第3章

数据存储

本章先介绍数据仓库的概念模型、逻辑模型和物理模型；接着对元数据存储进行了简单讲解，然后讨论了数据集市的特点和组成；在介绍了传统数据仓库的体系结构后，讨论了传统数据仓库系统在大数据时代所面临的挑战；最后论述了大数据存储技术。

本章的重点内容如下：

（1）数据仓库的数据模型；

（2）元数据存储；

（3）数据集市。

3.1 数据仓库的数据模型

为了真实地模拟或抽象地表示现实中的各种系统，需要建立数据模型。数据模型（Data Model）是对现实世界数据特征的抽象表达，是用来描述数据的一组概念和定义。在信息管理中需要将现实世界的事务转换为信息世界的数据才能对信息进行处理与管理，这就需要依靠数据模型作为这种转换的桥梁。

在数据仓库开发过程中，首先将现实世界中的客观对象抽象为概念模型，然后把概念模型转化为数据仓库支持的物理模型。其转化过程如图 3-1 所示。

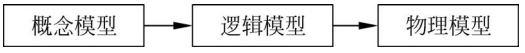

图 3-1 数据仓库的开发过程

如图 3-2 所示，在数据模型设计中要进行三级抽象。通过对现实世界的抽象，得出描述世界的概念模型。概念模型就是业务模型，是由企业决策者、商务领域知识专家和 IT 专家共同就企业级的跨领域业务系统需求进行分析的结果，随后分析系统的实际需求，构建数据逻辑关系模型，定义数据库结构。概念模型关联着数据仓库的逻辑模型和物理模型两部分。逻辑模型是进行数据管理、分析和交流的重要手段。物理模型描述数据在存储介质上的结构，以及将数据仓库的逻辑模型物理化到数据库的过程，可以构建数据仓库的物理分布模型，主要包含数据仓库的软硬件配置、资源情况以及数据仓库模式。

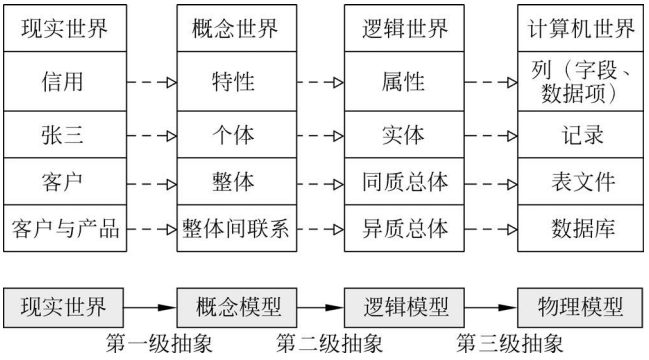

图 3-2　数据模型的三级抽象

3.1.1　数据仓库的概念模型

概念模型描述的是从客观世界到主观认识的映射，它是用于我们为一定的目标设计系统、收集信息而服务的一个概念性工具。

1. 概念模型设计的主要工作

（1）界定系统边界。进行任务和环境评估、需求收集和分析，了解用户迫切需要解决的问题及解决这些问题所需要的信息，要对现有数据库中的内容有一个完整而清晰的认识。

（2）确定主要的主题域及其内容。首先确定系统所包含的主题域，然后对每一个主题域的公共码键、主题域之间的联系、充分代表主题的属性组进行较为明确的描述。

2. 概念模型的设计方法

数据仓库的概念模型设计可以采用两种方法：E-R 模型和面向对象的分析方法。

（1）E-R 模型分析方法依次经过任务和环境评估、需求的收集和分析、主题选取和确定主题间关系、主题内容描述，最后绘制出如图 3-3 所示的系统 E-R 图。

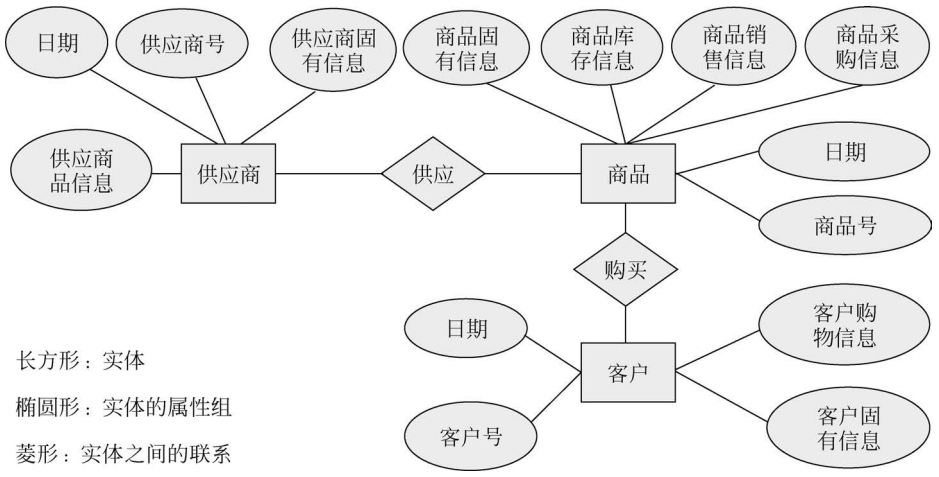

图 3-3　系统 E-R 图

【例 3-1】 假设有商品、客户和供应商 3 个实体。商品有如下属性组：商品固有信息、商品库存信息、商品销售信息和商品采购信息。客户有如下属性组：客户固有信息、客户购

物信息。供应商有如下属性组：供应商固有信息、供应商品信息。

（2）在面向对象的分析方法中，常用如图 3-4 所示的图形表示类表，类之间存在 3 种关系：继承、包含和关联。

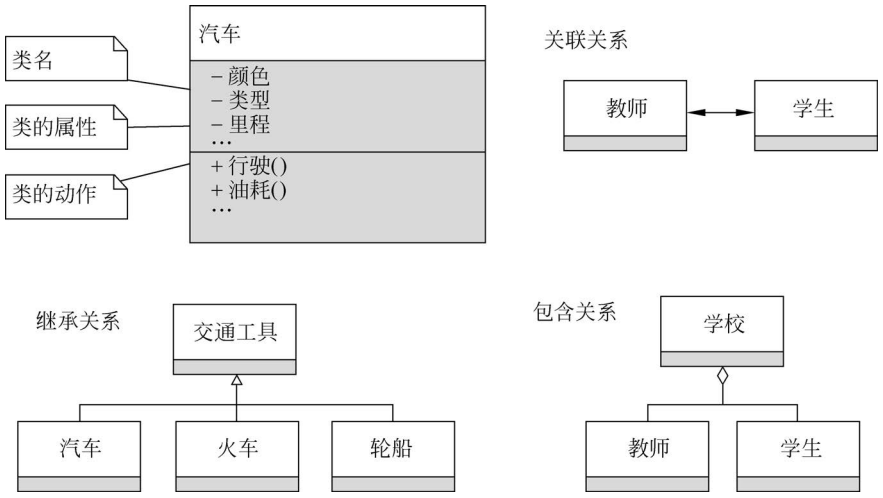

图 3-4　面向对象的图形表示

3.1.2　数据仓库的逻辑模型

逻辑模型是对数据仓库中主题的逻辑实现，从支持决策的角度去定义数据实体，更适合大量复杂查询。通常有两种逻辑模型表示法：星状模型和雪花模型。进行逻辑模型设计所要完成的主要工作包括分析主题域、定义逻辑模型、数据粒度的层次划分、确定数据分割策略、增加导出字段。

1. 星状模型与雪花模型

星状模型是数据集市维度建模中推荐的建模方法。如图 3-5 所示，星状模型是以事实表为中心，所有的维度表直接连接在事实表上，像星星一样。星状模型的特点是数据组织直观，执行效率高。因为在数据集市的建设过程中，数据经过了预处理，比如按照维度进行了汇总、排序等，数据量减少，执行的效率就比较高。

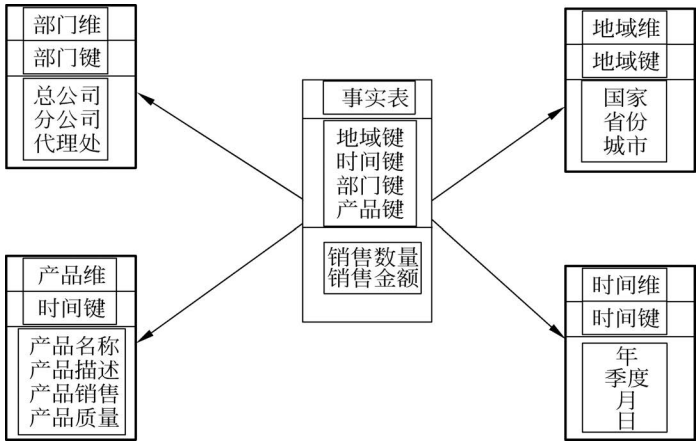

图 3-5　星状模型

雪花模型是介于星状模型和范式模型之间的。范式模型和雪花模型的区别在于雪花模型在维度上也是有冗余的。如图 3-6 所示的雪花模型中，地域维度不符合第三范式，因为地域维度中存在传递依赖：城市-省级-国家-地域。

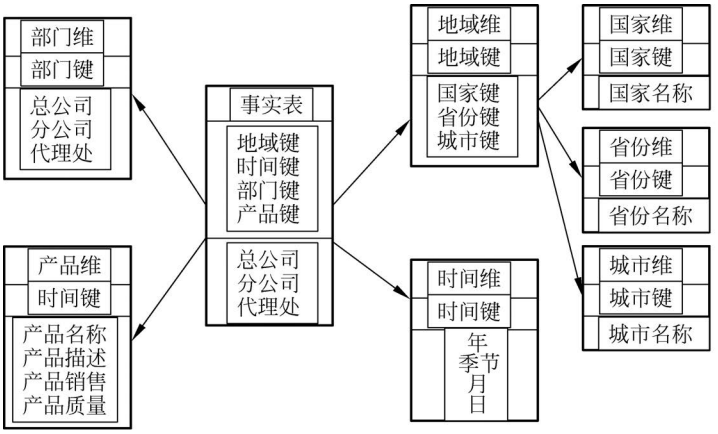

图 3-6　雪花模型

表 3-1 给出了星状模型与雪花模型的比较。

表 3-1　星状模型与雪花模型的比较

属　　性	星 状 模 型	雪 花 模 型
数据总量	多	少
可读性	容易	差
表个数	少	多
查询速度	快	慢
冗余度	高	低
实时表	一张或多张	一张或多张
维表	一级维表	多层级维表
扩展性	差	好

2. 数据粒度层次划分

数据粒度是指数据仓库中数据的细化和综合程度。根据数据粒度细化标准，细化程度越高，粒度越小；细化程度越低，粒度越大。确定粒度是数据仓库开发人员面对的一个重要设计问题。假设数据粒度划分合理，数据仓库设计和实现就能够顺利开展，否则，会导致数据仓库其他方面都很难进行。粒度的大小需要数据仓库设计时在数据量大小与查询的详细程度之间权衡确定。

设计粒度时需要注意什么问题？既然粒度对数据仓库这么重要，那么如何来设计粒度？确定粒度时的主要问题是使数据处于一个合适的级别，粒度级别不能太高也不能太低。低的粒度级别能提供详尽的数据。但要占用较多的存储空间和较长的查询时间，高的粒度级别能够高速、方便地进行查询，但不能提供过细的数据。总之，应结合业务的特点、数据量等方面综合考虑。

数据粒度层次划分遵循如下原则：

(1) 要接受的分析类型。粒度层次越高,就越不能进行细节分析。

(2) 可接受的最低粒度。粒度划分策略一定要保证数据的粒度确实能够满足用户的决策分析需要。

(3) 能存储数据的存储容量。若存储容量有限,则只能采用较高粒度的数据粒度划分策略。

单一粒度是指直接存储细节数据并定期在细节数据的基础上进行数据综合。从数据装载之后,所有细节数据都将保留在数据仓库中。存储期限(5～10 年)到了之后,才会导入后备设备(如磁带)中。双重粒度指对于细节数据只在数据仓库中保留近期的数据,当保留周期到达时,将距离当前较远的数据导出到磁盘上,从而为新的数据腾出空间。数据仓库中只保留在细节数据保留周期内的数据,对于这个周期之后的信息,数据仓库只保留其综合数据。

数据仓库之父比尔·恩门提出的数据粒度策略见表 3-2。

表 3-2 比尔·恩门的数据粒度策略

1 年内数据量(行)	5 年内数据量(行)	数据粒度策略
10 000	100 000	单一粒度,设计简单
100 000	1 000 000	如使用单一粒度,需认真设计
1 000 000	10 000 000	最好使用双重粒度
10 000 000	20 000 000	必须用双重粒度且需认真设计

3. 数据分割策略

数据分割将逻辑上统一的数据分散到各自的物理单元中,以便能分别处理,从而提高数据处理效率,数据分割后的数据单元称为分片。数据分割可按日期、地域、业务领域或按多个分割标准的组合进行。数据分割便于进行数据的重构、索引、重组、恢复。进行数据分割时需要考虑如下因素:

(1) 数据量的大小。若数据量较小,则可以不进行分割,或只用单一标准进行分割。若数据量很大,则应当采用多重标准的组合来较细致地分割数据。

(2) 数据分析处理的实际情况。数据分割与数据分析处理的对象是紧密联系的。

(3) 简单易行。选择用于数据分割的标准应当是自然的、易于实施的。

(4) 与粒度的划分策略相统一。同一粒度层次上的数据需要进行分割时,应当按照划分粒度层次时使用的标准进行分割。

(5) 数据的稳定性。数据仓库中的数据追加频率不同,有的快,有的慢,将不同变化频度的数据放在不同的表中进行更新处理。

4. 增加导出字段

导出字段是在原始数据的基础上进行总结或计算而生成的数据。这些数据可以在以后的应用中直接利用,避免了重复计算。

3.1.3 数据仓库的物理模型

物理模型是逻辑模型在数据仓库中的具体实现。构建数据仓库的物理模型与所选择的数据仓库开发工具密切相关。这个阶段所做的工作是根据信息系统的容量、复杂度、项目资源以及数据仓库项目自身的软件生命周期确定数据仓库系统的数据的存储结构、索引策略、数据存放位置、存储分配等。这部分工作应该是由项目经理和数据仓库架构师共同实施的。

进行逻辑模型设计主要完成以下几项工作：

(1) 确定数据的存储结构。数据仓库中包含巨量数据，为了提高数据的访问效率和可靠性，必须认真选择数据的存储结构。对于数据存储问题的解决，有两种可选的方式：分布式存储方式和集中存储方式。数据分布式存储是采用磁盘阵列在多个节点间分布的方式来存储数据。数据集中存储是将现有的 SAN 或 NAS 系统作为服务器的存储部分。

(2) 确定索引策略。在数据仓库中由于数据量很大，需要对数据的存取路径进行仔细设计和选择，建立专用的复杂的索引，以获得最高的存取效率。常见的索引技术有 B-Tree 索引、位索引技术、标识技术、广义索引及连接索引等。

(3) 确定数据存储策略表的归并，如分割表的存放、按列存储、存储分配优化。

(4) 存储分配优化是解决诸如数据块大小、缓冲区单元大小和个数、系统配置等相关的问题，通常不同的数据仓库厂商都会根据其产品的应用实例给出推荐的配置参数，设计人员可以参考这些数据，系统配置还要在系统维护过程中根据实际情况(数据的增长速度、用户查询的数量和额度)进行调整。

3.2 元数据存储

3.2.1 元数据的概念

元数据(Metadata)是描述数据的数据，用于建立、管理、维护和使用数据仓库。元数据管理是企业数据仓库的关键组件，贯穿于建立数据仓库的整个过程，直接影响着数据仓库的构建、使用和维护。

元数据包括数据从哪里来、流通多长时间、更新频率是多少、数据的含义是什么以及数据已经进行了哪些计算、转换和筛选等。

例如，每张数码照片都包含 EXIF 信息，就是用来描述数码图片的元数据。按照 Exif 2.1 标准，其中主要包含如下信息：

Image Description(图像描述、来源)指设备名；

Artist(作者)有些相机可以输入使用者的名字；

Make(生产者)指设备生产厂家；

Model(型号)指设备型号；

Orientation(方向)有的相机支持，有的不支持；

Software(软件)显示固件 Firmware 版本；

DateTime(日期和时间)；

……

3.2.2 元数据的分类方法

1. 按元数据的类型分类

按照数据类型，元数据可以分为以下两类。

(1) 基础元数据：基础元数据是指数据仓库系统中所有的数据源、数据集市、数据仓库和应用中的数据。

（2）数据处理元数据：数据处理元数据是数据仓库系统中与数据处理过程紧密相关的元数据，它包括数据加载、清洗、更新、分析和管理信息。

2. 按抽象层次分类

按照抽象层次，元数据可以分为以下 3 类。

（1）概念元数据：应用系统、预定义查询和分析应用相关的信息。

（2）逻辑元数据：应用数学语言的描述，从某种程度上是概念元数据更深层次的描述。

（3）物理元数据：关于数据仓库实现的最底层信息，包括事务规则、SQL 编码、关系索引文件和分析应用代码等。

3. 按用户角度分类

从用户角度，元数据可以分为以下两类。

（1）管理元数据：是存储关于数据仓库系统技术细节的数据，用于开发和管理数据仓库。包括数据仓库结构的描述、汇总用的算法、由操作环境到数据仓库环境的映射。

（2）用户元数据：从最终用户角度描述数据仓库，包括如何连接数据仓库、可以访问数据仓库的哪些数据、数据来自哪一个源系统。

4. 按元数据来源分类

从数据来源的角度，元数据可以分为以下 3 类。

（1）工具元数据：指由 ETL（数据抽取、数据转换、数据装载）组件、数据仓库设计工具等产生的元数据。

（2）资源元数据：指由操作系统、数据集市、数据库和数据字典生成的元数据。

（3）外部数据：指从本地数据仓库系统以外的其他系统输入的元数据，如业务系统数据库中的数据。

3.2.3　元数据的管理

从元数据的发展历史不难看出，对于相对简单的环境，元数据管理只需按照通用的元数据管理标准建立一个集中式的元数据知识库即可；对于比较复杂的环境，分别建立各部分的元数据管理系统，形成分布式元数据知识库，然后，通过建立标准的元数据交换格式，实现元数据的集成管理。

元数据管理功能包括数据抽取、数据建模、数据存储、数据展示。一般可以采用集中式的元数据知识库或分布式元数据知识库和标准的元数据交换格式实现元数据管理。元数据管理工具包括数据抽取工具、建模工具、前端展现工具和元数据存储工具等，如图 3-7 所示。

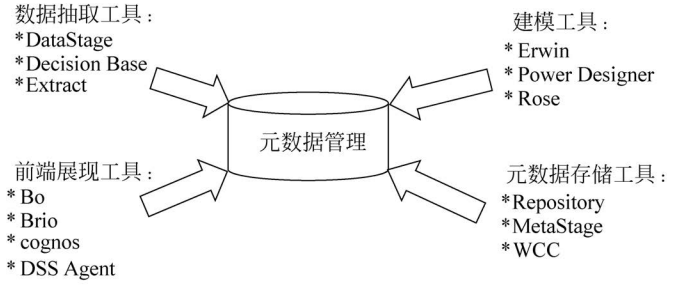

图 3-7　元数据管理工具

3.2.4 元数据的作用

元数据是进行数据集成所必需的。元数据定义的语义层可以帮助最终用户理解数据仓库中的数据，元数据是保证数据质量的关键，元数据可以支持需求变化。具体体现在以下几个方面。

1. 元数据是进行数据集成所必需的

数据仓库最大的特点就是它的集成性。这一特点不仅体现在它所包含的数据上，还体现在实施数据仓库项目的过程中。一方面，从各个数据源中抽取的数据要按照一定的模式存入数据仓库中，这些数据源与数据仓库中数据的对应关系及转换规则都要存储在元数据知识库中；另一方面，在数据仓库项目实施过程中，直接建立数据仓库往往费时、费力，因此在实践中，人们可能会按照统一的数据模型，首先建设数据集市，然后在各个数据集市的基础上再建设数据仓库。不过，当数据集市数量增多时很容易形成“蜘蛛网”现象，而元数据管理是解决“蜘蛛网”问题的关键。如果在建立数据集市的过程中，注意了元数据管理，那么在集成到数据仓库中时就会比较顺利；相反，如果在建设数据集市的过程中忽视了元数据管理，那么最后的集成过程就会很困难，甚至不可能实现。

2. 元数据定义的语义层可以帮助用户理解数据仓库中的数据

最终用户不可能像数据仓库系统管理员或开发人员那样熟悉数据库技术，因此迫切需要有一个“翻译”，能够使他们清晰地理解数据仓库中数据的含义。元数据可以实现业务模型与数据模型之间的映射，因而可以把数据以用户需要的方式“翻译”出来，从而帮助最终用户理解和使用数据。

3. 元数据是保证数据质量的关键

数据仓库或数据集市建立好以后，使用者在使用的时候，常常会产生对数据的怀疑。这些怀疑往往是由于底层的数据对于用户来说是不“透明”的，使用者很自然地对结果产生怀疑。而借助元数据管理系统，最终的使用者会很方便地得到各个数据的来龙去脉以及数据抽取和转换的规则，这样他们自然会对数据具有信心；当然也可便捷地发现数据所存在的质量问题。甚至国外有学者在元数据模型的基础上引入质量维，从更高的层次上来解决这一问题。

4. 元数据可以支持需求变化

随着信息技术的发展和企业职能的变化，企业的需求也在不断地改变。如何构造一个随着需求改变而平滑变化的软件系统，是软件工程领域中的一个重要问题。传统的信息系统往往通过文档来适应需求变化，但是仅依靠文档是远远不够的。成功的元数据管理系统可以把整个业务的工作流、数据流和信息流有效地管理起来，使得系统不依赖于特定的开发人员，从而提高系统的可扩展性。

3.3 数据集市

3.3.1 数据集市的概念

1. 数据集市的产生

企业级的数据仓库能够对数据进行存储、采集和分析，从而满足用户的不同需求。然

而，不同部门职责范围不同，需要采集和分析不同的数据。如果全部数据操作和处理都从数据仓库进行，会加重系统的负担，降低工作效率，造成资源浪费。最终用户对信息检索要求是高性能的，即越快越好，但数据仓库开发周期较长。

数据集市就是在这个背景下发展起来的，一方面符合部门级数据分析的需要，另一方面减轻了中央数据仓库的负担，提高了工作效率。数据集市是在数据仓库的基础上发展起来的，通常由各部门安排数据集市中存储的数据。相关数据表明，数据集市的投资占数据仓库投资的50%。

2. 数据集市的定义

数据集市是一种小型的部门级的数据仓库，主要面向部门级业务，并且只面向某个特定的主题，是为满足特定用户(一般是部门级别的)的需求而建立的一种分析型环境。它的投资规模比较小，更关注在数据中构建复杂的业务规则来支持功能强大的分析。数据集市常称为“部门级数据仓库”。

数据集市具有灵活、便捷、简单的优势，为决策人寻找信息提供方便，所以数据集市具有如下特征：面向部门、特定服务、规模小、成本低、使用简单、维护方便、由业务部门规划和实现、投资回收快、集成性、工具集完备等。数据集市是数据仓库的子集和一部分，继承了数据仓库的特征和优势。

对于数据集市，有时会出现一定的认识误区，如单纯用数据量大小来区分数据集市和数据仓库，认为数据集市比较容易建立，很容易升级到数据仓库。独立型数据集市的存在会给人造成一种错觉，似乎可以先独立地构建数据集市，当数据集市达到一定的规模可以直接转换为数据仓库，但这是不正确的，多个独立的数据集市的累加并不能形成一个企业级的数据仓库，这是由数据仓库和数据集市本身的特点决定的。如果脱离集中式的数据仓库，独立地建立多个数据集市，企业只会又增加了一些信息孤岛，仍然不能以整个企业的视图分析数据，数据集市为各个部门或工作组所用，各个集市之间又会存在不一致性。

数据仓库与数据集市的区别具体见表3-3。

表3-3 数据仓库与数据集市的区别

	数据仓库	数据集市
范围	企业级	部门级
主题	企业主题	部门或特殊的分析主题
数据粒度	最细粒度	较粗的粒度
历史数据	大量的历史数据	适度的历史数据
优化	探索	便于访问和分析、快速查询

3.3.2 数据集市的类型

按照不同的数据来源和建立方法，数据集市可以分为独立型数据集市和从属型数据集市。

(1) 独立型数据集市是指它的数据直接来源于各操作数据环境，当为各个部门建立相关数据集市后，这些数据集市之间相互独立，可能具有不同的数据存储类型，具体如图3-8所示。

（2）从属型数据集市的数据来自企业级数据仓库，是企业级数据仓库的子集。如图 3-9 所示，各数据集市中数据的组织、格式和结构在整个系统中保持一致。一般为那些访问数据仓库十分频繁的关键业务部门建立从属型数据集市，这样可以更好地提高查询反应速度。

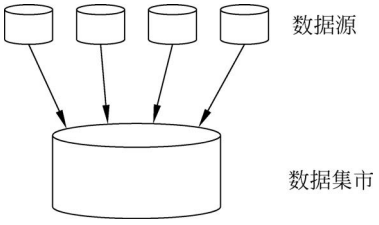

图 3-8　独立型数据集市

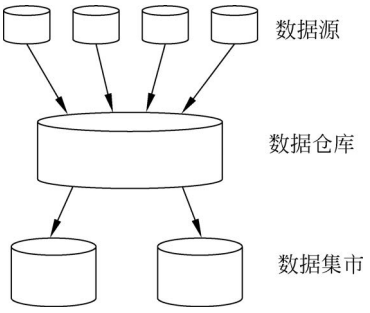

图 3-9　从属型数据集市

3.4　大数据存储技术

大数据（big data）是指所涉及的数据量规模大到无法通过传统 IT 技术和软硬件工具在合理时间内对其进行感知、获取、管理及收理的数据集合。

目前，大数据主要来源于搜索引擎服务、电子商务、社交网络、音频/视频、在线服务、个人数据业务、地理信息数据、传统企业活动、公共服务等领域。数据通常以每年增长 50％的速度激增，尤其是非结构化数据。随着科技的进步，出现了越来越多的传感器采集数据、移动设备、社交多媒体等，并且数据量将会继续高速增长。这些数据早已远远超越了目前人力所能处理的范畴。总之，大数据的时代已经到来。

大数据具备数据量大、类型繁多、价值密度低、速度快等特征，如图 3-10 所示。

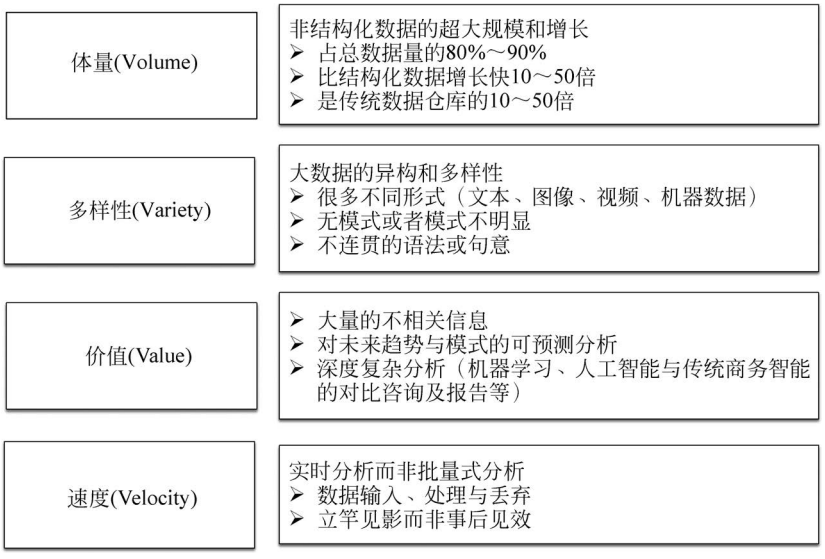

图 3-10　大数据的特征

3.4.1 传统数据库管理系统

传统数据库依次经历了层次数据库、网状数据库和关系数据库3个发展阶段。

1. 层次数据库

层次模型是数据库系统中最早出现的数据模型，层次数据库系统采用层次模型作为数据的组织方式。层次模型用树状结构来表示各类实体以及实体间的联系。层次模型的优点是数据结构比较简单清晰、查询效率高、提供了良好的完整性支持。层次模型的缺点是不合适表示非层次性联系，上下层1：N联系容易产生数据冗余，不能表达多对多关系，容易引起数据不一致。

2. 网状数据库

网状数据库采用网状模型，克服了层次数据库的缺点。它能够更为直接地描述现实世界，如一个节点可以有多个双亲，节点之间可以有多种联系；具有良好的性能，存取效率较高。其缺点是数据量越大，结构越复杂，不利于用户掌握；用户必须了解系统存储结构的细节，加重了编程的负担。

3. 关系数据库

关系数据库采用关系模型，它是建立在严格的数学概念基础上。关系模型简单清晰，无论实体还是实体之间的联系都用关系来表示。对数据的检索和更新结果也是关系。关系模型的存取路径对用户透明，从而具有更高的数据独立性、更好的安全保密性，也简化了程序员的工作和数据库开发建立的工作。所以关系模型诞生以后发展迅速，深受用户的喜爱。

关系数据库采用行式存储，即数据存放在数据文件内。数据文件的基本组成单位为块/页，块内结构包括块头、数据区。在数据库中，读某个列必须读入整行，行不等长，且修改数据可能导致行迁移。行数据较多时，可能导致行链接。

关系数据库在写入或者更新资料的过程中，能保证事务是正确可靠的。它具备以下特性。

(1) 原子性：在事务中执行的多个操作是原子性的，即要么操作全部执行，要么一个都不执行。

(2) 一致性：事务进行过程中整个数据的状态是一致的，不会出现数据矛盾等异常情况。

(3) 隔离性：两个事务不会相互影响、覆盖彼此数据等。

(4) 持久化：事务一旦完成，那么数据应该是被写到安全的、持久化存储的设备上。

关系数据库也存在以下不足。

(1) 面向对象编程与关系数据库间的不一致，这个问题影响的是开发效率。

(2) 关系数据库没有设计在集群上运行，传统的SQL Server、Oracle都是强依赖于磁盘系统来实现集群的。

(3) 关系数据库在单机容量达到上限的时候，做扩展是非常难的，往往要根据主键进行分表。一旦分表后，就已经违反关系数据库的范式了，因为“同一个集合的数据被拆分到多个表”。

(4) 当数据开始分布存储的时候，关系数据库逐渐演变成依赖主键的查询系统。

传统的关系数据库具有不错的性能，稳定性高，历经多年发展已日臻成熟，而且使用简

单，功能强大，也积累了大量的成功案例。在 20 世纪 90 年代的互联网领域，网站基本都是静态网页，主要以文字为主，访问量也不大，当时用单个数据库完全可以应对。近年来，动态网站随处可见，各种论坛、博客、微博异常火爆，用户数据量迅速增长，处理事务性数据得心应手的关系数据库，面对互联网的高并发、大数据量变得力不从心，暴露了很多难以克服的问题。具体如下。

(1) 数据库高并发读写。高并发的动态网站数据库并发负载非常高，往往要达到每秒上万次甚至百万次、千万次的读写请求。关系数据库应付上万次 SQL 查询没问题，但是应付上百万、千万次 SQL 数据请求，硬盘 I/O 就已经无法承受了。

(2) 海量数据的高效率访问。一般大型数据库在百万级的数据库表中检索数据可达到秒级，但面对数亿条记录的数据库表，检索速度效率是极其低下，令人难以忍受的。

(3) 数据库可扩展性和高可用性。基于 Web 的架构中，数据库无法通过添加更多的硬件和服务节点来扩展性能和负载能力，对于很多需要提供 24 小时不间断服务的网站来说，数据库系统升级和扩展只能通过停机来实现，但这无疑是一个艰难的决定。

关系数据库的改进和优化已经无法从根本上解决上述难题，无法满足大数据存储和处理要求。大数据需要非常高性能、高吞吐率、大容量的基础设备和新一代存储技术。

3.4.2 NoSQL 数据库

NoSQL 的准确含义是 Not only SQL，是对不同于传统的关系数据库的数据库管理系统的统称。NoSQL 用于超大规模数据的存储。NoSQL 作为新兴的数据库系统，由于其具备处理海量数据的能力，近年来受到各大 IT 公司的追捧。Amazon、Google、阿里巴巴等国内外大型企业已纷纷斥资进行研究并开发适用的产品。

NoSQL 数据库分为 Key-Value、Key-Document 和 Key-Column 这 3 类。常见的 NoSQL 产品有 Google 的 BigTable、基于 Hadoop HDFS 的 Hbase、Amazon 的 Dynamo、COUCHDB、MONGODB、Redis 等。

BigTable 是 Google 为解决其内部海量数据存储设计的分布式非关系数据库。稀疏的、分布式、持久化存储的多维度排序 Map 可处理 PB 级数据，并存储在上千台机器上，可应用在 Google 多个产品上。BigTable 使用 GFS 来存储日志和数据文件。BigTable 使用一个类似于 B＋树的三级结构来存储 Tablet 服务器(BigTable 组件之一)的放置信息。BigTable 的应用包括 Google Analytics、Google 地球个性化搜索等。Google Analytics 是帮助站长分析站点流量模式的服务，提供统计如每天内不同访问者的数量、每个 URL 的每天访问数等。Google 地球可以通过网页或客户端访问地球表面高分辨率卫星图像，而个性化搜索是用来记录用户查询和单击记录的可选服务。为提高可用性、降低延时，个性化搜索数据备份在多个 BigTable 集群上。

HBase 是 Apache 下 Hadoop 的存储系统，是一个高可靠性、高性能、面向列、可伸缩的分布式存储系统。HBase 是 BigTable 的开源实现，可在廉价 PC Server 上搭建大规模的结构化存储集群，利用 Hadoop Mapreduce 处理 HBase 中的海量数据。

GaussDB(DWS)是华为研发的一个企业级 AI-Native 分布式数据库。GaussDB 采用 MPP 架构，支持行存储与列存储，提供 PB 级别数据量的处理能力。可以为超大规模数据管理提供高性价比的通用计算平台，也可用于支撑各类数据仓库系统、BI 系统和决策支持

系统，为上层应用的决策分析提供服务。GaussDB 将 AI 能力植入数据库内核的架构和算法中，为用户提供更高性能、更高可用性、更多算力支持的分布式数据库。

如图 3-11 所示的 PolarDB 是阿里巴巴自主研发的下一代关系型云原生数据库，100％兼容 MySQL、100％兼容 PostgreSQL、高度兼容 Oracle。PolarDB 使用了存储计算分离架构，通过软硬件结合的设计，计算能力最高可扩展至 1000 核以上，存储容量最高可达 100TB，支持智能的读写分离，提供数据透明加密 TDE 和链路加密 SSL 能力。PolarDB 针对高弹性、大容量、高性能的业务场景而设计。

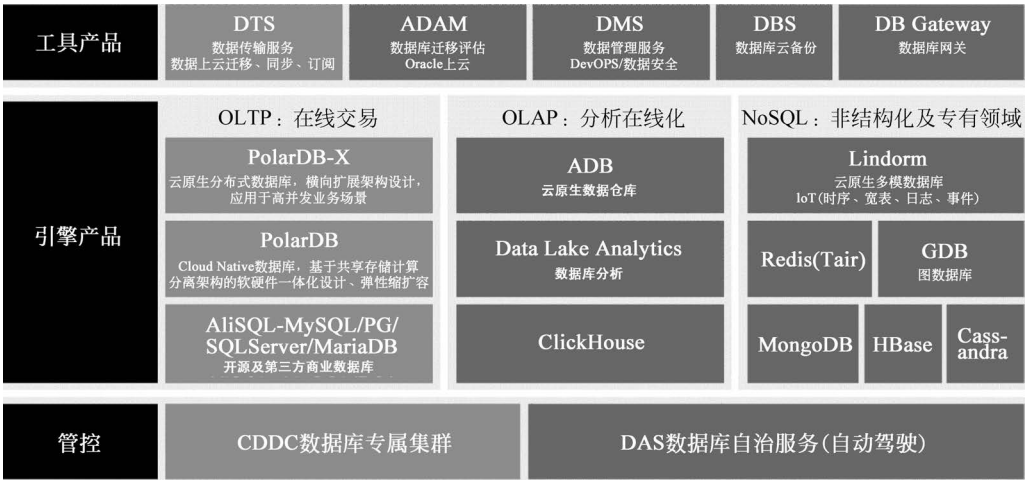

图 3-11 PolarDB 数据库

小结

本章介绍了数据仓库的数据模型，包括数据仓库的概念模型、逻辑模型和物理模型，并讨论了数据集市的特点和组成，讲解了传统数据仓库的体系结构和大数据存储技术。通过本章的学习，读者可以对数据仓库和数据集市有一个基本的认识，了解到大数据时代数据存储技术和国内外主流产品，为以后开展数据仓库和数据挖掘的应用研究奠定基础。

习题

1. 选择题

(1) 建立 E-R 模型的过程包括(　　)。

A. 任务和环境评估　　B. 需求的收集和分析

C. 主题选取，确定主题间关系　　D. 主题内容描述

(2) 以下关于采用面向对象方法进行概念模型设计的说法不正确的是(　　)。

A. E-R 模型中的实体转化为面向对象系统中的类

B. E-R 模型中实体的属性对应面向对象系统中类的属性

C. E-R 模型中实体间的关系表现为面向对象系统中类间的关系

D. 设计过程为需求分析→确定类间关系→选择类→描述类属性和动作

(3) 逻辑模型是对数据仓库中主题的逻辑实现,进行逻辑模型设计所要完成的主要工作有(　　)。

A. 分析主题域,定义逻辑模型　　B. 数据粒度的层次划分

C. 确定数据分割策略　　D. 增加导出字段

(4) 物理模型是逻辑模型在数据仓库中的具体实现,进行逻辑模型设计所要完成的主要工作有(　　)。

A. 确定数据的存储结构　　B. 确定数据的索引策略

C. 确定数据的存储策略　　D. 存储分配优化

(5) 以下哪个选项不属于元数据管理工具中的建模工具?(　　)

A. Erwin　　B. Decision Base

C. Power Designer　　D. Rose

2. 判断题

(1) 数据仓库的开发过程为:先将现实世界中的客观对象抽象为概念模型,然后把概念模型转化为数据仓库支持的数据模型。(　　)

(2) 元数据(Metadata)是描述数据的数据,用于建立、管理、维护和使用数据仓库。元数据管理是企业数据仓库的关键组件,贯穿于建立数据仓库的整个过程。(　　)

(3) 基础元数据是数据仓库系统中与数据处理过程紧密相关的元数据,它包括数据加载、清洗、更新、分析和管理信息。(　　)

(4) 数据集市是一种小型的部门级的数据仓库,通常面向部门级业务的多个主题,是为满足特定用户的需求而建立的一种分析型环境。(　　)

(5) NoSQL 产品对性能的关注远远超过 ACID,往往只提供行级别的原子性操作,即对同一个 key 的操作会是串行执行,保证数据不会损坏。(　　)

3. 简答题

(1) 什么是数据模型?数据仓库开发中主要涉及哪几类模型?

(2) 什么是数据仓库概念模型?概念模型设计主要完成哪些工作?

(3) 什么是数据仓库逻辑模型?逻辑模型设计主要完成哪些工作?

(4) 对星状模型和雪花模型进行简要说明。

(5) 什么是数据粒度?数据粒度层次划分原则有哪些?

(6) 什么是元数据?简要说明元数据的分类及其作用。

(7) 对比数据集市和数据仓库的区别,并说明常见的几种对数据集市的认知误区。

(8) 传统数据库经历了哪 3 个发展阶段?各自特点是什么?

(9) 在大数据时代,传统关系数据库存在哪些不足?

(10) 大数据时代 NoSQL 数据库分类和主要数据产品有哪些?中国企业的产品有哪些?