国外计算机科学经典教材

Data Modeling for Everyone

数据建模基础教程

(美) Sharon Allen 著
李 化 等译

清华大学出版社

国外计算机科学经典教材

数据建模基础教程

(美) Sharon Allen 著

李 化 等译

清华大学出版社

北 京

内 容 简 介

本书通过联系大量实际应用，一方面将关系数据库的基本理论贯穿其中，另一方面又融入作者十几年工作实践提炼出的实用经验和技巧，把什么是数据建模以及怎样建立高质量的数据模型全面生动地展现在读者面前。全书包括 3 个部分和 1 个术语表。第一部分讲述了数据建模的理论基础及其方法论，其中包括基本的关系理论以及模型分析的类型和层次。第二部分通过构造一个数据模型实例详尽地描述了在实际工作中如何应用这些理论和方法，其中包括事务系统中的概念、逻辑、物理 3 阶段建模以及数据仓库系统中的多维建模。第三部分讲述了建模人员怎样为开发组增加价值。附录中给出了书中所涉及的专业术语及其解释。

本书不要求读者具有关系建模的预备知识或实际编程经验。适合于希望在关系数据建模上获得实用技术指导的数据库设计人员、开发人员和 DBA 等。

Data Modeling for Everyone
Sharon Allen
EISBN：1-904347-00-2

北京市版权局著作权合同登记号　图字：01-2003-0857

图书在版编目(CIP)数据

数据建模基础教程/(美)艾伦(Allen, S.)著；李伟等译. —北京：清华大学出版社，2004.9
书名原文：Data Modeling for Everyone
(国外计算机科学经典教材)
ISBN 7-302-09004-1

Ⅰ. 数… Ⅱ. ①艾…②李… Ⅲ. 数据库技术—建立模型—教材 Ⅳ.TP311.13
中国版本图书馆 CIP 数据核字(2004)第 067214 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　**邮　　编**：100084
社 总 机：010-62770175　**客户服务**：010-62776969
组稿编辑：曹　康
文稿编辑：王　军
封面设计：康　博
版式设计：康　博
印 装 者：北京鑫海金澳胶印有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：26　**字数**：655 千字
版　　次：2004 年 9 月第 1 版　2004 年 9 月第1 次印刷
书　　号：ISBN 7-302-09004-1/TP・6364
印　　数：1～4000
定　　价：48.00 元

前　言

如果把信息管理系统比作一座公寓，那么它底层的数据模型就像是公寓的设计蓝图。也许若干年后，住在公寓里的人会希望改变大厅和卧室的布局、打通厨房和餐厅，或者增加窗户的宽度。经验丰富的设计师将允许我们做这些改变，使得我们能够很幸运地跟上流行的装修潮流；而一个糟糕的设计师却会告诉我们，打通厨房和餐厅会影响结构的安全，增加窗户的宽度之前必须先对窗梁进行加固——这时我们会觉得唯一的解决办法可能只有重建公寓了。和建筑的设计蓝图一样，软件的数据模型也决定了软件的灵活性、可扩充性、健壮性等内在的特性，这些性质是无法通过增加界面的美观程度来达到的。

可见，一个高级建模人才为开发小组带来的潜在价值可能是很难衡量的。然而，数据建模给人的感觉总是非常神秘，这一方面可能是由于数据模型很容易让人联想起深奥的理论，另一方面也可能是因为建立好的数据模型要求极其丰富的设计和实践经验——设计和建模并不是根据理论照本宣科的过程，而是根据现实需要和约束进行各种权衡和选择。这些知识在纯理论性的书本中通常难觅踪迹，而同时又是许多开发和设计人员所迫切需要的。本书的一大特色就是强调理论与实际的联系，把关系理论、项目管理、软件工程理论与工程实例紧密地糅合在一起，使读者能够更加快捷、形象地理解到理论的实际含义，并在潜移默化当中掌握一些权衡方法和设计技巧。

在翻译本书的过程中，我们深深地觉得，对于从事信息系统的管理、开发和设计的任何人而言，本书都具有极高的价值。本书的原作者用他丰富的经验为我们勾画出了数据建模人员的工作思路和设计技巧，使我们在思维方法和工作能力上能够迅速获得一个大的飞跃。其次，本书的翻译人员是由国防科技大学计算机学院数据库方向的博士和研究人员组成的，他们大都参与或负责过国家和大型企业的重大项目，具备丰富的理论和实践经验。在翻译的过程中，我们特别注重了文化背景和中国人的阅读习惯，对原作者的一些文字进行了适当的处理，使之更加符合中国人的习惯，对原书存在的很少的编排错误也进行了修改。

全书由李化组织翻译，参与翻译的有李化、潇东、陈大峰、杨征、齐宁、彭智、周志华、左亚利、张君、李满朝和胡班。全书最后由肖国尊统稿。如果读者发现本书存在任何不足之处，敬请批评指正。

译　者

2003 年 3 月 3 日

目　录

第 1 章　数据建模介绍 …… 1
1.1　什么是数据 …… 1
1.2　什么是数据建模 …… 2
1.3　数据的生命周期 …… 2
1.4　数据建模对我们有哪些好处 …… 6
1.5　谁是数据建模者 …… 7
1.6　定义角色 …… 7
1.7　数据建模者的开发章程 …… 9
1.8　职称 …… 10
1.9　As-Is 支持 …… 10
1.9.1　配置管理支持 …… 10
1.9.2　提供影响分析 …… 11
1.9.3　提议 IT 标准 …… 11
1.9.4　提供数据完整性评估 …… 12
1.9.5　调查现有技术和工具 …… 12
1.10　To-Be 支持 …… 12
1.10.1　设计新的数据结构 …… 12
1.10.2　提供专家建议 …… 13
1.10.3　提供可供选择的办法 …… 13
1.10.4　提供预期评估 …… 13
1.10.5　调查新的技术和工具 …… 13
1.11　小结 …… 13
第 2 章　关系建模 …… 15
2.1　数据库模型 …… 15
2.1.1　分层 DBMS …… 15
2.1.2　网络 DBMS …… 16
2.1.3　关系 DBMS …… 16
2.2　概念建模与逻辑建模的概念 …… 17
2.2.1　实体 …… 17
2.2.2　类别实体 …… 20
2.2.3　联接实体或交叉实体 …… 23
2.2.4　属性 …… 25

2.2.5 键……27
2.2.6 关系……31
2.2.7 关系模型业务规则……34
2.3 物理建模概念……35
2.3.1 表……35
2.3.2 视图……37
2.3.3 列……37
2.3.4 约束……38
2.4 建模语法……39
2.4.1 集成定义符号(IDEF1X)……39
2.4.2 框……39
2.4.3 线……42
2.4.4 终止符……44
2.4.5 实体-关系(ER)图或 Chen 示意图……46
2.4.6 信息工程(I/E)……47
2.4.7 Barker 表示……48
2.5 小结……49
第 3 章 关系理论简介……50
3.1 关系数据建模……50
3.1.1 关系理论的起源……51
3.1.2 关系 DBMS 目标……51
3.2 Codd 的 RDBMS 规则……52
3.3 规范化……55
3.3.1 关系通用性质……56
3.3.2 第一范式(1NF)……60
3.3.3 第二范式(2NF)……62
3.3.4 第三范式(3NF)……63
3.3.5 Boyce/Codd 范式……64
3.4 反规范化……66
3.4.1 派生列……66
3.4.2 故意重复……67
3.4.3 故意删除或禁用约束……67
3.4.4 对范式的故意撤销……67
3.5 小结……67
第 4 章 分析级别……69
4.1 模型开发……69
4.1.1 不是流程图……71

4.1.2 数据关系规则……72
4.2 概念分析……73
4.2.1 概念模型中的实体……73
4.2.2 概念模型中的关系……74
4.2.3 概念模型示例……74
4.3 逻辑分析……75
4.3.1 逻辑模型中的实体……75
4.3.2 属性……76
4.3.3 逻辑分析示例……79
4.4 物理分析……80
4.4.1 表……81
4.4.2 物理分析示例……83
4.5 逆向工程分析……85
4.6 详细分析……85
4.6.1 实体级……86
4.6.2 基于键(KB)……87
4.6.3 全属性(FA)……89
4.7 小结……90
第 5 章 项目中的数据模型……92
5.1 项目……92
5.1.1 项目管理……92
5.1.2 项目的生命周期……97
5.2 项目类型……102
5.2.1 企业项目……102
5.2.2 事务项目——OLTP……102
5.2.3 数据仓库——企业报表……103
5.2.4 项目类型比较……103
5.3 模型目标……104
5.3.1 抽象模型……105
5.3.2 数据元素分析模型……106
5.3.3 物理设计模型……106
5.4 选择正确的模型……107
5.4.1 项目类型……108
5.4.2 模型目标……108
5.4.3 客户需求……108
5.4.4 建模技巧……109
5.5 小结……110

第 6 章　创建概念模型 …… 111
6.1　业务建模 …… 111
6.2　目标 …… 112
6.3　目标范围 …… 113
6.4　方法 …… 114
6.4.1　自顶向下 …… 114
6.4.2　自底向上 …… 115
6.5　记录流程规则：自顶向下 …… 116
6.5.1　Solitaire 纸牌游戏中的活动 …… 116
6.5.2　Solitaire 纸牌游戏的流程步骤 …… 116
6.5.3　建立活动描述 …… 118
6.5.4　标出重要的元素 …… 119
6.5.5　定义元素 …… 120
6.5.6　验证我们的工作 …… 121
6.5.7　综合为概念 …… 121
6.6　记录流程规则：自底向上 …… 123
6.6.1　记录活动规则 …… 123
6.6.2　建立规则描述 …… 124
6.6.3　标出重要元素 …… 124
6.6.4　定义元素 …… 125
6.6.5　两种方法的比较 …… 127
6.7　建立概念模型 …… 128
6.7.1　优化概念定义 …… 128
6.7.2　加入关系 …… 130
6.8　检查业务规则 …… 140
6.8.1　检查关系 …… 141
6.8.2　发布模型 …… 146
6.9　小结 …… 146
第 7 章　创建逻辑模型 …… 147
7.1　概念模型——指南 …… 147
7.1.1　确认模型 …… 149
7.1.2　使用反馈 …… 149
7.1.3　主题领域的范围 …… 150
7.2　逻辑数据建模 …… 150
7.3　对 Card 主题领域进行建模 …… 150
7.3.1　Card 实体分析 …… 151
7.3.2　Card 类别分析 …… 152
7.3.3　Card 联系 …… 153

7.3.4 Card 实体的详细内容 …… 157
7.3.5 Deck 和 Back Design 分析 …… 165
7.3.6 影子实体 …… 165
7.4 对“Card Movement”主题领域进行建模 …… 166
7.4.1 Card Movement 实体分析 …… 166
7.4.2 Movement 实体的细节 …… 173
7.5 对“Event”主题领域建模 …… 176
7.5.1 Event 实体分析 …… 177
7.5.2 Event 联系 …… 177
7.6 全图 …… 179
7.7 质量保证检查 …… 181
7.7.1 范式—从第一范式到 BC 范式 …… 181
7.7.2 过多/过少的属性 …… 183
7.7.3 多余的关系 …… 183
7.7.4 正确的角色名称 …… 184
7.7.5 实例表 …… 185
7.7.6 相关专业领域专家 …… 186
7.7.7 对等模型 …… 186
7.7.8 最后的工作 …… 186
7.8 小结 …… 186
第 8 章 逻辑到物理的转换 …… 188
8.1 项目状态 …… 188
8.2 逻辑到物理 …… 189
8.3 逻辑名到物理名 …… 189
8.4 从类别中创建表 …… 194
8.4.1 只留父类表 …… 194
8.4.2 只留子类表 …… 196
8.4.3 可扩展类别 …… 198
8.4.4 Solitaire 纸牌游戏实例 …… 200
8.5 检查影子实体 …… 201
8.6 确定主键 …… 202
8.6.1 复查主键 …… 203
8.6.2 加入数据类型和数据大小 …… 212
8.7 质量检查和额外的字段/表 …… 215
8.7.1 实例化表 …… 215
8.7.2 命名和定义 …… 215
8.7.3 复核需求 …… 215

8.7.4 讲故事 …… 215
8.7.5 确定数据管理员 …… 216
8.7.6 建立测试 DDL …… 216
8.8 其他潜在的问题 …… 218
8.8.1 增加操作表 …… 218
8.8.2 数据量资料 …… 218
8.8.3 活跃度资料 …… 219
8.8.4 模型功能 …… 220
8.9 小结 …… 220
第 9 章 直接设计物理模型 …… 221
9.1 现实的限制 …… 221
9.2 从哪里开始 …… 222
9.3 Solitaire 调查系统 …… 222
9.3.1 对见到的数据建模 …… 223
9.3.2 应用命名标准 …… 223
9.3.3 建立查找表 …… 225
9.3.4 继续寻找重要的数据集 …… 226
9.3.5 检查文本字段 …… 226
9.3.6 继续物理化 …… 227
9.3.7 质量和折衷 …… 229
9.4 更具挑战性的任务 …… 230
9.4.1 数据元素的分类 …… 231
9.4.2 文本字段 …… 233
9.5 其他的物理表 …… 242
9.5.1 操作型表 …… 242
9.5.2 数据转移表 …… 243
9.5.3 档案表 …… 244
9.6 小结 …… 244
第 10 章 多维数据建模 …… 245
10.1 多维模型基础 …… 245
10.1.1 多维设计的优点 …… 247
10.1.2 星型模式 …… 249
10.1.3 雪片模型 …… 250
10.1.4 Solitaire 纸牌游戏总体模型 …… 253
10.1.5 目标事实 …… 254
10.1.6 Game 数据集市 …… 259
10.1.7 GameMove 数据集市 …… 275

10.1.8 完成 281
10.2 小结 281
第 11 章 逆向工程设计数据模型 283
11.1 从哪里开始 283
11.2 数据结构分析 285
11.2.1 模型工具支持 285
11.2.2 手动过程 292
11.2.3 结构评估 295
11.3 数据分析 301
11.3.1 SELECT COUNT 301
11.3.2 SELECT COUNT/GROUP BY 301
11.3.3 SELECT COUNT DISTINCT 301
11.3.4 SELECT MIN 301
11.3.5 SELECT MAX 302
11.3.6 SELECT 302
11.3.7 数据评估 302
11.3.8 代码中的数据规则 302
11.4 前端分析 305
11.4.1 界面标签 305
11.4.2 数据关系界面规则 309
11.4.3 派生值 310
11.5 历史性的/描述性的 311
11.6 加注释 313
11.7 创建一个逻辑模型 313
11.7.1 命名 313
11.7.2 键 315
11.7.3 类别 316
11.7.4 其他规则 318
11.7.5 关系命名 320
11.8 完成 321
11.9 小结 322
第 12 章 模型沟通 323
12.1 为什么要增加更多的元素 323
12.2 元素排列 324
12.3 附加文本 325
12.3.1 名称和题目 325
12.3.2 版本符号 329

12.3.3 注释 331
12.3.4 图例 333
12.4 视觉增强 335
12.4.1 图形/图像/图标 335
12.4.2 其他可选 336
12.5 发布 337
12.5.1 半存取和公开存取——Web 337
12.5.2 开发文件存取 338
12.5.3 归档文件存取——文档库 338
12.6 小结 339
第 13 章 进一步数据分析 340
13.1 数据质量方面 340
13.1.1 逼真度分析 341
13.1.2 关键度分析 343
13.1.3 敏感性和保密性分析 345
13.1.4 管理工作(Stewardship) 346
13.1.5 横向核对 348
13.1.6 流程确认 348
13.1.7 风险及降低风险分析 349
13.2 数据模型作为一个知识架构 351
13.3 小结 362
第 14 章 元数据建模 363
14.1 定义元数据 363
14.1.1 技术元数据 365
14.1.2 业务元数据 365
14.1.3 实时元数据 366
14.2 元数据的重要性 367
14.3 一个元数据模型 369
14.3.1 概念元数据模型 369
14.3.2 逻辑元数据模型 371
14.3.3 物理元数据模型 373
14.4 建模人员与元数据 373
14.4.1 建模人员——元数据贡献者 373
14.4.2 建模人员——元数据消费者 374
14.5 元数据建模的未来 374
14.6 小结 375

第 15 章　数据建模工作习惯 …… 376
15.1　最坏的习惯 …… 376
15.2　团队模块化策略 …… 376
15.2.1　傲慢 …… 377
15.2.2　不愿妥协 …… 377
15.2.3　沟通困难 …… 377
15.2.4　固步自封 …… 378
15.2.5　回避问题 …… 378
15.2.6　伪君子行为 …… 378
15.2.7　说话做事欠考虑 …… 378
15.2.8　经常批评别人 …… 379
15.2.9　语言不够通俗易懂 …… 379
15.2.10　缺乏主动性 …… 379
15.3　延迟进度 …… 379
15.3.1　陷入分析麻痹 …… 380
15.3.2　从不开展交流 …… 380
15.3.3　一次一个任务 …… 380
15.3.4　从不承认您错了 …… 381
15.4　模型管理不善 …… 381
15.5　最好的习惯 …… 382
15.5.1　听取同事的意见 …… 382
15.5.2　适当妥协 …… 382
15.5.3　共享资料 …… 382
15.5.4　善于接受 …… 382
15.5.5　准时 …… 383
15.5.6　透明 …… 383
15.5.7　尊重别人 …… 383
15.5.8　有效地沟通和交流 …… 383
15.5.9　自我激励 …… 383
15.6　固守时间表 …… 383
15.6.1　Pareto 规则 …… 384
15.6.2　收益递减法则(The law of Diminshing Returns) …… 384
15.6.3　处理预期的事物 …… 385
15.6.4　利用您的主动性 …… 385
15.6.5　寻求帮助 …… 386
15.6.6　模型管理 …… 386
15.7　理解数据和设计 …… 388
15.7.1　逻辑到物理的转换 …… 388

15.7.2　关于物理数据的谬论……389
15.8　项目教训……390
15.8.1　用户定制解决方案项目……390
15.8.2　值得密切注意的短语……391
15.8.3　购买的解决方案项目……393
15.8.4　遗留系统与辨别分析……395
15.8.5　模型复查……397
15.8.6　经验值……397
15.8.7　责任感 vs 权限……398
15.9　小结……398

第1章　数据建模介绍

本章我们将讨论数据建模涉及哪些方面，在特定的组织中究竟哪些人从事数据建模工作。这一章的内容包括：

- 数据与信息的比较
- 什么是数据建模
- 在处理企业数据的生命周期中，数据建模到底起到哪些作用
- 谁在负责组织的数据建模任务
- 数据建模者需要参与哪些事情

1.1　什么是数据

数据就是对事实的描述，信息的定义同数据不一样，信息是对数据的有效解释。因为数据的最终目的就是为了提供有效的信息，所以必须通过建模以达到将数据转换为信息这个目的。虽然以前数据是用数据库中的文本和数值来表示的，但是目前数据的表现形式已经扩展到图片、声音、虚拟三维对象，以及由这些对象组合而成的复杂多媒体对象。数据特点不断进化发展的过程向我们提出了挑战——究竟怎样才能有效地使用技术，让数据发挥最大的能力。

除了数据本身，人们对数据关注的范围也发生了改变。我们曾经认为大部分数据应用集中在库存和账目支付系统方面。看看下面的例子：地面跟踪系统使用动态的卫星上行链路来帮助我们查找路线；虚拟电子游戏可以让来自全球任何一个角落的参与者实时玩游戏并且与对方交流；网上购物系统可以记住我们是谁，以及我们可能会对什么商品感兴趣；还有，公众建筑物上所安装的摄影机可以使用面像识别软件来进行监视。

人们想要收集的数据越来越多，同时他们也需要处理这些数据，以便提供累计、趋势、由逻辑驱动的优先顺序、来源，以及根据数据(元数据)对其他重要内容生成的数据映射等信息。为了满足这些需求，我们需要提供结构化的数据来支持不同的功能，比如：

- 联机事务处理
- 操作数据存储
- 数据仓库

目前人们认为，应该尽可能地以最基本、最不可分割、最基础的可复用组件的方法来收集和储存数据。只有这样，才能为创建新定义的信息提供条件。我们常常需要组合这些基本的数据元素，设计合适的储存和处理工具来成功地完成这项任务。

我们发现这些数据元素的过程决定了当前它们之间的关联方式及定义方式，从而在将来可以识别和使用这些数据，这个过程就称为数据建模。

1.2 什么是数据建模

数据建模是一种技巧，它可以记录存货信息、形状、尺寸、内容，还有在业务处理流程中数据元素使用的规律。业务处理的范围可能像一个多元的跨国公司那么复杂，也可以像在码头上接收货箱那么简单。最终的产品就好像一张图表，旁边附加上所有必要的参考资料，可以完全清楚地说明一切。

建立模型是为了以文档形式记录较抽象的想法，我们称它为概念模型。而有些模型的建立纯粹是为了以文档形式记录元素的规律和结构，我们称它为逻辑模型。最后一种模型可能大家都知道了，如果数据模型是设计数据库，就称它为物理模型，这个模型是编写代码建立表、视图、完整性约束的基础。如果从不同的观点来看待一个单独的业务处理，可以看出这 3 种基本模型之间的连接很松散，并且还可以互相补充和支持。

一个数据模型不必为了支持一个应用程序而提供完整的后端代码，它可以不包含安全授予功能，不包含数据库连接代码。它也不必显示尺寸和空间的需求(尽管有些数据库建模软件会让您添加这些方面的内容)。数据模型是建立数据库的蓝图，而不是数据库本身，是开发项目成功的基础，不过它也只是设计应用程序的一个基本要素。

建模的过程就是对现存事物的描述，也就是物理数据的 As-Is 状态，这样我们就可以查看现有数据库中数据的结构，以及在当前环境下数据库管理的规则。以上观点还只是局限于在一个单独的应用程序中，或者像数据库一样只是限定模式的一组表格。例如需要使用上百个这样的描述来记录企业中所有的信息。它可能使用几百个逻辑模型来描述数据的更多的理论规则，而不是物理上实现的方法。同样，我们还可以用很多概念模型来描述所有业务处理流程，尽管在这个阶段概念模型比其他阶段层次更高，想从一个角度说明所有的企业活动。

注意：
模型帮助我们观察当前和过去是怎样管理数据的。

数据建模还可以建立新东西，这就是 To-Be 阶段。通常我们会创建多个“To-Be”设计，让开发团队从中选择一个来进行实现。

注意：
我们建模是为了选择解决方案，因为即使对于一个相同的问题也可能会有几种不同的解决方案，要知道哪个是正确的并不容易。

然而，不管怎样利用数据，我们都需要理解数据的元素形式，并且在它的生命周期内应该按照规范小心地处理它。在我们继续探讨数据建模技巧之前，先看看一个典型的数据元素。

1.3 数据的生命周期

数据的生命周期阶段到底遵循什么规律？刚开始时新数据被当作新的事实使用，接着它就变成了一个历史事实，最终变成被遗忘的事实(其实对于这个阶段我们越来越觉得难以相信，因

为客户总是重复地要求挖掘出他们归档的数据，或者从备份中把数据恢复出来)。

图 1-1 并不是应用程序开发项目的生命周期，尽管很多其他的处理可能也有类似的阶段。这是任何数据元素要经过的步骤。

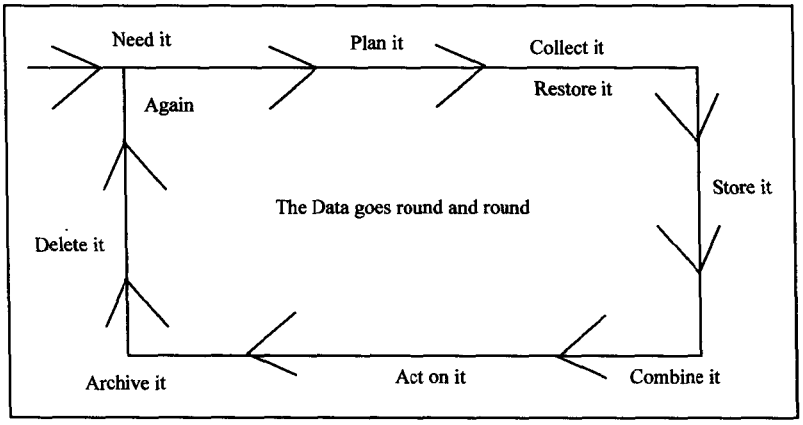

图　1-1

1. Need It

这是有人提出他需要一些数据的阶段。例如：昨天在衣阿华州得梅因市的气温是多少度？上个星期在 PG(宜在家长指导下观看的电影)电影开演之前，卖出了多少糖块？在第二次交接班时有多少个空停车位？下面让我们来看看用信用卡缴费的例子。假设您是一家小零售店的老板，想利用信用卡来支付货款，您关系了几个不同的信用卡发行公司了解到底要怎样做。这时您会发现不同的信用卡发行公司都会在您使用它们的服务时收取不同的费用，这些费用在不同的条件下还会有所不同，也就是说向客户提供服务而您得为此花钱。您需要分别收集在不同条件下每次信用卡交易的服务费金额，以便更好地分析利润。这种对数据元素需求的认识只是第一步。

几乎没有人会获取那些没有用的数据元素，数据必须对某人有价值，或者认为在将来它可能有价值，可以以后再拿出来使用它。在分析新数据元素的影响时，您可能又会发现其他一些可用的数据元素。

Need It 阶段是对新事实的认识，它是完成一个任务所必需的基础。现在我们已懂得为了得知在商业活动中的收益率，需要获得信用卡交易的服务费金额，通过查看信用卡服务的使用和花费，可以得到投资收益率(ROI，Return on Investment)，甚至制定下年度的预算。既然知道您需要信息，下一步应该做什么呢？

2. Plan It

接着就是考查和分析数据元素的阶段，关于频率、规模和业务规则的问题，还有获取和存储信息元素的方法，以及安全性、可靠性和服务质量都需要考虑到。

信用卡交易的服务费金额能有多少呢？它是按照销售额的百分比来收取的吗？或者是每次收取固定的金额？是当顾客用信用卡支付时公司马上要付出服务费，还是稍后付费呢？如果是这样的话，我们该如何把它和每次单独的交易关联起来呢？如果金额贷入我们的户头，那它是正值还是负值？