

iLike就业

WRO
PAT

SQL多功能教材

张 琨 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

iLike 就业 SQL 多功能教材

张 琨 编著

電 子 工 業 出 版 社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书全面介绍了 SQL 语言各方面的相关知识。全书共分为 9 课, 包括关系数据库和 SQL 概述; 数据库和表的操作; 数据的简单查询; 数据的查询进阶; 数据的插入、更新和修改; 视图与索引的操作; 存储过程与用户自定义函数; 触发器; 事务控制与并发处理。全书以 SQL 语句的综合应用实例为主线, 为读者全面地讲解了开发专业实例之前所应掌握的基础知识, 使其对 SQL 语句本身有一个比较深刻的理解和掌握, 并能够运用 SQL Server 数据库工具进行项目开发。

本书脉络清晰、深入浅出、涉及知识面广、实用性强, 非常适合数据库开发人员学习和使用, 也可以作为使用 SQL 的计算机专业或相关人员的参考手册。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

SQL 多功能教材 / 张琨编著. —北京: 电子工业出版社, 2013.1

(iLike 就业)

ISBN 978-7-121-18815-2

I. ①S… II. ①张… III. ①关系数据库系统—数据库管理系统—教材 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2012)第 254703 号

责任编辑: 戴 新

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

北京市海淀区翠微东里甲 2 号 邮编: 100036

开 本: 787×1092 1/16 印张: 14.75 字数: 384 千字

印 次: 2013 年 1 月第 1 次印刷

定 价: 32.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

SQL (Structured Query Language, 结构化查询语言) 是一种功能强大的数据库查询和程序设计语言, 用于存取数据及查询、更新和管理关系数据库系统。

SQL 是高级的非过程化编程语言, 是沟通数据库服务器和客户端的重要工具, 允许用户在高层数据结构上工作。它不要求用户指定对数据的存放方法, 也不需要用户了解具体的数据存放方式, 所以, 具有完全不同底层结构的不同数据库系统可以使用相同的 SQL 语言作为数据输入与管理的接口。SQL 语言集数据查询、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特点和优点, 它已成为关系数据库领域中的一种主流语言。

自 SQL 成为国际标准语言以来, 各个数据库厂家纷纷推出了各自支持 SQL 的软件或能够与 SQL 接口的软件, 这就使得大多数数据库使用 SQL 作为共同的数据库语言和标准接口成为可能, 不同数据库系统之间的相互操作也有了共同的基础。而且对数据库以外的领域也产生了很大影响, 有不少软件产品将 SQL 语言的数据查询功能与图形功能、软件工程工具、软件开发工具、人工智能程序结合起来。

全书共分为 9 课, 第 1 课介绍了关系数据库和 SQL 概述, 并着重介绍了 SQL Server 系列至今的发展及最新版本 SQL Server 2012 的特点及功能, 同时兼顾了该系列至今应用最广的 SQL Server 2008 和 2000 版本。全文实例均是在 SQL Server 2012 中的运行结果, 如无特别说明, 操作和代码在 SQL Server 2008 和 SQL Server 2000 中也都能执行。第 2 课介绍了 SQL 中常用的数据类型, 数据库和表的一些相关操作, 包括数据库的创建与删除, 表的创建、修改与删除。第 3 课和第 4 课介绍了与 SQL 查询 (SELECT) 语句相关的一系列的数据查询方法, 包括构成 SQL 的基础元素, 查询的基本结构及单列查询、简单搜索条件的查询、复杂搜索条件的查询、多表查询、聚合函数与组合查询等。第 5 课介绍了 SQL 中数据插入 (INSERT)、更新 (UPDATE) 与删除 (DELETE) 语句。第 6 课介绍了视图、索引的相关知识, 以及在 SQL 环境中创建、删除视图和索引的语法及实现实例。第 7 课介绍了 SQL 中的存储过程与函数。第 8 课介绍了 SQL 中触发器的有关概念和操作。第 9 课介绍了事务及其操作及并发处理的相关内容。这些内容使读者能熟练运用 SQL 语句, 并能够基于不同数据库工具熟练高效地进行项目开发。

本书以“实例”为主线, 用大量小例子作为全书主体结构进行功能和知识的讲解。本书可以作为学习和使用 SQL 的计算机专业或相关专业人员的参考手册, 使他们能够全面了解开发专业实例之前所应掌握的基本知识。

本书由张琨统稿, 贾若、王青和罗昕在本书的编写过程中亦提供了帮助。在本书的编写过程中, 尽管编者竭尽全力, 但由于自身水平有限和时间仓促, 书中难免有疏漏和不当之处, 敬请读者批评指正, 不胜感激!

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任 and 行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

目 录

第 1 课	关系型数据库与 SQL 概述	1
1.1	数据库系统概述	1
1.1.1	数据库系统的产生与发展	1
1.1.2	数据库的基本概念	3
1.1.3	数据库系统的组成	4
1.1.4	数据库系统体系结构	6
1.1.5	关系数据库	8
1.2	SQL 基础	10
1.2.1	SQL 的发展	10
1.2.2	SQL 的特点	11
1.2.3	SQL 数据库的体系结构	12
1.2.4	SQL 语言的组成	13
1.2.5	SQL 语句的结构	14
1.3	SQL 环境	14
1.3.1	环境介绍	14
1.3.2	SQL 的层次结构	15
1.4	SQL Server 2012 介绍	16
1.4.1	SQL Server 2012 概述	16
1.4.2	SQL Server 2012 的版本	16
1.4.3	SQL Server 2012 的特点	16
1.4.4	SQL Server 2012 的工具	17
1.5	本书使用的数据库和约定	24
第 2 课	数据库和表的操作	26
2.1	数据库操作实例	26
2.1.1	实例：数据库的创建	28
2.1.2	实例：数据库的删除	29
2.2	数据表操作实例	29
2.2.1	实例：表的创建	33
2.2.2	实例：表的修改	38
2.2.3	实例：表的删除	43
2.2.4	实例：表的重命名	44
第 3 课	数据的简单查询	45
3.1	列查询的实例	49
3.1.1	实例：单列查询	50
3.1.2	实例：多列查询	51
3.1.3	实例：查询所有的字段	51
3.1.4	实例：去除重复信息	52

3.2	查询结构排序的实例	53
3.2.1	实例：单列排序	53
3.2.2	实例：多列排序	54
3.3	简单查询的实例	55
3.3.1	实例：WHERE 子句单条件查询	55
3.3.2	实例：WHERE 子句运算符查询	56
3.3.3	实例：BETWEEN 子句条件查询	59
第4课	数据的查询进阶	60
4.1	复杂查询的实例	60
4.1.1	实例：使用 AND 和 OR 运算符	60
4.1.2	实例：使用 IN 运算符	63
4.1.3	实例：使用 NOT 运算符	64
4.1.4	实例：使用 LIKE 运算符	65
4.2	聚合函数与组合查询的实例	70
4.2.1	实例：聚合函数	70
4.2.2	实例：GROUP BY 子句	77
4.2.3	实例：HAVING 子句	78
4.3	多表查询的实例	78
4.3.1	实例：多表连接的目的	78
4.3.2	实例：简单的二表连接与多表连接	79
4.3.3	实例：多表连接的类型	82
4.3.4	实例：UNION 与 UNION JOIN 子句	89
第5课	数据的插入、更新和删除	92
5.1	数据插入的实例	92
5.1.1	实例：插入的基本语法	92
5.1.2	实例：整行的插入	92
5.1.3	实例：NULL 值的插入	94
5.1.4	实例：唯一值的插入	94
5.1.5	实例：特定字段的插入	95
5.1.6	实例：使用 VALUES 关键字	96
5.1.7	实例：使用 SELECT 语句	97
5.2	数据更新的实例	99
5.2.1	实例：UPDATE 语句的简单结构	99
5.2.2	实例：行值和列值的更新	100
5.2.3	实例：UPDATE 语句的复杂结构	103
5.2.4	实例：数据分步更新	104
5.3	数据删除的实例	106
5.3.1	实例：DELETE 语句的结构	106
5.3.2	实例：指定数据的删除	107
第6课	视图与索引的操作	110
6.1	视图操作的实例	113

6.1.1 实例: 视图的创建	114
6.1.2 实例: 视图数据的查询和修改	124
6.1.3 实例: 视图的删除	129
6.2 索引操作的实例	130
6.2.1 实例: 索引的创建	134
6.2.2 实例: 索引的删除	140
第 7 课 存储过程与用户自定义函数	142
7.1 存储过程创建和使用的实例	143
7.1.1 实例: 存储过程的创建和执行	144
7.1.2 实例: 存储过程的使用	158
7.2 用户自定义函数创建和使用的实例	162
7.2.1 实例: 用户自定义函数的创建和调用	163
7.2.2 实例: 操作用户自定义函数	167
第 8 课 触发器	169
8.1 触发器创建及操作的实例	170
8.1.1 实例: 创建触发器	170
8.1.2 实例: 触发器的操作	174
8.2 触发器应用的实例	176
8.2.1 实例: INSERT 触发器	177
8.2.2 实例: DELETE 触发器	180
8.2.3 实例: UPDATE 触发器	182
8.2.4 实例: INSTEAD OF 触发器	184
8.2.5 实例: 嵌套触发器	189
8.2.6 实例: 递归触发器	192
第 9 课 事务控制与并发处理	195
9.1 事务控制操作的实例	197
9.1.1 实例: 事务处理语句	197
9.1.2 实例: 启动事务	197
9.1.3 实例: 提交和回滚事务	201
9.1.4 实例: 事务的保存点	202
9.2 事务使用的实例	203
9.2.1 实例: 嵌套事务	203
9.2.2 实例: 批处理中的事务	205
9.2.3 实例: 存储过程和触发器中的事务	206
9.3 并发处理的实例	209
9.3.1 实例: 并发访问的问题	209
9.3.2 实例: 锁	210
9.3.3 实例: 事务的隔离级别	213
9.3.4 实例: 事务的阻塞及死锁	220

第1课

关系型数据库与 SQL 概述

数据库技术是信息系统的一个核心技术，是一种计算机辅助管理数据的方法。它研究如何组织和存储数据，如何高效地获取和处理数据。数据库技术不仅应用于事务处理，并且能够进一步应用到情报检索、人工智能、专家系统、计算机辅助设计等领域。本章将详细介绍有关数据库系统和 SQL 的基础知识。

1.1 数据库系统概述

数据库系统 (Database Systems) 是由数据库及其管理软件组成的系统，它是为适应数据处理的需要而发展起来的一种较为理想的数据处理的核心机构。数据库是一个实际可运行的为存储、维护和应用系统提供数据的软件系统，是存储介质、处理对象和管理系统的集合体，是为适应数据处理的需要而发展起来的一种较为理想的数据处理的核心机构。计算机的高速处理能力和大容量存储器提供了实现数据管理自动化的条件。

1.1.1 数据库系统的产生与发展

数据库技术起源于 20 世纪 60 年代末，在 70 年代初逐步形成，其主要目的是有效地管理和存储大量的数据资源，主要研究如何存储、使用和管理数据。近年来，数据库技术和计算机网络技术的发展相互渗透，相互促进，已成为当今计算机领域发展迅速、应用广泛的两大领域。在应用需求的推动下，在计算机硬件、软件发展的基础上，数据管理技术的发展大致经历了人工管理、文件系统和数据库系统三个阶段。

1. 人工管理阶段

20 世纪 50 年代以前，计算机主要用于数值计算。从当时的硬件看，外存只有纸带、卡片、磁带，没有直接存取设备，因此数据管理的工作由应用程序完成。在应用程序中不仅要规定数据的逻辑结构，还要设计数据的物理结构。从软件看，当时还未形成软件的整体概念，没有操作系统及管理数据的软件。从数据上看，这一阶段包含有四个特点。

- 数据不保存

应用程序所对应的数据不单独存在，在程序运行时由数据卡片或数据磁带输入，运算后直接取走结果，源数据与结果都不保存。

- 数据不共享

数据是面向应用的，一个数据集只能对应一个应用程序。应用程序之间不能共享数据。

- 数据不具有独立性

应用程序完全依赖于数据，不具有数据独立性，一旦数据的逻辑结构或物理结构发生变化，应用程序必须做相应的修改。

- 数据大量冗余

由于数据不能共享，因此即使两个应用程序使用到某些相同的数据，它们之间也无法相互利用对方的数据。

在人工管理数据阶段，程序与数据之间具有一一对应关系，如图 1.1 所示。

2. 文件系统阶段

20 世纪 50 年代后期到 60 年代中期，出现了磁鼓、磁盘等数据存储设备。新的数据处理系统迅速发展起来，大大增强了计算机的存取能力。软件方面，出现了操作系统，数据以文件的形式由操作系统的专门软件——文件管理系统进行统一管理。这种数据处理系统是把计算机中的数据组织成相互独立的数据文件，系统可以按照文件的名称对其进行访问，对文件中的记录进行存取，并实现对文件的修改、插入和删除，这就是文件系统。

文件系统阶段数据具有以下三个特点。

- 数据可以长期保存

数据统一由操作系统组织成数据文件，并长期保存在磁盘存储器上。用户通过文件管理系统可以很方便地对其进行访问与修改。

- 数据的独立性有一定的提高

由于文件的物理组织由操作系统完成，用户或者应用程序只需根据数据的逻辑结构，通过文件名进行访问，无须了解数据的物理结构，从而提高了数据的物理独立性。

- 由文件系统管理数据

文件管理系统还为用户提供了多种文件组织形式，如顺序文件组织、索引文件组织、直接文件存取组织等。

尽管文件管理系统的出现弥补了手工管理的一些缺陷，使得数据管理技术有了很大的提高，但是它在数据管理方面还存在一些不足：文件从整体来看仍是无结构的，其数据面向特定的应用程序，因此数据共享性、独立性还不够高；且由于数据冗余度大，导致管理和维护的代价也很大。

文件系统阶段应用程序与数据之间的关系，如图 1.2 所示。

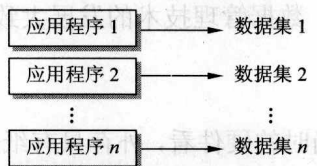


图 1.1 人工管理阶段应用程序与数据之间的关系

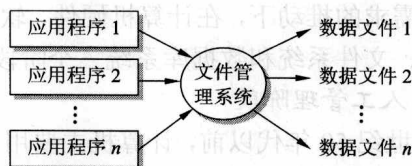


图 1.2 文件系统阶段应用程序与数据之间的关系

3. 数据库阶段

20 世纪 60 年代后期，由于计算机技术的迅速发展，磁盘存储技术大大提高，计算机更广泛地应用于管理领域。随着数据量的剧增，相关领域对数据管理提出了更高的要求——更高的独立性与共享性。文件管理技术已经不能适应上述要求。为了进一步减少数据冗余，满足多用户、多应用程序的数据独立与高度共享的需求，使数据为尽可能多的应用服务，统一管理数据的专门软件系统——数据库管理系统（Database Management System, DBMS）相应诞生了。

用数据库系统来管理数据比用文件系统具有明显的优势，从文件系统到数据库系统，标志

着数据管理技术的飞跃。这一阶段数据的特点有以下四个。

- 数据结构化

在文件系统中, 尽管记录内部有了某些结构, 但记录之间没有联系。而数据库系统则实现了整体数据的结构化, 这是数据库的主要特征之一, 也是数据库与文件系统的本质区别。

- 数据独立性高

数据库系统提供了数据的逻辑映射与物理映射, 这样, 数据库中的数据既有逻辑独立性, 又有物理独立性。用户只须通过简单的局部逻辑结构来操作数据, 无须考虑数据的全局逻辑结构及物理存储结构, 因此确保了较高的数据独立性。

- 数据的共享性高、冗余度低、易于扩充

数据的共享性直接影响到数据的冗余度。数据库中的数据是面向整个系统的, 而不是只面向某个应用, 它们可以被多个用户共享。这样既可以保证最小的数据冗余, 又可以避免数据的不相容性与不一致性。

- 数据由数据库管理系统统一管理和控制

在数据库系统中, 由数据库管理系统对数据进行统一的管理和控制。通过 DBMS 还可以保证数据库系统中数据的安全性、完整性, 并发访问控制和恢复。

数据库技术在数据管理中的特点与优势, 使得它在计算机数据管理中占据主导地位, 成为各类信息系统的核心基础。

数据库阶段应用程序与数据之间的关系, 如图 1.3 所示。

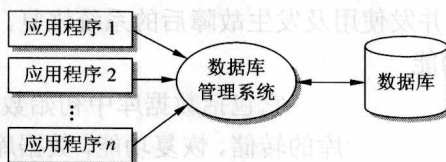


图 1.3 数据库阶段应用程序与数据之间的关系

1.1.2 数据库的基本概念

数据、数据库、数据库管理系统和数据库系统是与数据库技术密切相关的四个基本概念。

1. 数据 (Date)

数据是数据库的基本组成内容, 是对客观世界所存在的事物的一种表征, 人们总是尽可能地收集各种各样的数据, 然后对其进行加工处理, 从中抽取并推导出有价值的信息, 作为指导日常工作和辅助决策的依据。

数据是数据库中存储的基本对象。大多数人对于数据的第一反应就是数字。其实数字只是数据最简单的一种形式, 是对数据的一种传统和狭义的理解。广义来讲, 数据的种类很多, 文字、图形、图像、声音、学生的档案记录、货物的运输情况等, 这些都是数据。

这里可以对数据做如下定义: 描述事物的符号记录称为数据。描述事物的符号有多种表现形式, 它们都可以被数字化后存入计算机。

2. 数据库 (DataBase, 简称 DB)

数据库 (Database) 是按照数据结构来组织、存储和管理数据的仓库, 只不过这个仓库是位于计算机存储设备上的, 而且数据是按一定的格式进行存放的。

人们收集并抽取出一个应用所需要的大量数据之后, 应将其保存起来以供进一步加工处理, 进一步抽取有用信息。在科学技术飞速发展的今天, 人们的视野越来越广, 数据量急剧增

加。过去人们把数据存放在文件柜里,现在人们借助计算机和数据库技术科学地保存和管理大量的、复杂的数据,以便能方便而充分地利用这些宝贵的信息资源。

数据库是长期储存在计算机内的、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、描述和储存,具有较小的冗余度、较高的数据独立性和易扩展性,并可为各种用户共享。

3. 数据库管理系统 (DataBase Management System, 简称 DBMS)

在了解了数据和数据库的概念之后,就该讨论如何科学地组织和存储数据,又如何高效地获取和维护数据了。完成这些任务的是一个系统软件——数据库管理系统。

数据库管理系统是位于用户与操作系统之间的一种数据管理软件,主要包括以下几方面的功能。

● 数据定义功能

DBMS 提供数据定义语言 (Data Definition Language, 简称 DDL), 用户通过它可以方便地对数据库中的数据对象进行定义。

● 数据操纵功能

DBMS 还提供数据操纵语言 (Data Manipulation Language, 简称 DML), 用户可以使用 DML 操纵数据以实现对数据库的基本操作, 如查询、插入、删除和修改等。

● 数据库的运行安全

数据库在建立、运用和维护时由数据库管理系统统一管理、统一控制, 以保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复。

● 数据库的建立和维护功能

它包括数据库中初始数据的输入、转换功能, 数据库的转储、恢复功能, 数据库的重组功能 and 性能监视、分析功能等。这些功能通常是由一些实用程序运行完成的。

数据库管理系统是数据库系统的一个很重要的组成部分。

4. 数据库系统 (DataBase System, 简称 DBS)

数据库系统是指在计算机系统中引入数据库后的系统, 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。应当指出的是, 数据库的建立、使用和维护等工作只靠一个 DBMS 是远远不够, 还要有专门的人员来完成, 这些人被称为数据库管理员 (DataBase Administrator, 简称 DBA)。

在一般不引起混淆的情况下常常把数据库系统简称为数据库。数据库系统可以用图 1.4 来表示。

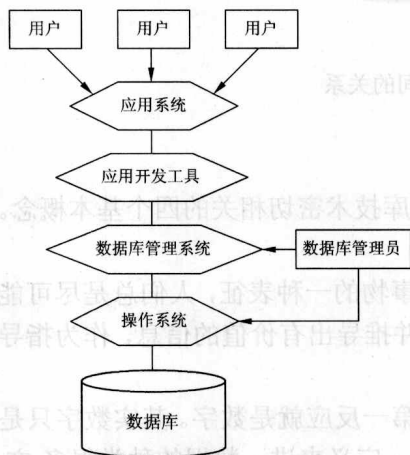


图 1.4 数据库系统

1.1.3 数据库系统的组成

数据库系统通常是指带有数据库的计算机应用系统, 因此, 数据库系统不仅包括数据库本身, 还包括相应的硬件、软件和各类人员。数据库系统是一个由使用人员和维护人员、加工设备和数据资源组成的完整的计算机应用系统。

1. 硬件平台

由于数据库系统数据量都很大,加之 DBMS 丰富的功能使得自身的规模也很大,因此整个数据库系统对硬件资源提出了较高的要求,包含以下三条。

- (1) 有足够大的内存,存放操作系统、DBMS 的核心模块、数据缓冲区和应用程序。
- (2) 有足够大的磁盘等直接存储设备存放数据库,有足够的磁带(或微机软盘)为数据做备份。
- (3) 要求系统有较强的通道传输能力,以提高数据传送效率。

2. 软件平台

数据库系统的软件主要包括以下五个因素。

- (1) DBMS,它是作为数据库的建立、使用和维护配置的软件。
 - (2) 支持 DBMS 运行的操作系统。
 - (3) 具有与数据库接口的高级语言及其编译系统,便于开发应用程序。
 - (4) 以 DBMS 为核心的应用开发工具。
- 应用开发工具是系统为应用开发人员和最终用户提供的高效率、多功能的应用生成器、第四代语言等各种软件工具。它们为数据库系统的开发和应用提供了良好的环境。
- (5) 为特定应用环境开发的数据库应用系统。

3. 人员

- (1) 数据库管理员 (DataBase Administrator, 简称 DBA)

在数据库系统环境下,有两类共享资源:一类是数据库,另一类是数据库管理系统软件。因此需要有专门的管理机构来监督和管理数据库系统。DBA 是这个机构的一个或一组人员,负责全面管理和控制数据库系统。DBA 的具体职责有以下几条。

- 决定数据库中的信息内容和结构
数据库中要存放哪些信息,DBA 要参与决策。因此 DBA 必须参与数据库设计的全过程,并与用户、应用程序员、系统分析员密切合作、共同协商,做好数据库设计。
 - 决定数据库的存储结构和存取策略
DBA 要综合各用户的应用要求,和数据库设计人员共同决定数据的存储结构和存取策略,以求获得较高的存取效率和存储空间利用率。
 - 定义数据的安全性要求和完整性约束条件
DBA 的重要职责是保证数据库的安全性和完整性,因此 DBA 负责确定各个用户对数据库的存取权限、数据的保密级别和完整性约束条件。
 - 监控数据库的使用和运行
DBA 还有一个重要职责就是监视数据库系统的运行情况,及时处理运行过程中出现的问题。比如系统发生各种故障时,数据库会因此遭到不同程度的破坏,DBA 必须在最短时间内将数据库恢复到正常状态,并尽可能不影响或减少对计算机系统其他部分的正常运行的影响。
 - 数据库的改进和重组重构
DBA 还负责在系统运行期间监视系统的空间利用率、处理效率等性能指标,对运行情况进行记录、统计和分析,依靠工作实践并根据实际应用环境,不断改进数据库设计。不少数据库产品都提供了对数据库运行状况进行监视和分析的实用程序,DBA 可以利用这些实用程序完成这项工作。
- 另外,在数据运行过程中,有大量数据不断插入、删除、修改,时间一长,会影响系统的

性能。因此, DBA 要定期对数据库进行重组织, 以提高系统的性能。

当用户的需求改变时, DBA 还要对数据库进行较大的改造, 包括修改部分设计, 即数据库的重构造。

(2) 系统分析员和数据库设计人员

系统分析员负责应用系统的需求分析和规范说明, 要和用户及 DBA 相合作, 确定系统的软硬件配置, 并参与数据库系统的概要设计。

数据库设计人员负责数据库中数据的确定、数据库各级模式的设计。数据库设计人员必须参加用户需求调查和系统分析, 然后进行数据库设计。在很多情况下, 数据库设计人员就由数据库管理员担任。

(3) 应用程序员

应用程序员负责设计和编写应用系统的程序模块, 并进行调试和安装。

(4) 用户

这里的用户是指最终用户 (End User)。最终用户通过应用系统的用户接口使用数据库。常用的接口方式有浏览器、菜单驱动、表格操作、图形显示、报表书写等, 这些接口方式给用户提供了简明直观的数据表示。

最终用户可以分为偶然用户、简单用户和复杂用户三类。

● 偶然用户

这类用户不经常访问数据库, 但每次访问数据库时往往需要不同的数据库信息, 一般是企业或组织机构的中高级管理人员。

● 简单用户

数据库的多数最终用户都是简单用户, 其主要工作是查询和修改数据库。他们一般都是通过应用程序员精心设计并具有友好界面的应用程序存取数据库。银行的职员、航空公司的机票预定工作人员、旅馆总台服务员等都属于这类用户。

● 复杂用户

复杂用户包括工程师、科学家、经济学家、科学技术工作者等具有较高科学技术背景的人员。这类用户一般都比较熟悉数据库管理系统的各种功能, 能够直接使用数据库语言访问数据库, 甚至能够基于数据库管理系统的 API (Application Programming Interface, 应用程序编程接口) 编写自己需要的应用程序。

1.1.4 数据库系统体系结构

从数据库管理系统的角度看, 数据库系统通常采用三级模式结构。数据库系统设计员可在视图层、逻辑层和物理层对数据进行抽取, 通过概念模式、外模式和内模式三级模式来描述不同层次上的数据特性。本节将介绍数据库系统的模式结构。

模式 (Schema) 是数据库中全体数据的逻辑结构和特征的描述, 它仅仅涉及到类型的描述, 而不涉及到具体的值。模式的一个具体值称为模式的一个实例 (Instance)。同一个模式可以有很多实例。模式是相对稳定的, 实例是相对变动的, 因为数据库中的数据总在不断地更新。模式反映的是数据的结构及其联系, 而实例反映的是数据库某一时刻的状态。数据库系统的三级模式结构如图 1.5 所示。

1. 模式

模式又称为概念模式, 是数据库全体数据的逻辑结构和特征的描述, 是所有用户的公共数

据视图。它不涉及到数据物理存储、访问技术等细节，是数据库的管理人员视图。概念模式的一个具体值称为模式的一个实例，实例反映的是数据库某一时刻的状态，所以是相对变动的。同一个模式可以有很多实例。

概念模式不仅要描述概念记录类型，还要描述记录间的联系和操作、数据的完整性、安全性等要求。概念模式既不涉及数据的物理存储细节和硬件环境，也与具体的应用程序，与所使用的应用程序开发工具以及程序设计语言无关。

2. 外模式

外模式又称为子模式或用户模式，是用户对于与数据库系统的接口，是数据库用户对于看见和使用的局部数据的逻辑结构和特征的描述。它由若干个外部记录类型组成，用户使用数据操纵语言 DML 对数据库进行操作，实际上是对外模式的外部记录进行操作。

外模式通常是模式的子集，一个数据库中可以有多个外模式。外模式是保证数据库安全性的一个有力措施，每个用户只能看见和访问到相应的外模式的数据，而看不见数据库中的其余数据。

DBMS 提供外模式描述语言（外模式 DDL）来严格地定义外模式。

3. 内模式

内模式又称为存储模式，它描述的是数据的物理结构和存储结构，是数据在数据库内部的表示方式。内模式定义所有的内部记录类型、索引和文件的组织方式，以及数据控制方面的细节。它是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式。一个数据库只能有一个内模式。

4. 两级映像与数据独立性

数据库系统的三级模式将数据划分为三个抽象级别，它把数据的具体组织工作留给了 DBMS 管理，使用户能够从逻辑层面上处理数据，而不必关心数据在计算机中的具体表示方式和存储方式。为了能够在内部实现这三个抽象层次的联系和转换，DBMS 在这个三级模式之间提供了两级映像。

● 模式/内模式映射

内模式到概念模式之间的映像提供了数据的物理独立性。数据的物理独立性是指当数据库的内模式发生改变时，数据的逻辑结构不变。当数据的物理结构发生了变化，只需改变模式/内模式映像，而使概念模式尽可能保持不变。也就是说数据的内模式改变，而模式可以不变，因而应用程序也不改变，从而实现了数据的物理独立。

● 概念模式/外模式映射

概念模式到外模式之间的映射提供了数据逻辑独立性。数据的逻辑独立性是指用户的应用程序与数据库的逻辑结构之间是相互独立的。数据的逻辑结构发生变化后，用户程序可以不变。如果要对数据库的概念模式进行修改，那么模式/外模式映射也要做相应的修改，但外模式很可能保持不变，即实现了数据的逻辑独立。

数据库的这种多层次的体系结构，提供了高度的数据独立性。其中数据库的全局逻辑描述

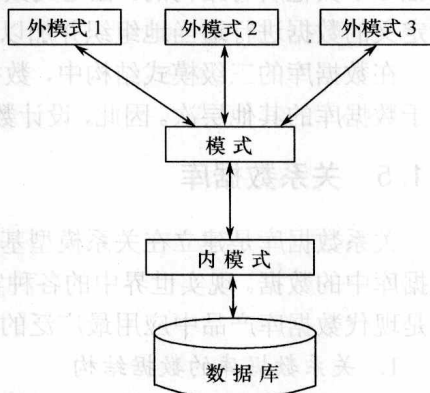


图 1.5 数据库系统的三级模式结构

是独立于其他所有结构的。在定义数据库结构时,应首先定义概念模式。内模式是将概念模式中定义的数据进行适当地组织并加以存储,以达到较好的运行效率。

在数据库的三级模式结构中,数据库模式,即全局逻辑模式是数据库的中心与关键,它独立于数据库的其他层次。因此,设计数据库系统的模式结构时,应首先确定数据库的逻辑模式。

1.1.5 关系数据库

关系数据库是建立在关系模型基础上的数据库,借助于集合代数等数学概念和方法来处理数据库中的数据。现实世界中的各种实体及实体之间的各种联系均用关系模型来表示。关系模型是现代数据库产品中应用最广泛的模型,而且正是关系模型构成了 SQL 的基础。

1. 关系数据库的数据结构

关系模型是建立在严格的数学概念的基础上的。依据用户的观点,关系模型中数据的逻辑结构是一张二维表,它由行和列组成。这里介绍一下关系模型中的一些术语。

● 关系 (Relation)

一个关系通常用一张表来表示。

● 元组 (Tuple)

表中的一行即为一个元组。

● 属性 (Attribute)

表中的一列即为一个属性,每一个属性的名称即为属性名。

● 主码 (Key)

表中的某个属性组,它可以唯一确定一个元组,也就成为了本关系的主码。

● 域 (Domain)

属性的取值范围。例如人的年龄一般在 1~150 岁之间,大学生年龄属性的域是 (14~38),性别的域是 (男,女),系别的域是一个学校所有系名的集合。

● 分量

元组中的一个属性值。

● 关系模式

对关系的描述,一般表示为:

关系名 (属性 1, 属性 2, ..., 属性 n.)

关系模型要求关系必须是规范化的,即要求关系必须满足一定的规范条件,这些规范条件中最基本的一条就是,关系的每一个分量必须是一个不可分的数据项,也就是说,不允许表中还有表。

2. 关系数据库的模式及完整性约束

(1) 关系数据库模式

关系数据库模式是关系数据库结构的描述,它包括若干域的定义及在这些域上定义的若干关系模式。关系模式对应于程序设计语言中定义类型的概念。

关系的描述称为关系模式,可以形式化地表示为 $R(U, D, \text{dom}, F)$, 其中 R 表示关系名, U 是组成该关系的属性名集合, D 是属性的域, dom 是属性向域的映像集合, F 为属性间数据的函数依赖关系集合。通常关系模式简记为 $R(U)$ 。

(2) 函数依赖

若对 $R(U)$ 的任意一个可能的关系 R , R 中不可能存在两个元组在 X 上的属性值相等,

而在Y上属性值不等,则称“X函数决定Y”,或称“Y函数依赖于X”,记作 $X \rightarrow Y$,X称为决定元素。

在R(U)中,如果 $X \rightarrow Y$,并且对于X的任何一个真子集X'都能函数决定Y,则称“Y完全函数依赖于X”。

在R(U)中,如果 $X \rightarrow Y$,但不完全依赖于X,则称“Y对X部分函数依赖”。函数依赖必须满足以下三个性质。

自反律:如果B是A的子集,则 $A \rightarrow B$ (平凡依赖)。

增广律:如果 $A \rightarrow B$,则 $AC \rightarrow BC$ 。

传递律:若 $A \rightarrow B$ 且 $B \rightarrow C$,则 $A \rightarrow C$ 。

上述规则是三条基本定理,由此还可以得到如下推论。

自含律: $A \rightarrow A$ 。

分解律: $A \rightarrow BC$,则 $A \rightarrow B$,且 $A \rightarrow C$ 。

合并律: $A \rightarrow B$,且 $A \rightarrow C$,则 $A \rightarrow BC$ 。

复合律: $A \rightarrow B$,且 $C \rightarrow D$,则 $AC \rightarrow BD$ 。

(3) 关系模式的规范化

● 第一范式

如果关系模式R中每一属性都是不可分的基本数据项,则R属于1NF。第一范式是对关系模式的最起码的要求,不满足第一范式的数据库模式不能称为关系数据库。

● 第二范式

如果关系模式R属于1NF,并且每一个非主属性都完全函数依赖于键码,则R属于2NF。

● 第三范式

如果关系模式R属于2NF,且不存在非主属性传递依赖于其他的非关键字属性,则R属于3NF。

● BCNF 范式

BCNF格式又称改进的第三范式,如果关系模式R属于3NF,并且没有一个非主属性是传递函数依赖于其他的非关键字属性,则R属于BCNF。

● 第四范式

如果关系模式R属于3NF,且没有非主属性且非函数依赖的多值依赖,则R属于4NF。

在以上几种范式之间,4NF包含BCNF,而BCNF又包含3NF。即如果一组元组满足4NF条件,那么它必然满足BCNF和3NF两个范式条件;而且如果它满足BCNF条件,那么它必然属于3NF。

在关系数据库中,对关系模式的基本要求是满足第一范式,寻求解决关系模式存在的插入、删除异常,数据冗余,以及修改等复杂的问题,这就是规范化的目的。

(4) 完整性约束

完整性规则是保证数据一致性的一种重要手段,可防止对数据的恶意破坏。关系模型的完整性规则是对关系的某种约束条件,关系的完整性约束分为三类:实体完整性、参照完整性和用户定义完整性。

● 实体完整性

指的是基本关系的主属性不能取空值。

● 参照完整性