商务智能

构造数据仓库

- 明确需求
  - 用户的主观分析需求
- 选择工具
  - 数据抽取（data extract）工具
  - 数据清洗（data cleaning）工具
  - 元数据(metadata)管理工具
  - 数据分析工具（look for patterns）
  - 数据挖掘工具（look for hidden patterns）
  - 数据展现工具
- 构建数据仓库

操作型处理

- 也叫事务处理，是指对数据库的日常联机访问操作，通常是对一个或一组记录的查询和修改，主要是为企业特定的应用服务的，所以也叫联机事务处理。
- On-Line Transaction Processing（OLTP）
  - 通常仅仅是对一个或一组记录的查询或修改；
  - 查询简单，但执行频率高；
  - 人们关心的是处理的响应时间、数据的安全性和完整性等指标。

分析型处理

- 也叫做信息型处理，主要用于企业管理人员的决策分析，为制订企业的未来经营管理计划提供辅助决策信息。
  - 需要对大量的事务型数据进行统计、归纳和分析；
  - 需要访问大量的历史数据；
  - 执行频率和对响应时间的要求都不高。

-典型的的分析型处理

  - 决策支持系统（DSS --Decision Support System）

事务处理环境不适宜 DSS 应用的原因

- 在传统的以数据库为核心的事务处理环境中不适宜建立 DSS 等分析型应用，其原因主要有以下六条：
  - 事务处理和分析处理的性能特性不同：

    用户每次操作处理的时间短，存取数据量小，但操作频率高，并发程度大。

    每次分析可能需要连续运行很长的时间，存取数据量大，但很少做这样的分析处理，也没有并发执行的要求。
  - 数据集成问题：

    事务处理一般只需要与本部门业务有关的当前细节数据

    分析处理的数据可能来自多种不同的数据源

    对于需要集成数据的 DSS 应用来说，在应用程序中对事务处理环境中的这些纷繁复杂的数据进行集成,将带来下述问题：

    大大加重程序员的负担

    重复计算

    极低的分析处理效率
  - 数据的动态集成问题：

    集成数据必须以一定的周期进行刷新（即采用动态集成策略），但传统的事务处理环境并不具备动态集成的能力。
  - 历史数据问题

    事务处理一般只需要当前数据

    分析处理更看重历史数据
  - 数据的综合问题

    事务处理需要的是当前的细节性操作数据，而分析处理需要的往往是大量的总结性分析型数据，而非数据库中的细节性操作型数据。

在分析前往往需要对细节数据进行不同程度的综合，传统的事务处理系统不具备这种综合能力

- 数据的访问问题

事务处理对于需要修改的数据必须实时‘更新’数据库

分析处理不需要实时的‘更新’操作，但需要定时‘刷新’

操作型数据和分析型数据

特　性	操 作 型 数 据（DB）	分 析 型 数 据（DW）
定位	面向应用的事务处理	面向主题的数据分析
DB 设计	E-R 模型	星型/雪花模型，数据立方体
数据	当前的、最新的	历史的，具有时间跨度
汇总	原始的，细节的	集成的，一致的
视图	详细的，关系的	总体的，多维的
操作类型	读/写（可变的）	读（稳定的）
存取请求	可预知的	事先未知的
访问记录	一次操作少量记录	一次操作大量记录
DB 规模	100MB ~ GB	TB
工作单位	短的，简单事务	复杂查询
性能要求	对性能要求高	对性能要求较宽松

数据仓库定义

数据仓库就是一个面向主题的、集成的、不可更新的、随时间不断变化的数据集合，用于支持经营管理过程中的决策制定。

数据仓库的四个特征：

- 面向主题

主题是较高层次上将企业信息系统中的数据综合、归类并进行分析利用的抽象。在逻辑意义上，是对应企业中某一宏观分析领域涉及的分析对象。

面向主题是指数据仓库内的信息是按主题进行组织的，为按主题进行决策的过程提供信息。

如果按照面向主题的方式进行数据组织，首先应该抽取主题，即按照管理人员的分析要求来确定主题，而与每个主题相关的数据又与有关的事务处理所需的数据不尽相同。在该例中，我们可以抽取出三个不同的主题（即分析对象）及其相关的数据：

- 集成

数据仓库中的数据是为分析服务的，而分析需要多种广泛的不同数据源以便进行比较、鉴别，因此数据仓库中的数据必须从多个数据源中获取，这些数据源包括多种类型数据库、文件系统以及 Internet 网上数据等，它们通过数据集成而形成数据仓库中的数据。

- 不可更新

数据仓库中的数据是经过抽取而形成的分析型数据，不具有原始性，主要供企业决策分析之用，执行的主要是‘查询’操作，一般情况下不执行‘更新’操作。同时，一个稳定的数据环境也有利于数据分析操作和决策的制订

- 随时间不断变化

数据仓库内的信息并不只是关于企业当时或某一时点的信息，而是系统记录了企业从过去某一时点到目前的各个阶段的信息，通过这些信息可以对企业的发展历程和未来趋势作出定量分析和预测。

因此数据仓库中的数据通常都带有时间属性，同时必须以一定时间段为单位进行统一更新。

数据仓库的关键技术

- 数据的 ETL（抽取、转换、装载）
- 存储和管理
- 数据的访问和表现

- 元数据

数据抽取

- 数据仓库中的数据来源于数据源，将数据源中数据通过网络进行抽取，并经加工、转换、综合后形成数据仓库中的数据，这就是数据仓库的数据抽取。

数据刷新

- 经过抽取进入数据仓库的数据，在经过一段时间后要重新修正，修改那些过时的数据，保存那些不变的数据，此种动作称为数据仓库的数据刷新。
- 数据刷新的过程与抽取类似，但刷新的数据量往往小于抽取的数据量。由于仅需要对修改过的数据进行刷新，因而其实现难度与复杂性要大于数据抽取。

一般数据刷新的方法包括：

- 时间戳

  适用情况

  若数据库中的记录有时间属性，则可根据 OLTP 数据库中的数据有无更新，以及在执行更新操作时数据的修改时间标志来实现数据仓库中数据的动态刷新。

  缺点:大多数数据库系统中的数据并不含有时间属性。

- DELTA 文件

  适用情况

  有些 OLTP 数据库的应用程序在工作过程中会形成一些 DELTA 文件以记录该应用所作的数据修改操作，可根据该 DELTA 文件进行数据刷新。

  优点:采用此方法可避免对整个数据库的对比扫描，具有较高的刷新效率。

  缺点:这样的应用程序并不普遍，修改现有的应用程序的工作量又太大。

- 建立映象文件

  实现方法

  在上一次数据刷新后对数据库作一次快照

  在本次刷新之前再对数据库作一次快照

  比较两个快照的不同，从而确定数据仓库的数据刷新操作。

  缺点:需要占用大量的系统资源,可能较大地影响原有数据库系统的性能

- 日志文件

  实现方法

  一般 OLTP 数据库都有日志文件，可根据 OLTP 数据库的日志信息来实现数据仓库的数据刷新。

  优点:日志是 OLTP 数据库的固有机制,不会影响原有 OLTP 数据库的性能,具有比 DELTA 文件和建立映象文件更高的刷新效率

历史完整性？

数据仓库是多维度多层次的

- 维度是观察数据对象的角度
- 层次是数据对象的综合程度

数据仓库的数据组织形式

- 简单堆积文件
- 轮转综合文件
- 简化直接文件
- 连续文件

元数据？

一个完整的数据仓库/数据集市体系结构一般由三个层次组成,它们是：
数据源
数据仓库
数据集市（Data Mart）

数据仓库与数据集市的关系类似于传统关系数据库系统中的 基表 与 视图 的关系。数据集市的数据来自数据仓库，它是数据仓库中数据的一个部分与局部，是一个数据的再抽取与组织的过程。

数据仓库与数据集市的关系

- 自顶向下的结构
- 自底向上的结构
- 总线结构的数据集市
- 企业级数据集市结构

自顶向下的结构

- 构建企业数据仓库
- 基于企业数据仓库构建数据集市
- 优点
    - 建立数据集市能够减轻 DW 访问负载
    - 各部门可以任意处理数据
    - 数据转换和整合在 DW 阶段统一完成
    - 数据缓冲功能
- 缺点
    - 成本高、见效慢、数据集市间不共享资源

自底向上的结构

- 构建数据集市

- 基于数据集市构建企业数据仓库
- 优点
  - 见效快、启动资金少
- 缺点
  - 各个部门都要进行数据清理整合
  - 可能造成“蜘蛛网”、数据不一致等问题
  - 并且总体上没有节约资金

总线结构的数据集市

  - 不建立数据仓库而直接建立数据集市
  - 各个数据集市不是孤立的，相互之间通过一种共享维表和事实表的“总线结构”紧密联系在一起。
- 优点
  - 共享维表和事实表，解决了建立数据集市的许多问题
- 缺点
  - 这种结构基于多维模型，应用限制于 OLAP
  - 多个数据源直接影响多个集市造成结构不十分稳定

数据仓库数据的间接访问

参加 ppt 4 26

比较项目	OLAP	OLTP
应用基础	数据仓库	DBMS
用户	决策者（高级管理人员）	一般操作员（管理人员）
目的	为决策和管理提供支持	为日常工作服务
数据特征	导出数据	原始数据
数据细节	综合性数据，细节程度低	细节程度高
时间特征	历史数据，横跨一个时段	当前数据
更新方法	周期性刷新	可实时更新
数据量需求	一次处理需大量数据	一次处理需少量数据

OLAP 中的几个基本概念

维（Dimension）

- 观察分析对象的角度
- 例如：可以从三个‘维’角度观察‘销售金额’这个对象
  - 时间维
    - 可从时间角度统计（所有）商品在不同时间段内的销售（总）金额，以便于分析其与时间之间的关系
  - 商品维
    - 根据商品的分类情况统计每一类商品的销售金额，以便于分析其与商品类型之间的关系
  - 地域维
    - 可根据每个连锁店所在的地域统计其销售（总）金额，以便于分析其与地域之间的关系

层（Layer）

- 在分析型应用中，对分析对象可以在不同的深度层面上进行分析与观察，并可能得到不同的分析结果。因此，‘层’ 反映了对分析对象的观察深度。
- 按如下的方法进行层次划分

- 按商品的价格分为
- 高档，中档，低档
- 按商品的供应商分为
- 外资，合资，国营，私营，个体
- 按购买商品的顾客信息分为
- 按照年龄层次来划分：老年，中年，青年，少年儿童，婴儿
- 按照所从事的职业来划分：……

维成员

- 维的一个取值称为该维的一个‘维成员’
    - 如果一个维是多层次的，则该维的‘维成员’可以是
        - 在不同维层次上的取值的组合
            - 例如：对具有日，月，年三个层次的‘时间维’来说，‘某年某月某日’、‘某年某月’、‘某月某日’、‘某年’都是其维成员，如：

1998 年，1 月，1998 年 1 月，1998 年 1 月 1 日，1 月 1 日

        - 在某个维层次上的取值
            - 例如：‘地域’维中的‘江苏’，‘南京’，……
            - 例如：‘商品’维中的‘电视机’，‘服装’，……
    - 对一个数据项（分析‘对象’）来说，维成员是该数据项在某维中位置的描述。

数据单元（单元格）

- 当多维数组的每一维都选中一个维成员，这些维成员的组合就唯一确定了一个观察对象的值，即：

（维成员 1，维成员 2，……，维成员 n，对象值）

- 这样一个值或存放该值的地方我们称其为一个‘数据单元’

OLAP 的基本数据模型

OLAP 中的数据构造方式 ？

星型模式

- 星型模式是一种多维表结构，它一般由两种不同性质的二维表组成：
    - 事实表（fact table）
        - 它存放多维表中的主要事实，我们称其为量（Measure）
    - 维表（Dimension Table）
        - 用以存放多维表中的维成员的取值
- 一般一个 n 维的多维表往往有 n 个维表和一个事实表，它们构成了一个星形结构，因而称其为‘星型模式’。
    - 在星型模式中主体是事实表，而有关维的细节则构作于维表内以达到简化事实表的目的，事实表与维表间有公共属性相连以使它们构成一个整体。

雪花模式

- 雪花模型是对星型模型的扩展
    - 雪花模型对星型模型的维表进一步层次化，原来的各维表可能被扩展为小的事实表，形成一些局部的“层次”区域。
- 优点
    - 最大限度地减少数据存储量，使维表尽可能地规范化。
- 缺点
    - 执行查询需要更多的连接操作，可能会影响查询性能。

切片，切块，上钻，下钻（参见书）

    - 切片（Slice）

        - 根据某一维上的某个维成员值选择统计数据进行分析
    - 切块（Dice）
        - 根据某一维上的某个维成员取值的区间选择统计数据进行分析
        - 根据多个维度上的维成员取值的区间选择统计数据进行分析
    - 数据概括（roll_up）
        - 将多维下标的取值提升到较高的概念层次上，从而形成新的统计查询结果，并进行分析。
    - 数据细化（drill_down）
        - 将多维下标的取值降低到较低的概念层次上，从而形成更细致的统计查询结果，并进行分析。

数据仓库设计的原则

- 面向主题原则
- 数据驱动原则
- 原型法设计原则

面向主题原则

- 建立数据仓库的目的
    - 构建数据仓库的目的是面向企业的管理人员，为经营管理提供决策支持信息。因此数据仓库的组织设计必须以用户决策的需要来确定，即从用户决策的主观需求（主题）开始。
- 数据仓库中数据的组织方法
    - 为了进行数据分析首先要有分析的主题，以主题为起始点，进行相关数据的设计，最终建立起一个面向主题的分析型环境。
    - 在数据库设计中则是以客体（Object）为起始点，即以客观操作需求为设计依据。
- 例如：‘商品销售’主题
    - 建立目的
        - 管理人员能够在适当的时候，订购适当的商品，并把它们分发到适当的商店中去销售，以提高商品的销售总金额。
    - 需要执行的分析操作
        - 分析什么样的商品，在什么样的时间和商店内畅销
            - 即分析商品的销售额与商品类型、销售时间及商店位置之间的变化关系
        - 管理人员将据此决定他们的经营策略

数据驱动原则

- 在数据仓库设计中，由于其所有数据均应建立在已有的数据库基础上，即是从已经存在于操作型环境中的数据出发进行数据仓库的设计，这种设计方法被称为 “数据驱动” 方法

原型法设计原则

- 数据仓库系统的原始需求不明确，且不断变化与增加，开发者最初并不能确切了解到用户的明确而详细的需求，用户所能提供的无非是需求的大方向或部分需求，更不能较准确地预见到以后的需求。
- 因此，采用原型法来进行数据仓库的开发是比较合适的，即从构建系统的基本框架着手，不断丰富与完善整个系统。

数据仓库设计的三级数据模型

- 概念模型
    - 为一定目标设计系统、收集信息而服务的概念型工具，是客观世界到机器世界的一个中间层次
    - E-R 法
- 逻辑模型
    - 描述了数据仓库的主题的逻辑实现
    - 关系模型
- 物理模型
    - 逻辑模型在数据仓库中的实现

数据仓库的设计步骤

- 数据仓库设计大致有如下几个步骤：
    1. 系统规划
        - 明确主题

          在数据仓库设计的开始，首先要做的事是有关分析人员需要确定具体领域的分析对象，这个对象就是主题。主题是一种较高层次的抽象，对它的认识与表示是一个逐步完善的过程。因此，在开始时不妨先确定一个初步的主题概念以利于设计工作的开始，此后随着设计工作的进一步开展，再逐步扩充与完善。（原型设计法）
        - 技术准备
    2. 概念设计
        - 确定系统边界
        - 确定主要的主题及其内容
        - OLAP 等分析应用的设计

        一般将数据划分为：详细数据、轻度总结、高度总结三种粒度，或者采用更多级的粒度划分方法

    3. 逻辑设计

        将 E—R 图转换成关系数据库的二维表

        定义数据源和数据抽取规则

        在逻辑模型的设计过程中，需要考虑以下一些问题：

        适当的粒度划分

        合理的数据分割策略

        定义合适的数据来源
    4. 物理设计
    5. 数据仓库生成
    6. 数据仓库的运行与维护

物理模型设计

- 在逻辑模型设计基础上确定数据的存储结构、确定索引策略、确定存储分配及数据存放位置等与物理有关的内容，物理模型设计的具体方法与数据库设计中的大致相似。其目的是为了提高数据仓库系统的访问性能。常用的一些技术有：
    - 合并表
    - 建立数据序列
    - 引入冗余
    - 表的物理分割
    - 生成导出数据
    - 建立广义索引
- 规范化/反规范化

物理模型设计 – 合并表

- 在常见的一些分析处理操作中，可能需要执行多表连接操作。为了节省 I/O 开销，可以把这些表中的记录混合存放在一起，以减低表的连接操作的代价。这样的技术我们称为 合并表 。

考虑创建一个数据数组，这样如果数据存放在一行中，那么一次 I/O 就足以检索到了。通常当数列中值的数量稳定、数据是按顺序访问的、数据的创建与修改在统计上是以非常有规律的方式进行等条件都满足时，创建一个数组才是有意义的。

在面向某个主题的分析过程中，通常需要访问不同表中的多个属性，而每个属性又可能参与多个不同主题的分析过程。因此可以通过修改关系模式把某些属性复制到多个不同的主题表中去，从而减少一次分析过程需要访问的表的数量。

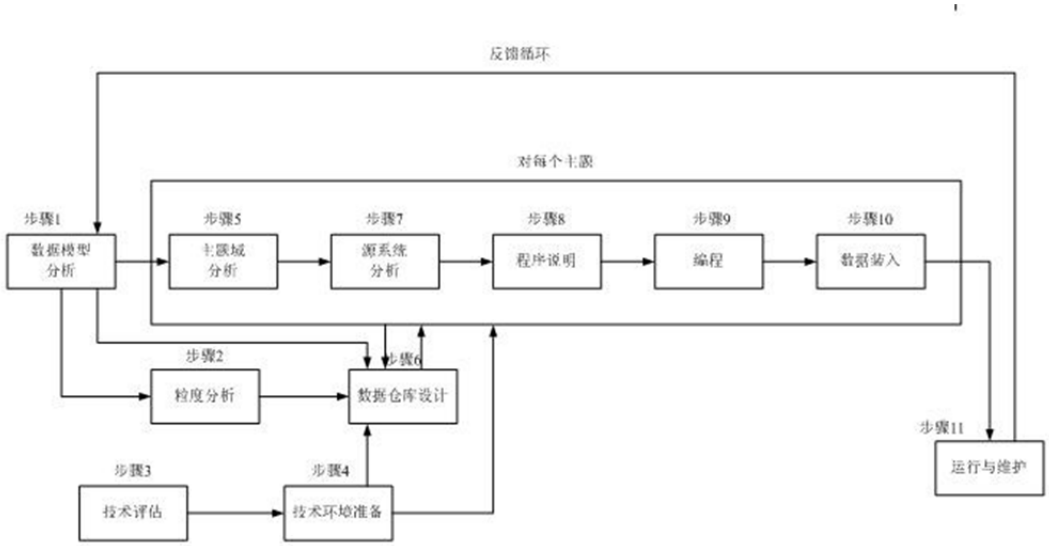
对于访问频率较高的属性，可以单独考虑其物理存储组织，以便选择合适的索引策略和特定的物理组织方式。

对于需要频繁更新的属性，也可以单独组织其物理存储，以免因数据更新而带来的空间重组、重构等工作。

在原始、细节数据的基础上进行一些统计和计算，生成导出数据，并保存在数据仓库中。

- 优点
  - 避免在分析过程中执行过多的统计或计算操作，减少输入/出的次数，提高分析操作的性能
  - 避免了不同用户进行重复统计操作可能产生的偏差

这样的广义索引的数据量是非常小的，可以在每次进行数据仓库数据加载工作时生成或刷新这样的广义索引。用户可以从已经建立的广义索引里直接获取这些统计信息，而不必对整个数据仓库进行扫描。



事实表

- 是维度建模的核心和基本表
- 每一事实表都对应着一个或若干个‘度量值’
  - 度量值是事实表的核心，也是趋势分析的对象
  - 通过事实表来记录维度值与度量值之间的关系
- 事实表中的一行对应一个度量值
  - 事实表中的所有度量值必须具有相同的粒度
  - 粒度划分：事务，周期快照，累积快照
- 最常用的度量值：数值类型
- 三种类型的度量值
  - 可做加法运算
  - 可沿着某些维度做加法运算
  - 不能做加法运算
    - 计数统计
    - 计算平均值
    - 取样统计
- 很少采用文本形式的度量值
- 度量值通常是一个可以连续取值的量
- 每个事实表都有两个或两个以上的外关键字(Foreign Key)
  - 通过外关键字建立事实表与维表之间的联系，从而可以通过维度表来存取事实表中的度量值
  - 可以由外关键字的组合构成事实表的主关键字(Primary Key)

- 维度表的定义通常包括 ？？？ 什么是行 什么是列
  - 尽可能多的列

- 尽可能少的行（相对于事实表）

维度建模的设计过程

- 选取要建模的业务处理过程
  - 分析需要
- 定义业务处理的粒度
  - 事实表中每一行的度量值的取值粒度
- 选择事实表中的维度
- 选择事实表中的度量值
  - 分析对象
  - 可以有多个度量值
- 通过计算而获得的可加性度量值也应该物理存储在事实表中，如：毛利润金额
- 不具有可加性的计算结果则应该由数据存取工具在访问过程中进行计算，如：毛利润率，单价，……

退化维度

- 维度表为空，具体的维度值直接存放在事实表中

维度的规范化处理

规范化	非规范化
雪花模型	星型模型
复杂的表关系	简单的表关系
节省存储空间	记录之间存在数据冗余
连接的复杂，高开销	连接简单，低开销
低维度浏览能力	高维度浏览能力
不支持物理加速技术	支持物理加速技术

代理关键字，避免直接使用操作型数据作为维度表和事实表的主关键字和外关键字

- 可以缓冲操作型数据的变化对数据仓库数据的影响
- 性能优势
- 操作型数据可能无法作为关键字
- 日期维度的特殊要求
- 历史一致性

历史一致性　07　47？？？

值链

- 由企业的关键业务组成
- 值链确定了企业主体活动的自然逻辑流程

三种互补的库存模型

周期库存快照

定期生成每种商品的库存水平（数量）

库存事务

记录影响库存水平的主要因素

商品的进/出仓库等事务

库存累积快照

记录每件商品的分发历史，直至其离开仓库为止

商品的库存事实表与销售事实表的区别

- 销售事实表是稀疏的，而库存事实表则是稠密的
  - 在销售事实表中记录每天实际发生的商品销售情况
  - 而库存事实表则需要记录每天、每种商品、在每个商场的库存情况（不管是否发生了实际的销售事实）
- 解决办法
  - 随着时间的推移可降低周期快照的频度
    - 最近 60 天内的以天为粒度单位的周期快照
    - 最近 3 年内的以周为粒度单位的周期快照

半加型事实 (Semiadditive Facts)

- 只在部分维度上具有可加性的度量值被称为‘半加型事实’
- 在商品营销中，绝大部分的度量值在所有的维度范围内都具有极好的可加性。
- 在库存快照模型中，‘库存量’可以跨‘产品’或‘商场’进行汇总（具有可加性），但不具有跨‘日期’的可加性。

几种常见的半加型事实

- 库存数量，银行帐户余额，温度，水位，含量，……
- 用于记录静态水平的度量值在跨日期维度以及可能的其它维度范围内都是不可加的。
  - 对于不可加的度量值，可用的聚集方法如：平均统计
  - 不能简单地利用 SQL 中的 AVG 函数来完成这样的平均统计计算工作
  - AVG_DATE_SUM

数据仓库总线结构

- 一种可以按增量开发方式分步建造企业数据仓库的方法
  - 计算机中的总线
  - 通过为数据仓库环境定义标准的总线接口，独立的数据集市就可以由不同的开发小组在不同的时间进行实现。只要遵循这个标准，独立的数据集市就可以插入到一起并有效地共享。
    - 数据集市

数据仓库总线矩阵（2/2）

- 矩阵的行：对应着数据集市
  - 如果数据来源不同，处理功能不同，或者矩阵行代表的内容无法在单个迭代过程中合理完成，就应当创建独立的矩阵行
- 矩阵的列：对应着共享的公共维度

一致性维度

- 一致性维度是进一步开发总线结构数据仓库系统的基础
- 一致性维度
  - 要么是同一的，要么是具有最佳粒度与细节性的维度在严格数学意义上的子集
  - 一致的维度具有如下特征
    - 一致的维度关键字
    - 一致的属性列名字
    - 一致的属性定义
    - 一致的属性值
- 一致的维度可能意味着是相同的维度表
  - 与它们相连的事实表具有完全相同的内容（不同的度量值）。例如：
    - 连接到销售事实表与库存事实表上的日期维度表是同一的，意味着销售事实表和库存事实表中的内容是相

同的

- 这样的维度表在物理上可能是同一张表，也可能是不同的表，但它们应该具有*相同数目的行*、*相同的关键字值*、*相同的属性标签*、*相同的属性定义与相同的属性值*。

- 原子型维度
  - 在最佳粒度层次上的维度定义（最小的粒度）
- 堆积维度 (roll-up dimensions)
  - 在较高层次上的维度定义（较大的粒度）
    - 每日快照 vs. 每周快照
  - 如果堆积维度是基本层次上原子型维度严格意义上的子集，则堆积维度与原子型维度保持一致。

一致性事实

- 同样的事实在不同的数据集市进行存储的一致性
  - 取值单位的一致性
  - 值的一致性
  - 自然关键字的一致性
- 一般说来，事实表数据不在多个数据集市明确的进行拷贝。
- 如果事实表存在于多个数据集市，那么支撑这些事实的定义和方程必须都是相同的
- 如果无法使事实完全保持一致，那么应该对不同的解释给与不同的名称

日期维度的角色模仿

- 后台只维持一个单一的日期维度表
- 为事实表中的每一个日期外关键字建立一个日期维表上的视图

优点：降低存储空间开销，方便使用

三种类型事实表的比较

特　征	事务粒度	周期快照粒度	累积快照粒度
代表的时间段	时间点	规律性可预见间隔	不确定时间跨度，一般是短期
粒度	每个事务事件一行	每段一行	每个生命期一行
事实表加载	插入	插入	插入与更新
事实行更新	不重新存取	不重新存取	行为发生任何时候都要重新存取
日期维度	事务发生日期	时间段终止日期	标准关键环节的多个日期
事实	事务活动	预定时间间隔的性能	给定生命期的性能

三种不同类型的实时分区

- 事务粒度 – 当天的记录（并非统计结果）
- 周期快照 – 最近一个周期内的统计结果
  - 对非/半加性事实的考虑
- 累积快照 – 只记录最近被更新的项

支架维度

- 将一组低基数属性单独构成客户维度的一个维度（称为支架维度），从而使整个模型呈雪花状
- 支架维度中的数据一般是从外部数据提供者那里获得的。

        - 如：县人口统计支架维度
- 使用维度支架的好处
    - 客户维度与支架维度具有相差悬殊的粒度
    - 具有不同的管理与加载次数
    - 可以节省客户维度表的存储空间
    - 如果用户的查询工具坚持使用星型结构，那么可以通过视图定义来隐藏维度支架

在数据仓库的维度模型中，部分维度属性是会随时间而发生变化的，若只是将这些变化的维度属性值作简单的修正，即在维度表中只保留该维度属性的当前值，这会直接影响到对事实表中该维度属性所对应的事实数据元组的访问，特别是无法根据维度属性值的变化情况来进行分析处理。

维度表的划分

- 稳定维度
- 渐变维度
- 快变维度

渐变维度的处理办法

- 类型 1：改写属性值
- 类型 2：添加维度行
    - 在新的元组上记录修改后的属性值，同时系统将为该元组生成新的代理关键字
    - 其它维度属性上的值不变
    - 可以考虑在维度表中增加两个日期属性：维度的 *生效日期* 和 *截止日期*
- 类型 3：添加维度列
    - 在新增加的属性列上记录修改后的属性值

什么是数据挖掘？

- 定义一：数据挖掘就是对数据库(数据仓库)中***蕴涵的、未知的、非平凡的、有潜在应用价值的模式(规则)***的提取。
- 定义二：数据挖掘就是从大型数据库(数据仓库)的数据中提取人们***感兴趣的知识***。这些知识是***隐含的、事先未知的潜在有用信息***。

数据挖掘中的几个基本概念

- 模式（pattern）
- 知识（discovered knowledge）
- 置信度（confidence）
- 兴趣度（interestingness）
- 非平凡性（nontrivial）
- 有效性（effectiveness）

模式

- 用高级语言表示的表达一定逻辑含义的信息，这里通常指数据库中数据与数据之间的逻辑关系。
    - 例如：在某超市的商品销售数据库中，我们可以找到以下信息：
        - 男性顾客在购买婴儿尿布时也往往同时购买啤酒
        - 在购买面包和黄油的顾客中，大部分的人同时也买了牛奶

知识

- 满足用户对兴趣度和置信度的要求的模式。

置信度

- 模式在某一数据集上成立的程度。
    - 例如：
        - 模式 R1：***在购买面包和黄油的顾客中，大部分的人同时也买了牛奶***
            - 该模式的置信度为：同时购买‘面包、黄油、牛奶’的顾客人数占同时购买‘面包、黄油’的顾

客人数的百分比，即：

- 通过数据挖掘所发现的模式的置信度的大小涉及到许多因素，如数据的完整性、样本数据的大小、领域知识的支持程度等。
  - 如果没有足够的置信度，模式不能成为知识。因此在数据挖掘过程中，通常要规定模式的***最小置信度***

兴趣度

- 在一定数据集上为真的知识被用户关注的程度。
  - 用户对知识感兴趣的程度也可以用知识的支持度（support）来表示。
    - 例如：模式 R1 的支持度为“***同时购买‘面包，黄油和牛奶’的顾客人数占总顾客人数的百分比***”，即：
- 只有当一个模式的‘***兴趣度***’达到一定的程度时，那么该模式才是一个有意义的模式，才能引起用户的注意，有助于用户的决策制订过程。
  - 因此，在数据挖掘过程中也要规定模式的‘***最小支持度***’，以淘汰哪些在极少情况下才会出现的模式。

非平凡性

- 平凡知识
  - 能够以确定的计算过程提取的模式称为平凡知识。
    - 例如：根据数据库中的薪水字段求得职员的平均薪水
  - 平凡的知识不是数据挖掘的目标。
- 在数据挖掘中，知识的发现过程都应具有某种不确定性和一定的自由度，也就是要发现*不平凡的知识*。

有效性

- 知识的发现过程必须能够有效地在计算机上实现。

常用的数据挖掘方法

- 特征规则挖掘
  - 面向属性归约方法
  - 数据立方方法
- 关联规则挖掘
- 序列模式分析
- 分类分析
- 聚类分析

面向属性归约方法

- 这是一种常用的特征规则的挖掘方法。
  - 它通过对属性值间概念的层次结构进行归约，以获得相关数据的概括性知识，通常又称为普化知识。
- 在实际情况中，许多属性都可以进行数据归类，形成概念汇聚点。
  - 这些概念依抽象程度的不同可构成描述它们层次结构的概念层次树。
  - 根据概念层次树可以对供挖掘用的数据进行预处理，以生成一个适合于进行数据挖掘工作的数据集。因此‘面向属性’的数据规约过程实际上是为进行数据挖掘工作而进行的数据预处理。

数据立方方法（2/2）

- 在数据立方方法中，常用的分析方法有：
  - 数据概括（roll_up 上翻）
    - 将属性值提升到较高的概念层次上
      - 如：从‘基本关系表’ 到 ‘概括关系表一’， 再到‘概括关系表二’的分析过程。
  - 数据细化（drill_down 下翻）
    - 将属性值减低一些层次
      - 如：从‘概括关系表二’ 到 ‘概括关系表一’， 再到‘基本关系表’的分析过程。
        - 要降低属性值的概念层次，通常需要在最初的基本关系表中重新进行统计计算。除非在多维数据库中已经保存有所需要的概念层次上的统计结果。

- 支持度(Support)
  - 同时购买 A 和 B 的客户人数占总客户数的百分比称为规则 R_1 的支持度。

- Support(A→B) = Probability(A∪B)
  - Probability(X)表示事件 X 出现的先验概念。在这里表示顾客购买商品 X 的概率，即：
    - Probability(X) =
  - Probability(A∪B)则表示顾客同时购买商品 A 和商品 B 的概率，即：
    - Support(A→B) =

- 置信度(Confidence)
  - 同时购买 A 和 B 的客户人数占购买 A 的客户人数的百分比称为规则 R_1 的置信度。
    - Confidence(A→B) = Probability(B / A)
      - Probability(B / A)表示在发生事件 A 的情况下事件 B 出现的可能性。在这里表示顾客购买商品 A 的同时也购买商品 B 的条件概率，即：
        - Confidence(A→B) =

Apriori 算法?

序列模式分析

- **序列模式分析与关联规则挖掘类似，也是为了找出数据对象之间的联系，但序列模式分析法的侧重点是为了找出数据对象之间的前因后果关系。**
  - **被分析对象具有前后的时序关系**
- **例如：**
  - **下雨 ---- 洪涝**
  - **电筒 ---- 电池**

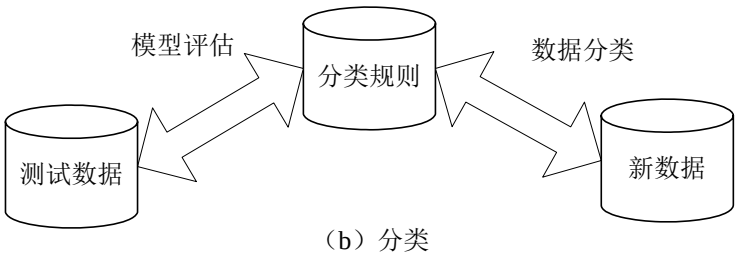
分类分析

- **数据分类是一个两个步骤的过程：**
  - **第 1 步：建立一个模型，描述给定的数据类集或概念集（简称训练集）。通过分析由属性描述的数据库元组来构造模型。每个元组属于一个预定义的类，由类标号属性确定。用于建立模型的元组集称为训练数据集，其中每个元组**

称为训练样本。由于给出了类标号属性，因此该步骤又称为有指导的学习。如果训练样本的类标号是未知的，则称为无指导的学习（聚类）。学习模型可用分类规则、决策树和数学公式的形式给出。

- 第 2 步：使用模型对数据进行分类。包括评估模型的分类准确性以及对类标号未知的元组按模型进行分类。

（a）学习



（b）分类

- 使用决策树进行分类分为两步：
  - 第 1 步：利用训练集建立并精化一棵决策树，建立决策树模型。这个过程实际上是一个从数据中获取知识，进行机器学习的过程。
  - 第 2 步：利用生成完毕的决策树对输入数据进行分类。对输入的记录，从根结点依次测试记录的属性值，直到到达某个叶结点，从而找到该记录所在的类。

聚类分析

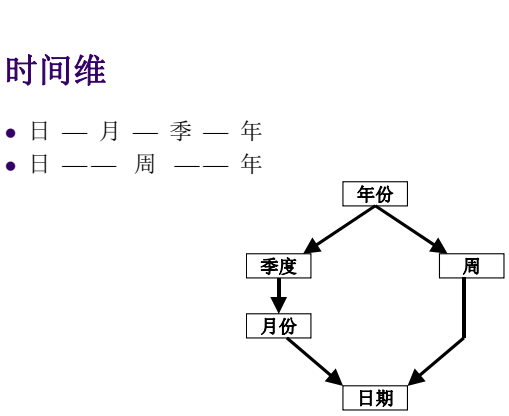
- 聚类分析又称群分析，它是研究分类问题的一种多元统计方法，所谓类，通俗地说，就是相似元素的集合
- 聚类分析分为距离聚类和相似系数聚类

数据挖掘一般可由下面 5 个步骤组成，它们是：

- 数据集成
- 数据归约
- 挖掘
- 评价
- 表示

综合：

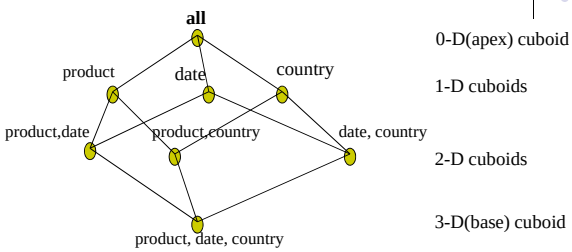
1、 数据立方体，举例，说明维和层，说明切片，切块，上钻，下钻。



数据立方体（1/4）

- 数据仓库的数据模式通常可以看成是定义在多个数据源上的数据视图，其中存储的分析型数据通常是一些经过统计而获得的总结性数据。
  - 获取这些总结性数据的常用方法是在视图中用统计函数进行计算，但这种方法的缺点是显见的：
    - 时间开销太大
- 物化视图
  - 为了提高对统计信息的查询速度，我们可以预先计算好数据视图中的统计信息并保存在数据仓库中，这称为‘物化视图’，即将虚的视图转变成实际的视图。
  - 存放物化视图的三维数据模型叫‘数据立方体’

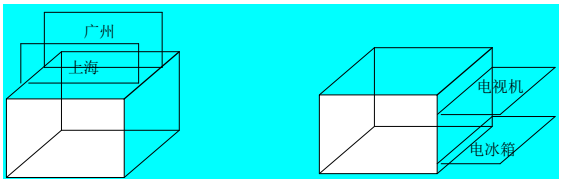
方体格（1/2）



- n维方体称为基本方体
- 0维方体代表最高层次的抽象，称作顶点方体
- 方体格构成数据立方体

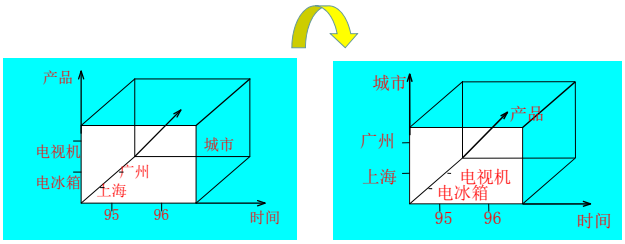
多维数据分析（5/12）

- 切片操作
  - 对三维数据，通过“切片”和“旋转”（选择特定切割方向），分别从城市到产品等不同的角度观察各年的销售情况



多维数据分析（12/12）

- 旋转操作



81

多维数据分析（8/12）

- 钻取操作（细化）
  - 对时间维进行下钻操作，获得新表如下：

表7.6 部门销售下钻数据

	1995年			
部门	1季度	2季度	3季度	4季度
部门1	200	200	350	150
部门2	250	50	150	150
部门3	200	150	180	270

77

多维数据分析（9/12）

- 上钻操作（概化）
  - 例如，1995年各季度各部门销售收入表如下

表7.6 部门销售下钻数据

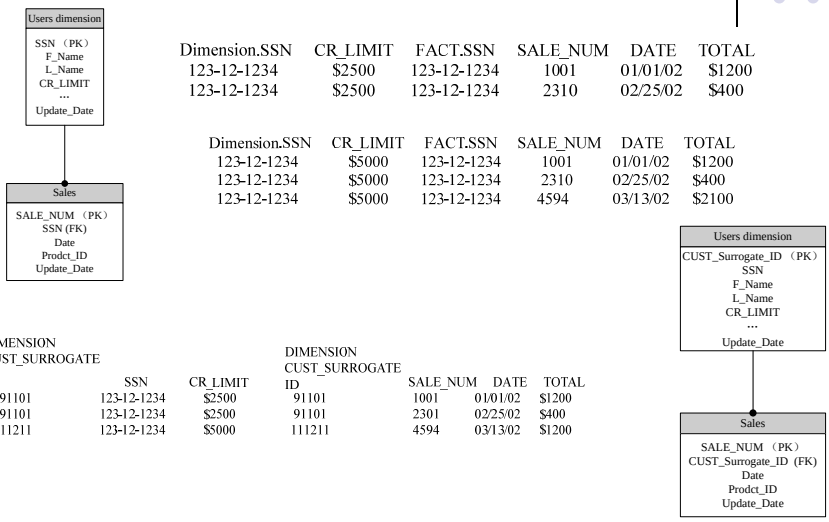
	1995年			
部门	1季度	2季度	3季度	4季度
部门1	200	200	350	150
部门2	250	50	150	150
部门3	200	150	180	270

78

2、给个例子，说明 DW 历史完整性问题，为什么不能用自然关键字

改例子，改表保证历史完整性

历史完整性



3、Apriori 算法

简答：

1、DW 定义，四大特色

数据仓库就是一个面向主题的、集成的、不可更新的、随时间不断变化的数据集合，用于支持经营管理过程中的决策制定。

问答：

DW 物理优化

物理模型设计

- 在逻辑模型设计基础上确定数据的存储结构、确定索引策略、确定存储分配及数据存放位置等与物理有关的内容，物理模型设计的具体方法与数据库设计中的大致相似。其目的是为了提高数据仓库系统的访问性能。常用的一些技术有：
  - 合并表
  - 建立数据序列
  - 引入冗余
  - 表的物理分割
  - 生成导出数据
  - 建立广义索引
- 规范化/反规范化

物理模型设计 – 表的物理分割（1/2）

- 类似于在逻辑设计阶段的数据分割
- 可以根据表中每个属性数据的访问频率和稳定性程度对表的存储结构进行分割
  - 对于访问频率较高的属性，可以单独考虑其物理存储组织，以便选择合适的索引策略和特定的物理组织方式。
  - 对于需要频繁更新的属性，也可以单独组织其物理存储，以免因数据更新而带来的空间重组、重构等工作。

54

物理模型设计 – 建立广义索引（1/2）

- 用于记录数据仓库中数据与 ‘最’ 有关的统计结果的索引被称为 ‘广义索引’ 。如：
  - 当月销售额最高的商店？
  - 当月销售情况最差的商品？
  - ……
- 这样的广义索引的数据量是非常小的，可以在每次进行数据仓库数据加载工作时生成或刷新这样的广义索引。用户可以从已经建立的广义索引里直接获取这些统计信息，而不必对整个数据仓库进行扫描。

58

一致性维度

事务处理环境不适宜 DSS 的原因

DW 刷新方法

数据挖掘：基于大量的完整的数据，结果是知识和有普遍意义规则

模式：数据库中数据与数据的逻辑关系

知识：满足用户对兴趣度和置信度的要求的模式

操作型处理

- 也叫事务处理，是指对数据库的日常联机访问操作，通常是对一个或一组记录的查询和修改，主要是为企业特定的应用服务的，所以也叫联机事务处理。
- On-Line Transaction Processing,（OLTP）
  - 通常仅仅是对一个或一组记录的查询或修改；
  - 查询简单，但执行频率高；
  - 人们关心的是处理的响应时间、数据的安全性和完整性等指标。

13

分析型处理

- 也叫做信息型处理，主要用于企业管理人员的决策分析，为制订企业的未来经营管理计划提供辅助决策信息。
  - 需要对大量的事务型数据进行统计、归纳和分析；
  - 需要访问大量的历史数据；
  - 执行频率和对响应时间的要求都不高。

-典型的的分析型处理

  - 决策支持系统 (DSS --Decision Support System)

元数据

- 关于数据的数据
  - 描述了数据的结构、内容、编码、索引等内容
- 通过元数据可以将数据仓库和复杂的数据源系统的变化隔离，是数据仓库开发和维护的一个关键因素，也是保证数据抽取质量的依据。
- 种类
  - 关于数据源的元数据
  - 关于数据模型的元数据
  - 关于数据仓库映射的元数据
  - 关于数据仓库使用的元数据

退化维度

- 维度表为空，具体的维度值直接存放在事实表中
  - 事务编号
  - 订单编号
  - 发票编号
  - 提货单编号
  - ……

POS 零售营销事务事　实
日期关键字（FK）
产品关键字（FK）
商场关键字（FK）
促销关键字（FK）
POS 事务编号
销售量
销售额
成本额
毛利润金额

维度的规范化处理（1/2）

规范化	非规范化
雪花模型	星型模型
复杂的表关系	简单的表关系
节省存储空间	记录之间存在数据冗余
连接的复杂，高开销	连接简单，低开销
低维度浏览能力	高维度浏览能力
不支持物理加速技术	支持物理加速技术