第一图书网, tushu007.com

<<Microsoft数据仓库工具箱>>

图书基本信息

书名：<<Microsoft数据仓库工具箱>>

13位ISBN编号：9787302283362

10位ISBN编号：7302283362

出版时间：2012-5

出版时间：清华大学出版社

作者：[美] Joy Mundy

页数：460

字数：736000

译者：包战,孔祥亮

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

内容概要

作为数据仓库和商业智能领域最有影响力的思想领袖，KimballGroup开发了一系列开拓性的技术，均已成为DW/BI系统设计、开发和管理的业界标准。
在这本畅销书的新版本中，KimballGroup中经验丰富的专家介绍了如何快速掌握SQLServer的新商业智能版本：SQLServer2008R2。

《Microsoft数据仓库工具箱（第2版）--使用SQLSever2008R2和MicrosoftBI工具集》涵盖了SQLServer2008R2中全套的数据仓库和BI工具，介绍了项目的整个生命周期，包括设计、开发、部署和维护。
《Microsoft数据仓库工具箱（第2版）——使用SQLSever2008R2和MicrosoftBI工具集》更新了上一版的大量内容，介绍了SQLServer2008R2的新功能，例如PowerPivot和MasterDataServices，还用翔实的示例说明如何更好地应用《Microsoft数据仓库工具箱（第2版）：使用SQL Server 2008 R2和Microsoft BI工具集》描述的技术。

作者分享了他们使用Microsoft工具构建DW/BI系统的经验，读者可以从中了解他们遇到的挑战，分享他们的成功。
还可以学习在使用Kimball生命周期建立自己的DW/BI系统时，应如何遵循4个基本原则：关注业务，构建信息基础架构，提供有意义的增量价值以及交付完整的解决方案。
有了这4个原则，就可以构建成功的DW/BI系统，以支持大多数公司都有的商业智能需求。

作者简介

Joy Mundy在斯坦福大学、WebTV和Microsoft SQL Server产品研发小组中一直关注DW/BI系统。
Joy在塔夫茨大学获得经济学学士学位，然后在斯坦福大学获得工程经济系统硕士学位。

Warren Thornthwaite自1980年起就开始了DW/BI生涯。
在离开Metaphor咨询公司后，他为斯坦福大学和WebTV工作。
Warren从密西根州立大学获得传媒学的学士学位，从宾夕法尼亚州的沃顿商学院获得决策学的MBA学位。

Ralph Kimball是Kimball Group的创立者，自从20世纪80年代中期开始，他就是DW/BI行业中维度方法的思想领袖。

书籍目录

第 i 部分 需求、现实情况和体系结构
第1章 定义业务需求
1.1 长期成功的最重要的决定因素
1.2 adventure works cycles简介
1.3 揭示业务价值
1.3.1 获得赞助商关系
1.3.2 定义企业级业务需求
1.4 设定业务需求的优先级
1.5 项目规划
1.6 收集项目需求
1.7 小结
第2章 业务过程维度模型设计
2.1 维度建模概念和术语
2.1.1 事实表
2.1.2 维度
2.1.3 整合事实和维度
2.1.4 总线矩阵、一致性维度和交叉探查
2.2 其他设计概念和技术
2.2.1 代理键
2.2.2 渐变维度
2.2.3 日期
2.2.4 退化维度
2.2.5 雪花模型
2.2.6 多对多维度或多值维度
2.2.7 层次结构
2.2.8 聚合维度
2.2.9 无意义维度
2.2.10 3种事实表类型
2.2.11 聚合
2.3 维度建模过程
2.3.1 准备工作
2.3.2 数据剖析和研究
2.3.3 构建维度模型
2.3.4 开发详细维度模型
2.3.5 模型测试和细化
2.3.6 评审和验证模型
2.4 案例研究：adventure works cycles订单维度模型
2.4.1 订单事实表
2.4.2 维度
2.4.3 确定订单业务过程的维度属性和事实
2.4.4 初始订单模型的最终草图
2.4.5 详细订单维度模型开发
2.4.6 最终的维度模型
2.5 小结
第3章 工具集

3.1 microsoft dw/bi 工具集
3.2 使用microsoft工具集的原因
3.3 microsoft dw/bi系统的体系结构
3.3.1 包含analysis services的原因
3.3.2 存储在关系数据库中的原因
3.3.3 etl不是可选的
3.3.4 master data services的作用
3.3.5 交付bi应用程序
3.4 microsoft工具概述
3.4.1 需要的产品
3.4.2 sql server开发和管理工具
3.5 小结
第4章 系统设置
4.1 系统规模的考虑事项
4.1.1 计算数据卷
4.1.2 确定应用复杂度
4.1.3 估计并发用户数
4.1.4 评估系统可用性需求
4.1.5 系统的规模
4.2 系统配置考虑事项
4.2.1 内存
4.2.2 一体化还是分布式
4.2.3 存储系统考虑事项
4.2.4 处理器
4.2.5 高可用性设置
4.3 软件安装和配置
4.3.1 开发环境的软件需求
4.3.2 测试和产品系统的软件需求
4.3.3 操作系统
4.3.4 sql server关系数据库设置
4.3.5 analysis services设置
4.3.6 integration services设置
4.3.7 reporting services设置
4.4 小结
第ii部分 建立和填充数据库
第5章 创建关系数据仓库
5.1 开始
5.2 完成物理设计
5.2.1 代理键
5.2.2 字符串列
5.2.3 空或非空
5.2.4 常规事务列
5.2.5 数据表和列的扩展属性
5.3 定义存储器并创建约束和支持对象
5.3.1 创建文件和文件组
5.3.2 数据压缩
5.3.3 实体和引用完整性约束

5.3.4 初始索引和数据库统计
5.3.5 聚合表
5.3.6 创建数据表视图
5.3.7 插入未知成员行
5.3.8 create table语句示例
5.4 分区表
5.4.1 分区表的工作方式
5.4.2 管理分区表
5.5 收尾
5.5.1 中间表
5.5.2 元数据设置
5.6 小结
第6章 主数据的管理
6.1 管理主引用数据
6.1.1 属性不完整
6.1.2 数据集成
6.1.3 系统集成
6.1.4 主数据管理系统和数据仓库
6.2 sql server主数据服务
6.2.1 模型定义功能
6.2.2 数据管理功能
6.3 创建简单的应用程序
6.3.1 业务场景
6.3.2 尽可能简单
6.3.3 创建mds模型
6.3.4 加载子类别成员
6.3.5 改进模型
6.3.6 导出到数据仓库
6.4 小结
第7章 设计和开发etl系统
7.1 确定需求
7.2 制定etl计划
7.3 sql server integration services概述
7.3.1 控制流和数据流
7.3.2 ssis程序包的体系结构
7.4 etl的主要子系统
7.5 提取数据
7.5.1 子系统1：数据剖析
7.5.2 子系统2：更改数据捕获系统
7.5.3 子系统3：提取系统
7.6 清理和一致化数据
7.6.1 子系统4：数据清理系统
7.6.2 子系统5：错误事件模式
7.6.3 子系统6：审核维度汇编器
7.6.4 子系统7：重复数据删除系统
7.6.5 子系统8：一致化系统
7.7 传递数据以用于展示

7.7.1 子系统9：渐变维度管理器
7.7.2 子系统10：代理键生成器
7.7.3 子系统11：层次结构管理器
7.7.4 子系统12：特殊维度管理器
7.7.5 子系统13：事实表构建器
7.7.6 子系统14：代理键管道
7.7.7 子系统15：多值维度桥接表构建器
7.7.8 子系统16：迟到数据的处理程序
7.7.9 子系统17：维度管理器
7.7.10 子系统18：事实提供程序系统
7.7.11 子系统19：聚合构建器
7.7.12 子系统20：olap多维数据集构建器
7.7.13 子系统21：数据传播管理器
7.8 管理etl环境
7.9 小结
第8章 核心analysis services olap数据库
8.1 analysis services olap概述
8.1.1 使用analysis services的原因
8.1.2 不使用analysis services的原因
8.2 设计olap结构
8.2.1 规划
8.2.2 起始工作
8.2.3 创建项目和数据源视图
8.2.4 维度设计
8.2.5 创建和编辑维度
8.2.6 创建和编辑多维数据集
8.3 物理设计的考虑因素
8.3.1 理解存储模式
8.3.2 分区计划
8.3.3 设计性能聚合
8.3.4 部署计划
8.3.5 处理整个多维数据集
8.3.6 开发增量处理计划
8.4 小结
第9章 实时商业智能的设计需求
9.1 实时分类
9.1.1 实时的含义
9.1.2 需要实时的人员
9.1.3 对实时的权衡
9.2 场景和解决方案
9.2.1 实时地执行报表
9.2.2 通过缓存向报表提供服务
9.2.3 用镜像和快照创建ods
9.2.4 用复制功能创建ods
9.2.5 建立biztalk应用程序
9.2.6 建立实时关系分区
9.3 小结

第iii部分 商业智能应用程序的开发
第10章 在reporting services中构建bi应用程序
10.1 bi应用程序概述
10.2 商业智能应用程序的价值
10.3 报表设计高层次的体系结构
10.3.1 回顾报表设计的业务需求
10.3.2 reporting services的体系结构
10.3.3 使用reporting services作为标准的报表设计工具
10.3.4 reporting services的评价
10.4 报表设计系统的设计和开发过程
10.4.1 报表设计系统的设计
10.4.2 报表设计系统的开发
10.5 报表的构建和传送
10.5.1 规划和准备
10.5.2 创建报表
10.5.3 报表设计的运行
10.6 即席报表设计选项
10.6.1 报表模型
10.6.2 共享数据集
10.6.3 报表部件
10.7 小结
第11章 powerpivot和excel
11.1 使用excel进行分析和报表设计
11.2 powerpivot体系结构
11.3 创建和使用powerpivot数据库
11.3.1 开始使用powerpivot
11.3.2 powerpivot表的设计
11.3.3 使用powerpivot创建分析表
11.3.4 powerpivot for excel的观察和指导原则
11.4 powerpivot for sharepoint
11.4.1 powerpivot sharepoint用户体验
11.4.2 服务器级别的资源
11.4.3 powerpivot的监控和管理
11.5 powerpivot在托管dw/bi环境下的作用
11.6 小结
第12章 bi门户和sharepoint
12.1 bi门户
12.1.1 bi门户的规划
12.1.2 对设计的影响
12.1.3 业务过程的类别
12.1.4 额外的功能
12.1.5 建立bi门户
12.2 把sharepoint用作bi门户
12.2.1 体系结构和概念
12.2.2 安装sharepoint
12.2.3 安装测试系统
12.2.4 完成bi门户

12.2.5 biportal站点模板的其他功能
12.2.6 研究sharepoint
12.3 小结
第13章 数据挖掘的加入
13.1 数据挖掘的定义
13.1.1 基本的数据挖掘术语
13.1.2 数据挖掘的业务应用
13.1.3 角色和责任
13.2 sql server数据挖掘体系结构概述
13.2.1 数据挖掘设计环境
13.2.2 构建、部署和处理
13.2.3 挖掘模型的访问
13.2.4 integration services和数据挖掘
13.2.5 其他功能
13.2.6 体系结构的总结
13.3 microsoft数据挖掘的算法
13.3.1 决策树
13.3.2 na?ve bayes算法
13.3.3 群集
13.3.4 顺序群集
13.3.5 时间序列
13.3.6 关联
13.3.7 神经网络
13.4 数据挖掘的过程
13.4.1 业务阶段
13.4.2 数据挖掘阶段
13.4.3 操作阶段
13.4.4 元数据
13.5 数据挖掘的示例
13.5.1 案例研究：给城市分类
13.5.2 案例研究：产品推荐
13.6 小结
第iv部分 dw/bi系统的部署和管理
第14章 设计和实施安全保护
14.1 确定安全管理员
14.2 保护硬件和操作系统
14.2.1 保护操作系统
14.2.2 使用windows集成安全认证
14.3 保护开发环境
14.4 保护数据
14.4.1 向内部用户提供开放的访问
14.4.2 分条列出敏感数据
14.4.3 保护各种类型的数据访问
14.5 保护dw/bi系统的组件
14.5.1 reporting services安全
14.5.2 analysis services的安全
14.5.3 关系dw的安全

14.5.4 integration services安全
14.6 使用情况的监控
14.7 小结
第15章 元数据规划
15.1 元数据的基础
15.1.1 元数据的目标
15.1.2 元数据种类
15.1.3 元数据库
15.2 元数据标准
15.3 sql server 2008 r2元数据
15.3.1 跨工具组件
15.3.2 关系引擎的元数据
15.3.3 analysis services
15.3.4 integration services
15.3.5 reporting services
15.3.6 master data services
15.3.7 sharepoint
15.3.8 外部元数据的源
15.3.9 对sql server元数据的期待
15.4 实用的元数据方法
15.4.1 元数据策略的创建
15.4.2 业务元数据报表
15.4.3 过程元数据报表设计
15.4.4 技术元数据报表
15.4.5 过程元数据的管理
15.5 小结
第16章 部署
16.1 建立环境
16.2 测试
16.2.1 开发测试
16.2.2 系统测试
16.2.3 数据质量保证的测试
16.2.4 性能测试
16.2.5 可用性的测试
16.2.6 测试小结
16.3 部署到生产环境中
16.3.1 关系数据库的部署
16.3.2 integration services程序包的部署
16.3.3 analysis services数据库的部署
16.3.4 reporting services报表的部署
16.3.5 master data services部署
16.4 数据仓库和bi文档
16.4.1 核心描述
16.4.2 其他文档
16.5 用户的培训
16.6 用户支持
16.7 台式计算机的准备和配置

16.8 小结
第17章 运行与维护
17.1 提供用户支持
17.1.1 bi门户的维护
17.1.2 bi应用程序的扩展
17.2 系统管理
17.2.1 dw/bi系统的控制
17.2.2 性能的监控
17.2.3 使用情况的监控
17.2.4 磁盘空间的管理
17.2.5 服务和可用性的管理
17.2.6 dw/bi系统的性能调整
17.2.7 备份和恢复
17.2.8 etl程序包的执行
17.3 小结
第18章 目前的需要及未来的展望
18.1 发展dw/bi系统
18.2 生命周期和常见的问题回顾
18.2.1 阶段ⅰ -- 需求、现实、体系结构和设计
18.2.2 阶段ⅱ -- 数据库的开发
18.2.3 阶段ⅲ-- 开发bi应用程序和门户环境
18.2.4 阶段ⅳ-- dw/bi系统的部署和管理
18.2.5 迭代和扩展
18.3 microsoft bi工具集中受欢迎的部分
18.4 未来的方向：改进的空间
18.4.1 查询工具
18.4.2 元数据
18.4.3 关系数据库引擎
18.4.4 analysis services
18.4.5 master data services
18.4.6 集成
18.4.7 顾客关注点
18.5 小结

章节摘录

版权页： 插图： Albert Einstein指出了使用维度模型的主要原因：能使任何事情尽可能简单，但绝不是简化。
结果证明，简单是相对的。
人们普遍认为，在数据仓储和商业智能中，维度模型是给用户显示信息的首选结构。
维度模型比典型的源系统规范化模型更便于用户理解，即使维度模型所包含的内容与规范化模型完全一致也是如此。
维度模型中的表更少，信息分组为对用户有意义的、一致的业务类别。
这些类别称为维度，有助于用户浏览模型，因为可以忽略与特定分析无关的全部类别。
但是，尽可能简洁并不意味着模型一定简单。
模型必须反映业务，而业务通常都比较复杂。
如果简化得过多，一般来说只表示了聚合数据，模型就会丢失对理解业务非常重要的信息。
无论如何进行数据建模，数据内容内在的复杂性都使大多数人最终愿意通过结构化报表和分析应用程序来访问DW／BI系统。
要达到第二个目标，即良好的性能，可能更需要依赖平台。
在关系环境下，维度模型能提供更好的查询性能，这是因为在创建维度时采用了反规范化的方法。
通过预先连接各种层次结构和查询表，优化程序考虑的连接路径较少，创建的中间临时表也更少。
通过采用维度结构而不是完全规范化的结构，SQL Server关系数据库的分析查询性能会更好——通常会好得多。
从更基本的层面上看，优化程序可以识别出维度模型，并利用其结构大幅度减少返回的行数。
这称为“ 星型连接优化 ”，当然这也是企业版的功能。
在Analysis Service OLAP环境下，专门把引擎设计成支持维度模型，主要通过在维度之内和之间进行预聚合来提高性能。
达到第三个目标需要采用各种设计模式，以创建出精确捕获和跟踪业务的模型。
下面从最基本的模式开始。
维度模型由一个或者多个中心事实表和与其相关的维度构成。
事实表位于中心，而维度环绕在其周围，类似于星型结构，因此又把维度模型称为星型模式。
为避免混淆，本书始终使用维度模型这一术语。
从关系数据建模的角度来看，维度模型是由一个规范化的事实表和反规范化的一些维度表组成的。
本节将定义维度模型、事实表和维度等基本组成部分，以及在处理随时间而发生的改动时涉及的一些重要概念。
2.1.1 事实表 每个事实表都包含与特定业务过程相关的度量，例如下一个订单，显示一个网页，接待一位患者或者处理一个顾客的服务请求。
事实表中的一条记录是一个度量事件，这些事件通常有数字值，用来量化大量的事件，如订购的数量、销量或呼叫持续时间等。
这些数字称为事实（或在Analysis Services中称为度量）。
事实表的键通常是多部分的键，由业务事件涉及的每个维度表的外键子集构成。
1.事实 大多数事实都是数字型的，且可以累加（如总销量或单位销量），这意味着可以对所有维度求和。
可加性很重要，因为DW／BI应用程序很少检索单个事实表记录。
用户查询一般一次选择数百或数千条记录并合计。
查询去年每个月的销售量仅返回12行结果，但它合计了成千上万行（或者更多）。
一些事实具有半可加性（如市场份额或账目结算），一些事实则具有非可加性（如单价等）。
并非所有的数字型数据都是事实，包括离散的描述信息，例如用来描述产品的包装尺寸或重量，以及用来描述商店面积的平方英尺。
通常这些不大变化的数字值是维度表中的描述性属性。

这种描述信息自然也用来约束查询，而不是在计算中求和。
决定将一个数据元素作为维度或事实的一部分时，这一区别是非常有用的。
一些业务过程跟踪没有任何实际度量的事件。
如果发生这种事件，源系统就添加一项，反之源系统就没有任何记录。
这类事件的常见示例有雇佣活动，例如聘用或者解雇某个雇员；参与事件的人数，例如学生何时加入某个班级等。
跟踪这类事件的事实表没有任何实际度量值，因此也称为没有事实的事实表。
通常的做法是向表中添加一列，该列称为EventCount，并且包含数字1。
这就为用户提供了一种统计事件数量的简便方法，即通过累加EventCount事实来实现。
有些事实是从其他事实派生或计算得到的，如纯收入是从销售总额减去营业税后得到的。
一些半可加性的事实可以使用基于查询上下文的派生列来处理，例如月底结算是累加多个账户，而不是对日期进行累加。
对于非可加性的单价，可以把它定义为平均单价，即Total Amount（总额）除以Total Quantity（总数量）。
有几种方法可以处理这种派生或计算得到的事实。
可以把它们作为ETL过程的一部分计算，并在事实表中存储：也可以把它们存储在事实表的视图定义里；还可以把它们包含在Analysis Services数据库的定义中。
唯一不能接受的方式是把这一过程留给用户处理。

编辑推荐

《Microsoft数据仓库工具箱(第2版):使用SQL Server 2008 R2和Microsoft BI工具集》描述如何使用Microsoft SQL Server产品集设计并构建一个成功的商业智能系统及其底层的数据仓库数据库。

第一图书网, tushu007.com

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com