  
D. 选择字段,选择记录,编辑记录,实现计算,建立新表,建立基于查询的报表和窗体
17. 在 Access 数据库中使用向导创建查询,其数据可以来自( )。  
A. 多个表      B. 一个表      C. 一个表的一部分      D. 表或查询
18. 在 Access 数据库中,下列查询的计算表达式中,求两门课的平均分数,正确的是( )。  
A. [语文]+[数学]/2      B. "([语文]+[数学])/2"  
C. ([语文]+[数学])/2      D. "[语文]"+"[数学]"/2
19. 在 Access 数据库中,要查询的条件是语文成绩处在 60 分数段的记录,则在语文字段的准则中应当输入( )。  
A. >60 and <70      B. >=60 and <70  
C. >60 or <70      D. >=60 or <70
20. 在 Access 数据库中,查询姓名字段中所有姓张的同学记录时,在姓名准则中应输入( )。  
A. 张      B. 张 \*      C. \* 张      D. \* 张 \*
21. 下列 SELECT 语句正确的是( )。  
A. SELECT \* FROM "学生表" WHERE 姓名="张三"  
B. SELECT \* FROM "学生表" WHERE 姓名=张三  
C. SELECT \* FROM 学生表 WHERE 姓名="张三"  
D. SELECT \* FROM 学生表 WHERE 姓名=张三

### 三、简答题

1. 举例说明在什么情况下,需要设计生成表查询。
2. 举例说明在什么情况下,需要设计追加查询。

3. 现在已知某单位“职工工资表”中的性别用汉字“男”或“女”表示,已有字段为“姓名,性别,标准工资,加班费,标准工资”,在查询中除已知字段外,现在要求自动求出“妇女保健补贴”字段中的数据,即每位妇女月补贴 25 元,如何自动求出所有职工的工资总额?

4. 设有下列关系模型的样本数据:

书店表: 由书店号、书店名、地址组成,书店号为主码。

图书馆表: 由图书馆号、图书馆名、城市、电话组成,图书馆号为主码。

图书表: 由 ISBN 号、书名、定价组成,ISBN 号为主码。

图书发行表: 由图书馆号、图书 ISBN 号、书店号、册数组成,图书馆号、图书 ISBN 号、书店号为主码。

试用 SQL 写出以下查询。

(1) 查找馆名为“A 馆”的图书馆从书店“B 店”购买的图书书名及其册数。

(2) 取出馆址在“西安”的馆名及电话号码。

(3) 取出书店“太白路店”发行的图书书名和数量。

5. 设有一个销售管理数据库系统,其关系模式如下:

---

销售员(工号,姓名,柜台,部门)

商品(商品号,商品名,价格)

业绩(工号,商品号,销售额)

---

用 SQL 表示下列数据查询操作。

(1) 查询“服装”部门女装柜台销售员的工号和姓名。

(2) 查询销售以“霓裳”开头的服装的销售员的工号、姓名和该类产品的销售额。

(3) 查询每种商品的商品号、各销售员销售该商品的最高销售额、最低销售额和所有员工的平均销售额。

#### 四、操作题

1. 创建一个数据表结构如下: 序号(自动编号),学号(文本,8,必填,非空),姓名(文本,8),性别(文本,2),出生日期(日期),党员否(是/否),入学成绩(整型,一位小数),籍贯(文本,10),简历(备注),照片(OLE)。

字段属性定义: 定义“学号”为主键,“入学成绩”定义有效性规则为大于 500 且小于 750。

查询要求: 创建一个籍贯包含“西安”和“北京”学生的查询,显示“姓名”“籍贯”; 创建一个入学成绩大于 600 的男生的查询,查询结果按成绩降序排列,显示“学号”“入学成绩”。

2. 创建“学生成绩.accdb”数据库,数据库包括学生表(学号,姓名,系名,性别,出生日期,爱好,照片,简历)、课程表(课程编号,课程名称,学分,开课时间)和成绩表(学号,课程编号,成绩),按下列要求进行操作。

(1) 创建表间关系。

(2) 录入数据。

(3) 写出满足如下条件的 SQL 语句。

① 选出语文、数学、计算机的各科成绩在 90 分以上的学生姓名。

② 选出“计算机系”的男学生的学号和姓名。

③ 选出姓“刘”的学生的姓名、性别。

(4) 创建一个查询,查询每位学生的总分,要求输出学号、姓名、总分,查询保存为“总分查询”。

(5) 创建一个选择查询,查找并显示简历信息为空的学生的“学号”“姓名”“性别”和“出生年月”四个字段内容,所建查询命名为“基本信息查询”。

(6) 创建一个选择查询,按系别统计各自男女学生的平均年龄,显示字段标题为“所属院系”“性别”,所建查询命名为“按系统计查询”。

(7) 创建一个操作查询,将没有书法爱好的学生的“学号”“姓名”和“出生年月”三个字段内容追加到目标表“临时表”的对应字段内,所建查询命名为“追加查询”。

(8) 创建一个查询,当运行该查询时,应显示参数提示信息“请输入爱好”,输入爱好后,在简历字段中查找具有指定爱好的学生,显示“学号”“姓名”“性别”“年龄”“照片”和“简历”字段的内容,所建查询命名为“按爱好查询”。

(9) 创建一个查询,查找学生的课程成绩大于或等于 80 且小于或等于 100 的学生情况,显示“姓名”“课程名称”和“成绩”三个字段的内容,所建查询名为“成绩查询”。

(10) 创建一个查询,按“课程编号”分类统计最高分成绩与最低分成绩的差,并显示“课程名称”“最高分与最低分的差”等内容。其中,最高分与最低分的差由计算得到,所建查询名为“高低分差别查询”。

3. 创建“库存管理系统”数据库,在库中创建“产品定额储备表”和“库存情况表”。表结构要求根据自己对实际的情况理解构造。按要求创建如下查询。

(1) 以“库存管理系统”数据库中的“产品定额储备”和“库存情况”两张表为数据源创建一个查询,查找并显示库存量超过 1000 只的产品名称和库存数量,查询名为“数量查询”。

(2) 以“库存管理系统”数据库中的“产品定额储备”和“库存情况”两张表为数据源,创建一个查询,按出厂价计算每种库存产品的总金额,并显示其产品名称和总金额。总金额的计算方法为:总金额=出厂价×库存数量。查询名为“资金查询”。