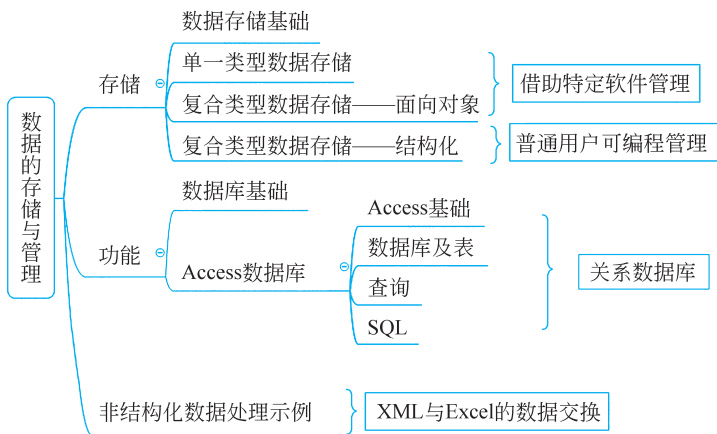


数据的存储与管理

数据存储是计算机系统中的重要组成部分,它是指将数据保存在计算机内存或外存中,以便随时访问和使用,本章的数据存储主要指外部存储,通常以文件形式存储数据,文件可以是文本、图像、音频等多种类型。

数据管理是指对数据的收集、分类、组织、编码、存储、查询和维护等活动,数据管理技术就是指与数据管理活动有关的技术。为了高效、准确地实现数据管理,用计算机进行数据管理的常用方式是通过特定软件、数据库或程序设计软件来完成。本章介绍的数据管理部分主要介绍基于关系数据库的数据管理基础。



3.1 数据的存储

第2章介绍了计算机中不同的数据类型,包括数字、字符和多媒体类型。不同数据类型的运算规则不同,数字可以进行加、减、乘、除等数学运算;字符可以进行字符串的连接、查找、替换等操作;多媒体类型中图形、图像和音频是不可分割的原子媒体类型,视频和动画通常是由多种类型的数据组成的。本节通过分析常见文件类型存储原理介绍不同类型的数据是如何存储的。

3.1.1 数据存储基础

数据存储就是将信息以各种不同的形式存储在外存的过程。数据通常以文件

的形式存储在外存设备中。文件存储时,为了读取和处理方便,需要对数据进行合理编排,通常会对数据进行不同层次的组织。

1. 基本概念

数据是对现实世界的抽象,数据用人为规定的符号表示从现实世界中观察、收集到的现象和事实。

数据对象是对一个实体的具体描述,如一个人的个人信息、一张照片等。

当一个数据对象中包含多个相同类型的数据单元时,通常把一个数据单元称为一个**数据元素**。

数据项是指具有独立含义的最小的数据存储单位,如果进一步分割则会失去数据本质的含义,如地名广州代表了一个城市,如果再进一步拆解为广,则与这地方本身代表的对象没有实际的关系。

数据的**逻辑结构**是从逻辑的角度(即数据间的联系和组织方式)观察数据、分析数据,与数据的存储位置无关。其是指数据使用者所看到的数据组织方式,在数据处理时按照逻辑结构进行数据访问。

数据的**存储结构**,也称物理结构,是指数据的逻辑结构在计算机中的实现形式,即数据在物理存储空间上的存放方式(连续存放还是分散存放)。数据的物理结构是从数据的管理者的角度研究数据项之间的关联。本书只讨论数据的逻辑结构,物理结构不展开讲述。

2. 数据类型汇总

从计算机的角度,数据泛指可以被计算机接收并能被计算机处理的符号。如 2.1 节~2.3 节介绍,计算机中数据的表现形式包括文本形式(数字、文字)和多媒体形式(图形、图像、动画、声音)。

根据文件存储内容可将文件分为单一类型和复合类型。

(1) **单一类型**: 指在文件内部只存储一种数据信息,如文本文件中只存储字符。单一类型的文件还包括图像、声音等。

(2) **复合类型——面向对象**: 指文件内部保存了不同类型的数据,如视频、Word 或 HTML 文档等。

(3) **复合类型——结构化**: 主要指基于关系数据库的数据存储。

数据分类如表 3.1 所示。

3.1.2 单一类型数据存储

计算机中基本的数据类型包括数字、文字、声音和图像。单一类型数据是指数据中只存在一种数据类型格式,并且数据之间没有明确的语义关系,如文本文件里面只有字符,如图 3.1 中展示了一个文本文件 jnu.txt。这类单一类型数据通常由最本原的手段获取,并且没有与其他类型的数据共同存储在一个文件中,如图像通常由照相机获取,声音由录音机获得等,文本文件由用户直接输入字符等。

表 3.1 数据分类

类 别	类型	特 点
单一类型	文本	只存储字符,如 TXT 等
	声音	存储的是声波文件,如 WAV、MP3 等
	图像	用数字任意描述像素点、强度和颜色,如 JPG、PNG、BMP 等
复合类型 ——面向对象	图形	由外部轮廓线条构成的矢量图,即由计算机绘制的直线、圆、矩形、曲线、图表等。例如,SVG、CDR 等
	标记文件	由标记和数据构成,如 HTML、XML 等
	视频	存储的内容包括声音、大量的图、文字等,如 MP4、AVI 等
	动画	存储的内容包括声音、图形、图像、文字等,如 SWF、FLC 等
	其他	包含不同的数据对象类型,如 Word、Excel 等文档
复合类型 ——结构化	数据库	存储一条条记录,如 mdb、dbf 等

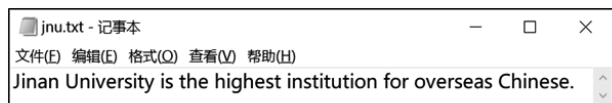


图 3.1 文本文件示例

图 3.1 jnu.txt 文本文件中的内容全部是英文字符,英文字符在存储时存储每个字符对应的 ASCII 码。图 3.2 展示的是文本文件的存储,每行存储 16 个字符,每行前面的数字代表字符位置(十六进制形式)。从图 3.2 中可以看到第一个字符存储的数字是 0x4a,即一个十六进制数 4a,其对应的十进制数是 74,也就是字母 J 对应的 ASCII 码(大写字母 A 对应的 ASCII 码是 65),以此类推,可以发现此文本文件包含的是一个连续存放的字符。

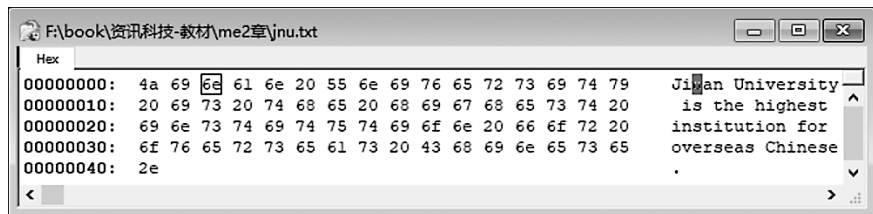


图 3.2 文本文件存储内容示例(一)

对于此类文件进行存储时,需要逐个读取其中的数据,再进行相应的数据计算,例如,如果要将图 3.1 中的 Jinan 修改为 JiNan,就需要修改对应的 ASCII 码数值为 78(大写字母 N 的 ASCII 码),其对应的十六进制数为 4e,修改后的效果如图 3.3 所示。

单一类型数据文件通常由专门的应用软件来完成,如 mspaint 进行修图的基本原理就是对相应的像素点进行颜色设置,通常选择一片区域或者相同数值的颜色值进行批量设置。

3.1.3 复合类型数据存储——面向对象

复合类型数据存储是指在同一个文件内部包含了不同种类的数据,如视频、动画、Word

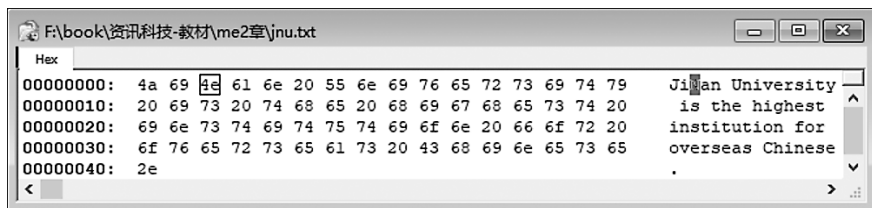


图 3.3 文本文件存储内容示例(二)

或 Excel 文档、数据库类型等。其中,视频、动画这样的类型没有固定的内部格式,将其归类为面向对象的复合数据类型;数据库中每个数据库文件通常有固定的结构,将其归为结构化的复合数据类型。

1. 视频文件

尝试通过一个视频处理软件 Camtasia 打开视频文件的原始素材,界面如图 3.4 所示,下方显示该视频文件中的 3 个轨道,分别是图像、音频和字幕。当 Camtasia 软件将这些素材组合成一个视频文件时,就需要将 3 个轨道分别作为一个数据对象进行进一步封装处理,其过程非常复杂(处理过程比较耗时,通常视频处理软件也非常庞大),在此不过多陈述。

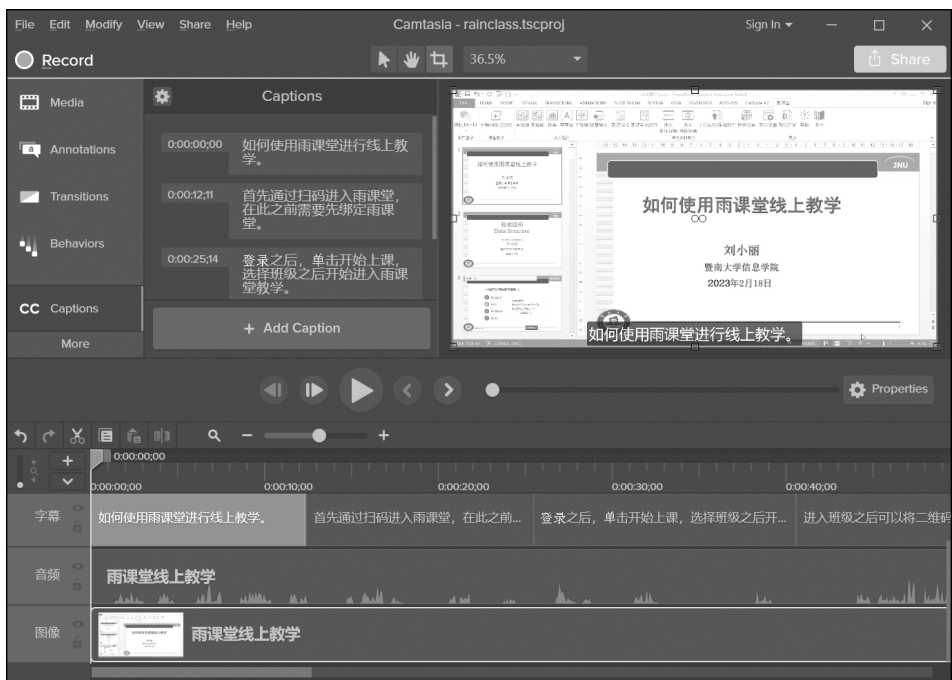


图 3.4 视频编辑界面

动画文件与视频文件比较类似,所不同的是视频文件中的动态画面是由一幅幅图片连续播放形成的,而动画中的动态画面可能是若干对象实体的形状/位置变化所得到的,所以相对来说视频文件比动画文件存储的图片要多、视频文件通常比较大。

2. 标记语言文件

标记语言文件是指文件内容包含了标记信息,其对应的处理软件会根据不同的标记展示不同的内容。简单地说,标记语言文件是使用标记语言来描述的文件。通常,标记语言文件可以将不同类型的数据进行标记编排,形成指定的格式文件。

名词释义: 标记语言(Markup Language)

标记语言是一种将文本以及文本相关的其他信息结合,展现出关于文档结构和数据处理细节的文字编码。与文本相关的其他信息(包括文本的结构和表示信息等)与原来的文本结合,使用标记进行标识。

常见的标记语言格式有 HTML(超文本标记语言)、XML(可扩展标记语言)、SGML(标准通用标记语言)、XAML(可扩展应用程序标记语言)等。HTML 文本是由 HTML 命令组成的描述性文本,HTML 中的标签可以将网络上的文档格式统一,使分散的 Internet 资源链接为一个逻辑整体,HTML 命令可以说明文字、图形、动画、声音、表格、链接等。如图 3.5(a)是一个 HTML 文件,该文件将文字、图片进行组合,用浏览器解析打开之后的界面如图 3.5(b)所示。



(a) HTML 文本



(b) 对应的网页

图 3.5 HTML 文件与对应的网页

HTML 文件中的每个数据元素都需要进行标记,用<标签>数据元素</标签>来标记,例如,HTML 文件 jnu.html 中包含了文本和图片两类数据。

(1) <title>暨南大学简介</title>: 标记该文件的标题。

(2) <style type="text/css"> </style>: 定义该文件用到的格式。

(3) <p align="center"></p>: 标记插入图片的对齐方式、名称、宽度和高度。通过这些描述就可以将图像文件作为段落插入页面中。

HTML 文件可以将文本、图片、声音、视频、动画等数据对象进行标记,在网页上展示不同的对象类型数据。XML 是用于标记电子文件使其具有结构性的标记语言,可以用来标

记数据、定义数据类型,它允许用户定义自己的标签和自己的文档结构。XML 是 SGML 的子集,非常适合 Web 传输。

综上所述,标记语言文件就像一个说明书,把不同类型的素材黏合在一起,但是每种素材都是依据素材本身的文件特点进行存储的。

3. Word 文件

Word 文件也可以包含文本、图像、动画、表格等,还可以对数据对象进行非常丰富的格式设置,但 Word 文件的存储与文本文件有很大的不同,Word 文件其实是一个 Zip 压缩文件,可以用 Zip 软件进行解压。

如图 3.6 对 jnu.docx 文件进行解压,上半部分是解压后的内容,下半部分是 Word 文件打开之后的内容,Word 文件对应的压缩包主要由 XML 文件构成,分为 4 部分。

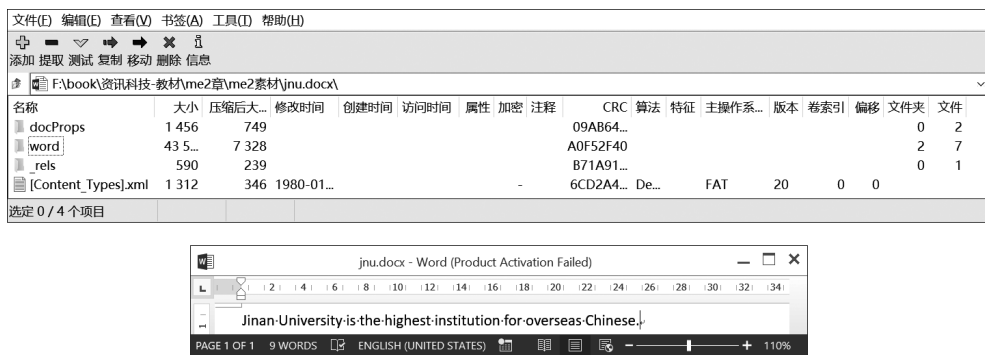


图 3.6 Word 文件与对应的数据元素

(1) docProps: 目录中的 XML 文件保存了 Word 文件的属性,其中包括 core.xml、app.xml 文件。

(2) word: 目录中包含了文件的字体、格式、风格、Web 设置等。

(3) _rels: 目录中会有一个扩展名为 rels 的文件,它里面保存了这个目录下各个 Part 之间的关系。

(4) [Content_Types].xml: 目录文件,记录每个 XML 文件的内容类型。

扩展名为 pptx、xlsx 文件的存储原理与 Word 类似。无论是多媒体类型文件、标记语言文件,还是 Word 文件,处理对象都是没有固定结构的数据,称为面向对象的数据存储,其操作实现比较复杂。

单一类型数据存储和面向对象的复合类型数据存储都归类为非结构化数据。

3.1.4 复合类型数据存储——结构化

普通程序员要想处理、存储复合类型数据,使用面向对象的形式,难度非常大,几乎是不可实现的。计算机科学家提出了一种结构化的数据存储方式,用于存储具有相同数据元素的批量数据。

1. 结构化数据定义

简单地说,结构化数据对象中的数据通常具有相同的格式,这些数据需要依据不同的数

据类型进行快速计算、修改。结构化的数据是指可以使用关系数据库表示和存储,由二维表结构进行逻辑表达和实现,严格地遵循数据格式与长度规范,主要通过关系数据库进行存储和管理。其一般特点:数据以行为单位,一行数据表示一个实体的信息,每列数据的属性相同。

结构化数据是高度组织和格式化的数据,与非结构化数据相比,是更容易使用的数据类型。存储和管理这些数据一般为关系数据库,当使用结构化查询语言(SQL)时,计算机程序很容易搜索这些数据。结构化数据具有的明确关系使这些数据运用起来十分方便。

2. 结构化数据举例

图 3.7 是一个关系数据库的基本表,首行是字段名称(也称属性),每行代表一条记录,即每个交易单的基本数据。人们可以很方便地进行查询或者更新,例如,如果想查找 2019 年 12 月份总消费情况,就可以直接筛选时间和使用类别,将使用金额相加即可。

交易号	卡号	时间	使用类别	使用金额	余额
201528	001	2019-10-28 19:52:26	消费	¥10.00	¥483.00
201529	001	2019-10-28 19:54:05	充值	¥20.00	¥503.00
201530	001	2019-10-28 19:54:09	消费	¥2.00	¥501.00
201531	001	2019-12-30 19:49:29	充值	¥200.00	¥701.00
201532	001	2019-12-30 19:49:34	消费	¥18.00	¥683.00
201533	001	2019-12-30 19:51:33	消费	¥20.00	¥663.00
201534	001	2019-12-30 21:28:21	充值	¥200.00	¥863.00
201535	001	2019-12-30 21:28:26	消费	¥19.00	¥844.00
201536	001	2019-12-30 21:28:30	消费	¥25.00	¥819.00
	001	2020-04-11 0:47:19			

图 3.7 结构化数据

对于图 3.7 中的数据,依然可以存储在 Word 文件中,但是已经失去了结构化的含义。

非结构化数据存储和传输比较方便,但是缺点十分明显:修改或者统计的难度非常大,尤其是当行数较多时,几乎无法实现。所以结构化的数据存储加快了处理数据的速度和效率,在某种程度上,结构化数据的存在,也是实现自动计算的第一步。

3.2 数据库基础

3.2.1 数据库的由来

从数据管理的角度看,数据库技术到目前共经历了如下 3 个阶段(见图 3.8)。

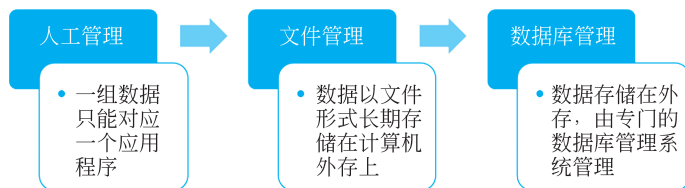


图 3.8 数据库技术的 3 个阶段

(1) 人工管理阶段:数据量小、独立,用户直接管理。数据和应用程序是严格绑定的,

一组数据只能对应一个应用程序,如果需要改变数据的结构,就要对应用程序作相应的更改,使用起来很不方便。

(2) 文件管理阶段:使用文件存取数据,冗余度高,管理维护难。数据以数据文件形式存储,数据文件是记录的集合,每条记录数据由一些字段按照特定结构组成,应用程序对数据的存取由文件管理系统完成。数据和应用程序具备了一定的独立性。实现了数据的共享和长期存储。

(3) 数据库管理阶段:专门的数据库系统管理数据,高效方便,易于共享维护。数据库系统负责管理数据及数据之间的关联,数据库系统的数据由数据库管理系统(Database Management System, DBMS)统一管理,允许多个用户或多个应用程序同时访问数据库。用户可以使用终端命令(如 SQL)对数据库进行访问,也可以借助高级语句(如 Java、C 等程序)对数据库进行访问。DBMS 还提供了数据的安全性、完整性和并发控制等功能。

当面对数据量较大、数据间关系比较复杂等情况时,数据的存储结构、共享性、安全性、独立性等问题会越来越突出,一般会采用数据库系统进行数据管理,数据库技术正是研究解决这些问题的专门技术。

数据库管理技术能够很好地对大量数据进行存储、管理及高效检索,而且数据库中的数据可以被多个用户、多个应用共享使用。数据的共享可以尽可能地避免数据的重复问题,节约存储空间,同时也能够减少由于数据重复造成的数据不一致现象。

3.2.2 数据库相关概念

数据库是存储在计算机内、有组织的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和存储,具有较小的数据冗余度,较高的数据独立性和可扩展性,并且数据库中的数据可为各种合法用户共享。简单来说,数据是描述事物的符号记录,模型是现实世界的抽象。数据库系统中用以提供信息表示和操作手段的形式构架。

1. 数据库管理系统

数据库管理系统是一个软件系统,主要用来定义和管理数据库,处理数据库与应用程序之间的联系。它建立在操作系统之上,对数据库进行统一的管理和控制。数据库管理系统的主要功能如下。

(1) **描述数据库**。数据库管理系统能够提供数据描述语言,描述数据库的逻辑结构、存储结构和保密要求等,使人们能够方便地建立数据库和定义数据库的结构。

(2) **操作数据库**。数据库管理系统能够提供数据操纵语言(Data Manipulation Language, DML),使用数据操纵语言能够方便地对数据库进行查询、插入、删除和修改等操作。

(3) **管理数据库**。数据库管理系统能够提供对数据库的运行和管理功能,保证数据的安全性和完整性,控制并发用户对数据库数据的访问,管理大量数据的存储。

(4) **维护数据库**。数据库管理系统能够提供数据维护功能,管理数据初始导入、数据转换、备份、故障恢复和性能监视等。

2. 数据模型

数据模型是对数据的特点及数据之间关系的一种抽象表示,是数据特征的抽象,是数据库管理的形式。现有的数据库系统都是基于某种数据模型的。

数据模型包括数据结构、数据操作和完整性约束 3 部分:数据结构用于描述系统的静态特性,是所研究的对象类型的集合。数据操作是对系统动态特性的描述,是指对数据库中各种对象允许执行的操作的集合,包括对数据库的检索和更新两大类操作。完整性约束是对数据模型中的数据的约束规则,以保证数据的正确、有效、相容。其中,数据结构是刻画一个数据模型性质最重要的方面。实际上,数据库系统中是按照数据结构的类型命名数据模型。

根据数据结构的组织方式可将数据模型分为 3 种。

(1) 层次模型是数据库系统最早使用的一种模型,它的数据结构是一棵“有向树”,便于表示一对多的关系。

(2) 网状模型以网状结构表示实体与实体之间的联系,可以表示多个从属关系的联系,也可以表示数据间的横向关系与纵向关系,它是层次模型的扩展。网状模型可以方便地表示多对多的关系,但结构复杂,实现的算法难以规范化。

(3) 关系模型以二维表结构来表示实体与实体之间的联系,它是以关系数学理论为基础的。关系模型的数据结构是一个“二维表框架”组成的集合,在关系模型中,操作的对象和结果都是二维表。关系模型是目前最流行的数据库模型,Access 就是一种关系数据库系统。

目前,主要的数据库系统都是基于关系模型的关系数据库系统。直观地说,关系模型中数据的逻辑结构如表 3.2 所示,这种表结构称为二维表。

表 3.2 学生二维表

学 号	姓 名	性 别	籍 贯	出生日期	入学总分	爱好特长
2018050001	刘伟箭	男	广东广州	08/10/2000	692	足球,篮球
2018050002	刘简捷	男	北京	12/08/1998	701	书法,国画
2018050003	杨行东	男	湖南长沙	06/12/1999	695	球类,游泳
2018050004	林慧繁	女	广西桂林	01/15/2000	703	舞蹈,唱歌

关系模型的主要优点包括以下 4 项。

- (1) 关系模型建立在严格的数学概念的基础上,有坚实的理论支持。
- (2) 数据结构简单、清晰,易于操作和管理。
- (3) 具有较高的数据独立性,有利于系统的扩充和维护。
- (4) 具有能够处理复杂关系对象的能力。

3. 关系数据库

客观存在并且可以相互区别的事物被称为实体,实体是彼此可以识别的对象,实体通过其特征相互区别。表征实体特征称为属性。属性是事物性质的抽象,实体及其属性构成信

息世界。

关系数据库是应用二维表格来表示和处理信息世界中的实体集合和属性关系的数据库。关系数据库不是按物理的存储方式来组织连接数据,而是通过建立表与表之间的关系来连接数据库中的数据。

关系数据库中数据的基本结构是表(Table),即数据按行、列有规则的排列、组织。数据库中每个表都有一个唯一的表名。

1) 关系模型的基本概念

关系: 一个关系在逻辑上对应一个按行、列排列的表。

属性: 表中的一列称为一个属性,或称一个字段(Field),表示所描述的对象的一个具体特征。

域: 属性的取值范围。例如,性别属性取值范围是“男”或“女”。

元组: 表中的一行称为元组,又称记录(Record)。

主键: 在一个表中不允许有两个完全相同的元组,表中能唯一标识元组的一个属性或属性集合称为主键(Key)。例如,学生表的主键为“学号”。

关系模式: 关系名及关系中的属性集合构成关系模式。

例如,学生表的关系模式如下:

学生表(学号,姓名,性别,籍贯,出生日期,入学总分,爱好特长)

2) 关系模型的完整性约束

关系数据库的重要特征之一是数据的完整性约束。数据的完整性是指数据的正确性和一致性。数据库管理系统提供了定义数据完整性约束条件的机制,能够检查数据是否满足完整性约束条件,从而防止数据库中存在不合语义的数据,防止由于错误的输入输出而造成错误的结果。

关系模型中的完整性约束条件主要包括实体完整性和参照完整性。还有一种约束是用户自定义完整性约束,它是在开发应用系统时,根据使用者的业务规则定义的完整性规则。

实体完整性是指构成关系主键的属性或属性集合不能为空。在关系模型中,系统要求不能有完全相同的两个元组。关系中各个元组的唯一性是通过主键保证的。如果主键为空,无法保证关系中不出现重复的元组。例如,学生表的主键是“学号”,如果存在姓名等其他各属性完全相同的学生,且“学号”属性为空,则无法保证元组的唯一性。

参照完整性用于约束多个表之间的数据一致性。例如,在课程管理中除学生表外,还要有学生的成绩表,用于记录学生所选修的各门课程的成绩。学生成绩表的主键由课程号和学号两个属性构成。若只有课程号属性,学号属性为空,则无法确定是哪个学生的成绩;若只有学号属性,课程号属性为空,则不能确定学生的成绩是哪门课程。

学号是学生表的主键,不是成绩表的主键,这类属性对于成绩表的关系称为外键。参照完整性要求外键的取值只允许有两种可能:一种是空值;另一种是等于与之相关表的主键值。这种约束表明,在进行课程管理时,学生成绩表中的学号必须存在学生表中,也就是只有已经注册的学生才能选课,才能有课程的成绩。如果学生表中的学号发生变化,学生成绩表中的学号也必须做相应的修改,这就是参照完整性的实例。参照完整性主要用于维护不同表之间数据的一致性。

实体完整性和参照完整性一般由数据库管理系统提供。在设计及定义系统时,指定有

关的主键和外键等,系统会自动进行有关的完整性检查。

用户定义的完整性规则是针对一个具体的应用提出的完整性约束要求。这类规则是根据具体应用要求,在需求分析和系统设计阶段制定的。用户定义的完整性规则既可以在定义数据表的同时进行,也可以在系统开发时通过程序实现。

数据库相关概念

- **数据**: 数据就是数据库中的基本存储对象。
- **数据库**: 长期存储在计算机系统中,有组织的可共享的大量数据的集合。
- **数据库管理系统**: 位于用户应用和操作系统之间的一层数据管理软件。用于科学存储和组织数据,高效获取和维护数据,包括数据定义,数据组织、存储和管理,数据操纵功能,数据库的事务管理和运行管理,数据库的建立和维护等。
- **数据库系统**: 数据库系统是指在计算机系统中引入数据库后构成的系统,通常包括数据库、数据库管理系统、应用程序、数据库管理员。

3.2.3 数据库的发展阶段

按照数据模型发展的主线,数据库技术的形成过程和发展可分为如下3个阶段:层次和网状数据库系统阶段、关系数据库系统阶段和新一代数据库系统阶段。

1. 层次和网状数据库系统

比较典型的是由IBM公司研制的层次数据库系统(Information Management System, IMS)和美国数据库系统语言研究会的网状数据库系统(Data Base Task Group, DBTG)。这一阶段数据库中数据之间的联系在层次和网状数据库系统中都是用存取路径来表示和实现的,拥有独立的数据定义语言和导航的数据操纵语言。

2. 关系数据库系统

支持关系模型的关系数据库系统是第二代数据库系统。1970年,IBM公司的研究员提出了数据库的关系模型,开创了数据库关系方法和关系数据理论的研究,促进了数据库的产业化。第二代关系数据库系统具有模型简单清晰、理论基础好、数据独立性强、数据库语言非过程化和标准化等特色,SQL为关系数据库语言标准。关系数据库系统可以理解为使用二维表来表示数据间的关系。

3. 新一代数据库系统

第三代数据库系统应支持数据管理、对象管理和知识管理。第三代数据库系统继承了第二代数据库系统已有的技术;保持第二代数据库系统的非过程化数据存取方式和数据独立性,这不仅能很好的支持对象管理和规则管理,而且能更好地支持原有的数据管理,支持多数用户需要的查询等。第三代数据库系统支持数据库语言标准;在网络上支持标准网络协议;具有良好的可移植性、可连接性、可扩展性和可互操作性;等等。新一代数据库系统改进了关系数据库系统存在数据模型、性能、扩展性、伸缩性等方面的缺点,出现了面向对象数据库(Object Oriented Database, OODB)、非关系数据库(Not Only SQL, NoSQL)等。

3.3 Access 数据库

3.3.1 Access 基础

Access 的主要功能如下。

- (1) 定义表,利用表存储相应的信息。
- (2) 定义表之间的关系,利用表中相关的字段链接不同的表。
- (3) 强大的数据处理能力,使用 Access 可以创建查询检索数据;创建窗体查看、输入及更改表中的数据;创建报表分析数据或将数据以特定的方式打印出来。
- (4) 导入、链接或导出 HTML 文档。通过网络发布和访问数据库的信息,建立对 Internet 的支持。
- (5) 开发应用程序。用户可以利用宏或 Visual Basic 将各种数据库对象链接在一起,形成一个数据库应用系统。

Access 提供了创建数据库、表、查询、窗体和报表等数据库对象的向导,用户可以利用可视化的工具创建和编辑各种数据库对象,可以不编写任何代码就创建一个完善的应用程序。

Access 数据库中包括表、查询、窗体、报表、宏和模块等不同的对象,这些对象用于收集、存储、检查和链接各种不同的信息。在 Access 中,一个数据库包含数据和与存储数据有关的所有对象。

1) 表

表(Table)是一种有关特定实体的数据的集合,表以行(Record,即记录)、列(Field,即字段)格式组织数据。表中的字段用来描述实体的属性,记录用来描述一个实体的完整信息。通常,一个数据库保存的主要信息,都是以表的形式表示和存储的。一个数据库可以同时有多个表。Access 允许用户在不同表之间定义关系。

2) 查询

查询(Query)是 Access 的重要组成部分。它是对数据库中数据的直接访问。利用查询可以通过不同的方式查看、更改、分析以及控制数据库中的数据,也可以为其他查询、窗体和报表提供数据,还可以为一个数据访问页提供数据源。

在 Access 中可以利用 QBE(Query By Example)工具创建查询,也可以通过 SQL 语句建立查询。

3) 窗体

窗体(Form)也是 Access 的重要组成部分。它是用户交互式访问数据库的界面,是数据库与用户之间的主要接口。使用窗体可以向表中输入数据,查看或更新表中的数据,以及根据用户的输入信息执行相应的操作。Access 还提供了打印窗体的功能。窗体使数据的输入、修改和查询变得非常容易和直观。

4) 报表

报表(Report)是输出数据库数据最有效的方法。在报表中可以控制每个数据的显示方式,对数据进行排序和分组,并给出每组记录的各种统计数据。在用 Access 打印报表前,用

户可以在屏幕上查看报表格式。报表中的数据可以来源于表或查询。

Access 提供了强大的报表设计工具,使用户能够在较短时间内设计出高质量报表。

5) 宏

宏是一个或多个操作的集合,其中每个操作可实现特定的功能。例如,设置一个宏,在用户单击“打开”按钮时运行该宏,以打开一个表。

在数据库的很多地方要用到宏,尤其是在窗体设计中。使用宏可以让用户非常方便地处理一些重复性操作。通过使用宏可以实现一些操作,如打开表、更改记录、插入记录、删除记录、建立查询、生成报表和打印数据等。

6) 模块

Access 支持 VBA(Visual Basic for Application)语言。模块是将 VBA 声明和过程作为一个单元进行保存的集合,专门用来存放 VBA 代码。模块有两个基本类型,即类模块和标准模块。一个模块一般包含多个过程(Procedure)或函数(Function)。

Access 数据库以一个文件存储数据库中的所有对象,其扩展名为 accdb。Access 窗口如图 3.9 所示,其中快速访问工具栏、标题栏、功能区、选项卡等的使用方法与 Excel 类似。位于窗口左下端的导航窗格是 Access 窗口的重要元素,数据库中的所有对象的名称在此罗列。打开数据库对象的方法:在导航窗格中双击对象名称;或在导航窗格中选择对象,然后按 Enter 键;或在导航窗格中右击对象名称,在弹出的快捷菜单中选择“打开”命令,快捷菜单中的命令根据不同的对象类型而有所不同。

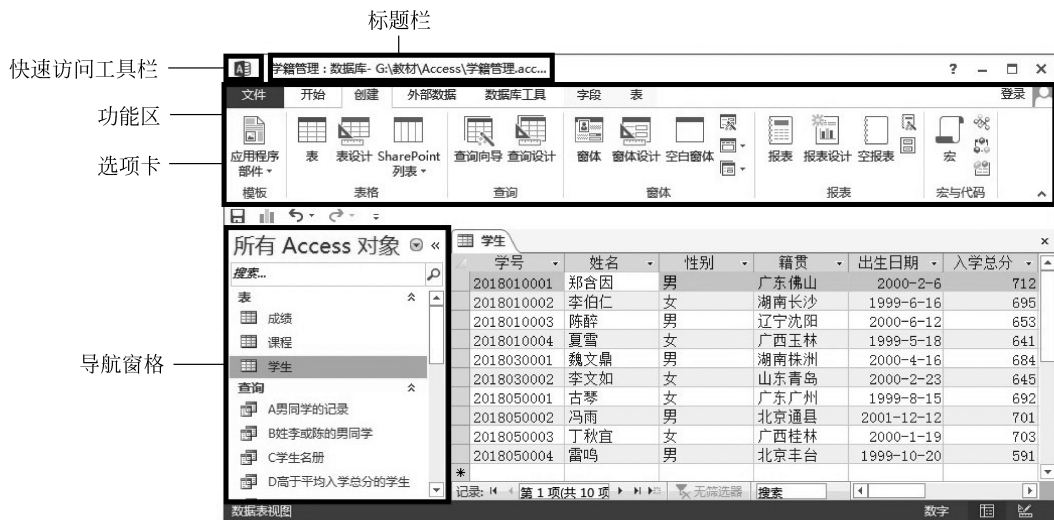


图 3.9 Access 窗口

3.3.2 数据库及表

Access 数据库以表的形式组织数据,一个数据库就是多个表的集合,表之间存在着引用和被引用的关系。此外,为了数据处理的需要,在数据库中还可以创建查询、窗体、报表、模块等多种对象,所有这些对象都存储在数据库文件(.accdb)中。因此,Access 数据库是相关联数据及相关对象的集合。其中,表对象是最核心的数据库对象。

Access 数据库表是一种结构化的二维表,结构化是指表的同一列数据有相同的字段名称、相同的数据类型、相同的数据存储宽度等。每列称为一个字段,字段的结构化是由字段属性描述的。要创建一个 Access 表,先要创建表结构,即通过 Access 的表设计器设计表中的每个字段及相关属性;再向表中添加数据,即数据是在结构化的框架下输入表中的。

本节将创建一个名为学籍管理的数据库。在该数据库中,共有 3 个表,分别用来存储学生、课程和成绩的相关信息。表中的字段和数据类型分别如表 3.3~表 3.5 所示。

表 3.3 学生表中的字段和数据类型

字段名称	字段数据类型	字段大小
学号	短文本	10 个字符
姓名	短文本	4 个字符
性别	短文本	1 个字符
籍贯	短文本	10 个字符
出生日期	日期/时间	
入学总分	数字	整型
住校否	是/否	
爱好特长	短文本	20 个字符
简历	长文本	

表 3.4 课程表中的字段和数据类型

字段名称	字段数据类型	字段大小
课程编号	短文本	6 个字符
课程名称	短文本	20 个字符
课程类别	短文本	2 个字符
学分	数字	整型

表 3.5 成绩表中的字段和数据类型

字段名称	字段数据类型	字段大小
学号	短文本	10 个字符
课程编号	短文本	6 个字符
修读学期	数字	整型
成绩	数字	整型

1. 创建数据库

在 Access 中创建一个数据库有两种方法。

(1) 先创建一个空数据库,再向数据库中添加表、查询、窗体和报表等数据库对象。这种方法比较灵活,但要求对数据库中的每个对象进行定义。

(2) 使用 Access 提供的数据库向导,在向导的提示步骤下进行操作,就可以很快创建一个数据库,并在数据库中建立所需的表、窗体和报表等数据库对象。数据库向导提供了多个数据库的模板供选择。使用数据库向导创建数据库的方法比较简单。

2. 在数据库中添加表

表是特定实体的数据集合,是数据库的基本对象,用来存储和管理数据。它是数据库的基础。表以行、列方式组织数据,表中的一行称为记录,描述一个实体的完整信息。表中的一列称为字段,描述实体的一个属性。例如,本节中要创建的学生表(见表 3.3),其中一个学生的所有信息(如学号、姓名、性别、籍贯、出生日期、入学总分、住校否、爱好特长、简历)称为一条记录;所有学生的同一属性(如学号)称为字段。

在数据库中表不是孤立存在的,相互之间有一定的关系。不同的表中可以有相同的字段,两个表通过表中相同的字段关联,建立表之间的关系。

在创建数据库后的首要任务是定义表结构以及表与表之间的关系,然后向表中输入数据,并创建其他数据库对象,以实现数据库系统的功能。在 Access 数据库中,系统为表对象提供了两种工作视图,即设计视图和数据表视图。使用设计视图可以创建及修改表的结构,如图 3.10 所示;使用数据表视图可以查看、添加、删除及编辑表中的数据。

学生	
字段名称	数据类型
学号	短文本
姓名	短文本
性别	短文本
籍贯	短文本
出生日期	日期/时间
入学总分	数字
住校否	是/否
爱好特长	短文本
简历	长文本

图 3.10 设计视图

3. 设置字段数据类型及属性

Access 数据库中提供的字段数据类型如表 3.6 所示。

表 3.6 字段数据类型

数据类型	用法	存储空间
短文本	文本或文本与数字的组合,如姓名等;也可以是不需要计算的数字,如邮政编码	最多可用 255 个字符
长文本	长文本及数字,如备注	最多为 65 535 个字符
数字	用于数学计算的数值型数据	1 字节、2 字节、4 字节或 8 字节
日期/时间	100—9999 年的日期和时间值	8 字节

续表

数据类型	用法	存储空间
货币	货币值或用于数学计算的数值型数据。使用货币数据类型可以精确到小数点左边 15 位和小数点右边 4 位	8 字节
自动编号	在添加记录时自动插入一个唯一的顺序号(每次递增 1)或随机数	4 字节
是/否	字段只包含两个值中的一个,如是/否、真/假及开/关	1 字节
OLE 对象	表中链接或嵌入的对象,如 Word 文档、Excel 电子表格、图像、声音或其他二进制数据	
超链接	文本或文本和数字的组合,以文本形式存储并用作超链接地址	超链接地址中每部分最多只能包含 2048 个字符
查阅向导 (Lookup Wizard)	创建字段,该字段可以使用列表框或组合框从另一个表或取值列表中选择一个值。在数据类型列表中选择此选项,将启动查阅向导进行定义	与用于执行查阅向导的主键字段大小相同,通常为 4 字节

在 Access 的数据类型中,默认值是短文本。在使用 Access 的数据类型时,需要注意下面 3 个问题。

(1) 长文本、超链接和 OLE 对象字段不能进行索引。

(2) 对数字、日期/时间、货币以及是/否等数据类型,Access 提供了预定义显示格式。用户可以设置格式属性选择所需的格式。

(3) 如果表中已经输入数据,在更改字段的数据类型时,修改后的数据类型与修改前的数据类型发生冲突,则有可能丢失一些数据。

4. 主关键字

主关键字(也称主键)是表中的一个或一组字段,这些字段的值能唯一标识表中所保存的每条记录。指定了表的主关键字后,为确保唯一性,当用户输入新记录到表中时,Access 将检查是否有重复的数据,如果有则禁止把重复的数据输入表中。同时,Access 也不允许在主关键字字段中输入 Null(空值)。

在设置主关键字时,应选择没有重复值的字段作为主关键字。如果选择的字段中有重复值或 Null,Access 将不会将其设置为主键。

如果在创建新表的过程中没有设置主键,保存时 Access 将询问是否要创建主键。如果回答是,Access 将创建一个“自动编号(ID)”字段作为主键。该字段在添加记录时自动插入一个具有唯一顺序的数字。

如果表中存在值唯一的一个或一组字段,用户可以自行创建主键。例如,“学生”表中的“学号”字段值就是唯一的,可以将该字段指定为主键。

5. 索引

索引是对数据库的虚拟排序,它并不影响数据在表中的位置和顺序。建立索引的目的是加快查询速度。当经常要对某个字段进行搜索或排序时,可以创建此字段的索引;也可以

根据需要,在一个表中创建多个索引。表在更改或添加记录时,索引可以自动更新。任何时候都可以在设计视图中为表添加或删除索引。

在表中使用多个字段索引进行排序时,首先按索引中的第一个字段进行排序,如果第一个字段有重复值,系统再按第二个字段进行排序,以此类推。

6. 表与表之间的关系

在数据库中,表的内部和表之间均存在联系。表内部的联系是指组成表的各个字段之间的关系。表之间的联系是指不同表之间的关系。两个表之间的关系有3种类型:一对一、一对多和多对多。

(1) 设有A、B两个表。在一对一关系中,A表中的每条记录仅能与B表中的一条记录匹配,并且B表中的每条记录仅能与A表中的一条记录匹配。

例如,在学校中,每个系只有一名系主任,每个系主任只领导一个系,则系与系主任之间具有一对一关系。

(2) 一对多关系是关系中最常用的类型。在一对多关系中,A表中的一条记录能与B表中的多条记录匹配,但是B表中的一条记录仅能与A表中的一条记录匹配。

例如,在学校中,一个班级有若干学生,每个学生只属于一个班级,则班级与学生之间具有一对多关系。

(3) 在多对多关系中,A表中的一条记录能与B表中的多条记录匹配,并且B表中的一条记录也能与A表中的多条记录匹配。

例如,一门课程可以同时有多个学生选修,一个学生也可以同时选修多门课程,则课程与学生之间具有多对多的关系。

由于现在的数据库管理系统不直接支持多对多关系,在处理多对多的关系时需要将其转换为一对多关系。一个多对多的关系实际蕴涵了两个一对多关系。两个表之间的多对多关系可以通过如下方法实现:定义第三个表,将其他两个表中定义为主键的字段添加到第三个表中,则这两个表与第三个表之间均是一对多的关系。多对多关系实际上是使用第三个表的两个一对多关系。

在学籍管理数据库中,学生表和课程表之间的关系是多对多关系,一门课程可以同时有多个学生选修,一个学生也可以同时选修多门课程。为处理这两个表之间的联系,创建第三个表,名为成绩,将学生表和课程表中的主关键字字段“学号”和“课程编号”作为成绩表的字段,成绩表的第三个字段是“成绩”,记录每个学生选修每门课程的成绩。创建成绩表后,则学生表和成绩表之间是一对多关系,即一个学生可以有多门课程的成绩;课程表和成绩表之间也是一对多的关系,一门课程可以同时有多个学生选修。

若两个表之间存在一对多的关系,如学生表和成绩表,则将学生表称为主表,将成绩表称为相关表。

在数据库及表创建完成后,还要建立表之间的关系,如图3.11所示。由于不能在已打开的表之间创建或修改关系,所以在创建关系前,需要关闭所有打开的表。

3.3.3 查询

查询是按照一定的条件或要求对数据库中的数据进行检索。它是数据库的核心操作。

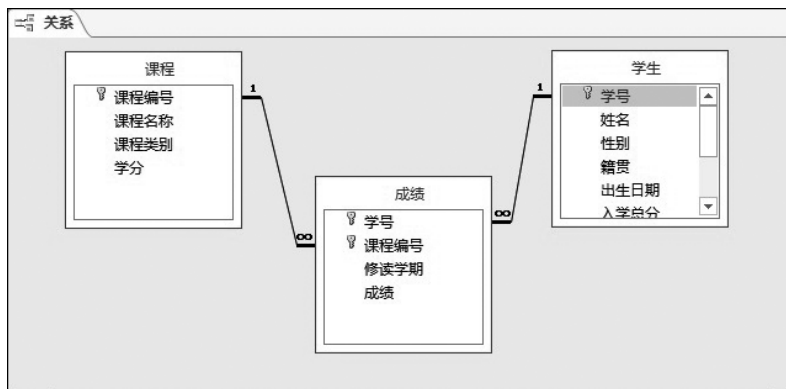


图 3.11 表之间的关系

在数据库中,一个查询可以从一个或多个表中检索数据,也可以对查询的结果做进一步的查询处理,还可以将查询结果用作窗体、报表的数据源。

Access 提供了强大的查询工具,可以按照不同方式查看数据库中的数据。使用查询,可以非常容易地实现下列目标。

- (1) 查找满足条件的字段。
- (2) 查找符合条件的记录。
- (3) 对某些字段进行计算,如求平均值、最大值和统计记录等。
- (4) 将查询的结果作为窗体或报表的数据源。
- (5) 修改、删除和更新表中的数据。

1. 查询的种类

在 Access 中可以创建下列类型的查询:选择查询、参数查询、交叉表查询、操作查询和 SQL 查询。

1) 选择查询

选择查询是最常见的查询类型,它可以从当前数据库中的一个或多个表中按照一定的条件检索数据,也可以使用选择查询对记录进行分组,并对记录进行合计、计数、平均值以及其他类型的统计计算。

查询的结果是一个包含一条或多条记录的记录集,该记录集是动态的,在记录集中的查询结果始终是最新的。

2) 参数查询

参数查询在执行时显示对话框以提示用户输入相关信息,再按用户输入的内容执行相应的查询操作。

3) 交叉表查询

交叉表查询是把一个表或查询作为数据源,将表或查询中的某个字段的统计值(合计、计数以及平均值)作为查询结果,并将它们分组,一组列在数据表的左侧,另一组列在数据表的上部。

使用交叉表查询,除了需要指明对哪个字段进行统计操作外,还需要指定进行分类的行和列字段,这些字段必须有离散值(不连续的值),以便能对数据源中的数据进行分类,以计

算统计结果。

4) 操作查询

操作查询是在一个查询中更改多条记录的查询,共有4种类型:删除查询、更新查询、追加查询与生成表查询。

(1) 删除查询:从一个或多个表中删除满足条件的一组记录。使用删除查询,将删除整条记录,而不只是记录中所选择的字段。

(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,可以更改已存在表中的数据。

(3) 追加查询:从一个或多个表中将满足条件的一组记录追加到一个表的尾部。

(4) 生成表查询:使用一个或多个表中的全部或部分数据新建表。

5) SQL 查询

SQL 查询是用户使用 SQL 语句创建的查询。除使用符合 SQL 语法规定的 SQL 命令外,还支持联合查询、传递查询、数据定义查询和子查询等特殊查询。

2. 查询的基本操作

Access 为查询对象提供了3种视图:设计视图、SQL 视图和数据表视图。查询的设计视图主要用于创建和修改查询,SQL 视图用于查看和修改 SQL 语句,数据表视图以行列方式查看查询结果中的数据。从查询“设计视图”切换到另外两种视图的方法:单击“设计”选项卡“结果”选项组中的“视图”按钮,打开“视图”列表,选择“SQL 视图”或“数据表视图”。

3. 建立查询的方式

1) 使用向导创建查询

Access 提供了4种查询向导:简单查询向导、交叉表查询向导、查找重复项查询向导和查找不匹配项查询向导。启动查询向导的方法:单击“创建”选项卡“查询”选项组中的“查询向导”按钮,打开“新建查询”对话框,可以根据需要选择适当的查询向导创建查询。

2) 使用查询的设计视图创建查询

除了使用向导建立查询外,还可以使用查询的设计视图创建查询。使用查询的设计视图可以从无到有创建一个查询。在查询的设计视图中,既可以建立简单查询(如选择查询),也可以建立较复杂的查询(如操作查询)。如果觉得向导生成的查询不能满足要求,也可以利用查询的设计视图进行修改。因此,使用查询的设计视图建立查询比使用向导灵活。

在查询的设计视图中,设置查询的条件。该窗口的上半部分是字段列表,下半部分是查询设计网格。在字段列表中显示作为查询数据源的表或查询,以及它们之间的关系,每个表或查询中的字段名称都出现在字段列表中。如果被添加到查询的设计视图中的表之间已经建立了关系,则表之间的关系也被自动添加进来;如果被添加到查询的设计视图中的表之间没有建立关系,则可以在该窗口中创建表之间的关系,也可以修改及删除关系。

在查询设计网格中有“字段”“表”“总计”“排序”“显示”“条件”等项目。其中,“字段”设置查询结果中要显示的字段;“排序”指定在查询结果中记录按哪个字段中的数据进行排序,它有两种类型:升序和降序;“显示”决定了选定的字段是否显示在查询结果中;“条件”中的每列则指定了筛选记录的限制条件。如果要在查询设计网格中指定字段的条件,则可以在

该字段的“条件”单元格中输入相应的表达式。如果在多个列上定义了筛选条件,各筛选条件可以用 And(与)或 Or(或)运算符连接:位于同一行的筛选条件表达式之间是 And 关系,处于不同行的筛选条件表达式之间是 Or 关系。“查询的设计视图”窗口如图 3.12 所示。

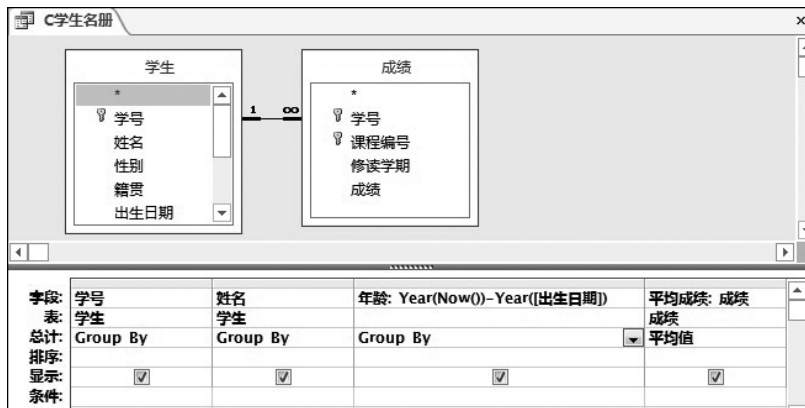


图 3.12 “查询的设计视图”窗口

例 3.1 使用设计视图创建一个名为“学生名册”的选择查询,从数据库中查询所有学生的学号、姓名、年龄和平均成绩。

- (1) 单击“创建”选项卡“查询”选项组中的“查询设计”按钮,打开“显示表”对话框。
- (2) 在“显示表”对话框中,先依次双击“学生”表、“课程”表和“成绩”表,再单击“关闭”按钮。
- (3) 在“查询的设计视图”窗口上部,双击“学生”表中的“学号”和“姓名”字段,将这两个字段添加到查询设计网格中。
- (4) 定义用于计算年龄的字段并为其指定别名:在查询设计网格中,单击第三列中的“字段”单元格,然后输入计算年龄的表达式“年龄: Year(Now())-Year([出生日期])”。
- (5) 单击“设计”选项卡“显示/隐藏”选项组中“汇总”命令,使查询设计网格中显示“总计”行。
- (6) 在查询设计网格“字段”行的第四列中输入“平均成绩:成绩”,并在该列的“总计”单元格中选择“平均值”。
- (7) 单击“设计”选项卡“结果”选项组中“运行”命令,以查看选择查询的运行结果。
- (8) 选择“视图”列表中的“SQL 视图”,切换到“SQL 视图”,查看创建查询时所生成的 SQL 语句。
- (9) 单击“保存”按钮,打开“另存为”对话框,在“另存为”对话框中将查询的名称指定为“学生名册”,再单击“确定”按钮。

例 3.2 使用设计视图创建一个交叉表查询,要求用“学号”和“姓名”字段作为行标题,用“课程名称”字段作为列标题,在行列交叉处显示各门课程的成绩。

- (1) 单击“创建”选项卡“查询”选项组中的“查询设计”按钮,打开“显示表”对话框。
- (2) 向查询中添加表。在“显示表”对话框中,先依次双击“学生”表、“课程”表和“成绩”表,再单击“关闭”按钮。