第一章　关系数据库系统的基本原理

为了使读者对关系数据库系统有个基本了解，也为了统一对数据库系统中一些名词、概念的理解，我们首先简要介绍一下关系数据库系统的基本原理。

第一节　关系数据库的基本概念

在日常生活中，有各种各样的库，例如书库、金库、血库以及存放一般东西的仓库。这些库都有两个特点：一是可以存放现实世界中一定的东西，二是这些东西是有条理有组织地存放在库中的。过去，各种数据资料都是以文字形式保存在纸上，例如各种帐本，银行储蓄户卡片，各种表报、公文和人事档案等。自从电子数字计算机问世以来，人们逐渐认识到计算机除了科学计算外，还有强大的存储和处理数据的能力。计算机的外存贮器是保存数据资料的一个好地方。

数据库就是在计算机外存中有组织地保存大量数据信息的“仓库”。“库”中的这些数据都是现实世界中某个方面的信息的集合，可供许多人，甚至是不懂计算机的人共同使用。

研究如何更好地组织和管理数据库，如何使数据库更方便、更有效地为用户服务就构成了计算机科学技术的一个分支——数据库系统的研究内容。

数据库为反映现实世界而采用的若干种不同的组织方法称为数据模型。现在流行的有关系型、网状型、层次型和面向对象型。其中关系型数据库以其理论基础扎实，表现直观，使用灵活而得到广泛应用，商品化的产品也最多。主要有 ORACLE，FOXBASE，FOXPRO，INGRES，INFOMIX，DB Ⅱ，SYSTEMR 和 Rdb 等。

人们往往用表格来表示所关心的现实世界中的信息。关系数据库正是利用类似于表格的“关系”(relation)来表示信息的。例如，我们可以将一个工厂里各车间的情况列成一个表格，如表 1-1。

表 1-1　工厂车间表

车间号	车间名	车间主任	电话号
101	总　装	李××	6101
102	电　器	张　×	6102
103	钣　金	宋××	6103
104	机　加	杨××	6104
105	热处理	刘××	6105
⋮	⋮	⋮	⋮

这个表格的格式由(车间号，车间名，车间主任，电话号〉组成，称为这个关系的“关系型”或“关系框架”，也有的直接将关系称为表(table)，表的格式就称为表的结构(structure)。其中车间号、车间名、电话号等称为记录中的“属性名”(attribute)，也有的称

需要全本请访问 www.ertongbook.com

为字段或域。每个属性都表示关系所反映的客观对象的一个性质，它只能在一定范围内取值。而这个取值的范围称为这个属性的“取值域”(domain)。不同的属性可在同一个取值域内取值，这些值只能是一个确定的数据类型，如电话号码只能是正整数，车间名为字母开头的字母数字串。

〈101,总装,李××,6101〉分别为车间号、车间名、车间主任、电话号各属性的值。在不混淆的场合属性的值也可简称为属性。表格中的一行是该“记录型”的一个“记录值”,简称为一个记录,也称为这个“关系”的一个“记录”。如〈102,电器,张×,6102〉以及〈105,热处理,刘××,6105〉等均为该关系中的记录，这样的记录也称为元组（tuple)。每个记录反映了工厂中一个车间的情况。在诸属性中应有一个属性（或若干个属性的联合）的值唯一标识一个记录，称这样的属性（或属性的联合）为这个关系的“关键字”(key)。在上面谈到的关系中的车间号即为其关键字。

总之，一个关系型由若干属性（名）组成，每个属性为一定的数据类型，其值取自一定的值域。

一个关系由若干记录组成，每个记录由一组属性值组成。当把一个关系（即其所有的记录）保存到计算机的辅助存贮设备上时，就构成一个文件(file),该文件有一文件名,即为其关系名。文件通过操作系统存取。

一个关系也可以看成是若干记录的集合，该集合的元素就是记录。因为集合中的元素都是各不相同的，是无序的，所以一个关系中不允许有重复的记录，记录之间被看作是无顺序的。

客观世界往往需要用若干个关系联起来才能反映所研究的问题，这些关系之间通过公共的属性名而互有联系。相互联系起来反映所研究的问题的各关系的全体称为一个关系数据库。也有的为省事，把一个关系称作一个数据库，建立一个关系称为建库。

关系数据库是通过类似于表格的方式来组织数据记录，从而形成一个数据库的。我们把数据库系统中实体与实体之间的联系方式，也就是记录类型之间、记录之间的联系方式称为数据库的数据模型。关系数据库的数据模型是关系型的。除关系型外，传统数据库的数据模型还有层次型和网状型。

第二节　数据库管理系统

如果要用好数据库中的数据，就需要有一组程序来管理这些数据。这组程序被称为数据库管理系统,简称 DBMS。

一、DBMS 的功能

DBMS 具有如下的功能：

(1) 描述记录类型及其相互间联系的能力；

(2) 有效地利用存贮空间，以存贮更多的信息；

(3) 为用户提供对数据库进行查询、增加记录、修改记录和删除记录的能力；

(4) 使用户方便、容易地使用数据库，更快地得到所需数据；

(5) 保证数据库的完整、安全与保密，一旦遭受破坏或发生故障，有能力及时恢复，尽

量减少损失；

（6）在网络环境下，能在操作系统配合下完成数据通讯。

二、DBMS的组成

DBMS包含的主要程序有：

（1）语言处理程序。包括数据描述语言（DDL）、数据操纵语言（DML）、命令解释等处理程序。

（2）系统运行控制程序。包括总控，并发控制，安全保密控制，数据完整性控制，通讯控制，查、删、改、增加数据的实施等程序。

（3）系统维护程序。

三、数据字典

数据字典描述了一个数据库系统的总体情况，它是理解具体数据库内容的指南。其大体有：

（1）关于关系及其属性的定义情况；

（2）关于用户的情况；

（3）数据文件的存贮登记及存取权限信息；

（4）系统运行日志，在各检查点保留系统运行的主要信息。

四、数据库管理员（简称 DBA）

数据库管理员的主要工作职责为：

（1）决定数据库的信息内容。

（2）充当数据库系统与用户的联络员。

（3）决定数据存贮结构和访问策略。

（4）决定数据库的保护策略。

（5）监视系统工作，改善系统性能，提高系统效率。

五、数据库的结构

在数据库系统中，用户看到的数据与计算机实际存放的数据的形式与结构是不同的，其实际上分了三个层次：第一个层次是用户直接能看到的数据及其格式，称为子模式或外模式，也称作视图。第二层是系统为了减少冗余，实现数据共享，把所有用户要求的数据进行综合、抽象成一个统一的数据格式，称为概念模式。概念模式也是数据库系统对所反映的客观世界的抽象认识与结果。第三层是为了提高存取效率，改善系统性能，方便实际存放数据而采用的数据存贮结构，称为内模式或物理模式。

外模式是针对用户的，不同的用户可能有不同的外模式。全系统有一个统一的概念模式，所有外模式都是从概念模式派生出来的。

第三节　数据库查询语言

数据库系统必须为用户提供某种手段来建立自己的数据库，并在这个数据库上进行查询、修改、增加、删除等操作。这种手段就是数据库查询语言。前者称为“数据定义语言”（DDL），后者称为“数据操纵语言”（DML）。与高级程序设计语言是用户与计算机的界

面类似，数据库语言是用户与 DBMS 间的界面。

无论是 DDL 还是 DML ,与 FORTRAN,PASCAL,C 等语言相比，其计算功能还很不完善，因此它们常常被嵌入其他高级语言（如 C 语言中,此时 C 语言称为主语言),这样组成的数据库语言称为宿主语言。

关系数据库系统的语言都是根据在关系上的运算操作而设计的。表示关系上的操作可以有多种方法，不同的方法基于不同的数学理论，即基于集合代数的观点形成关系代数和基于谓词逻辑的观点形成关系演算。其共同点是通过对已存在的关系进行运算产生的结果数据仍是一个关系。

关系代数的观点是把关系看成集合，关系的一个记录为集合的一个元素。对关系的运算用集合的操作符来表示,如关系间的交、并、差以及笛卡儿积等。此外,为了方便,还专门定义一些关系运算，主要有：

选择：$\sigma_F(R)$表示,在一个关系 R 中选取那些满足一定限制条件下的记录，从而形成一个新关系，此新关系就可表示成 $\sigma_F(R)$。

投影：用 $\pi_{A_1\cdots A_n}(R)$表示,在一个关系 R 的各记录中选出属性名为 $A_1,\cdots,A_n$的那些属性的值,形成一个新关系,这个新关系就表示成 $\pi_{A_1\cdots A_n}(R)$。

联接：用 $R\underset{F}{\bowtie}S$ 表示,将关系 R 和 S 的笛卡儿积中选出满足条件 F 的那些元组，去掉重复元组,形成一个新关系。最常用的是‘自然联接”,即关系 R 和 S 有公共属性 B,在两关系中把该属性值对应相等的记录联接起来，形成新的记录，去掉重复记录，形成新的关系。

对数据库的查询,就可以用交、并、差、笛卡儿积、选择、投影和联接等操作联合起来表示，形成一个关系代数表达式。一个关系代数表达式就表示了对数据库的一个查询。

关系演算是把一阶谓词演算的理论应用到关系中，语句的表达方式采用了谓词公式。这种语句的一般形式为

操作动词（目标属性表）目标的描述：

操作动词一般有 GET,PUT,DELETE 及 UPDATE 等。目标属性表指出要取哪些关系中的哪些属性的值。目标的描述是一个谓词公式，说明所取属性值应满足的条件。谓词公式采用元组变量或属性变量表示。元组变量的定义域为一个关系，取值为该关系中的记录。属性变量的定义域为一个关系中的一个属性，取值为该属性的在该关系中的属性值。

最著名的关系数据库查询语言是 SQL(Structured Query Language),它已成为国际的工业标准。ORACLE 关系数据库系统的查询语言就是 SQL。

第二章 ORACLE 系统概述

第一节 ORACLE 系统概况

ORACLE 系统是目前世界上较为流行的关系数据库管理系统，它是美国 ORACLE 公司推出的 DBMS 产品。DRACLE 公司是一家著名的专门从事数据库技术研究和开发的计算机软件生产厂家。自 1977 年创建以来，先后推出了多种版本的 ORACLE 产品。1986 年推出的 ORACLE V5.1 是一个具有分布处理功能的数据库管理系统；1988 年底推出的第六版则向全关系系统迈进了一步，同时加强了事务处理的功能；1992 年又推出了第七版,使多媒体技术在 ORACLE 数据库中得到了充分的应用。经过十多年的不断完善和扩充,ORACLE 产品已经发展成为性能优良、功能强大的大型组合软件。

目前 ORACLE 产品覆盖了大、中、小型机、微机等几十种机型,成为世界上使用最广泛的性能优良的数据库管理系统。ORACLE 产品一进入中国市场，就迅速为中国用户所接受。同时美国 ORACLE （中国）有限公司在与有关单位合作下推出了 5.1 版的汉化产品,为 ORACLE 产品在中国推广起到了积极的作用。

一、ORACLE 系统特点

ORACLE 关系数据库系统是目前比较成功的数据库产品。它之所以倍受用户的喜爱是因为它有下列特点：

(1) 硬件环境独立性

硬件环境独立性是 ORACLE 系统独有的特点。目前 ORACLE 系统可以运行在不同厂家的不同档次的计算机上，如 IBM,DEC,AT&T,SUN 等著名厂家的多种机型，其中有大型机、小型机及微型机。

(2) 软件环境独立性

众所周知,DBMS 是在操作系统的支持下工作的。而 ORACLE 对系统提供的软件环境要求相当宽松，也就是说可以在不同的操作系统支持下运行。例如，在 VM/CMS, MVS,VMS,UNIX,DOS 上均能有效的运行 ORACLE。

ORACLE 的硬件环境独立性和软件环境独立性对用户是很有吸引力的。这种独立性是通过采用 C 语言来开发 ORACLE 的全部产品而获得的。目前适用于 ORACLE 的使用环境如表 2-1 所示。

需要全本请访问 www.ertongbook.com

表 2-1 可运行 ORACLE 的机型与系统

	厂 家	型 号	操作系统
大型机	IBM	370,43xx,30xx	VM/CMS,MVS,UTS
	Amdohl	470,580	VM/CMS,MVS,UTS
	Sperry	1100	UNIX
小型机	DEC	VAX-11,8200,8500,8700,8800,PDP-11	VMS, UNIX, ULTRIX, RSX-11M
	DG	MV 系列	AOS/VS,DG/VX
	HP	9000	HP/UX
	AT&T	3B5,3B12,3B15	UNIX
	APOLLO	Domain	AEGIS
	HARRIS	700,800,1000	VOS
	HONEYWELL	DPS-6,7,8	GCOS
	PRIME	SERIES 50	PRJMOS
	STRTVS	32	VOS
	Sperry	5000,7000	UNIX
	WANG	VS 系列	VS
	SUN	所有型号	UNIX
微型机	IBM	PC/XT,AT,RT	DOS,XENIX,AIX
	AT&T	6300,7300,3B2	DOS,UNIX
	DEC	RAINBOW,MicroVAX	DOS,VMS
	MOTOROLA	6300,6600,2000	UNIX
	NCR	TOWER	UNIX
	TI	Professional	DOS
	WANG	PC	DOS

(3) 兼容性

由于 ORACLE 采用了 SQL 语言，而且在功能上又进行了许多扩充，从而使 ORACLE 与 SQL/DB 和 DB I 等大型关系数据库完全兼容，即可以直接使用现有的数据库及其应用程序。

(4)高性能的 RDBMS

为了克服关系数据库效率低的弱点，ORACLE 系统在实现中引入了多种优化技术、索引技术和簇技术，从而使 ORACLE 的性能有很大的提高。

(5) 网络通讯与分布处理

ORACLE 系统具有网络通讯能力，并适应于多种网络协议，支持分布于不同节点之间的数据共享与分布查询。ORACLE 的第五版 SQL * STAR 包括三个主要特征，即存贮地址的独立性、网络独立性和 DBMS 独立性。它不仅提供了分布式 RDBMS的功能,而且采用了开放式体系结构，支持多种硬件及操作系统环境，多种通讯协议和多种 DBMS 之间的互连，从而为建立分布式信息系统提供了强有力的支持。

(6) 丰富的软件工具

除基本系统 RDBMS 之外,ORACLE 还包括一组功能很强的软件工具。即为应用开发人员提供应用生成器，菜单管理，报表生成，电子表格接口，电子图形软件等一批第四代

语言工具。

二、ORACLE 产品结构

ORACLE 数据库系统包括以 ORACLE RDBMS 为核心的一批软件产品，这些软件产品合起来被称为 ORACLE 数据库管理系统。许多用户不需要也不会使用所有的 ORACLE 产品，但用户可以利用不同的产品采用不同的方法来完成各种各样的任务。图 2-1 给出了 ORACLE 系统的基本轮廓。

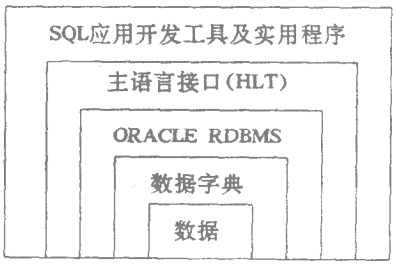


图 2-1 ORACLE 系统轮郭

(一) ORACLE RDBMS

ORACLE RDBMS 是 ORACLE 产品的核心，通常称之为 ORACLE 基本系统或核心系统。它包括核心数据库管理，以及帮助用户和数据库管理员（DBA)维护、监督和使用数据库的一些功能，由核心数据库管理模块和一些实用程序组成。核心数据库管理模块负责解释执行所有的 SQL 操作,实施数据管理和安全性、完整性、并发性控制。

ORACLE RDBMS 实用程序分为两类；一类是 DBA 实用程序 SQL * DBA ;另一类为用户实用程序：EXPORT/IMPORT, SQL * LOADER, CRT 和 ORACLE * TERMINAL。下面分别作一下简述：

(1) SQL * DBA是帮助 DBA 管理、控制和监督 ORACLE 数据库运转的工具。

(2) EXPORT/IMPORT 是完成 ORACLE 数据库的御出和装入的程序。

(3) SQL * LOADER 是将外部文件中的数据加载到 ORACLE 数据库中的程序。

(4) CRT 和 ORACLE * TERMINAL 。这两个实用程序均是给 ORACLE 支持的全屏幕软件工具产品定义终端显示特性，它们的差异仅在于其使用范围。例如前者适用于 SQL * FORMS 2.3 及以下版本，而后者适用于 SQL * FORMS 3.0 及以上版本。

(二) ORACLE 软件产品

ORACLE 软件工具给用户提供了一个友好的第四代开发环境。图 2-2 给出了 ORACLE 提供的软件产品。

由图可知,ORACLE 软件产品主要有下列四类：

1. EASY 产品系列

EASY * SQL 为友好的数据库用户接口。用户无需了解 SQL 语言，即可按照 EASY * SQL提供的全屏幕表格、菜单、窗口等提示实现对数据库的操作，并可生成数据库报表。

2. SQL 产品系列

(1) SQL * PLUS 为交互式命令接口。它可使用户在终端上以交互方式使用 SQL 语

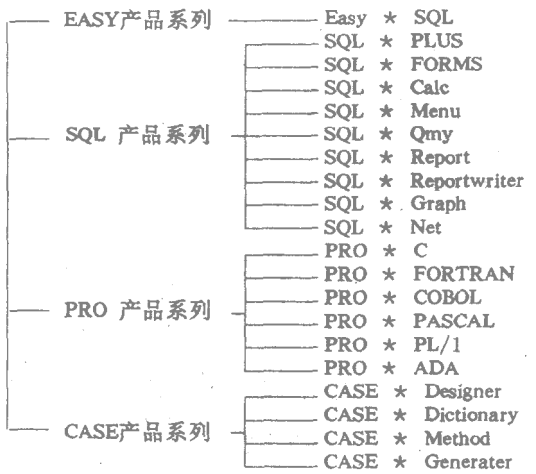


图 2-2 ORACLE软件产品构成

言来操纵数据库。

(2) SQL * FORMS 为第四代应用开发工具。它能使用户利用其全屏幕功能，按照应用原型来实现对数据库的各种存取和操纵。

(3) SQL * CALC 是 ORACLE 决策支持工具。通过将 CALS 子报表同 ORACLE 数据库紧密结合于一体,从而完成各种复杂的数据计算、分析、预测工作。

(4) SQL * MENU 为自动菜单生成系统。它提供用户设计 ORACLE 应用和其它产品的用户菜单驱动接口，使用户无须编程即可生成所需的菜单提示系统。

(5) SQL * QMX 为示例查询接口(QBE)。其采用二维表格作为用户界面来接受操作命令和显示结果。

(6) SQL * REPORT 为报表生成工具。将正文格式化能力与 SQL 查询能力相结合。

(7) SQL * REPORT WRITER 是第四代报表生成工具。它采用友好的用户界面辅助用户自动生成各种数据库报表，并可以同生成的菜单系统相关联，形成用户的应用系统。

(8) SQL * GRAPH 为 ORACLE 图形接口。它能够把对数据库的查询结果转换成圆饼图、直方图、曲线图或射线图。

(9) SQL * NET 是支持网络环境中数据库的分布式查询操作。

3. PRO 产品系列

利用 PRO 系列（预编译）产品，可在传统的高级程序设计语言所编写的应用程序中，通过嵌入 SQL 语句，方便灵活地访问和操纵数据库中的数据。PRO 产品主要有：

PRO * C,PRO * FORTRAN,PRO * COBOL ,PRO * PASCAL,PRO * PL/1,PRO * ADA。

4. CASE 产品系列

CASE 产品是提供给信息系统设计开发人员的。在 CASE 环境下,设计人员只需通过交互的表格和图形界面，提出系统设计要求，即可完成系统的开发任务。系统自动为其产生系统字典和有关文档。CASE 的目的就在于为系统的设计、实现和维护提供一个有效的

控制环境。

(1) CASE * DESIGNER 为 CASE 环境的图形接口，是多任务和多窗口的工作平台。

(2) CASE * DICTIONARY 是 CASE 环境的核心，其记录着开发项目期间所收集的各项信息，即：

①业务功能、事件、目标、实体、属性、域、关系、容量、频率、业务单位和数据流程；

②实现阶段将被翻译成程序模块、表和视图设计、索引和详细的量化信息。

(3) CASE * METHOD 为 CASE 环境的设计方法。其采用“自顶向下”的结构化方法，围绕着业务目标、任务和信息，使用图表表示应用开发中的定义说明。

(4) CASE * GENERATOR 是 CASE 环境的应用生成器。它将 CASE 字典中记录的设计定义自动生成先进的应用软件。

(三) ORACLE V6 对 ORACLE V5 的改变

从体系结构来看，ORACLE V6 对 ORACLE V5 进行了根本性的改造，从而大幅度地提高了事务处理能力。归纳一下主要有：

(1) 事务提交时，延迟写操作，从而提高了事务处理的性能。

(2) 由单数据库系统变为多数据库系统，因此有了创建数据库的概念，相应地数据库的存贮结构和空间分配方式也有改进。使用户和 DBA 对存贮空间的利用更为灵活。

(3) 有联机备份和恢复功能，日志登录形式也有改善。

(4) 增加了系统容错能力。

(5) 具有保存点(savepoints) 和语句级回滚功能。

(6) 改进了数据库管理工具 SQL * DBA。

(7) EXPORT 充了增量转贮的功能。

(8) 为了增加事务处理的并发度，增加了行级封锁功能。

(9) 提供了 SQL 过程语言 PL/SQL。PL/SQL 是对 SQL 的扩充，它使非过程的 SQL 语言具有了过程化的结构语言特色。

(1) ORACLE V6 中的 SQL * DBA 实用程序概括了 ORACLE V5 中的 IOR，ODS，AIJ，SGI，CCF 等实用程序的功能。

(2) ORACLE V6 中的 SQL * LOADER 实用程序取代了 ORACLE V5 中的 ODL 实用程序。

(3) ORACLE V6 的数据字典强化了 ORACLE V5 的数据字典，从而适应 ANSI/ISO SQL 标准和方便用户使用。

(4) ORACLE V6 对存贮结构做了如下的调整：系统中可用多个数据库(DB)，每个数据库可划分为多个表空间 (tablespace)，每个表空间可存放多个表，一个表对应一个数据段和多个索引段。索引段的存放更加灵活，一个索引段只存放一个索引，但索引段可不只限于表所在的表空间。

与五版的存贮结构相比较，它们有如下的关系：六版中的表空间对应于五版的分区，两种版本下的数据库文件都是一致的。六版的控制文件在五版中没有，是新增的文件。REDO 日志文件大致与五版的 AIJ 文件作用相同，但六版下的日志文件是必选的，而五版的 AIJ 文件是可选项；六版的回退段对应五版的 BI 文件，临时段对应五版的临时文件。

(5) 事务控制方式。六版中，在事务控制方面增加了如下功能；

① COMMIT 和 ROLLBACK 改为 SQL 语句，在五版中，它们是 SQL * PLUS 中的命令。

②增加了 SAVEPOINT 语句，用户可在事务中设置多个保存点，以减轻回滚负担。

③六版中隐式回滚是语句级的，五版是基于事务级的回滚。

④ 可用 SET TRANSACTION READ ONLY 设置只读事务，以保证一事务中多个读语句的一致性，即满足可重复读。

(6) 六版中提供了真正的行级封锁，从而提高了用户更新数据库的并行度。

第二节 ORACLE 体系结构

ORACLE 数据库系统包括两个主要方面：其一是 ORACLE 数据库，其二是 ORACLE 例程。图 2-3 为 ORACLE 数据库系统。图中：

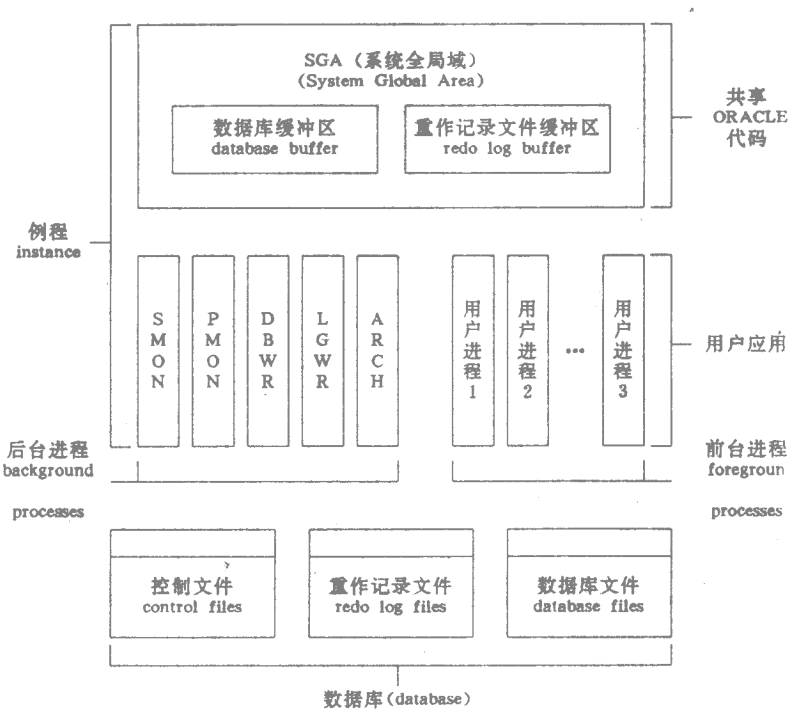


图 2-3 ORACLE 数据库系统

(1) 数据库(database) 是作为一个整体看待的数据集合。物理上，数据库是由数据库文件(database files)、控制文件(control files) 和重作记录文件(redo log files)等(操作系统文件)组成 逻辑上,数据库是由数据字典和用户表及其恢复数据组成。

(2)例程(instance)是存取和控制数据库的软件机制，由 ORACLE 共享代码、SGA 和一组后台进程组成。由此可见，例程提供了存取数据库的手段,它是指为了运行和存取

一个 ORACLE 数据库所要求的一组进程和内存空间（SGA）。

注意，例程和数据库彼此独立，一个例程每次仅能与一个数据库相联接。

一、ORACLE RDBMS 文件结构

ORACLE 数据库系统的运行要求四类操作系统文件：

（1）ORACLE RDBMS 执行代码文件　诸多操作系统文件（目标库文件、执行代码文件和命令文件）所需的空间将依赖于操作系统。

（2）数据库文件(database files) 至少一个操作系统文件，其大小起码应是 500KB。

（3）控制文件(control files) 至少一个操作系统文件，其大小根据安装时对数据库文件和重作记录文件的配置所决定。

（4）重作记录文件(redo log files) 至少两个操作系统文件，其大小起码应是 50KB。

（一）数据库文件

一个 ORACLE 数据库可由一个或多个数据库文件组成，其具备如下特性：

（1）一个数据库文件仅能与一个数据库相关。

（2）一个数据库文件的尺寸不得重新改变。

（3）一个数据库文件在磁盘上连续分配时性能最优。

（4）一个或多个物理文件构成数据库存贮的逻辑单位——表空间。

（5）例程被启动时，一个数据库中的联机表空间的全部数据库文件必须均是可存取的 。

1. 数据库文件的数目

（1）数据库文件数目的绝对最大值能够利用 SQL 语句来设置（缺省值为 255）.

（2）数据库文件数目的当前最大值能够通过 INIT.ORA 文件中的参数 DB－FILES 来设置(缺省值为 32).

2. 数据库文件的大小

ORACLE 数据库至少包括一个 500KB 的数据库文件来容纳初始的数据字典和初始的回退段。数据库创建时利用 SQL 语句 CREATE DATABASE　创建一个或多个足够大的数据库文件来满足数据库的初始用途。随后任何时候均能根据应用需要利用 SQL 语句 CREATE TABLESPACE(对新的表空间)或 ALTER TABLESPACE（对现存的表空间）添加任意大小的附加数据库文件。

ORACLE RDBMS 以 ORACLE 块为单位管理数据库文件的存贮空间。ORACLE 块的大小与操作系统标准 I/O 块的大小是不同的，通常是其倍数。数据库创建时能够通过设置 INIT.ORA 文件中的参数 DB－BLOCK－SIZE 来改变 ORACLE 块的大小,然而数据库一经创建后就不得改变 ORACLE 块的大小，除非重新创建数据库。

3. 数据库文件的添加和取消

数据库创建时利用 SQL 语句指定了组成一个数据库的 SYSTEM 表空间的初始数据库文件,随后可以利用 SQL 语句 CREATE TABLESPACE(对新的表空间)或 ALTER TABLESPACE（对现存在表空间）添加数据库文件来达到扩展数据库容量的目的。

一个表空间中的所有数据库文件仅在成功地取消该空间后，才能利用特定的操作系统命令予以删除。

（二）控制文件

需要全本请访问 www.ertongbook.com

控制文件是一个尺寸很小的二进制文件，仅与单个数据库相关联，被用来帮助ORACLE RDBMS运行时准确地对数据库和其它相应的操作系统文件进行定位（标识和验证数据库配置）。其所包含的内容有：物理数据库文件和重作记录文件的名字及其所驻留的存贮设备；数据库创建时间标志；数据库名。

数据库创建时控制文件被创建。 ORACLE 数据库运行时，控制文件将持续地被ORACLE 自动更新，故控制文件必须总是可存取的。

1. 控制文件的数目

例程启动时必须要求控制文件，控制文件的被破坏或丢失将导致整个数据库不可恢复的崩溃。要减少存贮介质故障消除控制文件的危险性，应当在不同的磁盘上维护多个等价的控制文件副本，从而提供额外的系统可靠性。因此，应在不同的磁盘设备上至少配置两个控制文件。

2. 控制文件的大小

控制文件的尺寸一般都相当小，它主要是被 INIT.ORA文件中的两个参数 DB - FILES 和 LOG - FILES所决定。数据库一经建立后就不能改变控制文件的大小。

3. 控制文件的添加和取消

控制文件的名字是被 INIT.ORA 文件中的参数 CONTROL - FILES所指定的。所以，通过改变该参数的值来增加新的控制文件或改变控制文件的名字或存贮位置。

（1）添加控制文件可利用操作系统文件拷贝命令将一个当前控制文件复制到一个新的文件，然后再将该文件名添加到 CONTROL - FILES 的值表中。

（2）取消控制文件可利用操作系统的删除命令将一个控制文件删除，然后再将该文件名从 CONTROL - FILES 的值表中删除。

（三）重作记录文件

每个 ORACLE数据库均要求重作记录文件。重作记录文件的基本作用是保护数据库以防止系统崩溃或存贮设备故障造成的数据损失，从而维护数据库的完整性。重作记录文件的主要内容是重作项，其记录着用于重新构造对数据库所作的全部改变所需要的前滚数据。

1. 重作记录文件的数目

一个ORACLE 数据库至少要求两个重作记录文件。

（1）重作记录文件数目的绝对最大值能够利用 SQL 语句来设置（缺省值是 255）。

（2）重作记录文件数目的当前最大值能够通过 INIT.ORA 文件中的参数 LOG - FILES 来设置(缺省值是 16)。

2. 重作记录文件的大小

各个重作记录文件的大小不要求完全相同，但一个重作记录文件的大小起码是50KB。ORACLE RDBMS 以操作系统标准 I/O 块为单位管理重作记录文件的存贮空间。重作记录文件的大小主要是由 INIT.ORA 文件中的参数 LOG - CHECKPOINT - INTERVAL 来确定。

3. 重作记录文件的使用

对数据库的所有改变均要被记录到重作记录文件中。任何时候仅有一个联机重作记录文件被写,而对各个联机重作记录文件的写按‘循环’方式进行。根据重作记录数据是否

被存档，重作记录文件能够用于两种模式：NOARCHIVELOG 模式和 ARCHIVELOG 模式。采用前者时，最后的联机重作记录文件被写满后即返回到第一个重作记录文件并覆盖其内容来继续写。采用后者，任何重作记录文件则不会被重写直到其内容被存档。NOARCHIVELOG 模式仅能用于系统崩溃或软件故障后的例程恢复，但不能用于存贮介质故障后的恢复。而 ARCHIVELOG 模式则既能用于系统崩溃或软件故障后的例程恢复，亦能用于存贮介质故障后的恢复。

无论是哪一种模式，一个联机重作记录文件被写满时将触发称为检查点的行为，即届时所有被修改的数据库缓冲区均将被写到数据库，从而确保数据库的所有改变均被真实地记录到磁盘上。除了联机重作记录文件被写满时发生检查点以外，还能够通过设置 INIT.ORA 文件中的参数 LOG－CHECKPOINT－INTERVAL 来指定联机重作文件期间特定的检查点发生的时间间隔。

二、ORACLE 数据库的存贮结构

(一)表空间(table space)

ORACLE 数据库被划分成称作表空间的逻辑区域，从而形成 ORACLE 数据库的逻辑结构。一个 ORACLE 数据库能够划为一个或多个表空间，而一个表空间则对应一个或多个数据库文件。数据库初始的表空间是 SYSTEM。引入表空间概念的目的是帮助 DBA 完成以下工作,即：

(1)决定数据库实体的空间分配；

(2)设置数据库用户的空间份额；

(3)控制数据库部分数据的可用性；

(4)分布数据于不同的设备之间来改善性能；

(5)备份与恢复数据。

1. SYSTEM 表空间

每个 ORACLE 数据库均有 SYSTEM 表空间，它是数据库创建时自动生成的。SYSTEM 表空间必须总要保持联机，因为其包含着数据库运行所要求的基本信息。

SYSTEM 表空间能够通过添加新的数据库文件来进行扩充。

2. 多个表空间的使用

一个小型应用的 ORACLE 数据库通常仅包括 SYSTEM 表空间，然而一个稍大型应用的 ORACLE 数据库采用多个表空间对于数据库的使用则会带来更大的灵活性。利用 SQL 语句 CREATE TABLESPACE 创建多个空间的目的是：

(1)按应用对数据的存取特点将 I/O 分散于不同的磁盘设备之间；

(2)将表数据和索引数据分隔进行存贮来减少存取冲突；

(3)对用户或应用实施空间分配策略；

(4)给不同数据库应用类型保留不同的表空间；

(5)控制数据库数据的部分可用性。

3. 用户与表空间的联系

表空间实际容纳着用户数据库实体的数据。用户创建其数据库实体时就必须指定表空间及在表空间中应具有的相应权利。所以对于一个用户来说，其要操纵一个 ORACLE 数据库中的数据库，则应该：

(1) 被授予在一个或多个表空间中的 RESOURCE 特权；

(2) 被指定缺省表空间；

(3) 被分配指定表空间的存贮空间使用份额；

(4) 被指定缺省临时段表空间。

若用户在多个表空间具有 RESOURCE 特权，则创建表、索引和簇时，应当指定该表、索引或簇的数据所驻留的表空间。创建表、索引或簇时没有显式地指定表空间，则表、索引或簇将驻留于用户的缺省表空间中。

应该注意，一个表空间能够包含许多表、索引或簇，但一个表、索引或簇却不能横跨多个表空间。

4. 表空间的联机和脱机

利用 SQL 语句 ALTER TABLESPACE 使得一个表空间处于联机或脱机状态来控制数据库的部分数据的可用性。

SYSTEM 表空间必须始终保持其可用性，即数据库运行期间其总是联机的而不能被脱机，这主要是因为 ORACLE RDBMS 控制数据库的存取活动离不开数据字典。

一个表空间脱机时，因为所有组成该表空间的数据库文件均处于非打开的状态，故该表空间中的数据不能够被访问。数据库运行过程期间访问一个表空间时遇到了某些错误，该表空间可能会自动地被从联机改变成脱机。

（二）段(segment)

表空间中的所有数据是按段的数据库空间分配的方式来存贮的。一个段是对数据库数据存贮所分配的一组数据库块（ORACLE 块），作为数据库中表空间之后存贮的下一个逻辑层次。尽管一个段不能横跨多个表空间，但其可能横跨于组成表空间的若干数据库文件之间。

1. 段的类型

ORACLE 数据库要求五种类型的段：

(1) 数据段(data segment)；

(2) 索引段(index segment)；

(3) 回退段(rollback segment)；

(4) 临时段(temporary segment)；

(5) 自举段(boot segment)。

2. 段的创建方式

大多数用户仅仅关注数据段和索引段，而 DBA 还将关注回退段、临时段和自举段。由于段的类型不同，段的创建方式亦不同：

(1) 用户利用 SQL 语句 CREATE TABLE, CREATE INDEX 和 CREATE CLUSTER 来创建数据段和索引段。

(2) DBA 利用 SQL 语句 CREATE ROLLBACK SEGMENT 来创建回退段。

(3) ORACLE RDBMS 处理 SQL 语句时自动地创建必要的临时段。

(4) ORACLE RDBMS 创建数据库时自动地创建自举段。

三、ORACLE RDBMS 进程结构

ORACLE RDBMS 6.0 的运行需要五个后台进程，来处理对 ORACLE 数据库进行异步读写和监督，ORACLE 进程的工作见图 2-3。

这五个后台进程是：

(1) SMON (system monitor)监督例程故障,完成可能的例程恢复,并清除未使用的临时段。

(2) PMON (process monitor) 监督访问数据库失败的用户进程，完成进程恢复，并清除缓冲区和释放资源。

(3) DBWR (database buffer writer)负责将 SGA 的数据库缓冲区中被改变的数据块写到数据库中。

(4) LGWR (redo log buffer write)负责将 SGA 的重作记录缓冲区中重作记录项写到重作记录文件中。

(5) ARCH (archiver) 负责适时地将联机重作记录文件拷贝到磁带上 。

进程 SMON 和 PMON 将会被定期地“唤醒”来检查其是否被需要来进行处理，亦会当其它进程检测需要之时被调用 。事务提交时，DBWR 不立即对数据库进行写,而 LGWR 则立即对联机重作记录文件进行写。采用 ARCHIVELOG 模式时 ARCH 被激活。

从图 2-3 中可看出,SGA 也是十分重要的部分。

SGA (System Global Area)为系统全局区域,它是 ORACLE 多服务器中的所有用户进程存取数据库数据时所共享的存贮器区域，包含着关于单个例程的数据和控制信息，亦作为用户应用程序和 ORACLE RDBMS 进行信息通讯的缓冲区 。SGA 是 ORACLE 减少 I/O 请求的关键之处,其存贮着一份当前(若干)数据库数据块(页)的拷贝以便每个数据库用户快速地访问。

SGA 包含四个部分,即：

(1) 定长数据缓存区(fixed data area)；

(2) 变长数据缓存区(variable data area)；

(3) 数据库缓冲区(database buffer)；

(4) 重作记录缓冲区(redo log buffer)。

其中，定长数据缓存区存贮着 ORACLE RDBMS 控制系统所需要的数据结构；变长数据缓存区存贮着(若干)数据字典的数据块(页)拷贝 数据库缓冲区存贮着(若干)数据库文件的数据块(页)拷贝 重作记录缓冲区存贮着(若干)重作记录文件的数据块(页)拷贝。

ORACLE RDBMS 被启动时系统会显示 SGA 各个组成部分的尺寸大小，ORACLE 版本五中,DBA 利用实用程序 SGI 检查 SGA 各个组成部分的尺寸大小；ORACLE 6.0 中,DBA 利用 SQL * DBA 命令 SGA 检查各个组成部分的尺寸大小。然而，除了定长数据缓存区之外,SGA 其余部分的尺寸大小要受到 INIT.ORA 文件中的许多系统参数的影响。用户可根据应用需要调整这些参数以合理地使用存贮器空间。

SGA 随着 ORACLE RDBMS 的启动而产生，随着 ORACLE RDBMS 的停机而消失,这就是 SGA 的动态特性。

第三章 SQL＊PLUS 语言

第一节 概 述

SQL＊PLUS 是 ORACLE 系统用来存贮、查询、操纵、控制数据库中信息的交互式工具软件。它具有交互性、非过程化等特点。SQL＊PLUS 作为 ORACLE 系统最基本的接口。主要包括标准 SQL 语言和 SQL＊PLUS 命令两部分。其中 SQL 语言用于实现数据定义、数据操纵和数据控制 而 SQL＊PLUS 命令则用来编辑、存贮 SQL 命令，控制显示结果的格式，实现格式化报表输出等。因此，SQL＊PLUS 是一种增强型的 SQL 语言。

一、SQL 语言

作为 ORACLE 数据库系统核心的 SQL(Structured Query Language)语言是一种类似于英语的结构化查询语言，它是目前国际标准化组织 ISO 推荐的标准关系数据库语言。

SQL 语言具有简单易学的用户界面，丰富且完备的数据处理能力，功能很强。它既可以满足具有某些数据库经验的用户需要，又可以使新的数据库用户容易学会如何使用 ORACLE RDBMS。

SQL 语言的命令按照功能通常可分为如下三类：

（一）数据定义子语言(DDL)

用来定义数据库中的各种对象(表、视图、索引、同义词、簇等)。

（二）数据操纵子语言(DML)

用来检索或更新数据库中的数据。

(三)数据控制子语言(DCL)

用来控制和管理数据库的使用。

二、SQL＊PLUS 命令

SQL＊PLUS 命令是指在标准 SQL 基础上又扩充的一些命令，按其功能划分主要有：格式化报表生成命令、编辑命令和辅助实用命令等。

三、运行 SQL＊PLUS

在开始使用 SQL＊PLUS 之前，必须安装 ORACLE RDBMS 和 SQL＊PLUS，并将 ORACLE 核心调入内存。在操作系统提示符下，若要进入 SQL＊PLUS 状态，则键入“SQLPLUS 用户名／口令 ’即可。例如键入“ SQLPLUS SCOTT/TIGER ’即可进入到 SQL＊PLUS 状态，这时显示器上的提示符为：

SQL＞

在这种提示符下，就可以输入 SQL 语言的语句和 SQL＊PLUS 命令。其中 SQL 语句可以长于一行，必须用分号结束；SQL＊PLUS 命令用于编辑、存贮、检索 SQL 命令，控制输出结果格式，SQL＊PLUS 命令必须在一行中输入，不必以分号结束。这两种命令输入

完，按[Return]键执行之。这里“SCOTT/TIGER”是 ORACLE 系统提供的一个用户的“用户名／口令”，在这个用户下有一个样本数据库，读者可以以这个用户的名字和口令在 ORACLE 系统中注册，并利用它提供的样本数据库进行学习和演示。

若要退出 SQL＊PLUS 环境，只要在‘SQL>’状态下键入 EXIT 命令即可退出 SQL＊PLUS。

本章和后面所用到的样本数据库为一教学数据库，见图 3－1。

S

SNO	SN	SD	SA
学号	学生姓名	学生所属系	年龄
92001	丁　一	CS	19
92002	王　二	CS	20
92003	张　三	MA	21
92004	李　四	CI	19
92005	刘　五	CS	22

C

CNO	CN	CT
课程号	课程名称	课时
C1	高等数学	144
C2	程序设计	72
C3	微机原理	80
C4	数据结构	80
C5	编译原理	60
C6	操作系统	90
C7	计算方法	60
C8	数 据 库	60

SC

SNO	CNO	GR
92001	C1	85
92001	C4	80
92001	C7	90
92003	C8	95
92003	C3	76
92003	C6	81
92004	C2	90
92004	C5	65
92005	C1	90
92005	C5	80
92005	C7	79

TDEPT

TDNO	TDNAME	OFNO
部门号	部门名称	办公室地址
601	硬　件	101
602	软　件	132
603	信息中心	135
604	开发公司	151

T

TNO	TN	TA	JOB	MGR	SAL	COMM	TC	TDNO
教师号	教师姓名	年龄	职务	直接上级	工资	职务津贴	所讲课程	部门号
1234	丁　力	28	讲　师	1242	113		C2	601
1236	王一平	36	主　任	1250	122	30	C4	603
1241	马光明	48	教　授	1253	180		C1	602
1242	付小光	40	主　任	1250	131	30		601
1245	刘　刚	45	讲　师	1241	131		C3	602
1250	王　伟	50	校　长		160	50		601
1251	丁　雨	52	副教授	1253	140		C5	602
1253	郑玉华	50	主　任	1250	150	30		602
1255	张　晓	45	讲　师	1236	131		C6	603
1259	王　兰	53	副教授	1236	150		C7	603
1260	肖　红	43	讲　师	1251	122		C8	602
1265	李　光	50	副教授	1236	150			603

图 3－1　教学数据库

第二节　　数据定义

众所周知，建立数据库系统的第一步就是建立数据模型。而关系数据库实际上是表（关系）的集合，因此关系数据库的数据定义实际上是对各个表的描述。表以一定的逻辑结构和物理结构存于数据库中，所谓数据定义主要是对这些结构加以描述。它应该包括对数

需要全本请访问 www.ertongbook.com