对象-关系数据库管理系统原理与实现

李战怀　李红燕　徐秋元　等著

清 华 大 学 出 版 社
北　京

需要全本请访问 www.ertongbook.com

数据库技术在商业领域的巨大成功刺激了其他领域对数据库技术需求的迅速增长。这些新的领域为数据库应用开辟了新的天地，而应用中提出的新的数据管理的需求也直接推动了数据库技术的研究与发展。20 世纪 80 年代，面向对象数据库系统(object oriented database system，OODBS)的研究与发展就是一个极好的例证。

20 多年来，国内外学术界、工业界努力探索数据库技术与对象技术的结合，沿着三条路线展开了面向对象(OO)数据模型和面向对象数据库系统的研究：

一条是以面向对象的程序设计语言为基础，研究持久的程序设计语言，支持 OO 模型；

一条是建立新的面向对象数据库系统支持 OO 数据模型；

一条是以关系数据库和 SQL 为基础，把面向对象技术融入数据库系统的对象关系数据库系统(object relational database system，ORDBS)。

回顾上述 3 个方向所做的研究及开发工作，可以看到各自的发展是不平衡的。其中 ORDBS 方向的成果最为卓著。它在传统关系数据库的基础上吸收了面向对象主要思想，同时又保持了关系数据库的优点，成功开发了诸如 Postgres、Illustra 等原型系统。近年来，各大 RDBMS 厂商也已推出了其产品的对象-关系版本，从而满足了许多新的数据库应用需求。

同样，面向对象数据库研究领域吸引了我国许多数据库工作者，取得了大量的研究成果，开发了多个面向对象数据库管理实验系统。本书介绍的就是其中之一。

从 1991 年起，西北工业大学数据库课题组开始了面向对象数据库技术的研究。该课题组以本书作者李战怀教授和他的导师徐秋元教授为核心，他们从探索新一代 DBMS 研制技术以及研制具有自主版权的数据库产品出发，在八五国防预研项目、国家自然科学基金、863、航空基金、陕西省自然科学基金等项目的支持下开发了客户机-服务器结构的对象-关系数据库系统 Angel。他们从系统底层做起，历时 8 年，投入大量精力和人力研发了 Angel-Ⅰ、Angel-Ⅱ。该系统技术先进，功能丰富，在国防科工委与航空工业部主持的技术鉴定会上得到专家们的高度评价，并且于 1996 年获得国防科技进步二等奖。

本书就是对 Angel 系统以及以此为背景所进行的研究成果(包括 3 篇博士论文)的系统介绍。这对于学习大型数据库系统软件的设计和开发有很好的参考价值，特别是对学习

ORDBS 的基本概念、实现技术、关键技术中难点要点以及实际应用的解决方案是极其有帮助的。

从本书所介绍的研究和开发工作充分说明，研制具有我国自主版权的数据库系统一直是我国数据库界长期的执着的追求。我相信这种追求一定会带来丰硕的成果。

中国人民大学　数据与知识工程研究所
王　珊

我们西北工业大学数据库研究组从20世纪80年代初期开始，在徐秋元教授和李战怀教授的领导下一直从事工程数据库系统的研究，20世纪80年代中期以后，国际数据库界以新型应用领域（CAD、VLSI设计、CASE等）的需求为背景，广泛开展了新一代数据库管理系统的研究，由于这些研究的背景与我们一直关注的应用背景有很大的相似之处，因而引起了我们巨大的关注，经过反复讨论，我们决定自行研发一个数据库管理系统。

采用何种数据模型是我们首先需要解决的问题，当时国际数据库界主流的研究方向是面向对象系统和基于PRLOG的知识库管理系统，但也有一些观点认为要通过对关系模型增加面向对象特征来增加数据模型建模能力，如加州大学伯克利分校的M. Stonebrake教授就是这一观点的代表者。通过对各种模型，如面向对象模型、基于逻辑的模型、嵌套关系模型、复杂对象模型、Postgres的数据模型等广泛的研究，我们提出了一种具有面向对象特征的非第一范式的模型，对这一模型的介绍以“一种适用于工程数据管理的数据模型”名称发表于第十一届全国数据库会议论文集中，基于此模型实现的数据库管理系统我们称之为Angel。

Angel的实现分两个阶段。1991—1995年是第一阶段，在这个阶段我们完全从底层做起，在Sun工作站上实现了一个基于客户机-服务器结构的对象关系数据库管理系统，我们称之为Angel-Ⅰ。1995年年底在国防科工委与原航空工业部主持下开了鉴定会，主要结论是：国内领先，有关研究与国际同步。1996年获得国防科技进步二等奖。1996—1998年是第二阶段，这一阶段主要的工作是提高系统的健壮性并开发客户端开发工具，在此期间，我们完全重新设计了整个系统的体系结构，实现中更多地采用了面向对象的实现方法，在存储管理、查询执行和优化方面做了大量工作，实际上重新实现了整个系统。同时在客户端，我们为Angel开发了ODBC DRIVER，嵌入C的预编译器，以及导航式图形化查询界面。这一阶段开发的系统我们称之为Angel-Ⅱ。整个Angel系统的实现总计花费了四十多人年的工作。

Angel系统的研究与实现，得到国防预研、国家自然科学基金、国家863网络与数据库专题、航空基金、陕西省自然科学基金等多方支持，在此向有关单位致以深深的谢意。

本书以李战怀、李红燕、邓江的博士论文为基础，加上这些年来我们围绕Angel系统发表的论文和技术报告编著而成。虽然主要内容是现成的，但要整理成一本书还是需要花费很大的精力。李红燕副教授从2002年开始整理有关资料，更系统地对以前的工作进行了总

结，到 2003 年年底，经过 1 年多时间的努力，终于完成了这本书。作者要特别感谢蒋泽军教授等对 Angel 系统实现作出贡献的所有项目组成员，但由于人员众多，恕不一一列名致谢。

由于众所周知的原因，出版一本专著性质的书是非常不容易的，作者特别感谢清华大学出版社对本书出版给予的支持，感谢编辑们的辛勤劳动，没有他们的帮助，本书的出版是不可能的。

出版本书的目的是为了把我们在研究对象-关系数据库管理系统方面的心得体会介绍给大家，希望起到抛砖引玉的作用。由于作者的水平与能力所限，书中肯定存在很多不足，如蒙赐教，将不胜感激。

作　者

2006 年 2 月

需要全本请访问 www.ertongbook.com

第 1 章　绪论 …… 1

1.1　**研究目的及意义** …… 1
1.1.1　**扩展关系数据库** …… 2
1.1.2　**面向对象数据库** …… 2
1.1.3　**对象-关系数据库** …… 3
1.2　**对象-关系技术的研究进展** …… 3
1.2.1　**对象-关系技术的萌发** …… 3
1.2.2　**对象-关系技术的现状** …… 4
1.2.3　**对象-关系技术的前景** …… 6
1.3　**相关研究工作概述** …… 6

第 2 章　数据模型 …… 8

2.1　**设计思想** …… 8
2.1.1　**关于值与对象的讨论** …… 8
2.1.2　**对象标识** …… 10
2.1.3　**对象结构** …… 11
2.1.4　**继承性** …… 12
2.1.5　**封装** …… 13
2.1.6　**方法** …… 13
2.2　**形式描述** …… 14
2.2.1　**型与值** …… 14
2.2.2　**类与关系** …… 16

第 3 章　查询代数 …… 20

3.1　**保持模式结构的操作** …… 20
3.1.1　**并** …… 21
3.1.2　**差** …… 22

3.1.3 交 …… 23
3.1.4 选择 …… 23
3.2 改变模式结构的操作 …… 25
3.2.1 投影 …… 25
3.2.2 联结 …… 27
3.2.3 NEST 和 UNNEST …… 28
3.3 代数操作的嵌套使用 …… 29
3.3.1 扩展选择 …… 29
3.3.2 扩展投影 …… 30

第 4 章 体系结构 …… 32

4.1 客户机-服务器体系结构 …… 32
4.1.1 设计原则 …… 32
4.1.2 平台及通信工具 …… 33
4.1.3 功能分布 …… 33
4.1.4 服务器的多进程结构 …… 34
4.2 系统功能划分 …… 35
4.2.1 客户端 …… 35
4.2.2 服务器端 …… 35
4.2.3 语言处理 …… 35
4.2.4 模式管理 …… 36
4.2.5 事务处理 …… 36
4.2.6 查询执行 …… 36
4.2.7 对象管理 …… 36
4.2.8 存储管理 …… 36
4.3 系统环境 …… 37
4.3.1 系统运行环境 …… 37
4.3.2 用户界面 …… 37

第 5 章 数据库语言 …… 38

5.1 主要特点 …… 38
5.2 ADBL 的数据定义功能 …… 39
5.2.1 数据库的创建 …… 39
5.2.2 表的创建 …… 40
5.2.3 索引的创建 …… 44
5.2.4 视图的创建 …… 44
5.3 ADBL 的模式操纵功能 …… 45
5.3.1 表的修改 …… 46
5.3.2 表的删除 …… 47

5.4 ADBL 的查询功能 …… 47
5.4.1 对构造层次的查询 …… 49
5.4.2 对继承层次的查询 …… 52
5.4.3 对传递闭包的查询 …… 53
5.5 ADBL 的数据操纵功能 …… 54
5.5.1 INSERT 语句 …… 54
5.5.2 UPDATE 语句 …… 56
5.5.3 DELETE 语句 …… 56
5.5.4 对视图的操作 …… 57
5.6 ADBL 的数据控制功能 …… 59
5.6.1 完整性保护 …… 59
5.6.2 安全性保护 …… 61
5.6.3 事务管理 …… 65
5.6.4 封锁机制 …… 67

第 6 章 基类扩充 …… 68

6.1 对 built-in 类型的扩充 …… 68
6.1.1 域 …… 68
6.1.2 模式类型 …… 69
6.1.3 抽象数据类型 …… 69
6.2 用户定义函数机制 …… 70
6.2.1 UDF 在 ADBL 中的使用 …… 71
6.2.2 UDF 的继承机制 …… 72
6.2.3 UDF 的实现 …… 73
6.3 利用 UDF 实现操作符定义及操作符重载 …… 76

第 7 章 嵌入式 ADBL …… 78

7.1 语法结构 …… 78
7.1.1 可执行语句 …… 78
7.1.2 说明性语句 …… 79
7.2 预编译实现方法 …… 81
7.3 嵌入式 ADBL 的程序首部处理 …… 83
7.3.1 SQL 变量 …… 83
7.3.2 SQL 通信区 …… 87
7.4 嵌入式 ADBL 的程序体处理 …… 87
7.4.1 不含输出宿主变量的 AEDBL 语句 …… 88
7.4.2 含有输出宿主变量的 AEDBL 语句 …… 88
7.4.3 特殊 AEDBL 语句的处理 …… 89
7.5 游标机制 …… 89

7.5.1 游标的声明 …… 90
7.5.2 游标的打开 …… 91
7.5.3 对当前实例的取值 …… 91
7.5.4 游标的关闭 …… 92
7.5.5 带游标的 ADML 操作 …… 92
7.5.6 游标的撤销 …… 95

第 8 章 复杂对象 …… 96

8.1 扩展的非第一范式结构 …… 96
8.2 复杂对象语义 …… 97
8.3 对象引用 …… 99
8.3.1 引用作为完整性约束 …… 99
8.3.2 引用作为类型构造器 …… 101
8.3.3 两种特殊的引用方式 …… 106
8.3.4 引用信息的外存表示 …… 107
8.4 复杂对象信息组成 …… 108
8.5 复杂对象表示形式 …… 109
8.5.1 复杂对象语言形式 …… 109
8.5.2 复杂对象内存形式 …… 109
8.5.3 复杂对象外存形式 …… 112
8.5.4 复杂对象显示形式 …… 116
8.6 对象管理 …… 116
8.6.1 对象缓冲区管理 …… 117
8.6.2 对象存取 …… 119
8.6.3 驻留对象管理 …… 121
8.7 对象聚集 …… 122
8.7.1 传统聚集策略 …… 122
8.7.2 Angel 聚集策略 …… 122
8.8 对复杂对象操作的实现 …… 125
8.8.1 ADML 操作 …… 125
8.8.2 授权操作 …… 127
8.8.3 并发控制操作 …… 128

第 9 章 查询处理与优化 …… 130

9.1 查询语义 …… 130
9.1.1 路径语义 …… 130
9.1.2 对层次的讨论 …… 131
9.1.3 层次的确定 …… 132
9.1.4 层次的匹配 …… 136

9.2 查询处理 …… 137
9.2.1 总体处理方案 …… 137
9.2.2 嵌套查询处理 …… 143
9.2.3 继承层次查询处理 …… 150
9.2.4 传递闭包查询处理 …… 152
9.3 查询优化 …… 155
9.3.1 代数查询优化 …… 156
9.3.2 索引机制 …… 163
9.3.3 基于代价评估的优化方法 …… 165

第 10 章 模式管理与动态模式修改 …… 176

10.1 模式管理 …… 176
10.1.1 数据字典 …… 176
10.1.2 ANGEL $ TABLES 表 …… 177
10.1.3 模式管理器 …… 181
10.1.4 模式管理的应用 …… 184
10.2 动态模式修改 …… 185
10.2.1 ORDBMS 中的 DSM …… 185
10.2.2 DSM 在 Angel 系统中的实现 …… 190

第 11 章 安全机制与完整性保护 …… 206

11.1 安全机制 …… 206
11.1.1 安全性定义 …… 207
11.1.2 安全性检查 …… 218
11.1.3 安全机制的实现 …… 218
11.2 完整性保护 …… 220
11.2.1 ORDBMS 主动性规则的研究 …… 221
11.2.2 Angel 对规则系统的支持 …… 224
11.2.3 S E-C-A 模型的实现 …… 226
11.2.4 S E-C-A 模型中的规则行为分析 …… 229
11.2.5 语义完整性约束 …… 232

第 12 章 事务管理 …… 236

12.1 事务 …… 236
12.1.1 基本概念 …… 236
12.1.2 事务管理器 …… 239
12.2 常规事务管理 …… 241
12.2.1 封锁与并发控制 …… 242
12.2.2 日志与恢复 …… 247

12.2.3 事务管理器的实现 …… 250
12.3 长事务管理 …… 250
12.3.1 保存点机制 …… 251
12.3.2 具体实现 …… 252

第 13 章 开放数据库连接驱动程序的设计与实现 …… 256

13.1 开放数据库连接简介 …… 256
13.1.1 ODBC 接口 …… 257
13.1.2 ODBC 组件 …… 258
13.1.3 驱动程序类型 …… 259
13.1.4 一致级 …… 260
13.2 设计目标 …… 262
13.2.1 驱动程序结构 …… 263
13.2.2 Winsock 简介 …… 265
13.3 重要数据结构 …… 266
13.3.1 三级句柄 …… 266
13.3.2 数据类型 …… 267
13.3.3 返回状态与出错信息 …… 268
13.4 Angel 数据传输协议 …… 269
13.4.1 连接 Angel 服务器 …… 269
13.4.2 发送命令 …… 270
13.4.3 返回结果 …… 270
13.5 具体实现 …… 272
13.5.1 初始化连接 …… 273
13.5.2 终止连接 …… 275
13.5.3 处理 SQL 语句 …… 276
13.5.4 取回结果 …… 279
13.5.5 处理终止语句或事务 …… 282
13.5.6 信息查询与目录函数 …… 283
13.6 数据源管理 …… 284

第 14 章 复杂对象推理的逻辑语言 …… 285

14.1 A-logic 的语法 …… 285
14.1.1 类型 …… 285
14.1.2 A-logic 语言 …… 286
14.2 A-logic 的语义 …… 299
14.3 讨论 …… 302

附录 A Angel 数据语言文本 …… 304

参考文献 …… 315

需要全本请访问 www.ertongbook.com

第1章 CHAPTER 1 绪 论

对象-关系数据库技术是目前数据库领域的主流技术，本书将完全从低层研究开发的对象关系数据库管理系统 Angel 为背景，全面深入地论述了对象关系数据库系统的原理与实现技术。作为绪论，本章介绍研究对象-关系数据库管理系统的目的及意义，并对对象-关系技术的研究进展进行总结和回顾。

1.1 研究目的及意义

应用的需求始终是推动技术进步的动力。自 20 世纪 70 年代初 E. F. Code 提出关系数据库理论，定义关系数据模型以来，关系数据库管理系统(relational database management system，RDBMS)以其简洁、严密的数据模型，简明易学、功能强大的描述性查询语言而备受用户青睐，并迅速取代网状及层次型数据库系统，成为数据库技术发展的主流。之后，随着 RDBMS 理论基础及其实现方法的日趋成熟，关系数据库技术日臻完善。

近年来，随着计算机辅助设计(CAD)、计算机辅助制造(CAM)、计算机辅助软件工程(CASE)、全球信息系统(GIS)、图像处理、超文本应用等领域的飞速发展及其在传统领域中应用的深化，要求数据库管理系统(database management system，DBMS)能够有效地管理复杂对象。比如在工程应用领域，一个客观复杂实体往往由数十个，甚至成百上千个简单实体组成，为了减小数据库应用系统的设计复杂度、提高其执行效率，要求 DBMS 不但能根据实体丰富的语义进行建模、提供有效的存储与操纵手段，以及模拟复杂实体的复杂行为，而且在逻辑上还要将一个复杂实体的表示和操纵作为一个整体看待，在操纵数据的同时考虑实体间的复合语义，即各简单实体的存在方式(独立或依赖)以及实体间的引用方式(共享或排他)。

然而，传统 RDBMS 由于采用满足第一范式(first normal form，1NF)的平关系模型，在面对各种新的应用领域时存在以下不足。

(1) 不能完整地描述复杂实体。对一个逻辑上作为整体的复杂实体只能采用分解成多个关系的方法进行描述，从而在内部数据库结构和由这种结构所描述的外部实体之间缺乏一对一的对应关系，在查询时这将导致系统效率的严重降低。

(2) 缺乏稳定的元组标识。RDBMS 中完全由关键字属性值作为元组的惟一标识，当关键字值不稳定时，引用时会产生数据的一致性问题，给维护带来很大困难。

(3) 缺乏数据抽象。在关系模型中只有一个非常简单的结构构造概念——关系，缺乏

诸如聚合(把由多个子实体组成的复合实体抽象成一个聚合实体)、概化(把相似实体归类为一个更一般的实体型)与特化(把抽象的实体型细化成更特殊的实例)等在许多实际应用中非常重要的抽象概念。

(4) 缺乏丰富的类型系统。

这些不足使得传统关系型数据库技术在面对来自"非经典"应用领域中的巨大挑战时显得力不从心,新的数据库技术应运而生。

1.1.1 扩展关系数据库

这类研究相对来说是"改良性"的:即在传统关系模型上增添各种特点,或打破原有约束以支持各种新型的、复杂结构的数据。早期对关系模型的扩展主要包括非第一范式(non-first normal form,NF^2)和抽象数据类型(abstract data type,ADT)。

非第一范式打破了关系模型中第一范式的要求,从而能够有效地描述复杂实体。这方面的工作主要集中在 20 世纪 80 年代,所作的扩充也各有不同,包括:P. Pistor 等提出的 AIM-P 中,允许关系的属性为任意域[1];Jaeschke 与 Schek 所扩充的关系模型允许关系带有集合值属性[2];由 Thomas 和 Fischer 提出的"嵌套关系模型"[3]。其中,嵌套关系模型融合了前两种模型,它允许关系中集合操作与聚合操作以任意层次相互嵌套,建模能力最强,因此后续的研究大多围绕嵌套关系进行,包括代数和演算系统[4]、查询语言[5]、规范化[4][6]等。

抽象数据类型原本是程序设计中的重要概念,它强调数据的封装,是面向对象思想的起源之一。而在数据库系统中,ADT 表示用户定义类型,用户不仅要定义类型的规格(数据长度及适用函数),还需要用外部语言予以实现。DBMS 对新类型进行登记,并在适当时机调用相应函数。将 ADT 引入 DBMS 的先驱是加州伯克利大学的 ADT-Ingres[7],并在其后续系统 Postgres[8]中进一步阐明了相关的优化方法。ADT 是扩展数据库类型系统的强大武器,在很多系统中都采用这种方法。

关系模型的这类扩展,克服了平关系不能自然地表达复杂实体结构的缺点,并因为与平关系有着自然的联系,从而可以充分利用平关系系统的研究成果和原有的理论基础。另一方面,这种方法虽然确实增强了关系数据库(RDB)的建模能力,但在数据聚合、概化和特化、刻画复杂对象行为以及对象标识的稳定性等方面,仍未能有根本性改善。

1.1.2 面向对象数据库

这类研究将面向对象思想融入数据库技术,从而发展为全新概念的面向对象数据库管理系统(object-oriented database management system,OODBMS)。尽管 OODB 缺乏通用数据模型、形式基础及强有力的实验活动,但一般说来,它至少包含有如下核心概念[9]:

(1) 对象和对象标识。在面向对象系统和语言中,现实世界的客体统一地用对象来模拟,每一对象有与之相联系的惟一标识符。

(2) 属性和方法、封装和消息传递。每一对象都有一个状态和行为。对象的状态是对象属性值的集合,而对象的行为是作用于对象状态上的方法(程序代码)集。对象的属性值

本身也是一个对象。被封装在对象中的状态和行为只有通过显式的消息发送才能存取。

(3) 类和 ADT。类是把具有相同属性和方法的对象组织在一起的一种手段,对象和类之间的关系是 instance-of 联系。类的含义与 ADT 相似。

(4) 类层次和继承。面向对象系统允许用户从已有类中导出新类。新类继承父类的所有属性和方法。它们之间的关系是子类——超类联系。用户可以为新类指定附加的属性和方法。一个类可以有任意多个子类。

面向对象数据库技术除提供传统关系数据库系统的功能之外,还利用类来组织复杂对象,利用方法来模拟复杂对象行为,利用继承机制来实现对象属性和方法的重用。虽然 OODBMS 在一些特定领域(如 CAD)内能较好地满足应用需求,但由于它不支持标准的结构化查询语言(structured query language,SQL),从而在通用性方面失去了优势。还有,目前绝大多数 OODBMS 都不能很好地支持视图、自动查询优化处理、动态模式演变、并发控制、安全授权、故障恢复及参数化的系统性能调整,等等,所以 Won Kim 认为 OODB 更像一个面向对象程序设计语言(object-oriented programming language,OOPL)的持久存储系统而不是数据库管理系统[9]。

1.1.3 对象-关系数据库

现实与需求之间的矛盾推动了对数据库新技术的研究。20 世纪 90 年代初,将面向对象技术与关系数据库技术相结合的新型数据库管理系统——对象-关系数据库管理系统(object-relational database management system, ORDBMS)——登上了数据库舞台。ORDBMS 由于支持已被广泛使用的 SQL,具有良好的通用性,同时又具备面向对象特性,能支持复杂对象及其复杂行为。从近几年的发展趋势看,ORDBMS 已取代 OODBMS 成为新一代 DBMS 的主要选型,并逐渐与 RDBMS 分庭抗礼。

从前面的讨论可以看出,应用领域要求新的 DBMS 不但要有强大的建模能力,而且必须要有高级的查询手段,使得新型对象-关系技术的研究要同时具有理论价值和长远的应用前景。

1.2 对象-关系技术的研究进展

1.2.1 对象-关系技术的萌发

自 1986 年起,将面向对象技术引入到数据库领域来,已走过了近二十年的历程。其间,对于二者相结合的研究在学术界和工业界都很活跃,各研究机构和软件厂商做了许多工作,探讨如何将二者有机地结合起来,发表了为数众多的研究报告,推出了若干原型系统和商品化的数据库管理系统。概括起来说,将面向对象技术结合到数据库系统中的方式大致可归纳为如下四种。

1. 数据库系统工具包/部件

为了满足不同领域的需求,通过提供 DBMS 核心及工具包的方式对 DBMS 进行各个层

次上的扩充。这些需求可能包括不同的查询语言、存取方法、存储组织和不同的事务机制。

基于该方法的代表之一是 EXODUS[10]。它提供了一个对象存储管理器，一个支持持久对象的程序设计语言，用于书写新的存取方法和查询操作等，以及一个查询优化器生成器（它可以从基于规则的语言说明出发，生成一个适用于特定领域的查询语言的查询优化器）。

2. 支持持久对象的程序设计语言

这是一种在面向对象程序设计语言的基础上，增加数据库功能的方法。具体地说，它采用一种 OOPL（如 Smalltalk、C++等）的类型系统及编程模式，通过增加新的功能，使其数据成为持久的和可共享的，并使其程序原子化执行。该方法所涉及的主要问题有：能否尽量减少对对象使用与定义所做的限制，是否能使所有类型都成为持久类型，如何实现数据的持久性，如何保证类型系统和类型的安全性，如何支持导航的透明性，采用何种持久数据结构的垃圾收集技术，等等。

3. 面向对象的数据库系统

这是一种最为革命的方式。它将数据库技术与面向对象技术相结合，产生了面向对象的数据库管理系统。由于有关 OODBMS 的数据模型、查询方式、查询语言等都没有统一的定义，因而 OODBMS 的研究涉及到了很多方面的问题，诸如数据模型、查询语言、索引技术、查询优化和处理技术、系统结构以及用户界面等。

目前比较有代表性的 OODBMS 有 Gemstone[11]、Orion[12]、O_2[13] 等。

4. 对关系数据库系统的扩充

这是一种以传统的 RDBMS 为基础进行演化的方式。其主要思想是在 RDBMS 之上扩充面向对象特征，如增加一个对象管理层来支持方法与继承，并负责对象与 RDBMS 中关系、元组之间的转换；将 RDBMS 的类型系统开放，支持用户定义的抽象数据类型，等等。

加州大学 Berkeley 分校的 Postgres 系统是这种方式的早期代表。Postgres 是 Ingres 的扩展，其目标之一是提供对复杂对象的存储和查询。它采取了一种把过程作为数据类型的方法，并采用了预先计算和查询改写技术来减轻这种以过程为中心的方式所带来的开销。

1.2.2 对象-关系技术的现状

回顾按照上述四个方式所做的研究及开发工作，可以明显看到各自的发展是不平衡的。有的方式取得了重大成果，有的方式已成为现在的主导方式，而有的方式却似乎失去了继续发展的必要。当然，也有新的方法被提出，新的探索在进行。

就第一种方式来说，由于所提供的工具包既烦琐又不够灵活，在数据库系统设计的某些方面也不够完全，因而其目前的发展现状与原来期望的相差甚远。当然，还有一个重要原因是对象-关系系统已能提供足够的可扩充性和满足大多数应用领域的需求，这就用不着从一个 DBMS 核心和工具包开始去费力地构造数据管理设施了。目前 EXODUS、GENESIS、DASDBS 等工具包已经很少使用了。

第二种方式的研究中取得了若干有意义的成果，它们对 OODBMS 的发展产生了积极

的影响：当今许多 OODBMS 产品的导航式程序设计界面就得益于持久的程序设计语言的研究，还有关于持久模型、持久数据的垃圾收集模式等方面的研究成果也可以直接应用到 OODBMS 中去。但从商业实践的角度来看，却没有一个商品化的产品出现。

有关 OODBMS 的研究非常活跃。OODBMS 的优点包括：

（1）提供了封装和继承的概念，支持数据库设计与编程的可重用性；

（2）提供了对象标识，允许对象之间的引用与共享，可以自然地描述客观复杂对象及复杂对象之间的联系；

（3）由于类与抽象数据类型之间的相似性，OODB 为类型系统的扩展提供了方便的支持手段。

虽然 OODBMS 弥补了 RDBMS 的种种缺陷，自诞生之日起也表现了空前的活力，但其商品化现状显然与以前预期的相差甚远。其主要原因有：

（1）OODBMS 缺乏统一的标准，尽管已提出了一个标准草案 ODMG-93，但许多厂商只是部分地遵从它；

（2）大多数 OODB 缺乏查询设施，虽然有少数系统提供了重要的查询设施，但其查询语言不与 ANSI SQL 兼容；

（3）OO 模型一般使用对象引用和类层次支持对象间联系，对象间引用仅仅提供了对象间的二元关系，在处理 m-n 联系以及递归联系时与网状和层次模型一样困难；

（4）视图虽然在关系模型中得到了广泛支持，但对 OODB 而言，尚不能对 ANSI/X31/SPARC 建议的三级模式体系结构提供有力的支持，因而，用户常常面对的是大粒度的概念模式，极少有或根本没有定义视图的工具。近来虽然出现了一些对 OODBMS 支持视图的建议，但其通用性和灵活性仍然较差。

此外，OODBMS 产品在安全性、完整性、健壮性、可伸缩性、模式演变等许多方面均不如 RDBMS 产品，加之 OODBMS 的应用开发工具很少，对客户机-服务器计算环境的支持不够，与现有 RDB 也不够相容（如不支持 ANSI SQL 的超集），等等。这些因素使目前 OODBMS 市场占有率仅相当于 RDBMS 的百分之一[14]。

就目前看来，扩充关系数据库系统的方式最占优势，也最能满足数据库应用的需求。对象-关系数据库系统就是这种方式走向成熟的产物。ORDBMS 最初以“第三代数据库系统”[15]的面貌出现，它在传统 RDB 的基础上扩展了几乎所有的面向对象思想（如丰富的类型系统、继承性、函数与封装、惟一标识、规则-触发器等），同时又保持了 RDB 的各种优点（如描述性查询语言、视图机制、客户机-服务器体系结构、授权机制、模式演变等）。由于它建立在比较成熟的关系数据库技术基础之上，并且可以利用和借鉴关系数据库系统、面向对象数据库系统和数据库系统工具包这三个方向上的研究成果，因此在数据库界涌现了诸如 Postgres、Illustra、UniSQL 等原型系统和商用系统。近两年来，各大 RDBMS 厂商也已推出了其产品的对象-关系版本，如 Informix Universal Server、IBM DB2/UDB 以及 Oracle8 等。

在数据库市场上最近还出现了一种新方法——面向对象的客户包装层。其基本思想是在关系数据库之上再加一个对象包装层，以支持开发在传统数据库之上运行的面向对象的客户端应用。Persistence Software、Ontologic、HP、Next 等厂商都提供了这样的产品。这些产品一般都生成 C++ 或 Smalltalk 类，并以之作为底层数据库中数据的代表，使得程序设计员能以一种更加自然的方式与底层数据库中的数据打交道。

1.2.3 对象-关系技术的前景

预计 10 年之后，对象-关系数据库系统会成熟，并将有一系列的产品推向数据库市场。那时，对象-关系数据库服务器将提供对面向对象的 ADT 的全面支持，包括 ADT 间的继承以及在多种程序设计语言上实现 ADT 的能力；提供对行类型的支持，以及与此相关的扩充 SQL 功能；提供其他重要的 SQL 功能，如面向对象的视图、授权、触发器、完整性约束等。这些服务器与高度功能化的、面向对象的、带缓存的客户前端相结合，可以提供这样一种开发环境：能在各个层次上采用相同的对象模型对数据库进行描述，既支持查询请求，又支持导航式的程序设计。方法和查询既可以在客户端缓存的数据上运行，也可以在服务器上运行，具体情况依代价的高低而论。触发器、参照完整性约束的执行也是如此。

总之，今后比较成熟的对象-关系数据库系统应该是高度集成的，并具有客户机-服务器体系结构。为使这一前景变成现实，还需要做大量的研究工作。其中服务器功能和性能方面的工作包括对象查询的处理，SQL 中视图、更新、约束、触发器等的扩充，扩充的存取方法的支持等；在客户集成方面的工作有对对象-关系数据进行查询和编程的界面的设计，客户端更新对缓存和数据库更新的触发，方法在客户和服务器两端的正确执行等。此外，要将关系数据库的并行查询技术运用到 ORDB，还需要重点解决诸如多媒体和 GIS 数据模型等重要 ADT 操作的并行执行问题；而对于遗产数据源的存取则需要解决对象-关系、关系和其他数据源环境中的查询优化问题，以及半结构化数据的有效处理问题等。当然，还有一项重要的工作是标准化，国际标准化组织正在讨论的新一代数据库查询语言标准 SQL3[16] 正在朝着这个方向前进，SQL3 已将面向对象的一些主要特性扩充进了 SQL 标准[37]。

从现实需求的角度来看，ORDBMS 不但很好地适应了新的应用的要求，而且对传统商务数据处理应用深化中提出的基于复杂数据的决策支持查询问题，也提供了良好的解决方案，这给 ORDBMS 带来了巨大的市场潜力。据估计，随着数据库应用领域的进一步扩展，ORDBMS 的市场占有率将在 2005 年达到 RDBMS 的 1.5 倍[14]。无论这个预计是否准确，可以肯定的是，ORDBMS 将获得极大的发展，并迅速占领市场。

1.3 相关研究工作概述

从 1991 年起，西北工业大学数据库课题组承担了国防“八五”预研课题“工程数据库管理系统”的研究任务，着手研制面向工程领域的新一代数据库管理系统。从全面掌握、探索新一代 DBMS 研制技术以及将来拥有自主版权的数据库产品角度出发，课题组从底层研制开发出一个 DBMS——Angel 原型系统。该系统本着“以满足工程应用的需求为目标，在技术上博采众长，有所创新”的原则，独创性地提出了具有面向对象风格的扩展非第一范式数据模型，并于 1995 年 11 月通过部级鉴定，被认为“是一个比较典型的对象-关系数据库管理系统”。

从 1996 年起，针对原型系统在执行效率和健壮性方面的缺点，同时为探索拥有自主版权的数据库管理系统的实用化方法，课题组承担了国家“863”计划“工程数据库管理系统 Angel 的实用化开发”课题的研究任务。其主要工作有：对实现方法的改进（包括对象缓冲

需要全本请访问 www.ertongbook.com