# 数据仓库

## 什么是数据仓库

数据仓库是一个面向主题的、集成的、时变的、非易失的数据集合，支持管理者的决策过程。

* **面向主题的：**数据仓库围绕一些重要主题，如顾客、供应商、产品和销售组织。数据仓库关注决策者的数据建模与分析，而不是单位的日常操作和事物处理。因此，数据仓库通常排除对于决策无用的数据，提供特定主题的简明视图。
* **集成的：**通常，构造数据仓库是将多个异构数据源，如关系数据库、一般文件和联机事物处理记录集成在一起。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。
* **时变的：**数据存储从历史的角度（例如，过去5~10年）提供信息。数据仓库中的关键结构都隐式或显式的包含时间元素。
* **非易失性的：**数据仓库总是物理的分离存放数据，这些数据源于操作环境下的应用数据。由于这种分离，数据仓库不需要事物处理、恢复和并发控制机制，通常，它只需要两种数据访问操作：数据的初始化装入和数据访问。

概言之，**数据仓库是一种语义上一致的数据存储，它充当决策支持数据模型的物理实现，并存放企业战略决策所需要的信息，数据仓库也常常被看做一种体系结构，通过将异构数据源中的数据集成在一起而构建，支持结构化和专门的查询、分析报告和决策制定。**

数据仓库的构建需要数据集成、数据清理和数据统一。数据仓库的应用常常需要一些决策支持技术。

从异构数据库集成的角度来看，数据仓库也是非常有用的。许多组织机构收集了形形色色的数据，并由多个异构的、自治的和分布的数据源维护大型数据库。集成这些数据，并提供简便、有效的访问是人们非常期望的，并且也是一种挑战。

**对于异构数据库的集成，传统的数据库做法是：在多个异构数据库上，建立一个包装程序和一个集成程序（或中介程序）。当查询在客户站点提交时，首先使用元数据字典对查询进行转换，将它转换成相应异构站点上的查询，然后，将这些查询映射和发送到局部查询处理器。由不同站点返回的结果被集成为全局回答**。这种查询驱动的方法需要复杂的信息过滤和集成处理，并且与局部数据源上的处理竞争资源。这种方法是低效的，并且对于频繁的查询，特别是需要聚集操作的查询，开销可能很大。

对于异构数据库集成的传统方法，数据仓库提供了一种有趣的替代方案。**数据仓库使用更新驱动的方法，而不是查询驱动的方法**。**这种方法将来自多个异构源的信息预先集成，并存储在数据仓库中，供直接查询和分析**。与联机事物处理数据库不同，数据仓库不包含最近的信息。然而，数据仓库为集成的异构数据库系统带来了高性能，因为数据被复制、预处理、集成、注释、汇总，并重新组织到一个语义一致的数据存储中，数据仓库的查询处理并不影响在局部数据源上进行的处理。此外，数据仓库可以存储并集成历史信息，支持复杂的多维查询。

## 数据库与数据仓库

联机操作数据库系统的主要任务时执行联机事务和查询处理。这种系统称作**联机事务处理（Online Transaction Processing, OLTP）系统**。它们涵盖了单位的大部分日常操作，如购物、库存、制造、银行、工资、注册、记账等。另一方面，数据仓库系统在数据分析和决策方面为用户或“知识工人”提供服务。这种系统可以用不同的格式组织和提供数据，以便满足不同用户的形形色色的需求，这种系统称作**联系分析处理（Online Analyical Processing, OLAP）系统**。

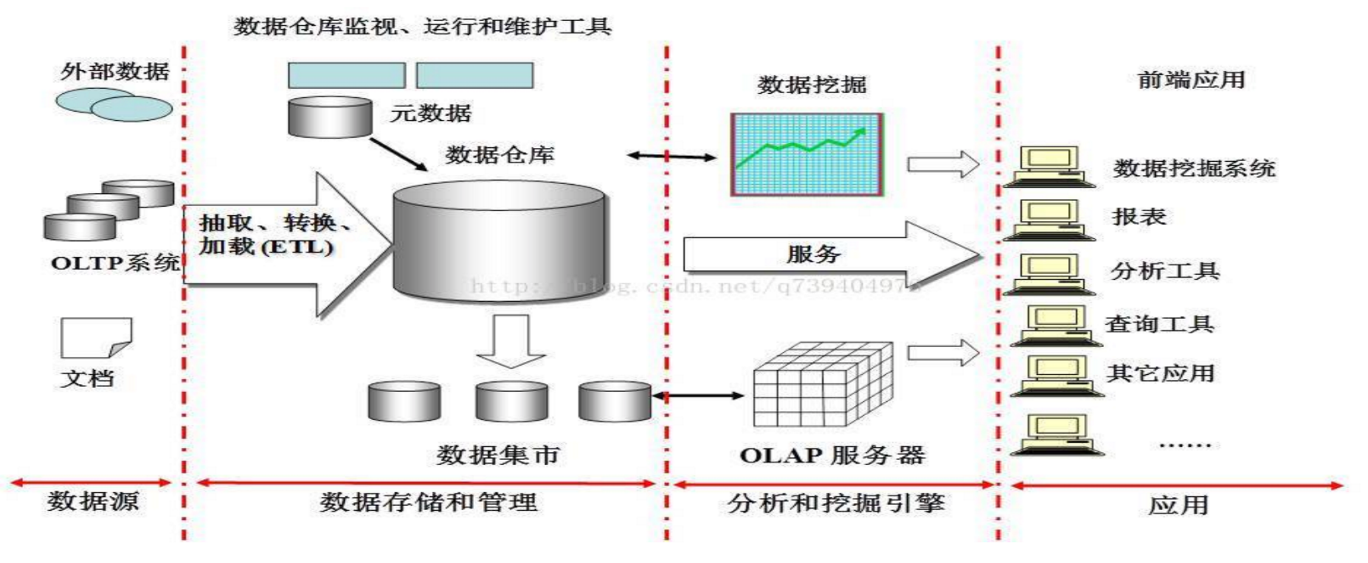
OLTP和OLAP的主要区别概述如下：

* **用户和系统的面向性：**OLTP是面向顾客的，用于技术专业人员的事务和查询处理。OLAP是面向市场的，用于知识工人（分析人员）的数据分析。
* **数据内容：**OLTP系统管理当前数据。OLAP系统管理大量历史数据，提供汇总和聚集机制。并在不同的粒度层上存储和管理信息。
* **数据库设计：**OLAP系统采用实体-联系（ER）数据模型和面向应用的数据库设计。而OLAP系统通常采用星行或雪花模型和面向主题的数据库设计。
* **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单位的数据。OLAP系统常常跨越数据库模式的多个版本，OLAP系统还处理来自不同单位的信息，以及由多个数据库集成的信息。由于数据量巨大，OLAP数据也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，这种系统需要并发控制和恢复机制。然而，对OLAP系统的访问大部分是只读操作，尽管许多是复杂的查询。



## 数据仓库体系结构

通常，数据仓库采用三层体系结构，如图所示：



1. 底层是仓库数据库服务器，它几乎总是一个关系数据库系统。使用后端工具和使用程序，由操作数据库或其它外部数据源提取数据，放入底层。这些工具和实用程序进行数据提取、清理和变换，以及装入和刷新，以更新数据仓库。这一层还包括元数据库，存放关于数据仓库和它的内容信息。
2. 中间层是OLAP服务器，其典型的实现使用（a）关系OLAP模型，（b）多维OLAP模型。
3. 顶层是前端客户层，它包括查询和报告工具、分析工具和数据挖掘工具。

## 数据仓库模型

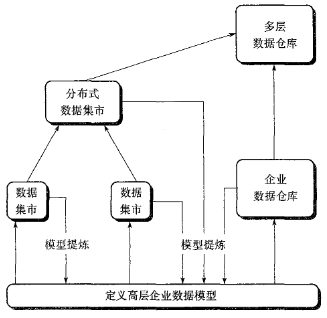
从结构的角度看，有三种数据仓库模型：企业仓库、数据集市和虚拟仓库。

**企业仓库：**企业仓库搜集了关于主题的所有信息，跨越整个企业。它提供企业范围内的数据集成，通常来自一个或多个操作数据库系统或外部信息提供者，并且是多功能的。

**数据集市：**数据集市包含企业范围数据的一个子集，对于特定的用户群是有用的，其范围限于选定的主题。例如，销售数据集市可能限定其主题为顾客、商品和销售。包括在数据集市中的数据通常是汇总的。数据集市的实现周期一般是以周计，而不是以月计或数以年计。

**虚拟仓库：**虚拟仓库是操作系统上视图的集合。为了有效的处理查询，只有一些可能的汇总视图被物化。

对于开发数据仓库系统，一种推荐的方法是以递增、进化的方式实现数据仓库，如图所示，首先，在一个合理短的时间内（如一两个月），定义一个高层次的企业数据模型，在不同的主题和可能的应用之间，提供企业范围的、一致的、集成的数据视图。这个高层模型将大大减少今后的集成问题，尽管在企业数据仓库和部门数据集市的开发中，他还需要进一步提炼。其次，基于上述相同的企业数据模型，可以并行的实现独立的数据集市和企业数据仓库。再次，可以通过中心服务器集成不同的数据集市，构造分布数据集市。最后，构造一个多层数据仓库，这里，企业仓库是所有仓库数据的唯一管理者，仓库数据分布在一些依赖的数据集市中。



## 数据提取、变换和装入

数据仓库系统使用后端工具和实用程序来加载和刷新它的数据，这些工具和实用程序包含以下功能：

* **数据提取：**由多个异构的外部数据源收集数据
* **数据清理：**检测数据中的错误，可能时订正它们
* **数据变换**：将数据由遗产或宿主格式转换成数据仓库格式
* **装入：**排序、汇总、合并、计算视图、检查完整性，并建立索引和划分
* **刷新：**传播由数据源到数据仓库的更新。

## 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，元数据库应当包括以下内容：

* 数据仓库结构的描述，包括仓库模式、视图、维、分层结构、导出数据的定义，以及数据集市的位置和内容。
* 操作元数据，包括数据血统（迁移数据的历史和它所使用的变换序列）、数据流通（主动的、档案的或净化的）和管理信息（仓库使用的统计量、错误报告和审计跟踪）。
* 用于汇总的算法，包括度量和维定义算法，数据所处的粒度、划分、主题领域、聚集、汇总、预定义的查询和报告。
* 由操作环境到数据仓库的映射，包括源数据库和它们的内容，信关描述、数据划分、数据提取、清理、转换规则和默认值，数据刷新和净化规则，以及安全性（用户授权和存取控制）
* 关于系统性能的数据，除刷新、更新和复制周期的定时和调度的规则外，还包括改善数据存取和检索性能的索引和概要。
* 商务元数据，包括商务术语和定义，数据拥有者信息和收费策略。

数据仓库包含不同的汇总层，元数据是其中一种类型，其他类型包括当前的细节数据（几乎总是在磁盘上）、老的细节数据（通常在三级存储器上）、稍加汇总的数据和高度汇总的数据（可以，也可以不物理的存入仓库）。

与数据仓库中的其他数据相比，元数据扮演很不相同的角色，并且由于种种原因，它也是重要的角色，例如，元数据用作目录，帮助决策支持系统分析者对数据仓库的内容定位；当数据由操作环境到数据仓库环境转换时，作为数据映射的指南；对于汇总的算法将当前细节数据汇总成稍加综合的数据，或将稍加综合的数据汇总成高度综合的数据，它也是指南。元数据应当持久存放和管理（即存放在磁盘上）。

# 数据仓库建模

数据仓库建模即是创建数据立方体，数据立方体是一种多维数据模型，数据仓库和OLAP工具均是基于多维数据模型。

## 数据立方体

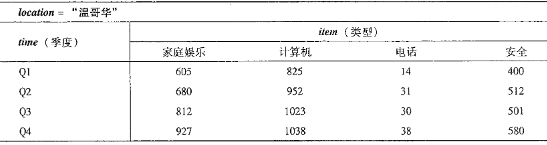
数据立方体允许以多维对数据建模和观察。它由维和事实定义。

**维是一个透视或实体**。例如，创建一个数据仓库sales，记录商店的销售，涉及维time、item、branch和location。这些维使得商店能够记录商品的月销售、销售商品的分店和地点。每个维都可以有一个与之相关的表。该表称为维表，他进一步描述维，例如，item的维表可以包含属性item\_name、brand和type。

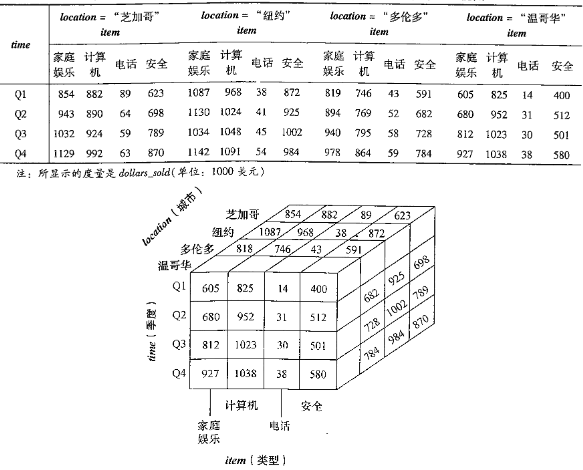
**通常，多维数据模型围绕诸如销售这样的中心主题组织，主题用事实表表示**。**事实是数值度量的。把它们看作数量，是因为我们想根据它们分析维之间的联系**。例如，数据仓库sales的事实包括dollars\_sold（销售额）、units\_sold（销售量）和amount\_budgeted（预算额）。事实表包括事实的名称或度量，以及每个相关维表的码。

尽管我们经常把数据立方体看作3-D几何结构，但是在数据仓库中，数据立方体是n维的。

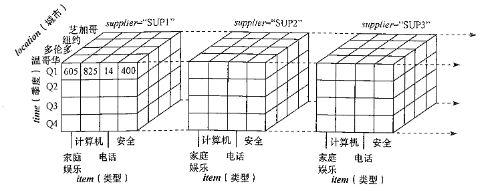
销售数据的time和item维的2-D视图：



销售数据的time、item和location维的3-D视图：

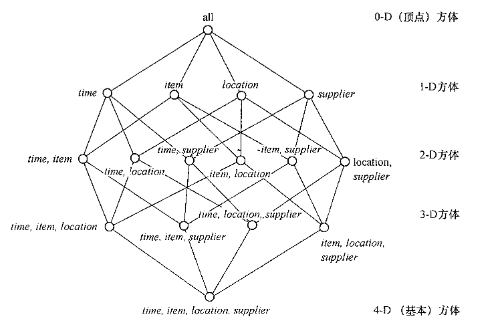


销售数据的time、item、location和supplier维的4-D视图：



如果我们按这种方法继续下去，则我们可以把任意n维数据立方体显示成（n-1）维“立方体”的序列。

上图所示的数据立方体称作方体。给定维的集合，我们可以对给定诸维的每个可能的子集产生一个方体。结果形成方体的格，每个方体在不同的汇总级显示group by数据。方体的格称作数据立方体。下图显示time、item、location和supplier的数据立方体的方体格。



存放最低层汇总的方体称作基本方体，0-D方体存放最高层的汇总，称作顶点方体。

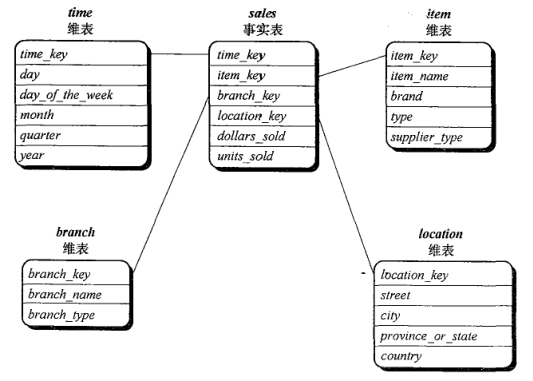
## 多维数据模型的模式

最流行的数据仓库的数据模型是多维数据模型，这种模型可以是星型模式、雪花模式或事实星座模式。

### 星型

星型模式：其中数据仓库包括（1）一个大的中心表（事实表），它包含大批数据并且不含冗余；（2）一组小的附属表（维表），每维一个。这种模式图很像星光四射，维表显示在围绕中心表的射线上。

销售的星型模式显示在图中，从四个维time、item、branch和location考虑销售。该模式包含一个中心事实表sales，它包含四个维的码和两个度量dollars\_sold和units\_sold。

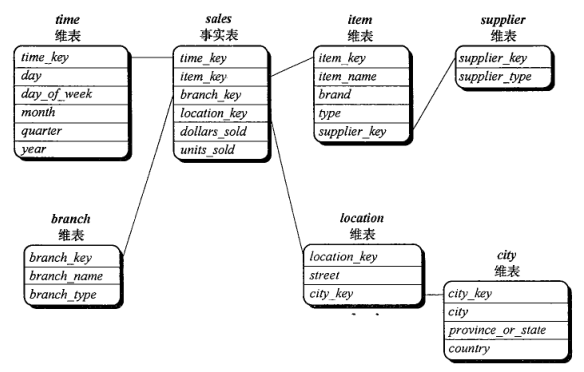


### 雪花

**雪花模式：**雪花模式是星型模式的变种，其中某些维表被规范化，因而把数据进一步分解到附加的表中，结果模式图形成类似于雪花的形状。

**雪花模式和星型模式的主要不同在于，雪花模式的维表可能是规范化形式，以便减少冗余。**这种表易于维护，并节省存储空间。然而，与典型的巨大事实表相比，这种空间的节省可以忽略，此外，由于执行查询需要更多的连接操作，雪花结构可能降低浏览的效率。因此，尽管雪花模式减少了冗余，但是**在数据仓库设计中，雪花模式不如星型模式流行**。

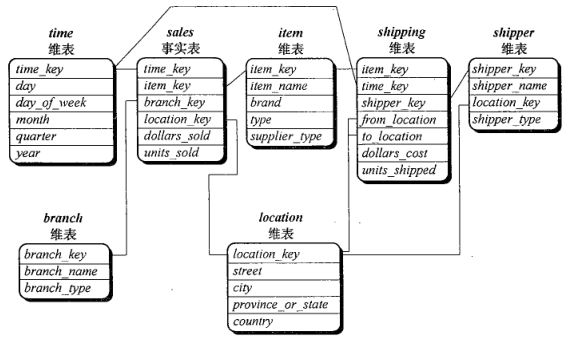
雪花模式如图所示，这里，事实表sales与上图的星型模式相同，两个模式的主要差别是维表。星型模式中item的单个维表在雪花模式中被规范化。导致新的item表和supplier表。类似的，星型模式中单个维表location也被规范化成两个新表：location和city。



### 事实星座

事实星座：复杂的应用可能需要多个事实表共享维表，这种模式可以看做星型模式的汇集，因此称作星系模式或事实星座。

一个事实星座模式的例子如图所示，该模式说明了两个事实表，sales和shipping。Sales表的定义与星型模式相同。Shipping表有五个维或码——item\_key, time\_key, shipper\_key, from\_location和to\_location,两个度量——dollars\_cost和units\_shipped。事实星座模式允许事实表共享维表。例如，事实表sales和shipping共享维表time,item和location。



在建立数据仓库时，数据仓库和数据集市之间是由区别的，数据仓库收集了关于整个组织的主题（如顾客、商品、销售、资产和员工）信息，因此是企业范围的。**对于数据仓库，通常使用事实星座模式，因为它能对多个相关的主题建模**。另一方面，数据集市是数据仓库的一个部门子集，它针对选定的主题，因此是部门范围的。**对于数据集市，流行采用星型模式或雪花模式，因为它们都适合对单个主题建模**，尽管星型模式更流行更有效。

## 概念分层

概念分层定义一个映射序列，将低层概念集映射到叫高层，更一般的概念。考虑维location的概念分层。

## 度量分类

“如何计算度量？”，我们首先研究如何对度量分类。注意，数据立方体空间的多维点可以用维-值对的集合来定义，例如，<time=“Q1”，location=“温哥华”，item=“计算机”>。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空间的每个点求值，通过对给定点的各维-值对聚集数据，计算该点的度量值。

度量根据其所用的聚集函数可以分成三类：分布的、代数的和整体的。

分布的：一个聚集函数如果能用如下方式进行计算，则它是分布的。假设数据被划分为n个集合，将函数用于每一部分，得到n个聚集值。如果将函数用于n个聚集值得到的结果与将函数用于整个数据集（不划分）得到的结果一样，则该函数可以用分布方式计算。例如sum()/count()

代数的：一个聚集函数如果能够用一个具有M个参数的代数函数计算（其中M是有界正整数），而每个参数都可以用一个分布式聚集函数求得，则它是代数的。例如avg()/min\_N()/max\_N()

整体的：一个聚集函数如果描述它的子聚集所需的存储没有一个常数界，则它是整体的，例如median()/mode()/和rank()。

大部分数据立方体应用需要有效的计算分布的和代数的度量，对此存在许多有效的技术，相比之下，有效的计算整体度量是比较困难的。然而，对于某些整体函数的近似计算，有效的技术是存在的。

## OLAP操作

“在OLAP中，如何使用概念分层？”在多维数据模型中，数据组织在多维空间，每维包含由概念分层定义的多个抽象层。这种组织为用户从不同角度观察数据提供了灵活性，有一些OLAP数据立方体操作用来物化这些不同视图，允许交互查询和分析手头数据。

**上卷：**通过沿一个维的概念分层向上攀升或者通过维规约在数据立方体上进行聚集。

**下钻：**下钻是上卷的逆操作，下钻可以通过沿维的概念分层向下或引入附加的维来实现。

**切片和切块：**切片操作在给定的立方体的一个维上进行选择，导致一个子立方体。

**转轴：**是一种目视操作，它转动数据的视角，提供数据的替代表示。

**其他OLAP操作：**有些OLAP系统还提供其他钻取操作。例如，**钻过**执行涉及多个事实表的查询。**钻透**操作使用关系SQL机制，钻透到数据立方体的底层，到后端关系表。

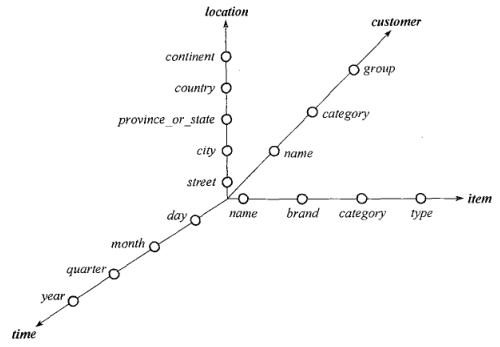
其他OLAP操作可能包括列出表中最高或最低的N项，以及计算移动平均值、增长率、利润、内部返回率、贬值、流通转换和统计功能。

OLAP提供了分析建模机制，包括推导比率、方差等以及计算多个维上度量的计算引擎。它能在每一粒度和所有维的交上产生汇总、聚集和分层。OLAP也支持预测、趋势分析和统计分析函数模型。

## 星网查询模型

多维数据库查询可以基于星网模型，星网模型由从中心点发出的射线组成，其中每一条射线代表一个维的概念分层。概念分层上的每个“抽象级”称为一个足迹，代表诸如上卷、下钻等OLAP操作可用的粒度。

数据仓库的一个星网查询模型显示如图所示。该星网由四条射线组成，分别代表维location、customer、item和time的概念分层。每条线由一些足迹组成，代表该维的抽象级。例如，time线有4个足迹：day、month、quarter和year。一个概念分层可以涉及单个属性，或若干属性，为了考察商品销售，用户可以沿着time维上卷，由month到quarter，或沿着location维下钻，由country到city。



通过用较高层抽象值替换低层抽象值，概念分层可以用于泛化数据。通过用低层抽象值替换高层抽象值，概念分层也可以特殊化数据。

# 数据仓库的设计与使用

## 数据仓库的分析框架

关于数据仓库的设计，必须考虑四种不同的视图：自顶向下视图、数据源视图、数据仓库视图和商务查询视图。

* **自顶向下视图**使得我们可以选择数据仓库所需的相关信息。这些信息能够满足当前和未来的商务需求
* **数据源视图**揭示被操作数据库系统收集、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记录，存放在个别数据源表或集成的数据源表中。
* **数据仓库视图**包括事实表和维表。他们提供存放在数据仓库内的信息，包括预计算的总和与计数，以及提供历史背景的关于源、日期和时间等信息。
* **商务查询视图**是从最终用户的角度透视数据仓库中的数据。

建立和使用数据仓库是一项复杂的任务，因为它需要**商务技巧、技术技巧和计划管理技巧**。关于商务技巧，建立数据仓库涉及理解这样的系统如何存储和管理他们的数据；如何构造一个提取程序，将数据由操作数据库转换到数据仓库；如何构造一个仓库刷新软件，合理地保持数据仓库中的数据相对于操作数据库中数据的当前性。使用数据仓库涉及理解它所包含的数据的含义，以及理解商务需求并将它转换成数据仓库查询。

关于技术技巧，数据分析者需要理解如何由定量信息作出估价，以及如何根据数据仓库中的历史信息得到的结论推导事实。**这些技巧包括发现模式和趋势，根据历史推断趋势和发现异常或模式漂移的能力，并根据这种分体提出条理清晰的管理建议**。最后，**计划管理技巧涉及需要与许多技术人员、经销商和最终用户沟通，以便以及时和讲求效益的方式提交结果**。

## 数据仓库的设计过程

数据仓库可以使用自顶向下方法、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。当技术成熟并且已经掌握，对必须解决的商务问题清楚并且已经很好理解时，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。这样可以以相当低的代价推进，在做出重要承诺之前评估技术带来的利益。在混合方法下，一个组织既能利用自顶向下方法的规划性和战略性的特点，又能保持自底向上方法一样快速实现和立即应用。**

从软件工程的角度来看，数据仓库的设计和构造包含以下步骤：**规划、需求研究、问题分析、仓库设计、数据集成和测试，最后，部署数据仓库**。大型软件系统可以用两种方法开发：瀑布式方法和螺旋式方法。

一般而言，数据仓库的设计过程包含如下步骤：

1. 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，应当选用数据仓库模型，然而，如果处理是部门的，并关注某一类商务处理的分析，则应该选择数据集市。
2. 选取商务处理的粒度。对于处理，该粒度是基本的，在事实表中是数据的原子级。
3. 选取用于每个事实表记录的维。
4. 选取将安放在每个事实表记录中的度量。

**一旦涉及和构造好数据仓库，数据仓库的最初部署就包括初始化安装、首次展示规划、培训和熟悉情况。平台的升级和维护也要考虑。数据仓库管理包括数据刷新、数据源同步、规划灾难恢复、管理存取控制和安全、管理数据增长、管理数据库性能以及数据仓库的增强和扩充。范围管理包括控制查询、维、报告的数量和范围，限制数据仓库的大小，或限制进度、预算和资源。**

## 数据仓库分层

### 什么是数据仓库分层

分层是数据仓库解决方案中，数据架构设计的一种数据逻辑结构，通过分层理念建立的数据仓库，它的可扩展性非常好，这样设计出来的模型架构，可以任意的增减、替换数据仓库中的各个组成部分。

### 数据仓库分层的原因

清晰数据结构：每一个数据分层都有它的作用域和职责，在使用表的时候能更方便地定位和理解

减少重复开发：规范数据分层，开发一些通用的中间层数据，能够减少极大的重复计算

统一数据口径：通过数据分层，提供统一的数据出口，统一对外输出的数据口径

复杂问题简单化：将一个复杂的任务分解成多个步骤来完成，每一层解决特定的问题

### 数据仓库具体的分层

标准的数据仓库分层：ODS（操作数据存储层），DW（数据仓库层），MID（数据集市层），APP（应用层）。

**ODS：操作数据存储层**，是最接近数据源中数据的一层，数据源中的数据，经过抽取、洗净、传输，也就说传说中的 ETL 之后，装入本层。本层的数据，总体上大多是按照源头业务系统的分类方式而分类的。

一般来讲，为了考虑后续可能需要追溯数据问题，因此对于这一层就不建议做过多的数据清洗工作，原封不动地接入原始数据即可，至于数据的去噪、去重、异常值处理等过程可以放在后面的DWD层来做。

**DW：数据仓库层**，它的数据是干净的数据，是一致的准确的，也就是清洗后的数据，它的数据一般都遵循数据库第三范式，数据粒度和ods的粒度相同，它会保存bi系统中所有历史数据，DW层又细分为DWD（Data Warehouse Detail）层、DWM（Data Warehouse Middle）层和DWS（Data Warehouse Service）层。

1. 数据明细层：DWD（Data Warehouse Detail）

该层一般保持和ODS层一样的数据粒度，并且提供一定的数据质量保证。同时，为了提高数据明细层的易用性，该层会采用一些维度退化手法，将维度退化至事实表中，减少事实表和维表的关联。

另外，在该层也会做一部分的数据聚合，将相同主题的数据汇集到一张表中，提高数据的可用性，后文会举例说明。

2. 数据中间层：DWM（Data WareHouse Middle）

该层会在DWD层的数据基础上，对数据做轻度的聚合操作，生成一系列的中间表，提升公共指标的复用性，减少重复加工。

直观来讲，就是对通用的核心维度进行聚合操作，算出相应的统计指标。

3. 数据服务层：DWS（Data WareHouse Servce）

又称数据集市或宽表。按照业务划分，如流量、订单、用户等，生成字段比较多的宽表，用于提供后续的业务查询，OLAP分析，数据分发等。

一般来讲，该层的数据表会相对比较少，一张表会涵盖比较多的业务内容，由于其字段较多，因此一般也会称该层的表为宽表。

在实际计算中，如果直接从DWD或者ODS计算出宽表的统计指标，会存在计算量太大并且维度太少的问题，因此一般的做法是，在DWM层先计算出多个小的中间表，然后再拼接成一张DWS的宽表。由于宽和窄的界限不易界定，也可以去掉DWM这一层，只留DWS层，将所有的数据在放在DWS亦可。

**MID：数据集市层**，它是面向主题组织数据的，通常是星状和雪花状数据，从数据粒度来讲，它是轻度汇总级别的数据，已经不存在明细的数据了，从广度来说，它包含了所有业务数量。从分析角度讲，大概就是近几年

**APP：应用层**，数据粒度高度汇总，但不一定涵盖所有业务数据，只是mid层数据的一个子集。

### 数据分层实例

如下图，可以认为是一个电商网站的数据体系设计。我们暂且只关注用户访问日志这一部分数据。

在ODS层中，由于各端的开发团队不同或者各种其它问题，用户的访问日志被分成了好几张表上报到了我们的ODS层。

为了方便大家的使用，我们在DWD层做了一张用户访问行为天表，在这里，我们将PC网页、H5、小程序和原生APP访问日志汇聚到一张表里面，统一字段名，提升数据质量，这样就有了一张可供大家方便使用的明细表了。

在DWM层，我们会从DWD层中选取业务关注的核心维度来做聚合操作，比如只保留人、商品、设备和页面区域维度。类似的，我们这样做了很多个DWM的中间表

然后在DWS层，我们将一个人在整个网站中的行为数据放到一张表中，这就是我们的宽表了，有了这张表，就可以快速满足大部分的通用型业务需求了。

最后，在APP应用层，根据需求从DWS层的一张或者多张表取出数据拼接成一张应用表即可。

备注：例子只是为了简单地说明每一层的作用，并不是最合理的解决方案，大家辩证地看待即可。



### 技术实践

数据层的存储一般如下：

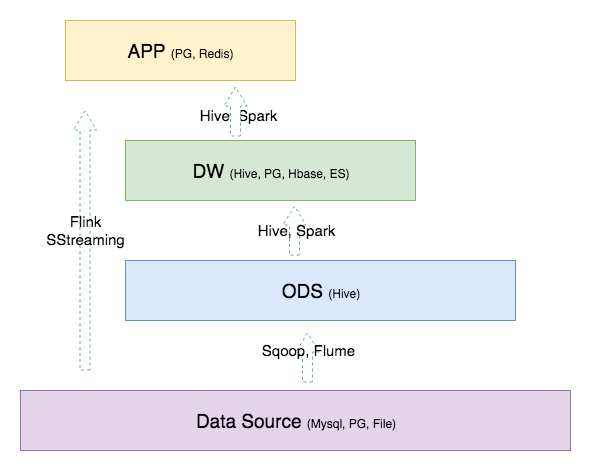
Data Source：数据源一般是业务库和埋点，当然也会有第三方购买数据等多种数据来源方式。业务库的存储一般是Mysql 和 PostgreSql。

ODS 层：ODS 的数据量一般非常大，所以大多数公司会选择存在HDFS上，即Hive或者Hbase，Hive居多。

DW 层：一般和 ODS 的存储一致，但是为了满足更多的需求，也会有存放在 PG 和 ES 中的情况。

APP 层：应用层的数据，一般都要求比较快的响应速度，因此一般是放在 Mysql、PG、Redis中。

计算引擎的话，可以简单参考图中所列就行。目前大数据相关的技术更新迭代比较快，本节所列仅为简单参考。



## 数据仓库用于信息处理

数据仓库工具可以分为访问与检索工具、数据库报表工具、数据分析工具和数据挖掘工具。

有三类数据仓库应用：信息处理、分析处理和数据挖掘。

* **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。数据仓库信息处理的当前趋势是构造低价格的基于Web的访问工具，然后与Web浏览器集成在一起。
* **分析处理**支持基本的OLAP操作，包括切片与切块、下钻、上卷和转轴。
* **数据挖掘**支持知识发现，包括找出隐藏的模式和关联，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。

## 从联机分析处理到多维数据挖掘

数据挖掘领域已经对各种类型的数据的挖掘做了大量研究，这些数据类型包括**关系数据、数据仓库的数据、事务数据、时间序列数据、空间数据、文本数据和一般文件**。**多维数据挖掘把数据挖掘与OLAP集成在一起，在多维数据库中发现知识**。在数据挖掘的许多不同范例和结构中，由于以下原因，多维数据挖掘特别重要。

* 数据仓库中数据的高质量
* 环绕数据仓库的信息处理基础设施
* 基于OLAP的多维数据探索
* 数据挖掘功能的联机选择

# 数据仓库的实现

数据仓库包含海量数据。OLAP服务器要在数秒之内回答决策支持查询。因此，至关重要的是，**数据仓库系统要支持高校的数据立方体计算技术、存取技术和查询处理技术**。

## 数据立方体的有效计算：概述

多维数据分析的核心是有效地计算许多维集合上的聚集。用SQL的术语，这些聚集称为分组（group-by）。每个分组可以用一个方体表示，而分组的集合形成定义数据立方体的方体的格。下面，考察与数据立方体有效计算相关的问题。

1. compute cube操作与维灾难
2. 部分物化：方体的选择计算

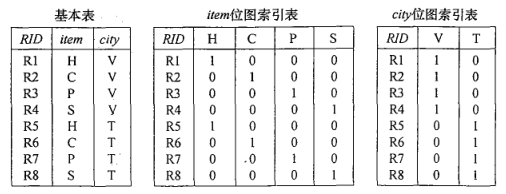
给定基本方体，方体的物化有三种选择

1. 不物化
2. 完全物化
3. 部分物化

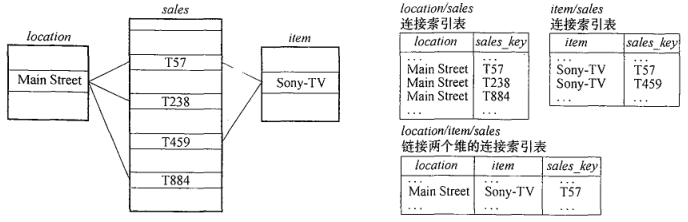
## 索引OLAP数据：位图索引和连接索引

为了提供有效的数据访问，大部分数据仓库系统支持索引结构和物化视图（使用方体）。下面考察如何使用位图索引和连接索引对OLAP数据进行索引。

**位图索引**是record\_ID（RID）列表的一种替代表示。在给定属性的位图索引中，属性域中的每个值v，有一个不同的位向量Bv，如果给定的属性域包含n个值，则位图索引中每项需要n个位（即n位向量）。如果数据表给定行上改属性值为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。



连接索引方法的流行源于它在关系数据库查询处理方面的应用。传统的索引将给定列上的映射到具有该值的行的列表上，与之相反，连接索引登记来自关系数据库的两个关系的可连接行。例如，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（RID，SID）对，其中RID和SID分别来自关系R和S的记录标识符。



## OLAP查询的有效处理

物化方体和构造OLAP索引结构的目的是加快数据立方体查询处理的速度，给定物化的视图，查询处理应按如下步骤进行：

1. 确定哪些操作应当在可利用的方体上执行
2. 确定相关操作应当使用哪些物化的方体