第 12 章 数据仓库

12.1 什么是数据仓库

数据仓库，英文名称为 Data Warehouse，可简写为 DW 或 DWH。数据仓库，是为企业所有级别的决策制定过程，提供所有类型数据支持的战略集合。它出于分析性报告和决策支持目的而创建。为需要业务智能的企业，提供指导业务流程改进、监视时间、成本、质量以及控制。

12.2 数据仓库能干什么？

1）年度销售目标的指定，需要根据以往的历史报表进行决策，不能拍脑袋。

2）如何优化业务流程

例如：一个电商网站订单的完成包括：浏览、下单、支付、物流，其中物流环节可能和中通、申通、韵达等快递公司合作。快递公司每派送一个订单，都会有订单派送的确认时间，可以根据订单派送时间来分析哪个快递公司比较快捷高效，从而选择与哪些快递公司合作，剔除哪些快递公司，增加用户友好型。

12.3 数据仓库的特点

1）数据仓库的数据是面向主题的

与传统数据库面向应用进行数据组织的特点相对应，数据仓库中的数据是面向主题进行组织的。什么是主题呢？首先，主题是一个抽象的概念，是较高层次上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，它是对应企业中某一宏观分析领域所涉及的分析对象。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的一个完整、一致的描述，能完整、统一地刻划各个分析对象所涉及的企业的各项数据，以及数据之间的联系。所谓较高层次是相对面向应用的数据组织方式而言的，是指按照主题进行数据组织的方式具有更高的数据抽象级别。

2）数据仓库的数据是集成的

数据仓库的数据是从原有的分散的数据库数据抽取来的。操作型数据与 DSS 分析型数据之间差别甚大。第一，数据仓库的每一个主题所对应的源数据在原有的各分散数据库中有许多重复和不一致的地方，且来源于不同的联机系统的数据都和不同的应用逻辑捆绑在一起；第二，数据仓库中的综合数据不能从原有的数据库系统直接得到。因此在数据进入数据仓库之前，必然要经过统一与综合，这一步是数据仓库建设中最关键、最复杂的一

步，所要完成的工作有：

（1）要统一源数据中所有矛盾之处，如字段的同名异义、异名同义、单位不统一、字长不一致等。

（2)进行数据综合和计算。数据仓库中的数据综合工作可以在从原有数据库抽取 数据时生成，但许多是在数据仓库内部生成的，即进入数据仓库以后进行综合生成的。

3）数据仓库的数据是不可更新的

数据仓库的数据主要供企业决策分析之用，所涉及的数据操作主要是数据查询，一般情况下并不进行修改操作。数据仓库的数据反映的是一段相当长的时间内历史数据的内容，是不同时点的数据库快照的集合，以及基于这些快照进行统计、综合和重组的导出数据，而不是联机处理的数据。数据库中进行联机处理的数据经过集成输入到数据仓库中，一旦数据仓库存放的数据已经超过数据仓库的数据存储期限，这些数据将从当前的数据仓库中删去。因为数据仓库只进行数据查询操作，所以数据仓库管理系统相比数据库管理系统而言要简单得多。数据库管理系统中许多技术难点，如完整性保护、并发控制等等，在数据仓库的管理中几乎可以省去。但是由于数据仓库的查询数据量往往很大，所以就对数据查询提出了更高的要求，它要求采用各种复杂的索引技术；同时由于数据仓库面向的是商业企业的高层管理者，他们会对数据查询的界面友好性和数据表示提出更高的要求。

4）数据仓库的数据是随时间不断变化的

数据仓库中的数据不可更新是针对应用来说的，也就是说，数据仓库的用户进行分析处理时是不进行数据更新操作的。但并不是说，在从数据集成输入数据仓库开始到最终被删除的整个数据生存周期中，所有的数据仓库数据都是永远不变的。

数据仓库的数据是随时间的变化而不断变化的，这是数据仓库数据的第四个特征。这一特征表现在以下 3 方面：

（1）数据仓库随时间变化不断增加新的数据内容。数据仓库系统必须不断捕捉 OLTP 数据库中变化的数据，追加到数据仓库中去，也就是要不断地生成 OLTP 数据库的快照，经统一集成后增加到数据仓库中去；但对于确实不再变化的数据库快照，如果捕捉到新的变化数据，则只生成一个新的数据库快照增加进去，而不会对原有的数据库快照进行修改。

（2）数据仓库随时间变化不断删去旧的数据内容。数据仓库的数据也有存储期限，一旦超过了这一期限，过期数据就要被删除。只是数据仓库内的数据时限要远远长于操作型

环境中的数据时限。在操作型环境中一般只保存有 60~90 天的数据，而在数据仓库中则需要保存较长时限的数据（如 5~10 年），以适应 DSS 进行趋势分析的要求。

（3）数据仓库中包含有大量的综合数据，这些综合数据中很多跟时间有关，如数据经常按照时间段进行综合，或隔一定的时间片进行抽样等等。这些数据要随着时间的变化不断地进行重新综合。因此，数据仓库的数据特征都包含时间项，以标明数据的历史时期。

11.4 数据仓库发展历程

数据仓库的发展大致经历了这样的三个过程：

1）简单报表阶段：这个阶段，系统的主要目标是解决一些日常的工作中业务人员需要的报表，以及生成一些简单的能够帮助领导进行决策所需要的汇总数据。这个阶段的大部分表现形式为数据库和前端报表工具。

2）数据集市阶段：这个阶段，主要是根据某个业务部门的需要，进行一定的数据的采集，整理，按照业务人员的需要，进行多维报表的展现，能够提供对特定业务指导的数据，并且能够提供特定的领导决策数据。

3）数据仓库阶段：这个阶段，主要是按照一定的数据模型，对整个企业的数据进行采集，整理，并且能够按照各个业务部门的需要，提供跨部门的，完全一致的业务报表数据，能够通过数据仓库生成对对业务具有指导性的数据，同时，为领导决策提供全面的数据支持。

通过数据仓库建设的发展阶段，我们能够看出，数据仓库的建设和数据集市的建设的重要区别就在于数据模型的支持。因此，数据模型的建设，对于我们数据仓库的建设，有着决定性的意义。

11.5 数据库与数据仓库的区别

了解数据库与数据仓库的区别之前，首先掌握三个概念。数据库软件、数据库、数据仓库。

数据库软件：是一种软件，可以看得见，可以操作。用来实现数据库逻辑功能。属于物理层。

数据库：是一种逻辑概念，用来存放数据的仓库。通过数据库软件来实现。数据库由很多表组成，表是二维的，一张表里可以有很多字段。字段一字排开，对应的数据就一行一行写入表中。数据库的表，在于能够用二维表现多维关系。目前市面上流行的数据库都是二维数据库。如：Oracle、DB2、MySQL、Sybase、MS SQL Server 等。

数据仓库：是数据库概念的升级。从逻辑上理解，数据库和数据仓库没有区别，都是通过数据库软件实现的存放数据的地方，只不过从数据量来说，数据仓库要比数据库更庞大得多。数据仓库主要用于数据挖掘和数据分析，辅助领导做决策。

在 IT 的架构体系中，数据库是必须存在的。必须要有地方存放数据。比如现在的网购，淘宝，京东等等。物品的存货数量，货品的价格，用户的账户余额之类的。这些数据都是存放在后台数据库中。或者最简单理解，我们现在微博，QQ 等账户的用户名和密码。在后台数据库必然有一张 user 表，字段起码有两个，即用户名和密码，然后我们的数据就一行一行的存在表上面。当我们登录的时候，我们填写了用户名和密码，这些数据就会被传回到后台去，去跟表上面的数据匹配，匹配成功了，你就能登录了。匹配不成功就会报错说密码错误或者没有此用户名等。这个就是数据库，数据库在生产环境就是用来干活的。凡是跟业务应用挂钩的，我们都使用数据库。

数据仓库则是 BI 下的其中一种技术。由于数据库是跟业务应用挂钩的，所以一个数据库不可能装下一家公司的所有数据。数据库的表设计往往是针对某一个应用进行设计的。比如刚才那个登录的功能，这张 user 表上就只有这两个字段，没有别的字段了。但是这张表符合应用，没有问题。但是这张表不符合分析。比如我想知道在哪个时间段，用户登录的量最多？哪个用户一年购物最多？诸如此类的指标。那就要重新设计数据库的表结构了。对于数据分析和数据挖掘，我们引入数据仓库概念。数据仓库的表结构是依照分析需求，分析维度，分析指标进行设计的。

数据库与数据仓库的区别实际讲的是 OLTP 与 OLAP 的区别。

操作型处理，叫联机事务处理 OLTP（On-Line Transaction Processing），也可以称面向交易的处理系统，它是针对具体业务在数据库联机的日常操作，通常对少数记录进行查询、修改。用户较为关心操作的响应时间、数据的安全性、完整性和并发支持的用户数等问题。传统的数据库系统作为数据管理的主要手段，主要用于操作型处理。

分析型处理，叫联机分析处理 OLAP（On-Line Analytical Processing）一般针对某些主题的历史数据进行分析，支持管理决策。

表 操作型处理与分析型处理的比较

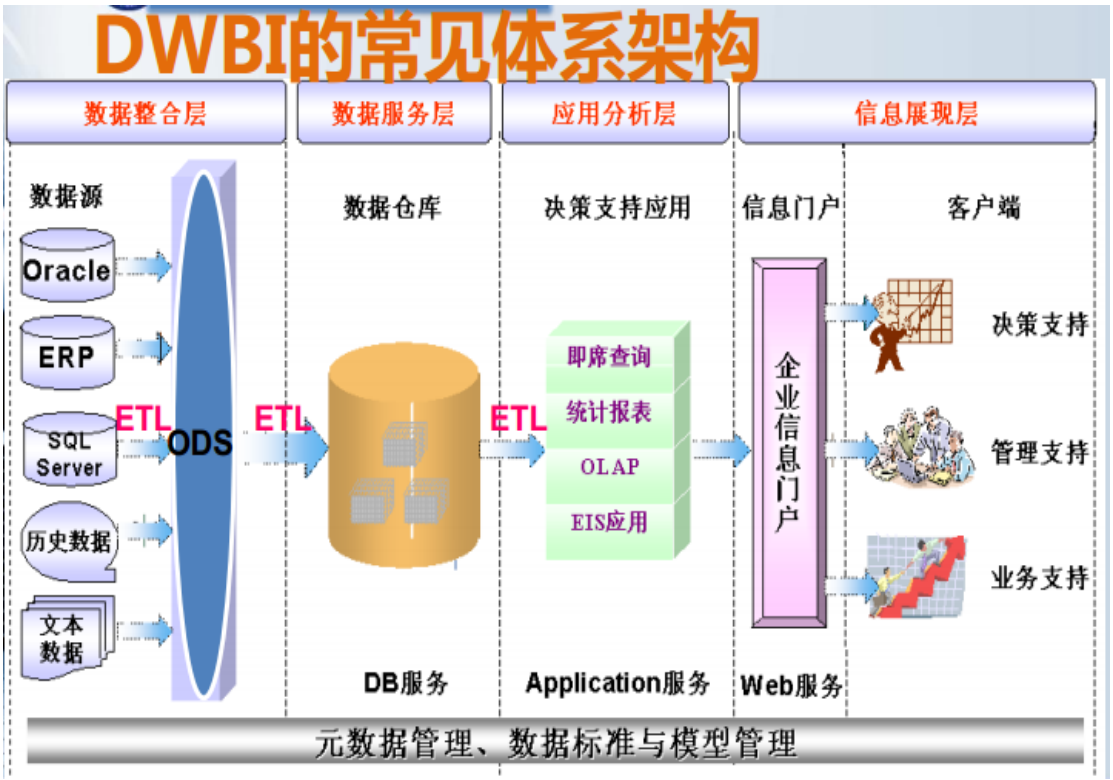
操作型处理	分析型处理
细节的	综合的或提炼的
实体——关系（E-R）模型	星型模型或雪花模型

存取瞬间数据	存储历史数据，不包含最近的数据
可更新的	只读、只追加
一次操作一个单元	一次操作一个集合
性能要求高，响应时间短	性能要求宽松
面向事务	面向分析
一次操作数据量小	一次操作数据量大
支持日常操作	支持决策需求
数据量小	数据量大
客户订单、库存水平和银行账户查询等	客户收益分析、市场细分等

11.6 数据仓库架构分层

11.6.1 数据仓库架构

数据仓库标准上可以分为四层：ODS（临时存储层）、PDW（数据仓库层）、DM（数据集市层）、APP（应用层）。



1）ODS 层：

为临时存储层，是接口数据的临时存储区域，为后一步的数据处理做准备。一般来说 ODS 层的数据和源系统的数据是同构的，主要目的是简化后续数据加工处理的工作。从数

据粒度上来说 ODS 层的数据粒度是最细的。ODS 层的表通常包括两类，一个用于存储当前需要加载的数据，一个用于存储处理完后的历史数据。历史数据一般保存 3-6 个月后需要清除，以节省空间。但不同的项目要区别对待，如果源系统的数据量不大，可以保留更长的时间，甚至全量保存；

2）PDW 层：

为数据仓库层，PDW 层的数据应该是一致的、准确的、干净的数据，即对源系统数据进行了清洗（去除了杂质）后的数据。这一层的数据一般是遵循数据库第三范式的，其数据粒度通常和 ODS 的粒度相同。在 PDW 层会保存 BI 系统中所有的历史数据，例如保存 10 年的数据。

3）DM 层：

为数据集市层，这层数据是面向主题来组织数据的，通常是星形或雪花结构的数据。从数据粒度来说，这层的数据是轻度汇总级的数据，已经不存在明细数据了。从数据的时间跨度来说，通常是 PDW 层的一部分，主要的目的是为了满足用户分析的需求，而从分析的角度来说，用户通常只需要分析近几年（如近三年的数据）的即可。从数据的广度来说，仍然覆盖了所有业务数据。

4）APP 层：

为应用层，这层数据是完全为了满足具体的分析需求而构建的数据，也是星形或雪花结构的数据。从数据粒度来说是高度汇总的数据。从数据的广度来说，则并不一定会覆盖所有业务数据，而是 DM 层数据的一个真子集，从某种意义上来说是 DM 层数据的一个重复。从极端情况来说，可以为每一张报表在 APP 层构建一个模型来支持，达到以空间换时间的目的数据仓库的标准分层只是一个建议性质的标准，实际实施时需要根据实际情况确定数据仓库的分层，不同类型的数据也可能采取不同的分层方法。

11.6.2 为什么要对数据仓库分层？

1）用空间换时间，通过大量的预处理来提升应用系统的用户体验（效率），因此数据仓库会存在大量冗余的数据。

2）如果不分层的话，如果源业务系统的业务规则发生变化将会影响整个数据清洗过程，工作量巨大。

3）通过数据分层管理可以简化数据清洗的过程，因为把原来一步的工作分到了多个步骤去完成，相当于把一个复杂的工作拆成了多个简单的工作，把一个大的黑盒变成了一个白

盒，每一层的处理逻辑都相对简单和容易理解，这样我们比较容易保证每一个步骤的正确性，当数据发生错误的时候，往往我们只需要局部调整某个步骤即可。

11.7 元数据介绍

当需要了解某地企业及其提供的服务时，电话黄页的重要性就体现出来了。元数据（Metadata）类似于这样的电话黄页。

1）元数据的定义

数据仓库的元数据是关于数据仓库中数据的数据。它的作用类似于数据库管理系统的数据字典，保存了逻辑数据结构、文件、地址和索引等信息。广义上讲，在数据仓库中，元数据描述了数据仓库内数据的结构和建立方法的数据。

元数据是数据仓库管理系统的重要组成部分，元数据管理器是企业级数据仓库中的关键组件，贯穿数据仓库构建的整个过程，直接影响着数据仓库的构建、使用和维护。

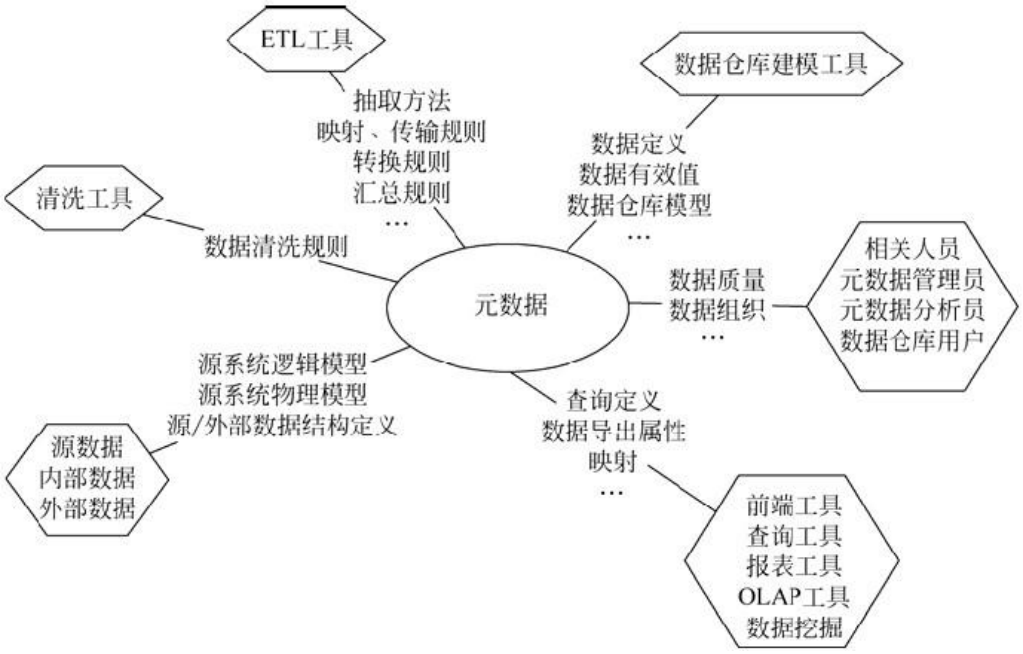
（1）构建数据仓库的主要步骤之一是 ETL。这时元数据将发挥重要的作用，它定义了源数据系统到数据仓库的映射、数据转换的规则、数据仓库的逻辑结构、数据更新的规则、数据导入历史记录以及装载周期等相关内容。数据抽取和转换的专家以及数据仓库管理员正是通过元数据高效地构建数据仓库。

（2）用户在使用数据仓库时，通过元数据访问数据，明确数据项的含义以及定制报表。

（3）数据仓库的规模及其复杂性离不开正确的元数据管理，包括增加或移除外部数据源，改变数据清洗方法，控制出错的查询以及安排备份等。

元数据可分为技术元数据和业务元数据。技术元数据为开发和管理数据仓库的 IT 人员使用，它描述了与数据仓库开发、管理和维护相关的数据，包括数据源信息、数据转换描述、数据仓库模型、数据清洗与更新规则、数据映射和访问权限等。而业务元数据为管理层和业务分析人员服务，从业务角度描述数据，包括商务术语、数据仓库中有什么数据、数据的位置和数据的可用性等，帮助业务人员更好地理解数据仓库中哪些数据是可用的以及如何使用。

由上可见，元数据不仅定义了数据仓库中数据的模式、来源、抽取和转换规则等，而且是整个数据仓库系统运行的基础，元数据把数据仓库系统中各个松散的组件联系起来，组成了一个有机的整体，如图所示



2）元数据的存储方式

元数据有两种常见存储方式：一种是以数据集为基础，每一个数据集有对应的元数据文件，每一个元数据文件包含对应数据集的元数据内容；另一种存储方式是以数据库为基础，即元数据库。其中元数据文件由若干项组成，每一项表示元数据的一个要素，每条记录为数据集的元数据内容。上述存储方式各有优缺点，第一种存储方式的优点是调用数据时相应的元数据也作为一个独立的文件被传输，相对数据库有较强的独立性，在对元数据进行检索时可以利用数据库的功能实现，也可以把元数据文件调到其他数据库系统中操作；不足是如果每一数据集都对应一个元数据文档，在规模巨大的数据库中则会有大量的元数据文件，管理不方便。第二种存储方式下，元数据库中只有一个元数据文件，管理比较方便，添加或删除数据集，只要在该文件中添加或删除相应的记录项即可。在获取某数据集的元数据时，因为实际得到的只是关系表格数据的一条记录，所以要求用户系统可以接受这种特定形式的数据。因此推荐使用元数据库的方式。

元数据库用于存储元数据，因此元数据库最好选用主流的关系数据库管理系统。元数据库还包含用于操作和查询元数据的机制。建立元数据库的主要好处是提供统一的数据结构和业务规则，易于把企业内部的多个数据集市有机地集成起来。目前，一些企业倾向建立多个数据集市，而不是一个集中的数据仓库，这时可以考虑在建立数据仓库（或数据集市）之前，先建立一个用于描述数据、服务应用集成的元数据库，做好数据仓库实施的初期支持工作，对后续开发和维护有很大的帮助。元数据库保证了数据仓库数据的一致性和准确性，为企业进行数据质量管理提供基础。

3）元数据的作用

在数据仓库中，元数据的主要作用如下。

（1）描述哪些数据在数据仓库中，帮助决策分析者对数据仓库的内容定位。

（2）定义数据进入数据仓库的方式，作为数据汇总、映射和清洗的指南。

（3）记录业务事件发生而随之进行的数据抽取工作时间安排。

（4）记录并检测系统数据一致性的要求和执行情况。

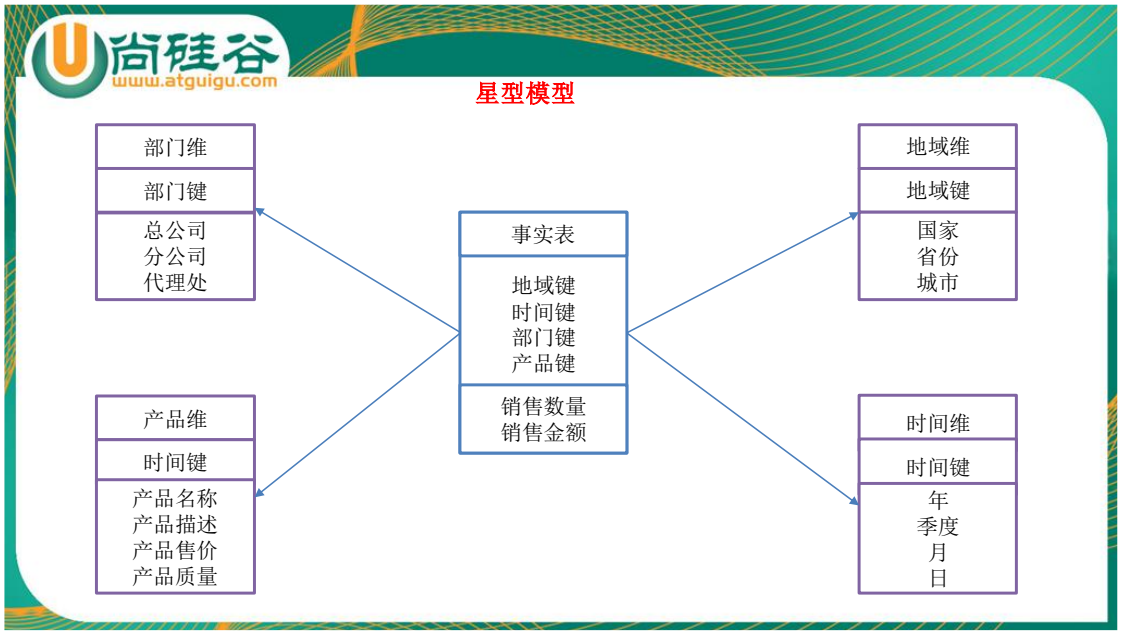
（5）评估数据质量。

11.8 星型模型和雪花模型

在多维分析的商业智能解决方案中，根据事实表和维度表的关系，又可将常见的模型分为星型模型和雪花型模型。在设计逻辑型数据的模型的时候，就应考虑数据是按照星型模型还是雪花型模型进行组织。

11.8.1 星型模型

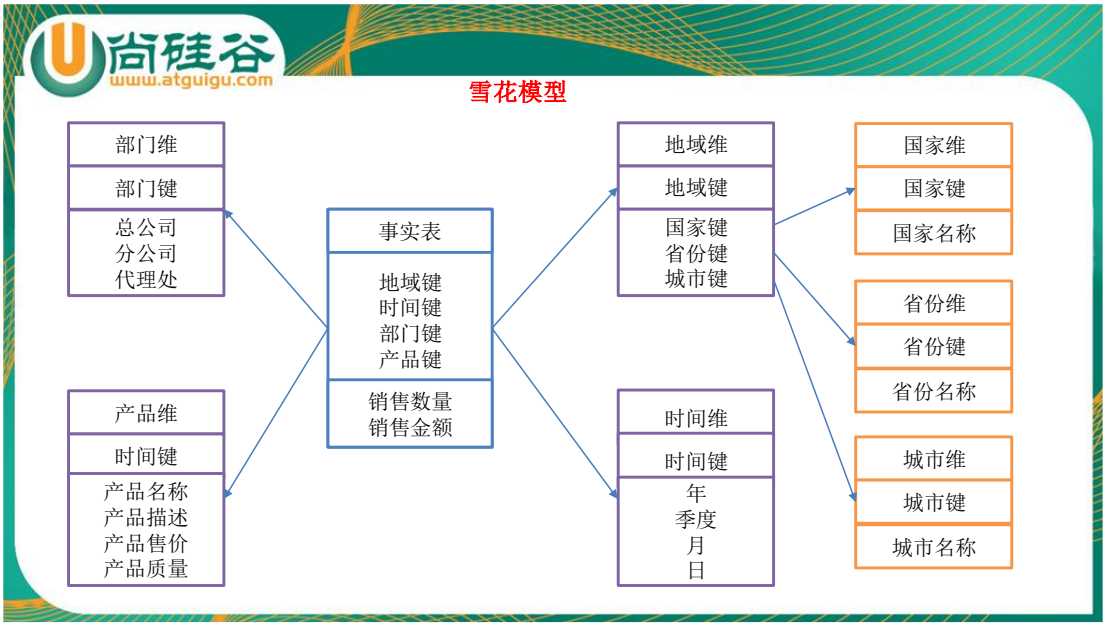
当所有维表都直接连接到“ 事实表”上时，整个图解就像星星一样，故将该模型称为星型模型。



星型架构是一种非正规化的结构，多维数据集的每一个维度都直接与事实表相连接，不存在渐变维度，所以数据有一定的冗余，如在地域维度表中，存在国家 A 省 B 的城市 C 以及国家 A 省 B 的城市 D 两条记录，那么国家 A 和省 B 的信息分别存储了两次，即存在冗余。

11.8.2 雪花模型

当有一个或多个维表没有直接连接到事实表上，而是通过其他维表连接到事实表上时，其图解就像多个雪花连接在一起，故称雪花模型。雪花模型是对星型模型的扩展。它对星型模型的维表进一步层次化，原有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，这些被分解的表都连接到主维度表而不是事实表。如图所示，将地域维表又分解为国家，省份，城市等维表。它的优点是：**通过最大限度地减少数据存储量以及联合较小的维表来改善查询性能。**雪花型结构去除了数据冗余。



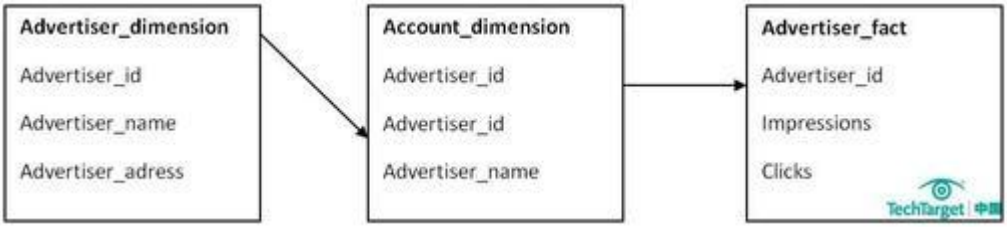
星型模型因为数据的冗余所以很多统计查询不需要做外部的连接，因此一般情况下效率比雪花型模型要高。星型结构不用考虑很多正规化的因素，设计与实现都比较简单。雪花型模型由于去除了冗余，有些统计就需要通过表的联接才能产生，所以效率不一定有星型模型高。正规化也是一种比较复杂的过程，相应的数据库结构设计、数据的 ETL、以及后期的维护都要复杂一些。**因此在冗余可以接受的前提下，实际运用中星型模型使用更多，也更有效率。**

11.8.3 星型模型和雪花模型对比

星形模型和雪花模型是数据仓库中常用到的两种方式，而它们之间的对比要从四个角度来进行讨论。

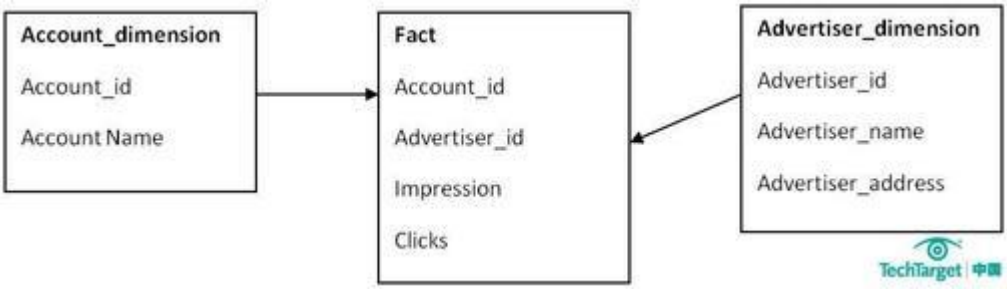
1）数据优化

雪花模型使用的是规范化数据，也就是说数据在数据库内部是组织好的，以便消除冗余，因此它能够有效地减少数据量。通过引用完整性，其业务层级和维度都将存储在数据模型之中。



雪花模型

相比较而言，星形模型使用的是反规范化数据。在星形模型中，维度直接指的是事实表，业务层级不会通过维度之间的参照完整性来部署。



星形模型

2）业务模型

主键是一个单独的唯一键(数据属性)，为特殊数据所选择。在上面的例子中，Advertiser_ID 就将是一个主键。外键(参考属性)仅仅是一个表中的字段，用来匹配其他维度表中的主键。在我们所引用的例子中，Advertiser_ID 将是 Account_dimension 的一个外键。在雪花模型中，数据模型的业务层级是由一个不同维度表主键-外键的关系来代表的。而在星形模型中，所有必要的维度表在事实表中都只拥有外键。

3）性能

第三个区别在于性能的不同。**雪花模型在维度表、事实表之间的连接很多，因此性能方面会比较低。**举个例子，如果你想要知道 Advertiser 的详细信息，雪花模型就会请求许多信息，比如 Advertiser Name、ID 以及那些广告主和客户表的地址需要连接起来，然后再与事实表连接。

而星形模型的连接就少的多，在这个模型中，如果你需要上述信息，你只要将 Advertiser 的维度表和事实表连接即可。

4）ETL

雪花模型加载数据集市，因此 ETL 操作在设计上更加复杂，而且由于附属模型的限制，不能并行化。

星形模型加载维度表，不需要再维度之间添加附属模型，因此 ETL 就相对简单，而且可以实现高度的并行化。

总结

雪花模型使得维度分析更加容易，比如“针对特定的广告主，有哪些客户或者公司是在线的?”星形模型用来做指标分析更适合，比如“给定的一个客户他们的收入是多少?”