第四章 数据仓库与联机分析处理

张静
（华东理工大学计算机系）
（Jingzhang@ecust.edu.cn）

主要内容

- 数据仓库的概念
- 数据仓库的建模：数据立方体与OLAP
- 数据仓库的设计与使用
- 数据仓库实现
- 数据泛化：面向属性的归纳
- 小结

从事务数据到决策知识

- DBMS的发展轨迹清晰的表明，它是在服务于联机事务处理（Online Transaction Processing，OLTP）中不断完善和丰富起来的。
- 特别是基于E.F.Codd提出的关系理论的数据库技术，将数据集分成了甚少冗余的实体（Entity），并利用关系（Relationship）将这些实体组织成一个有机的整体，较好地满足了OLTP的应用需求。
- 其主要目的在于操作数据，而不在于分析数据，因此它提供了强大的数据存取、增添、删除、修改等操作的功能体系，却不能为预测和决策服务提供多少有用的信息。

基于事务数据库的决策支持系统DSS（Decision Support System）的缺陷

- 数据缺乏组织性
  - DSS需要集成的数据，全面而正确的数据是有效的分析和决策的首要前提，相关数据收集得越完整，得到的结果就越可靠。

基于事务数据库的决策支持系统DSS（Decision Support System）的缺陷

- 业务数据本身大多以原始的形式存储，难以转换为有用的信息
  - 事务处理的目的在于使业务处理自动化和简单化，因此数据的表达上尽可能简化以利于存储，即使是描述同一属性，在不同的库中也可能有不同的表达方式。
    - 例如，考虑对某人“性别”的编码，在数据库A中编码为‘m’，而在数据库B编码为‘男’，则DSS分析时该如何使用这些数据？

基于事务数据库的决策支持系统DSS（Decision Support System）的缺陷

- 服务于联机事务处理(OLTP)的关系数据库是面向操作的而不是面向分析的
  - OLTP要求数据库系统具有实时响应能力，数据库系统也是一个时变的系统。
  - 一个DSS分析与推理可能需要数秒或者数小时，甚至更长的时间，如果其基于的数据在不停的变化，会导致决策分析的求解过程永远无法收敛。
  - 本质上传统数据库是很难为数据分析提供有意义的数据，这二者本身就是一对矛盾体。

DSS所期望的理想数据源

- 数据格式统一
  - 该数据库中同一问题的属性字段都采用同一种表达方式来描述。具有一致的命名规则，一致的变量单位，一致的编码结构和一致的特性描述等。
- 独立
  - 该数据库与事务数据库隔离开来，割断这两者间的相互牵制。事务数据库必然要求能响应且实时响应对它的读写事务操作，而DSS分析过程是对现有数据的一个推理演算，它不需要修改数据库中的数据，否则会影响其它DSS的分析过程，因此它对于DSS而言是一个只读型的数据库。

DSS所期望的理想数据源

- 集成了某一主题所需的全部数据
  - 是指用户使用数据库辅助决策时所关心的重点问题，每一个主题对应一个客观分析领域。
- 在DSS分析期间相对稳定
  - 是指数据一旦进入数据库，一般情况下将被长期保留，变更很少。
- 保持与具体应用同步的“最新”数据
  - 是指数据库中存储的是一个时间段的数据，而不仅仅是某一个时间点的数据。当数据源的信息变更后，DSS期望的数据库应该也能反映这种变更，以便基于正确的数据进行分析。

数据仓库的概念

- DSS所期望的数据库正是数据仓库（Data Warehouse，DW）。它正是为了建立这种新的分析处理环境而出现的一种数据存储和组织技术。
- 数据仓库
  - 涉及数据清理和数据集成，可以看作是数据挖掘的一个重要预处理步骤。
  - 提供联机分析处理（OLAP）工具，用于各种粒度的多维数据分析。

数据仓库的概念

- 数据仓库
  - 数据仓库是一个面向主题的(Subject Oriented)、集成的(Integrated)、时变的(Time-Variant) 、非易失(Nonvolatile) 数据集合，用于支持管理决策。

数据仓库的特征

- 面向主题的
  - 基于传统关系数据库建立的各个应用系统，是面向应用进行数据组织的；而数据仓库中的数据是面向主题进行组织的。
  - 主题是指一个分析领域，是指在较高层次上企业信息系统中的数据综合、归类并进行利用的抽象。
    - 例如保险公司建立数据仓库，所选主题可能是顾客、保险金和索赔等，而按照应用组织的数据库则可能是汽车保险、生命保险和财产保险等。
  - 面向主题的数据组织方式，就是在较高层次上对分析对象的数据一个完整、一致的描述，能完整、统一地刻划各个分析对象所涉及的各项数据以及数据之间的联系。

数据仓库的特征

- 集成的
  - 通过集成多个异种数据源而构成。
    - 关系数据库、一般文件和联机事务处理记录。
  - 使用数据清理和数据集成技术。
    - 在不同的数据源中，确保命名约定、编码结构、属性度量等的一致性。
      - 例如，旅馆价格：由住宿费、税收、附带的早餐费等等构成。
  - 数据被移到数据仓库时就进行了数据转换。

数据仓库的特征

- 时变的
  - 数据仓库的时间范围明显长于操作数据库
    - 操作数据库：当前的有用信息。
    - 数据仓库：从历史的角度提供信息（例如：过去的5-10年）
  - 数据仓库的每一个关键结构都隐式或显示的包含时间元素
    - 但操作数据库的关键结构可以包含也可以不包含“时间元素”

数据仓库的特征

- 非易失的
  - 数据仓库总是物理地分离存放数据，这些数据源于操作环境下的应用数据
  - 操作性的数据更新不会发生在数据仓库的环境下
    - 数据仓库不需要事务处理、恢复和并发控制机制
    - 它只需要两种数据访问：
      - 数据的初始装入和数据访问

数据仓库系统

数据仓库系统

- 是以数据仓库技术为基础，以联机分析处理（OLAP）和数据挖掘（Data Mining）等工具为手段进行数据分析处理的一整套解决方案。
- 或者说数据仓库系统以数据仓库为基础，通过查询工具和分析工具，完成对信息的提取，满足用户进行管理和决策的各种需要的系统。

操作数据库与数据仓库的区别

- 操作数据库
  - 联机事务处理OLTP (on-line transaction processing)
- 数据仓库
  - 联机分析处理OLAP (on-line analytical processing)
- OLTP和OLAP是两类主要的数据处理方法

OLTP vs. OLAP

- 联机事务处理OLTP (on-line transaction processing)
  - 传统的关系DBMS的主要任务
  - 以日常事务处理为主，是一种操作型处理
  - 特点是处理事务量大，但事务内容比较简单且重复率高，人们主要关心的是响应时间、数据安全性和完整性。

OLTP vs. OLAP

- 联机分析处理**OLAP (on-line analytical processing)**
  - 数据仓库系统的主要任务
  - 以数据分析和决策为目标
  - 需要访问大量历史性、汇总性和计算性数据，分析内容复杂，主要是管理人员的决策分析。
  - 具有汇总、合并和聚集功能，以及从不同角度观察信息的能力，支持多维分析和决策。

OLTP vs. OLAP

- 用户和系统的面向性
  - OLTP是面向顾客的，用于办事员、客户和信息技术专业人员的事务和查询处理；OLAP是面向市场的，用于帮助经理、主管和分析人员等进行数据分析。
- 数据内容
  - OLTP系统管理当前数据。这种数据一般都太琐碎，难以用于决策。OLAP系统管理大量历史数据，提供汇总和聚集机制，并在不同的粒度级别存储和管理信息。
- 数据库设计
  - OLTP系统通常采用实体-联系(ER)模型和面向应用的数据模式，而OLAP系统通常采用星形或雪花模型和面向主题的数据模式。

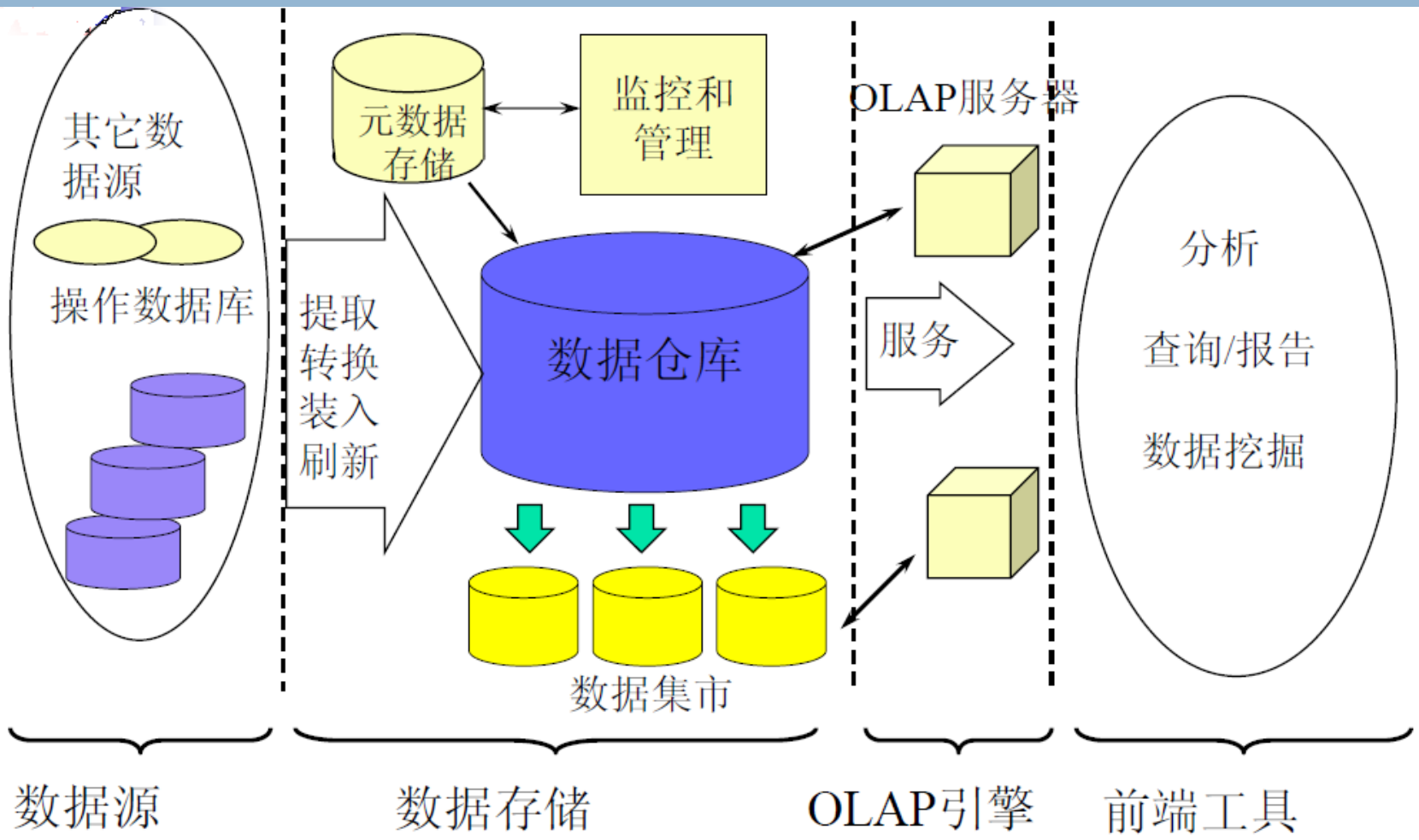
OLTP vs. OLAP

- 视图
  - OLTP系统主要关注一个企业或部门内部的当前数据，而不涉及历史数据或不同组织的数据； OLAP系统则通常跨越数据库模式的多个版本，处理来自不同组织的信息和多个数据存储集成的信息。此外，由于数据量巨大，OLAP数据一般存放在多个存储介质上。
- 访问模式
  - OLTP系统的访问主要由短的原子事务组成。而OLAP系统的访问由一些只读操作组成，尽管可能是很复杂的查询操作。

OLTP vs. OLAP

特　性	OLTP	OLAP
特征	操作处理	信息处理
面向	事务	分析
用户	办事员、DBA、数据库专业人员	知识工人(如经理、主管、分析员)
功能	日常操作	长期信息需求，决策支持
DB设计	基于E-R、面向应用	星形/雪花、面向主题
数据	当前的，确保最新	历史的，跨时间维护
汇总	原始的，高度详细	汇总的、统一的
视图	详细，一般关系	汇总的、多维的
工作单位	短的，简单事务	复杂查询
存取	读/写	大多为读
数据冗余	非冗余性	时常有冗余
操作	主关键字索引/散列	大量扫描
访问记录数量	数十个	数百万
用户数	数千	数百
DB规模	100MB到GB	100GB到TB
优先	高性能，高可用性	查询吞吐量，响应时间
度量	事务吞吐量	查询吞吐量，响应时间

数据仓库结构（多层体系结构）



数据仓库的体系结构

- 数据源
  - 是数据仓库系统的基础，是整个系统的数据源泉。通常包括企业内部信息和外部信息。
  - 内部信息包括存放于企业操作型数据库中(通常存放在RDBMS中)的各种业务数据和办公自动化(OA)系统包含的各类文档数据。
  - 外部信息包括各类法律法规、市场信息、竞争对手的信息以及各类外部统计数据和各类文档等。

数据仓库的体系结构

- 数据抽取、转换和加载
  - 这个部分负责从外部数据源获取数据，数据被区分出来，进行拷贝或重新定义格式等处理后，准备装入数据仓库。
  - 数据抽取在技术上主要涉及互连、复制、增量、转换、调度和监控等方面。
  - 数据仓库中的数据并不要求与联机事务处理系统保持实时同步，因此数据抽取可以定时进行，但多个抽取操作执行的时间、相互的顺序、成败对数据仓库中信息的有效性则至关重要。

数据仓库的体系结构

- 数据的存储与管理
  - 是整个数据仓库系统的核心，它负责数据仓库的内部维护和管理。
  - 在现有各业务系统的基础上，对数据进行抽取、清理，并有效集成，按照主题进行重新组织，最终确定数据仓库的物理存储结构，同时组织存储数据仓库元数据(具体包括数据仓库的数据字典、记录系统定义、数据转换规则、数据加载频率以及业务规则等信息)。
  - 按照数据的覆盖范围，数据仓库存储可以分为企业级数据仓库和部门级数据仓库(通常称为“数据集市”(Data Mart)。
  - 数据仓库的管理包括数据的安全、归档、备份、维护和恢复等工作。这些功能与目前的DBMS基本一致。

数据仓库的体系结构

- OLAP服务器
  - 对分析需要的数据按照多维数据模型进行再次重组，以支持用户多角度、多层次的分析，发现数据趋势。
  - 其具体实现可以分为：ROLAP、MOLAP和HOLAP。
  - ROLAP基本数据和聚合数据均存放在RDBMS之中；MOLAP基本数据和聚合数据均存放于多维数据库中；而HOLAP是ROLAP与MOLAP的综合，基本数据存放于RDBMS中，聚合数据存放于多维数据库中。

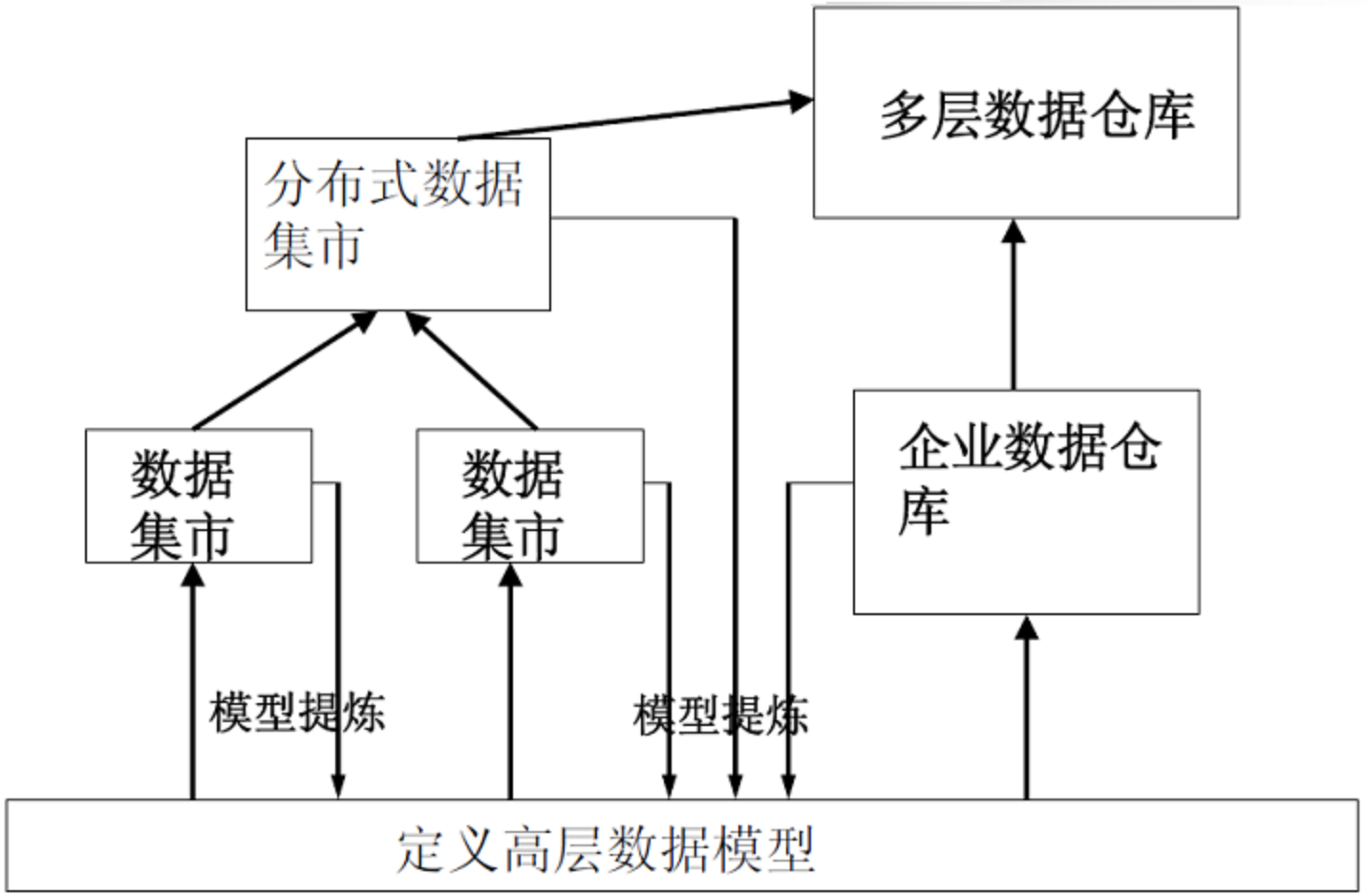
数据仓库的体系结构

- 前端工具与应用
  - 它面向最终用户，前端工具主要包括各种数据分析工具、报表工具、查询工具、数据挖掘工具以及各种基于数据仓库或数据集市开发的应用。
  - 其中数据分析工具主要针对OLAP服务器，报表工具、数据挖掘工具既针对数据仓库，同时也针对OLAP服务器。

数据仓库模型

- 企业仓库
  - 搜集了关于主题的所有信息，跨越整个组织
- 数据集市
  - 包含企业范围数据的一个子集，对于特定的用户是有用的，其范围限于选定的主题，如：商场的数据集市可能限定其主题为顾客、商品和销售。
    - 独立的数据集市和依赖的数据集市
      - 独立的数据集市，数据来自一个或多个操作系统或外部信息提供者，或者来自在一个特定的部分或地域局部产生的数据。
      - 依赖的数据集市，数据直接来自企业数据仓库。
- 虚拟仓库
  - 操作数据库上视图的集合。
  - 只有一些可能的汇总视图被物化。

数据仓库开发的推荐方法



元数据库

- 元数据是关于数据的数据。在数据仓库中，元数据是定义仓库对象的数据。
- 元数据库应当包括
  - 数据仓库结构的描述
    - 包括仓库模式、视图、维、层次结构和导出的数据定义，以及数据集市的位置和内容
  - 操作元数据
    - 包括数据血统（移植数据的历史和用于它的转换序列），数据流通（主动的、档案的或净化的），以及监视信息（仓库使用统计，错误报告，审计跟踪）
  - 汇总用的算法
    - 包括度量和维定义算法，数据所处粒度、分割、主题领域、聚集、汇总、预定义的查询与报告。

元数据存储

- 由操作环境到数据仓库的映射
  - 包括源数据库和他们的内容、网间连接程序描述、数据分割、数据提取、清理、转换规则和缺省、数据刷新和剪裁规则、安全（用户授权和存取控制）。
- 关于系统性能的数据
  - 刷新、更新和复制周期的定时和调度的规则
  - 改善数据存取和检索性能的索引和配置
- 商务元数据
  - 商务术语和定义，数据拥有者信息和收费策略

主要内容

- 数据仓库的概念
- 数据仓库的建模：数据立方体与OLAP
- 数据仓库的设计与使用
- 数据仓库实现
- 数据泛化：面向属性的归纳
- 小结

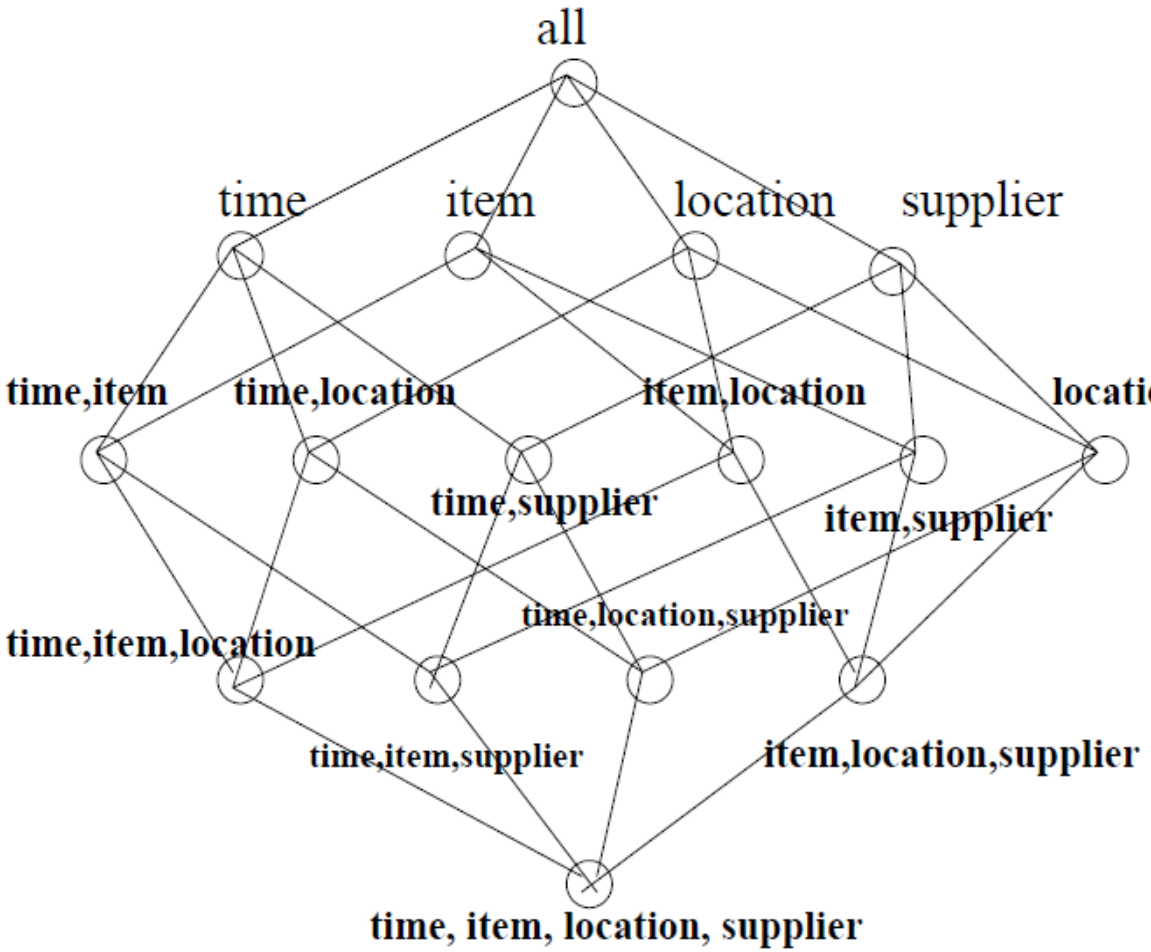
数据立方体

- 数据仓库建立在多维数据模型上，多维数据模型把数据看成数据立方体的形式
- 一个数据立方体，像sales,允许以多维对数据建模和观察
  - 维：维表，例如item的维表包含属性(item_name, brand，type)，time的维表包含属性(day，week, month, quarter, year)
  - 主题：事实表，包含事实的名称和度量(例如dollars_sold) 以及每个相关维表的关键字

数据立方体

- 数据立方体可以是n维的
  - 我们可以把任意n-D数据立方体显示成（n-1）-D数据立方体的序列
- 数据仓库语义中
  - 一个n-D 底层方体称为基本方体。 最高层的0-D方体，存放最高层的汇总， 称为顶点方体。所有的方体格组成了数据立方体。

立方体：一个方体格



0-D(顶点）方体

1-D 方体

2-D 方体

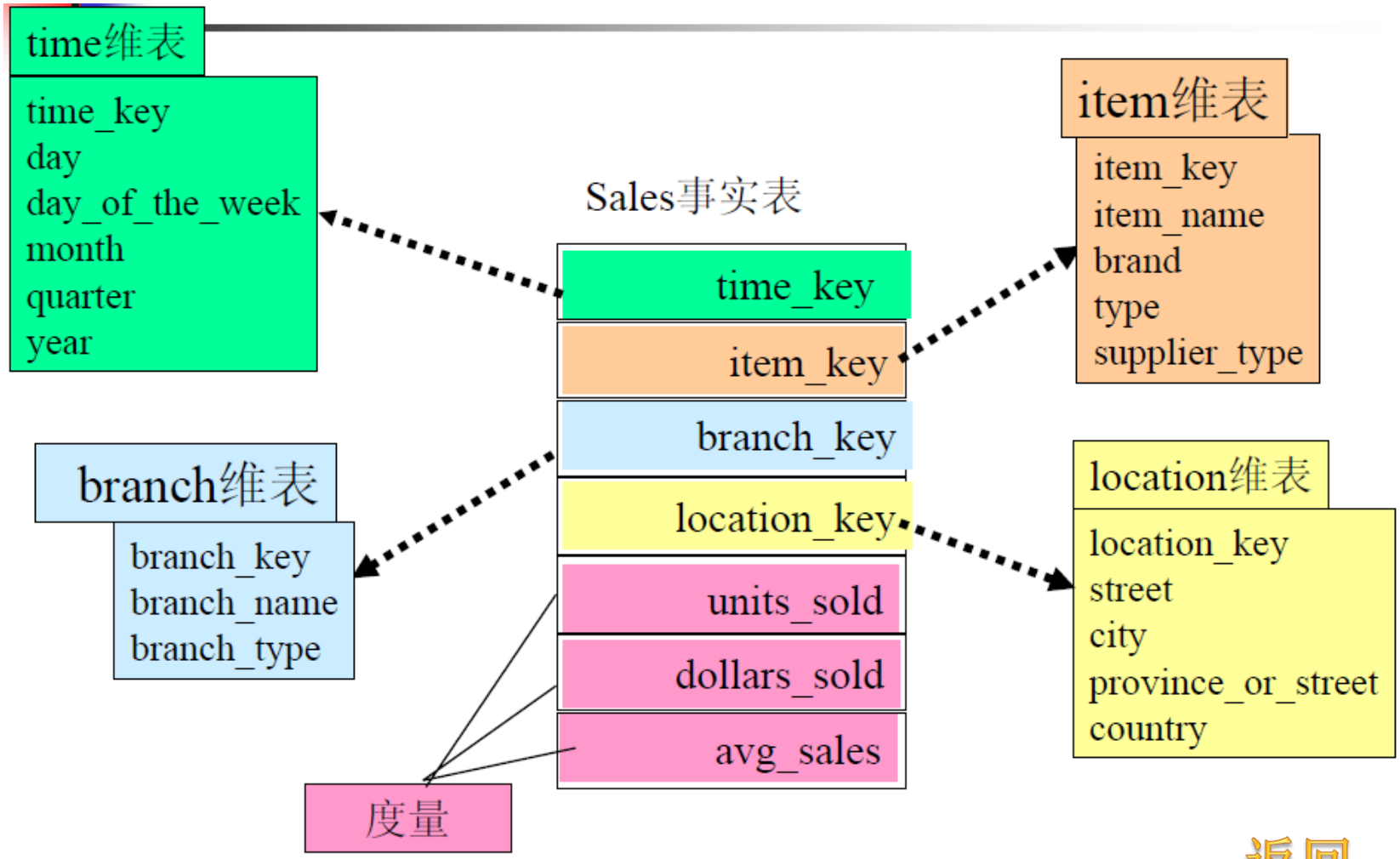
3-D 方体

4-D(基本) 方体

多维数据模型的模式

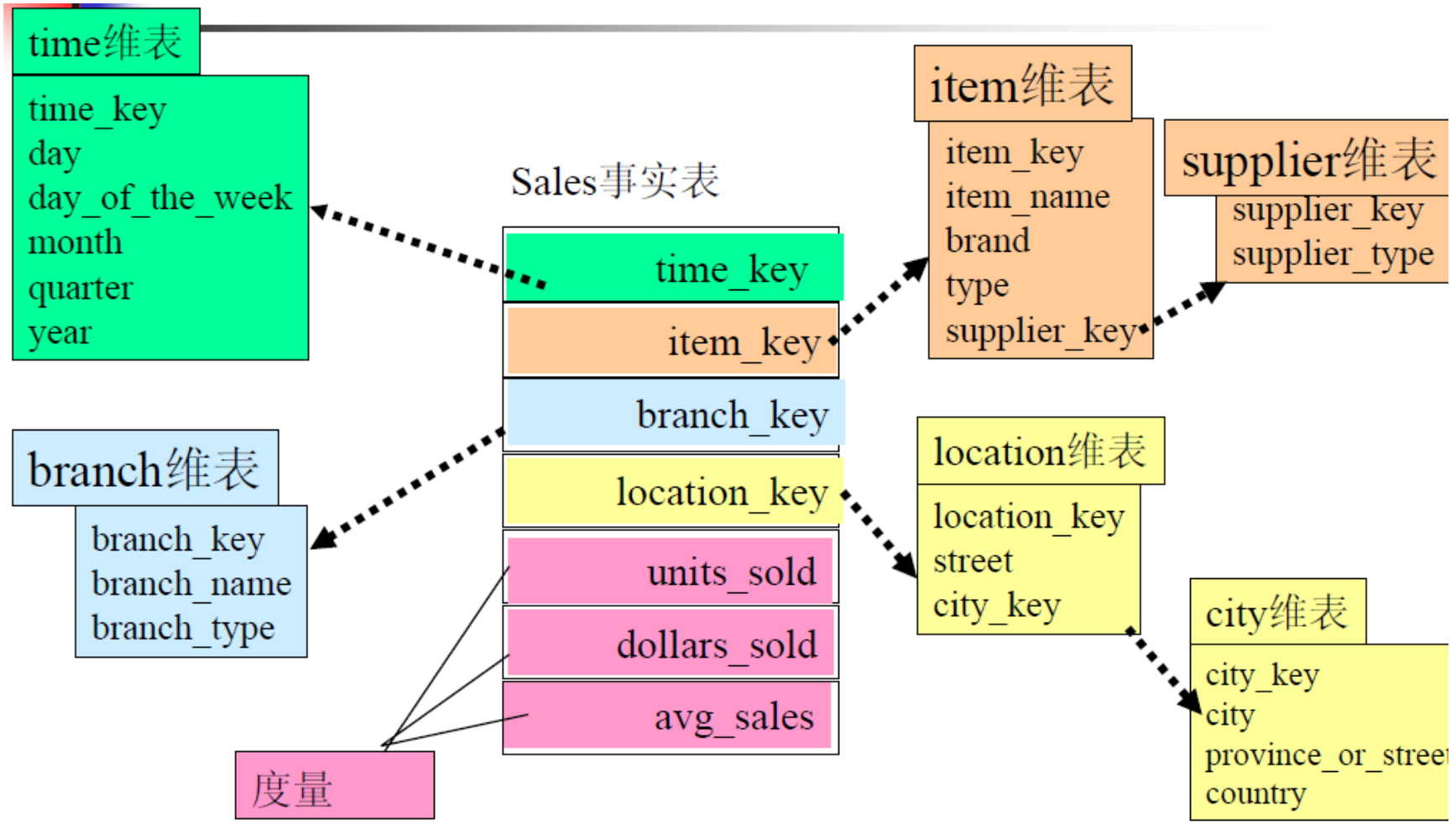
- 多维数据模型
  - 星形模式：中间是事实表（包含大批数据和不含冗余的中心表），连接一组维表，每维一个。
  - 雪花模式：雪花模式是星形模式的变种，其中某些维表是规范化的，因而把数据进一步分解到附加的维表中，它的图形类似于雪花的形状
    - 雪花模式与星形模式的区别在于：雪花模式的维表可能是规范化形式，以便减少冗余。但大量地链接操作会降低查询性能。
  - 事实星座模式：多个事实表共享维表，这种模式可以看作星形模式集，因此称为星系模式或事实星座

星形模式的例子


time维表
time_key
day
day_of_the_week
month
quarter
year
Sales事实表
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales
item维表
item_key
item_name
brand
type
supplier_type
branch维表
branch_key
branch_name
branch_type
location维表
location_key
street
city
province_or_street
country
度量

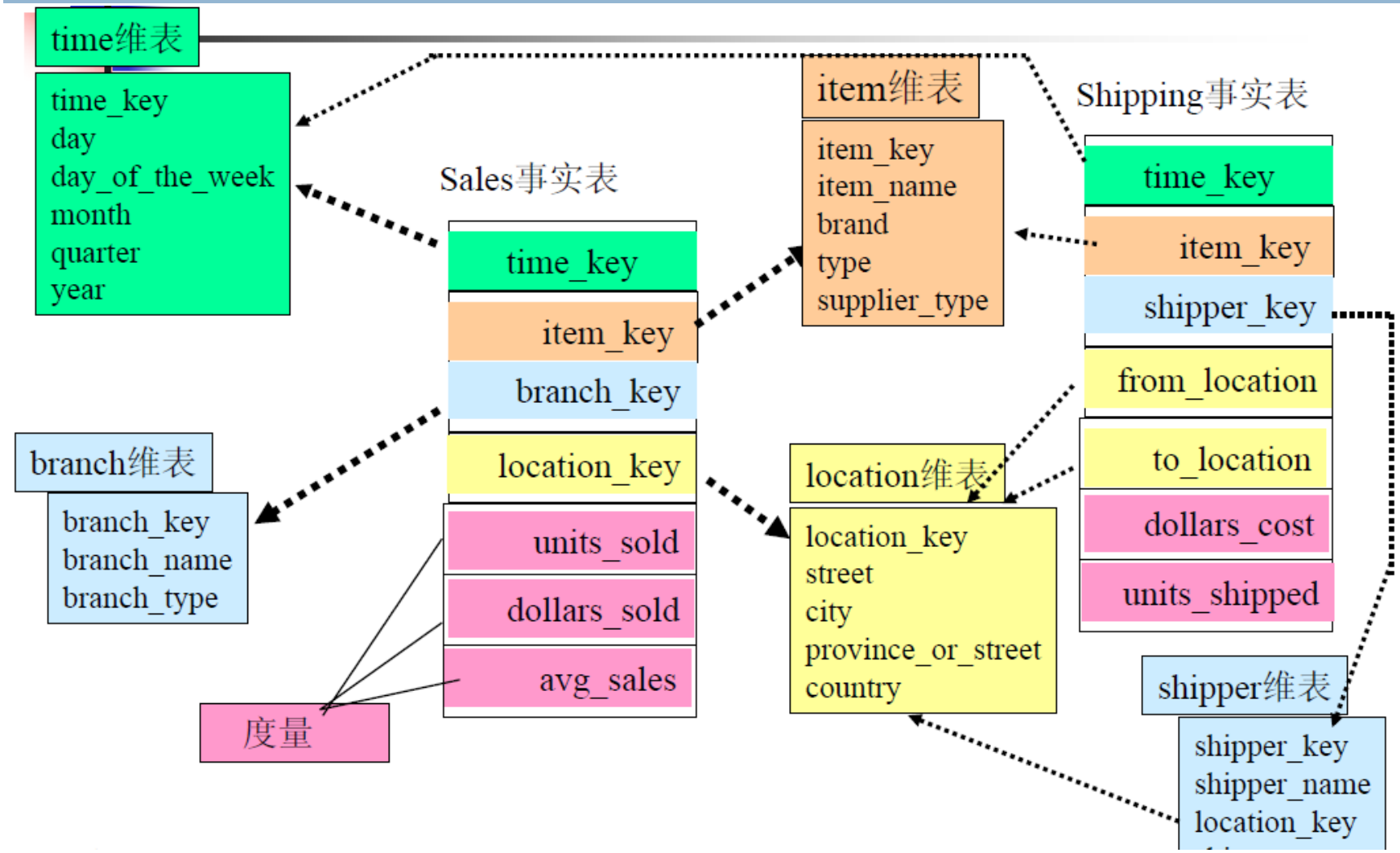
返回

雪花模式的例子


time维表
time_key
day
day_of_the_week
month
quarter
year
Sales事实表
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales
item维表
item_key
item_name
brand
type
supplier_key
supplier维表
supplier_key
supplier_type
branch维表
branch_key
branch_name
branch_type
location维表
location_key
street
city_key
city维表
city_key
city
province_or_street
country
度量

返回

事实星座模式的例子



返回

数据仓库VS数据集市

- 数据仓库
  - 收集关于整个组织的主题（如顾客、商品、销售、资产和人员）信息，是企业范围的。
  - 通常使用事实星座模式，因为它能对多个相关主题建模。
- 数据集市
  - 数据仓库的一个部门子集，针对选定的主题，是部门范围的。
  - 通常使用星形或雪花模式，只对单个主题建模。

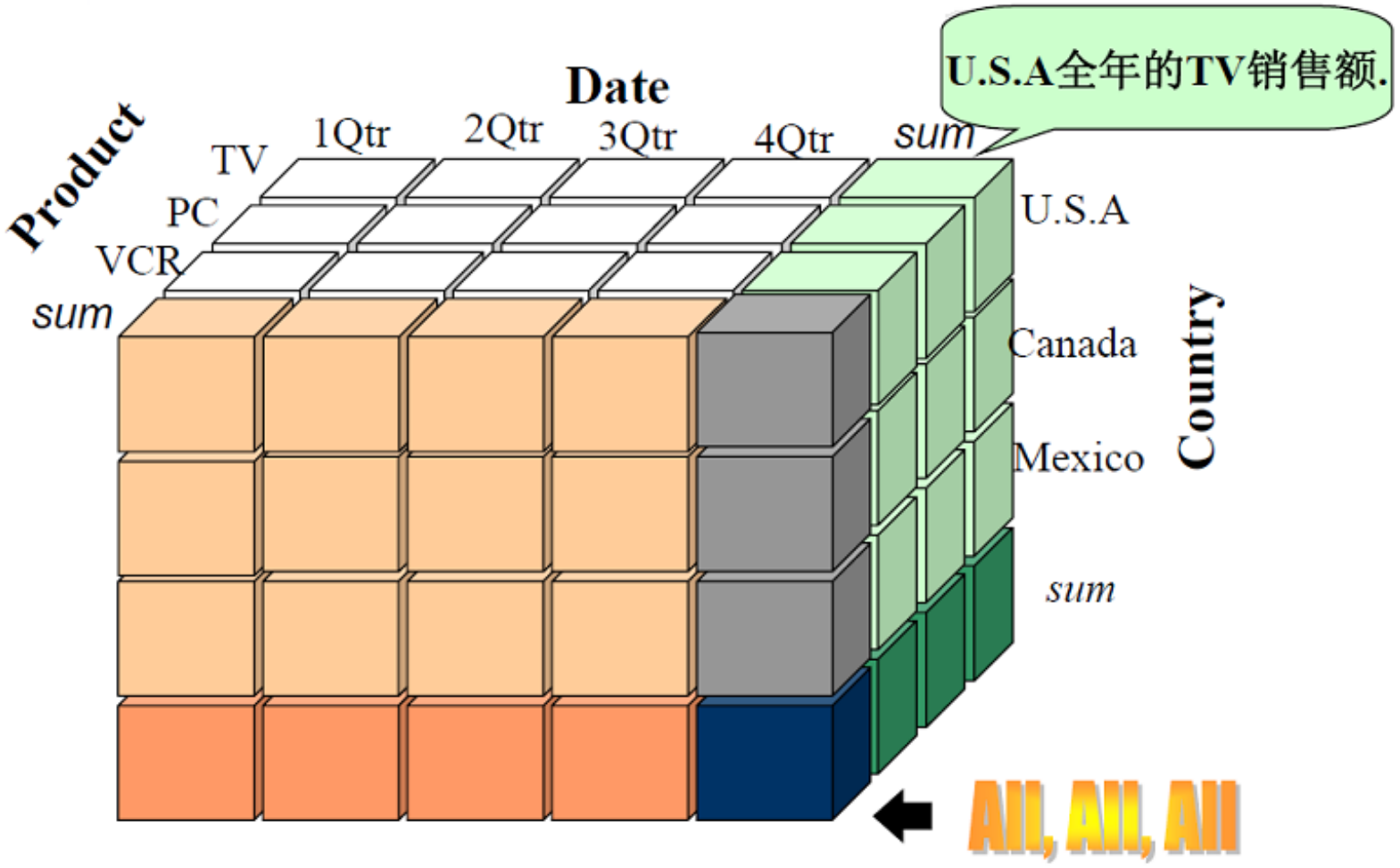
度量的分类和计算

- 数据立方体空间的多维点由维-值对定义
  - 例如：**<time="Q1", location="Vancouver", item="computer">**
- 数据立方体度量
  - 是一个数值函数，该函数可以对数据立方体的每一个点求值。
  - 通过对给定点的各维-值对聚集数据，计算该点的度量值。

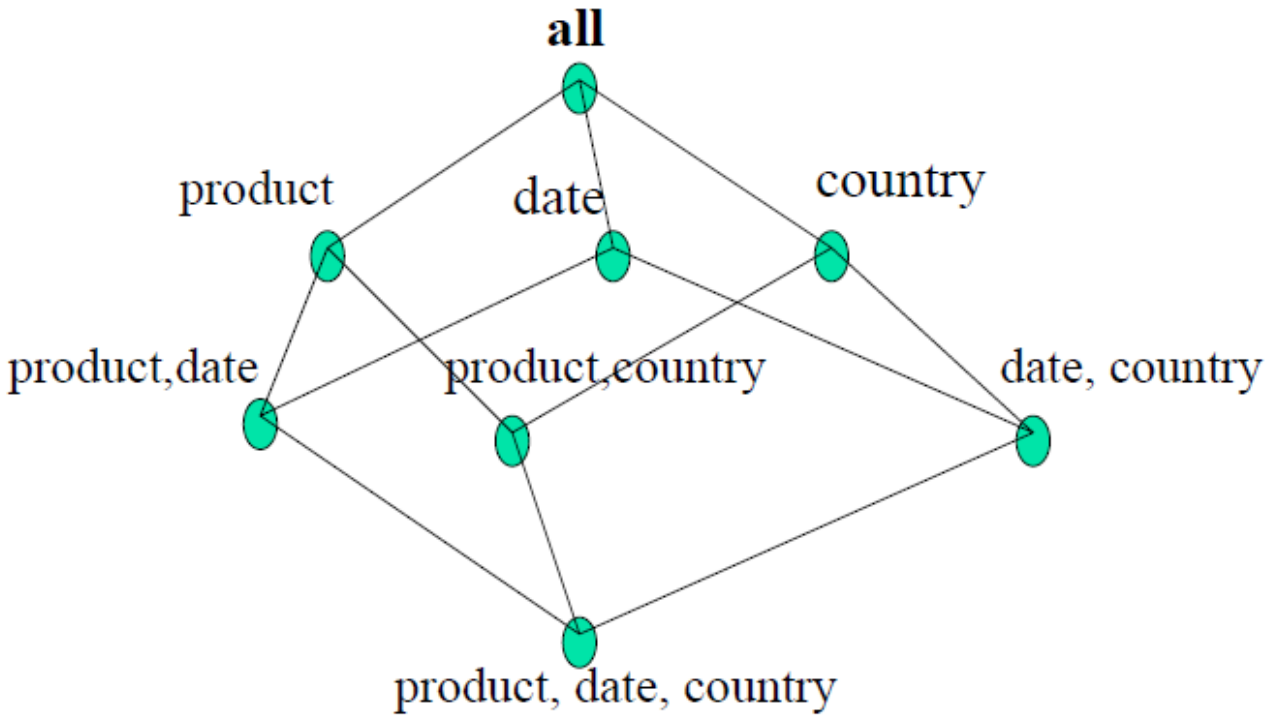
度量的三种分类（根据聚集函数）

- 分布的（distributive)：如果将函数用于n个聚集值得到的结果，与将函数用于所有的数据得到的结果一样
  - 例如, count(), sum(), min(), max().
- 代数的（algebraic）:如果它能够由一个具有M个参数的代数函数计算（其中M是一个有界整数）而每个参数都可以用一个分布聚集函数求得。
  - 例如,avg()可以由sum()/count()计算，且sum()和count()都是分布聚集函数。同理，min_N(), standard_deviation()。
- 整体的（holistic）：如果描述它的子聚集所需的存储没有一个常数界。即不存在一个具有M个参数的代数函数进行这一计算（其中M是常数）。
  - 例如, median(), mode(), rank().

数据立方体的例子


U.S.A全年的TV销售额.
Date
1Qtr
2Qtr
3Qtr
4Qtr
sum
Product
TV
PC
VCR
sum
U.S.A
Canada
Mexico
sum
Country
All, All, All

对应于立方体的方体格



0-D(顶点) 方体

1-D方体

2-D方体

3-D(基本) 方体

典型的OLAP操作

- 上卷（Roll up）:汇总数据
  - 通过维的概念分层向上攀升或者通过维归约来实现
- 下钻(roll down):上卷的逆操作
  - 从高层的汇总到低层汇总或详细数据，或者引入新的维来实现

表 A （单位：万美元）

部门	销售
部门 1	90
部门 2	60
部门 3	80

按时间维向下钻取 ⇓

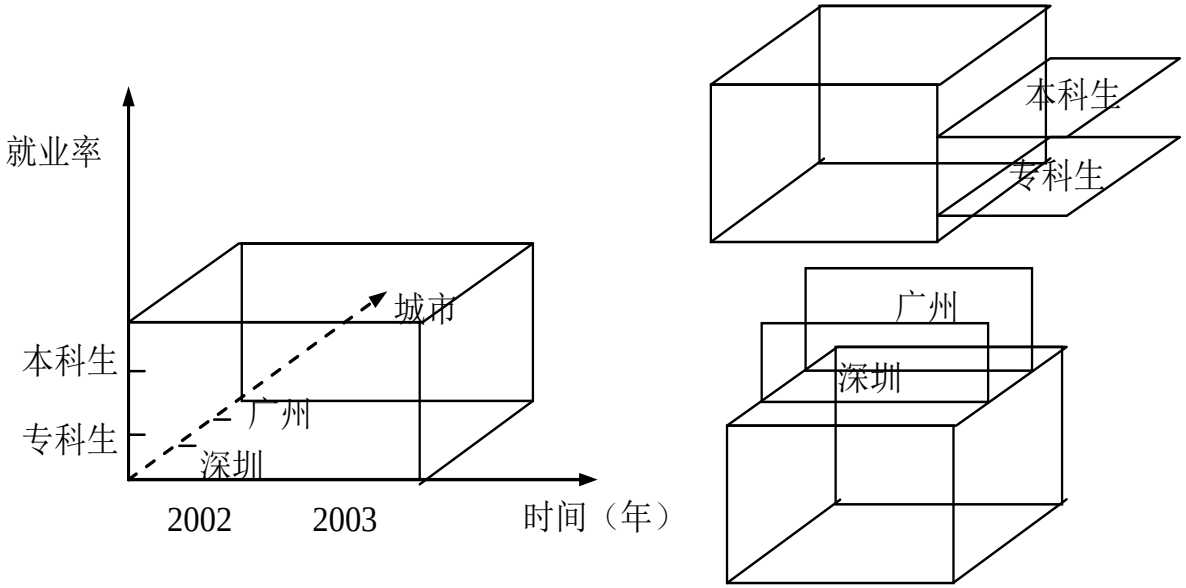
⇑ 按时间维向上钻取

表 B（单位：万美元）

	1995 年			
部门	1 季度	2 季度	3 季度	4 季度
部门 1	20	20	35	15
部门 2	25	5	15	15
部门 3	20	15	18	27

典型的OLAP操作

- 切片（Slice）和切块（dice）：
  - 映射和选择
  - 切片操作在给定的数据立方体的一个维上进行选择，导致一个子方。
  - 切块操作通过对两个或多个维执行选择，定义子方。



典型的OLAP操作

- 转轴（Pivot）
  - 是一种目视操作，它转动数据的视角，提供数据的替代表示。

表 A（单位：万美元）

	1995 年				1996 年			
部门	1 季度	2 季度	3 季度	4 季度	1 季度	2 季度	3 季度	4 季度
部门 1	20	20	35	15	12	20	25	14
部门 2	25	5	15	15	20	18	23	12
部门 3	20	15	27	27	18	20	17	25

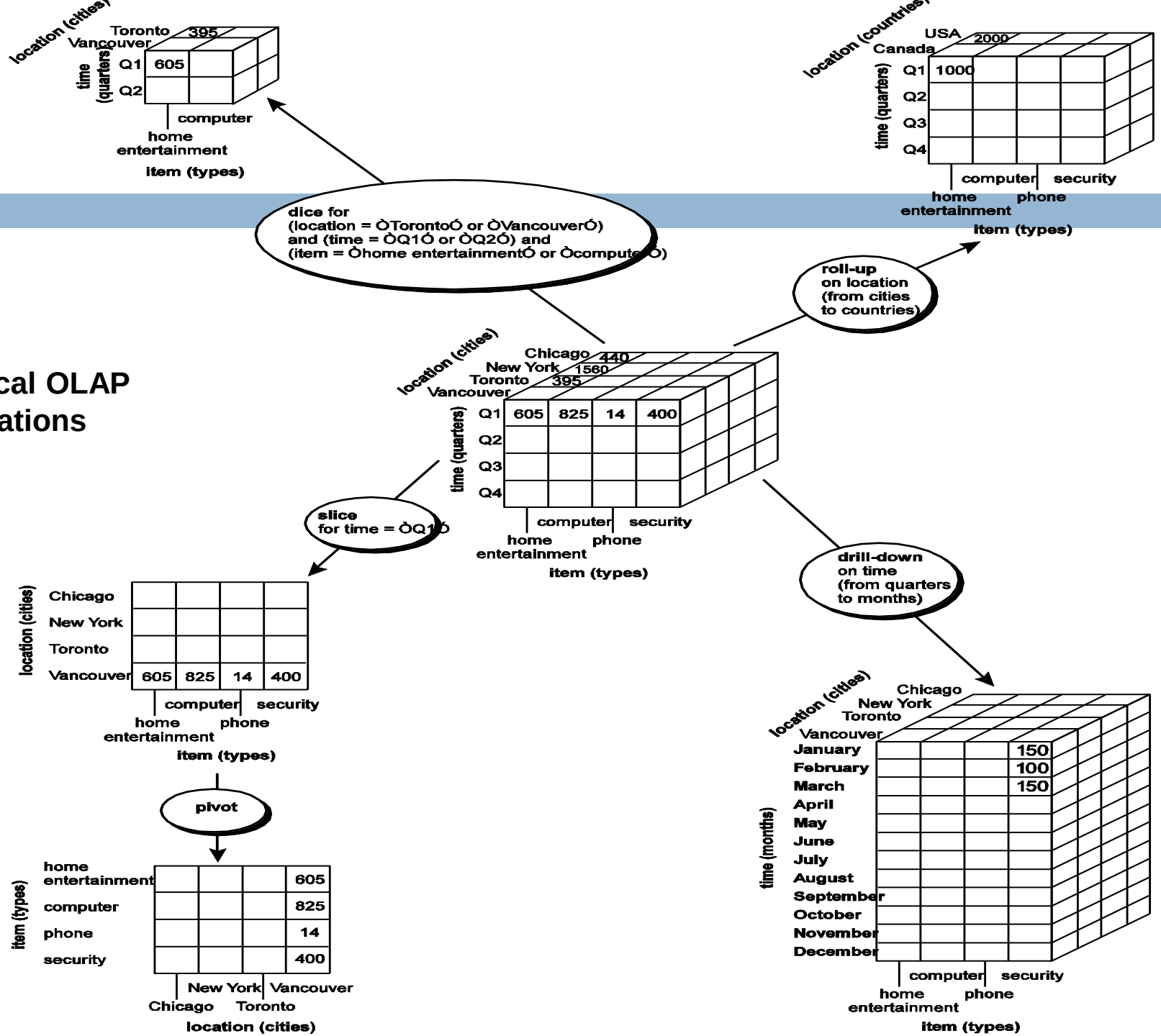
表 B（单位：万美元）

	1 季度		2 季度		3 季度		4 季度	
部门	1995 年	1996 年	1995 年	1996 年	1995 年	1996 年	1995 年	1996 年
部门 1	20	12	20	20	35	25	15	14
部门 2	25	20	5	18	15	23	15	12
部门 3	20	18	15	20	27	17	27	25

典型的OLAP操作

- 其他的操作
  - 钻过（drill across):涉及多个事实表的查询
  - 钻透（drill through）:钻到数据立方体的底层，到后端关系表(使用SQL)

Typical OLAP Operations



主要内容

- 数据仓库的概念
- 数据仓库的建模：数据立方体与OLAP
- 数据仓库的设计与使用
- 数据仓库实现
- 数据泛化：面向属性的归纳
- 小结

数据仓库的设计

- 数据仓库设计的四种视图
  - 自顶向下视图
    - 允许选择数据仓库的所需的相关信息，这些信息能够满足当前和未来商务的需求
  - 数据源视图
    - 揭示被操作数据库系统捕获、存储和管理的信息。这些信息可能以不同的详细程度和精度建档，存放在由个别数据源到集成的数据源表中。
  - 数据仓库视图
    - 由事实表和维表构成，提供存放在数据仓库内部的信息。
  - 商务查询视图
    - 从最终用户的角度透视数据仓库的数据

数据仓库的设计过程

- 使用自顶向下方法、自底向上方法或二者结合的混合方法设计
  - 自顶向下方法：由总体设计和规划开始(当技术成熟并已掌握，这种方法是有用的)
  - 自底向上方法：以实验和原型开始(在商务建模和技术开发的早期阶段，这种方法是有用的)
  - 混合方法：一个组织既能利用自顶向下方法的有计划的战略性的特点，又能保持像自底向上方法一样快速实现和立即应用。
- 从软件工程的观点
  - 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：涉及功能渐增的系统的快速产生，相继版本的时间间隔很短

数据仓库的设计过程

- 典型的数据仓库设计过程
  - 选取待建模的商务处理
    - 如果一个商务过程是整个组织的，并涉及多个复杂的对象，应该选用数据仓库模型，如果处理是部门的，并关注某一类商务处理，则应选择数据集市。
  - 选取商务处理的粒度
  - 选取用于每个事实表记录的维
  - 选取事实表中每条记录的度量

主要内容

- 数据仓库的概念
- 数据仓库的建模：数据立方体与OLAP
- 数据仓库的设计与使用
- 数据仓库实现
- 数据泛化：面向属性的归纳
- 小结

数据立方体的有效计算

- 数据立方体可以被看成是方格体
  - 最底层的方体是基本方体
  - 最上层方体（顶点方体）只包含一个元
  - 那麽一个具有L层的n维立方体有多少个方体?

$$T = \prod_{i=1}^{n} (L_i + 1)$$

    - Li是维i（除去虚拟的顶层all，因为概化到all等价于去掉一个维）的层次数
- 数据立方体的物化
  - 方体的物化有三种选择
    - 不预先计算任何“非基本”方体（不物化）
    - 预先计算所有方体（全物化）
    - 在整个可能的方体集中有选择地物化一个适当的子集（部分物化）
  - 特点
    - 第一种选择导致在运行时计算昂贵的多维聚集，可能很慢。
    - 第二种选择可能需要海量存储空间，存放所有预先计算的方体。
    - 第三种选择在存储空间和响应时间二者之间提供了很好的折衷。

立方体操作

- 立方体在**DMQL**中的定义和计算

  define cube sales[item, city, year]: sum(sales_in_dollars)

  compute cube sales

- 把它变成类**SQL**语言

  SELECT item, city, year, SUM (amount)

  FROM SALES

  CUBE BY item, city, year

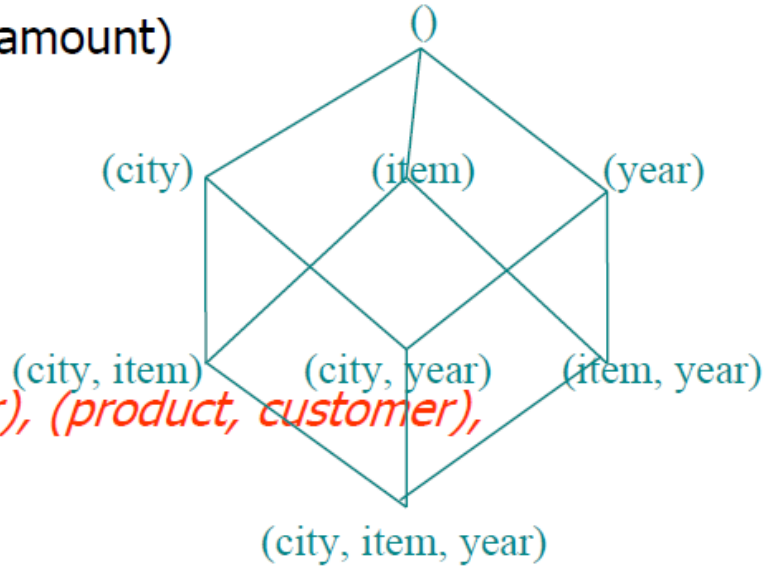
- 需要计算以下的分组

  (date, product, customer),

  (date,product),(date, customer), (product, customer),

  (date), (product), (customer)

  ()



索引OLAP 数据: 位图索引

- 在特定栏上的索引
- 这一栏上的每一个值都对应于一个位向量
- 位向量的长度：基本表中特定栏属性值的个数。
- 如果基本表中的给定行的属性值为v，则在位图索引的对应行，表示该值的位为1，该行的其它位均为0
- 对于基数较大的域不大适合

基本表

Cust	Region	Type
C1	Asia	Retail
C2	Europe	Dealer
C3	Asia	Dealer
C4	America	Retail
C5	Europe	Dealer

Region上的索引

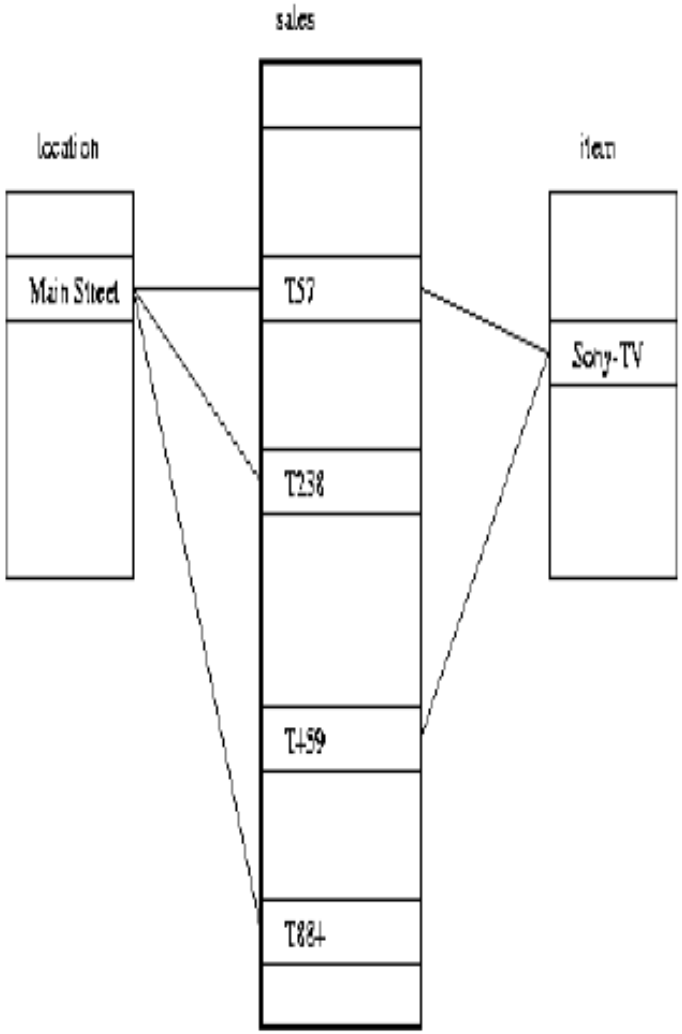
RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

Type上的索引

RecID	Retail	Dealer
1	1	0
2	0	1
3	0	1
4	1	0
5	0	1

索引OLAP数据:连接索引

- 连接索引：如果两个关系R(RID ,A)和S(B,SID)在属性A和B上连接，则连接索引记录包含JI（RID,SID)对，其中RID和SID分别来自R和S的记录标识符。
- 传统的索引将给定列上的值映射到具有该值的列表上，而连接索引登记来自两个关系数据库的可连接行
- 在数据仓库中，连接索引把星形模式的维值连接到事实表中的行，事实表和它对应维表的连接属性是事实表的外关键字和维表的主关键字
  - 连接索引可以跨越多维，形成复合连接索引



OLAP查询的有效处理

- 物化方体和构造OLAP索引结构的目的
  - 加快数据立方体中的查询处理。
- 给定物化的视图，查询处理应按如下步骤进行
  - 确定哪些操作应当在可利用的方体上执行
    - 将查询中的下钻，上卷等转换成对应的SQL 和/或OLAP操作，例如，数据立方体上的切片和切块可能对应于物化方体上的选择和/或投影操作
  - 确定相关操作应当使用那些物化的方体
    - 涉及找出可能用于回答查询的所有物化方体，使用方体之间的“支配”联系知识，剪去上集合，估计使用剩余物化方体的代价，并选择代价最低的方体。

OLAP服务器结构

- 关系OLAP (ROLAP)
  - 使用关系和扩充关系DBMS存放并管理数据仓库，OLAP中间件支持其余部分。
  - 包括每个DBMS后端的优化，聚集导航逻辑的实现，和附加的工具和服务
  - 比MOLAP技术具有更大的可伸缩性
- 多维OLAP (MOLAP)
  - 基于数组的多维存储引擎，支持数据的多维视图。
  - 将多维视图直接映射到数据立方体数组结构。
  - 能够对预计算的汇总数据进行快速索引
- 混合OLAP (HOLAP)
  - 结合ROLAP和MOLAP技术，提高了用户的灵活性
- 特殊的SQL服务器
  - 在星形和雪花模式上支持SQL查询

数据仓库的使用

- 三种数据仓库应用
  - 信息处理
    - 支持查询和基本的统计分析，并使用交叉表、表、图表或图进行报告
  - 分析处理
    - 数据仓库数据的多维分析
    - 支持基本的OLAP 操作，包括切片、下钻、上卷和转轴
  - 数据挖掘
    - 隐含模式中的知识发现
    - 支持关联模式，构造分析模型，进行分类和预测并使用可视化工具提供挖掘结果

主要内容

- 数据仓库的概念
- 数据仓库的建模：数据立方体与OLAP
- 数据仓库的设计与使用
- 数据仓库实现
- 数据泛化：面向属性的归纳
- 小结

面向属性的归纳（Attribute-Oriented Induction）

- Proposed in 1989 (KDD '89 workshop)
- Not confined to categorical data nor particular measures
- How it is done?
  - Collect the task-relevant data (*initial relation*) using a relational database query
  - Perform generalization by attribute removal or attribute generalization
  - Apply aggregation by merging identical, generalized tuples and accumulating their respective counts（合并相同的广义元组以及它们的计数）
  - Interaction with users for knowledge presentation

Attribute-Oriented Induction: An Example

Example: Describe general characteristics of graduate students in the University database

- Step 1. Fetch relevant set of data using an SQL statement, e.g.,

  Select * (i.e., name, gender, major, birth_place, birth_date, residence, phone#, gpa)

  from student

  where student_status in {“Msc”, “MBA”, “PhD” }

- Step 2. Perform attribute-oriented induction
- Step 3. Present results in generalized relation, cross-tab, or rule forms

Class Characterization: An Example

Initial Relation

Name	Gender	Major	Birth-Place	Birth_date	Residence	Phone #	GPA
Jim Woodman	M	CS	Vancouver,BC, Canada	8-12-76	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-75	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-70	125 Austin Ave., Burnaby	420-5232	3.83
…	…	…	…	…	…	…	…
Removed	Retained	Sci,Eng, Bus	Country	Age range	City	Removed	Excl, VG,..

Prime Generalized Relation

Gender	Major	Birth_region	Age_range	Residence	GPA	Count
M	Science	Canada	20-25	Richmond	Very-good	16
F	Science	Foreign	25-30	Burnaby	Excellent	22
…	…	…	…	…	…	…

Gender \ Birth_Region	Canada	Foreign	Total
M	16	14	30
F	10	22	32
Total	26	36	62

Basic Principles of Attribute-Oriented Induction

- Data focusing: task-relevant data, including dimensions, and the result is the *initial relation*
- Attribute-removal: remove attribute *A* if there is a large set of distinct values for *A* but (1) there is no generalization operator on *A*, or (2) *A*'s higher level concepts are expressed in terms of other attributes
- Attribute-generalization: If there is a large set of distinct values for *A*, and there exists a set of generalization operators on *A*, then select an operator and generalize *A*
- Attribute-threshold control: specified/default
- Generalized relation threshold control: control the final relation/rule size

Mining Class Comparisons

- Comparison: Comparing two or more classes
- Method:
  - Partition the set of relevant data into the target class and the contrasting class(es)
  - Generalize both classes to the same high level concepts
  - Compare tuples with the same high level descriptions
  - Present for every tuple its description and two measures
    - support - distribution within single class
    - comparison - distribution between classes
  - Highlight the tuples with strong discriminant features
- Relevance Analysis:
  - Find attributes (features) which best distinguish different classes

Concept Description vs. Cube-Based OLAP

- **Similarity:**
  - Data generalization
  - Presentation of data summarization at multiple levels of abstraction
  - Interactive drilling, pivoting, slicing and dicing
- **Differences:**
  - OLAP has systematic preprocessing, query independent, and can drill down to rather low level
  - AOI has automated desired level allocation, and may perform dimension relevance analysis/ranking when there are many relevant dimensions
  - AOI works on the data which are not in relational forms

主要内容

- 什么是数据仓库?
- 多维数据模型
- 数据仓库的系统结构
- 数据仓库实现
- 从数据仓库到数据挖掘
- 小结

小结

- 数据仓库
  - 数据仓库是面向主题的、集成的、时变的和非易失的有组织的数据集合，支持管理的决策过程。
- 数据仓库的多维数据模型
  - 星形模式、雪花模式和事实星座模式
  - 一个数据立方体有大量事实（或度量）和许多维组成
- **OLAP**操作
  - 上卷、下钻、切片、切块和转轴
- **OLAP** 服务器
  - 关系**OLAP(ROLAP)**, 多维**OLAP(MOLAP)**,混合**OLAP(HOLAP)**
- 数据立方体的有效计算
  - 部分物化、全物化和不物化
  - 多路数组聚集
  - 位图索引和连接索引

参考文献（1）

- S. Agarwal, R. Agrawal, P. M. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. In Proc. 1996 Int. Conf. Very Large Data Bases, 506-521, Bombay, India, Sept. 1996.
- D. Agrawal, A. E. Abbadi, A. Singh, and T. Yurek. Efficient view maintenance in data warehouses. In Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data, 417-427, Tucson, Arizona, May 1997.
- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data, 94-105, Seattle, Washington, June 1998.
- R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In Proc. 1997 Int. Conf. Data Engineering, 232-243, Birmingham, England, April 1997.
- K. Beyer and R. Ramakrishnan. Bottom-Up Computation of Sparse and Iceberg CUBEs. In Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99), 359-370, Philadelphia, PA, June 1999.
- S. Chaudhuriand U. Dayal. An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 26:65-74, 1997.
- OLAP council. MDAPI specification version 2.0. In http://www.olapcouncil.org/research/apily.htm, 1998.
- J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. Data Mining and Knowledge Discovery, 1:29-54, 1997.

参考文献（2）

- V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data, pages 205-216, Montreal, Canada, June 1996.
- Microsoft. OLEDB for OLAP programmer's reference version 1.0. Inhttp://www.microsoft.com/data/oledb/olap, 1998.
- K. Ross and D. Srivastava. Fast computation of sparse datacubes. In Proc. 1997 Int. Conf. Very Large Data Bases, 116-125, Athens, Greece, Aug. 1997.
- K. A. Ross, D. Srivastava, and D. Chatziantoniou. Complex aggregation at multiple granularities. In Proc. Int. Conf. of Extending Database Technology (EDBT'98), 263-277, Valencia, Spain, March 1998.
- S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of OLAP data cubes. In Proc. Int. Conf. of Extending Database Technology (EDBT'98), pages 168-182, Valencia, Spain, March 1998.
- E. Thomsen. OLAP Solutions: Building Multidimensional Information Systems. John Wiley & Sons, 1997.
- Y. Zhao, P. M. Deshpande, and J. F. Naughton. An array-based algorithm for simultaneous multidimensional aggregates. In Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data, 159-170, Tucson, Arizona, May 1997.

习题P117-118

- 习题4.3
- 习题4.4
- 习题4.5
  - （a）
  - （b）

思考题

- 试分析对数据仓库建模的星形模式和雪花形模式相似点和不同点？各自的优缺点？
- 试分析3种数据仓库的应用：信息处理、分析处理和数据挖掘的区别是什么？

结　束