第一章　引论

30年来，数据库理论与技术的发展极其迅速，其研究取得了丰硕成果,其应用日益广泛,时至当今信息社会,它几乎(也应该)无所不在。以三大经典(层次、网状、关系)系统为代表的传统数据库技术对于管理结构简单、操作简单、完全格式和结构化且较稳定的数据已经证明是很成功的，因而在一类重要但有限的传统商务和管理的事务性应用领域中获得了普遍而有效的应用。然而，过去的十几年尤其是近年来，在这个数字化信息爆炸社会里，新的过程与工程型、时间关键型、动态不确定型、主动规则型、非规范超大型等现代(非传统)应用不断涌现,而且今后还将继续大量涌现,有发展成为数据库应用主流的可能，它们已经并将继续对数据库技术提出新的要求与挑战，从而促使了现代（非传统）数据库技术的研究与发展。

1.1　现代数据库应用要求

数据库应用正在并将继续从传统领域不断地迅速向新的现代领域扩展,如 CAD/CAM、CIMS 或工厂自动化、过程控制等工程应用;合作与协同计算 数据通信、电力调度、电话交换等网络管理；电子银行事务、证券与股票交易、电子数据交换和电子商务；雷达跟踪、空中交通管制、武器制导、指挥控制或 C^3I 系统 图像处理与模式识别、实时仿真等。这些应用有着与传统应用不同的行为特性和数据特性。

需要全本请访问 www.ertongbook.com

1.1.1 现代应用的行为特性

不同类型的应用可能有各自的不同特征、限制与要求，但与传统应用比较而言，现代应用有下列一般性动态特征：

1. 复杂性

其行为方面的复杂性主要表现在：

(1) 应用语义复杂　有各种问题域中的语义特性、要求、约束与规则；

(2) 活动复杂　各种任务活动内部及活动之间有紧密联系，即存在活动的内部活动(intra-action)和相互活动(inter-action)；

(3) 操作复杂　不仅是传统的插入、删除、修改，还有关于时间、空间的操作及用户自定义操作；

(4) 行为对象复杂　需要抽象对象型（即 ADT—抽象数据类型),复杂对象,多维(值、空间、时间)概念,多形态(规则、方法、过程、文档等)及多介质等；

(5)执行处理复杂　其执行处理不仅要考虑其逻辑结果本身的正确性，还要考虑逻辑结果产生的时间、场合等。

2. 合作与协同性

各应用任务的执行不是独立的，它们在语义结构、数据、行为(操作)、时间、地点等方面有相互依赖性,因而有事务级的各种通信与同步问题要解决。传统数据库对此完全未考虑。

3. 时间性

现代应用的时间性表现在两个方面：一是它们所使用的数据随时间而变化,即不但要处理对象的‘当前’值,还要处理其变化的历史；二是应用处理的定时性，即查询和事务有定时限制，如截止时间等。

4. 主动性

许多应用能自动监控特定的情形（条件），一旦该情形发生（条件成立），能自动反应而执行预先指定的活动，无须人工干预。

5. 过程不可逆性

在许多时间关键型、实时过程型和工程型应用领域中，其应用工程是不可还原和重现的,或者由于工程是连续和‘长寿’的,还原和重现也是毫无实际意义的。

6. 可靠性

工程领域中的许多应用，尤其是一些关键性应用不能失误，因此它们具有容错性,当发现其中的任务要失败时有‘应急’能力,可自动执行功能替换或补偿任务；或者干脆就是并行地执行任务的多个版本，让至少有一定的版本正确执行，其结果一致，而让不正确版本的结果无效。

1.1.2 现代应用的数据特征

现代应用要求的数据较之传统应用的数据表现出的不同特点，主要在下列几个方面：

1. 数据结构

在数据结构表示方面，现代应用要求更高，例如：

(1) 对象类型　不但要求传统的基本数据型，如整型、实型、字符串等,还要求复杂的或构造的数据类型,如集合、向量、矩阵、时间序列、递归结构及抽象数据类型等；

(2) 对象联系有复杂的联系，如 n-元联系等；多种类型的、多种语义的联系,如关于时间、空间、模态的联系;非显式定义的、依据对象定义而存在的隐含联系（它对许多方面，例如，数据发掘是很有用的）；

(3) 对象表示　除了表示惯常的结构化、格式化的数据，还要表示非结构、半结构的数据,非格式、超格式(hyperformated)的数据等。

2. 数据特性

在现代应用中的数据，本身表现出了与一般传统应用数据的不同特性,主要有：

(1) 多维性　每一数据对象不再唯一由其值来表示，每一值

还有一与之相联的时间,即数据是二维的(值、时间),更进一步,值本身也可能是多维的，即数据对象不只是代表线上的点，还可能是平面、空间等的点、线、面、体；

(2) 易变性数据对象频繁地动态变化，其变化不仅表现在数据的值上,而且也表现在它的定义上,即数据的定义(如类型、联系、构造等)也动态改变；

(3) 多态性　数据对象不仅是传统意义下的“值”，还可能是过程、规则、方法、模型等,也可能是声音、影像或图像等。

3. 数据使用

现代应用对数据的使用也表现出了许多不同特性，主要有：

(1) 数据操作　不仅要进行通常的插入、删除、修改、查询,还要进行各种其他类型的操作、特殊操作,如执行(过程)、点燃(规则)、邻域搜索(图像)、浏览、时态查询等等,还要能即席构造操作和用户自定义操作等；

(2) 互操作性　要求数据对象在不同模式（不是同一模式的不同视图）下的互操作(interoperation）及数据对象之间的交互作用(interaction)；

(3)使用方式　传统的数据使用都是被动的、单向的,即只能经应用（联机用户请求或应用程序）中的操作而被动地存取，其作用方向也只是应用对数据，而现代应用要求数据使用的主动性和双向作用，即数据的状态或状态变迁可主动地驱动操作，除了应用作用于数据外，数据也可作用于应用；

(4) 变化历史　许多现代应用要求数据的状态（值）和模式（定义）的变化记录，即要求数据的多版本；

(5) 数据正确性　在现代应用中，数据的正确性不仅表现在数据值上,而且表现在其值产生的时间(乃至地点)上,即数据有值和时间(乃至空间)二维(乃至三维)正确性。

1.1.3　现代应用对数据库的要求

由上两节所述现代应用的行为和数据特征，可直接得出现代

应用对数据库系统的要求，归纳为：

1. 强有力的数据建模能力

数据模型就是帮助人们研究设计和表示应用的静态与动态特性及完整性约束的完全数学地定义的概念集合。这是任何数据库系统的基础。而现代应用要求更强的建模能力，要求数据库系统提供的建模技术与工具支持：

(1)丰富的基础数据类型　除整、实、字符串及布尔等原子型外，还有记录、表、集合等基本构造型和抽象数据型(ADT)；

(2)复杂信息建模　有多种数据抽象技术，如聚集（aggregation)、概括/特化(generalization/specification)、分类(classification)和组合(association)等，有建模复杂对象的能力和 ADT；

(3)数据模型级的新型操作　例如大多数工程型应用所包含(或要求)的复杂的数学操作、时间操作、多介质操作等。

2. 新的查询设施

这包含两个方面：一是特制 ádhoc)查询语言功能(提供一种特制查询语言或具有这种语言能力的服务），如特制的图形浏览器、相联存取(associated access）设施、使用语义知识的查询设施、实时查询技术、时间联接与时间索引技术等；二是新的查询优化策略，如语义查询优化、时间查询优化、整体查询优化(对一个查询的集合产生一单个查询计划，而不是各查询计划的集合）技术等。

3. 支持元数据处理

许多工程型应用是数据密集和不断发展变化的，所以要求提供能自由地改变和更新有关应用语义和结构的工具，以及支持维护关于应用数据本身的专门数据记录，它们包括关于数据的所有权、建立时间、用途、变更历史等那样的信息。

4. 强有力的数据存储与共享能力

关于数据的存储与共享能力包含两个方面：一是可以存储各种类型的“数据”，不仅是传统含义下的数据，还可以是图形、“过程”、规则、事件及控制知识等；不仅是传统的结构化数据，还可以是非结构和超格式化数据；不仅是单一介质的数据，还可以是多介

需要全本请访问 www.ertongbook.com

质的数据。二是人们能够（假定他们有相应的权限）存取和修改这些数据，而不管他们是何种形式及物理地在何处。

5. 高级的现代事务管理

包括从事务的模型、特征，直到实现技术。现代应用要求：

(1) 复杂事务模型　具有内部构造和相互间通信与合作的事务"原子的、隔离的、平淡的操作序列"的事务概念已不适用。

(2)新的和非—ACID或扩展—ACID 特性　如结构依赖性、语义相关性、实时性、主动性等。

(3) 新的或较弱的正确性准则　传统的可串行化概念已不适用或太强。它对工程（尤其是合作计算的）环境极不合适；它极大地降低了事务执行的并发度,从而也降低了系统性能,对时间关键型应用更不合适。

(4）新的实现技术　例如基于优先级的调度策略，多隔离度或无锁的并发控制协议和机制，"部分"的恢复方法等。

6. 先进的图形设施

图形是各种工程型应用普遍要求的，如各种工具的图形接口、图形数据结构、对数据库或模式的图形浏览式存取、语义联系的可视化说明以及应用本身（如各种 CAD、CASE等)的处理等,所以要求完善集成(seamlessly integrated）先进的图形设施到数据库系统，而不是像现有大多数系统那样数据库与图形设施接口之间有明显的界限分离。

7."识时"的系统执行

如 1.1.1 节和 1.1.2 节中所述，现代应用的数据与任务（事务）往往都有与之相联的时间，因此系统必须"识时"(time-cognizant 地进行数据存取与事务处理。有关讨论在第四章中进行。

8. 触发器或主动能力

现代应用常常要求表示和监控数据与活动的静态特性（静态约束)和动态变化(动态约束),即要求触发器或主动机制。这种机制不仅用于数据库系统本身（如实现完整性检验、一致性和安全性控制等），还直接支持应用语义的需要。具体在第五章中讨论。

1.2 现代数据库研究

现代数据库(Advanced Databases)就是支持现代应用、满足上述现代应用要求的数据库，它是一个范畴，不是一类（像关系、网状)数据库,更不是一种(像 ORACLE、DB2）数据库。有关现代数据库的功能特性在下一节中给出，本节着重讨论现代数据库研究的主要问题、趋向与方法。

基于上节所述，现代应用的行为与数据特征及其对数据库的要求,数据库工作者自 20 世纪 70 年代中期以来进行了大量的卓有成效的研究，其主要工作为：

1.2.1 知识表示

关于数据库建模方法、模型及其语言的研究开始得最早。根据建模方法和模型的面向（着眼的方面）可分类成三种：

1. 语义模型

着眼于信息的静态方面，即对象、对象间的联系和完整性约束。典型的语义模型例如有 E-R(P. P. Chen)、SDM(H. Hammer and D. Mcleod)、SAM * (S. Y. W. Su),还有 TAXIS(J. M. Mylopoulos 等)和 DAPLEX(D. W. Shipman)模型与语言等。

2. 过程模型

着眼于信息的动态方面，说明对象及其联系何时及如何发生变化，以及变化是怎样彼此相关的。这种模型的典型代表有 SHM^+ (M. L. Brodie 等)、TAXIS、REMORA(C. Rolland and C. Richard)和 OBCM(Y. Wand and R. Weber)。

3. 事件模型

着眼于信息与时间之间的联系，它用于不仅要建模关于企业组织，而且要建模它的环境及两者之间的交互作用的信息。表示时间与信息交互作用的有效方式就是“事件”，由于事件有一定的先决条件、触发条件和产生的活动，故通过实践可以很容易地说明

这种交互作用。这种模型的代表有 EVENT(R. King and D. Mcleod)、ODE(R. Agrawal and N. H. Gehani)、HiPAC(U. Dayal 等)。

4. 时态模型

引入时间维，它描述完整的数据历史、状态约束和事件发生。这种模型可以是将时间引入其他数据模型（如 E-R 模型、语义模型、关系模型等)而成。这种模型的典型代表有 THM(U. Schiel)、TODM(G. Ariav)、TDM(A. Segev and A. Shoshani)、TEDM(W. W. Chu 等)等。

1.2.2 体系结构

针对不同的应用特征和要求，人们开发了不同的数据库系统体系结构，我们无法汇总所有这些不同结构，但从构造方法学上却能给出一个公共模式“：相关技术 + 数据库能力”。这里的‘ + ’就是‘ 完善集成 ”(seamless integration)。例如：

OODB（面向对象数据库）= OOT（面向对象技术）+ DBC(数据库能力)

TDB(时态数据库) = TDP(时态数据处理) + DBC

RTDB(实时数据库) = RTP(实时处理) + DBC

ADB(主动数据库) = AM(主动机制) + DBC

…

1.2.3 事务模型

针对现代应用的特征和要求，人们从系统、结构、语义和特性方面对现代事务进行了广泛的研究。传统的原子、平淡事务模型已不适用，人们开发了各种复杂事务模型。对于现代数据库系统中的事务，可形式地定义：

1. 系统模型

一组相关‘ 活动 ’的并发‘ 经历 ”(history)。

2. 结构模型

具有内部构造和相互联系的“操作步”的“经历”。

3. 语义模型

若干相关“事件”的一个“经历”。

在事务特性方面的研究表明现代事务具有结构复杂性、行为相关性、时间性、主动性等，传统的 ACID(atomicity, consistency, isolation, durability) 特性必须扩展。

1.2.4 执行模型

一个数据库系统的执行模型包括：

1. 负载模型

各种应用(事务)的类型、到达(或生成)的分布率、结构与执行特征(在结构、数据、行为及时间等方面的相关性)、施加的控制与约束、所需的各种资源。

2. 处理模型

各种应用（事务）的资源需求及控制与约束的满足策略，如优先级分派、调度、开发控制、恢复等的算法、协议及其相关技术与机制。

3. 正确性准则

正确的含义、内容、测度标准。

1.2.5 资源模型

它包含两个方面：①支持相关应用所需各类资源(如数据、内存、CPU、外存及其 I/O 等)的类型/形态、体系结构、技术指标及使用代价 ②这些资源的分配、使用、回收及再分配的策略、技术与方法。

1.2.6 性能模型

系统性能的监视与分析、调整及优化的策略、技术与方法。

现代数据库研究方法的当前趋势表现有三种：一是扩充最流行的关系数据库，这主要是通过放松 INF 要求，引入“嵌套关系”，

或者增加复杂对象、规则或面向对象能力等来扩充现有关系系统的数据模型和／或数据库系统功能特性；第二是建立一个可扩展的核心数据库系统（而不是一个完整的系统），该核心可根据需要扩展和改制成支持一定应用的数据库系统，其扩展和改制可以是数据建模、查询处理、存取方法、存储结构、并发控制和恢复部件等；第三就是如前面所述将基本的数据库设施和特定的领域技术“完善集成”而建立支持相关现代应用的数据库系统。

1.3 现代数据库的特性

基于上两节的讨论，本节给出现代数据库应有的功能特性。要说明的是，下面列出的是支持各种现代应用所要求的数据库功能特性的并集，并非要求任一现代数据库系统都必须具有所有这些功能特性。另外，这些特性也不全都是现代数据库所特有的，有的传统数据库系统也有(或强或弱)其中的某些(或多或少)特性。这些现代数据库特性有：

1.3.1 复杂信息建模

使用丰富的基本数据类型和数据抽象技术。

1.3.2 多版本管理

能管理数据、文档、程序及其他对象的多个版本,以及对象的历史(版本及其相联时间)。

1.3.3 模式进化及元数据处理

可维护关于应用的数据语义及结构变化的信息。

1.3.4 约束与触发器机制

两者都可是数据库系统中的“主动”元件，即每当发生适合于它们工作的情形时它们就执行的部件。两者的不同在于：①前者

是其值为真的布尔函数，后者是 E-C-A(Event-Condition-Action)规则；②前者是在系统行为(如插入、删除、修改操作)发生前，立即直接防止某种状态（如银行数据库中，取钱后账号的结余为负）出现，而后者是当与之相联的“事件”（数据库状态变迁、事务开始或结束、某个特定时刻)发生时，检测与之相联的‘条件”，再决定它的行动；③前者“防止某种状态出现”的行动是简单地阻止相应系统行为的发生，而后者在条件不成立时无任何行动，条件成立时则执行与之相联“活动”（该活动可以是任何系统行为序列）。有关的讨论在第五章中进行。

1.3.5 现代事务管理

支持复杂事务模型和灵活的事务框架、正确性准则及其实现技术与机制。

1.3.6 视图机制

可以不同的观点(包括静态和动态方面)看待相同的数据，这些数据是数据库中存储的实际数据的子集，且彼此在静态结构和／或动态行为方面可以相同或有一定的不同。

1.3.7 强有力的数据共享

能存储与存取在不同地方、具有不同形态和格式、不同介质的数据，甚至通过 WWW 网来存储与操作。

3.8 新的查询与数据操纵设施

具有灵活和特制的查询及其优化技术，以及新的、多种类型的数据操纵设施。

3.9 分布与合作计算

提供数据与任务(事务)的划分／分解、分布及通信的处理、控制和协调与同步机制。

需要全本请访问 www.ertongbook.com

1.3.10 一体化的图形设施

提供用户接口、数据构造、数据模式、应用处理的高级图形设施的统一集成(不同于各自单个的图形接口)。

1.3.11 “识时”处理机制

能处理与数据和事务相联的时间，这种时间可以是现实世界的‘有效时间’或系统中的‘事务时间’,但不能只是‘用户定义时间’(具有时间值域的属性)。有关方面在第三、第四章中讨论。

1.3.12 先进的资源模型

具有新型的 CPU、数据库缓冲区(或内存)、外存及 I/O 管理的策略、工具与机制。例如‘内存数据库’技术(见第六章),基于优先级的调度策略(见第四、第六章)。

1.3.13 有效的数据库保护

能维护数据库的永久性、一致性、可恢复性和安全性，提供满足现代应用要求的提交、并发控制、恢复、授权策略与技术。如‘预提交’、‘乐观’的并发控制、‘部分恢复’等(见第四、第五、第六章)。

1.3.14 可扩展性

允许改变、增加、删除某些相关部件(如查询处理技术、调度算法、并发控制策略与恢复机制等) 以定制成适合于特定应用的数据库系统。

1.4 现代数据库的主要领域与本书的范围

1.4.1 现代数据库领域

前面已经说到,现代数据库就是支持现代(非传统)应用、满足

其除传统应用要求以外还有特殊要求的数据库。由于应用特征及其要求的变化多样，加上数据库工作者的不同立场与观点、爱好与特长、策略与方法，从而形成了数据库世界的五彩缤纷的景象。下面按人们如何在数据库系统中采用或引入相关技术以设计和开发出相应的现代数据库系统来归纳它们，从而试图给出现代数据库技术的一个谱系，进而界定本书的范围。其目的是想给读者以“什么是现代数据库”的一个完整轮廓，尽管这种归纳可能是不十分(或甚至是十分不)准确、全面的。

第一，整体系统方面　相对传统数据库而言，在数据模型及其语言、事务处理与执行模型、数据库组织与物理存储等各层上都集成了新的技术、工具与机制的有：

· 面向对象数据库(Object-Oriented Database)；

· 时态数据库(Temporal Database)；

· 实时数据库(Real-Time Database)；

· 主动数据库(Active Database)。

第二，体系结构方面　在数据库的基本原理上无根本性的变化，但在系统的体系结构方面采用和集成了新的技术的有：

· 内存数据库(Main Memory Database)；

· 分布式数据库(Distributed Database)　包括当前时兴的客户/服务器数据库；

· 并行数据库(Parallel Database)；

· 多数据库或联邦数据库　Federal Database)；

· 数据仓库(Data Warehouse)。

第三，应用方面　数据库系统原理也无根本性变化，而以一定的应用特性为出发点，在某些方面采用和引入一些非传统数据库技术，从而加强或改进了系统对有关应用的支撑能力，这尤其表现在数据模型、语言及查询处理方面。属于这类的有：

· 工程数据库(Engineering Database)　支持 CAD/CAM 的数据库是其典型代表，尤其是 CASE 数据库；

· 科学与统计数据库(Scientific and Statistic Database)；

·超文档数据库(Hyperdocument Database)包括超文本/超媒体数据库(Hypertext/Hypermedia Database),多媒体数据库(Multimedia Database)也可放入这一类;

·空间数据库(Spatial Database)包括地理数据库(Geographic Database);

·演绎数据库(Deductive Database)包括智能数据库和知识库,数据发掘(Data mining)也可放入这一类。

如本节开头所说,这种归类可能不一定正确和完全,但我们的目的不是分类现代数据库,而是想告诉读者哪些当前可以属现代数据库的范畴,然后选定本书所讨论的范围。

1.4.2 本书讨论范围

我们不打算也不可能在本书中讨论上述所有的方面,而是侧重于整体系统方面,分别在下面的第二、第三、第四和第五章中讨论面向对象数据库、时态数据库、实时数据库和主动数据库。对第二和第三个方面分别选择内存数据库和超文档数据库作代表,在第六和第七章中讨论。这样选择主要是考虑:①现代应用的普遍而迫切需要;②数据库的未来发展趋势;③本书的目的是为了国防军事领域及工程、过程控制和时间关键等类型的应用;④作者本人主要从事的研究与实践工作。

第二章　面向对象的数据管理

2.1　面向对象的基本概念

“面向对象”(OO, object-Orientation）是当前计算机科学与技术界普遍流行的一个术语。它具有广泛的背景和长久的历史，最早可追溯到 20 世纪 60 年代末和 70 年代初,进入 80 年代后引起了人们的普遍关注。它最先起源于程序设计语言和风范，但很快被引入计算机科学与技术的其他领域，如数据库、操作系统、软件方法与软件工程乃至系统结构。这里我们讨论有关面向对象数据库方面的问题。

面向对象实际是一种方法学，它将所考虑的现实世界或系统看成是一个对象的集合，每一对象都具有特性和行为，即包含了一组数据和该数据集上的一组操作。面向对象的基本概念是对象、对象标识、类和继承（尽管没有一个统一的“面向对象”的判定标准,但这些作为公共的基础是人们能普遍接受的),下面分别叙述。

2.1.1　对象

“对象”就是具有自身状态及响应外部请求即操作自身状态的能力的实体。对象的自身状态就是其“属性”（或称特性、实例变量、数据部件、槽)的值集。外部请求称为“消息”,为一个对象定义的所有消息的集合构成它的“外部(或抽象)接口”。响应消息而对状态进行操作(因而可改变状态)的程序称为“方法”,为一个对象所定义的所有方法的集合则确定了它的“行为”。对象的自身状态和方法隐藏在内部，对其用户是不可见的，用户只能通过发消息

（即对操作或方法的调用）而向对象提请求来实现与对象的交互作用，这称为‘封装’原理。这种原理提高了模块性和可维护性，因为用户的应用不涉及对象的具体实现与内部表示，故其实现可以改变而不影响应用或用户。

形式地，一个对象就是一个四元组：

$$<ID, DS, MS, RS>$$

其中 DS 为数据集，即对象的状态；MS 为方法集，即对象的行为；RS 为关于对象之间关系的规则集，即关于对象的复杂构造、外部请求或对象间交互作用的规则；ID 为对象的标识，下面要对其专门讨论。

对象的结构如图 2-1 所示。

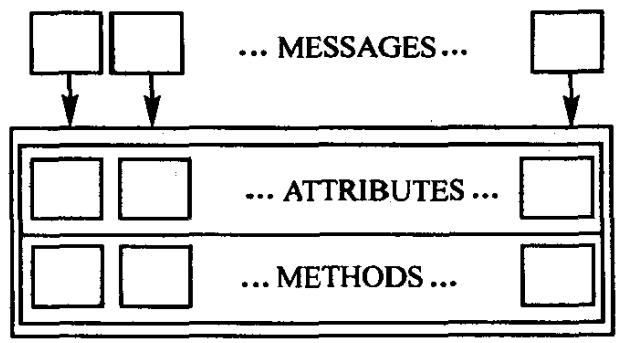


图 2-1　对象

2.1.2　对象标识符

每一对象与一个唯一标识符相联，该标识符无关于它的状态和结构变化，这意思就是对象独立于其值而存在。较之简单地描述对象的值，如关系模型中的关键字，标识符是一个存储指示概念，它不能像改变描述对象的值那样改变，甚至可以改变对象的型（type）而不变其标识符（尽管这不是大多数面向对象系统所支持的）。

由于标识符是存储指示概念，故具有完全相同值的两个不同对象也是可辨别的，但它们若作为关系中的元组，则不能区别，因为一个关系就是一个集合，而集合中的元素是不能重复的。面向

对象中的标识符与层次数据库中的记录指针类似，其主要不同是前者为逻辑指针，后者为物理指针，这样标识符可以用于访问完整性，而记录指针则不能。也因为如此，在像关系数据库这样的“面向值”的系统中,对象标识要通过显式地(explicitly)引入对象标识符来实现，而这就给用户施加了确保标识符的唯一性和访问完整性的职责。

2.1.3 类

上面已谈到，每一对象具有一组属性和一组方法，它们分别表示对象的结构与行为。共有同样结构和行为的对象组合成“类”(class)，所以类是组合具有相同属性和方法的对象在一起的“模板”(template)。类中的成员称为‘实例对象’或‘实例’,所以类实现了数据建模中的‘is-instance-of’联系。类和通常的“型”(type)不同：类是一种运行时的概念，它不用于检查正确性，在运行时可以变化；而型是一种说明时的概念，它用于检查正确性，在运行时不能变化。多个类的定义可与一单个型的定义一致，一个类只与一个型相联，而一个型可以由多个类来实现。

类的属性的值域可以是一个类，依次地，它的属性的值域又可以是一个类，即一个对象的属性值可以是一个或一组对象。依此下去,从而形成‘复杂(或复合)对象’嵌套结构。这种嵌套结构形成一个表示类与其属性间的组成关系的有向、可能有圈的图。这种类组成层次关系等价于数据建模中的“聚合”层次结构，是一种类间的‘is-part-of’联系。

2.1.4 继承

面向对象系统允许用户由现有类导出新的类。这个新的类继承原类的全部属性和方法，此外还可定义它自己额外的属性和方法,以及重定义继承的方法。新类称为原类的‘子类’,它是原类的特化。原类称为导出类的“超类”，它是导出类的一般化。故继承实现了概念建模中的‘IS-A’,联系,它减少了说明变量信息的需要，

需要全本请访问 www.ertongbook.com