数据仓库与数据挖掘

主讲人：龚卫华（博士）
研究方向：网格计算，数据库系统

教材与参考书

- 陈文伟，数据仓库与数据挖掘教程，清华大学出版社
- 安淑芝等编著.数据仓库与数据挖掘.清华大学出版社.
- Jiawei Han，Micheline Kamber.数据挖掘概念与技术.范明等译.机械工业出版社.
- 张云涛,龚玲著.数据挖掘原理与技术.电子工业出版社.**(IBM软件学院)**

主要内容与考核方式

- 内容提要：
  - 数据仓库->DW的设计和OLAP操作
  - 数据挖掘->关联规则、聚类和分类算法
- 考核方式：
  - 实验：20%
    - Sql server 2000 Analysis Service
  - 平时：10%
  - 开卷试题：70%

绪论

数据爆炸问题

- 自动数据收集工具和成熟的数据库技术使得大量的数据被收集，存储在数据库、数据仓库或其他信息库中以待分析。

- 我们拥有丰富的数据，但却缺乏有用的信息
- 解决方法：**数据仓库技术和数据挖掘技术**
  - 数据仓库(Data Warehouse)和在线分析处理(OLAP)
  - **数据挖掘：**在大量的数据中挖掘感兴趣的知识（规则，规律，模式，约束）

数据库技术的演化 (1)

- 1960s和以前:
  - 文件系统
- 1970s:
  - 层次数据库和网状数据库
- 1980s早期:
  - 关系数据模型, 关系数据库管理系统(RDBMS)的实现

数据库技术的演化 (2)

- 1980s晚期:
  - 各种高级数据库系统(扩展的关系数据库,面向对象数据库等等.)
  - 面向应用的数据库系统 (空间数据库，时序数据库，多媒体数据库等等）
- 1990s:
  - 数据挖掘, 数据仓库（Inmon），多媒体数据库和网络数据库
  - 95年数据仓库流行：IBM的BI，微软的SQL Server绑定OLAP服务器
- 2000s
  - 流数据管理和挖掘
  - 基于各种应用的数据挖掘
  - XML数据库和整合的信息系统

数据仓库的用途（三种）

- 信息处理
  - 支持查询和基本的统计分析，并使用交叉表、表、图表和图进行报表处理
- 分析处理
  - 对数据仓库中的数据进行多维数据分析
  - 支持基本的OLAP操作，切块、切片、上卷、下钻、转轴等
- 数据挖掘
  - 从隐藏模式中发现知识
  - 支持关联分析，构建分析性模型，分类和预测，并用可视化工具呈现挖掘的结果

数据仓库的应用价值

- 传统的数据库针对OLTP应用理想，但不适合决策分析。原因：
  - 1. 决策处理的系统响应时间
    - 可能很长，遍历大部分数据
  - 2.决策数据需求的问题
    - 动态更新，数据需要正确的集成、汇总、概括。
  - 3. 决策数据操作的问题
    - 日常事务不能满足决策需要，希望对数据进行多种形式的操作。
- 传统DB的操作型数据与DW的分析型数据区别

操作型数据	分析型数据
细节的	综合的或提炼的
在存取瞬间是准确的	代表过去的
可更新	不更新
操作需求事先可知道	操作需求事先不知道
生命周期符合SDLC	完全不同的生命周期
对性能要求高	对性能要求宽松
一个时刻操作一个单元	一个时刻操作一个集合
事务驱动	分析驱动
面向应用（OLTP）	面向分析（DSS）
一次操作数据量小	一次操作数据量大
支持日常操作	支持管理需求

操作型DBS与数据仓库

- 操作型DBS的主要任务是联机事务处理OLTP（On Line Transaction Processing）
  - 日常操作: 购买，库存，银行，制造，工资，注册，记帐等
- 数据仓库的主要任务是联机分析处理OLAP（On Line Analytical Processing）
  - 数据分析和决策支持（DSS），支持以不同的形式显示数据以满足不同的用户需要

OLTP VS. OLAP （1）

- 用户和系统的面向性
  - 面向顾客（事务） VS. 面向市场（分析）
- 数据内容
  - 当前的、详细的数据 VS. 历史的、汇总的数据
- 数据库设计
  - 实体—联系模型(ER)和面向应用的数据库设计 VS. 星型/雪花模型和面向主题的数据库设计

OLTP VS. OLAP (2)

- 数据视图
  - 当前的、企业内部的数据 VS. 经过演化的、集成的数据
- 访问模式
  - 事务操作 VS. 只读查询（但很多是复杂的查询）
- 任务单位
  - 简短的事务 VS. 复杂的查询
- 访问数据量
  - 数十个 VS. 数百万个

OLTP VS. OLAP (3)

- 用户数
  - 数千个 VS. 数百个
- 数据库规模
  - 100M-几GB VS. 100GB-数TB
- 设计优先性
  - 高性能、高可用性 VS. 高灵活性、端点用户自治
- 度量
  - 事务吞吐量 VS. 查询吞吐量、响应时间
- 国际评测标准(http://www.tpc.org/)
  - TPC-C VS. TPC-H

为什么需要一个分离的数据仓库?

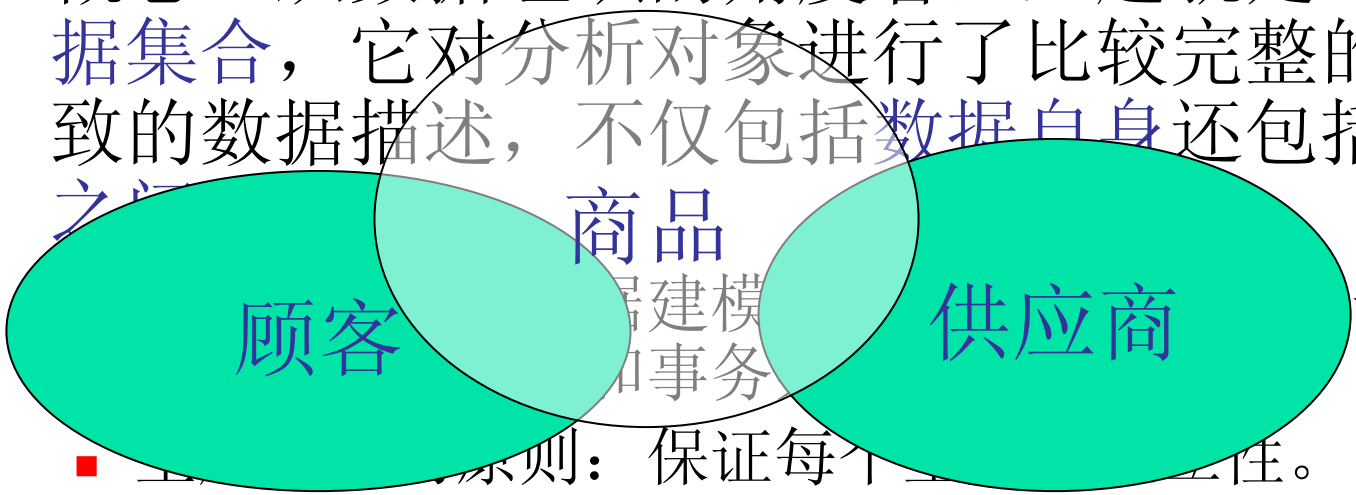
- 提高两个系统的性能
  - DBMS是为OLTP而设计的：存储方式,索引, 并发控制, 恢复
  - 数据仓库是为OLAP而设计：复杂的 OLAP查询, 多维视图，汇总
- 不同的功能和不同的数据:
  - **历史数据:** 决策支持需要历史数据，而这些数据在操作数据库中一般不会去维护
  - **数据汇总：** 决策支持需要将来自异种源的数据统一（如聚集和汇总）
  - **数据质量:** 不同的源使用不一致的数据表示、编码和格式，对这些数据进行有效的分析需要将他们转化后进行集成

数据仓库的定义

- 数据仓库的定义很多，但却很难有一种严格的定义
  - 它是一个提供决策支持功能的数据库，它与公司的操作数据库分开维护。
  - 为统一的历史数据分析提供坚实的平台，对信息处理提供支持
- 数据仓库区别于其他数据存储系统
  - “数据仓库是一个面向主题的、集成的、随时间而变化的、不容易丢失的数据集合，支持管理部门的决策过程.”—W. H. Inmon（数据仓库之父）

数据仓库关键特征一——面向主题

- 面向主题，是DW显著区别于面向应用的传统DB的一个特征
- 概念：从数据组织的角度看，主题就是一些数据集合，它对分析对象进行了比较完整的、一致的数据描述，不仅包括数据自身还包括数据之[illegible]



[illegible]据建模[illegible]于组织[illegible]事务[illegible]

  - [illegible]原则：保证每个[illegible]性。
  - 围绕一些主题，例如哪些顾客采购产品数量多？哪些产品销售量大？哪些供应商提供的产品具有竞争力？
  - 主题之间可能存在重叠关系，如

数据仓库关键特征二——数据集成

- 一个数据仓库是通过集成多个异种数据源来构造的。
  - 关系数据库、一般文件、联机事务处理记录
- 使用数据清理和数据集成技术。
  - 确保命名约定、编码结构、属性度量等的一致性，度量单位。
  - 当数据被移到数据仓库时，它们要经过转化。

数据仓库关键特征三——随时间而变化（1）

- 数据仓库是从历史的角度提供信息
  - 数据仓库的时间范围比操作数据库系统要长的多。
    - 操作数据库系统: 主要保存当前数据。
    - 数据仓库:从历史的角度提供信息（比如过去 5-10 年）
  - 数据仓库中的每一个关键结构都隐式或显式地包含时间元素，而操作数据库中的关键结构可能就不包括时间元素。

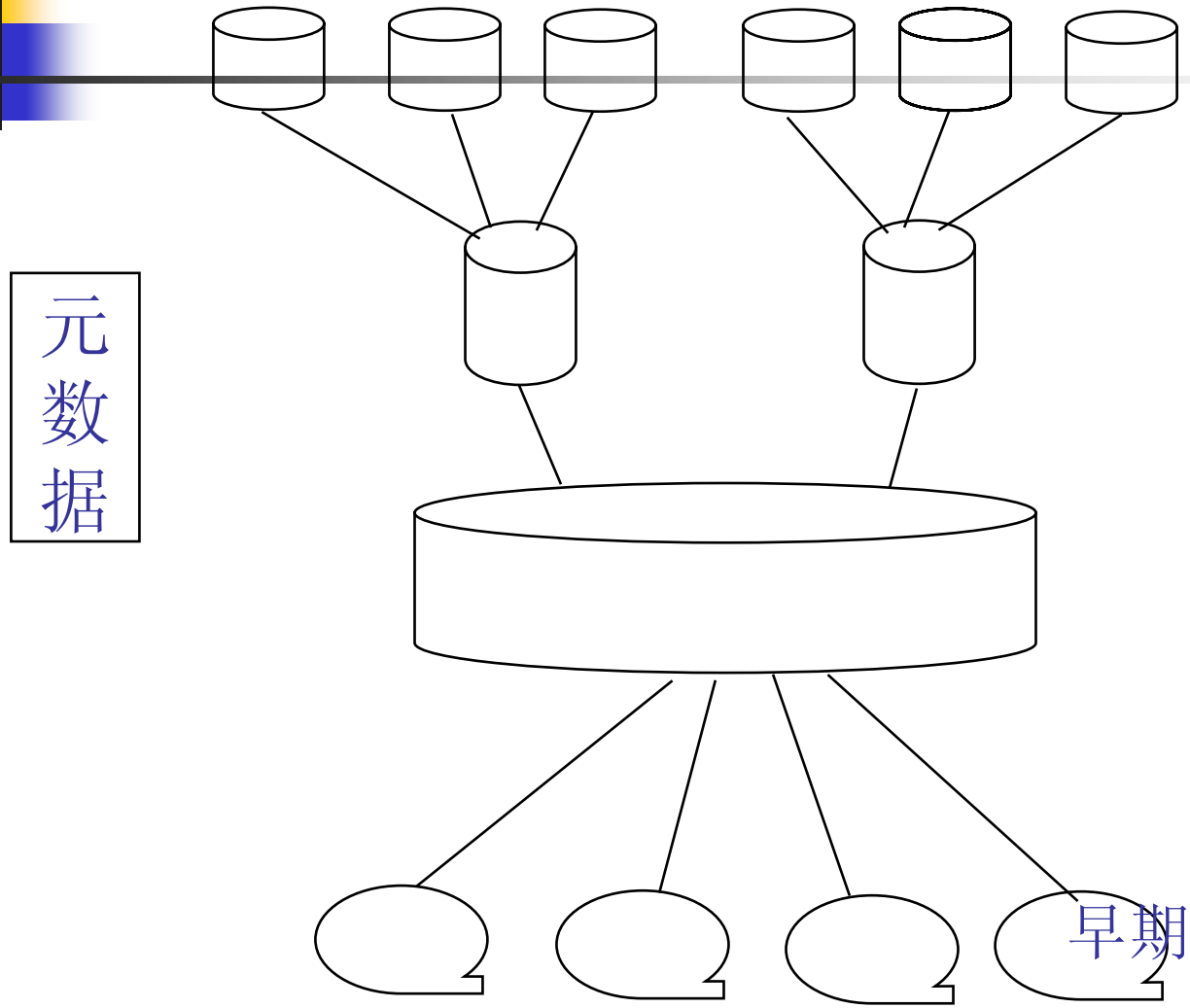
数据仓库关键特征三——随时间而变化（2）

- 数据仓库的数据追加
  - 定义：数据仓库的数据初装完成后，再向DW输入数据的过程。
  - 捕捉变化数据
    - 时标方法
    - DELTA文件：由应用生成，记录了应用改变的所有内容。优点：效率高，避免扫描整个DB。
    - 前后映像文件的方法：比较抽取数据的DB的前后快照。缺点：需占用大量资源。
    - *日志文件：DB的固有机制，不会额外增加工作量和占用系统资源。

数据仓库关键特征四——数据不易丢失

- 尽管数据仓库中的数据来自于操作数据库，但他们却是在物理上分离保存的。
  - 操作数据库的更新操作不会出现在数据仓库环境下。
  - *不需要事务处理，恢复，和并发控制等机制*
  - 只需要两种数据访问：
    - 数据的初始转载和数据访问（读操作）

数据仓库中的数据分级



元数据

- 概念：元数据是关于数据的数据，对DW中的各种数据进行详细的描述与说明，说明每个数据的上下文关系。

（相当于传统数据库系统中的数据字典）

- 元数据在DW中的作用
  - 用作目录，帮助DSS分析者对数据仓库的内容定义
  - 作为数据仓库和操作性数据库之间进行数据转换时的映射标准
  - 用于指导当前细节数据和稍加综合的数据之间的汇总算法，指导稍加综合的数据和高度综合的数据之间的汇总算法。

元数据的形式有以下几种：

- 数据仓库结构的描述
  - 仓库模式、视图、维、层次结构、导出数据的定义，以及数据集市的位置和内容
- 汇总用的算法
- 由操作环境到数据仓库的映射
- 关于系统性能的数据
  - 索引，profiles，数据刷新、更新或复制事件的调度和定时
- 商务元数据
  - 商务术语和定义、数据拥有者信息、收费政策等

元数据的分类

- 按类型分：
  - 基本数据（数据源、DW、应用程序管理）的元数据
  - 数据处理（数据装载、更新处理、分析处理、数据抽取、转换等）的元数据
  - 企业组织机构（用户、用户权限）的元数据
- 按抽象级别分：
  - 概念级（业务的全部描述）
  - 逻辑级（DB的关系方案，逻辑多维模型等）
  - 物理级
- 按承担的任务分：静态元数据（数据格式）和动态元数据（数据的状态与使用方法）
- 从用户角度分：技术元数据（开发、维护和管理信息技术环境中产生的数据）和业务元数据（使企业环境的服务更易于为终端用户所理解）

元数据的内容

- 数据源的元数据
  - 数据源的所有者描述信息、业务描述、存取方法、口令等。
- 数据模型的元数据
  - 企业概念模型，DW数据模型
- 数据准备区的元数据
  - 数据清洗规范、数据增强和映射转换、数据传输的安全性设置等
- DBMS元数据
  - 分区设置、索引、视图定义、数据备份等。
- 前台元数据
  - 现有的查询和报告定义、网络安全用户特权概况、身份验证、打印工具规范、最终用户文档等。

粒度与分割 （1）

- 粒度：DW中的数据单位中保存数据的细化或综合程度的级别。

  粒度越大，细化越低，综合程度越高。

  ➢ 分类：

  （1）按时间段综合数据的粒度：

  影响DW中的数据量的多少，也影响DW所能回答询问的种类

  （2）样本数据库：采样频率高低。

- 分割：将数据分散到各自的物理单元中以便能分别独立处理，以提高数据处理效率。

粒度与分割 （2）

- 分割的优点
  - 容易重构, 容易重组,自由索引,顺序扫描,易恢复, 易监控
- 分割的标准
  - 时间（必需）
  - 商业领域
  - 地理位置（区域）
  - 组织单位（机构）
  - 所有上述综合

数据仓库的数据组织及存储

数据仓库的数据组织形式：

（1）简单堆积文件：以天为单位堆积

（2）轮转综合文件：日、周、月、年

（3）简单直接文件：间隔一定的时间间隔

（4）连续文件：直接前后连接

数据仓库的存储方式：

（1）虚拟存储：没有专门数据仓库数据存储

（2）关系表存储：关系型数据库

（3）多维数据库存储：多维数组结构文件进行数据存储

数据仓库的说明—标准手册

- 与管理人员、开发人员、决策分析人员及计划人员等相关。
- 基本内容（12项）
  - 描述什么是DW
  - 描述对DW输送数据的源系统
  - 如何使用DW
  - 如何获得帮助
  - 谁负责什么
  - DW的迁入计划
  - DW的数据如何面向应用的数据相关联
  - 如何为决策分析系统使用DW
  - 什么时候不向DW中加数据
  - DW中没有什么类型的数据
  - 可利用的元数据的说明
  - DW的记录系统是什么

数据仓库的构建与使用

- 数据仓库的构建包括一系列的数据预处理过程
  - 数据清理：检测数据中的错误并作可能的订正
  - 数据集成：从多个外部的异构数据源收集数据
  - 数据变换：将数据由历史或主机的格式转化为数据仓库的格式
- 数据仓库的使用热点是商业决策行为，例如：
  - 增加客户聚焦
  - 产品重定位
  - 寻找获利点
  - 客户关系管理

数据仓库与异种数据库集成

- 异种数据库的集成方法
  - 传统的异种数据库集成:（*查询驱动*）
    - 在多个异种数据库上建立包装程序（wrappers）和中介程序（mediators ）
    - 查询驱动方法——当从客户端传过来一个查询时，首先使用元数据字典将查询转换成相应异种数据库上的查询；然后，将这些查询映射和发送到局部查询处理器
  - 数据仓库:（*更新驱动*）
    - 将来自多个异种源的信息预先集成，并存储在数据仓库中，供直接查询和分析

查询驱动方法和更新驱动方法的比较

- 查询驱动的方法
  - 需要负责信息的过滤和集成处理
  - 与局部数据源上的处理竞争资源
  - 对于频繁的查询，尤其是涉及聚集（汇总）操作的查询，开销很大（决策支持中常见的查询形式）
- 更新驱动的方法（带来高性能）
  - 数据经预处理后单独存储，对聚集操作提供良好支持
  - 不影响局部数据源上的处理
  - 集成历史信息，支持多维查询

数据仓库设计的三级数据模型

- 与操作型DB的不同之处：
  - DW的数据模型中不包含纯操作型的数据
  - DW中扩充了码结构，增加了时间属性作为码的一部分
  - 增加了一些导出数据
- 三级模型结构：
  - **概念级：** 描述主题，主题间的关系
  - **逻辑数据模型：**描述DW的主题的逻辑实现，即每个主题所对应的关系表的关系模式的定义
  - **物理数据模型：**

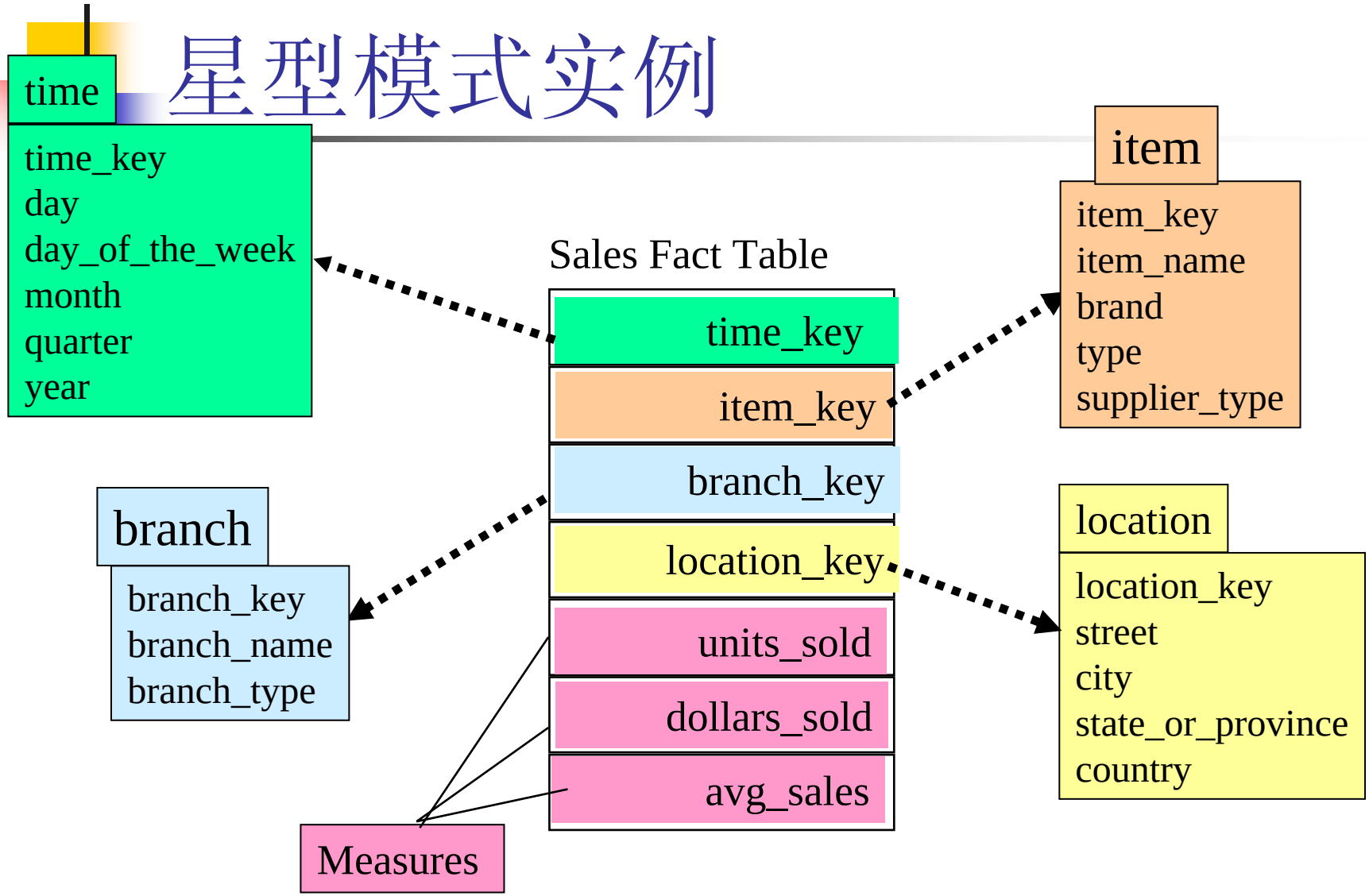
  性能因素：I/O存取空间时间利用率和维护代价
- Inmon的三级结构：高层数据模型、中间层数据模型和低层数据模型。

数据仓库的概念模型

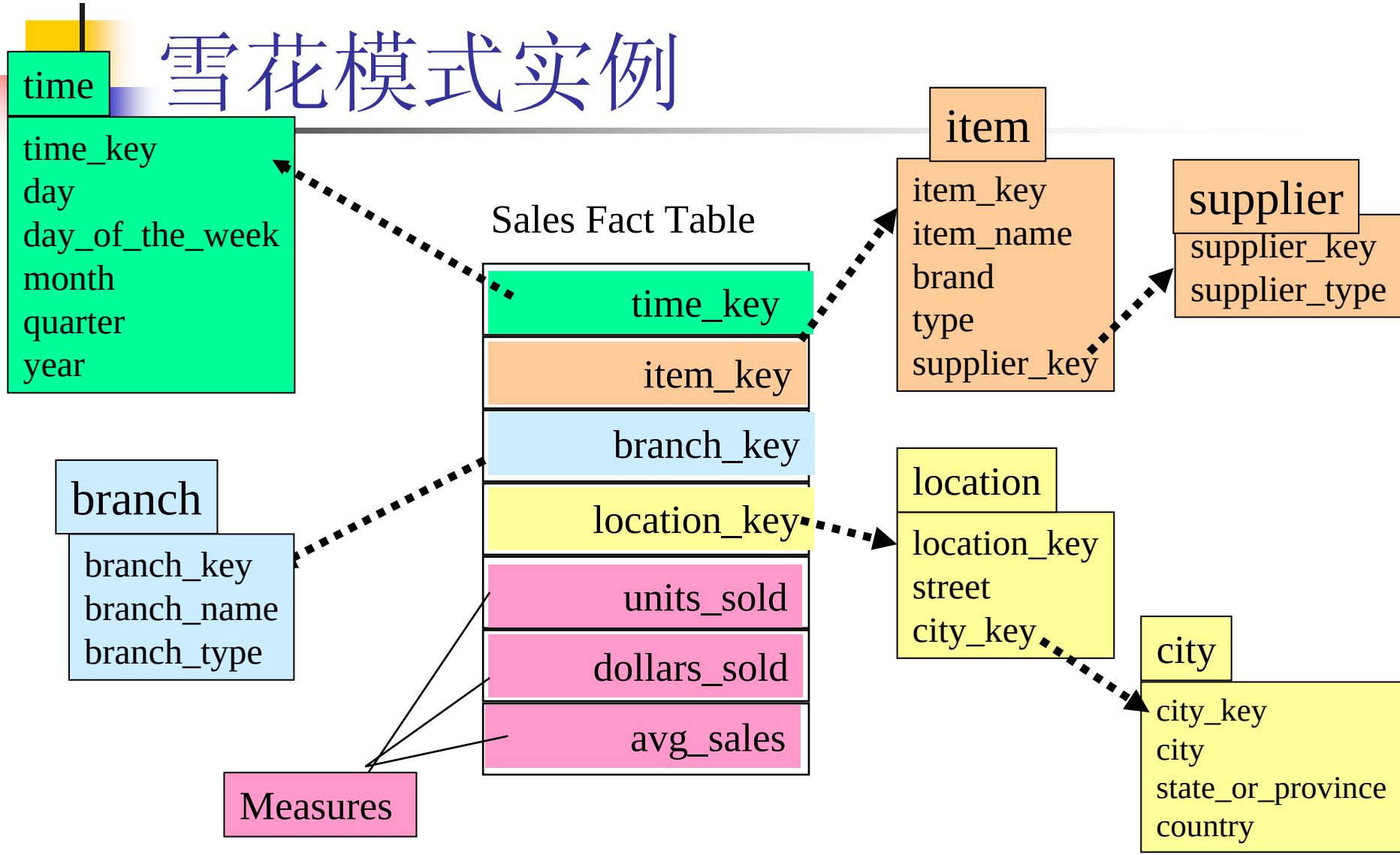
最流行的数据仓库概念模型是**多维数据模型**。这种模型可以以星型模式、雪花模式、或事实星座模式的形式存在。

- 星型模式（Star schema）：事实表在中心，周围围绕地连接着维表（每维一个），事实表含有大量数据，没有冗余。
- 雪花模式（Snowflake schema）： 是星型模式的变种，其中某些维表不是规范化的，因而把数据进一步分解到附加表中。结果，模式图形成类似于雪花的形状。
- 事实星座（Fact constellations）：多个事实表共享维表，这种模式可以看作星型模式集，因此称为星系模式（galaxy schema），或者事实星座。

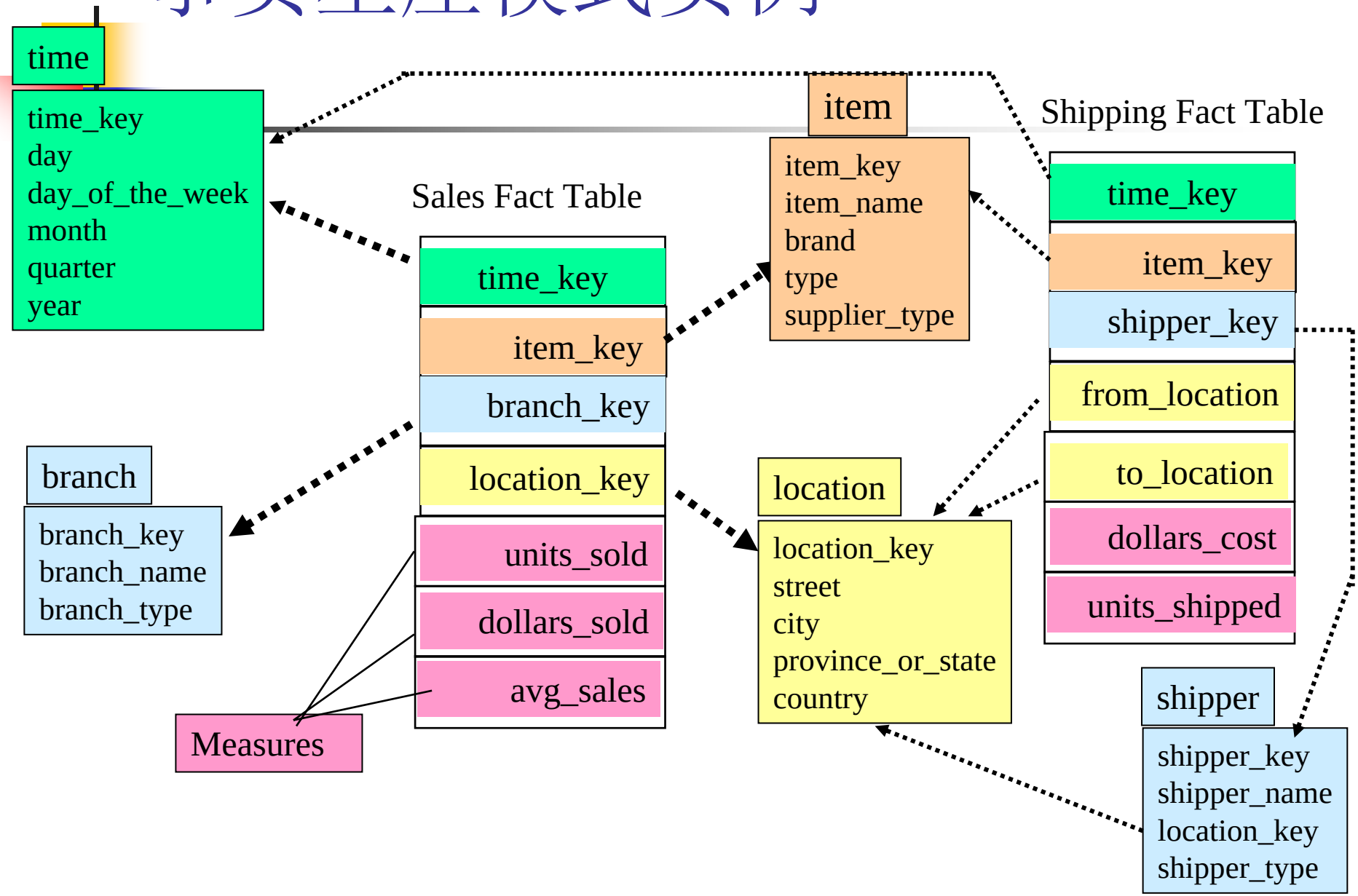
星型模式实例


time
time_key
day
day_of_the_week
month
quarter
year
item
item_key
item_name
brand
type
supplier_type
Sales Fact Table
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales
branch
branch_key
branch_name
branch_type
location
location_key
street
city
state_or_province
country
Measures

雪花模式实例


time
time_key
day
day_of_the_week
month
quarter
year
Sales Fact Table
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales
item
item_key
item_name
brand
type
supplier_key
supplier
supplier_key
supplier_type
branch
branch_key
branch_name
branch_type
location
location_key
street
city_key
city
city_key
city
state_or_province
country
Measures

事实星座模式实例



数据仓库设计：一个商务分析框架 (1)

- 数据仓库给商业分析专家提供了什么？
  - 通过提供相关数据与信息，获得竞争优势
  - 通过有效的收集精确的描述组织的数据，获得生产力的提高
  - 通过提供不同级别（部门、市场、商业）的客户视图，协助客户关系管理
  - 通过追踪长期趋势、异常等，降低成本
- 有效构建数据仓库的关键：理解和分析商业需求
  - *通过提供一个商业分析框架，综合各种不同的数据使用者的视图*

数据仓库设计：一个商务分析框架 (2)

- 数据仓库设计的四种视图
  - 自顶向下视图
    - 允许我们选择数据仓库所需的相关信息
  - 数据源视图
    - 揭示被操作数据库系统所捕获、存储和管理的信息
  - 数据仓库视图
    - 由事实表和维表所组成
  - 商务查询视图
    - 从最终用户的角度透视数据仓库中的数据

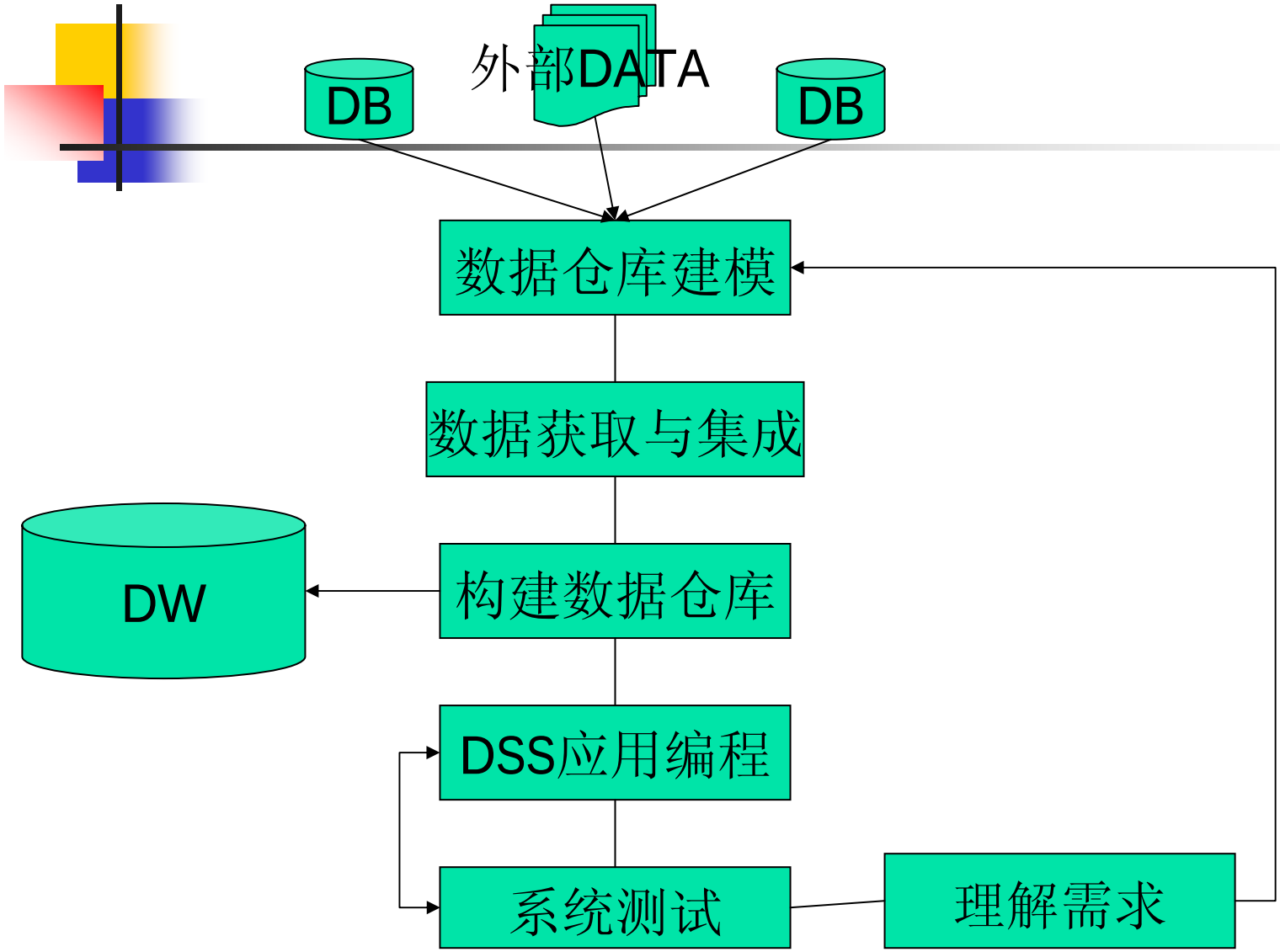
数据仓库设计：一个商务分析框架(3)

- 数据仓库的构建与使用涉及多种技能
  - 商业技能
    - 理解系统如何存储和管理数据
    - 数据如何提取
    - 数据如何刷新
  - 技术方面的技能
    - 如何通过使用各种数据或量化的信息，来提供决策支持的模式、趋势、判断等
    - 如何通过审查历史数据，分析发展趋势等
  - 计划管理技能
    - 如何通过与不同的技术、厂商、用户交互，来及时、有效、经济的提交结果

数据仓库的设计过程 (1)

- 自顶向下法、自底向上法或者两者的混合方法
  - 自顶向下法：由总体设计和规划开始
    - 在技术成熟、商业理解透彻的情况下使用
  - 自底向上法：以实验和原型开始
    - 常用在模型和技术开发的初期，可以有效的对使用的技术和模型进行评估，降低风险
  - 混合方法：上述两者的结合
- 从软件过程的观点
  - 瀑布式方法：在进行下一步前，每一步都进行结构化和系统的分析
  - 螺旋式方法：功能渐增的系统的快速产生，相继版本之间间隔很短

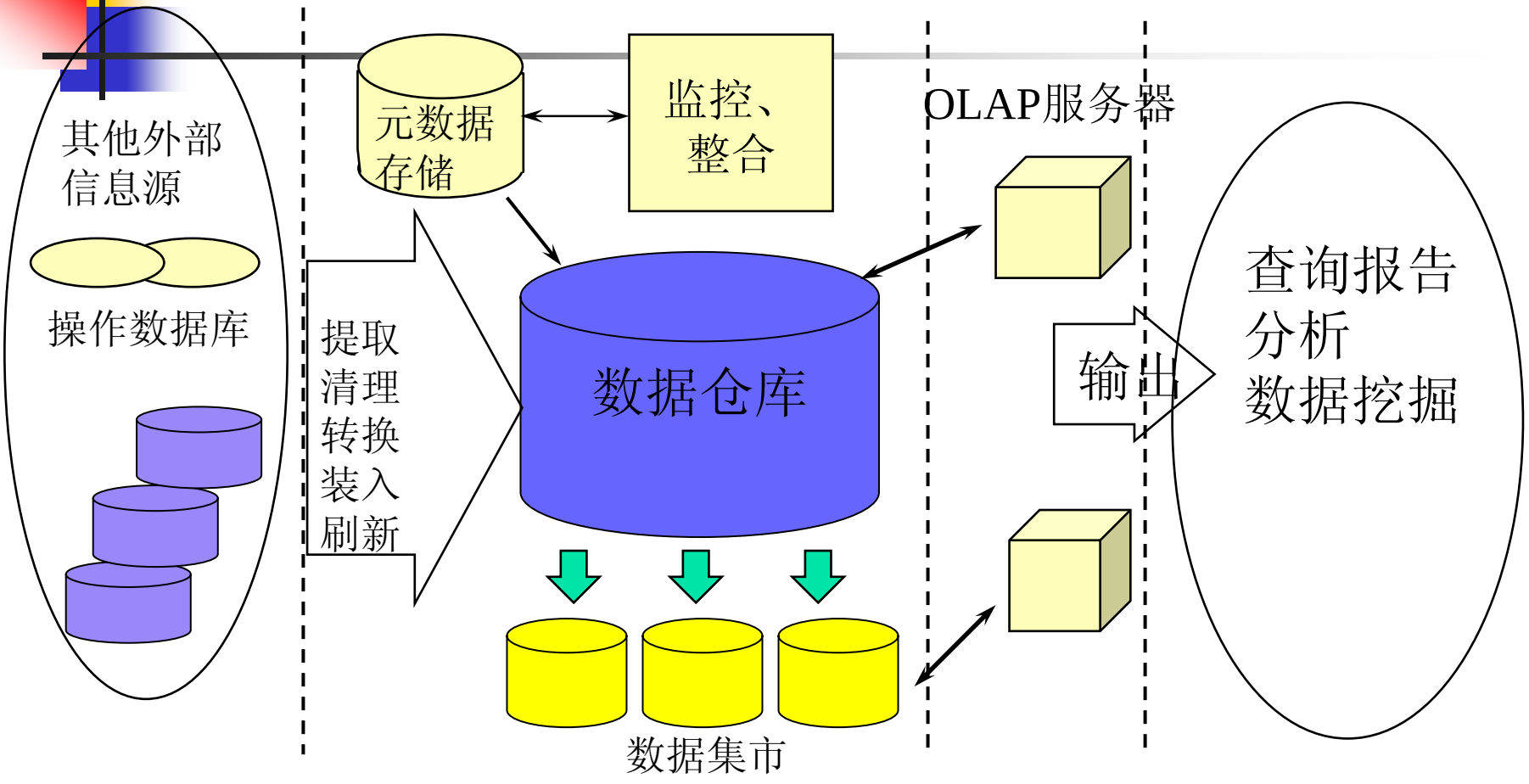
数据仓库的螺旋式开发方法



数据仓库的设计过程 (2)

- 典型的数据仓库设计过程
  - 选取待建模的*商务过程*
    - 找到所构建的数据仓库的主题，比如：销售、货运、订单等等
  - 选取商务过程的*颗粒度*
    - 数据起始于多细的颗粒度，比如：记录每条详细订单，或是开始于每日的汇总数据
  - 选取用于每个事实表记录的*维*
    - 常用的维有：时间、货物、客户、供应商等
  - 选取将安放在事实表中的*度量*
    - 常用的数字度量包括：售价、货物数量等

三层数据仓库架构 (1)



三层数据仓库架构 (2)

- 底层：数据仓库的数据库服务器
  - 关注的问题：如何从这一层提取数据来构建数据仓库（通过Gateway（ODBC,JDBC,OLE/DB等）来提取）
- 中间层：OLAP服务器
  - 关注的问题：OLAP服务器如何实施（关系型OLAP，多维OLAP等）
- 前端客户工具层
  - 关注的问题：查询工具、报表工具、分析工具、挖掘工具等

三种数据仓库模型

从体系结构的角度去看，数据仓库模型可以有以下三种：

- 企业仓库
  - 搜集关于跨越整个组织的主题的所有信息
- 数据集市
  - 企业范围数据的一个子集，对于特定的客户是有用的。其范围限于选定的主题，比如一个商场的数据集市
    - 独立的数据集市 VS. 从属数据集市（数据来自于企业数据仓库）
- 虚拟仓库
  - 操作数据库上的一系列视图
  - 只有一些可能的汇总视图被物化

数据仓库开发：困难与方法

- 数据仓库开发上的困难
  - 自顶向下的开发方法从全系统的角度提供解决方案，使得（模块）集成的问题最小；但是该方法十分昂贵，需要对组织进行长期研究和建模分析。
  - 自底向上方法提供了更多的开发灵活性，价格便宜；但往往会遇到集成问题（每个模块单独运行都没有问题，但是一集成就出异常）
- 解决方法：
  - 使用递增性、演化性的开发方法：

    高层数据模型→企业仓库和数据集市并行开发→通过分布式模型集成各数据集市→多层数据仓库

数据仓库开发——一个推荐的方法：

先小后大，先低后高