作者：希赛教育软考学院 来源：希赛网 2014年01月27日

数据的规范化

第23章 数据库设计

从历年试题来看，在下午考试的软件设计试题中，也会出现数据库设计试题，主要考点包括数据的规范化、E-R模型、数据库的逻辑设计与物理设计等内容。

23.1 数据的规范化

关系模式满足的确定约束条件称为范式，根据满足约束条件的级别不同从低到高分为1NF、2NF、3NF、BCNF、4NF、5NF等。不同的级别范式性质不同。

关系模式的规范化就是把一个低一级的关系模式分解为高一级关系模式的过程。

关系模式分解必须遵守两个准则：

无损联接性：信息不失真（不增减信息）。

函数依赖保持性：不破坏属性间存在的依赖关系。

规范化的基本思想是逐步消除不合适的函数依赖，使数据库中的各个关系模式达到某种程度的分离。规范化解决的主要是单个实体的质量问题，是对于问题域中原始数据展现的正规化处理。

规范化理论给了我们判断关系模式优劣的理论标准，帮助我们预测模式可能出现的问题，是数据库逻辑设计的指南和工具，具体有以下两点：

用数据依赖的概念分析和表示各数据项之间的关系。

消除E-R图中的冗余联系。

上一节 本书简介 下一节

作者：希赛教育软考学院 来源：希赛网 2014年01月27日

函数依赖

函数依赖的概念通俗地说，就像自变量x确定之后，相应的函数值f（x）也就唯一地确定了一样。函数依赖是衡量和调整数据规范化的最基础的理论依据。

比如记录职工信息的结构如下：

职工工号（EMP_NO）

职工姓名（EMP_NAME）

所在部门（DEPT）

我们说EMP_NO函数决定EMP_NAME和DEPT,或者说EMP_NAME、DEPT函数依赖于EMP_NO,记为：EMP_NO→EMP_NAME,EMP_NO→DEPT.

第 23 章：数据库设计 作者：希赛教育软考学院 来源：希赛网 2014年01月27日

码

关系R<U,F>中的一个属性或一组属性K,如果给定一个K则唯一决定U中的一个元组，也就是U完全函数依赖于K,我们就称K为 R的码。一个关系可能有多个码，选其一个作为主码。

包含在任一码中的属性称为主属性，不包含在任何码中的属性称为非主属性。

关系R中的属性或属性组X不是R的码，但X是另一个关系模式的码，我们称X是R的外码。

主码和外码是一种重要的表示关系间关联的手段。数据库设计中一个重要的任务就是要找到问题域中正确的关联关系，孤立的关系模式很难描述清楚业务逻辑。

第 23 章：数据库设计 作者：希赛教育软考学院 来源：希赛网 2014年01月27日

1NF

第一范式（1NF）是最低的规范化要求。

如果关系R中所有属性的值域都是简单域，其元素（即属性）不可再分，是属性项而不是属性组，那么关系模式R是第一范式的，记做R∈1NF.这一限制是关系的基本性质，所以任何关系都必须满足第一范式。第一范式是在实际数据库设计中必须首先达到的，通常称为数据元素的结构化。

比如：

职工工号	职工姓名	住　址
4110973	吕利英	陕西省西安市北大街 17 号[710001]

上表为非第一范式。分解如下：

职工工号	职工姓名	所在省	所在市	详细地址	邮　编
4110973	吕利英	陕西省	西安市	北大街 17 号	710001

这就满足了第一范式。经过处理后，我们就可以以省、市为条件进行查询和统计。

满足1NF的关系模式会有许多重复值，并且增加了修改其数据时引起疏漏的可能性。为了消除这种数据冗余和避免更新数据的遗漏，我们需要更加规范的第二范式（2NF）。

第 23 章：数据库设计 作者：希赛教育软考学院 来源：希赛网 2014年01月27日

2NF

如果一个关系R属于1NF,且所有的非主属性都完全地依赖于主属性，则称为第二范式，记做

R∈2NF.

为了说明问题，现举一个例子来说明。

有一个获得专业技术证书的人员情况登记表结构为：省份、姓名、证书名称、证书编号、核准项目、发证部门、发证日期、有效期。

这个结构符合1NF,其中 "证书名称"和"证书编号"是主码，但是因为"发证部门"只完全依赖于"证书名称",即只依赖于主关键字的一部分（即部分依赖），所以它不符合2NF,这样首先存在数据冗余，因为证书种类可能不多。其次，在更改发证部门时，如果漏改了某一记录，存在数据不一致性。再次，如果获得某种证书的职工全部跳槽了，那么这个发证部门的信息就可能丢失了，即这种关系不允许存在某种证书没有获得者的情况。我们可以用分解的方法消除部分依赖的情况，而使关系达到2NF的标准。方法是从现有关系中分解出新的关系表，使每个表中所有的非关键字都完全依赖于各自的主关键字。我们可以将其分解成两个表，分别是：省份、姓名、证书名称、证书编号、核准项目、发证日期、有效期，证书名称、发证部门。这样就完全符合2NF了。

上一节　本书简介　下一节

3NF

如果一个关系R属于2NF,且每个非主属性不传递依赖于主属性，这种关系是3NF, 记做R∈3NF.

从2NF中消除传递依赖，就是3NF.比如有一个表（职工姓名，工资级别，工资额），其中职工姓名是关键字，此关系符合2NF,但是因为工资级别决定工资额，也就是说非主属性"工资额" 传递依赖于主属性"职工姓名",它不符合3NF,我们同样可以使用投影分解的办法将其分解成两个表：职工姓名、工资级别，工资级别、工资额。

上一节　本书简介　下一节

BCNF

一般满足3NF的关系模式已能消除冗余和各种异常现象，能够获得比较满意的效果，但无论2NF还是3NF都没有涉及主属性间的函数依赖，所以有时仍会引起一些问题。由此我们引入BC范式（BCNF,Boyeet和Codd提出）。通常认为BCNF是第三范式的改进。

BC范式的定义：如果关系模式R∈1NF,且R中每一个函数依赖关系中的决定因素都包含码，则R是满足BC范式的关系，记做R∈BCNF.

当一个关系模式R∈BCNF,则在函数依赖范畴里，就认为已彻底实现了分离，消除了插入、删除的异常。

综合1NF、2NF和3NF、BCNF的内涵可概括如下：

非主属性完全函数依赖于码（2NF的要求）。

非主属性不传递依赖于任何一个候选码（3NF的要求）。

主属性对不含它的码完全函数依赖（BCNF的要求）。

没有属性完全函数依赖于一组非主属性（BCNF的要求）。

上一节　本书简介　下一节

多值依赖和4NF

第四范式是BC范式的推广，是针对有多值依赖的关系模式所定义的规范化形式。

多值依赖：关系模式R<U,F>∈1NF,X、Y是U的非空子集，Z=U-X-Y也非空，若任取一组值对<x,z>,都可决定一组<y1\y2\...\yn>值，且此决定关系与z值无关，我们就称Y多值依赖于X,记做X→→Y.

定义：关系模式R<U,F>∈1NF,若对任一多值依赖X→→Y, X必包含R的主键，称R是第四范式的，记做：R∈4NF.

例如，表23-1表示关系R（ypm,bm,sccj）。

表23-1　关系R的内容

用品名（ypm）	部门（bm）	生产厂家（sccj）
办公桌	生产经营部	美时办公用品公司
办公桌	总经理办公室	华鹤家具有限公司
办公椅	生产经营部	美时办公用品公司
办公椅	总经理办公室	华鹤家具有限公司
办公椅	计划部	华鹤家具有限公司

分析上表，这是实际工作中常见的登记表，抛开是否规范不说，这样的登记表一目了然。但从规范化的角度来看，对ypm的一个值，不论sccj取什么值，总有一组确定的bm与之对应，所以有ypm→→bm,同样分析有ypm→→sccj,此关系是全码，这说明R不满足4NF,此种关系模式有数据冗余和修改量大等弊端。可用分解法消除不满足4NF的多值依赖。解决办法是：把R分解为R1（ypm,bm），R2（ypm,sccj）。

上一节　本书简介　下一节

非规范化处理

规范化设计所带来的性能问题在实际应用中可能令人无法承受。如果出现这种情况，就要进行非规范化。非规范化就是为了获得性能上的要求所进行的违反规范化规则的操作。由于非规范化几

乎必然导致冗余，占用更多的存储空间，因此它需要对性能和空间的平衡进行考虑，需要不断地尝试和评估过程。进行非规范化有很多方法，不过大部分都与实际应用有关系，包括冗余属性、合并表，等等，可以根据实际的应用进行选择，找到最有效的方法。

作者：希赛教育软考学院　来源：希赛网　2014年01月27日

数据库设计概述

数据库设计涉及范围很广，要设计一个性能良好的数据库并非易事。从本质上讲，数据库设计的过程是将数据库系统与现实世界密切地、有机地、协调一致地结合起来的过程。数据库的设计质量与设计者的知识、经验和水平密切相关。作为数据库应用系统的重要组成部分，数据库设计的成败往往直接关系到整个应用系统的成败。

以数据库为基础的数据库应用系统与其他计算机应用系统相比往往具有数据量庞大、数据保存时间长、数据关联复杂、用户要求多样化等特点。

数据库设计中面临的主要困难和问题有：

同时具备数据库知识与应用业务知识的人很少。懂得计算机与数据库的人一般都缺乏应用业务知识和实际经验，而熟悉应用业务的人又往往不懂计算机和数据库。

项目初期往往不能确定应用业务的数据库系统的目标。

缺乏完善的设计工具和设计方法。

需求的不确定性。用户总是在系统的开发过程中不断提出新的要求，甚至在数据库建立之后还会要求修改数据库结构或增加新的应用。

应用业务系统千差万别，很难找到一种适合所有业务的工具和方法，这就增加了研究数据库自动生成工具的难度。因此，研制适合一切应用业务的全自动数据库生成工具是不可能的。

完善的数据库系统应具备如下特点：

功能强大。

能准确地表示业务数据。

使用方便，易于维护。

便于检索和修改数据。

在合理的时间内响应最终用户的操作。

为以后改进数据库结果留下空间。

维护数据库的工作较少。

具备有效的安全机制来确保数据安全。

冗余数据最少或不存在。

便于进行数据的备份和恢复。

数据库结构对最终用户透明。

数据库设计特点

数据库设计的很多阶段都可以和软件工程的各阶段对应起来，软件工程的某些方法和工具同样也适合于数据库工程，但数据库设计还有很多自己的特点。

从数据结构即数据模型开始，并以数据模型为核心展开。这是数据库设计的一个主要特点。

静态结构设计与动态行为设计分离。静态结构设计是指数据库的模式结构设计，包括概念结构、逻辑结构和物理结构的设计。动态行为设计是指应用程序设计，包括功能组织、流程控制等方面的设计。传统的软件工程往往注重处理过程的设计，不太注重数据结构的设计，在结构程序设计中只要可能就尽量推迟数据结构的设计，而数据库设计正好与之相反，需要把主要精力放在数据结构的设计上，如数据库的表结构、视图等。

试探性（tentative）：数据库设计比较复杂，又缺少完善的设计模型和统一的过程，设计的过程往往是试探的过程（cut and try process），因此设计的结果往往不是唯一的。有时多种方案并存，供设计者选择，而且这种选择并不是完全客观的，有时多少取决于用户的偏爱和观点。

反复性（iterative）：由于数据库设计是一种试探性的过程，这就决定了数据库的设计是一个反复推敲和修改的过程，而不可能"一气呵成".

多步性（multi-stage）：传统的数据库设计采用直观设计法或单步设计法，它由设计者通过用户的调查访问，确认需求，熟悉用户应用问题的语义，结合结构限制与DBMS功能，凭设计人员的经验进行分析、选择、综合与抽象之后，建立数据模型，并用DDL写出模式。由于这种设计方法要求设计人员有比较丰富的经验和熟练的技巧，不易为一般人所掌握，且因缺乏工程规范支持和科学根据，现已抛弃不用。新的数据库设计是分步（阶段）进行的。前一阶段的设计结果可作为后一阶段设计的依据，后一阶段也可向前一阶段反馈其要求，反复修改，逐步完善。

数据库设计方法

能够有效地指导数据库设计，使数据库设计更加合理的原则称为数据库设计方法学。首先，一个好的数据库设计方法学，应该能在合理的期限内，以合理的工作量，产生一个有实用价值的数据库结构。这里的"实用价值"是指既能满足用户关于功能、性能、安全性、完整性及发展需求等方面的要求，同时又服从特定DBMS的约束，并且可以用简单的数据模型来表达。其次，数据库设计方法学还应具有足够的灵活性和通用性，不仅能够供具有不同经验的人使用，而且应该不受数据模型及DBMS的限制。最后，数据库设计方法学应该是可再生的，即不同的设计者使用同一方法设计同一

问题时，应该得到相同或相似的设计结果。

已有的数据库设计方法可分为4类，即直观设计法、规范设计法、计算机辅助设计法和自动化设计法。直观设计法又称单步逻辑设计法，它依赖于设计者的知识、经验和技巧，缺乏工程规范的支持和科学根据，设计质量也不稳定，因此越来越不适应信息管理系统发展的需要。为了改变这种状况，1978年10月，来自30多个欧美国家的主要数据库专家在美国新奥尔良市专门讨论了数据库设计问题，提出了数据库设计规范，把数据库设计分为需求分析阶段、概念结构设计阶段、逻辑结构设计阶段和物理结构设计阶段4个阶段。目前，常用的规范设计方法大多起源于新奥尔良方法，如基于3NF的设计方法、LRA方法、基于E-R模型的数据库设计方法、基于视图概念的数据库设计方法等。下面对几种设计方法做一简单介绍。

1.基于3NF的数据v库设计方法

这是由S.Atre提出的数据库设计的结构化设计方法，其基本思想是在需求分析的基础上，识别并确认数据库模式中的全部属性和属性间的依赖，将它们组织成一个单一的关系模式，然后再分析模式中不符合3NF的约束条件，用投影和连接的办法将其分解，使其达到3NF条件。其具体设计步骤分为5个阶段，如图23-1所示。

1）设计企业模式

利用上述得到的3NF关系模式画出企业模式。具体包括：

分析应用环境，并设定环境中所使用的种种资料。

确定每一种报表各自所包含的数据元素。

确定数据元素之间的关系，如确定主关键字和一般的数据元素。

对每一组或若干组数据元素推导出3NF的关系模式。

在3NF关系模式的基础上画出数据库的企业模式。

2）设计数据库逻辑模式

根据上一步得到的企业模式选定数据模型，从而得出适用于某个DBMS的逻辑模式。

根据逻辑模式导出各种报表与事务处理所使用的外模式。

3）设计数据库物理模式（存储模式）

根据数据库的逻辑模式和给定的计算机系统设计物理模式。

4）评价物理模式

对物理模式估算空间利用情况，并推算输入/输出的概率。

必要时根据物理模式调整各种报表与事务处理的外模式。

对外模式进行存取时间的估算。

5）数据库实现

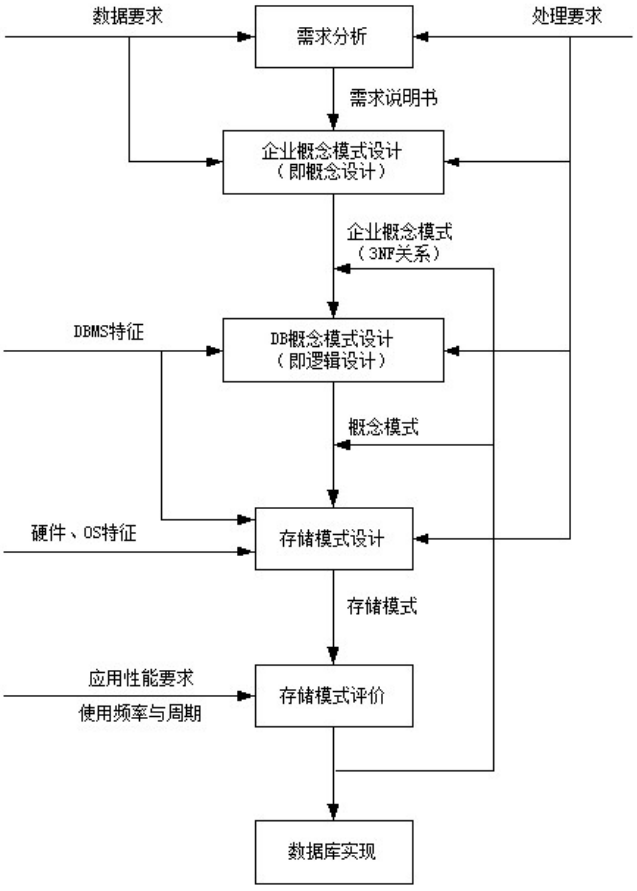

图23-1 基于3NF的DB设计方法

2.LRA方法

数据库设计的LRA方法（Logical Record Access）方法即逻辑记录存取法。它从用户的信息要求和处理要求出发，分3个阶段完成数据库的设计。

需求分析。向现有和潜在的用户了解和收集有关的信息内容和处理要求，并将它们文本化。

逻辑设计，又分信息结构设计（ISD）和信息结构改进（ISR）两步。前者主要是分析各种用户的信息要求，确定实体、属性及实体之间的联系，并综合成一个初始的数据库信息结构。信息结构改进的主要任务是根据设计的初始信息结构、处理要求和DBMS的特点，设计DBMS能处理的模式。它分3步完成，即首先定义局部信息结构，并将局部信息结构合并成全局信息结构；然后分别将全局信息结构和局部信息结构转换为DBMS所能支持的逻辑数据库结构和局部逻辑数据库结构；最后根据数据传送量、应用的处理频度、逻辑记录存取数等因素，改进逻辑数据库结构。

物理设计。LRA方法的物理设计与其他方法的物理设计类似。LRA方法的特点是提供一种定量估算的方法，使得能够对数据库逻辑结构的性能进行分析，在可供选择的若干个结构中选择一个处理效率较高的结构，或者据此对现有的逻辑结构进行改进。为此，LRA使用逻辑记录存取数表示在一个应用程序执行过程中对一个记录类型所要查找的记录的个数，记做LRA数。根据所有应用程序的LRA数及它们在单位时间内要执行的次数，就可以知道哪些应用程序可能要求的I/O次数最多，哪些应用程序在性能上起着主导地位，从而决定如何改进逻辑结构以提高处理效率。

3.基于实体联系（E-R）的数据库设计方法

1976年由P.P.S.Chen提出的E-R方法主要用于逻辑设计。其基本思想是在需求分析的基础上，用E-R图构造一个纯粹反映现实世界实体之间内在联系的企业模式，然后再将此企业模式转换成选定的DBMS上的概念模式。E-R方法简单易用，又克服了单步逻辑设计方法的一些缺点，因此成为比较流行的方法之一。但由于它主要用于逻辑设计，故E-R方法往往成为其他设计方法的一种工具。

4.基于视图概念的数据库设计方法

此方法先从分析各个应用的数据着手，为每个应用建立各自的视图，然后再把这些视图汇总起来合并成整个数据库的概念模式。合并时必须注意解决下列问题：

消除命名冲突。

消除冗余的实体和联系。

进行模式重构。

在消除了命名冲突和冗余后，需要对整个汇总模式进行调整使其满足全部完整性约束条件。

5.面向对象的关系数据库设计方法

面向对象的数据库设计（即数据库模式）思想与面向对象数据库管理系统（OODBMS）理论不能混为一谈。前者是数据库用户定义数据库模式的思路，而后者是数据库管理程序的思路。用户使用面向对象方法学可以定义任何一种DBMS数据库，即网络型、层次型、关系型、面向对象型均可。对象的数据库设计从对象模型出发，属于实体主导型设计。

数据库设计（模式）是否支持应用系统的对象模型是判断是否是面向对象数据库系统的基本出发点。应用系统对象模型向数据库模式的映射是面向对象数据库设计的核心和关键，其实质就是向数据库表的变换过程。有关的变换规则简单归纳如下。

一个对象类可以映射为一个以上的库表，当类间有一对多的关系时，一个表也可以对应多个类。

关系（一对一、一对多、多对多及三项关系）的映射可能有多种情况，但一般映射为一个表，也可以在对象类表间定义相应的外键。对于条件关系的映射，一个表至少应有3个属性。

单一继承的泛化关系可以对超类、子类分别映射表，也可以不定义父类表而让子类表拥有父类属性；反之，也可以不定义子类表而让父类表拥有全部子类属性。

对多重继承的超类和子类分别映射表，对多次多重继承的泛化关系也映射一个表。

对映射后的库表进行冗余控制调整，使其达到合理的关系范式。

面向对象关系数据库设计效果可归纳为：

数据库结构清晰，便于实现OOP.

数据库对象具有独立性，便于维护。

需求变更时程序与数据库重用率高，修改少。

6.计算机辅助数据库设计方法

计算机辅助数据库设计是数据库设计趋向自动化的一个重要步骤，它的基本思想并不是完全由机器代替人，而是提供一个交互式过程，人机结合，互相渗透，帮助设计者更快更好地进行设计工作。在数据库设计过程中，目前还没有全自动设计方法，只能使用计算机进行局部辅助设计，而且一般立足于不同的设计规程和模型化工具，如关系数据库模式的辅助设计工具的应用。许多自动化设计工具，如Oracle Designer,已经具有如下一些稳定的特性：

确定业务和用户需求的功能。

对业务处理建模的功能。

对数据流建模的功能。

对实体及其相互关系建模的功能。

生成创建数据库对象的DDL语句的功能。

对数据库设计周期的支持。

业务处理二次工程。

数据库和应用软件的版本控制。

文档和用户反馈信息报告的生成。

7.敏捷数据库设计方法

近年来，一种新的软件开发方法学--敏捷方法学被逐步应用于数据库设计。它提出了在可控制方式下的进化设计。迭代式开发是它的一个重要特点，即整个项目生命周期中运行多个完整的软件生命周期循环。敏捷过程在每次迭代中都会度过一个完整的生命周期且迭代时间较短。

敏捷方法的一些原则包括：拥有不同技能和背景的人能够紧密合作；每个项目组成员都有自己的数据库实例；开发人员数据库经常集成到共享主数据库；数据库包含计划和测试数据；所有的变化要求数据库重构；每个数据库重构都可以通过编写SQL DDL（对于计划变化）和DML（对于数据迁移）来自动完成；自动地更新所有开发人员的数据库；清晰地分离所有的数据库获取代码等。

敏捷数据库设计方法保持多个数据库在一个系统中，而且不需要专职的DBA.可以通过开发一些简单工具来帮助解决数据库进化过程中大量的重复性工作。

目前，敏捷方法在数据库设计中的应用尚有一些没有解决的问题，如集成数据库和24×7小时实施等，还有待进行进一步的研究工作。

此外，其他的设计方法还有属性分析法、实体分析法及基于抽象语义规范的设计法等，有兴趣的读者可以参阅有关资料。在实际的设计过程中，各种方法可以结合起来使用，例如，在基于视图概念的设计方法中可用E-R方法来表示各个视图。

上一节　本书简介　下一节

作者：希赛教育软考学院　来源：希赛网　2014年01月27日

数据库设计的基本步骤

分步设计法遵循自顶向下、逐步求精的原则，将数据库设计过程分解为若干相互独立又相互依存的阶段，每一阶段采用不同的技术与工具，解决不同的问题，从而将问题局部化，减少了局部问题对整体设计的影响。目前，此方法已在数据库设计中得到广泛应用并获得较好的效果。

在分步设计法中，通常将数据库的设计分为需求分析、概念结构设计、逻辑结构设计和数据库物理设计4个阶段，如图23-2所示。

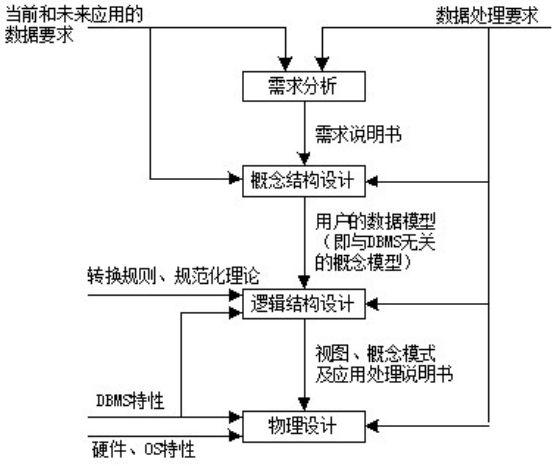

图23-2 数据库的设计步骤

1.需求分析

需求分析是指收集和分析用户对系统的信息需求和处理需求，得到设计系统所必需的需求信息，建立系统说明文档。其目标是通过调查研究，了解用户的数据要求和处理要求，并按一定格式整理形成需求说明书。需求说明书是需求分析阶段的成果，也是今后设计的依据，它包括数据库所涉及的数据、数据的特征、使用频率和数据量的估计，如数据名、属性及其类型、主关键字属性、保密要求、完整性约束条件、更改要求、使用频率、数据量估计等。这些关于数据的数据称为元数据（metadata）。在设计大型数据库时，这些数据通常由称为数据字典（data dictionary）的计算机软件（专用软件包或DBMS）来管理。用数据字典管理元数据有利于避免数据的重复或重名以保持数据的一致性及提供各种统计数据，因而有利于提高数据库设计的质量，同时可以减轻设计者的负担。

2.概念结构设计

它是数据库设计的第二阶段，其目标是对需求说明书提供的所有数据和处理要求进行抽象与综合处理，按一定的方法构造反映用户环境的数据及其相互联系的概念模型，即用户的数据模型或企业数据模型。这种概念数据模型与DBMS无关，是面向现实世界的、极易为用户所理解的数据模型。为保证所设计的概念数据模型能正确、完全地反映用户（一个单位）的数据及其相互关系，便于进行所要求的各种处理，在本阶段设计中可吸收用户参与和评议设计。在进行概念结构设计时，可先设计各个应用的视图（view），即各个应用所看到的数据及其结构，然后再进行视图集成（view integration），以形成一个单一的概念数据模型。这样形成的初步数据模型还要经过数据库设计者和用户的审查与修改，最后形成所需的概念数据模型。

3.逻辑结构设计

这一阶段的设计目标是把上一阶段得到的与DBMS无关的概念数据模型转换成等价的，并为某个特定的DBMS所接受的逻辑模型所表示的概念模式，同时将概念设计阶段得到的应用视图转换成外部模式，即特定DBMS下的应用视图。在转换过程中要进一步落实需求说明，并满足DBMS的各种限制。该阶段的结果是DBMS所提供的数据定义语言（DDL）写成的数据模式。逻辑设计的具体方法与DBMS的逻辑数据模型有关。逻辑模型应满足数据库存取、一致性及运行等各方面的用户需求。

4.数据库物理设计

物理设计阶段的任务是把逻辑设计阶段得到的满足用户需求的已确定的逻辑模型在物理上加以实现，其主要的内容是根据DBMS提供的各种手段，设计数据的存储形式和存取路径，如文件结构、索引的设计等，即设计数据库的内模式或存储模式。数据库的内模式对数据库的性能影响很大，应根据处理需求及DBMS、操作系统和硬件的性能进行精心设计。

实际上，数据库设计的基本过程与任何复杂系统开发一样，在每一阶段设计基本完成后，都要进行认真的检查，看看是否满足应用需求，是否符合前面已执行步骤的要求和满足后继步骤的需要，并分析设计结果的合理性。在每一步设计中，都可能发现前面步骤遗漏或处理不当之处，此时，往往需要返回去重新处理并修改设计和有关文档。所以，数据库设计过程通常是一个反复修改、反复设计的迭代过程。

第 23 章：数据库设计 作者：希赛教育软考学院 来源：希赛网 2014年01月27日

需求分析

需求分析是数据库设计过程的第一步，是整个数据库设计的依据和基础。需求分析做得不好，会导致整个数据库设计重新返工。

第 23 章：数据库设计 作者：希赛教育软考学院 来源：希赛网 2014年01月27日

需求分析的任务

需求分析的目标是通过对单位的信息需求及处理要求的调查分析得到设计数据库所必需的数据集及其相互联系，形成需求说明书，作为后面各设计阶段的基础。因此，这一阶段的任务是：

确认需求，明确设计目标。

分析和收集需求的数据。

整理文档。

1.确认需求、确定设计目标

数据库设计的第一项工作就是通过实地调查研究，弄清现行系统的组织结构、功能划分、总体工作流程，明确用户总的需求目标；通过分析，确定相应的设计目标，即确定数据库应支持的应用功能和应用范围，明确哪些功能由计算机完成或准备让计算机完成，哪些环节由人工完成，以确定应用系统实现的功能。

要确定数据库应用领域，以有效地利用计算机设备及数据库系统的潜在能力，充分满足用户的应用要求，同时，还应当尽可能地考虑将来的应用需要，以提高数据库的应变能力，避免运行过程中对数据库做过多或较大的修改，以延长数据库的生命周期。

2.分析和收集数据

这是整个需求分析的核心任务。它包括分析和收集用户的信息需求、处理需求、完整性、安全性需求及对数据库设计过程有用的其他信息。

信息需求是指设计目标范围内涉及的所有实体、实体的属性及实体间的联系等数据对象，包括用户在数据处理中的输入/输出数据及这些数据间的联系。在收集中，要收集数据的名称、类型、长度、数据量、对数据的约束及数据间联系的类型等信息。

处理需求是指为了获得所需的信息而对数据加工处理的要求。它主要包括，处理方式是实时还是批处理，各种处理发生的频度、响应时间、优先级别，以及安全保密要求等。所要收集的其他信息还有企业在管理方式、经营方式等方面可能发生的变化等。

分析和收集数据的过程是数据库设计者对各类管理活动进行深入调查研究的过程，调查对象包括数据管理部门的负责人、各使用部门的负责人及操作员等各类管理人员，通过与各类管理人员相

互交流，逐步取得对需求的一致认识。由于这种交流的双方知识背景不同，涉及不同层次、不同业务领域的管理人员，因而这一调查研究过程十分复杂和困难。为了减少因分析和收集数据的失误而导致整个系统的失败或延长研制周期，人们研究了许多用于需求分析的方法和技术，力求清楚地表达用户需求，系统地分析和收集需求数据。按方法的规范化程度可分为弱方法学（week methodologies）、强方法学（strong methodologies）和格式化方法（formatted approaches）。按方法描述需求的观点可分为面向数据的方法与面向过程的方法两大类。下面对前三类方法做一个简单介绍。

1）弱方法学

这类方法仅是一些原则性指南，并无专门的语言或特定的记法，有较大的自由度，使用者可以发挥自己的创造性，适用面广，尤其适用于复杂环境的需求分析。如BSP（Business System Planning）、面向对象的分析法（Object Oriented System Analysis）等方法属于这类方法。

2）强方法学

这是形式化方法，它以一种具有确定的语义成分及语法规则的语言作为描述手段，对使用者有强的制约，便于实现自动化，但一般只适用于特定的应用环境。这类方法有PSL/PAS（Problem Statement Language/Problem Statement Analyzer）、SREM（Software Requirement Engineering Methodology）和TAGS（Technology for the Automated Generational System）等。

3）格式化方法

此法介于上述两种方法之间，它虽然采用规范化的记法，但一般没有确定的语义成分。这类方法有BIAIT（Business Information Analysis and Integration Technique）、SA（Structured Analysis and Design Technique）等。其中有的是采用提问对答方式来进行信息分析的，如BIAIT,但大多数采用规范化的图形标记法及文字说明作为表达方式。由于这类方法的用法的自由度适中，且采用直观的图形方式，因而使用较为广泛。目前，这类方法中有不少已开发了相应的自动化支持工具。

3.整理文档

分析和收集得到的数据必须经过筛选整理，并按一定格式和顺序记载保存，经过审核成为正式的需求说明文档，即需求说明书。实际上，需求说明书是在需求分析的过程中逐渐整理形成的，是随着这一过程的不断深入而反复修改与完善的，是对系统需求分析的全面描述，由用户、领导和专家共同评审，是以后各设计阶段的主要依据。这一步的工作是进行全面的汇总、整理与系统化，以形成标准化的统一形式。

需求说明书作为应用部门的业务人员和数据库设计人员的"共同语言",要求准确地表达用户需求，无二义性，可读性强，为数据库的概念设计、逻辑设计和物理设计提供全面、准确和详细的资料。

需求说明书通常包括下述内容。

应用功能目标

明确数据库的应用范围及应达到的应用处理功能。

标明各用户视图范围

根据结构与职能关系图、数据流程图和管理目标与功能相关表等，确定不同部门或功能的局部视图范围。

应用处理过程需求说明

数据流程图：它主要反映应用部门原始业务处理的工作流程。

任务分类表：标明不同任务的功能及使用状况。

数据操作特征表：标明任务与数据间联系及不同数据的不同操作特征与执行频率。

操作过程说明书：标明各任务的主要逻辑执行步骤及程序编制的有关说明。

数据字典

数据分类表。

数据元素表。

各类原始资料，即收集所有单据、报表、文件及设计所需的原始资料，并根据数据分类表的数据标识统一分类编号。

数据量

数据约束

数据约束指应用对数据的特殊要求。它主要有下述几个方面：

数据的安全保密性。

数据的完整性，主要指数据正确性的约束范围和验证准则及一致性保护的要求。

数据的响应时间，指某些特定应用要求的数据存取时间限制。

数据恢复，指转储及恢复的时机与范围等要求。

上一节　本书简介　下一节

作者：希赛教育软考学院　来源：希赛网　2014年01月27日

确定设计目标

数据库设计的首要任务是要确定应该支持的应用功能和应用领域。理想情况下，最好能覆盖一个企业的所有应用领域并尽可能地考虑到将来的应用需求。但由于财力、技术及管理基础等因素，这样做往往行不通，实际开发的数据库系统一般只包含基本的应用功能。因此首先要确定一个合理的需求目标，据此划定数据库设计的范围，即涉及的应用领域和部门。为此，可引用前面系统分析中的有关成果，如企业的职能机构与确定的目标有关的功能域等。

作为企业或部门，最好有自己的总体"信息计划".计划范围可以很广，包括当前的和未来的信息系统战略、每个子系统的功能、各子系统间的相互依赖关系及数据分类情况。以此计划作为指南，就能很快地确定数据库的设计范围。如果企业或部门没有"信息计划"或计划没有足够的信息，开发人员就必须通过实地调查研究，弄清现行系统的组织结构、功能划分、总体工作流程、系统目标、存在困难与问题等，确定数据库的设计范围。其主要方法是根据最高决策机构的意图，了解数据库所涉及的部门和信息主题，向有关部门及领导进行调查，确定数据库的服务范围。

确定数据库服务范围时一定要考虑将来需求的变动情况，这种变动可能来自将来业务的调整、经营策略的改变、政策环境的变化，等等。对于可能影响数据库的因素，就应把它纳入设计范围，使数据库适应这些变动。

数据收集与分析

确定数据库服务范围后就可着手进行系统需求信息的收集工作。信息收集所采取的方法主要有以下几种。

向各级用户发放问题调查表，以了解各部门所涉及的所有业务的工作名称、功能、目标结果等，形成工作总表。

查阅各有关部门现成的数据文档。这里文档包括表格、票据及报告、目前所使用的数据文件、旧系统的一切文档等。

与用户交谈，进一步明确每一业务的功能、处理过程、业务所需数据元素及各元素间的依赖关系、业务执行的规律、与其他业务的联系等。交谈对象必须具有代表性、普遍性，既可以是各业务的代表，也可以是各级管理决策者。交谈的方式可不拘一格。

所有信息收集工作不应以任何用户观点而应以数据观点来进行，这样可更方便、准确地进行数据分析整理。信息的收集整理工作包括以下几步：

①编辑整理出所有系统产生和使用的数据元素。

②定义所有业务中的事务及其所用到的数据。事务是由操作序列组成的、完成某一特定任务的、不可分割的工作单元。

③找出明显的或隐含的操作规则和政策。

④列出将来可能的变动情况及其对系统的影响。

为了高效地进行需求分析，必须采用好的工具和方法。结构化分析方法就是一种广泛应用的需求分析方法。它是以数据流图为主要工具、逐步求精地建立系统模型的一种系统分析方法。这种方法还采用了如数据字典、判定表、判定树等一些辅助工具，这些工具是配合使用的。

有关结构化分析方法及其数据流图与数据字典的有关介绍请参考相应章节的详细介绍，这里不再赘述。要说明的是，作为结构化分析方法的一个有力工具，其中的数据字典与DBMS中的数据字典在内容上有所不同，DBMS数据字典是用来描述（或）定义数据库系统运行所涉及的各种对象的，但在功能上它们是一致的。利用数据字典可更方便地管理软件开发过程或数据库系统，提高软件开发效率或数据库系统的运行效率。

下面对面向数据的方法做一介绍。

面向数据的方法

面向数据的需求分析方法侧重于数据对现实世界的描述作用。它从分析目标功能域中的管理活动开始，弄清楚各种不同类型管理活动中涉及的实体、实体间的联系及它们的特性，从而得到一个描述这些实体、实体间联系及它们特性的数据集。然后用目标范围内的数据处理要求对该数据集进行验证，并加以必要的补充修改，最后整理成完整的数据库设计需求说明书。这种方法的基本步骤如下。

1.分析管理活动类型，确定调查对象

管理活动可分为例行事务活动和计划与控制活动两大类。例行事务活动都是周期性地（每日、每周或每月）重复的活动，一般都有固定的程序，所需的数据也均有一定的格式。如企业中的质量管理、成本管理、供销活动等均属例行事务活动，这类活动均属于各层次事务处理人员的职责范围。计划与控制类活动都与做出决定有关，如决定整个组织的目标、经营方向、方针，确定产品品种、销售策略、库存控制策略等，这类活动主要由高层与中层管理人员执行，它们的特点是，大都没有确定的程序，需要的数据也无固定的格式。由于这两类活动有着不同的特点，所涉及的数据对数据库设计具有不同的意义，因此在对管理活动开展调查活动之前，必须首先识别这两类活动。对这两类管理活动进行调查的一种行之有效的方法是让有关的管理人员填写询问表，询问表的格式可见表23-2.

表23-2 业务活动询问表

部门：书库管理 填表人：王×× 职务：库管理员

业 务 项 目	具 体 活 动
1．发运	接收装运单、审核、对合格装运单打包、退回不合格装运单
2．入库	收入库单、审核、质检、入库登记
3．清仓盘点	清点库存、核对台账、……
……	

填表者只要简明地填上自己负责的事务功能名称及为完成该功能需要做的事情。这里仅仅要求说清楚做什么，无须说明如何做的细节。这样的询问表必须发到每一个岗位上的管理人员手中，并及时收回。对收回的活动询问表进行分析，按活动的性质区分出两类不同的管理活动，并做适当标志，再分别汇总。表23-3给出了活动汇总表的示例。

表23-3 活动汇总表

例行事务类活动			计划与控制类活动		
部 门	业务项目	管理人员	部 门	业务项目	管理人员
销 售	订单处理	周×× 张××	销 售	制定销售计划	王××
……	……	……	……	……	……
物资供应	库房管理	赵××	生产计划	制定生产计划	吴××
……	……	……	……	……	……

根据这张汇总表不难找到两类管理活动相应的调查对象，然后对他们分别进行调查。

2.对例行事务活动的调查

例行事务活动是一个单位运行中的基本事务活动，是发生和汇集所有原始数据的处所。它们所涉及的数据对象及其联系，表示了一个单位的基本状态，只要单位的性质不变，表示其基本状态的数据对象及其联系也基本不变。因此，对这些活动调查所收集到的数据是设计一个数据库的主要原始材料，是建立一个稳定数据库的基础，因而对这类例行事务活动的调查必须十分细致和全面。

调查工作可按前面所讲的基本任务展开，通过调查应该明了以下内容：

执行每项事务功能的全部活动过程及相关的实体、实体间的联系和相应的事务规则，收集有关

的单据、票证等数据载体。

每项事务对数据的使用要求和使用方法。

调查可按下述步骤进行。

第一步，详细调查业务过程。方法是让被调查者详细地描述他们每天例行的业务过程，调查者可用录音（或自然语言）加以记录，也可采用绘制流程图的办法加以记录，或者先录音，以后再整理成流程图。采用自然语言描述这些业务过程时，调查记录表的右边要留有空栏，以便在分析过程中做注释。

下面是以自然语言描述的一个出版社邮购部的订单处理业务。

业务名称： 订单处理

本项业务处理客户邮来的订单．在接到客户订单以后，首先登记流水账，然后审阅订单，检查每一个重要数据项是否遗漏或模糊，所订的图书是否属停止出版发行，必要时报请主管领导鉴别确认，当无法确认时，把订单退回给客户并指明问题。如订单附有付款，则将附有付款的订单副本送财务处入账；对没有付款的订单，检查客户文件，判断客户是否属于已经建立信用的个人或团体，若没有建立信用，则填发收到订单的回执及付款通知。对于一个新客户，应成立户头，立即将该客户的有关信用添加到客户文件中，对已经付款或信用良好的订单检查书库的库存账，看是否能满足所有订购数要求，若可满足，填写发运单及发票（对已付款者，加盖收讫字样）。它们将和图书一起发运。若订单只能满足一部分，则只填写可满足部分的发运单及已付款的发票，附上一份未满足部分的回执。将暂时缺货的图书，填写到拖欠订书单账上，一旦这些图书从印刷厂运到，立即交付拖欠订书。对于已经售完的图书的订单，待积满一定数量后，再要印刷厂重印……	

底线含义：__表示实体，~~~表示联系，====表示数据载体，E_i表示实体，R_i表示联系（i=1~4）。

对于以周、月、季度或年为重复周期的事务活动可采用同样的方法实施调查。

第二步，识别实体、实体间的联系，收集相关的数据载体。一般说来，每个事务活动总要涉及一种或多种实体，如签订一份供货合同，就涉及到供货者、购货者及供应的物品3类实体。一个活动涉及到多种实体，也就意味着这些实体间存在某种联系。事务活动涉及实体或实体间联系的方式不外乎以下几种：

删除一个实体，如删除某个客户或某类物品等。

修改实体描述信息，如修改客户地址、物品的规格、数量等。

查问某个实体的情况，如查阅某个客户出版社的情况等。

发生了某种有多个实体参与的事件，如签订购销合同或取消某份合同等。

根据上述规律，对前面得到的调查材料进行详细分析，逐一识别每一基本事务活动中涉及的实体、实体间的联系，分别用记号E、R在流程图上或自然语言描述的注释栏内加以标记，并另外列表汇总。汇总表包括实体或联系的标志号、名称及相关的数据载体等，对于联系还须说明有关的事务规则，事务规则是单位执行一种事务的规定和法则。图23-3（a）、（b）分别是实体汇总表和联系汇总表的示例。

实体汇总表

标志号	实体名	有关的数据载体
E_1	客户	客户文件，订单，发运单，发票……
E_2	图书	书库目录，台账，订单……
E_3	书库	订单，发票，台账……
……		

（a）

联系汇总表

标志号	联系名	参与的实体	数据载体	事务规则
R_1	订书	出版社、客户、图书	订单	出版社有多个客户，一个客户可订购多种图书，一种图书可被多个客户订购
R_2	发运	客户、图书	发运单	每次对客户至少发送一种图书，一种图书可对多个客户发运
……	……	……	……	

（b）

图23-3 实体汇总表和联系汇总表示例

第三步，调查各类用户（部门）的数据处理要求，其内容包括：

处理方式（联机或批处理）

处理内容

处理频度

数据量

安全保密要求

下面就上例给出一个简略的调查示例。

（a）处理要求

联机事务：

用户组A:订单处理。

A_1. 登录新的订单（订单号，客户名，地址，订购书名，书号，种类，数量，付款情况，交书日期）（100份/天）。

A_2. 查找订购某一种类图书的所有客户（不经常）。

……

用户组B:书库管理。

B_1. 登录新书（书号，书名，作者，出版日期，价格）（200本/天）。

……

批处理：

A_{14}. 按周统计订货情况：

书号	书名	种类	总的订购数	与上周相比的变化率

……

（b）数据量（按最大可能估计）

客户数：400个

库存书的种类数：200种

每月订单数：1000张

……

第四步，识别例行事务的可能变化，如满足客户新的查询要求等。将这些可能有的要求也用第三步中的方式一一加以描述，在描述条目的编号前加上字母A以表示另加的条目。如：

AA_1:询问某本书的卷册数、页数（10/天）。

AA_2:询问某本书的版本数（10/天）。

AA_3:询问某一主题的书目（20/天）。

第五步，收集数据载体，识别数据元素。这一步主要工作是两项：将所有有关的数据载体，如订单、发票、发运单、台账、各种凭证及所要求的统计报表等收集并整理成册；另一项就是收集有

关实体及实体联系的特性，参观有关的数据载体，识别描述这些实体、实体间联系的数据元素及这些数据元素本身的特点。这里数据元素是指不可分割的数据项，有时就称数据项。数据元素的特点包括它的类型、值、值域及对识别一个实体或联系的作用等。在收集和描述实体的数据元素时，应从全局而不是从某项特定的应用来考查实体应该具有的性质。

第六步，验证需求的可满足性。对于上面收集到的基本数据集，对所有需求逐一加以验证，看其是否均能支持，即考查每一需求所要的数据是否都有了，必要时加以补充或调整。例如在核对某一联机事务时，发现该需求必须知道图书的版本号，所以在对图书的描述中应加进"版本号"的数据元素。在验证时，应特别注意因事务级上可能的变化而引起的新的数据要求，如在第四步识别到的AA_1、AA_2、AA_3,则图书的描述数据还应增加卷册数、页数及主题词等数据元素。

3.对计划与控制类活动的调查

这是一类与决策有关的活动，调查对象是单位的高层和中层管理人员，需要收集的信息应包括：

单位的运行方针（明显的及隐含的）。

每一个计划或控制活动的定义。

执行计划或控制活动的定义。

单位的发展规划或设想。

进行这一类活动的调查，并无严格的程序，但可参照对例行事务活动的调查方法，尤其是对那些程序化的或确定型的决策控制活动可按前面的调查流程进行。

通过对上述调查材料的分析，可能识别出新的数据需求、新的数据实体、实体间的联系和相关的事务规则，估计未来的变化对数据库的影响。根据这些分析结果，对前面收集的基本数据集做出必要的补充或调整。

例如，对一个出版社的上述两类管理人员进行调查，可能得到下述信息：

同一客户一次可订购多种图书，一次订购的图书可能分多次发运。

出版社对发行量超过1万册的图书的作者实行奖励。

出版社对客户实行分级的批量优惠政策。

正在考虑为顾客增设新的咨询服务项目，回答客户有关出版社出版的图书的情况、发运情况、供应点情况及优惠政策等询问。

数据流方法（又称结构化分析法）和基于数据的方法（data-based methodology）的主要差异是完成系统描述的技术不同。结构化系统分析法由定义系统功能和数据（信息）流开始，再由这些系统功能和数据流产生数据描述和数据模型或语义模型。面向数据的方法或基于数据的方法，则是首先定义数据结构和数据模型，然后再定义系统功能。这两类方法的最终系统描述都包含了系统中的数据流向、数据加工中的处理类别和数据本身的结构。

上一节　本书简介　下一节

数据库设计遵循软件工程的原则和方法，其需求分析阶段的文档工作主要是以下几点。

1.整理调查材料

由于调查分析工作可能分成若干调查组同时进行，同一项业务有不同调查对象，也可能分批分次进行调查，这样汇集的调查材料难免有不一致、不规范及粗糙、模糊等问题，必须进行整理，使之成为正规、系统、清晰可读的资料。整理工作包括：

检查材料的完整性和系统性，即检查各项调查内容是否齐全，有关数据载体（单据、票证、表格等）是否收集齐全，应建的文档是否都建了，逻辑顺序是否合理，必要时加以追补和修改。

消除模糊点，修改错误，必要时可再次访问调查对象或请示主管领导。

消除命名冲突和冗余。命名冲突指各类数据的同名异义（homonym）及同义异名（synonym）问题。对于同名异义问题可通过鉴别同名数据元素的语义加以识别，然后加以消除。对于同义异名的识别则要困难一些，办法之一是对数据元素按语义进行分类，将语义相近的数据元素归在一起，以便缩小搜索范围。例如按照实体名、日期、总计、单据等分类。类别分得越细，鉴别搜索范围就越小，然后再对小范围中不同名的数据元素的语义逐个鉴别其是否属同义。在此过程中也能识别出冗余数据，一旦发现应将其除去。

对已经识别的上述两类冲突，分别列表存档，留待以后进一步分析处理。

保持文档格式的一致性和规范化。

对所有原始材料及所做的文档进行编码。

2.用户审核确认

以上整理出来的调查分析材料仅仅表示了分析人员对单位的一种理解，这种理解只有和用户的想法相一致才是有效的，因此必须有一个统一认识的过程，这个过程就是用户审核确认的过程。用户审核确认的主要内容包括：

确认数据库的设计目标及范围是否合理。

验证每个业务功能流程的正确性及相关的数据对象是否齐全。

检查有关的事务规则是否正确完整。

数据的处理要求是否完全。

是否考虑到将来可能的变化或发展。

由于调查过程的复杂性，调查材料中不可避免地存在着误解与遗漏，如不及时加以解决，可能会有大的反复，所以用户的审核和确认至关重要，关系到数据库系统能否成功运行，必须给予足够的重视。

3.建立数据库设计的需求说明文档

用户对调查分析材料审核和确认后，就可着手建立数据库设计的需求说明文挡，即需求说明书。该需求说明书不是囊括全部调查分析材料，而只是选择对数据库设计直接有用的信息，一般包括实体类、联系类、数据的使用要求及冲突表等方面的内容。

（1）实体类：列出涉及的所有实体类并加以描述。

实体类名称及其语义说明，如图书这个实体类包括出版社发行的所有书籍。

可能的最大实例数，如图书总数为1万册。

描述实体类的数据元素，其格式如下：

其中，简要描述是对该数据元素的语义说明，存在概率表示该数据元素出现空值的概率，重复因子表示该元素值重复出现的最大次数，若其值为1,表示不重复，可用做实体的标识；如其值大于1,

表示可能重复出现，就不宜用做实体标识。

（2）联系类：对每个联系类要说明以下几点。

联系类名称、参与联系的实体及其他语义信息。

最大的实例数。

（3）事务处理说明：主要包括事务处理的下述内容。

过程描述。

执行者。

I/O数据。

执行频率。

执行条件与后果。

（4）数据使用说明：对数据的使用要求应说明下述内容。

各种数据在相应事务中的使用类型，使用类型可为C（建立）、Q（查询）、I（插入）、M（修改）、D（删除）、N（不能操作）、ALL（Q、I、M、D），等等。可用数据-处理关系矩阵表来表示。

用户组，即参与同一类事务活动的用户，可用事务活动名标识。

处理方式，分联机处理和批处理两类。联机处理如即席查询和更新等操作；批处理如各种例行报表的生成等。

数据使用的描述，说明数据在事务处理活动中被加工处理的过程。

数据的使用频度，按每日、每周使用数据的平均数和最高数表示，低于每周（月、日）一次者，标以不经常。

（5）冲突表：将已经识别的命名冲突用如图23-4所示的表格列表说明。

同名异义表

数据名称	语义说明	文档号

（a）

同义异名表

数据名称	语义说明	文档号
……	……	

（b）

图23-4 冲突表

（6）运行需求说明 用户在数据库系统运行时对数据操作方面的要求说明。主要包括以下内容。

安全与保密性要求。

完整性要求。

响应时限要求。

后备与恢复要求。

系统扩展要求等。

作者：希赛教育软考学院 来源：希赛网 2014年01月27日

概念结构设计

概念结构设计阶段所涉及的信息不依赖于任何实际实现时的环境，即计算机的硬件和软件系统。概念结构设计的目标是产生一个用户易于理解的，反映系统信息需求的整体数据库概念结构。概念结构设计的任务是，在需求分析中产生的需求说明书的基础上按照一定的方法抽象成满足应用需求的用户（单位）的信息结构，即通常所称的概念模型。概念结构的设计过程就是正确选择设计策略、设计方法和概念数据模型并加以实施的过程。

上一节 本书简介 下一节

作者：希赛教育软考学院 来源：希赛网 2014年01月27日

概念结构

概念模型是现实世界到机器世界的一个过渡的中间层次，它有以下特点。

概念模型有丰富的语义表达能力，能表达用户的各种需求，是对现实世界的抽象和概括，它真实、充分地反映了现实世界中事务和事务之间的联系，能满足用户对数据的处理要求。

易于交流和理解。由于概念模型简洁、明晰、独立于机器，是数据库设计人员和用户之间的主要交流工具，因此可以用概念模型和不熟悉计算机的用户交换意见，使用户能积极参与数据库的设计工作，保证设计工作顺利进行。

概念模型易于更改。当应用环境和应用要求发生变化时，能方便地对概念模型修改和扩充，以反映这些变化。

概念模型很容易向关系、网状、层次等各种数据模型转换，易于导出与DBMS有关的逻辑模型。

概念数据模型的作用是：提供能够识别和理解系统要求的框架；为数据库提供一个说明性结构，作为设计数据库逻辑结构即逻辑模型的基础。

概念模型的描述工具应该能够体现概念模型的特点，如E-R模型。近年来，由于面向对象数据模型具有更丰富的语义、更强的描述能力而越来越受到人们的重视，不但出现了商品化的面向对象DBMS,而且开始实际应用于概念模型的设计中，作为数据库概念设计的工具。Teory等人提出的扩展的E-R模型（称为E-E-R模型）增加了类似面向对象数据模型中的普遍化和聚合等语义描述机制，为这种最为人们熟悉的数据模型注入了新的生机，为概念模型的描述增加了一种理想的选择。

概念结构的设计策略主要有自底向上、自顶向下、由里向外和混合策略。在具体实现设计目标时有两种极端的策略或方案，一是建立一个覆盖整个单位所有功能域的全局数据库，称为全局方案（global approach）或全局策略；另一种则是对每一个应用都建立一个单独的数据库，称为应用方案（application approach）或应用策略。

全局策略要求设计一个能够支持单位所有应用需求的单一数据库，此法在数据库应用早期曾受到广泛的推崇。但实践证明，对于一个较大的单位，采用此种策略设计数据库，会使设计任务十分

复杂，开发周期过长，因而很难取得成功。应用策略则对每个应用单独设计一个数据库，优点是简化了分析工作且单库单用，运行效率较高，但也存在明显的缺点，会造成大量数据冗余、数据的不一致性、破坏整个单位中数据的完整性及难以实现数据共享。因此，这一策略与传统的文件系统并无本质上的差别，从而与全局策略一样可能导致失败。

介于全局和应用两种策略之间的一个折中的策略是根据对事务活动的分析，先设计出一个单位的初步信息结构，该结构表示单位活动中涉及的主要实体、实体间的联系，然后根据对信息流的分析，按自然合理的分组原则将这些实体及实体间的联系分别组成若干便于管理的较小数据库，再进行每个数据库的详细设计。

折中策略在大多数情况下提供了一种最好的选择。需要注意的是，对初步信息结构中实体及实体联系的划分不应依赖于某个应用及职能部门的界限，可以跨越多个应用域或职能部门，否则就变成了应用策略了。此外，对于信息结构不太复杂的较小单位，也可以考虑选择全局策略。

上一节　本书简介　下一节

作者：希赛教育软考学院　来源：希赛网　2014年01月27日

概念结构设计的方法和步骤

概念结构设计常用的方法是实体分析法（entity analysis）和属性综合法（attribute synthesis）。

实体分析法又称自顶向下（top-down）方法。它从总体概念入手，从分析一个单位的事务活动开始，首先识别用户所关心的实体及实体间的联系，建立一个初步的数据模型框架，然后再用逐步求精的方法加上必需的描述属性，形成一个个完整的局部数据模型，称为用户视图，最后再将这些视图集成为一个统一的数据模式，称为用户视图的集成，这种统一的数据模式（即全局信息结构）通常用E-R图表示。可见，实体分析法是通过3个不同的步骤完成概念模型设计的：建立用户视图（局部信息结构），视图集成为统一的数据模型（全局信息结构），用E-R图描述全局信息结构。这种方法的优点是：减少了分析中所涉及的对象数，简化了分析过程，且采用图形表示法，因而更为直观，易于理解，有利于用户在设计过程中的介入，所以被广泛采用。

属性综合法又称自底向上（bottom-up）方法。其基本点是将需求分析中收集的数据元素作为分析对象，高层实体及联系通过低层属性综合而成的设计技术。实际上，这是一种基于统计分析推导的方法，即通过对数据元素与应用任务联系的定性定量统计分析技术来推导出相应的信息结构。其处理过程可分为属性分类、实体构成、联系的确定等相对独立的步骤。这种方法适用于较为简单的设计对象而不适于稍为复杂的应用环境，因为要对几百甚至数千个数据元素进行综合处理就不容易，何况在数据分析的前期阶段，数据库管理员通常并不完全清楚在数据库模式中究竟包含哪些数据元素。

上一节　本书简介　下一节

作者：希赛教育软考学院　来源：希赛网　2014年03月13日

数据抽象和局部视图设计

数据模型是数据库系统的核心和基础，各种机器上实现的DBMS软件都是基于某种数据模型的且有一个共同的特点，因为它们是在具体的机器上实现的，所以在许多方面给出了细致严格的限制。而现实世界中应用环境是复杂多变的，各种事务的表现形式也与机器世界中相距甚远。在进行数据库设计时，如果将现实世界中的客观对象直接转换为机器世界中的对象，就会感到非常不方便，注意力往往被牵扯到更多的细节限制方面，而不能集中在最重要的信息的组织结构和处理模式上。因此往往是将现实世界中的客观对象首先抽象为不依赖任何具体机器的信息结构，这种信息结构不是DBMS支持的数据模型，而是概念级模型。然后再把概念级模型转换为具体机器上DBMS支持的数据模型。

由于各个部门对于数据的需求和处理方法各不相同，对同一类数据的观点也可能不一样，它们有自己的视图，所以可以首先根据需求分析阶段产生的各个部门的数据流图和数据字典中的相关数据设计出各自的局部视图。

在实体分析法中，局部视图设计的第一步是确定其所属的范围，即它所对应的用户组，然后对每个用户组建立一个仅由实体、联系及它们的标识码组成的局部信息结构（局部数据模式）框架，最后再加入有关的描述信息，形成完整的局部视图（局部数据模式）。这样做的目的是为了集中精力处理好用户数据需求的主要方面，避免因无关紧要的描述细节而影响局部信息结构的正确性。整个过程可分为以下几个步骤：

①确定局部视图的范围。

②识别实体及其标识。

③确定实体间的联系。

④分配实体及联系的属性。

1.确定局部视图的范围

需求说明书中标明的用户视图范围可以作为确定局部视图范围的基本依据，但它通常与子模式范围相对应，有时因为过大而不利于局部信息结构的构造，故可根据情况修改；但也不宜分得过小，过小会造成局部视图的数量太大及大量的数据冗余和不一致性，给以后的视图集成带来很大的困难。

局部视图范围确定的基本原则是：

各个局部视图支持的功能域之间的联系应最少。

实体个数适量。一个局部视图所包含的实体数量反映了该局部视图的复杂性，按照信息论中"7±2"的观点，人们在同一时刻可同时顾及的事情一般在5~9之间，以6或7最适当。因此，一个局部视图内的实体数不宜超过9个，否则就会过于复杂，不便于人们理解和管理。

2.识别实体及其标识

在需求分析中，人们已经初步地识别了各类实体、实体间的联系及描述其性质的数据元素，统称为数据对象，它们是进一步设计的基本素材。这一步的任务就是在确定的局部视图范围内，识别哪些数据对象作为局部视图的基本实体及其标识，定义有关数据对象在E-E-R模式中的地位。

1）数据对象的分类

为了确定数据对象在局部E-E-R数据模式中的地位，首先必须对所有已识别的数据对象加以适当

的分类，以便于根据它们所属的对象类来确定它们在相应局部E-E-R模式中的身份。数据对象分类的原则是同一类中的对象在概念上应该具有共性。例如高校中的教师这个概念是指在职的教学人员，他们的性质由姓名、性别、出生年月、工作单位、职称、专业特长等数据项加以描述。因此，教授、副教授、讲师和助教均可归入教师这一类，但他们在概念上并不完全相同，除了共性之外，还各有其特殊部分，如教授、副教授有研究方向、指导研究生等描述项。职称也不一样，因此存在分类层次问题，为此可运用面向对象数据模型中子类与超类的概念。

2）识别实体与属性

建立局部E-E-R数据模式，还须识别每个对象类在局部E-E-R模式中的地位：实体、属性或联系。实体和属性之间在形式上的界限并不明显，常常是现实世界对它们已有的大体的自然区分，它随应用环境的不同而不同。在给定的应用环境中区分实体和属性的总的原则是要在此应用环境中显得合理，且同一个对象类在同一局部视图内只能作为一种成分，不能既是实体又是属性或联系。此外，为了对已给定的需求目标，更合理地确认局部E-E-R模式中的实体和属性，以便在逻辑设计阶段从E-E-R模式得到更接近于规范化的关系模式，可按下述一般规则来区分实体与属性。

（1）描述信息原则

在E-E-R模式中的实体均有描述信息，而属性则没有。据此，可将一个需要描述信息的对象类作为实体，而将只需有一个标识的对象类归为属性。例如，仓库这个对象类在某些事务处理中需要用到仓库的面积、地点、管理员姓名等描述信息，则宜将其归入实体。但如人的年龄、物品的重量等对象类，在一般应用中都不需要描述信息，所以它们在通常情况下都作为属性。

（2）多值性原则

所谓多值性是指若一个描述存在多个值描述其描述对象，则即使该描述项本身没有描述信息也宜划为实体。例如加工种类与其描述的工件之间就符合多值性原则，因为每个工件实体可能需要多个工种的加工，尽管工种除了工种名外不需要其他的描述，但还是将工种作为实体好。需要注意的是，这一原则最好与存在性原则结合起来使用，如果将被描述对象删除后，描述项没有再存在的意义，则即使此描述是多值的也不宜作为实体。例如零件的颜色可以是多值的，但颜色离开了其描述的对象就没有单独存在的意义，因此还是作为属性为宜。

（3）存在性原则

设对象类R的描述A和B的值集分别为$\{a_1,a_2,\ldots，a_n\}$和$\{b_1,b_2,\ldots，b_n\}$.若从R中去掉B的某个值b_i，从而去掉与它的所有联系，如果b_i的消失对应用不产生任何影响，则B作为属性，否则B应作为实体。

（4）多对一联系性

属性不再与其描述对象以外的其他对象类发生联系。相反，如果一个对象类的某个描述项与另一个对象类存在着多对一联系，则此描述项即使本身没有描述信息，也宜将其作为实体。例如前面讲的工件与加工种类的例子中，如果还有一个车间实体，加工种类与车间之间存在多对一联系，因此将加工种类划为实体更合适。

（5）组合标识判别原则

若一个对象类的标识是由其他对象类的标识组成的，该对象一般应定义为联系。相反，如果组成某对象类标识的各成分不是其他实体的标识，且作为实体在应用的上下关系中很自然，则可以定义为实体。例如，一个工厂生产的零件须由名称及规格组成组合标识，在一般应用中应将零件作为实体较为适宜。

实体与属性的识别过程是个相互作用的反复过程，随着实体的确认，属性也将趋于明朗，反之亦然。在此过程中根据应用需求对已经识别的实体和属性做适当指派，发现问题再来调整，在识别过程中必须遵循前面所讲的总原则。

3）对象的命名

在需求分析中得到的数据对象通常都是有名称的，但由于这些名称未经规范化，常常存在诸如同名异义、异名同义等许多命名上的冲突、不一致性及语义不清等问题，是造成数据不一致性及数据冗余的一个重要原因，此外，数据对象原有的名称有的过于冗长，给使用带来很大不便。为此，需要按一定的规范对每一类数据对象重新命名，给它指定一个简洁明了且唯一的名字，避免异义同名和异名同义存在。命名规范一般包含以下原则和规定。

数据对象名应清晰明了便于记忆，并尽可能采用用户熟悉的名字。

名字要反映数据对象的主要特点并力求简洁，以利于减少冲突和方便使用。

遵守缩写规则。缩写规则包括一般缩写规则和系统中专用名称的缩写规定，对于较大的复杂系统应编制缩写字典，便于参照。凡有缩写规定的不得使用全称，以免混淆。

规定统一的命名约定并加以遵守。例如对属性的命名可以采用如下形式的约定：

实体名·分类词--修饰词

其中，分类词指单位内通用的数据项名，如名称、号码、小计、总计、合计、单位、摘要、日期、时间，等等。每个单位应有一个标准的分类词表，加于分类前面的实体名和后面的修饰词可用以说明该分类词在特定的上下文中具有的特殊含义。例如，合同上的数据项日期可命名为：

合同·日期--年月

制定命名约定的基本原则是简明一致、避免混淆，具体可根据单位内数据对象的复杂程度自行制定。

在完成了实体与属性的识别后，必须按照命名规范仔细地审核每类数据对象的名字，纠正不符合规范的命名，务必使每个对象名符合规范要求。

4）确定实体的标识（identifier）

实体的标识是能够唯一地标识一个实体的属性或属性组，也就是该实体的关键字。确定实体标识在实体识别与规范化命名之后进行，首先确定每个实体的候选关键字。一个实体可能有若干候选关键字，选择其中对有关用户最熟悉的一个候选关键字作为主关键字（或主码），并将每个实体的候选关键字、主关键字记入数据字典。

3.确定实体间的联系

实际上，识别联系的主要任务是在需求分析阶段完成的。这里的工作一是从局部视图的角度进行一次审核，检查有无遗漏之处；二是确切地定义每一种联系。

现实世界中的诸多形式的联系大致可分为3类：存在性联系、功能性联系和事件联系。

存在性联系如学校有教师，教师有学生，工厂有产品，产品有顾客等；功能性联系如教师讲授课程，教师参与科研，仓库管理员管理仓库等；事件联系如学生借书、产品发运等。

根据上述分类仔细检查在给定的局部视图范围内是否有未识别的联系，在确认所有的联系都已识别并无遗漏之后，还须对联系进行正确的定义。定义联系就是对联系语义的仔细分析，识别联系的类型，确定实体在联系中的参与度。

1）二元联系的类型与定义

二元联系是指两个实体类之间的联系。根据参与联系的两个实体类值之间的对应关系分为一对

一、一对多及多对多3种类型。对每一种联系类型，要确定实体在联系中的参与度，并以m:n的形式标在E-E-R图上要说明的实体旁。若m>0,表明该实体参与联系是强制性的，若m=0则是非强制性的。下面分别讨论上述3类联系的定义。

（1）一对一联系

这是一种最简单的联系类型。若对于实体集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具有一对一联系，记为1:1.例如在一个施工单位中，如果规定每项工程最多只能由一名工程师负责管理，而一名工程师最多也只能负责一项工程，则工程师与工程间的这种管理联系便是一对一联系。按照实体参与联系的强制性情况，又可分为以下3种情况。

（a）两类实体都是强制性的

假如规定每个工程师一定要负责一项工程，每项工程也一定要有一位工程师负责，便属于此种情况。如果工程师的标识为职工号，工程的标识为工程号，对于工程师与工程间的1:1联系，可用职工号或工程号作为标识。

（b）其中仅有一类实体是强制性的

若规定每项工程必须由一名工程师负责，但并不是所有工程师都必须负责一项工程（因为有可能没有那么多工程），此时，每一项工程一定对应着唯一负责联系，所以工程号可用做联系的标识。

（c）两类实体均为非强制性的

如果工程师不一定安排负责管理工程，有的工程项目暂时可以不安排工程师负责管理，这种情况表示凡分到工程项目的工程师与工程项目之间总存在一一对应的联系，因此职工号或工程号均可选为负责联系的标识，定了其中一个为标识，另一个就作为候选关键字。

（2）一对多联系

若对于实体集A中的每一个实体，实体集B中有n个实体（n≥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有一对多的联系，记为1:n.以专业与学生间的关系为例，如规定一个专业可以管理许多学生，每个学生只能属于一个专业，这种联系就是一对多联系。对这种联系我们只须关心"多"端实体的强制性，分两种情况进行讨论。

（a）"多"端的实体是强制性的。

此时，每个学生必须归属某个专业，即每个学生都有一个确定的专业，但每个专业都不唯一地对应一个学生，故可以选择学号作为联系的标识。

（b）"多"端的实体是非强制性的。

对本例而言系指有些学生（如新生）不属于任何专业的情况。此时实际上仅表示已经分配专业的学生与专业之间的联系，对这些学生中的每一个都有一个确定的专业，因此，应以学号为联系的标识，而专业代号只作为联系的一般属性。

（3）多对多联系

若对于实体集A中的每一个实体，实体集B中有n个实体（n≥0）与之联系，反过来，对实体集B中每一个实体，实体集A中也有m个实体（m≥0）与之联系，则称实体集A与实体集B具有多对多联系，记为m:n.教师与学生这两个实体类间的教与学的联系就是多对多的联系。这时，只有<教师，学生>对才能确定一个特定的教学联系。因此，一般情况下可以用两个关联实体的标识拼凑

（concatenate）作为联系的标识，但这种方法对某些情况就不能构成有效的联系标识。当一个实体值在同一个联系上可能存在多个不同的联系值时，就会出现这种情况。如教师与其讲授的课程之间的联系，同一个教师可讲授几门不同的课程，也可以多次讲授同一门课程，这是一种特殊的多对多联系，这种情况可用如图23-5（d）所示的值图（occurrence diagram）表示。值图是表示具体的实体及其关联的一种图示法，其中圆点表示具体的实体，连线表示实体间的联系。而通常的E-E-R图或E-R图只能表示型而不表示值，所以称为型图（type diagram）。显然，对于与讲授课程间的联系，如在教师档案中要求记录担任教学工作的情况，就需要在联系标识中增加表示授课日期的属性，即其合适的联系标识可能为（教师号，课程号，授课日期）。

（4）实体类内部的联系

这种联系发生在同一类实体的不同实体之间，因此称为内部联系或自联系，它也是一种二元联系，其表示方式与前面的二元联系并无不同，要注意的是仔细区别同一实体类中的不同实体在联系中扮演的不同角色及联系标识的选择。例如在职工类实体中间就存在着管理者与被管理者的联系。一个职工可以管理别的职工，称为管理者或领导者。一个管理者可以管理多个职工，而一个职工最多只从属于一位管理者，从而构成了一对多联系。若规定所有职工都要受管理（最高管理者考虑自己管理自己），但不是所有职工都是管理者，则此联系在"多"端呈现强制性。其中每个联系实体包含两个职工号值：职工号和管理员职工号，以区别不同实体在联系中的角色。

若略去实体与其属性图，以上实体间的联系可用图23-5所示。

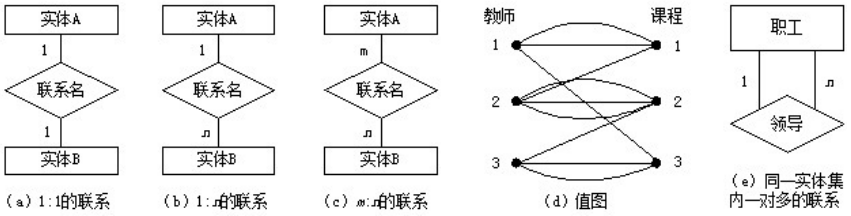

图23-5 实体间的联系

2）多元联系的识别与定义

两个以上的实体类之间的联系称为多元联系。例如在供应商向工程供应零件这类事件中，如果任一供应商可向任一工程供应任一种零件，则为了确定哪个供应商向哪个工程供应了何种零件，就必须定义一个三元联系，因为只有供应商、工程及零件三者一起才能唯一地确定一个联系值。其联系的标识由参与联系的实体类的标识拼接而成，在此例中由供应商、工程、零件3个实体类的标识拼接而成。

需要注意的是，涉及到多个实体的事件是否属于多元联系完全取决于问题的语义，不可一概而论。例如，如果上例中的问题说明变成每个工程需要订购一定的零件，而任一供应商可向任一工程供应零件。这里有两层意思，一是只有工程定了才能确定订购的零件，二是只有供应商及工程确定了，才能确定一个供应关系。根据这-情况，应定义两个二元联系，如图23-6（a）所示。

假如问题的说明是任一供应商向任一工程供应零件，但某个供应商向某项工程供应的零件是一定的，则在供应商与工程之间的关系确定后，供应的零件也就确定了。由此可知，只需定义一个二元联系就可以，如图23-6（b）所示，其中供应的零件作为供应联系的一个属性。

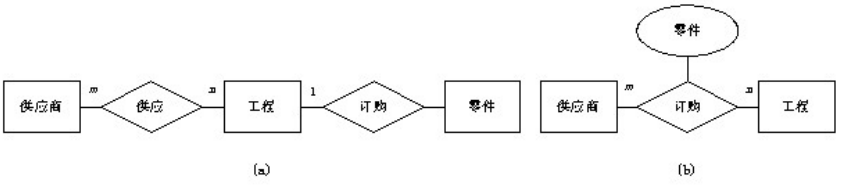

图23-6 多元联系

总之，具体问题应该具体分析，以便使定义的模式确切地表达问题的语义。

3）消除冗余联系

若出现两个或两个以上的联系表示的是同一概念，则存在着冗余的联系，具有冗余联系的E-E-R模型转换为关系模型可能会得到非规范化的关系，因此必须予以消除。

出现冗余联系的一个重要原因是存在传递联系。例如，图23-7中表示了产品与零件之间的"组成"联系，零件与材料之间的"消耗"联系及产品与材料间的"使用"联系。其中，材料与零件间的联系是1:n,零件与产品之间的组成联系是n:m,其实由这两个联系必然得出产品与材料之间的使用联系m:n.因此，图中产品与材料间的联系是冗余的，应将其去掉。

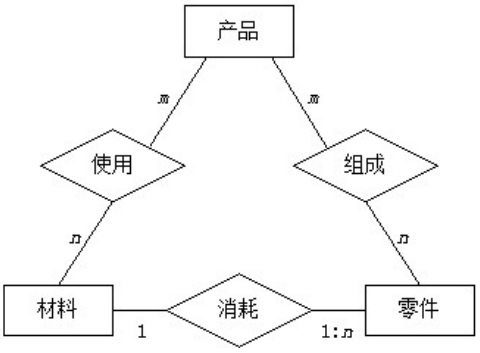

图23-7 存在冗余联系

4）警惕连接陷阱（connection traps）

连接陷阱是一种存在语义缺陷的联系结构，它是由于在定义联系过程中对语义的理解出现偏差而造成的，因而无法由它得到需要的信息。连接陷阱可分为扇形陷阱、断层陷阱和深层的扇形陷阱3种情况。

（1）扇形陷阱（fan traps）

两个实体类间的一对多联系，由一个实体值引出多个同一类型的联系值，其值图是一种扇形结构，故称为扇形联系。扇形陷阱则指由一个实体引出两种不同类型的扇形联系，形成双扇形结构。图23-8（a）是这种结构的一个例子，图23-8（b）是一个值图。从图23-8（b）值图可以看出，从这种联系结构无法获得哪个职工属于哪个专科的信息，其原因是我们将专科与职工之间的联系通过医院来连接的。如采用图23-9（a）所示的联系结构，图23-9（b）是它的值图。新的结构能较自然地表示医院、专科及职工之间的层次关联。如果我们假定任一医院的职工无例外地分属于医院的各个专科的话，该结构可以确定一个职工所属的专科或医院，但如果允许某些职工直属医院而不属于任何专科，那么这种结构还是不合适的。

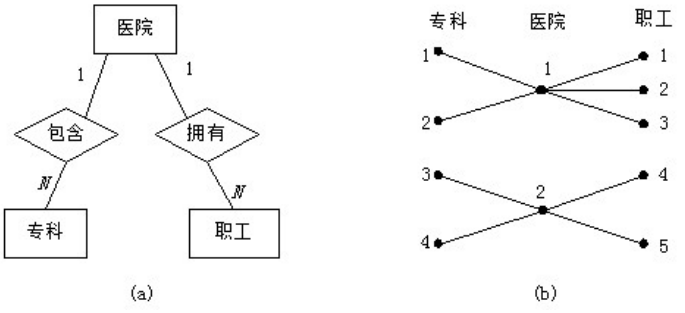

图23-8 扇形陷阱及其值图1

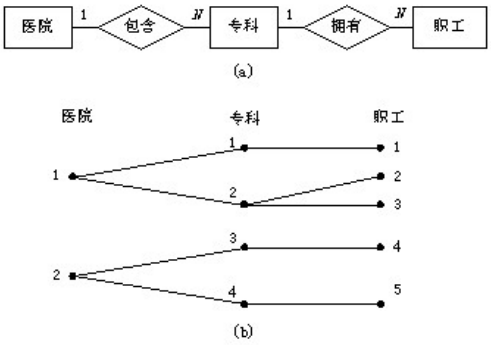

图23-9 扇形陷阱及其值图2

（2）断层陷阱

断层陷阱是指因为型图所含的传递联系而掩盖了某些特定的直接联系的现象。例如图23-9（a）的联系结构虽然隐含了医院与其职工的联系（传递联系），但却没有提供部分职工直属于医院的联系路径，因而出现了断层陷阱。解决办法是设置一个虚构的专科或增加一个联系，如在本例中增加一个医院-职工联系。增加虚构实体和现实世界不符，因此可考虑增加一个联系，但增加新的联系在某些情况也可能带来新的麻烦，见下面的讨论。

（3）深层的扇形陷阱

我们考虑一个"教师指导学生参加课题"的例子，若每个学生可在多位教师指导下参加多个课题研究，每位教师可指导多名学生，但现在只允许一名教师指导一名学生参加一个课题，不允许多名教师指导同一名学生参加某个课题。对此可先建立一个由两个多对多的二元联系组成的模式，如图23-10（a）所示。利用联系的分解法则将其分解为如图23-10（b）所示的结构，它是以学生为中心的双扇形结构，图23-10（c）是它的值图，从该联系结构无法得到哪位教师指导哪个学生参加何课题的信息，这表明存在扇形陷阱。图23-11（a）是对它的一种改进，在其中增加了一个教师与课题的联系。该联系结构能确切地提供"教师1指导学生1参与课题1","教师2指导学生2参与课题1"的信息，但从此图无法确定教师1指导学生2参与了哪一个课题。对教师2和学生l也是如此，因为参加课题l或2均是正确的语义。之所以如此，原因在于新增加的教师与课题间的多对多联系带来了两个新的双扇形结构，即以教师为中心的及以课题为中心的双扇形结构。增加的新联系虽然消除了原来的陷阱，却产生了新的陷阱。对这类问题的有效解决办法是将3个实体间的联系定义成一个单一的三元联系，它的E-E-R模式如图23-11（b）所示。现在，每一个联系值唯一地确定了另一个联系值，从中可获得哪位教师指导某学生参加哪一课题的确定信息。可见问题的实质是对于应该定义成三元联系的问题千万不能用二元联系代替。

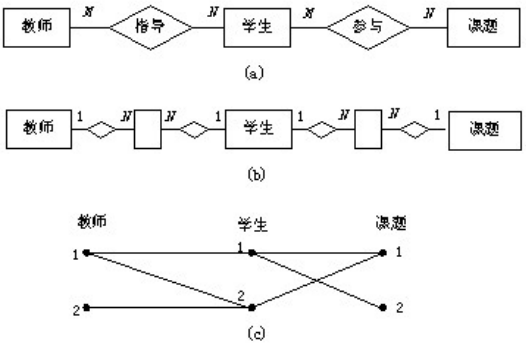

图23-10 深层的扇形陷阱

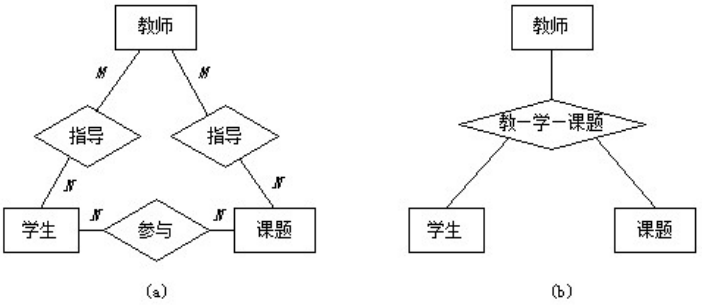

图23-11 改进的扇形结构

4.分配实体及联系的属性

在需求分析中收集的数据对象集内，除去已识别的实体、联系及标识外，剩下的主要是非标识属性。问题的实质就是将这些非标识属性恰当地分配给有关实体或联系。不过分配这些属性时，应避免使用用户不易理解的属性间的函数依赖关系及其有关准则，而应该从用户需求的概念上去识别框架中实体或联系必须有的描述属性，并按下述两条原则分配属性。

1）非空值原则

所谓非空值原则是指当一个属性的分配在几个实体或联系中可以选择时，应避免使属性值出现空值的分配方案。例如前面曾经讲过工程师负责工程的模式中，其联系是一对一且两个实体类均属非强制性的情况。现有一个属性项"施工期限",按依赖关系考虑可加给工程师、工程或负责联系三者中的任何一个，因为工程师与工程在这里是一对一的联系，工程师定了，工程的施工期限也定了。但有的工程师可能没有分配到负责工程的任务，若把施工期限作为工程师的属性，在此情况下便会出现空值；若作为工程的属性，则工程可能还未纳入计划，也会出现空值，可见将"施工期限"分配给负责联系最为适宜。

2）增加一个新的实体或联系

在分配属性过程中，有时会出现有的数据项在框架模式中似乎找不到适合依附的实体或联系的情况。对于这种情况，常常可通过在原模式中增加一个新的实体或联系加以解决。例如，图23-12表示一个病区/病人模式，这是一个一对多，且"多"端为强制性的联系，所有明显的属性均已分配完后，尚有属性项手术名及手术号待分配。这里手术名依赖于手术号，一个病人可能接受多种手术，一个病区可能接纳不同种手术的病人。在此情况下，手术号与手术名两个数据项不论作为哪个实体或联系的属性均不合适，为此可以在原来的模式中增加一个新的实体--手术，再加一个病人与手术的联系就可以了，如图23-13所示。

按上述属性分配原则建立的模式将有利于转换成规范化的关系模式。

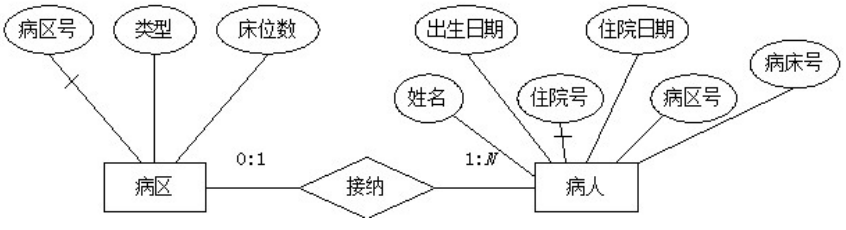

图23-12 病区/病人模式

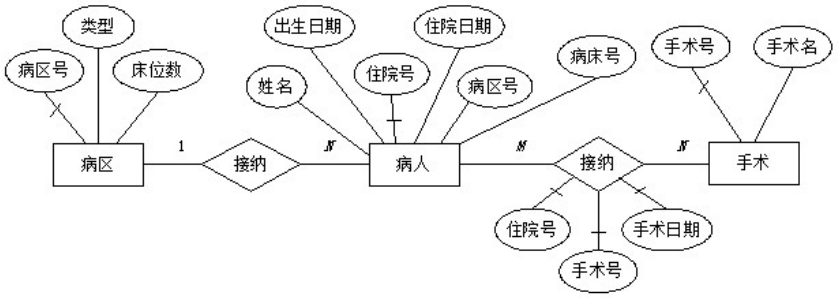

图23-13 改进后的模式

在属性综合法中，其高层的实体、实体间的联系是通过对底层数据元素的分析、归类和聚集而逐步形成的。从具体的设计过程而言，属性综合法是一种统计分析的方法，它利用统计数据元素与事务处理的关系导出结果。此法的优点是只要有一本好的数据字典和一张能反映用户需求的数据流程图，设计者即使不太熟悉业务，也能应用这种方法开展工作。其缺点是当应用环境比较复杂、数据元素较多时，很难用人工进行统计分析。

5.属性综合法构造局部E-R-R数据模式的步骤

这里简要介绍在局部视图范围确定后，属性综合法构造局部E-E-R数据模式的基本步骤，它共有5步，通常称为"五步"法。

1）数据元素的归类

数据元素归类的目标是识别实体和联系，并把属性归类成实体的属性和联系的属性。

（1）数据元素归类的基本原则

具有下述特征的数据元素一般属于实体。

被大量的事务处理所使用。

与大量的其他数据元素一起使用。

与其他单个数据元素一起使用的次数少于被所有事务处理交叉使用的总次数，亦即前者的使用率较低。

由两个以上的数据元素组成的集合，若满足下述条件，则可作为联系的候选者。

集合中的每个数据元素都可归类为实体。

集合中每个数据元素的使用频率均较高。

当数据元素集合在交叉事务处理中出现时，与其他数据元素一起使用的次数较多。

具有下述特征的数据元素，一般是一个与实体相关的属性。

使用它的事务处理相对而言较少。

它仅同少量的其他数据元素一起使用。

一般而言，与其他元素一起使用的次数与它被所有事务处理交叉使用的总次数相比，使用率是高的。

具有下述特征的数据元素，一般为与联系相关的属性。

有半数的事务处理要使用它。

与半数的其他数据元素一起使用。

它同其他数据一起使用的次数与它被所有事务交叉使用的总次数相比，使用率是平均的。

（2）数据元素的归类的具体步骤

①确定单位中事务处理/数据关系。

其基本方法是将局部视图范围内部的事务处理与涉及的全部数据元素分别编号，在此基础上分

别列出每个事务处理使用的数据元素号。

在建立事务处理/数据关系时，不要把事务处理不使用的数据元素也包括进去。对于异名同义的数据元素，应用一个具有代表性的数据元素代表之，且一个数据元素若在一个事务处理中被多次使用，在事务处理/数据关系中只写一次。这步工作最好重复一二次，以便更接近于实际情况，保证统计分析结果的正确性。

②定义事务处理使用向量、关系矩阵、全体数据关系向量。

定义事务处理使用向量

事务处理向量的初值为零，以后根据各个事务处理对数据元素的使用逐一修改事务处理使用向量。办法是：对每个事务处理使用的数据元素都要在事务处理向量的相应单元中加1,对每个事务处理重复上述修改操作，最后得到的事务处理使用向量表示了使用每个数据元素的事务处理个数。

定义关系矩阵

关系矩阵是反映各个事务处理数据元素间的所有数据关系，它的行和列均为数据元素的编号（即引用号），其初始值为0.关系矩阵的填法是：取出某一事务处理中的任一个数据元素的引用号作为行号，再分别把该事务处理中其他数据元素的引用号作为列号，将由此确定的关系矩阵单元的值加1.例如，设有事务处理1的数据元素是：1、3、7、14、20、2l,在关系矩阵中的填入过程是：将数据元素1的引用号1作为关系矩阵的行号，再把事务处理的其他数据元素引用号3、7、14、20、21作为列号，将关系矩阵中相应单元的值加1.然后，再填入数据元素3的关系，方法同前，依次类推，直到事务处理1中的所有数据元素的关系填完为止。在事务处理1的数据元素填完后，用同样的方法依次填入其他事务处理的数据元素的数据关系。

定义全体数据关系向量

全体数据关系向量的形式与事务处理使用向量相类似，不过其中每个单元的值是关系矩阵中相应行中非零单元的个数。

③生成使用率向量。

利用关系矩阵与事务处理使用向量可以构造一个包含每个数据元素使用率的向量。

④确定数据归类标准。

从上面得到的事务处理使用向量、全体数据关系向量和使用串向量3个向量的基础上画出相应的直方图，使用分界点取法，确定数据元素下面3类特征的高值、平均值和低值：

交叉使用事务处理

与其他数据元素一起使用

使用率

根据事务处理/数据关系的个数和数据的范围，划分成3个不同的分界点。划分的结果若与实际情况存在很大的差别就应回过头去重新选择参数和重新计算数据归类标准。

根据数据归类标准与各个数据和元素在各种不同情况下的出现值，采用上述统计分析结合数据元素的归类基本原则，可初步地分出一些实体、联系和属性。但由于统计分析方法常常受数据选取范围、数量的大小等因素的影响，往往不够准确，仅给出一些定量的信息作为参考。如欲精确地进行数据归类，必须与语义相结合，根据实际情况而定。

2）形成实体及其标识

在上述分析得到的数据元素初步归类的基础上，再结合语义，就可标明实体及其标识。

3）分析实体间的联系

联系是由两个或两个以上的实体来标识的，因此通过分析实体属性间的联系，并结合实体分析法中定义联系的原则和方法，定义实体间的联系。若实体的标识是组合属性，则选取其中有代表性的一个。一般选取事务处理使用值较大、与其他数据一起使用值较大、使用率较低的数据元素。

至此，就可建立局部视图的E-E-R模式框架，并注意消去冲突和冗余联系，其方法与实体分析法相同。

4）属性的指派

结合前面得到的与数据元素相关的3个向量：事务处理使用向量、全体数据关系向量和使用率向量，以及数据元素归类的基本原则和数据归类的标准，建立4张表：归为实体属性的数据元素表；归为联系属性的数据元素表；既可归为实体属性，又可归为联系属性的数据元素表；每个数据元素与其他数据元素一起使用的百分比表。前面3张表的结构是一样的，均由数据元素引用号和数据元素两部分组成，仅表名和内容不同而已。

属性指派的任务就是要把归为属性的数据元素分配给每一个具体的实体类和联系。

（1）实体属性的指派

首先，从归为实体属性的数据元素表中取出第一个数据元素，查阅关系矩阵，看它与哪个实体一起使用过，再按下述步骤决定该指派给哪个实体。

①若只有一个实体的标识属性与它一起使用，则将其归为这一标识属性所指的实体。

②若与多个实体的属性一起使用，且删去低百分比值的标识属性后，至少剩下一个标识属性，则进行删去工作，再进行下一步，否则直接进入下一步。

③逐一删去每一个标识属性，若删去后使用该数据元素失去存在的意义，则该数据元素就归为该标识属性所属的实体，否则就不用于该标识属性所指的实体。

④然后对表①中的其他数据元素重复上述过程，就可把表中的所有数据元素指派给相应的实体。

（2）联系属性的指派

首先，从表②中取出第一个数据元素，从关系矩阵找出它与实体标识属性一起使用的百分比，再执行下述步骤。

①分析联系标识属性的组合（它涉及两个以上有关实体）。

② 查阅联系的标识组中数据元素与其他数据元素一起出现的百分比表中出现的次数。

③若出现次数为一，则该数据元素应属于这次出现的数据元素的组合所代表的联系。

④若删去百分比低的数据元素的属性组合后，至少剩下一个标识属性组合，则进行删除后再做下一步，否则直接做下一步。

⑤分析每个标识属性组合与从表②中取出的数据元素之间的关系，若删去某组合会使该数据元素失去存在意义，则应把该数据元素归为该标识属性组合所指的联系，否则不属于该组合所指联系的属性。

对表②中的其余数据元素重复执行上述过程就完成了表②中数据元素的分配工作。

（3）两可属性的处理

对既可归为实体属性，又可归为联系属性的数据元素，可按前面1）中步骤作为实体的属性处理，若找不到恰当的实体，再按联系的属性处理（方法同前），直到找到比较合适的归类。

（4）实体、联系及其属性的最后形成

综合前面3步，并根据实际业务情况对数据元素做适当调整，以形成最后结果。

5.描述信息结构

用E-E-R模型或选定的其他数据模型来描述前面分析得到的实体、联系及它们的属性，并确定各实体之间的类型，即形成局部视图的数据模式。

由上面的讨论可知，属性综合法是以概率划界的，而划界的标准是主观确定的。因此，如果只使用一个参数，因缺乏可比性，可能会造成主观随意性而产生错误。最好是提供若干标准，以便作为参数选择，但这样做工作量很大，需要用辅助工具实现。

第 23 章：数据库设计　作者：希赛教育软考学院　来源：希赛网　2014年03月13日

视图的集成

视图集成就是要将反映各用户组数据的局部数据模式综合成单位中某个确定范围内的单一数据视图，即全局数据模式，所以又称模式汇总。该全局数据模式（E-E-R模式）是未来数据库结构的基础，因此视图集成是数据库设计过程中一个十分重要的步骤，也是一项较为复杂和困难的任务。

当所有局部视图设计完毕，就可开始视图集成。整体E-R图包含了每个局部E-R图的信息，它合理地表示了一个简单而又完整的数据库概念模型。

1.视图集成的原理和策略

视图集成的实质是所有局部视图的统一与合并，在此过程中主要使用3个基本概念：等同（identity）、聚合（aggregation）和普遍化（gener-aligation）。基于这3个概念，有3种相应的基本集成方法。

1）等同

等同又称一致性合并，是指如果两个或多个数据对象具有相同的语义，则把它们归并为同一个对象，以消除不一致性，而不管它们的名字原来是否相同；同样，如果两个对象有相同的名字，但表示不同的对象，则应通过改名把它们区分开来。

等同包括简单数据对象间的等同和多个数据对象的聚合与另外几个数据对象聚合之间的等同。等同的数据对象其语法表示形式不一定相同，它与通常说的同义异名是一个概念。识别等同时还要注意鉴别同名异义和语义相同但值域不同两种情况。同名异义虽有相同的表示形式，但语义却不同。等同的概念貌似简单，但在实践中要做出判断却并不容易，须做仔细分析，主要依靠对有关数据对象类值域的分析比较。

2）聚合

聚合表示和数据对象间的一种组成关系，例如数据对象学生可看成是学号、姓名、性别、年龄、系别、学籍、专业等数据元素的聚合。聚合集成主要用于实体的属性分配，有关的思想和方法请参看前面的实体分析法。

3）普遍化

普遍化是对某一概念范围内具有共性对象的一种抽象。在视图集成中，它用于对现实世界中的事物进行归类。虽然普遍化和聚合均表示事物的层次结构，且同一个数据对象可能同时参与普遍化和聚合两种联系，但它们是两个截然不同的概念。聚合是将若干不同类型的数据对象组合成一个高

级对象的过程，而普遍化是按某公共属性的值对事物进行抽象和归类的过程。

综合地运用上述3种方法可有效地进行视图的集成。

视图集成从策略上可分成两类：二元集成和n（n>2）元集成。每一类又有两种可能的集成方式：二元集成分为平衡式和阶梯式，n元集成分为一次集成与多次集成，因而总共有4种可能的集成方案。如图23-14所示。

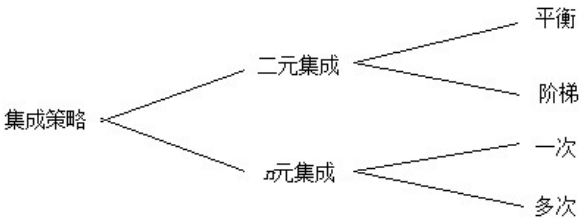

图23-14 视图集成的策略

1）二元集成

（1）平衡式集成

这种集成方式把整个集成过程分成若干层次，在每层中进行两两集成，其结果进入下一层集成，集成的对数逐层减少，最后得到统一的视图。Teory和Fry等人证明，只要选择恰当的集成初始序列，就有可能使这种集成的分析比较次数达到最少。两个分别具有N和M个对象的视图，其集成结果，在不考虑聚合和分解引起的增减的情况的下，可能有N+M-X个对象，其中X为视图间对象的重叠度，若两个视图中有i个对象等同，则X=i-1.为了最大限度地减少后面的集成中涉及的对象数，选择初始集成序列时应使每对集成视图的X值尽可能地大，若在每个集成层次上均按此原则进行，则就可能使总的集成效率达到最高。只有当配对的是关系极为密切的视图，即对应于事务处理中联系最紧密的功能域时，才能使配对视图中的X值为最大。由此可见，确定视图初始集成序列中的分组原则与划定局部范围的原则是一致的。因而有关的方法也是可参照使用的。

（2）阶梯式集成

这是一种流水线作业形式的集成方式，它无须考虑视图的初始序列，当然也不保证X值为最大，但省去了处理的麻烦，且这种方案适合和已经存在的局部集成模式进行综合。数据库的应用范围常常随着单位的发展而需要逐步扩展，扩展的每一种功能所涉及的数据，通常与已经集成的数据模式关联最为密切，而阶梯式集成为这种情况提供了最为适宜的策略。

由上可知，二元集成是一种较简单且行之有效的策略。它的优点在于可使每个集成步骤上的分析比较过程简单化和一致化，因而使用较广泛。它的主要缺点是集成操作的总次数较多，且在最后必须分析检查是否满足总体性能，必要时须做调整。

2）n元集成

（1）一次集成

此法一次集成n个视图。其优点是能充分地考虑全局需求，不必到最后再来分析调整，且集成步数量少。缺点是集成效率将随着视图数及视图中对象数的增加而明显降低，因为集成过程中的基本操作是等同性检查，一次集成n个视图，为了判别一个视图中单个数据对象的等同性，必须与其他n-1个视图中的每个数据对象进行比较，所以只有当n较小时，这种策略才有意义。

（2）多次集成

此法先用与平衡式二元集成相同的机理，将待集成的视图分成若干组，每组的视图数可以是两个或多个，但个数不能太多，然后按组集成，形成若干中间视图，再对这些中间视图进行分组、集

成，最后得到全局视图。它的优点是，具有平衡式二元集成法效率较高的优点，集成操作的总次数却又较少，其次，其分层的集成过程在概念上正好与大多数具有层次功能结构的事务单位相吻合。事务管理人员可凭借其丰富的事务知识有效地进行底层集成；到了高层，单位中高级管理人员具有的综合知识及全局观点，有利于运用聚合或普遍化概念进行高层次上的集成。此策略的效果同样取决于初始集成序列的划分。它的缺点是一致性差。

2.视图集成的方法和步骤

局部视图集成是一个相当困难的工作，往往必须凭设计者的经验与技巧才能很好地完成。尽管如此，集成的方法很多，它因所用的概念设计数据模型、集成策略、集成过程的输入/输出量及识别和解决冲突的方法的不同而各有其独特的执行过程，但总的来说都分成两个阶段：预集成阶段和集成阶段。

1）预集成阶段

在这个阶段需做的工作如下。

确定总的集成策略，包括视图集成的优先次序、一次集成的视图数及初始集成序列等。

检查集成过程要用到的信息是否齐全。

揭示和解决冲突，为下阶段视图归并奠定基础。

2）集成阶段

集成阶段的主要任务是归并和重构局部视图，最后得到统一的全局视图。全局视图须满足下述要求：

完整性和正确性。整体视图应包含各局部视图所表达的所有语义，正确地表达与所有局部视图应用相关的数据观点。

最小化。原则上，现实世界同一概念只在一个地方表示。

可理解性。即应选择最易为设计者和用户理解的模式结构。

一个视图的基本框主要由实体和联系组成，所以集成主要是实体与联系的集成，整个集成过程也就是这两类基本集成过程反复交替地执行过程。

3.等同的识别和冲突的发现与解决

1）冲突的表现及处理策略

（1）同名异义

为了发现不同视图间的同名异义问题，可以列出所有同名数据对象，然后逐一判别其语义。对同名异义冲突通常采用换名加以解决，既可对同名者之一换名，也可对两者都给以重新命名。识别语义的主要方法是进行值域分析。

（2）异名同义

识别异名同义比较困难，一般由设计者对所有对象一个不漏地逐一鉴别。它同样采用换名的方法解决。若归并时试图将它们合并为一个对象，则可以把其中之一的名称作为合并后的对象名；若集成后，它们仍以两个不同的对象存在，则可对其一换名。当然，若原名都不合适，则可以对两者都重新命名。

（3）同名不同层次

如果两个对象同名，但其中之一是作为一个视图中的实体，而另一个是另一视图中的属性，则在集成时就会发生同名不同层次冲突。解决这种冲突的办法有两个，一是将属性转换为实体，二是将实体变换成属性。

例如，设一局部视图中有一部门实体DEPT,而在另一与之集成的视图有职工实体EMP,且EMP有属性DEPT,于是发生了同名不同层次冲突。此时，可将EMP的DEPT属性去掉，另设一个实体DEPT与EMP建立联系，这时再与另一视图集成时就容易多了。

再如，设一局部视图中有一名为STOR的仓库实体，其中含有一属性部门号（DEPT-NO）；另一局部视图中有一单位实体DEPT,其中仅含有一个属性DEPT-NO.对这类同名不同层次的冲突可将DEPT实体变换为其所在视图中与DEPT相关的另一实体的属性，然后再进行集成。

必须注意的是，实体变换为属性时通常要满足一些特定条件，比如该实体通常只含有一个与同名属性具有共同特征的属性，且一定存在一个与该实体存在联系的另外实体。

（3）虽同名同义，但对象联系测度不同

所谓联系测度是指实体的联系是一对一、一对多还是多对多。若同名同义对象在一个局部视图中为一对多联系，在另一局部视图中为多对多联系，则集成时发生联系测度冲突。一般而言，一对多包含一对一，多对多包含一对多，所以解决这种冲突的方法往往取较高测度为集成后的相应联系的测度。

（4）数据特征不相容

如果一同名同义属性在一局部视图中可作为关键字属性，但在另一局部视图中不具有关键字属性特征。或者，如果一组属性在不同视图中具有相反的相互依赖关系。这两种情况均会发生数据特征不相容冲突。对于第一种不相容冲突，集成时往往需要重新选择关键字。关于第二种不相容冲突，解决的策略则依赖于实际应用环境。例如设两个不同局部视图中都含有课程和教室属性。其中之一存在课程决定教室的属性依赖关系；而另一局部视图中课程与教室的依赖关系刚好相反。当将这两个属性集成到一个实体中时，其间原有的这种对应联系不存在。此时，若这种联系的丢失不影响，集成可正常进行。但如同第一种冲突，有时可能要重新指定关键字，而且，有时第二种冲突也可能具有第一种冲突的特征，本例就可能是这种情况。

不相容冲突因不同的环境而可能有多种不同形式，应根据具体情况采用不同策略加以灵活处理。

2）等价对象类之间的映射

另一类貌似并不等同但通过适当映射可转化为等同的数据对象，称为等价的数据对象类。通常有以下两种等价情况。

一是用不同的名称和值域描述同一种事物。例如，出生日期（日期型）与年龄（整型）、职工号（字符型）与工作证号（整型）等。

二是用不同的量度单位度量同一事物。对于"重量"、"长度"等量度属性都可能出现这一情况。如对于一个物品的重量单位在一个视图中定义为吨，而在另外的视图中定义为公斤。这两种情况实质是用不同的形式来描述同一事务或概念，这种描述同一事物或概念的数据对象类称为等价数据对象类。这些等价对象类之间可以按一定的规则映射，如出生日期与年龄之间、吨与公斤之间都可按公式进行相互转换，对那些没有确定的规则可循的等价对象类之间的映射可用对照表加以实现。

等价对象类是由于各个用户组对数据对象的观点不同而造成的。它们可以通过映射转化为等同数据类，再按等同数据对象进行集成。所以在集成前应仔细识别所有等价数据对象类，并确切说明它们之间的映射。

3）定义数据对象类的值域

一个数据对象类所有可能的实例的集合称为该对象类的值域，或简称为域。数据对象类的值域

是分析该数据对象类语义的重要依据，是识别各个视图中数据对象类在所讨论的概念上是等同、有共性或不相干的主要依据。因此仔细定义每个数据对象的位域是识别来自不同视图的数据对象类是否等同的主要手段。如果我们用Dom（A）表示对象类A的值域。各对象类在值域上存在4种相关情况：

域等同：Dom（A）=Dom（B）

域包含：Dom（A）Dom（B）或Dom（B）Dom（A）

域重叠：Dom（A）∩Dom（B）≠φ

域分离；Dom（A）∩Dom（B）=φ

对所有数据对象按上述4种域相关情况归类后，便可用不同的方法进行集成。

4.实体类的集成

下面我们按照前面定义的4种值域相关情况，采用二元集成策略，对各视图中的实体进行集成。为了方便起见，用大写字母，如A、B等表示实体类，用Attrs（A）表示实体类A的属性。

1）域等同

A和B是来自不同视图的两个实体，如果它们的值域等同，则集成过程就是建立一个单一的实体类，设为C,作为全局模式中的实体类，其属性Attrs（C）为A和B的属性之并，值域等同于A或B的值。因此，这种情况下，集成操作就变成了求并操作，可表示为：

Attrs（C）=Attrs（A）∪Attrs（B）

Dom（C）=Dom（A）（或Dom（B））

2）域包含

若两个实体类A和B,有Dom（A） Dom（B），即实体类A是实体类B的子集，在集成模式中把实体B换名为B_1,并建立一个新的实体A_1代替视图中的实体A,实体A_1的属性是实体A的属性与实体B的属性的差集，即去掉A与B中相同的那部分属性，而仅保留A中不同于B中的那部分属性。A_1与A、B的关系及B_1与B的关系可表示成：

Attrs（B_1）=Attrs（B）

Attrs（A_1）=Attrs（A）-Attrs（B）

Dom（A_2）=Dom（A）

Dom（B_1）=Dom（B）

3）域重叠

属于这种情况的实体类A和B,不仅存在部分共性，且两个实体类中的部分实例是重叠的。例如，一个工厂中参加业余大学学习的职工在不同视图中被表示成学生和职工两个实体类，但业余大学的学生具有学生和职工双重身份，是它们域的重叠部分。对这类实体的集成方法是，在全局模式中建立3个实体类，一是A、B两实体的联合，表示为AB,其域是A、B的域的并集，包含了A、B的所有实例，属性则为这两类实体的属性的交集，是A和B的公共属性；另外，两个实体记为A_1和B_1,是AB实体集的两个子集，其域分别等同于A和B的域，而它们的属性分别为A、B的属性中不在AB中出现的那部分。

4）域分离

设A和B是来自不同视图的两个实体类，如果它们之间存在下述情况之一，则归入域分离集成方法。一是A和B存在某些共性，但它们的实例在所关心的概念上是不相关的；二是两者虽然不存在公

共实例，但可统一于某个有意义的概念，第一种情况如教授与研究生，两者虽然有系名、姓名、年龄、住址等相同的属性，但在通常意义上，它们的值域不可能等同、包含或重叠。对于这种情况，集成时不必对它们进行归并，按原样转入全局模式。对于第二种情况可引入普遍化机制进行概念集成。在集成的全局模式中建立3个实体类A_1、B_1和C,其中A_1、B_1相应于局部视图中的实体类A和B,而C则相应于A和B的普遍化概念。全局模式和局部模式间属性和域的关系与域重叠的情况相同。

对于来自n个视图的n个实体类（A_1,A_2,…，A_n），其集成过程可以重复采用二元集成策略，对某些域关系可采用n元集成。

5.联系类的集成

联系类集成是通过语义分析来归并和调整来自不同视图的联系结构。联系的语义主要由联系的元数、实体在联系中的角色和参与度及联系的值域等表示。根据待集成联系类的元数和实体在联系中所起角色的异同，可将联系类的集成分成3种类型：

元数和角色均相同的联系类的集成。

不同角色的联系类的集成。

不同元数的联系类的集成。

其中每一种类型又可按实体的参与度或值域等语义因素分为8种情况。下面对它们逐一进行讨论。

1）元数和角色相同联系类的集成

情况1:实体的参与度相同

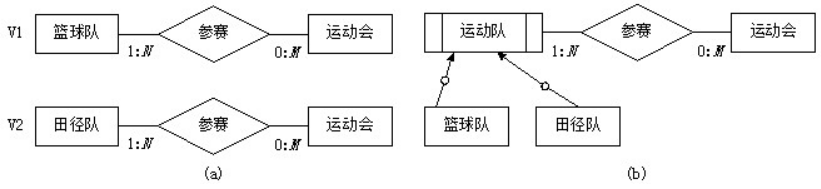

图23-15 实体的参与度相同的联系类集成

集成这类联系类时，通常只须将其中任一联系类原样转入全局模式即可，并根据情况对实体类和属性做简单的归并处理。例如，图23-15（a）表示来自视图V_1、V_2的两个联系类，除了左端的实体不同外，符合前面所讲的情况。左边的两个实体类，由于它们的域是不相交的，可进行普遍化集成，且可能已在实体集成中识别。图23-15（b）是其集成结果。

情况2:实体的参与度不同

这种情况表明，尽管联系的元数和实体在联系中担任的角色相同，但实体在不同视图中参与该联系的程度并不相同，即该联系类在不同的视图中所受的结构制约不同。对这类联系的集成，通常引入一个子集结构来协调两者在参与度上的差异。例如，教师与课程间的联系可能有两种观点，一是担任教学任务的观点，二是工作量统计的观点。按第一种观点，教师可能不担任任何教学任务，但从第二种观点，每个教师至少担任一门课程的教学任务。所以在两种不同观点的授课联系中，教师的参与度是不同的。在集成时可引入"任课教师"作为教师实体类的一个子集，以统一参与度，从而可将两个联系在全局模式中归并为一个联系。

2）相同元数、不同角色联系的集成

情况3:域包含

域包含指一个视图中的一个联系类为另一视图的一个联系的子集。例如教师与项目的联系，在一个视图中为"参与"联系，而在另一视图中可能是"领导"联系，两个联系中的实体类相同，但在其中

充当的角色却不同。但如果事务规则说明项目的领导者一定是项目的参与者，则"领导"联系类便是"参与"联系类的一个子集，也就是"领导"联系中的所有<教师，项目>实体对一定包含在"参与"联系的实体对集合<教师，项目>中。这种联系类的子集关系的语义制约是：当加入一个新的"领导"联系实例时，它必须同时加入"参与"联系，对于删除操作也是如此。

情况4:域分离

在这种情况下，参与两个不同视图中联系类的实体虽然相同，但它们表示的却通常是意义上互不相干的两件事。集成时，它们变成全局模式中两个实体类间的两个联系。

情况5:域重叠

这种情况的两个联系类的域之间有重叠部分，即一个联系类的某些实例同时是另一个联系类的实例，但两个联系类的域不是包含关系。集成时，在全局模式中，两个视图中相同的实体类两两合而为一，并在两个实体类之间建立3个联系，两个为集成前的原有联系，另一个为包含前两联系类实例的父集，与原有联系构成一个子集层次结构。例如教师与研究生之间，可能有上课和指导两种不同的联系。如果指导研究生论文的教师也要给研究生上课，则两个联系的实例就有重叠部分（假定不是所有上课的教师都指导论文）。

3）不同元数联系的集成

不同元数的两个联系类之间有3种可能的情况：元数较少的联系类可从元数较多的那个联系类导出；两个联系类在施加某些语义限制后可以兼容；不属于以上两种情况。对应地有3类集成处理：可归并的、有条件可归并的和不可归并的。

情况6:可归并的不同元数的联系

如果两个元数不同的联系类存在语义上的等价性，即它们实质上表示了相同的信息，则此两联系类是可归并的。通常总是将元数较多的联系转入全局模式。因此，仔细地鉴别两个不同元数的联系是能否归并的关键。例如图23-16中机器与零件之间的联系，在视图V1中以"组成"作为联系，在视图V_2中则以"包含"表示一个机器所包含的零件；实际上V2的联系可由V1导出，所以，只需把V_1中的联系转入全局模式即可。

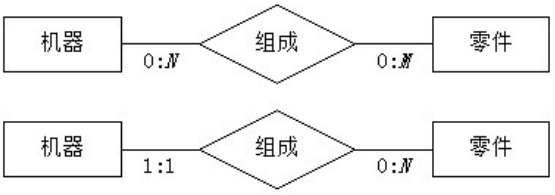

图23-16 可归并的不同元数的联系

情况7:有条件可归并的联系

这是指在某些条件成立的前提下，元数较少的联系可以从元数较多的联系导出的情况。换句话说，这种情况比情况6多了一个限制条件--可导出性条件。因此，集成这两个类的关键是识别元数较少的联系类从元数较多的联系类导出的条件，一旦识别成功，就可按情况6所述可归并联系相同的方法进行处理，即将元数较多的联系移入全局模式。例如图23-17中，V_1表示了工程向供应商购买部件的联系，V_2也表示了类似的联系。为了确定此两者是否可归并，首先要分析可归并的条件。如果V_2的联系中<工程，部件>对与<供应商>之间保持一对一的关系，而且R_I中供应商的所有属性都包含在R_2中，则该两个联系类是等价的，可以合并。按照可归并联系的集成原则，将元数较多的V_1中的联系移入全局模式。

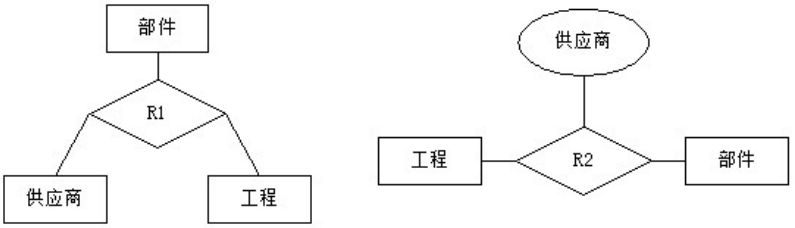

图23-17 有条件可归并的联系

情况8:不可归并的联系

如果两个联系类具有完全不同的语义，无法通过增加对联系的结构制约或语义制约而使它们兼容，则它们是不可归并的，集成时只能照原样分别转入全局模式中。例如设在视图V_1中有教师指导研究生参加课题研究的三元联系（在V_2中另有一个教师管理课题的联系），如图23-18所示。如果教师指导研究生所参加的课题和教师所管理的课题并不一致，则此两种联系所表达的语义是互不相干的，因而也就不能进行合并，只能照原样移入全局模式中。

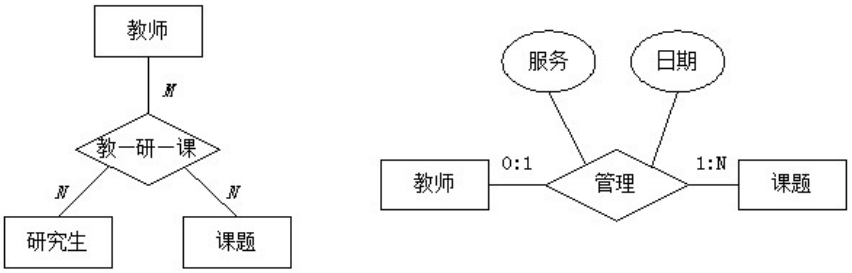

图23-18 不可归并的联系

6.新老数据模式的集成

当需要对已经建立的数据库系统进行扩充、修改，以扩大其应用范围，满足企业业务上发展的需要时，就会出现新老数据模式的集成问题。

现有的数据库系统的情况可能是：

由若干面向单项应用的独立数据库组成。

支持一定范围内多项应用的综合数据库。

不论是那一种情况，均可应用前面所讲的方法进行集成处理，但其具体内容稍有不同。

1）原有系统为多个独立数据库时的集成

这种情况下的集成包括单个数据库的集成和扩充的数据模式的集成。由于这些独立的数据库仅仅各自反映单个用户组的需求，而且很可能是在不同时期由不同的设计人员设计的，因而这些数据库之间及其和扩充的数据模式之间，必然地存在许多冲突和冗余。把它们集成为一个能支持所有应用，并能保证数据的一致性、完整性及最小冗余的全局模式后，原有的应用程序很可能都不能运行。因此，事前需要有周密的计划，原则是既要满足新的需要，同时尽可能地保持原来的数据模式，以便原有的应用程序稍做改动后仍能运行。

2）原有系统为单一的综合数据库时的集成

这时原有的数据模式已是经过集成的数据模式，再和扩充部分集成时，应尽量地向原模式靠拢，以使得原数据库支持的应用程序基本不变。

进行新老数据模式集成时，首先必须理解原有数据模式。如果原有数据库未留下概念设计的文档，就得从分析模式入手，这一过程不但十分困难和烦琐，而且容易出错或遗漏。因此，对于比较复杂的数据模式，最好先将逻辑模式翻译成相应的E-E-R图，将其与用户交互，确认新的需求，根据用户意见，对E-E-R模式进行调整，形成新的用户视图，再按前面所述的方法集成。可见，新老数据

模式的集成比之全新的视图集成受到的限制更多，因此，在某种意义来说也更加困难。

第 23 章：数据库设计　作者：希赛教育软考学院　来源：希赛网　2014年01月27日

逻辑结构设计

数据库逻辑结构设计的任务就是把概念结构设计阶段设计好的基本E-R图（或E-E-R图）转换为具体机器上的DBMS产品所支持的数据模型相符合的逻辑结构。这一阶段是数据库结构设计的重要阶段。

数据库逻辑设计的基础是概念设计的结果，而其成果应包括为某DBMS支持的外模式、概念模式及其说明，以及建立外模式和概念模式的DDL程序。因此，进行逻辑设计前，必须了解数据库设计的需求说明和概念设计的成果（包括E-E-R图和其他文档），并仔细阅读有关DBMS的文件。数据库的外模式和概念模式是用户所看到的数据库，是应用程序访问数据库的接口。因此在数据库逻辑设计阶段，还必须提供应用程序编制的有关说明，最好试编一些典型的访问数据库的应用程序，以检验所设计的概念模式是否满足使用要求。概念模式是数据库的基础，它的设计质量对数据库的使用和性能及数据库在今后发展过程中的稳定性均有直接的影响。为了设计出能够正确反映一个项目的数据间的内在联系的好的概念模式，设计时必须正确处理各种应用程序之间、数据库性能与数据模式的合理性和稳定性之间的各种矛盾，对设计中出现的各种矛盾要求要权衡利弊，分清主次，按统筹兼顾的原则加以正确处理。

逻辑结构设计一般分为以下几个步骤：

①将概念结构向一般关系模型转化。

②将第一步得到的结构向特定的DBMS支持下的数据模型转换。

③依据应用的需求和具体的DBMS的特征进行调整与完善。

下面以常用的E-R模型和扩充E-R模型为主，针对关系数据库的逻辑设计介绍基本原则和方法。

第 23 章：数据库设计　作者：希赛教育软考学院　来源：希赛网　2014年03月13日

E-R图向关系模型的转换

这里讨论基本E-R数据模型向关系模型的转换及扩充E-R数据模型的转换。

1.基本E-R模型转换向关系模型的转换

基本E-R模型主要包含实体和联系两个抽象概念，实体和联系本身还可能附有若干属性。其转换的基本原则是，实体和联系分别转换成关系，属性则转换成相应关系的属性。因此，E-R模型向关系模型的转换比较直观，但不同元数的联系具体转换方法稍有不同，下面根据不同的情况分别讨论。

1）一对一联系（1:1）

设有两个实体E_1和E_2之间为一对一联系。此情况存在3种可能的转换方案。

方案1:将实体E_1、E_2和联系名R分别转换成为关系E_1、E_2和R,它们的属性分别转为相应关系的属性，即得到

E_l（k_1,a）

E_2（k_2,b）

R（k_1,k_2,r）（k_2为候选关键字）

其中属性下面带一横线者为关系的关键字。

方案2:将实体E_l转换为关系E_l,将实体E_2与联系名R一起转换成关系E_2',E_2'的属性由E_2和R的属性加上E_1的关键字组成，其关键字为k_2,k_1为其候选关键字。转换后的关系为：

E_1（k_l,a）

E_2’（k_2,b,k_1,r），（k1是候选关键字）

方案3:与方案2类似，不过是把实体E_1与联系R一起转换成关系E_1’,其结果为：

E_1’（k_l,a,k_2,r），（k2是候选关键字）

E_2（k_2,b）

上述3个方案实际上可归结为转换成3个关系和转换成两个关系两种。如果每个实体的属性数较少，而联系的属性与两个实体之一关系又较密切。则可采用方案2或方案3,其优点是可减少关系数，有利于减少联接运算提高查询效率，但如果每个实体的属性较多，且合并后，会造成较大数据冗余和操作异常，则以采用方案1为宜。

2）一对多联系（1:n）

这种情况存在两种转换方案，其一是把两个实体类和一个联系类分别转换成对应的关系，实体的属性转换为对应关系的属性，其标识属性即为对应关系的关键字，而联系转换得到的关系其属性由两个实体的标识属性和联系类本身的属性组成，并以多端实体类的标识属性为其关键字。其转换结果为3个关系。第二个方案是转换成两个关系，设少端和多端的两个实体类分别为E_l、E_2,联系名R.转换时，将El转换为一个关系E_l,E_2和R合起来转换成一个关系E2?,E2?的属性由E2和R的属性加上E1的标识属性组成，并以E 2的标识属性为其关键字。

3）多对多联系（m:n）

两个实体类之间多对多联系组成的E-R模型向关系模型转换时，将两个实体和一个联系分别转换成关系，实体类的属性分别转换成对应关系的属性，其标识属性为其关键宇，由联系转换得到的关系的属性由两个实体类的标识属性和联系本身的属性组成，其关键字是由两个联系的实体类的标识属性组成的。

4）多元联系

实体类分别转换为相应的关系，3个实体类间的多元联系转换为以该联系名为关系名的关系，关系的属性由各实体的标识属性及其联系的属性组成，并以各实体的标识属性为其关键字。如图23-19所示的部件（PART）、工程（PROJECT）和供应者（SUPPLIER）三者之间的联系为P-J-S,其属性为QTY.转换时，把PART、PROJECT、SUPPLIER和联系PJS分别转换为相应的关系，其结果如下：

PART（P#,PN）

PROJECT（PNO,PNAME）

SUPPLIER（S#,SN）

PJS（P#,PNO,S#,QTY）

5）自联系

自联系是同一实体集的实体间的联系。例如对于职工实体类内部有领导与被领导的联系，在部件这个实体集的实体之间有组成成分与组成者之间的联系等，均属于实体类的自联系。在这种联系中，参与联系的实体虽然来自同一实体类，但所起的作用不一样。

6）弱实体类的转换

一个实体类，如果它的存在依赖于另一实体类，则称为弱实体类。例如职工的亲属（DEPENDENTS）是依赖于职工（EMPLOYEE）实体类而存在，因为实体集亲属（DEPENDENTS）是弱实体类，它们之间的关系如图23-20所示。由于弱实体类不能独立地存在，而是由其他实体标识而存在，所以不能单独地转换成一个关系，因此上图可转换成如下两个关系：

EMPL（empno,name,Birthdate）

DEPENDENTS（empno,name,sex,age,kinship）

其中kinship表示职工亲属与职工的关系，可以取值为"配偶"、"儿子"、"女儿",等等。

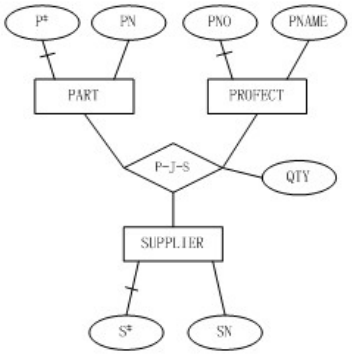

图23-19 多元联系

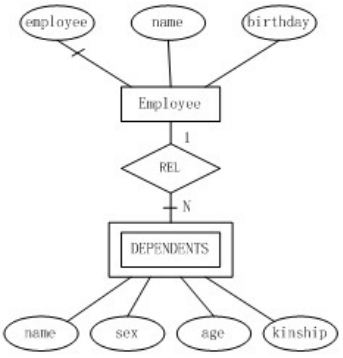

图23-20 弱实体类的转换

7）带有非原子属性的实体的转换

属性的原子性是关系模型对每个关系的基本要求。但E-R数据模型允许用集合或聚合（aggregate）作为属性，这类实体的转换与一般实体的转换有所不同。假如k是标识属性，a是普通原子属性，r是集合属性，$r=\{r_i|i=1,1,\ldots,n\}$,g是聚合属性，由原子属性g_1、g_2聚合而成，则此实体E可转换成下列两个关系：

E（k,a,g_1,g2）

E′ （k,r_i），i=1,2,…，n

其中k表示关系的主关键字。

2.扩充E-R模型向关系模型的转换

扩充E-R模型是基本E-R模型的扩充。它主要扩充了两点，一是一个实体集可能是另一个实体集中的某个属性，即一个实体集可以附属于另一个实体集，二是增加了一种叫ISA的特殊联系，这种联系建立了两个实体集间的继承（inheritance）关系，通过这种联系可以构成实体集之间普遍化/特殊化层次结构。下面讨论这些扩充部分的转换，这些转换方法也可推广到其他具有这些概念的数据模型。

1）一个实体集同时是另一实体集的属性

设实体集E1有属性k1、a1、k2,其中k1为E1的标识属性，而属性k2是另一实体E2的标识属性，E2有属性k2、e1、e2,则可转换成两个关系：

E_1（k_1,a_1,k_2）

E_2（k_2,e_1,e_2）

图23-21（a）是这种情况的一个实例，它可转换成工厂与厂长两个关系：

工厂（厂名，性质，地址，厂长工作证号）

厂长（身份证号，姓名，性别，年龄）

2）两个实体集间ISA联系的转换

这种情况中的实体E_0是实体集E_1的超集，而实体E1是实体集E_0的子集。E_1继承E_0的全部属性，同时，E_1可以有其自己的属性。图23-21（b）是这种联系的一个实例。

对这种E-E-R图，将其超类和子类分别转换成两个关系，并且子实体集以其超类实体集的关键字为关键字。即：

E_0（k,a）

E_1（k,b,c）

据此，图23-21（b）所示的实例可转换成下面两个关系：

学生（学号，姓名，年龄，系别）

研究生（学号。导师姓名，研究方向，攻读学位）

3）一个超集具有多个子集时的转换

设超实体集E0的属性集为{k,a_1,…，a_n},它有m个子实体集E_1,…，E_m,子实体集的属性集为Attr（E_i）（1≤i≤m），子实体可能不相交，也可能重叠。这种情况可把E_0,E_1,…，E_m分别转换成如下关系：

E_0（k,a_1,…，a_n）

E_i（k,Attr（E_i））（1≤i≤m）

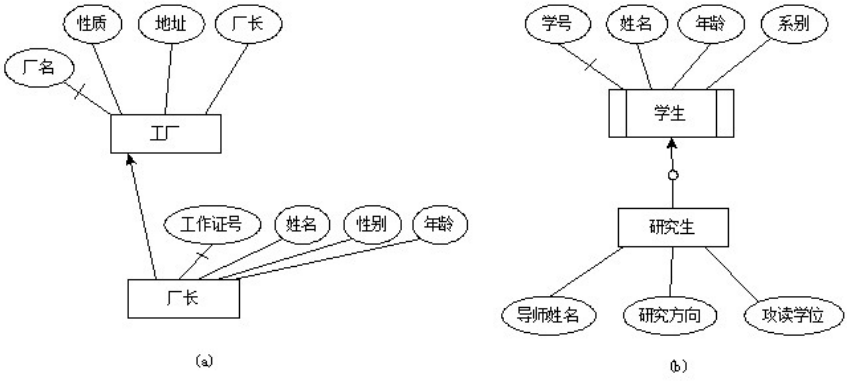

图23-21 扩充E-R模型向关系模型的转换

3.一般关系模型向特定的关系模型的转换

在逻辑结构方面，一般关系模型结构与目前较常用的DBMS支持的关系模型结构并无明显冲突。设计时需要注意下述问题。

DBMS支持的数据类型。

一般而言，DBMS只支持有限的几种数据类型，而E-R数据模型及E-E-R模型则无此限制，据此转换而得到的关系模型，必然保留了E-R或E-E-R模型中的数据类型。因此，在向具体的DBMS所支持的关系模型转换时，对DBMS不支持的数据类型必须做相应修改。如果用户坚持要使用原来的数据类型，就可能导致数据库的数据类型与应用程序中的数据类型不一致，应用程序必须负责这两者之间数据类型的转换。

DBMS对关系模式的数量、一个关系中属性的个数、关系名与属性名的长度等的限制。如果一般关系模型与DBMS的这些限制存在冲突，则按特定DBMS的要求进行修正。

必须注意的是，在向特定DBMS转换之前，须先对转换所得到的关系模式进行规范化处理。

上一节　本书简介　下一节

设计用户子模式

在ANSI/SPARC建议的数据库三级结构中，用户子模式（也称外模式）是用户所看到的数据库的数据逻辑结构。各个用户（或用户组）可以有各自的外模式。外模式是概念模式的子集，但在结构和形式上可以不同于概念模式，甚至可采用不同的数据模型，不过一般都是同一数据模型。

关系数据库的外模式由与用户有关的基表及按需要定义的视图构成。设计外模式时，可参照概念设计中的局部E-R（或E-E-R）图。在关系模型中，设计外模式是比较简单的。

上一节　本书简介　下一节

数据模型优化

由E-R（或E-E-R）图表示的概念模型转换得到的关系模式经过规范化以后，基本上可以反映一个企业数据的内在联系，但不一定能满足应用的全部需要和系统要求，因此，还必须根据需求分析对模式做进一步的改善和调整，其内容主要是改善数据库的性能和节省存储空间两个方面。

1.改善数据库性能的考虑

查询速度是关系数据库应用中影响性能的关键问题，必须在数据库的逻辑设计和物理设计中认真加以考虑，特别是那些对响应时间要求较苛刻的应用，应予以特别注意。

就数据库的逻辑设计而论，可从下列几个方面提高查询的速度。

1）减少联接运算

联接（joins）运算对关系数据库的查询速度有着重要的影响，联接的关系越多，参与联接的关系越大，开销也愈大，因而查询速度也愈慢。对于一些常用的、性能要求较高的数据库查询，最好是一元查询，这与规范化的要求相矛盾。有时为了保证性能，往往不得不牺牲规范化要求，把规范化的关系再合并起来，称为逆规范化（denormalization）。当然，这样做会引起更新异常。总之，逆规范化有得有失，设计者可根据实际情况进行权衡。

2）减小关系大小及数据量

被查询的关系的大小对查询速度影响较大。为了提高查询速度，可以采用水平分割或垂直分割等方法把一个关系分成几个关系，使每个关系的数据量减少。例如，对于大学中有关学生的数据，既可以把全校学生的数据集中在一个关系中，也可以用水平分割的方法，分系建立关系，从而减少了每个关系的元组数。前者对全校范围内的查询较方便，后者则可以显著提高对指定系的查询速度。也可采用垂直分割的方法，把常用数据与非常用数据分开，以提高常用数据的查询速度。例如，高校中教职工档案的属性很多，有些需经常查询，有些则很少查询，如果放在一起，则关系的数据量就很大，影响查询速度，把常用属性和非常用属性分开，就可提高对常用属性的查询速度。

3）尽量使用快照（snapshot）

快照是某个用户所关心的那部分数据，与视图一样是一种导出关系（derived relation），但它与视图有两点不同。一是视图是虚关系，数据库中并不存储作为视图的导出关系，仅仅保留它的定义。快照则是一个由系统事先生成后保留在数据库中的实关系。二是视图随数据当前值的变化而变化，快照则不随原来关系中数据的改变而及时改变，它只反映数据库中某一时刻的状态，不反映数据库的当前状态，犹如照片只反映某一时刻的情景，不能反映情景变化一样，之所以称它为快照，原因就在于此。但它与照片又有不同，快照不是一成不变的，它可以由系统周期性地刷新（refresh），或由用户用命令刷新。刷新时用当前值更新旧值。在实际应用中，快照可满足相当一部分应用的需要，甚至有些应用就是需要快照，而不是当前值。例如注明列出"某年某月某日截止"的统计或报表就是快照。由于快照是事先生成并存储在数据库中的，因而可大大缩短响应时间。目前不少DBMS,如Oracle、MS SQL Server等支持快照。对不支持快照的DBMS,用户也可以把需要作为实关系使用的导出关系作为一个独立关系存于数据库中，但这种做法只能供查询使用，对它们的刷新及管理由用户负责。

2.节省存储空间的一些考虑

尽管随着硬件技术的发展，提供用户使用的存储空间越来越大，但毕竟仍是有限度的，而数据库，尤其是复杂应用的大型数据库，需要占用较大的外存空间。因此，节省存储空间仍是数据库设

计中应该考虑的问题，不但要在数据库的物理设计中考虑，而且还应在逻辑设计中加以考虑。数据库逻辑设计中可采取以下措施。

1）缩小每个属性占用的空间

减少每个属性占用的空间，是节省存储空间的一个有效的措施。通常可以有两种方法。

用编码表示属性。例如用编码代替校名、系名、专业名称等比直接用文字表示要短得多，因而有效地节省存储空间。

用省略符表示。例如，用DS表示数据结构，用OS表示操作系统等。凡是能用省略符表示的尽量用省略符表示，同样能节省空间。

这两种方法的缺点是失去了属性值含义的直观性。

2）采用假属性（dummy attribute）

采用假属性可以减少重复数据占用的存储空间。设某关系模式R的属性A和B之间存在函数依赖A→B,B的每一个值需要占用较大的空间，但B的域中不同的值却比较少，A的域具有较多的不同值，则B的同一值可能在多个元组中重复出现，从而需要占用较多的空间。为了节省空间，可利用属性B的域中不同值少的特点，对B的值进行分类，用B′表示B的类型，则A→B可分解成两个函数依赖，即

$$A \rightarrow B', B' \rightarrow B$$

这样，就可用B′代替原来元组较多的关系R中的属性B,而另外建立一个较小的关系R′来描述B′与B的对应关系。这里B′在原关系R中起了属性B的替身的作用，所以称B′为假属性。例如，在职工关系中，职工的经济状况这一属性，通常由职工号决定。一个大型企业的职工人数虽多，工资级别、其他经济来源等，如每一职工逐一填写，就要占用较多的空间。为了节省空间可把经济状况分为几种类型，在元组较多的职工关系中用经济状况的类型代替原来的的经济状况，这里经济状况的类型就是假属性。另外建立一个较小的关系来描述每种经济状况类型的具体内容。

数据库设计与数学问题求解不同，它是一项综合性工作，受到各种各样要求和因素的制约，有些要求往往又是彼此矛盾的，因此，设计结果很难说是最佳的，常常是有得有失。设计者必须根据实际情况，综合运用上述原则和有关理论，在基本合理的总体设计的基础上，做一些仔细的调整，力求最大限度地满足用户各种各样的要求。

上一节　本书简介　下一节

作者：希赛教育软考学院　来源：希赛网　2014年01月27日

数据库物理设计

数据库物理设计是利用已确定的逻辑结构及DBMS提供的方法、技术，以较优的存储结构、数据存取路径、合理的数据存储位置及存储分配，设计出一个高效的、可实现的物理数据库结构。数据库物理设计过程如图23-22所示。

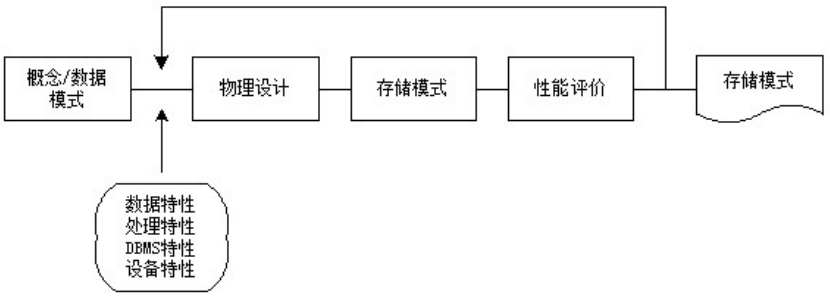

图23-22 数据库物理设计过程

由于不同的DBMS提供的硬件环境和存储结构、存取方法，以及提供给数据库设计者的系统参数及变化范围有所不同，因此，为了设计出一个较好的存储模式，设计者必须了解以下几方面的问题，做到心中有数。

了解并熟悉应用要求，包括各个用户对应的数据视图，即数据库的外模式（子模式），分清哪些是主要的应用，了解各个应用的使用方式、数据量和处理频率等，以便对时间和空间进行平衡，并保证优先满足应用的时间要求。

熟悉使用的DBMS的性能，包括DBMS功能，提供的物理环境、存储结构、存取方法和可利用的工具。

了解存放数据的外存设备的特性，如物理存储区域的划分原则，物理块的大小等有关规定和I/O特性等。

存储模式和概念模式不一样，它不是面向用户的，一般的用户不一定、也不需要了解数据库存储模式的细节。所以数据库存储模式的设计可以不必考虑用户理解的方便，其设计目标主要是提高数据库的性能，其次是节省存储空间。

数据库物理设计内容包括记录存储结构的设计、记录集簇（record clustering）的设计、存取路径（access path）的设计和物理设计的性能评价4个方面。

1）记录存储结构的设计

概念模式表示的是数据库的逻辑结构，其中的记录称为逻辑记录（logical record），而存储模式（内模式）则是逻辑记录的存储形式，是由存储记录组成的。记录存储结构的设计就是设计存储记录的结构形式，它涉及不定长数据项的表示、数据项编码是否需要压缩和采用何种压缩、记录间互联指针的设置，以及记录是否需要分割以节省存储空间等在逻辑设计中无法考虑的问题。

2）记录集簇的设计

记录集簇的设计是通过把一些经常用于同一访问的记录在外存空间中集中存放在一起，或存放在相邻的区域，以提高数据库的性能。在有些DBMS中，记录的存放位置由操作系统决定，DBMS无法控制，对这类DBMS在数据库的设计中也就毋庸加以考虑。但有些DBMS如Oracle和SQL/DS的DBMS提供了集簇（cluster）功能，供数据库设计者控制记录的存放。

3）存取路径（access path）的设计

这是利用DBMS提供的文件结构、索引技术等技术手段，选择适宜的文件结构和索引技术等使得主要应用能够对每个记录型或关系进行有效的访问，即提供合适的存取路径。

4）物理结构设计的性能评价

数据库物理设计是综合性的，既要满足各个应用对性能的要求，又要适应各种物理上的限制，如时间效率、空间效率、维护代价和各种用户要求，而且数据库的性能还与计算机系统的运行环境有关。例如，计算机系统是单用户还是多用户，负荷的轻重，数据库所用的磁盘是专用还是共享等，它们均影响到数据库的性能，但这些因素及其影响在数据库设计时又很难确切估计。因此，目

前，数据库物理设计主要还是采用启发式的方法，即根据一般的原则和设计要求，运用DBMS提供的各种手段，设计出初步方案，通过基准程序测试（benchmark testing）进行调整。

上一节　本书简介　下一节

存储记录的设计

在这里我们介绍几种常用的数据项存储技术和数据压缩技术。

1.数据项的存储技术

在存储记录中，数据项的表示方法常用的有以下4种。

定位法（Positional technique）：这是用得最多的一种方法。它按每个数据项的最大可能长度分配定长字段（field），数据项按照从左到右的顺序写入，不足部分用空白字符补充。这种方法的优点是简单，缺点是存储空间利用率不高，尤其当数据项值的长度参差不齐时更是如此。

相对法（relational technique）：此法并不规定每个字段的长度，而是用特殊分隔符分隔各个数据项的存储字段。

索引法（indexed technique）：每个字段用一个指针指向其首址。

标记法（1abeled technique）：每一字段用一标记开头。

在这4种方法中，定位法只能表示定长记录，其余3种方法可以表示可变长记录。所有DBMS支持可变长记录。

2.数据压缩方法

有些DBMS允许对数据进行压缩以减少存储空间。但用压缩数据进行存储，在访问时须进行转换或复原，从而增加了开销，降低了系统的效率。因此，在数据库中数据压缩技术仅应用于少数存储空间的容量成为主要矛盾的系统中，且一般也只采用一些简单的压缩方法。下面介绍几种可能的方法。

消零或空白符法（null suppression）：此法是用一特殊符号及表示零或空白符个数的数字来代替数据项中出现的一连串零或空白符。例如用@6表示000000,#5表示连续5个空白符。

模式代替法（pattern substitution）：此法是用一省略符代替数据项中经常出现的字符串，此法需要建立省略符与被替代字符串之间的对应表，称为模式表。转换时须查阅模式表。

索引法（indexing）：这是模式代替法的变种。它把那些经常出现的模式单独存储，在需要的地方用指针引用它。显然，只有当模式比指针长得多时才有意义。

上一节　本书简介　下一节

关系数据库的集簇设计

下面以Oracle DBMS为例，说明集簇在数据库设计中的应用。

集族是为了提高某个属性（属性组）的查询速度，把在这个或这些属性上有相同值的元组集中存放在一个物理块或物理上相邻的区域。我们先举个简单的例子说明集簇对查询速度的影响。设有一EMP（职工）关系，并建有出生年份的索引。如欲查询1935年出生的职工，假定1935年出生的职工有40人。在极端情况下，这40个不同的职工所对应的元组分散在40个不同的物理块中。执行这种查询，即使不考虑访问索引的I/O次数，读出该40个职工也要执行40次I/O操作，即访问40个物理块。如果按出生年份集中存放EMP元组，则每读一个物理块就可获得多个合乎条件的元组，可以显著地减少访问磁盘的次数。

Oracle允许以单属性或复合属性组成集簇键（cluster key），按集簇键集中存放元组。具有同一集簇键值的元组尽可能放在同一物理块中，如果放不下，可以链接多个物理块。对于支持区域（extent）概念的操作系统（此处区域系指磁盘上一段物理上邻接的存储区，如一道磁道或一个磁面），Oracle的这些链接块还可以集中在一个区域内。

Oracle中创建集簇语句的基本形式为：

CREATE CLUSTER集簇名（集族属性1,数据类型；集簇属性2,数据类型；……）；

其中，集簇名是需要产生的集簇的名称，括号中是集簇属性，至少一个，也可能是多个。每个属性后面是该属性的数据类型说明，属性间用逗号分隔。如果原来的关系为非集簇的，在创建后可将原来的关系转换为集簇关系。下面以EMP为例，说明集簇的创建过程。设创建的集簇名为CLE,集簇的关系名为CL-EMP,YR为出生年份，其创建过程如下：

① CREATE CLUSTER CLE（YR,NUMBER（4））；

② 按集簇CLE定义-个表CL-EMP:

```
CREATE TABLE CL-EMP
CLUSTER CLE（YR）
AS SELECT * FROM EMP;
```

③ 将EMP表的内容按集簇的原则复制到CL-EMP:

```
INSERT INTO CL-EMP
SELECT * FROM EMP;
```

④ 用集簇的EMP表代替原来的EMP表：

```
DROP TABLE EMP;
RENAME CL-EMP TO EMP;
```

集簇建成后，对用户而言，仍像集簇以前那样使用EMP表，只是感到按出生的年份查询时，速度明显地变快了。集簇后，集簇键相同的元组集中在一起，集簇键不必在每个元组中重复存储，只需在一组存一次，因此，集簇还可以节省存储空间。

Oracle的集簇功能也适用于多个关系。例如设EMP和DEPT是表示职工和部门的两个表，有公共属性DEPTNO（部门号）。在查询职工情况时，如涉及职工所在部门情况时，必须进行两张表的联接操作。如果把DEPTNO相同的EMP和DEPT元组在物理上集簇在一起，则可以大大提高联接的速度。其集簇过程如下：

① 创建集簇DEPT-EMP:

CREAT CLUSTER DEPT-EMP（DEPTNO,NUMBER）；

② 按集簇DEPT-EMP建立两个集簇表EMP1、DEPT1:

CREAT TABLE EMPl CLUSTER DEPT-EMP（DEPTNO）

AS SELECT * FROM EMP;

CREAT TABLE DEPTl CLUSTER DEPT-EMP（DEPTNO）

AS SELECT * FROM DEPT;

③ 把 EMP、DEPT分别变成集簇表EMP1、DEPT1:

INSERT INTO EMP1

SELECT * FROM EMP;

INSERT INTO DEPT1 FROM DEPT;

④ 以集簇表代替原来的表：

DROP TABLE EMP;

DROP TABLE DEPT;

RENAME EMP1 TO EMP;

RENAME DEPT1 TO DEPT;

与单表集簇一样，集簇键的值在一个组中只需存储一次，以节省存储空间。

必须注意：在集簇键中，至少应有一个属性被说明为NOT NULL.否则，若整个集簇键都是NULL,则有关元组将会无从存放。

集簇可以提高某些特定应用的性能，但对与集簇键无关的访问，则无所裨益。且当集簇键改变时，将引起整个关系大搬家，若一个元组的集簇键变动了，则该元组也要做相应的搬动。建立集簇的开销也很大，整个关系要进行移动，此关系上原来的索引均须重新建立，所以集簇一般在数据库建立和重组时进行，而且只有在满足一定条件的情况下才可考虑建立集簇。这些条件是：

按集簇键访问或连接是该表的主要应用，与集簇键无关的其他访问很少或者是次要的。尤其当主要应用的语句中包含与集簇键有关的ORDER BY、GROUP BY、UNION、DISTINCT等内容时，使用集簇键特别有利，可省去对结果的排序。

集簇键的值相对稳定，以减少集簇键修改时引起的维护开销。

对应每个集簇键值的平均元组数既不能太小，也不能太大。太小则集簇效果不明显，甚至浪费块的空间；太大就要采用多块链接，同样不利于提高性能。

在数据库物理设计中，对照上述条件分析，根据需要设置必要的集簇。在投入运行后，如果发现所设置的集簇收效不大，甚至有害，或者因为应用改变了，这种集簇没有必要了。则可用DROP CLUSTER语句撤销该集簇。但在撤销前，必须先恢复非集簇表，其过程与建立时相似。

作者：希赛教育软考学院 来源：希赛网 2014年01月27日

存取路径的设计

存取路径设计的目标是，提供访问每个记录型或关系的存取路径，使得主要应用能对它们做最有效的访问。它包括文件结构和索引选择两个方面。

1.数据库访问类型

虽然对数据库的访问要求是多种多样的，但就文件结构的选择而言，大致可分为3种类型。

1）访问文件的全部或相当多的记录

属于这类访问的应用-般要访问20%~100%的记录。例如生成职工工资发放表、登录学生成绩等处理中，需要访问全部记录。再如查阅所有教授、副教授及高工的情况需要访问相当多的记录。它们均属于这一类型。

2）访问某一特定记录

例如查询某人的通信地址或电话号码，查阅某个航班的起飞时间，查阅某个职工的情况等。这类应用访问的记录数为1或0.

3）访问某些记录

这种访问介于上述两者之间，既不是访问某一特定记录，也不是访问大量记录。其访问的记录一般少于总记录数的20%.例如查询职称为教授的女教师或查询因病休学的学生等。

2.文件类型的选择

有些DBMS是以操作系统的文件管理系统为其物理层的基础，但更多的DBMS是独立设计其物理层，其原因如下。

DBMS为了实现其功能，需要在文件目录、文件描述块、物理块等部分增加一些信息，传统的文件系统无此功能。

数据库中的文件为所有用户共享，要求其结构能兼顾多方面的要求，提供多种访问途径，提供并发控制、故障恢复和保密等方面的手段，而传统文件则无法满足这些要求。

传统文件系统主要面向批处理，在数据库系统中，往往要求即席访问、动态修改、要求文件结构能够适应数据的动态变化，提供快速访问路径。

如以操作系统中的文件系统为DBMS物理层的基础，则因DBMS对操作系统的依赖性，不利于DBMS的移植。何况，有些操作系统，如UNIX,仅提供字符流的存取功能，不提供各种文件结构，只能靠DBMS本身来实现。

数据文件的数据量变化大，传统文件无法适应这样的变化。

因此，DBMS的文件结构虽然继承了文件系统的某些技术，但又与传统文件系统有区别。数据库物理设计不是设计和实现新的文件结构，而是在DBMS所提供的各种文件结构范围内，根据应用的要求和数据的特征，选择适当的文件结构。

各个DBMS提供的文件结构和访问方法虽然有所不同，但常用的通常有以下几种。

1）直接文件（direct file）

在这种文件中，通过记录的关键字映射成记录的地址，直接访问该记录，与记录在文件中的位置及文件的大小无关，因此是随机的。这种文件不但查找方便，插入、删除、修改也很方便。关于关键字到地址的转换用各种散列函数实现。直接文件虽然能快速随机地查找文件中的某些记录，但由于某些原因，在目前的数据库系统中很少作为通用的访问方法，仅用于需要快速随机查询的场

合，如数据字典的查询。

影响直接文件在数据库系统中使用的原因主要有以下几点：

关键字所映射的地址空间并不决定于实际的数据量，而取决于键所映射的地址范围，因而往往造成存储空间的极大浪费。

不同的键可能映射到同一地址，对应每个映射地址有一个桶（bucket）。当桶装满了，就会溢出，这时就要进行溢出处理，增加了访问的开销，若出现严重的溢出情况，就会使性能恶化。

只能通过关键字直接访问，访问方法比较单一，适用于上述第二种类型的某些访问，对于其他类型的访问就非常不便。

不便于处理变长记录。

对于通用的DBMS很难找到合适的、通用的散列函数。

2）堆文件（heap file）

在这种文件结构中，记录按其插入的先后次序存放。记录可以存放在一个邻接的存储区中，也可以存放在多个不相邻接的存储区中，甚至可以存放在一个个物理上不相邻接的物理块中，通过指针或逻辑地址到物理地址的映射表等机制连成一体。这种文件结构的优点是建立容易，插入也很方便。缺点是一般只能采用顺序的扫描方式进行搜索。适宜于第一种类型的访问，对于其余两种类型的访问，效率很低。其次，删除、修改困难，一般在删除时只做删除标志，并不立即进行物理上的删除，以免引起大量记录的移动，而是在积累了一定数量的删除记录后，再集中清理。一般小文件都用堆文件，因为对于小文件不值得提供其他复杂的访问机构。尤其当内存缓冲区较大时，有些小文件可直接调入内存处理，速度很快。对于大文件，堆文件主要用于第一种类型的访问。

3）索引文件（indexed file）

实际上，索引文件是在前述的堆文件上增加索引。索引相当于一个映射机构，把某一键值转换为记录的地址。它与散列法的区别在于：索引法在有了记录后，才有索引项，才占有存储空间，因而不会浪费存储空间。当然，索引本身也要占用一定的空间，但索引键比记录本身要小得多，为此而付出的代价是可以接受的。此外，索引键的选择也比较灵活，可以是单属性，也可以是多个属性的组合，而且还可根据需要建立多种或多级索引。索引文件按其是否根据索引键排序而分为索引顺序文件和索引非顺序文件。

在堆文件的基础上加上索引后，既可利用原来的堆文件处理大量的记录，又可以利用索引访问个别记录，较好地满足了前述3种类型访问的需要，因而在数据库系统中得到广泛应用，尤其是B树动态索引用得最多。

3.索引的选择

原则上讲，索引选择可以采用穷举法，对每种可能的方案进行代价估算，从中选择最佳的方案，但实际上这是行不通的，原因如下。

数据库中文件之间的相关性：例如，关系数据库中的联接操作，涉及多个文件，计算联接操作的代价时，往往与其他关系参与联接操作的方法有关，因而各个文件往往不能孤立地进行设计。

可能出现组合爆炸问题，难以进行计算：即使对于由5个文件组成的数据库，每个文件只有5个属性的话，如果只在单个属性上建立索引，其存取结构的可能方案就有2.6×1011种之多。实际的数据库文件一般远不止5个，每个文件的属性也不止5个，而且有可能需要在多个属性上建立索引，这样的求解空间显然太大了。

访问路径与DBMS的优化策略有关：一个事务究竟如何执行，不仅取决于数据库设计者提供的

访问路径，还决定于DBMS的优化策略。若设计者所认为的事务执行方式与DBMS实际执行事务的方式不同，就会出现设计结果与实际情况的偏差。

代价估算比较困难：首先因为代价模型与系统有关，很难形成一套通用的代价估算公式。其次，代价估算还与数据库本身的特性有关，为了获得必需的设计参数，必须对数据进行统计分析，而在数据库设计阶段，对数据特性的了解往往是不充分的。

设计目标较为复杂：鉴于上述原因，很难进行比较精确、优化的数据库物理设计。目前常用的是一种简化了的设计方法，其思想和基本方法如下。

K.Y.Wbang等人分析了各种联接方法后认为，在一定条件的限制下，某些联接方法是可分离的，换句话说参与联接的各个关系的代价可以独立地估算，对于像用嵌套循环法进行两个关系联接等不可分离的联接方法，即使按分离的方法处理，也不会引起显著的误差。因此，在进行数据库物理设计时，可以分别设计各个文件，从而避免了前面提到的组合爆炸问题，大大缩减了求解空间。K.Y.Wbang等人的这一理论称为可分离（separabilty）理论。

按照可分离理论，根据处理要求，对记录在文件中的存放方式：有序或无序，或按某一属性（或属性组）进行集簇等，做出初步的抉择，再在运行中进行适当调整，而不必穷举各种可能，从而把问题归结为一个文件要建立哪些索引。而且经过简单分析后，可确定在有些属性上无须建立索引，当然也就不需要进行代价的分析和比较。不需要建立索引的属性有：

在查询条件中不出现或很少出现的属性。

属性值很少的属性。因为对于这种情况，如果在其上建立索引，则每个索引项后面附有大量的tid（元组标识符--由块号和记录在块中的地址组成），顺序索引集的溢出块将会很多，用索引去检索，不如直接进行顺序扫描。

建立索引的一般原则如下。

如果某个（或某些）属性经常作为查询条件，则考虑在这个（或这些）属性上建立索引。

如果某个（或某些）属性经常作为表的联接条件，则考虑在这个（或这些）属性上建立索引。

如果某个属性经常作为分组的依据列，则考虑在这个（或这些）属性上建立索引。

为经常进行连接操作的表建立索引。

一个表可以建立多个索引，但只能建立一个聚簇索引。

需要注意的是，索引一般可以提高查询性能，但会降低数据的修改性能。因为在修改数据时，系统要同时对索引进行维护，使索引与数据保持一致。维护索引要占用相当多的时间，而且存放索引信息也会占用空间资源。因此在决定是否建立索引时，要权衡数据库的操作。如果查询多，并且对查询的性能要求比较高，则可以考虑多建一些索引。如果数据更改较多，并且对更改的效率要求比较高，则应该考虑少建一些索引。

上一节　本书简介　下一节

物理结构设计的性能评价

评价物理结构设计完全依赖于具体的DBMS,主要考虑操作开销，即为使用户获得及时、准确的

数据所需的开销和计算机的资源的开销。具体可分为如下几类。

查询和响应时间：响应时间是从查询开始到查询结果开始显示所经历的时间。一个设计得好的应用程序可以减少CPU时间和I/O时间。

更新事务的开销：主要是修改索引、重写物理块或文件及写校验等方面的开销。

生成报告的开销：主要包括索引、重组、排序和显示结果的开销。

主存储空间的开销：包括程序和数据所占用的空间。一般对数据库设计者来说，可以对缓冲区做适当的控制，包括控制缓冲区个数和大小。

辅助存储空间的开销：辅助存储空间分为数据块和索引块两种，设计者可以控制索引块的大小、索引块的充满度等。

实际上，数据库设计者只能对I/O服务和辅助空间进行有效控制。对于其他的方面则只能进行有限的控制或者根本不能控制。

例题分析

例题1（2011年5月试题2）

阅读下列说明，回答问题I至问题3,将解答填入答题纸的对应栏内。

【说明】

某服装销售公司拟开发一套服装采购管理系统，以便对服装采购和库存进行管理。

【需求分析】

（1）采购系统需要维护服装信息及服装在仓库中的存放情况。服装信息主要包括：服装编码、服装描述、服装类型、销售价格、尺码和面料，其中，服装类型为销售分类，服装按销售分类编码。仓库信息包括：仓库编码、仓库位置、仓库容量和库管员。系统记录库管员的库管员编码、姓名和级别。一个库管员可以管理多个仓库，每个仓库有一名库管员。一个仓库中可以存放多类服装，一类服装可能存放在多个仓库中。

（2）当库管员发现有一类或者多类服装缺货时，需要生成采购订单。一个采购订单可以包含多类服装。每类服装可由多个不同的供应商供应，但具有相同的服装编码。采购订单主要记录订单编码、订货日期和应到货日期，并详细记录所采购的每类服装的数量、采购价格和对应的多个供应商。

（3）系统需记录每类服装的各个供应商信息和供应情况。供应商信息包括：供应商编码、供应商名称、地址、企业法人和联系电话。供应情况记录供应商所供应服装的服装类型和服装质量等级。一个供应商可以供应多类服装，一类服装可由多个供应商供应。库管员根据入库时的服装质量情况，设定或修改每个供应商所供应的每类服装的服装质量等级，作为后续采购服装时，选择供应商的参考标准。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图（不完整）如图23-23所示。

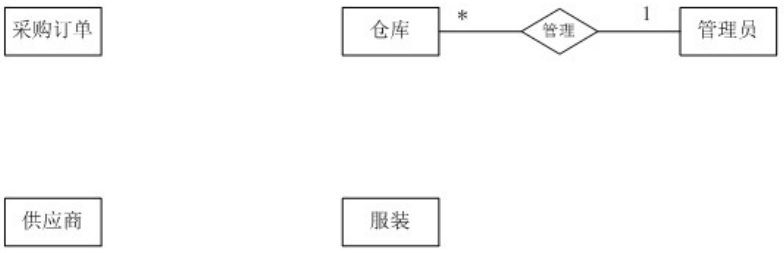

图23-23 不完整的实体联系图

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式（不完整）：

库管员（库管员编码，姓名，级别）

仓库信息（ （1） ,仓库位置，仓库容量）

服装（服装编码，服装描述，服装类型，尺码，面料，销售价格）

供应商（供应商编码，供应商名称，地址，联系电话，企业法人）

供应情况（ （2） ,服装质量等级）

采购订单（ （3） ）

采购订单明细（ （4） ）

【问题1】

根据需求分析的描述，补充图23-23中的联系和联系的类型

【问题2】

根据补充完整的图23-23,将逻辑结构设计阶段生成的关系模式中的空（1）~（4）补充完整，并给出其主键（用下划线指出）。

【问题3】

如果库管员定期需要轮流对所有仓库中的服装质量进行抽查，对每个仓库中的每一类被抽查服装需要记录一条抽查结果，并且需要记录抽查的时间和负责抽查的库管员。请根据该要求，对图2-1进行修改，画出修改后的实体间联系和联系的类型。

例题1分析：

本题考查数据库概念结构设计、概念至逻辑结构转换等内容。

此类题目要求考生认真阅读题目，根据题目的需求描述，给出实体间的联系。

【问题1】

本题主要考查根据题目描述补充完整ER图。

在本题中，根据题目描述"一个仓库中可以存放多类服装，一类服装可能存放在多个仓库中",我们可以知道服装与仓库间存在多对多的联系"存放";根据题目描述"一个供应商可以供应多类服装，一类服装可由多个供应商供应。"我们可以知道，供应商与服装之间存在多对多的供应关系；然后我们根据题目描述"一个采购订单可以包含多类服装。每类服装可由多个不同的供应商供应"可知，在服装、供应商和采购订单之间存在一个采购联系，其中三端都是多端。

【问题2】

该问题要我们补充完整各关系模式中缺失的属性并给出各关系模式的主键。要补充各关系模式缺失的属性应该根据题目的描述来完成。第1空是要我们补充仓库信息关系模式所缺失的属性，根据题目的描述，仓库信息包括：仓库编码、仓库位置、仓库容量和库管员，因此第1空应该填（仓库编码，库管员编码），这里用库管员编码而不用库管员的原因是库管员编码是库管员关系模式的主

键，而本关系模式的主键是仓库编码。

第2空是要我们补充供应情况所缺失的属性，供应是供应商与服装之间的联系，而这里是一个多对多的联系，多对多的联系在转换为单独的关系模式时，属性包括两端实体的主键其自身的一些属性，因此第2空应该填（供应商编码，服装编码），而该关系模式的主键为（供应商编码，服装编码）。

第3空要与第4空一起来考虑。第3空要我们补充采购订单关系模式所缺失的属性，根据题目的描述，采购订单主要记录订单编码、订货日期和应到货日期，并详细记录所采购的每类服装的数量、采购价格和对应的多个供应商。这里由于有关系模式采购订单明细，所以第3空应该填（订单编码，订货日期，应到货日期），而第4空应该填（订单编码，服装编码，供应商编码，数量，采购价格）。而采购订单的主键为订单编码，另外，由于题目描述"一个采购订单可以包含多类服装。每类服装可由多个不同的供应商供应",可知采购订单明细的主键为（服装编码，供应商编码）。

【问题3】

本题描述"如果库管员定期需要轮流对所有仓库中的服装质量进行抽查",我们可以知道抽查与库管员、仓库及服装这三个实体有关系，而且三端都是多端。这样就很容易画图ER图（见试题答案）。

例题1参考答案：

【问题1】

答案见图23-24所示

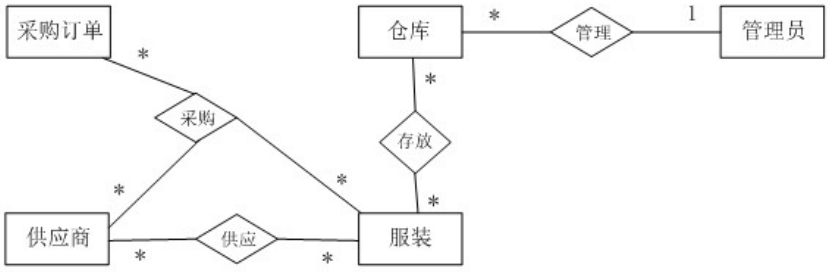

图23-24 本题答案图

【问题2】

（1）仓库编码，库管员编码

（2）供应商编码，服装编码

（3）订单编码，订货日期，应到货日期

（4）订单编码，服装编码，供应商编码，数量，采购价格

【问题3】

本题答案见图23-25所示

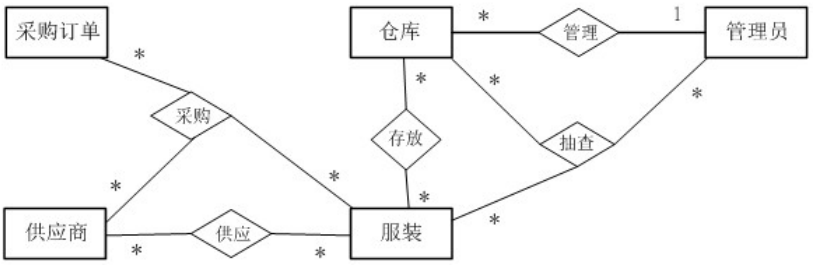

图23-25 本题答案图