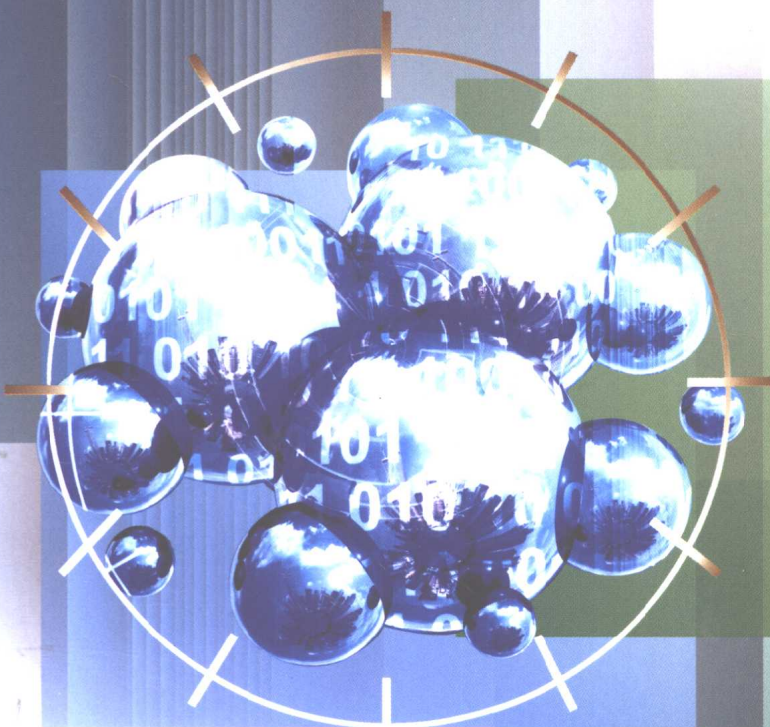




Microsoft®

SQL Server™ 2000

OLAP 解决方案—— 数据仓库与 Analysis Services



- SQL Server 2000 的 Analysis Services 为微软数据仓库提供了快速访问的解决方案，而数据仓库是信息界的明日之星。
- OLAP 的功能在 SQL Server 2000 中得到了极大的提高和增强，它以 Analysis Services 的形式闪亮登场。

沈兆阳 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



SQL Server 2000 OLAP 解决方案 —— 数据仓库与 Analysis Services

沈兆阳 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

SQL Server 2000 是一套集易学、易用以及功能强大于一身的数据库系统软件,除了传统的基本数据库功能外,还增加了联机分析处理(OLAP)的功能。

本书主要讲述如何使用 SQL Server 2000 的 Analysis Manager 来创建多维数据集,使用 Excel 的数据透视表功能来分析数据,并将结果发布到 Internet,从而实现数据的联机分析处理。书中主要讲述的概念除了 OLAP 以外,还包括数据仓库、数据转换服务(DTS)等。

本书适用于从事数据仓库设计、系统分析管理的中高级用户。

本书繁体字版名为《资料仓储与 Analysis Services——SQL Server 2000 OLAP 解决方案》,由文魁资讯股份有限公司出版,版权属沈兆阳所有。本书简体中文版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可,任何单位和个人不得以任何形式或任何手段复制或传播本书的部分或全部内容。

北京市版权局著作权合同登记号:图字 01-2001-3456 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名:SQL Server 2000 OLAP 解决方案——数据仓库与 Analysis Services

作 者:沈兆阳

责任编辑:高志风

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者:北京市牛山世兴印刷厂

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:21.75 字数:472 千字

版 次:2001 年 9 月第 1 版 2001 年 9 月第 1 次印刷

书 号:ISBN 7-302-04695-6/TP·2793

印 数:0001~5000

定 价:31.00 元

前言

数据仓库是信息业界的明日之星，数据库与联机事务处理(OLTP)是过去十几年来最热门的信息领域，它们的目标是以计算机来取代许多当前的作业，使得工商企业自动化。在现今竞争激烈的信息领域之中，人们已经开始把注意力由传统的数据库与 OLTP 加以转移，转而将注意力集中在积极层面的应用领域之上，工商企业也开始注意到他们所拥有的大量计算机数据，这些数据是公司极为重要的资产。

传统的数据库与 OLTP 平台并不是为了分析数据而设计的，为了要充分满足数据分析的请求，近几年来兴起了一种新的信息技术——数据仓库，工业分析师们预测在未来的 20 年之内，数据仓库将在信息业界占有可观的一席之地。

微软公司在 SQL Server 2000 上提供了 Analysis Services，它是数据仓库的解决方案，也是微软决策支持服务的主要组件。我们在数据仓库中常会看到一个名词 OLAP，OLAP 是 Online Analytical Processing 的缩写，即联机分析处理，它目前是决策支持的解决方案。

本书的第 1 至 4 章介绍了数据仓库的一些基本概念，详细地介绍了一个数据仓库的架构、数据仓库所使用的设计原理与数据结构，并且对一个数据仓库系统应具备的管理工具与使用的软硬件工作平台也做了详细的介绍。

为了使得内容更生动逼真，作者以 SQL Server 2000 的 Northwind 数据库为模板，循序渐进的引导读者从无到有添加一个数据仓库所使用的数据库，以作为 Analysis Services 的目标数据库。在本书中作者还使用 DTS 设计器来介绍如何由 Northwind 数据库将数据转移至目标数据库。

多维数据集是分析数据的基础，在本书中作者介绍了如何使用向导与编辑器来设计维度与多维数据集，并说明使用向导与编辑器的优缺点。除了分析管理器所提供的工具之外，Excel 也是一个用来分析多维数据集的前端工具，它是通过数据透视表来存取多维数据集的数据。本书除了详细的介绍如何使用 Excel 2000 的数据透视表功能来分析数据仓库的数据外，还介绍了多重维度表达式，它使得数据的分析多元化。

微软将数据仓库与 Web 整合在一起，用户可以使用 IE 5.x 通过互联网来分析数据，这使得数据仓库的功能向前迈进了一大步。当然，数据安全也是数据库的重要课题之一，本书于第 11 章详细的介绍了数据仓库用户与存取权限的管理。

本书的编写方法是采用“面向操作”的方式，就算是读者一时间无法理解文句中的内容，也可以通过实际的操作来体会。

Agg 15/5

目 录

第 1 章 数据仓库介绍	1
1.1 背景	1
1.2 开源与节流	1
1.3 分析数据的需求	2
1.4 什么是数据仓库	3
1.5 数据仓库与 OLTP 系统的差异	4
1.6 数据仓库的数据源	5
1.7 数据的整理与加载	6
1.8 需求分析	7
1.9 数据仓库的执行成员	7
1.10 制作数据仓库的步骤	8
1.11 自动化处理	10
1.12 数据仓库的备份与还原	10
1.13 数据仓库的种类	11
1.14 系统安全性	12
1.15 设计一个数据仓库的考虑因素	13
1.16 数据仓库的未来发展	14
1.17 本章小结	14
第 2 章 数据仓库架构	19
2.1 数据仓库的基本程序	19
2.2 抽取与加载数据	20
2.3 抽取数据的时间点	21
2.4 整理与转换数据	21
2.5 备份与备存程序	22
2.6 查询管理程序	22
2.7 数据仓库程序与系统管理器	23
2.8 加载管理器	24
2.9 仓库管理器	26
2.10 查询管理器	29
2.11 数据仓库的详细数据	30
2.12 数据仓库的架构	31

2.13	事实数据与维度数据.....	33
2.14	分区数据	33
2.15	集合数据与元数据.....	35
2.16	数据集市	36
2.17	数据集市使用时机.....	39
2.18	数据集市架构	40
2.19	制作数据集市的考虑因素.....	41
2.20	本章小结	42
第 3 章	数据仓库设计	47
3.1	星型雪花架构	47
3.2	识别事实与维度数据.....	50
3.3	设计事实表	51
3.4	设计维度表	58
3.5	设计星型雪花架构	59
3.6	查询向导	61
3.7	多重维度架构	62
3.8	使用集合信息	63
3.9	设计集合表	65
3.10	应生成哪些集合表.....	66
3.11	元数据	67
3.12	本章小结	70
第 4 章	管理工具与硬件平台	75
4.1	为何需要管理工具	75
4.2	管理工具分类	75
4.3	组合管理	76
4.4	调度管理	76
4.5	事件管理	77
4.6	数据库管理	78
4.7	数据仓库程序管理	79
4.8	备份与还原管理	80
4.9	数据源管理	80
4.10	元数据管理	80
4.11	运行性能监控与调整.....	81
4.12	SMP 硬件平台	81
4.13	群集	82
4.14	大量并行处理	83
4.15	本章小结	84

第 5 章 设计数据仓库架构	87
5.1 Northwind 数据库简介	87
5.2 Northwind 数据库架构介绍	89
5.3 Northwind 数据库表格的关系	95
5.4 需求分析	101
5.5 识别事实与维度数据	104
5.6 设计事实表	106
5.7 设计员工维度表	108
5.8 设计产品维度表	109
5.9 设计供货商维度表	110
5.10 设计顾客维度表	110
5.11 设计时间维度表	111
5.12 星型雪花架构	112
5.13 多重维度架构	113
5.14 本章小结	114
第 6 章 数据转移	117
6.1 DTS 设计器简介	117
6.2 新建目的数据库	118
6.3 激活 DTS 设计器	119
6.4 创建数据源与目的地	120
6.5 转移事实表	123
6.6 保存包	132
6.7 转移员工维度表	133
6.8 转移顾客、产品、供货商、时间维度表	138
6.9 执行包	143
6.10 设置表的主键与外键	144
6.11 本章小结	146
第 7 章 Analysis Services	149
7.1 安装 Analysis Services	149
7.2 分析管理器	151
7.3 新建数据仓库的数据库	154
7.4 为数据库指定数据源	155
7.5 激活多维数据集创建向导	157
7.6 选择事实数据表	158
7.7 创建星型架构维度	160
7.8 创建雪花架构维度	165
7.9 创建时间维度	168

7.10	存储多维数据集	170
7.11	处理多维数据集	172
7.12	浏览多维数据集的数据	173
7.13	本章小结	177
第 8 章	维度与维度编辑器	179
8.1	新建星型架构维度	179
8.2	为级别中的成员排序	181
8.3	在维度中新建级别	183
8.4	新建雪花架构维度	187
8.5	指定默认的成员	190
8.6	创建级别的属性	192
8.7	新建时间维度	194
8.8	使用维度编辑器新建多维数据集	199
8.9	为多维数据集创建衍生度量值	204
8.10	使用集合函数	205
8.11	计算平均值	208
8.12	隐藏内部使用的度量值	210
8.13	本章小结	211
第 9 章	数据透视表	213
9.1	创建 Excel 连接	213
9.2	浏览多维数据集的内容	217
9.3	细分成员	219
9.4	将多个维度加至坐标轴上	222
9.5	格式化数据透视表	226
9.6	创建图表	229
9.7	将数据透视表加入网页	234
9.8	创建本地多维数据集	236
9.9	使用 Excel 2000 创建多维数据集	239
9.10	本章小结	248
第 10 章	多重维度表达式	251
10.1	创建简单的 MDX 表达式	251
10.2	显示成员的名称	253
10.3	测试成员名称	260
10.4	显示成员的属性	263
10.5	将字符串转换为数值	264
10.6	计算百分比	266
10.7	参考前期的数值	269

10.8	使用 MDX Sample Application.....	271
10.9	将集合加至列滚动轴上.....	274
10.10	将集合加至行滚动轴上.....	276
10.11	将两个维度放在同一个轴上.....	277
10.12	在查询中创建计算成员.....	279
10.13	本章小结	283
第 11 章	数据安全	285
11.1	Analysis Services 数据安全架构.....	285
11.2	创建示范用户与组.....	287
11.3	为所有的用户新建角色.....	289
11.4	为数据库创建角色.....	291
11.5	管理数据库角色	296
11.6	完全限制一个维度.....	299
11.7	限制维度中的成员.....	301
11.8	维度的可视化控制.....	303
11.9	为维度指定默认的成员.....	306
11.10	单元格的安全性	308
11.11	本章小结	311
第 12 章	Analysis Services 的其他功能	313
12.1	备份与还原	313
12.2	虚拟多维数据集	316
12.3	设计存储体	323
12.4	使用网页存取分析数据.....	330
12.5	其他的功能	332
12.6	本章小结	335

第 1 章 数据仓库介绍

1.1 背 景

在过去的二十年之中，全世界已经投资了数以兆计的资源在计算机科技上，其主要目的就是使工商企业走上自动化之路。其实这股风潮的影响范围是全面性的，并非只是工商企业，大至全球性的跨国企业，小至两代所组成的单一家庭，甚至是活动用户(Mobile User)，都受到了这股风潮的影响。就工商企业组织结构面而言，现在已经有了所谓的“信息主管”(Information Director)的职位出现，目前可以说是信息化的时代。

自从全球信息网(World Wide Web/WWW)兴起以后，信息领域又向前迈进了一大步。就地域而言，它包括了全世界的每一个角落；就用户而言，它包括了各个层次的用户，上至工商企业的董事长，下至一年级学生。信息科技已经不再是一个陌生的名词，在任何时间、任何地点都存在着信息。

在已经过去的 20 世纪，工商企业信息化的目标绝大部分都是锁定在使用工作流程自动化，除了增加公司各项业务的执行绩效、提升企业的竞争能力之外，进而节省公司的经营成本。所谓自动化，指的是以计算机来执行原本应该以人工来进行的作业，这一范畴完全是锁定在工商企业现有的业务方面。

1.2 开源与节流

工商企业经营的大原则不外乎就是开源与节流，过去所致力于的自动化作业，我们可以把那些工作归于节流的范畴，这是过去几十年以来信息业界所致力目标。为了充分满足这种需求，数据库(Database)与联机事务处理(OnLine Transaction Processing/OLTP)的技术应运而生。

数据库与 OLTP 是过去十几年来最热门的信息领域之一，它们的目标完全是在以计算机来取代许多当前的作业，使得工商企业自动化。现以一家贸易公司为例：产品部门每天都要为新产品创建产品基本数据，业务部门每天都要输入订单数据，出货部门每天都要修订订单数据，这一切都是以信息科技来处理公司当前的业务。

如果上述的贸易公司想要拓展现有的业务或是增加营业项目，这时候公司的管理层次该如何处理呢？总不能凭空想象，当然一家公司的总经理可以根据已存储于计算机的历史数据来加以分析，但是如果计算机中已经存储了几年甚至是十几年的数据，面对如此庞大而且杂乱无章的数据，他应该如何处理。

在如今竞争激烈的信息领域之中，人们已经开始把注意力由传统的数据库与 OLTP 加以转移（似乎他们的技术方面已经够成熟了，已充分满足了大家的需求），从而将注意力集中在积极方面的应用领域之上，工商企业也开始注意到他们所拥有的大量计算机数据，这些数据是公司极为重要的资产。

1.3 分析数据的需求

如果工商企业能够充分利用他们所拥有的历史数据，加以分析、评估，相信一定可以改善他们管理决策的品质，以做出正确的抉择，从而提高公司的获利能力。

虽然工商企业已经知道这种数据分析的需求，但是他们大都面临到一个问题，因为目前经营运用的计算机系统并不是针对分析数据而设计的。也许有些管理人员已经经历过下面的过程：

- ◆ 总经理希望得到一项分析性的数据
- ◆ 将需求提交给公司的信息部门
- ◆ 信息部门根据需求设计程序
- ◆ 几天后信息部门将报表送交总经理
- ◆ 总经理发觉报表的格式不符合需求
- ◆ 总经理提交修正的需求给信息部门
- ◆ 信息部门根据新需求修改程序
- ◆ 几天后数据部门将新报表送交总经理

在上面的整个过程之中，受到影响的层次包括了 3 个部分：总经理、信息部门以及整个公司。各位也许会觉得很奇怪，为什么影响层次会涉及整个公司呢？我们将在下面加以说明。

当总经理提出需求给信息部门之后，也许要经过数次的需求修订，才能够得到令他满意（或是可以接受）的分析报表，整个周期少则一两天，多则要一两个星期。等到该总经理拿到他所需要的分析数据之后，很可能是缓不济急。

信息部门在收到了总经理的分析需求之后，开始指派程序员根据需求编写程序或是应急用的脚本(Script)文件，这增加了信息部门的工作开销。程序员所设计出来的程序是针对

特殊需求而设计的，往往只会使用一次（或是数次），这样的设计很不实际。

当程序员设计好程序开始进行作业时，因为统计分析需要使用大量的数据，所以会使用到许多的系统资源，这样很可能会影响到许多目前的作业。在这段期间内，也许某些用户会觉得计算机的运行速度很慢，当业务员输入一笔订单按下<Enter>按键后发觉要等几分钟才会完成交易。

当总经理收到了分析报表之后，他会开始作决策分析，但是他拿到的分析报表都是正确的数据吗？现以 BMW 汽车为例，在数据库中汽车的品牌名称很可能会出现 BMW、Bmw、bmw...，请问如果数据源有问题的话，那怎么可能会得到正确的分析数据呢？

1.4 什么是数据仓库

传统的数据库与 OLTP 平台并不是为了分析数据而设计的，用户可以在一个 OLTP 平台上安装数个应用系统，就应用范围而言，它们的数据很可能是不正确的甚至是互相抵触的。

为了要充分满足数据分析的需求，近几年来兴起了一种新的信息技术——数据仓库 (Data Warehouse)，工业分析师们预测在未来的 20 年之内，数据仓库将在信息业界占有一席之地。

概念上而言，一个数据仓库包含了一系列的关键性信息，它们可以用来管理并引导企业走向最可能的获利之路。这些信息似乎包含了所有与一个企业相关的数据，现以一超级市场为例，上架的产品、产品的价格、摆放的位置、折扣等信息都可以包含在数据仓库内。

现在我们要为数据仓库正式下一个定义，实际上各家学说已经为数据仓库下了一些定义，但是内容则是见仁见智，我们为数据仓库下的定义是：

数据仓库不仅包含了分析所需的数据，而且包含了处理数据所需的应用程序，这些程序包括了将数据由外部媒体转入数据仓库的应用程序，也包括了将数据加以分析并呈现给用户的应用程序。

根据我们为数据仓库所下的定义，一个数据仓库包括了数据以及负责管理与分析工作的程序管理器(Process Manager)，其主要目的是在及时提供可用的数据，使分析人员可以取得所需的正确统计信息，以作为管理决策的参考依据。

就数据的层面而言，一个数据仓库包含了下列类型的数据：

- ◆ 事实数据(Fact Data)
- ◆ 元数据(Meta Data)
- ◆ 维度数据(Dimension Data)

◆ 集合数据(Aggregation Data)

就应用程序的范围而言，一个数据仓库包含了下列类型的应用程序：

- ◆ 加载管理器(Load Manager)
- ◆ 仓库管理器(Warehouse Manager)
- ◆ 查询管理器(Query Manager)

请读者暂时不要去深究上面所提及的数据类型与应用程序，我们将于后面的章节为各位做详细的介绍。

数据仓库是一个整合式的、面向主题的、历史性的以及只读性的数据集，其主要目的是提供给管理员做管理决策的参考根据：

- ◆ 整合式的数据：将不同来源的数据以统一的规范，经过整理后存储在一起，所谓统一的规范指的是相同的数据类型、格式、度量等。
- ◆ 面向主题的数据：数据仓库存储的是与某一主题相关的数据，主题可以不止一个，但是不需要存储与主题无关的数据。
- ◆ 历史性的数据：数据仓库中的数据都是过去所发生的事实数据，但是并非存储过去的所有数据，而是在某一时间范围内的数据。
- ◆ 只读性数据：当我们将过去的的数据转入数据仓库之后，就不应该加以更新，但是可以以固定且周期性的方式定期加入新的历史数据。

1.5 数据仓库与 OLTP 系统的差异

数据库与 OLTP 的应用已经有很多年了，相信各位读者已经对它们有了一些基本的认识。数据仓库的发展并不是为了要取代传统的 OLTP 系统，它们是两个不同功能的系统，因此可以将它们看作是两个独立的系统。

OLTP 系统具有以下的特点：

- ◆ 是针对工商企业目前业务的自动化而设计的
- ◆ 是在协助工作人员执行既有的活动
- ◆ 可让多个用户同时使用信息系统
- ◆ 存储的数据比较偏重细节
- ◆ 数据的来源是用户日常工作的输入
- ◆ 数据库内容会随时被添加、删除以及更新
- ◆ 数据的处理是以交易为单位，交易时间短而且占用系统的资源少
- ◆ 设计数据库是采用正规化(Normalization)的设计方式

数据仓库系统具有以下的特点：

- ◆ 是针对工商企业数据分析需求而设计的，是一种新的业务
- ◆ 是在协助工作人员执行管理决策
- ◆ 一般而言，同时间不会有太多用户同时使用系统
- ◆ 存储的数据偏重较高级别的集合信息
- ◆ 数据的来源是 OLTP 系统
- ◆ 数据库的内容是历史数据，不会随时加以更新
- ◆ 数据的处理不是以交易为单位，一个用户的需求会占用大量的系统资源
- ◆ 设计数据库采用非正规化的设计方式

除了上述的特点之外，数据仓库还具有下列的特点：

- ◆ 将分散各处不同性质的数据整合到单数据集合之中
- ◆ 使用适合数据分析的结构来存储数据
- ◆ 包含由日常交易数据转换而来的数据，以提供管理员决策分析。
- ◆ 要能够及时从数据仓库中取得信息
- ◆ 数据仓库的数据在内部必须具有一致性
- ◆ 用户可以对取得的分析数据做细分与比较
- ◆ 数据仓库系统必须包含易于使用的浏览工具
- ◆ 在数据仓库中的数据必须完整而且正确

我们由上面的语句可以看出，数据仓库的特点与 OLTP 系统可以说是完全不同，也可以说它们之间是互补的。

1.6 数据仓库的数据源

数据仓库当然有其数据源，但是它的数据模式并不是来自终端用户(End User)每天所输入的数据。数据仓库的数据源包括下面两项：

- ◆ 工商企业的 OLTP 数据
- ◆ 由外部单位所购买的数据

事实上一个工商企业数据仓库的数据绝大部分都是来自其本身所拥有的 OLTP 数据，但是处理上并不是以日为单位。一般而言，数据仓库数据的时效会比较长，一家公司也许会拿去年的数据来加以分析，也许会希望分析前三年的数据，我们可以称这些保存数据的时间为“数据仓库时距”。

除了数据仓库时距以外，一个公司也会要求多久时间要将新的历史数据加入数据仓库。

例如：以上一个月为基准点、以上一季为基准点或是以上一年为基准点，我们可以称这种基准点为“数据仓库周期”。如果一个公司的数据仓库周期为一季，则该公司每隔三个月就要将数据由 OLTP 系统加载至数据仓库。

对于数据仓库的数据是来自公司所拥有的 OLTP 系统这一点，相信各位读者都能够理解，但是为什么数据仓库的数据可以由外部单位购买呢？假设用户是一位个人计算机公司的总经理，当用户需要得知明年个人计算机的全球性销售量预估数据时，该怎么办呢？是不是可以向 IDC 或是 Dataquest 购买数据来进行分析呢？这些数据不是由外部单位所购买的？

就市场面来看，数据仓库兴起了，现今社会对分析性数据的需求是越来越殷切，目前已经有一些专门出售数据的公司，相信在未来这将是一块很大的市场。用户可以视需要由外部公司购买数据，并将数据与数据仓库整合在一起。

1.7 数据的整理与加载

我们在前面曾经提到过，数据仓库的数据是来自公司的 OLTP 系统，但 OLTP 系统对数据的着眼点与数据仓库是完全不同的。

如果我们以一家汽车販售公司为例，该公司的职员一定很关心一笔订单的单价与购买数量的相关信息，但是对于一位顾客购买了哪一厂牌的车，可能就不关心，至于是在哪一个区域所售出的，很可能在输入数据时根本就懒得输入。

就该汽车販售公司而言，如果在输入汽车厂牌时没有对输入格式做统一的规范，例如：BMW 可能会以 BMW、Bmw、bmw、BMw 的类型出现，如果我们没有要求一定要输入一部汽车的販售区域，该项数据将是残缺不全的。

根据数据仓库要求输入必要而且正确的数据，但是 OLTP 系统应用已经很久了，用户无法管制过去的数据，而且只要是由人所输入的数据就一定会有错误发生。

在将数据由 OLTP 系统转入数据仓库之前，必须对 OLTP 的数据加以整理，以保证每一数据都具有统一的格式，而不会生成数据不一致的问题。

除了数据一致性的考虑之外，用户还要检查并更正错误的数据。程序员可以自行编写程序或是使用现有的工具来做数据检验的工作，这是一项繁杂的工作，但却是一项必要的工作，否则分析出来的结果将没有意义。

一般而言，检验数据所使用的硬件平台应该要与经营运用的 OLTP 系统区别开来，这是基于系统安全性与运行性能的考虑。在检验数据的过程之中，可能需要重复执行某一些步骤，例如：更正数据后要再执行数据一致性的检验、执行完数据一致性检验后又要更正数据，一直到对数据的正确性满意为止。

将数据分析完毕之后，就可以开始将数据加载至数据仓库系统，因为数据仓库所使用的数据量都非常之大，很可能需要耗费几小时甚至是几天的时间，用户可以分批并利用系统不忙碌的时段来加载数据。

加载数据至数据仓库的工作并不是经常要执行的，我们在前面曾经提到过数据仓库周期的概念，每经过一个数据仓库周期之后，必须把新的历史数据加载数据仓库。当然，可以运用递增式(Incremental)的方式来加载部分数据，这样可以节省数据加载的时间。请注意，在加载新历史数据之前也是要对数据加以整理。

1.8 需求分析

数据仓库与 OLTP 系统一样，在制作一个数据仓库之前也需要做需求分析。我们在前面曾提到过，数据仓库的数据是来自 OLTP 系统，但是并不是要将所有的 OLTP 数据都加入数据仓库之中。

事实上数据仓库所关心的事与 OLTP 系统是不同的，既然着眼点不同，所以没有必要将 OLTP 系统的数据完全拷贝至数据仓库之中。数据仓库需求分析的第一步是我们应访问公司内部的主管，由他们那里得到对数据分析面的需求。

需求分析在制作一个数据仓库的过程中是很重要的一个步骤，当一个数据仓库设计并制作完成后，如果要修改其组织结构将是一件很困难而且很耗时的工作，所以一个数据仓库在设计完成后理应能够支撑一段时间。

当然，用户的需求是无止境的，我们不可能永远不变动数据仓库的架构，也不可能禁止用户提出新的需求，但是在设计之初最好是能够将所有的需求搜索齐全（或是尽量搜索齐全），在设计方面做整体性的设计，在制作方面可以分阶段执行。

1.9 数据仓库的执行成员

在开始进行一个数据仓库的计划，第一件事就是召集足够的执行成员，并且成立一个项目小组。就功能方面来加以区分，一个数据仓库计划需要使用到下列的成员：

- ◆ 项目经理(Project Manager)
- ◆ 数据库管理员(Database Administrator/DBA)
- ◆ 系统管理员(System Administrator)
- ◆ 系统分析员(System Analyst)
- ◆ 数据库设计师(Database Designer)

- ◆ 程序员(Programmer)
- ◆ 技术员(Technician)
- ◆ 操作员(Operator)

项目经理对数据仓库计划负全部的责任,他负责整个项目的协调工作,此外对于项目的进度周期、资源的购置分配、工作优先职位制定、风险评估等工作也是由项目经理负责。

数据库管理员负责项目内与数据库相关的整体性管理工作,他是由项目经理指挥调度,数据库架构的设计、运行性能的调整、磁盘空间的配置等工作都是由数据库管理员负责。

系统管理员负责计算机硬件设备与操作系统的管理工作,他是由项目经理调度指挥工作,并指挥技术员与操作员来执行日常的工作,计算机系统运行性能的调整、日常工作的设计与制定、异常状况的处理等工作都是由系统管理员负责。

系统分析员负责与用户进行连接协调,以进行数据仓库的需求分析,他是由项目经理调度指挥,并指挥程序员与数据库设计师来进行数据库与程序设计的工作。

数据库设计师与程序员负责数据库设计与程序设计的工作,他们是由系统分析员调度指挥。技术员负责计算机中心各项设备的保养与维护工作,操作员负责各种日常工作的处理,如定期工作的执行、数据库的备份等,他们都是由系统管理员指挥调度。

虽然制作一个数据仓库计划需要以上专长的成员,但是一个人可以在计划中身兼数种角色,例如:项目经理可以身兼一部分系统分析员的工作、一位数据库管理员可以兼任数据库设计师的工作、技术员与操作员可以由一人担任。

1.10 制作数据仓库的步骤

一个数据仓库计划的制作包含了许多工作,我们可以将整个过程分为一些步骤,以降低整个计划执行上的风险。某一些步骤在执行上是有其先后顺序的,而某一些步骤却可以平行的执行。只有每一个步骤都有高品质产出,这样才会有高品质的数据仓库系统。

一般而言,一个数据仓库计划的执行周期大约是 18 到 24 个月,这是一个很长的周期,所以应该要有一个完善的计划。制作数据仓库目前并没有既定的步骤与流程,但可以将整个过程分为下列的步骤:

- ◆ 信息策略的制定
- ◆ 作业案例的分析
- ◆ 整体需求的制定
- ◆ 用户的教育训练
- ◆ 系统架构的制定