



全国1001所高校学子的明智选择

(第3版)

全国计算机等级考试 全能教程

二级Access数据库程序设计

全国计算机等级考试命题研究组◎编写

赠 考试系统一套

下载地址: www.buptpress.com



资源下载

- **权威性:**
命题研究组资深专家最新研究成果, 作者长期工作在教学一线。
- **全真性:**
所有试题均紧扣最新大纲的要求, 布局均与考试真题完全相当。
- **实战性:**
配套系统提供现场做题环境, 便于考生在考前60天实战冲刺。
- **前瞻性:**
在深入研究历年真题的基础上, 提供严密的知识体例供学习及应试。

全国大学生最关注的等考品牌

北邮·等考



北京邮电大学出版社
www.buptpress.com

全国计算机等级考试全能教程

——二级 Access 数据库程序设计(第3版)

全国计算机等级考试命题研究组 编写

北京邮电大学出版社
·北京·

内 容 简 介

本书根据全国计算机等级考试最新考试大纲和官方教程,在研究历年真题的基础上编写而成。本书章节安排与官方教程一致。每章末尾设置有“应试加油站”,该板块汇集重点提示、解题技巧等部分,引导考生掌握重点内容,增强考生的解题能力和综合应用能力。在正文中提供一章进行无纸化考试专题辅导。本书附有多套无纸化考试样卷,供考生考前实战演练。

本书配有考试系统,考试系统中的配套软件完全模拟真题考试环境,便于考生实战演练。

本书可供全国计算机等级考试考生复习使用,特别适合考前冲刺使用,同时也非常适合相关等级考试培训班用作培训教材,以及大、中专院校师生的教学用书。

图书在版编目(CIP)数据

全国计算机等级考试全能教程. 二级 Access 数据库程序设计/全国计算机等级考试命题研究组编写.--3 版.--北京:北京邮电大学出版社,2016.4

ISBN 978-7-5635-4716-6

I. ①全… II. ①全… III. ①电子计算机—水平考试—教材②关系数据库系统—程序设计—水平考试—教材
IV. ①TP3

中国版本图书馆 CIP 数据核字(2016)第 045971 号

书 名: 全国计算机等级考试全能教程——二级 Access 数据库程序设计(第 3 版)

作 者: 全国计算机等级考试命题研究组

责任编辑: 姚 顺

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish @ bupt. edu. cn

经 销: 各地新华书店

印 刷: 保定市中国画美凯印刷有限公司

开 本: 889 mm×1 194 mm 1/16

印 张: 11.5

字 数: 339 千字

版 次: 2016 年 4 月第 3 版 2016 年 4 月第 1 次印刷

ISBN 978-7-5635-4716-6

定 价: 32.80 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前 言

全国计算机等级考试为国内影响最大、参加人数最多的计算机类水平考试,在推广、普及计算机应用知识和技术中发挥了重要作用,并为用人单位的人员考核提供了客观、公正的评价标准。

为了引导考生顺利通过全国计算机等级考试,我们根据新大纲的要求,结合典型试题,按教育部考试中心指定教材的篇章结构,由从事全国计算机等级考试试题研究人员及在等级考试第一线从事命题研究、教学、辅导和培训的老师精心编写了《全国计算机等级考试全能教程——二级 Access 数据库程序设计(第3版)》。

1. 本书特色

☑ **突出实用性和高效性:**书的章名、节名与教育部考试中心指定教程同步,每章还设计了以下板块。

➤ **考题链接:**精选出常考题型与历年真题穿插在知识点中的讲解,一方面有利于考生对知识点的理解,另一方面也让考生明白试题是如何考的。

➤ **应试加油站:**该板块汇集考试重点整理、解题技巧等部分,引导考生掌握重点内容,增强考生的解题能力和综合应用能力。

➤ **习题:**针对本节知识点设计一部分题目,方便读者一点一练,巩固提高。

☑ **突出标准性与严谨性:**本书由从事全国计算机等级考试试题研究人员及在等级考试第一线从事命题研究、教学、辅导和培训的老师分工编写,层次清晰,结构严谨,导向准确。

☑ **结构科学,实用性强:**紧扣新大纲要求,精讲考点、重点与难点,深入分析典型范例,抓住等级考试题眼,并提供实战训练。

☑ **注重无纸化考试的辅导:**针对无纸化考试的特点,本书在深入研究无纸化考试题库的基础上,将真题进行分类,提炼出题型,按类型进行解析,便于考生专项攻克,提高复习效率。

☑ **全真模拟,实战提高:**根据新大纲、新考点、新题型进行最新命题,书末提供4套真题,供考生考前实战演练。

☑ **系统结合,一本速通:**系统中包括考试模拟系统,提供数套真题供考生联系,考试环境与真实考试一致,帮助考生顺利过关。

2. 读者对象

本书以全国计算机等级考试考生为主要读者对象,适合于考生在等考前复习使用,也可作为相关考试培训班的辅助教材,以及大、中专院校师生的参考书。

3. 本书作者

本书由冯伯虎、周胜、董清爽主编,参与本书编写、系统制作的人员还有卢振侠、周汉、顾锦江、李建荣、吴蕾、何光明、陈志荣、毛辉杰、郭丽红、吴海涛、余永红、蒋晶、赵强。

由于作者水平有限,书中难免存在疏漏和错误之处,恳请专家和广大读者批评指正。在学习过程中,遇到疑难问题,可以通过以下方式与我们联系:bjbaba@263.net。

编 者

全国计算机等级考试全能教程丛书

顾问委员会

成员名单 (排名不分先后)

邓杰海	邓卫民	刁爱军	董清爽	董自涛	方星星
冯伯虎	傅伟玉	高胜利	耿艳	胡波	黄琦
贾春燕	贾晓燕	孔德丽	李赛红	李文宏	刘伟
刘志强	刘子明	吕永强	牛雪飞	史国川	汤小燕
王成杰	王芳	夏良	徐国明	赵军	赵明
钟彩华	周胜	祝爱芬			

本书主编 冯伯虎 周胜 董清爽

目 录

第1章 数据库基础	1
1.1 数据库基础知识	1
1.1.1 计算机数据管理的发展	1
1.1.2 数据库系统	2
1.1.3 数据模型	4
1.2 关系数据库	6
1.2.1 关系数据库模型	6
1.2.2 关系运算	8
1.2.3 关系模型的完整性	9
1.2.4 规范化理论	9
1.3 数据库设计基础	11
1.3.1 数据库设计的原则及步骤	12
1.3.2 数据库设计过程	12
1.4 Access 简介	14
1.4.1 Access 2010 简介	14
1.4.2 Access 2010 数据库的系统结构	15
1.5 应试加油站	16
1.5.1 考试重点整理	16
1.5.2 解题技巧	16
1.6 过关练习与答案	17
1.6.1 过关练习	17
1.6.2 参考答案	19
第2章 数据库和表	20
2.1 创建数据库	20
2.1.1 使用模板创建数据库	20
2.1.2 创建空白数据库	20
2.2 数据库对象的组织	20
2.3 数据库对象的操作	20
2.3.1 数据库对象的打开	21
2.4 表与表的设计	21
2.4.1 表的结构	21
2.4.2 创建表结构	22
2.4.3 设置字段属性	22
2.4.4 主码	25
2.5 输入与维护表数据	27
2.5.1 数据输入	27

2.5.2 导入、导出数据	27
2.5.3 链接数据	27
2.6 表的使用	27
2.6.1 排序记录	27
2.6.2 筛选记录	28
2.6.3 设置数据表格式	28
2.7 表的索引	29
2.8 建立表间关联关系	29
2.8.1 表与表之间的关系	29
2.8.2 编辑与删除表间关联	29
2.9 应试加油站	30
2.9.1 考试重点整理	30
2.9.2 解题技巧	30
2.10 过关练习与答案	31
2.10.1 过关练习	31
2.10.2 参考答案	32
第3章 查询	33
3.1 查询的概述	33
3.1.1 查询的种类	33
3.1.2 查询的应用	35
3.2 查询条件	35
3.2.1 表达式	35
3.2.2 标准函数	37
3.3 查询的建立方法	39
3.3.1 使用查询向导	39
3.3.2 使用查询设计器	39
3.4 交叉表查询	40
3.5 参数查询	40
3.6 操作查询	40
3.7 SQL 查询	41
3.7.1 SQL 的数据定义	41
3.7.2 SQL 的数据操纵	42
3.7.3 SQL 查询	43
3.7.4 SQL 视图	45
3.8 编辑和使用查询	45
3.9 应试加油站	46
3.9.1 考试重点整理	46
3.9.2 解题技巧	46



3.10 过关练习与答案	48	5.4 报表排序和分组	65
3.10.1 过关练习	48	5.5 使用计算控件	65
3.10.2 参考答案	49	5.6 创建子报表	66
第4章 窗体	50	5.6.1 子报表的概念	66
4.1 窗体的基础知识	50	5.6.2 在已有报表中创建子报表	66
4.1.1 窗体的作用	50	5.6.3 将某个报表添加到其他已有报表来 创建子报表	67
4.1.2 窗体构成	50	5.7 打印输出报表	67
4.1.3 窗体视图	51	5.7.1 预览报表	67
4.1.4 窗体分类	51	5.7.2 打印报表	67
4.2 创建窗体	51	5.8 应试加油站	68
4.2.1 自动创建窗体	51	5.8.1 考试重点整理	68
4.2.2 自动创建分割窗体	52	5.8.2 解题技巧	68
4.2.3 使用“空白窗体”创建窗体	52	5.9 过关练习与答案	69
4.2.4 利用向导创建窗体	52	5.9.1 过关练习	69
4.2.5 创建数据透视表窗体、数据透视图窗体	52	5.9.2 参考答案	69
4.3 设计窗体	52	第6章 宏	70
4.3.1 常用控件	52	6.1 宏的概念	70
4.3.2 创建计算控件	53	6.2 宏的创建	70
4.3.3 窗体与控件的属性	53	6.2.1 创建独立宏	70
4.4 窗体修饰	55	6.2.2 创建宏组	71
4.4.1 应用主题	55	6.2.3 创建条件操作宏	71
4.4.2 条件格式的应用	55	6.2.4 设置宏的操作参数	71
4.4.3 提示信息的添加	55	6.3 宏的调试和运行	71
4.4.4 设置窗体布局	55	6.3.1 宏调试	72
4.4.5 使用 Tab 键设置控件次序	56	6.3.2 运行宏	72
4.5 应试加油站	56	6.4 通过事件触发宏	73
4.5.1 考试重点整理	56	6.4.1 事件的概念	73
4.5.2 解题技巧	57	6.4.2 通过事件触发宏	73
4.6 过关练习与答案	58	6.5 应试加油站	73
4.6.1 过关练习	58	6.5.1 考试重点整理	73
4.6.2 参考答案	59	6.5.2 解题技巧	74
第5章 报表	60	6.6 过关练习	75
5.1 报表的基础知识	60	6.6.1 过关练习	75
5.1.1 报表的定义	60	6.6.2 参考答案	75
5.1.2 报表的结构	60	第7章 模块与VBA程序设计	76
5.1.3 报表的视图	61	7.1 模块的基本概念	76
5.1.4 报表的分类	62	7.1.1 类模块	76
5.2 创建报表	62	7.1.2 标准模块	76
5.2.1 使用“报表”工具创建报表	62	7.2 创建模块	77
5.2.2 使用“报表设计”工具创建报表	62	7.3 VBA程序设计基础	78
5.2.3 使用“空报表”工具创建报表	63	7.3.1 使用VBA编程的场合	78
5.2.4 使用“报表向导”创建报表	63	7.3.2 面向对象程序设计的基本概念	78
5.2.5 使用“标签向导”创建报表	63	7.3.3 VB编程环境:VBE	80
5.3 报表的编辑	63		



7.3.4 数据类型	82	7.9.2 解题技巧	109
7.3.5 常量、变量与数组	83	7.10 过关练习与答案	114
7.3.6 运算符与表达式	86	7.10.1 过关练习	114
7.3.7 函数	89	7.10.2 参考答案	118
7.4 VBA 流程控制语句	92	第 8 章 VBA 数据库编程	119
7.4.1 声明语句	92	8.1 数据库引擎及其接口	119
7.4.2 赋值语句	93	8.2 VBA 访问的数据库类型	120
7.4.3 标号和 Goto 语句	93	8.3 数据访问对象	120
7.4.4 执行语句	93	8.4 ActiveX 数据对象(ADO)	121
7.5 过程调用与参数传递	96	8.5 数据库编程分析	125
7.5.1 过程调用	96	8.6 应试加油站	127
7.5.2 参数传递	97	8.6.1 考试重点整理	127
7.6 VBA 常用操作方法	98	8.6.2 解题技巧	128
7.6.1 打开和关闭操作	98	8.7 过关练习与答案	129
7.6.2 输入框(InputBox)	100	8.7.1 过关练习	129
7.6.3 消息框(MsgBox)	100	8.7.2 参考答案	130
7.6.4 VBA 编程验证数据	102	全国计算机等级考试二级 Access 模拟题(一)	131
7.6.5 计时事件(Timer)	102	全国计算机等级考试二级 Access 模拟题(二)	135
7.6.6 鼠标和键盘事件处理	102	全国计算机等级考试二级 Access 模拟题(三)	139
7.6.7 数据文本读写	103	全国计算机等级考试二级 Access 模拟题(四)	143
7.6.8 用代码设置 Access 选项	103	附录	147
7.7 用户定义类模块创建和引用	104	模拟题(一)答案解析	147
7.7.1 用户定义类模块概念	104	模拟题(二)答案解析	156
7.7.2 用户定义类模块创建和引用	106	模拟题(三)答案解析	162
7.8 VBA 程序的运行错误处理与调试	106	模拟题(四)答案解析	169
7.8.1 程序的运行错误处理	106		
7.8.2 程序的调试	108		
7.9 应试加油站	109		
7.9.1 考试重点整理	109		

第 1 章 数据库基础

1.1 数据库基础知识

数据库是 20 世纪 60 年代末发展起来的一项重要技术,它的出现使数据处理进入了一个崭新的时代。它能把大量的数据按照一定的结构存储起来,在数据库管理系统的集中管理下,实现数据共享。

1.1.1 计算机数据管理的发展

1. 数据与数据处理

数据是指存储在某一载体上能够被识别的物理符号。数据包含两个方面的内容,一是对事物特征的描述,表示事物的属性,如大小、形状、数量等;二是存储的形式,如数字、文字、图形、图像、声音、动画、影像等。

数据处理是指将数据转换成信息的过程。信息是一种被加工成特定形式的数据,这种数据形式对数据接受者来说是有意义的。

2. 数据管理技术

在应用需求的推动下,数据管理技术得到迅猛发展,在整个利用计算机进行数据管理的发展过程中又经历了人工管理、文件系统、数据库系统三个阶段。当前的计算机数据处理是基于数据库的一种计算机应用和发展,它是按特定需求对数据进行加工的过程。

(1) 人工管理阶段

20 世纪 50 年代以前,计算机主要用于数据计算。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,没有操作系统及数据管理的软件;从数据看,数据量小,用于数据结构的模型没有完善。所以这一阶段的管理由用户直接管理,存在以下主要特点:

- 数据不能长期保存。
- 数据相对于程序不具有独立性。
- 数据不共享。

(2) 文件管理阶段

20 世纪 50 年代后期到 60 年代中期,计算机外部存储设备中出现了磁鼓、磁盘等直接存取的存储设备;计算机操作系统中已有了专门的管理数据软件,称为文件系统。在数据的处理方式上不仅有了文件批处理,而且能够在需要时随时从存储设备中查询、修改或更新数据。这时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,并可以按文件的名字进行访问,所以称为文件管理阶段。这一阶段的特点如下:



- 数据可组织成文件长期保存在计算机中,并可经常进行查询、修改和删除等操作。
- 数据具有较低的独立性。
- 数据共享性低,冗余度大。

(3) 数据库系统阶段

20 世纪 60 年代后期,计算机性能大幅度提高,特别是大容量磁盘的出现,使存储容量大大增加并且价格下降。为满足和解决实际问题中多个用户、多个应用程序共享数据的要求,使数据能为尽可能多的应用程序服务,在软件方面就出现了统一管理数据的专用软件系统,克服了文件系统管理数据时的不足,这就是数据库管理技术。

(4) 分布式数据库系统阶段

数据库技术与通信网络技术的结合产生了分布式数据库系统。网络技术的发展为数据库提供了分布式运行的环境,从主机—终端体系结构发展到客户机/服务器(Client/Server,C/S)体系结构。

(5) 面向对象数据库系统阶段

数据库技术与面向对象程序设计技术相结合产生了面向对象的数据库系统。它采用面向对象的观点来描述现实世界实体(对象)的逻辑组织、对象之间的限制和联系等,克服了传统数据库的局限性,能够自然地存储复杂的数据对象以及这些对象之间的复杂关系,大大提高了数据库管理的效率,降低了用户使用的复杂性。

从本质上说,Access 仍然是传统的关系型数据库系统,但它在用户界面、程序设计等方面进行了扩充,提供了面向对象程序设计的功能。

1.1.2 数据库系统

1. 数据库

数据库(DataBase,DB)是存储在计算机存储设备上、结构化的相关数据集合。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。数据库中的数据按一定的数据模型组织、描述和存储,具有较小的冗余度、较高的数据独立性和易扩展性,并可供各种用户共享。对于数据库中数据的增加、删除、修改和检索等操作均由系统软件进行统一的控制。

2. 数据库管理系统

数据库管理系统(DataBase Management System,DBMS)是位于用户与操作系统之间的一层数据管理软件。市场上可以看到各种各样的数据库管理系统软件产品,如 Oracle、SQL Server、Access、Visual FoxPro、Informix、Sybase 等。其中,Oracle、SQL Server 数据库管理系统适用于大中型数据库;Access 是微软公司 Office 办公套件中一个极为重要的组成部分,是目前世界上最流行的桌面数据管理系统,它适用于中小型数据库应用系统。

数据库管理系统的主要功能包括以下几个方面:

(1) 数据定义功能。数据库管理系统提供数据定义语言,通过它可以方便地对数据库中的相关内容进行定义。如对数据库、基本表、视图、查询和索引等进行定义。

(2) 数据操纵功能。数据库管理系统提供数据操纵语言,实现对数据库的基本操作,如对数据库中数据的插入、删除、修改和查询等操作。

(3) 数据库的运行管理。这是数据库管理的核心部分,所有数据库操作都是在系统的统一管理下进行,以保证数据的安全性、完整性以及多用户对数据库的并发使用。

(4) 数据的组织、存储和管理。数据库中需要存放多种数据,DBMS 需要确定以何种文件结构和存取方式物理地组织这些数据,如何实现数据之间的联系,以便提高存储空间利用率以及提高查找、增加、删除、修改等操作的时间效率。

(5) 数据库的建立和维护。



(6) 数据通信接口。DBMS 需要提供与其他软件系统进行通信的功能。

为了提供上述功能,数据库管理系统提供以下 4 类语言和程序:

(1) 数据定义语言及其翻译处理程序

数据定义语言(Data Definition Language, DDL)供用户定义数据库的外模式、模式、内模式、各级模式间的映射、有关的约束条件等。

(2) 数据操纵语言及其变异(或解释)程序

数据操纵语言(Data Manipulation Language, DML)实现对数据库的检索、插入、修改、删除等基本操作。

(3) 数据库运行控制程序

数据库管理系统提供了一些负责数据库运行过程中的控制与管理的系统运行控制程序,包括系统初始化程序、文件读写与维护程序、存取路径管理程序、缓冲区管理程序、安全性控制程序等。

(4) 实用程序

数据库管理系统通常还提供一些实用程序,包括数据初始装入程序、数据转储程序、数据库恢复程序、性能检测程序、通信程序等。

3. 数据库应用系统

数据库应用系统是由系统开发人员利用数据库系统资源开发出来的、面向某一类实际应用的应用软件系统。

4. 用户

用户指与数据库系统打交道的人员,包括以下 3 类人员:

- 数据库应用系统开发员:开发数据库系统的人员。
- 数据库管理员:全面负责数据库系统的正常运行和维护的人员。
- 最终用户:使用数据库应用系统的人员。

5. 数据库系统

数据库系统(DBS)是指引入数据库后的计算机系统。一般由数据库、数据库管理系统及其开发工具、应用系统、数据库管理员和用户构成。

数据库技术是信息系统的核心和基础,它的出现极大地促进了计算机应用向各行各业的渗透。从一般的小型事务处理到大型的信息系统,越来越多的领域开始采用数据库技术存储与处理其信息资源。

数据库系统的主要特点如下:

- 采用特定的数据结构,以数据库文件组织形式长期保存。
- 实现数据共享,冗余度小。
- 具有较高的独立性。
- 具有统一的数据控制功能。

数据库系统在其内部采用了三级模式和二级映射的抽象结构体系。三级模式分别为概念级模式、内部模式和外部模式,二级映射分别为概念级到内部级的映射、外部级到概念级的映射。

(1) 数据库系统的三级模式

1) 概念模式。概念模式是数据库系统中全局数据逻辑结构的描述,是全体用户(应用)公共数据视图。该模式与具体的硬件环境、软件环境及平台无关。概念模式可用 DBMS 中的 DDL 语言定义。

2) 外模式。外模式也称子模式或用户模式,是用户所看到和理解的数据模式,是从概念模式导出的子模式。外模式给出了每个用户的局部数据描述。DBMS 一般提供相关的外模式描述语言(外模式 DDL)。

3) 内模式。内模式又称为物理模式,它给出了物理数据库的存储结构和物理存取方法,如数据存储的文件结构、索引、集簇及存取路径。DBMS 一般提供相关的内模式描述语言(内模式 DDL)。



(2) 数据库系统的二级映射

1) 外模式到概念模式的映射

概念模式描述系统的全局逻辑结构,外模式描述每个用户的局部逻辑结构。对应于一个概念模式可以有任意多个外模式。对应于每一个外模式,数据库系统都有一个从外模式到概念模式的映射,该模式给出了外模式与概念模式的对应关系。

应用程序是依据数据的外模式编写的,当数据库模式改变时,通过对各个外模式到概念模式的映射作相应改变,可以使外模式保持不变,从而不必修改应用程序,保证了数据与程序的逻辑独立性,简称数据的逻辑独立性。

2) 概念模式到内模式的映射

由于数据库只有一个概念模式和一个内模式,所以数据库中从概念模式到内模式的映射是唯一的。这种映射定义了数据全局逻辑结构与存储结构之间的对应关系。

当数据库的存储结构发生改变时,数据库管理员通过修改内模式到概念模式映射,可使概念模式保持不变,使应用程序不受影响,保证了数据与程序的物理独立性,简称数据的物理独立性。

1.1.3 数据模型

1. 数据模型的组成

数据模型所描述的内容包括数据结构、数据操作和数据约束 3 部分。

(1) 数据结构。数据结构主要描述数据的类型、内容、性质和数据之间的联系等。数据结构是数据模型的基础,数据操作和数据约束均建立在数据结构的基础之上。

(2) 数据操作。数据操作主要描述在相应数据结构上的操作类型和操作方式。数据库的操作主要有检索和更新两大类。

(3) 数据约束。数据约束是一组完整性规则的集合,主要描述数据结构内数据间的语法、语义联系,它们之间的制约与依存关系,以及数据动态变化的规则,以确保数据的正确、有效和相容。

2. 数据模型的分类

数据模型按不同的应用层次分为概念数据模型、逻辑数据模型和物理数据模型。

(1) 概念数据模型。概念数据模型简称概念模型,它是一种面向客观世界、面向用户的模型,与具体的平台和数据库管理系统无关。概念模型是整个数据模型的基础,较为著名的概念模型有 E-R 模型、扩充的 E-R 模型、面向对象模型等。

(2) 逻辑数据模型。逻辑数据模型又称数据模型,是面向数据库系统的模型,着重于在数据库系统一级的实现。较为成熟的逻辑数据模型有层次模型、网状模型、关系模型和面向对象模型等。

(3) 物理数据模型。物理数据模型又称物理模型,是面向计算机物理表示的模型。

3. E-R 模型

(1) 实体(Entity)

实体是客观存在并可相互区别的事物。实体可以是实际事物,也可以是抽象事件。例如,一个学生、一个部门属于实际事物;一次订货、借阅若干本图书是比较抽象的事件。同一类实体的集合称为实体集。例如,全体职工的集合、全馆图书等。

(2) 属性(Attribute)

属性刻画了实体的特性。一个实体往往可以有若干个属性。例如,职工实体可以用若干个属性(职工编号、姓名、性别、出生日期、职位等)来描述。属性的具体取值称为属性值,一个属性的取值范围称为该属性的值域或值集。

(3) 联系(Relationship)

实体集之间的对应关系称为联系,它反映现实世界事物之间的相互关联。如生产者和消费者之间的



供求关系。

两个实体集之间的联系实际上是实体集之间的函数关系,有3种类型:

1) 一对一(One to One)的联系。如果对于实体集 A 中的每一个实体,实体集 B 中至多有一个实体与之联系,反之亦然,则称实体集 A 与实体集 B 具有一对一的联系,记为 1:1。

2) 一对多(One to Many)的联系。如果对于实体集 A 中的每一个实体,实体集 B 中有 n 个实体($n \geq 0$)与之联系,反之,对于实体集 B 中的每一个实体,实体集 A 中至多只有一个实体与之联系,则称实体集 A 与实体集 B 具有一对多的联系,记为 1:n。

3) 多对多(Many to Many)的联系。如果对于实体集 A 中的每一个实体,实体集 B 中有 n 个实体($n \geq 0$)与之联系,反之,对于实体集 B 中的每一个实体,实体集 A 中也有 m 个实体($m \geq 0$)与之联系,则称实体集 A 与实体集 B 具有多对多的联系,记为 $m:n$ 。

E-R 模型可以用一种非常直观的图的形式来描述现实世界的概念模型。这种图称为 E-R 图, E-R 图有 3 个要素。

- 实体:用矩形并在框内标注实体名称来表示。
- 属性:用椭圆形表示,并用连线将其与相应的实体连接起来。
- 联系:用菱形表示,菱形框内写明联系名,并用连线分别与有关实体连接起来。

每个数据库管理系统都是基于某种数据模型的。在目前数据库领域中,常用的数据模型有层次模型、网状模型、关系模型和面向对象模型等。

4. 层次模型

层次模型是最早发展起来的数据模型,它是把客观问题抽象为一个严格的自上而下的层次关系。层次模型用树形结构表示各类实体以及实体间的联系,它具有以下特点:一是有且仅有一个根结点无双亲,这个结点即为树的根;二是其他结点有且仅有一个双亲。因此,层次模型只能反映实体间的一对多的联系。现实世界中许多实体间存在着自然的层次关系,如组织机构、家庭关系和物品分类等。

5. 网状模型

网状模型的数据结构是一个网络结构,其基本特征是:一个双亲允许有多个子女;反之,一个子女也可以有多个双亲。广义地讲,任意一个连通的基本层次联系的集合就是一个网状模型。

与层次模型不同,网状模型中的任意结点间都可以有联系,适用于表示多对多的联系,因此,与层次模型相比,网状模型更具有普遍性。也可以认为层次模型是网状模型的特例。

6. 关系模型

关系模型是各种数据模型中最为重要的模型。关系模型是建立在数学概念基础上的,在关系模型中,把数据看成一个二维表,这个二维表就叫作关系。

与层次和网状模型相比,关系模型的数据结构单一,不管实体还是实体间的联系都用关系来表示;同时关系模型是建立在严格的数学概念基础上的,具有坚实的理论基础;此外,关系模型还将数据定义和数据操纵统一在一种语言中,易学易用。

7. 面向对象模型

面向对象数据模型中的主要概念有对象、类、方法、消息、封装、继承和多态等。其中,最基本的概念是对象(Object)和类(Class)。对象是现实世界中实体的模型化,每一个对象有唯一的标识符,把“状态”和“行为”封装在一起。其中,对象的“状态”是该对象属性值的集合,对象的“行为”是在对象状态上操作的方法集。一个对象由一组属性和一组方法组成,属性用来描述对象的特征,方法用来描述对象的操作。一个对象的属性可以是另一个对象,另一个对象的属性还可以用其他对象描述,以此来模拟现实世界中的复杂实体。

面向对象的数据模型主要具有以下优点:

- 可以表示复杂对象,精确模拟现实世界中的实体。



- 具有模块化的结构,便于管理和维护。
- 具有定义抽象数据类型的能力。

1.2 关系数据库

1.2.1 关系数据库模型

一个关系可看作一个二维表,由各表示一个实体的若干行或各表示实体(集)某方面属性的若干列组成。可以用数学语言将关系定义为元组的集合。关系有“型”和“值”之分,关系模式是对关系的型即关系数据结构的描述,关系可看成按型填充所得到的值。一般来说,关系模式是稳定的,而关系本身却会跟随所描述的客观事物的变化而不断变化。客观世界中事物的性质是互相关联的,往往还要受到某些限制。在关系数据模型中,这些关联和限制表现为一系列数据的约束条件,包括域约束、键(码)约束、完整性约束和数据依赖等。

1. 关系模型的组成

关系模型由关系数据结构、关系操作集合和关系完整性约束 3 大要素组成。

(1) 关系数据结构

关系模型把数据库表示为关系的集合。在用户看来,关系模型中数据的逻辑结构是一张二维表。关系模型的数据结构单一,在关系模型中,现实世界的实体以及实体间的各种联系均可用关系来表示。

(2) 关系操作集合

关系模型中常用的操作包括选择、投影、连接、除、并、交、差等以及查询操作和插入、删除、更新操作。查询操作是其中最主要的部分。

(3) 关系的完整性约束

数据库的数据完整性是指数据库中数据的正确性、相容性和一致性。这是一种语义概念,包括两方面的意思:

- 1) 数据库中的数据与现实世界中应用需求的数据的正确性、相容性和一致性。
- 2) 数据库内数据之间的正确性、相容性和一致性。

数据的完整性由完整性规则来定义,关系模型的完整性规则是对关系的某种约束,因此也称为完整性约束。它提供了一种手段来保证当用户对数据库进行插入、删除、更新时不会破坏数据库中数据的正确性、相容性和一致性。

2. 关系模型的基本概念

在关系模型中,数据结构用单一的二维表结构来表示实体及实体间的联系。

(1) 关系

在关系模型中,一个关系就是一张二维表,每一个关系有一个关系名。

(2) 属性

二维表中垂直方向的列称为属性,每一个列有一个属性名,列的值称为属性值。

(3) 元组

二维表中水平方向的行称为元组,一行就是一个元组。元组对应数据表中的一条记录,元组的各分量分别对应于关系的各个属性。关系模型要求每个元组的每个分量都是不可再分的数据项。

(4) 域

属性所取值的变化范围称为属性的域(Domain)。域约束规定属性的值必须是来自域中的原子值,即那些就关系模型而言已不可再分的数据,如整数、字符串等,而不应包括集合、记录、数组这样的组合数据。



(5) 关系模式

关系的描述称为关系模式(relation schema)。它可以形式化地表示为

$$R(U, D, \text{Dom}, F)$$

其中, R 为关系名, U 为组成关系的属性名集合, D 为属性组 U 中属性所来自的域, Dom 为属性向域的映像集合, F 为属性间数据依赖关系的集合。

在关系数据库中, 关系模式是型, 关系是值, 关系模式是静态的, 关系是关系模式在某一时刻的状态或内容, 关系是动态的。

(6) 候选码(候选键)

如果关系中的一个码移去了任何一个属性, 它就不再是这个关系的码, 则称这样的码为该关系的候选码或候选键。

(7) 主码(主键)

一个关系中往往有多个候选码, 若选定其中一个用来唯一标识该关系的元组, 则称这个被指定的候选码为该关系的主码(主键)。

(8) 外码(外键)

当关系中的某个属性或属性组虽然不是这个关系的主码, 或只是主码的一部分, 但却是另一个关系的主码时, 则称该属性或属性组为这个关系的外码或外键。

(9) 主属性和非主属性

关系中包含在任何一个候选码中的属性称为主属性, 不包含在任何一个候选码中的属性称为非主属性。

3. 关系形式

关系是属性值域笛卡儿积的一个子集。

(1) 笛卡儿积(Cartesian Product)

设有一组域 $D_1, D_2, \dots, D_n$, 这些域可以部分或者全部相同。域 $D_1, D_2, \dots, D_n$ 的笛卡儿积定义为如下集合:

$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, 2, \dots, n\}$$

其中, 每一个元素 $(d_1, d_2, \dots, d_n)$ 称为一个 n 元组(或简称元组), 元素中的每一个值 d_i 称为一个分量。

若干个域的笛卡儿积具有相当多的元素, 在实际应用中可能包含许多“无意义”的元素。人们通常感兴趣的是笛卡儿积的某些子集, 笛卡儿积的子集就是一个关系。

两个集合 R 和 S 的笛卡儿积是元素对的集合, 该元素对是通过选择 R 的某一元素(任何元素)作为第一个元素, S 的元素作为第二个元素构成的, 该乘积用 $R \times S$ 表示。笛卡儿积的结果可表示为一个二维表, 表中的每行对应一个元组, 表中的每列对应一个域。

(2) 关系的形式定义

笛卡儿积 $D_1 \times D_2 \times \dots \times D_n$ 的子集 R 称为在域 $D_1 \times D_2 \times \dots \times D_n$ 上的一个关系(Relation), 通常表示为

$$R(D_1, D_2, \dots, D_n)$$

其中, R 表示关系的名称, n 称为关系 R 的元数或度数(Degree), 而关系 R 中所含有的元组个数称为 R 的基数(Cardinal Number)。

关系是笛卡儿积的子集, 所以关系也是一个二维表, 表的每行对应一个元组, 表的每列对应一个域。由于域可以相同, 为了加以区分, 必须为每列起一个名字, 称为属性(attribute), n 组关系必有 n 个属性。

关系可以有 3 种类型: 基本关系(通常又称为基本表或基表)、查询表和视图表。基本表是实际存在的表, 它是实际存储数据的逻辑表示; 查询表是查询结果对应的表; 视图表是虚表, 是由基本表或其他视图表导出的表, 不对应实际存储的数据。



由上述定义可以知道,域 $D_1, D_2, \dots, D_n$ 上的关系 R , 就是由域 $D_1, D_2, \dots, D_n$ 确定的某些元组的集合。

(3) 关系模型对关系的限制

在关系模型中,对关系作了下列规范性限制:

- 关系中不允许出现相同的元组。
- 不考虑元组之间的顺序,即没有元组次序的限制。
- 关系中每一个属性值都是不可分解的。
- 关系中属性顺序可以任意交换。
- 同一属性下的各个属性的取值必须来自同一个域,是同一类型的数据。
- 关系中各个属性必须有不同的名字。

1.2.2 关系运算

关系代数是集合代数为基础发展起来的。在关系代数的操作中,其操作对象和操作结果均为关系。关系代数也是一种抽象的查询语言,它通过对关系的操作来表达查询。关系代数的基本运算有两类:传统的集合运算(并、交、差等)和专门的关系运算(选择、投影、连接),有些查询需要几个基本运算的组合。

1. 传统的集合运算

并、差、交是集合的传统运算形式,进行集合运算的关系 R 与 S 必须具有相同的模式,即 R 和 S 必须具有相同的属性集。

(1) 并(Union)

设有关系 R, S (R, S 具有相同的模式),则关系 R 与关系 S 的并由属于 R 或者属于 S 的元组组成。记作:

$$R \cup S = \{t | t \in R \vee t \in S\}$$

式中,“ $\cup$ ”为并运算符, t 为元组变量,结果 $R \cup S$ 为一个新的与 R, S 同类的关系。

(2) 差(Difference)

设有关系 R, S (R, S 具有相同的模式),则关系 R 与关系 S 的差由属于 R 而不属于 S 的元组组成。记作:

$$R - S = \{t | t \in R \wedge t \notin S\}$$

式中,“-”为差运算符, t 为元组变量,结果 $R - S$ 为一个新的与 R, S 同类的关系。

(3) 交(Intersection)

设有关系 R, S (R, S 具有相同的模式),则关系 R 与关系 S 的交由既属于 R 又属于 S 的元组组成。记作:

$$R \cap S = \{t | t \in R \wedge t \in S\}$$

式中,“ $\cap$ ”为交运算符,结果 $R \cap S$ 为一个新的与 R, S 同类的关系。

2. 专门的关系运算

在 Access 数据库中,查询是经常使用的数据库操作,了解专门的关系运算有助于查询的设计。

(1) 选择(Selection)

选择是在关系 R 中选择满足给定条件的元组,即从行的角度进行操作。记作:

$$\sigma_F(R) = \{t | t \in R \wedge F(t) = \text{true}\}$$

式中, $\sigma_F(R)$ 表示由从关系 R 中选择出满足条件 F 的元组所构成的关系, F 表示选择条件。

(2) 投影(Projection)

投影是从关系 R 中选择若干属性列,并且将这些列组成一个新的关系,即从列的角度进行操作。



设有关系 R , 其元组变量为 $t^k = \langle t_1, t_2, \dots, t_k \rangle$, 那么关系 R 在其分量 $A_{i1}, A_{i2}, \dots, A_{in}$ ($n \leq k, i1, i2, \dots, in$ 为 1 到 k 之间互不相同的整数) 上的投影记作:

$$\Pi_{i1, i2, \dots, in}(R) = \{t \mid t = \langle t_{i1}, t_{i2}, \dots, t_{in} \rangle \wedge \langle t_1, t_2, \dots, t_k \rangle \in R\}$$

(3) 连接(Join)

连接又称为 θ 连接, 它是将两个关系拼接成一个更宽的关系, 生成的新关系中包含满足联接条件的元组。记作:

$$R \bowtie_{i\theta j} S = \sigma_{i\theta j}(R \times S)$$

式中, “ $\bowtie$ ” 为连接运算符, $i\theta j$ 是一个比较式, 其中 i, j 分别为 R 和 S 中的域, θ 为算术比较符。该式说明, R 与 S 的 θ 连接是 R 与 S 的笛卡儿积再加上限制 $i\theta j$ 而成, 显然, $R \bowtie_{i\theta j} S$ 中元组的个数远远少于 $R \times S$ 的元组个数。

连接运算有多种类型, 自然连接是最常用的连接运算。在连接运算中, 按关系的属性值对应相等为条件进行的连接操作称为等值连接, 自然连接是去掉重复属性的等值连接。

1.2.3 关系模型的完整性

关系模型的完整性规则是对关系的某种约束条件。关系模型有三类完整性约束: 实体完整性、参照完整性和用户定义的完整性。其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件, 被称作是关系的两个不变性, 应该由关系系统自动支持。

1. 实体完整性(Entity Integrity)

实体完整性规则要求表中的主键不能取空值或重复的值。例如, 在关系学生(学号, 姓名, 性别, 年龄, 专业, 班级)中, “学号”属性为主键, 则“学号”不能取空值, 也不能取重复值。

对于实体完整性规则说明如下:

(1) 实体完整性规则是针对基本关系而言的。一个基本表通常对应现实世界的一个实体集。例如, 学生关系对应于学生的集合。

(2) 现实世界中的实体是可区分的, 即它们具有某种唯一性标识。例如, 学生关系中的“学号”, 学生可以有重名的, 但学号没有重复的。相应地, 关系模型中以主码作为唯一性标识。

(3) 主码中的属性即主属性不能取空值。所谓空值就是“不知道”或“无意义”的值。

2. 参照完整性(Referential Integrity)

参照完整性规则就是定义外码与主码之间的引用规则。若属性(或属性组) F 是基本关系 R 的外码, 它与基本关系 S 的主码 K 相对应(基本关系 R 和 S 不一定是相同的表), 则 R 中的每个元组在 F 上的值必须为:

(1) 或者取空值(F 的每个属性值均为空值);

(2) 或者等于 S 中的某个元组的主码值。

3. 用户定义的完整性(User-defined Integrity)

任何关系数据库系统都应该支持实体完整性和参照完整性。除此之外, 不同的关系数据库系统根据其应用环境的不同, 往往还需要一些特殊的约束条件, 用户定义的完整性就是针对某一具体关系数据库的约束条件。它反映某一具体应用所涉及的数据必须满足的语义要求。

1.2.4 规范化理论

规范化理论研究的是关系模式中各属性之间的依赖关系及其对关系模式的影响。规范化理论给我们提供判断关系模式优劣的理论工具, 帮助我们预测可能出现的问题, 提供了自动产生各种模式的算法, 因此是设计人员的有力工具, 也使数据库设计有了严格的理论基础。