微软院校认证课程系列教材

关系数据库应用基础

Microsoft 著

高等教育出版社

需要全本请访问 www.ertongbook.com

策划编辑　冯　英
责任编辑　蒋　青
封面设计　张　楠
版式设计　史新薇
责任校对　尤　静
责任印制

图书在版编目（CIP）数据

关系数据库应用基础／Microsoft **著**. **北京：高等教育出版社**，2003.7
(**微软院校认证课程系列教材**)
ISBN 7–04–013184–6

I. **关**… II. M… III. **关系数据库–数据库管理系统–技术培训–教材** IV. TP311.138

中国版本图书馆 CIP **数据核字**（2003）**第** 058655 **号**

出　　版	**高等教育出版社**	**购书热线**	010 – 64054588
社　　址	**北京市西城区德外大街** 4 **号**	**免费咨询**	800 – 810 – 0598
邮政编码	100011	**网　　址**	http://www.hep.edu.cn
总　　机	010 – 82028899		http://www.hep.com.cn
排　　版	**高等教育出版社照排中心**		
印　　刷			
开　　本	787×1092 1/16	**版　　次**	**年　月第　版**
印　　张	16.5	**印　　次**	**年　月第　次印刷**
字　　数	400 000	**定　　价**	60.00 **元**(**含光盘**)

版权所有　侵权必究

目　录

第 1 章　逻辑数据设计 1
1.1　设计过程 1
1.1.1　MSF 过程模型 1
1.1.2　收集需求 2
1.1.3　MSF 设计过程模型 2
1.1.4　数据和 MSF 设计过程模型 3
1.2　数据需求 3
1.2.1　数据需求的特征 3
1.2.2　数据需求的分类 4
1.2.3　确定功能性数据需求 4
1.2.4　确定非功能性数据需求 5
1.2.5　验证数据需求 5
1.3　逻辑数据设计的介绍 6
1.3.1　为什么需要逻辑数据设计 6
1.3.2　定义逻辑数据服务 6
1.3.3　实体和属性 7
1.3.4　数据的关系 8
1.3.5　实体/关系模型 10
1.4　规范化逻辑数据设计 12
1.4.1　实现实体的关系 12
1.4.2　规范化的基础 13
实验 A　确定数据相关的用况和数据需求 18
练习 1　确定要求数据的用况 18
练习 2　确定隐式数据需求 19
练习 3　确定非功能性需求 19
实验 B　从数据需求中定义实体和属性 20
练习 1　确定数据实体 20
练习 2　确定实体属性 21
实验 C　创建逻辑数据模型 21
练习 1　确定两实体间的关系 22
练习 2　确定基数和存在关系 22
实验 D　确定逻辑模型中的键 23
练习　确定键 23
实验 E　规范化数据 23
练习　创建规范化的数据设计 24
习题 26
实验附录　Ferguson & Bardell 公司案例 26
1. 公司 26
2. 技术 28
3. 时间和记账 30
4. 另外一些谈话 33
第 2 章　物理数据设计 36
2.1　物理数据设计的介绍 36
2.1.1　描述物理数据模型 36
2.1.2　确定数据库的表和字段 37
2.1.3　常用数据类型 37
2.2　实现关系 38
2.2.1　实现一对一关系 38
2.2.2　实现一对多关系 39
2.2.3　实现多对多关系 39
2.3　数据优化技术 39
2.3.1　优化的目标 39
2.3.2　为数据的创建进行优化 40
2.3.3　为数据的检索进行优化 40
2.3.4　为数据的更新进行优化 41
2.3.5　删除数据的注意事项 42
2.4　实现数据完整性 43
2.4.1　数据完整性概述 43
2.4.2　实现数据完整性需求 45
实验 A　转化逻辑数据设计 47

练习 将逻辑数据设计转换为物理数据设计 ····· 48
实验 B 优化物理数据设计 ····· 50
练习 确定可优化的区域 ····· 50
实验 C 确定数据完整性需求 ····· 52
练习 1 确定数据有效性需求 ····· 52
练习 2 确定数据完整性需求 ····· 54
实验 D 确定数据完整性的实现 ····· 54
练习 确定完整性的实现 ····· 55
习题 ····· 56
第 3 章 Access 2000 简介 ····· 57
3.1 Access 数据库简介 ····· 57
3.1.1 使用 Access ····· 57
3.1.2 数据存储 ····· 59
3.1.3 数据检索 ····· 59
3.1.4 数据输入与显示 ····· 60
3.1.5 数据在网络上的分发 ····· 61
3.2 Access 入门 ····· 61
3.2.1 启动 Access 并打开现有的数据库 ····· 62
3.2.2 使用向导创建数据库 ····· 62
3.2.3 使用数据库窗口 ····· 65
3.2.4 使用各种视图类型 ····· 66
3.2.5 获取帮助 ····· 70
3.3 书店应用程序概述 ····· 72
3.3.1 背景 ····· 72
3.3.2 现行数据管理 ····· 72
3.3.3 需求 ····· 72
实验 创建数据库 ····· 72
练习 1 使用数据库向导创建数据库 ····· 73
练习 2 存储和检索数据 ····· 73
习题 ····· 77
第 4 章 数据库设计 ····· 78
4.1 开始设计 ····· 78
4.1.1 设计数据库 ····· 78
4.1.2 分析数据库应用程序的输出需求 ····· 79
4.2 设计报表、查询和窗体 ····· 79
4.2.1 设计报表 ····· 80
4.2.2 设计查询 ····· 80
4.2.3 设计窗体 ····· 81
4.2.4 课堂练习 设计报表 ····· 81
4.2.5 课堂练习 设计查询 ····· 82
4.2.6 课堂练习 设计窗体 ····· 83
4.3 设计表 ····· 84
4.3.1 设计表和字段 ····· 84
4.3.2 组织数据 ····· 85
4.3.3 确定表与表之间的关系 ····· 85
4.3.4 确定数据完整性需求 ····· 86
4.3.5 课堂练习 设计表 ····· 86
习题 ····· 87
第 5 章 数据存储 ····· 89
5.1 创建表 ····· 89
5.1.1 使用表向导 ····· 89
5.1.2 课堂练习 创建表 ····· 90
5.2 自定义表 ····· 92
5.2.1 复查表的设计 ····· 92
5.2.2 使用设计视图自定义表 ····· 93
5.2.3 处理字段 ····· 94
5.2.4 处理数据类型 ····· 95
5.2.5 设置字段属性 ····· 96
5.2.6 定义查阅字段 ····· 97
5.2.7 创建索引简介 ····· 100
5.2.8 规范化表简介 ····· 100
5.2.9 课堂练习 自定义表 ····· 101
5.3 在表中添加数据 ····· 102
5.3.1 使用窗体 ····· 102
5.3.2 使用数据表视图 ····· 103
5.3.3 导入数据 ····· 104
5.3.4 链接数据 ····· 104
5.3.5 课堂练习 在表中添加数据 ····· 105

需要全本请访问 www.ertongbook.com

5.4 处理表 ····· 107
5.4.1 定义两表间的关系 ····· 108
5.4.2 用子数据表查看多个表 ····· 110
实验 A 创建表 ····· 110
练习 1 使用“表向导”创建表 ····· 111
练习 2 在设计视图中自定义表 ····· 112
练习 3 通过数据导入来填充表 ····· 114
实验 B 处理相关的表 ····· 115
练习 1 定义关系 ····· 115
练习 2 输入数据到相关的表中 ····· 116
习题 ····· 118
第 6 章 数据检索 ····· 120
6.1 排序表记录 ····· 120
6.2 在表中查找信息 ····· 120
6.3 筛选数据 ····· 121
6.3.1 确定筛选类型 ····· 122
6.3.2 使用筛选 ····· 122
6.3.3 多次使用筛选 ····· 125
6.4 查询数据库 ····· 125
6.4.1 使用查询 ····· 126
6.4.2 创建查询 ····· 126
6.4.3 查询属性 ····· 128
6.4.4 筛选查询 ····· 128
6.4.5 打印查询 ····· 129
6.4.6 课堂练习 创建查询 ····· 129
6.5 自定义查询 ····· 129
6.5.1 使用查询设计视图 ····· 130
6.5.2 指定查询条件 ····· 131
6.5.3 创建特殊的查询 ····· 134
6.5.4 使用表达式 ····· 136
6.5.5 使用“表达式生成器” ····· 138
6.5.6 使用计算的值 ····· 139
6.5.7 课堂练习 自定义查询 ····· 140
实验 A 创建简单查询 ····· 141
练习 1 创建和执行查询 ····· 141
练习 2 筛选查询输出结果 ····· 142
实验 B 创建含有计算字段的查询 ····· 144
练习 创建 Sales Analysis 查询 ····· 145
习题 ····· 146
第 7 章 生成窗体 ····· 148
7.1 创建窗体 ····· 148
7.1.1 使用窗体向导来创建窗体 ····· 149
7.1.2 使用自动窗体来创建窗体 ····· 150
7.1.3 图表和数据透视表向导介绍 ····· 151
7.2 自定义窗体 ····· 152
7.2.1 修改窗体内容 ····· 153
7.2.2 定义字段的查阅列表 ····· 154
7.2.3 自定义窗体外观 ····· 156
7.2.4 自定义窗体功能 ····· 157
7.2.5 自定义窗体区域 ····· 158
7.2.6 课堂练习 创建和自定义窗体 ····· 160
7.3 使用窗体 ····· 161
7.3.1 为窗体添加控件 ····· 162
7.3.2 创建计算控件 ····· 163
7.3.3 使用控件属性 ····· 163
7.3.4 在窗体中插入图形 ····· 165
7.3.5 在窗体中输入数据 ····· 166
7.3.6 创建子窗体 ····· 166
7.3.7 打印窗体 ····· 167
7.3.8 把窗体导出到 HTML ····· 168
7.3.9 课堂练习 在窗体中显示计算值 ····· 168
实验 A 创建窗体 ····· 170
练习 1 使用“窗体向导”创建窗体 ····· 170
练习 2 在设计视图中自定义窗体 ····· 171
实验 B 在窗体中显示计算值 ····· 172
练习 扩展“图书存量总额”窗体 ····· 173
习题 ····· 175
第 8 章 生成报表 ····· 176
8.1 创建报表 ····· 176
8.1.1 使用报表 ····· 177
8.1.2 使用“自动报表”创建报表 ····· 178

8.1.3 使用“报表向导”创建报表……178
8.2 自定义报表……181
8.2.1 在设计视图中修改报表……182
8.2.2 自定义报表的外观……183
8.2.3 自定义报表的区域……183
8.2.4 预览和打印报表……185
8.2.5 课堂练习 创建和自定义报表……186
8.3 增强报表……188
8.3.1 对记录排序和分组……188
8.3.2 使用条件格式……190
8.3.3 使用控件和函数……191
8.3.4 生成特殊报表……192
8.3.5 课堂练习 增强报表……193
实验 创建报表……195
练习 1 使用报表向导创建报表……196
练习 2 在设计视图中自定义报表……198
练习 3 在报表中显示计算的值……198
习题……199
第 9 章 生成数据访问页……201
9.1 创建数据访问页……201
9.1.1 数据访问页介绍……201
9.1.2 使用向导创建数据访问页……202
9.1.3 使用数据访问页……203
9.1.4 将数据访问页发布到 Web 服务器……204
9.1.5 控制对数据访问页的访问……205
9.2 自定义数据访问页……206
9.2.1 在数据访问页上添加和删除字段……206
9.2.2 使用控件……207
9.2.3 设置数据访问页的属性……209
9.2.4 自定义导航控件……211
9.2.5 课堂练习 创建和自定义数据访问页……211
9.2.6 将 HTML 页转换为数据访问页……213
9.3 数据访问页的使用……213
9.3.1 在数据访问页上对记录分组……214
9.3.2 在数据访问页上对记录排序……216
9.3.3 课堂练习 对数据访问页进行分组和排序……216
实验 创建数据访问页……217
练习 1 使用“页向导”创建数据访问页……218
练习 2 自定义数据访问页……219
习题……220
第 10 章 创建数据库应用程序……221
10.1 向数据库添加切换面板……221
10.1.1 创建切换面板页……222
10.1.2 创建切换面板项目……223
10.2 为数据库应用程序定义启动选项……225
10.2.1 指定启动时显示的窗体/数据页……225
10.2.2 设置数据库启动选项……226
10.2.3 自定义菜单 工具栏和状态栏……226
10.3 数据库安全……227
10.3.1 设置数据库密码……227
10.3.2 加密和解密数据库……229
10.4 维护数据库……230
10.4.1 备份和恢复数据库……231
10.4.2 压缩和修复数据库……231
10.4.3 复制数据库……232
10.4.4 拆分数据库……233
10.4.5 在 Access 2000 中使用早期版本的数据库……233
实验 创建数据库应用程序……234
练习 1 创建切换面板……234
练习 2 定义启动选项……237
习题……238
词汇表……239

第 1 章　逻辑数据设计

学习完本章后，将能够：

- 说明业务解决方案和以数据为中心的解决方案的设计过程
- 介绍逻辑数据设计
- 规范化逻辑数据设计

1.1　设计过程

在本节中，将学习设计过程以及 MSF(Microsoft Solutions Framework)过程模型，同时将了解这些过程如何构成方案设计的应用程序模型，这些设计过程之间又是如何交互的。

1.1.1　MSF 过程模型

Microsoft® 推荐使用针对应用程序开发的“ MSF 过程模型 ”作为业务解决方案软件开发项目的项目生命周期模型。

MSF 过程模型是一个阶段模型，即每个项目被分为各个不同的阶段，以便整个开发小组和负责人了解项目正处于整个软件开发生命周期中的哪个阶段，以及将要完成哪些关键任务。

MSF 过程模型是以里程碑为标志的，即每个阶段结束时必须达到一个里程碑，在里程碑处应该获得一定的可交付物。这些里程碑标志了开发小组的工作进程，每个里程碑代表一个过渡点，让项目开发小组和客户进行交流以达成共识，并可以进行复查来纠正可能存在的错误。

MSF 过程模型的各个阶段如下：

- 构思 (Envisioning) 阶段

该阶段中，开发小组与客户共同定义业务需求和项目总体目标。此阶段以“ 构思已认可(Vision Approved) ”里程碑作为结束标志，说明开发小组和客户对项目方向已达成一致

- 计划 (Planning) 阶段

该阶段中，开发小组和客户确定需要实现的功能、实现方法和时间进度计划。此阶段以“ 项目计划已受认可 (Project Plan Approved) ”里程碑作为结束标志，说明开发小组、客户和负责人在最后的交付物和交付时间等问题上已达成一致

- 开发 (Developing) 阶段

该阶段中，开发小组将实现设计和构造应用程序。此阶段以“ 范围完成 (Scope Complete) ”里程碑作为结束标志，说明了实现软件功能的代码已编写完成，此时可以对产

品进行外部测试并提高其稳定性

- **稳定** (Stabilizing) **阶段**

该阶段中，开发小组的工作重点转移到解决所有已知问题(如代码缺陷或功能错误)的阶段。此阶段不会进行新的开发活动，而是稳定和优化应用程序。此阶段以“发布 (Release)”里程碑作为结束标志，此后产品的主要工作转由运营小组负责

1.1.2 收集需求

在建立任何业务和数据解决方案之前，必须理解存在的问题以及期望的解决方案，而要做到这点并不容易。要想做出正确决定，得到完整可靠的信息是非常重要的。在收集信息时，需要考虑业务需求、业务流程、现有的系统和参与的人员。

收集信息和需求的过程会引起项目的业务调整。该过程中不仅会生成对当前系统中所执行操作和流程的说明，还会生成对未来的应用程序应如何满足这些需求的说明。

1.1.3 MSF 设计过程模型

MSF 设计过程模型可分为三个设计阶段：概念设计、逻辑设计和物理设计。

(1) 概念设计

概念设计的目标包括确定业务需求以及了解用户的行为和需求。设计小组从用户和业务的角度考虑问题，使用一定的场景来定义问题和解决方案，这些场景能够完整而精确地反映需求。

概念设计并不考虑构建解决方案所需的方法和技术，它相当于房屋建造过程中使用的粗略草图设计。概念设计其实就是一些由客户和设计人员共同创建的、比较容易理解的模型。

(2) 逻辑设计

逻辑设计的目标包括定义解决方案的组织结构以及解决方案不同部分之间的通信方式。设计小组从项目小组的角度考虑问题，将解决方案定义为一系列相互协作的服务。然而，设计小组并不使用特定的技术来定义这些服务。

根据在概念设计的场景中确定的业务问题，逻辑设计阐述解决方案的抽象模型，这些模型包括：对象和服务、用户界面原型与数据库逻辑设计。逻辑设计将计划进一步细化，并详细描述解决方案的规格说明，这类似于为建筑物设计施工图。

(3) 物理设计

物理设计的目标包括在逻辑设计的基础上应用现实世界中的技术约束，这些约束包括实现过程和性能问题。设计小组从开发人员的角度考虑问题，定义解决方案的服务和技术。此时，项目开发小组可以开始考虑实现解决方案的最佳方法以及所要使用的合适工具。

在物理设计中，根据逻辑设计的结果来产生组件、用户界面规格说明和数据库物理设计。这里用建造房屋来比喻：承包商对建造房屋结构的各个部分提出各自的方案，物理设

计就像是审核并选择这些方案(诸如布线、排管道、安装加热系统和通风系统等)。承包商细化了设计师的设计，使其满足现实世界中的实际情况。

1.1.4 数据和 MSF 设计过程模型

MSF 设计过程模型的解决方案设计流程是从概念设计到逻辑设计，再从逻辑设计到物理设计。有时很难确定每个阶段明确的开始和结束时间。要理解概念设计、逻辑设计和物理设计，关键就在于要知道每个设计阶段将产生怎样的实际输出结果。

在概念设计中，要调查、记录和分析解决方案的数据需求，这有助于确定在业务解决方案中需要存储和处理的信息。

在逻辑设计中，将根据数据需求初步产生一系列数据实体。期间需要制订并记录这些实体内容，也需要确定这些实体间的关系。逻辑设计确保了解决方案的数据设计能够正确地表示概念需求并映射到概念需求中。

在物理设计中，由逻辑设计产生的实体模型将映射到数据库中的表、字段和关系。该数据库就是逻辑模型的物理表示。

1.2 数据需求

在概念设计期间，了解如何建立数据需求将有助于定义最终产品。

1.2.1 数据需求的特征

概念级别上的数据需求具有一些标志性特征，这些特征有助于巩固概念设计，并将导出一组更好的用况 (use case)。用况通常是指描述用户(或者用户和其他系统协作交互)行为的一个简短的句子(例如，学生从图书馆借书)。在确定需求时，要考虑以下特征：

- 设计需求

为了获得数据库的逻辑实现和物理实现，需定义数据需求，也就是需定义将要存储的数据类型。

- 显式数据需求

数据需求应该很明确，并确保无二义性。用况必须明确显示每个用户或系统将处理何种类型的数据。

- 隐式数据需求

最难定义的数据就是“看不见”的数据。在数据设计阶段，不可避免地将一直存在隐式需求和不可预料的情况。

例如，Ferguson & Bardell 公司正计划开拓海外业务。显式数据需求就是，公司的计划和发票系统必须支持对账单和发票的跟踪功能，而隐式数据需求就是，当客户在其他国家支付账单时，该系统必须能进行实时的汇率换算，因而，在该解决方案中就应该包含货币汇率。虽然没有显式地表述该需求，但由于 Ferguson & Bardell 公司已经有意开拓国外市场，所以设计时也需要加以考虑。

1.2.2 数据需求的分类

一般来说，系统必须满足两类数据需求，在设计小组考虑系统的概念设计时，这两种类型的数据需求相互补充，这样设计小组就能够了解完整的系统构图。

(1) 功能性数据需求

在解决方案的概念设计阶段进行迭代时，必须理解当前系统中数据的使用方式以及今后与数据的交互方式，因为这能够帮助设计人员生成一组功能性数据需求。

功能性数据需求包括任何一种实际需求，也就是说，为了使以数据为中心的用况能够正确地运行，必须在数据系统中跟踪该需求。另外，它也是指参与者为了执行功能或任务而必须了解的信息和知识。每种功能性数据需求都由用况的一个参与者和一个对象直接跟踪。

(2) 非功能性数据需求

非功能性数据需求是对功能性数据需求更完全的描述。非功能性数据需求可能会改变设计思想和设计构架，因此在解决方案的概念设计中考虑非功能性数据需求是十分重要的。

考虑如下一些常见问题能够帮助设计人员确定非功能性需求：

- 可伸缩性

数据的取值范围是什么？总共有多少用户需要访问该系统？有多少用户需要并发地访问该系统？需要跟踪多少数据？在给定的时间间隔内要处理多少事务？

- 可用性

系统必须支持何种预期的使用级别和准备级别？可以接受的系统最长脱机工作时间(脱机工作时间超过这个最长时间就会导致出现问题)是多久？系统必须在哪些时间段可用？系统的使用高峰是什么时候？系统在哪些时间段可以脱机工作？

- 可维护性

系统的更新、维护和修复操作需要简单到什么程度？当前数据系统是否合适，怎样支持它们？每隔多久才修改系统或数据结构？每隔多久备份系统？

- 扩展性

为了今后在系统中添加功能，设计应该简单到什么程度？是按当前的容量进行设计还是按将来的容量进行设计？

- 互操作性

设计的是独立工作的系统还是开放式系统？设计的系统是否能和其他系统一起工作？数据来自何处？有多少系统存储该数据？当前系统中有哪些要给应用程序提供数据？要为哪些其他系统提供信息？需要导入和导出什么？

1.2.3 确定功能性数据需求

在确定解决方案的总体数据设计过程中，第一步必须为业务解决方案建立数据需求。如果解决方案没有数据需求，那就不需要存储数据，更不需要什么逻辑数据组织了。

需要全本请访问 www.ertongbook.com

要从解决方案的用况中确定功能性数据需求，应该执行以下任务：

- 检查每个用况

因为整个系统是围绕所有用况之间的交互而进行的，所以必须检查每个用况。确定对信息执行的操作顺序。

- 找出影响每个用况的因素

哪些是完成该用况的必要幕后操作？又有哪些系统执行的操作对于完成用况是必须的？在执行某个操作之前，参与者必须知道些什么？

- 确定应在源信息上执行何种操作

如何使用数据？例如：如何处理咨询人员的工作时间，以得到计费账单。

- 确定操作具有的特征

完成每个操作需要哪些步骤？哪些操作需要组合进行？例如，如何使用事务来处理信息。

1.2.4 确定非功能性数据需求

不考虑系统的非功能性需求会影响解决方案的成功执行。例如，Ferguson & Bardell 公司案例中的咨询人员需要将账单信息提交到公司的数据库中。解决方案的可伸缩性和可用性就是需要考虑的非功能性需求。

与多数其他设计过程一样，非功能性需求的确定过程本质上就是一个迭代的过程。通常在概念设计阶段，就应该不断地重复考虑数据需求，这样才能生成一组正确的非功能性需求。

执行以下这些步骤可以帮助设计小组确定非功能性数据需求：

- 考虑每个信息源

设计小组很可能会在忽略信息源的情况下考察数据需求，因此他们应该搜索所有信息源，以找到所有可能的需求，当然也包括非功能性数据需求。

- 查找影响用况的因素

与处理功能性数据需求一样，设计小组应该考虑以下问题：要完成这个用况需要进行哪些幕后操作？又需要系统进行哪些操作？

- 查找独立于用况的操作

考虑受非功能性数据需求影响的系统操作。例如，Ferguson & Bardell 公司系统中创建发票这一用况是否需要安全机制，以防止未经授权的用户篡改发票？

- 确定非功能性数据需求并将它包括在系统用况中

1.2.5 验证数据需求

对数据需求的验证可以保证指定的业务需求和业务约束得以满足。

在定义数据需求之后，应该确保可以根据业务需求来跟踪这些数据需求，这将有助于满足业务需求。

设计小组对数据需求的验证是必不可少的，因为如果在概念设计所确定的用况中有一部分是错误的，或是只表示了一部分预期解决方案，那么数据需求可能也会发生改变。

当解决方案的概念设计完成后，需要让设计小组的每个成员、客户以及关键负责人在

有关概念设计所表示的内容上达成一致。概念设计不仅应该表示出预期解决方案，还应表示出解决方案中的数据。

以下两种技术可用来确保概念设计中数据需求的有效性：

- 对源进行确认

确保用户在概念设计上达成共识，因为当前状态用况、数据需求和场景都是基于这些用户的。

- 聚集验证

让项目小组复查用况、场景和数据，以保证概念设计已经完整了。有时候只有在将概念设计及其数据需求与所有的输入对应起来时，才能发现该设计并不正确，它并不能表示预期的输出。

1.3 逻辑数据设计的介绍

在本节中，将学习逻辑数据设计。在设计过程这个阶段中，将设计数据结构，并规范解决方案的概念设计。

1.3.1 为什么需要逻辑数据设计

逻辑数据设计是从概念设计到物理设计这一自然过程的中间阶段。根据概念数据设计来定义逻辑数据设计方案的整个过程需要完成以下操作：

- 将概念数据需求转化为实际数据需求

必须把概念设计时确定的概念数据需求转化为实际的实体和关系，这些实体和关系定义了数据的交互方式。

- 帮助设计人员组织和理解数据

必须确保概念设计阶段确定的数据元素能够精确地描述业务解决方案所需要的数据。

- 关注数据关系的语义细节部分

必须深入研究解决方案的数据需求，并考虑数据如何在现有系统和将被构建的解决方案之间进行交互。

- 提供通信机制

逻辑设计过程中会记录一些数据，这些数据可以提供给其他感兴趣的人员进行阅读和理解。

1.3.2 定义逻辑数据服务

在逻辑数据设计中首先要解决的问题是数据服务，数据服务确定了管理和操作数据的方式，逻辑设计的开发提供了完成这些服务的方法。

(1) 将数据需求组织为实体

在收集完不同用途的数据需求后，就要将这些数据需求组织到相似的信息组中，对这些需求的分组被称为实体。实体允许你管理和跟踪所有相关数据。

(2) 确定数据服务

必须确定对数据进行哪些操作。例如，Ferguson & Bardell 公司的时间表。在使用该时间表的过程中包括许多步骤，很多员工可以编辑此表，或从表中读取信息，自动化程序还会将时间表中的信息提取出来放入另一数据库中，这些信息将被用来生成报表。

在确定了某个实体的使用方式后,就可以确定是否需要一个服务来作为该实体的接口。

(3) 确定服务的逻辑分组

在开发数据服务时，将作用于通用数据和流程上的服务分为一组，会带来很多好处。这样的分组确保解决方法的一致性，例如，服务是否考虑到了这样一个过程，即对员工对象的操作或对一组银行账号对象的处理。

例如，考虑来自 Ferguson & Bardell 公司案例中的咨询人员、咨询经理和行政助理。在某种程度上，他们都有与存储在系统中的时间和账单信息交互的要求。某个数据服务可能会处理来自咨询人员的时间表的提交，另一个数据服务可能会为行政助理将时间表中的数据放入账单中。所有 Ferguson & Bardell 公司系统的用户都需要使用时间表，但这并不意味着只有一个数据服务将时间表的所有数据当作同一类型来处理。

将逻辑数据服务按功能分组之后，就能设计处理解决方案所需的逻辑功能服务了。

(4) 确定组和实体之间的关系

服务所处理的数据不是单独存在的，现实场景中数据之间需要存在某种类型的关系，还需要清楚定义服务和数据间的关系。明确的关系提供了更好的逻辑设计定义，也增强了逻辑设计的功能。

在逻辑层面上，了解数据间存在什么关系是很重要的。以后，这些关系将变为逻辑数据模型的一部分，该模型是数据的逻辑表示，而用户也将在逻辑数据模型上处理日常事务。

通过确定所有可能的关系，开发小组就能更好地理解完整的情况。

(5) 精简设计

组织和分析实体之后，要再一次检查初始设计，并在所需之处进行精简，重复检查设计是产生完整设计必不可少的过程。

1.3.3 实体和属性

在本节，将学习在逻辑模型中表示的逻辑实体和它们的属性。

(1) 实体和属性的概述

当进入数据设计的逻辑设计阶段时，首要任务之一就是将数据需求和其他相关信息中取得的实体公式化。

实体可以是任何人、地点、事物或概念。

属性是进一步定义和描述实体实例的特性，一个实体可以有多个属性。

将实体作为名词，将属性作为形容词。

在逻辑设计时确定的实体和属性将在以后被用到，以帮助物理设计建模。

(2) 定义实体

在定义实体时，要牢记实体是：

- 存储某种信息的对象

实体就是要保存信息的对象。实体的示例有：职员、客户和时间表。

- 逻辑设计的起始点

设计数据库的第一步就是确定实体。

- 在数据库表中相应的行

实体的实例将与表中的行相对应。

在需求分析时，以下技术将帮助定义实体：

- 从收集的信息中找出用况
- 在用况中，查找名词和名词短语，这里的名词描述的是人、地点、事物或概念

请看如下示例：

“咨询人员每星期将他们的时间输入到时间表中，然后将时间表交给行政助理，行政助理将表中的时间输入发票生成程序，然后按提交的时间将发票发送给客户。”

粗体字就是可能的实体。在确定实体之后，编写短小精悍的实体定义是很有用的，这将有助于今后确定实体间的关系。

(3) 定义属性

在确定了实体之后，还需确定实体的属性。属性具有以下特征：

- 描述实体

属性描述了解决方案中的实体。例如，汽车的属性可能有颜色、牌子、型号和生产年份。

- 必须和最接近描述的实体有关

属性必须和实体联系在一起，它是依赖于实体而存在的。例如，颜色属性必须应用在对象上才能描述实际的物体。

- 定义数据库表中的列

物理设计实现后，属性就成为了数据库表中的列。

确定属性并没有特别的方法，反复考虑实体特性是一种比较有效的方法。如果属性很难确定，那么就应该检查实体定义得是否充分。

1.3.4 数据的关系

在本节，将学习关系的概念，关系是如何将实体联系在一起，以及关系如何在逻辑数据设计时提供高级功能。

还将学习如何通过基数和存在进一步定义关系。

(1) 数据关系概述

在定义完实体及其属性后，还要确定实体间的联系。在逻辑设计过程中，这些联系被称为关系。

关系在实体间起到了绑定作用。本质上来说，关系定义了实体相互之间的共同特性。定义的实体允许为解决方案构造现实数据模型。

(2) 确定实体间的关系

关系将实体连接起来。例如，职员属于部门，同样地，部门中也有职员。

关系可以被认为是联系实体的动词或动词短语，它们被称为关系动词。

你可以通过以下步骤来确定关系：

1) 查看实体定义。

2) 确定描述实体间相互行为的动词或动词短语，这些动词就是实体间的关系。

考虑以下示例(关系动词用粗体字表示)：

- 咨询人员提交时间表
- 行政助理开发票给客户
- 客户询问咨询人员

(3) 基数和存在概述

基数和存在是对实体间关系的进一步定义。

基数允许指定出现在关系每一边实体的实例数目。例如，一个咨询人员在同一时刻能操作多个项目。

存在允许在给定特定的基数时，指定关系中的实体是否能独立于关系的其他实体而存在。例如，每个项目必须有一个咨询人员。这个示例显示了一种存在条件的类型，必须满足该条件才能使该关系有意义。

(4) 确定基数

基数是关系的一个可确定属性。可以为一个关系分配以下三种基数类型：

- 一对一

某一实体的单个实例只与另一个实体的单个实例相联系。

一对一关系的示例如下：一个公民只能有一张身份证。

- 一对多

某一实体(父)的单个实例与另一个实体(子)的零个或多个实例相联系。

一对多关系的示例如下：图书馆的每个借阅者都可以一次借走若干本书。

- 多对多

某一实体的多个实例与另一个实体的多个实例相联系。

多对多关系的示例如下：一家音像店可以将某盘录像带租给很多不同的客户，而某个特定的客户也可以一次租多部录像带，因此，多盘录像带被租给了多个客户。

(5) 确定存在

关系中的存在确定了两个实体之间的强制联系或可选联系。

当一个或多个实体依赖于另一实体而存在时，就要指定关系的存在，以使关系保持有效。例如，发票只有当支付咨询人员工作报酬的客户存在时才能存在。在这种场景下，发票依赖于客户的存在才有效。

存在被分为两种类型：

- 强制存在

一个实体依赖于另一个实体的实例而存在。

强制存在的示例如下：发票只有在支付报酬的客户存在时才可以存在。

- 可选存在

一个实体能脱离其他实体而单独存在。

可选存在示例如下：咨询人员可以驾驶公司的汽车，但公司的汽车不依赖这些咨询人员也能存在。

1.3.5 实体/关系模型

在本节，将学习如何从实体、属性和关系中创建逻辑数据模型。

(1) 语法

逻辑数据模型用图来表示，该图被称为实体/关系 (ER) 图。该图使用了通用的语法来表示组成逻辑数据模型的实体、属性、关系、基数和存在。

一般而言，实体在 ER 图中用矩形来表示，而矩形内部则列出实体的属性。例如，Ferguson & Bardell 公司的员工 (Employee) 实体 ER 图，如图 1-1 所示：

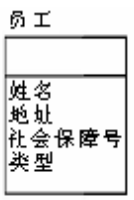


图 1–1 ER 图简单示例

关系定义了所有实体之后，就要将实体与关系联系起来，实体间用线段来表示关系，有三种常用的关系类型：

- 一对一关系通常用连接两个实体的线段来表示
- 一对多关系通常用一端带圆点的线段来表示，圆点一侧的实体是子实体
- 多对多关系通常用两端都带圆点的线段来表示

还可以将关系动词写在关系线段的边上，以指定所代表的关系。

通常用关系线段两端上的数字来表示基数。例如，写在关系线段一端的“ 1 ”，表示关系的父实体端只有一个实例；写在另一段的“ 1 ∞ Infinity ”，表示关系的子实体有不止一

需要全本请访问 www.ertongbook.com