

武器装备测试数据立方体模型的构建

汪浩洋, 吴伟, 向超, 唐健, 汤佳骏

(上海机电工程研究所, 上海 201109)

摘要: 针对大规模、高频率数据交换的实际需求, 结合武器装备测试数据的特点, 构建了装备测试系统的数据立方体模型; 首先介绍了数据立方体的基本概念以及常用的基本模型, 之后分析了装备测试系统的实际需求, 以概念模型、逻辑模型以及物理模型三级数据模型的方法实现整体设计; 数据立方体为装备测试数据提供多维的观测角度, 满足了实际任务对高效率数据交换的要求, 为之后的数据分析应用提供了良好的数据基础平台。

关键词: 数据立方体; 武器装备; 多维数据模型

Establishment of Data Cube for Weapon Equipment Test

WANG Haoyang, WU Wei, XIANG Chao, TANG Jian, TANG Jiajun

(Shanghai Eletro-Mechanical Engineering Institute, Shanghai 201109, China)

Abstract: In response to the actual needs of data exchange with large scale and high frequency, combining with the characteristics of weapon equipment test data, a data cube model of the equipment test system is established. This paper firstly introduces the fundamental concepts of data cubes and basic models, and then analyzes the actual requirements of the equipment test system, and realizes the overall design with the three-level data model: conceptual model, logical model and physical model. The data cube provides multidimensional observation angles for equipment test data, meets the requirements of high efficiency data exchange for actual tasks, and provides a good data foundation platform for subsequent data analysis applications.

Keywords: data cube; weapon equipment; multidimensional data model

0 引言

随着从数据中获取知识的概念提出, 以及信息化技术的发展, 数据作为一种资源被广泛重视并收集使用。武器装备测试系统在信息化实践过程中积累了大量的测试数据, 并随着武器装备的研制生产, 其规模越来越庞大。但人们更为关注数据背后蕴含的、能有效指导实践的知识, 这意味需要对海量数据进行挖掘分析, 提炼出有价值的信息。基础的数据存储以及管理方式虽然统筹存储数据, 解决了数据孤岛问题, 但难以满足大规模的数据挖掘处理以及复杂分析的需求, 因此需要构建一套高效便捷的数据管理架构。

在数据采集、存储、应用的全过程中, 形成了多种优化方法以提高实际效率。文献 [1-2] 对数据仓库抽取、转换以及加载数据的功能过程进行了梳理, 优化了任务调度策略, 在军备保障以及政务数据存储中得到了良好的运用。文献 [3-9] 则是构建了数据立方体模型, 利用其易与联机实时分析方法 (On Line Transaction Processing, OLAP) 结合的优点实现数据处理过程的优化。文献 [3] 与文献 [4] 将数据立方体模型与政区地图相结合, 实现了不同地区维度、不同统计方向下的数据统筹分析。文献 [7] 介绍了数据立方体在港口环境下的实际应用, 总结了数据立方体的构造以及软件实现, 运用实际数据进行了测试验证, 保证了数据立

方体的有效性。在文献 [8] 对犯罪预防分析与文献 [9] 对舆情监测分析中, 数据立方体模型起到了强力支撑作用, 为复杂多要素情境提供多维度分析视角。文献 [10] 将文本概念加入数据立方体, 形成更丰富的应用环境; 文献 [11] 与文献 [12] 则是将流数据与数据立方体结合, 提出基于流数据的概要数据立方体, 极大提高了索引效率。文献 [13-14] 提出了有向图方法对数据立方体进行详细描述; 文献 [15-16] 分别就紧凑、封闭等数据立方体细分领域进行深入研究, 提升数据立方体的数据查询效率。

本文根据国内外研究及应用成果, 将数据立方体概念引入到武器装备测控领域, 分析了数据立方体的设计思路, 依据装备测试数据的特点构建装备测试数据的数据立方体模型, 在不同维度上实现对测试数据的观测, 为之后的数据分析工作提供良好的数据交换平台。

1 数据立方体

1.1 数据立方体基本概念

自 Codd. E. F 在 20 世纪 70 年代提出关系型数据模式以来^[17], 关系型数据库以其结构简单清晰、易于维护等优点得以快速发展。数据立方体是多维数据模型的一种具体表现形式, 是基于关系数据库的一类多维矩阵, 能够多维度展示数据, 给用户 提供多个观测和分析数据的视角^[18-19]。

收稿日期:2021-01-31; 修回日期:2021-03-24。

作者简介:汪浩洋(1998-),男,湖北黄冈人,硕士研究生,主要从事测试数据分析、数据挖掘等方向的研究。

引用格式:汪浩洋,吴伟,向超,等.武器装备测试数据立方体模型的构建[J].计算机测量与控制,2021,29(9):142-146.

数据立方体是对基本事件的属性、度量值以及它们之间关联的表现形式进行重构得到的, 由维和事实定义。维是事实信息的属性, 至少包含维码、维层次、维成员等基本属性, 由维的基本属性构成的与之相关联的表称为维表。事实是数值度量的, 事实表包括事实的名称或度量, 以及每个相关维表的维码。

从名称上看, 数据立方体容易被理解是三维结构, 但实际上它可以由 n 维构成, 立方体是对多维数据存储的一种比喻。多维数据模型围绕中心主题组织构建, 不同视角分析数据的实际需求以及主题所关联的特性决定数据立方体的维度。也即是说, 我们可以根据实际的分析需求对构成数据立方体的维度进行增减, 优化数据结构, 例如微软曾构建过高达 64 维的多维数据模型。显然, 这是多维数据模型面向主题构建特性的体现。图 1 展示了围绕武器装备试验次数构建的数据立方体模型, 该数据立方体提供时间、地点、装备试验次数 3 个数据分析视角, 若想要分析不同环境下的试验次数, 则需设计一个环境维度重新构造数据立方体模型。

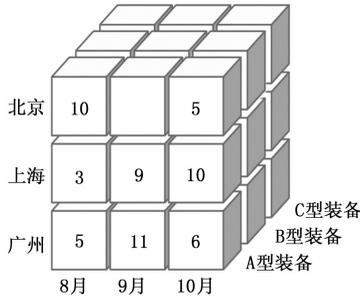


图 1 三维数据立方体示意图

数据立方体是一个概念模型, 它与物理模型的映射关系是通过确定维度以及维度内的成员、粒度、层次等要素来实现的。维度成员是构成维的基本单位, 对应着该维的一个取值, 例如对于时间维, 第一季度是它的一个维成员。粒度是一个维度内为提供详细分类信息而定义的划分单位。粒度越粗表示细节程度越低、综合程度越高; 粒度越细表示细节程度越高、分类信息越详实。信息详实程度往往与数据总量呈正相关, 即信息越详实, 数据规模越庞大。因此在资源有限的情况下, 需要进行合理的粒度划分, 优化数据结构, 提高查询分析效率。粒度的粗细对应不同的级别, 级别组成维度的层次结构。对于时间维的一个层次结构 {年, 半年, 季} 而言, 年是第一级别, 半年是第二级别, 季是第三级别, 这些级别间存在着“父子”包含关系。需要注意的是, 同一维度内可以存在多个层次结构, 例如时间维内可以有 {年, 半年, 季, 月} 的层次结构, 也可以有 {年, 月, 日} 的层次结构。

各维度的维成员在数据立方体内唯一逻辑标识确定的单位称为数据格, 该数据格存放聚集值。由所有维度最低级别的维成员确定的数据格称为基本格, 也称基本方体; 最高级别的维成员所确定的数据格称为顶点方体; 高层次

级别的数据格聚集值由低层次数据格汇聚得到。在图 1 中时间维度的“9 月”成员、地点维度的“上海”成员以及武器装备型号维度的“A 型装备”成员所唯一确定的方体存储的量值为 9, 即是在 9 月份上海有 9 次 A 型装备试验。

在数据立方体中并非每个方体内都包含量值, 不具备任何信息的方体被称为空方体。空方体不同于量值为 0 的方体, 它通常是数据缺失或其他异常情况导致的, 需要被特定标识。数据立方体中的空数据格作为一种特殊的数据格, 体现了数据立方体模型的稀疏性, 用基本空方体数与数据立方体的基本方体总数的比值来衡量稀疏性程度。空方体的这种特殊性会导致数值矩阵计算难以进行, 所以在实际操作中忽略空数据格是一个不错的选择。

1.2 数据立方体的模式

最初的关系型数据库设计应用实体-联系数据模型, 通过键码连接二维数据表实现数据存储检索^[20]。这种关系模型在查询时通过键值跳转查询, 对数据的全局展示较差, 并不利于数据的多维分析, 也不适用于联机分析处理。目前较为常用的数据仓库模型是多维数据模型, 主要有星形模型、雪花模型及事实星座模型 3 种样式。

星形模式数据库包括一个中心事实表与一组附属维表两大部分, 各维表连接到一个大的中心事实表上, 其连接图形如星形。在该模型下, 中心事实表包含事实数据信息以及各个维表的维码, 并且不含冗余项; 维表数量与事实表中的维码相对应, 每个维表仅包含一组属性。图 2 为武器装备测试数据的星形模型展开。

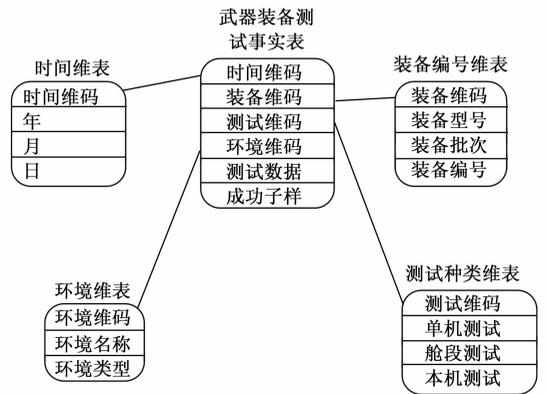


图 2 武器装备测试数据的星形模型表示

雪花模型和事实星座模型都能够由星形模型衍化而来, 可以看做是星形模型的特殊形式。雪花模型是在星形模式的基础上, 将某些维表进行规范化处理, 将数据进一步分解到附加表中得到的。相比于星形模式, 雪花模式的优点在于某些维表是规范化的形式, 易于维护, 并且节省了存储空间。不足之处在于规范附加维表后, 执行查询需要更多的连接操作, 带来浏览效率的降低, 可能会影响系统的性能。因此, 尽管雪花模式有一些优点, 适用于一些特定的场合, 比如地址的规范, 但是在多维数据模型中星形模型应用更广泛。

事实星座模型是将多个关联的星形模型合并相同维表得到的。在实际的复杂应用情景中，存在有多个事实表相互关联，表现为部分维表相同的情况，有多个中心事实表而共用相同维表的模型称为事实星座模型。

2 武器装备测试数据立方体的构建

数据立方体的构建需要分析实际应用场景，确定实际需求，从而明确数据仓库的主题，而后依据中心主题进行数据立方体设计。本节首先分析武器装备测试系统的实际需求，明确维度数量、维度间的关系以及各维度的层次结构，之后运用概念模型、逻辑模型以及物理模型三级数据模型方法实现数据立方体的构建。

2.1 武器装备测试系统需求分析

武器装备测试系统管理人员最关注武器装备生产质量是否合格，最关注的参数是武器装备的测试参数指标。由其研制过程来看，数据来源包括原理样机测试数据、工程样机测试数据以及定型测试数据，涵盖了设计、定型、生产的各阶段。因此可以以测试质量分析为中心主题构建一个数据立方体模型。但在实际操作过程中，由于武器装备系统的复杂性以及各层次管理人员的不同分工管理，单一的测试数据中心主题不足以满足实际需求。顶层设计人员需要从宏观把控武器装备型号立项，掌握全局信息，确定发展方向；中层管理人员需要确定测试项目及指标设计是否合理，从原理设计方面对其进行试验验证；技术人员需要检验生产质量，依据测试数据进行判读分析；配套方单独提供配套信息。因此在结合各方需求后，基本可以确定 4 个中心主题：武器装备设计分析、武器装备测试项目分析、武器装备测试数据分析以及配套信息分析。由此，确定武器装备测试数据立方体采用事实星座模式，由多个事实中心及其配套维表组成。事实星座模型设计可由多个星形模式合并相同维表得来，因此在下文分析中仅以武器装备测试数据分析的星形模式为例分析数据立方体的构建方法及步骤，在此基础上容易得到事实星座模式构建方法。

2.2 概念模型设计

概念模型设计是在较高抽象层次上的设计，它面向武器装备测试系统全局建立，为数据立方体的整体构建提供了全局的概念视图。一方面，数据立方体是对武器装备测试系统的历史数据进行集成和整合而组成的数据大集合，需要充分考虑使用者的功能需求；另一方面，需要对武器装备测试系统的数据内容及其特性有深入而清晰的理解，才能有针对性地设计完整方案。分析数据特性以及功能需求后，通过确定系统边界和定义主题域来反映数据仓库的概念模型。

武器装备测试数据分析主题的需求是有效判别武器装备生产质量，数据来源是测试数据信息。经过分析可以得到感兴趣的分析视角，确定维度信息，即时间维度、武器装备编号维度、测试种类维度、测试环境维度。时间维度中划分年、月、日 3 个粒度级别；武器装备编号维度中划分型号、批次、编号 3 个粒度级别；测试总类维度中划分

全弹测试、舱段测试、单机测试 3 个粒度级别；测试环境维度中信息较为繁杂，并没有明显的“父子”包含关系，多以并列关系存在。图 3 展示了武器装备编号维度的详细层次结构关系。

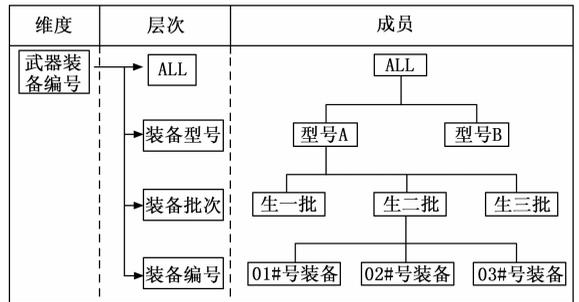


图 3 武器装备编号维度的层次结构

考虑多维数据模型的多维特性，超出三维的数据模型无法有效展现，本文采用信息包图表示多维数据。信息包图采用二维表格的形式表示多维数据，它有 3 个重要对象：维度、粒度和指标对象。根据上文的分析，详细确定了武器装备测试数据分析主题的维度以及粒度层次信息；指标对象信息是实际度量值，在武器装备测试主题中包括测试数据以及成功子样。将各确定的维度对象、粒度对象以及指标对象信息汇总后，以信息打包的方式得到信息包图，如图 4 所示。



图 4 武器装备测试数据信息包图

2.3 逻辑模型设计

逻辑模型是从概念模型到物理模型转化的中间步骤。按照概念模型阶段构建的基本对应关系，以及选定的数据模型形式要求，将概念模型转化为相应的逻辑模型。这一阶段的目的是确定各主题中包含的实体、各实体间的关联关系是什么以及实体间是否有约束关系。按照转换规则以及目的，武器装备测试数据分析主题确定指标实体以及维度实体两类实体。首先需要定义指标实体以及维度实体，其次是建立两者间的联系。指标实体包括数据指标和逻辑指标，数据指标即是主题域中所包含的数据，可以由信息包图的指标对象构成；逻辑指标是指区分数据的逻辑关系，可以由各维度的最低级别类别构成，得到唯一确定的逻辑标识。这样得到的数据与逻辑指标共同构成了星型图的指标实体。维度实体由各维度的各层次对象构成，例如时间维度的维度实体为 {年，月，日}。

指标实体包含维度实体的最低类别成员，因此指标实

体与维度实体之间的关系可以通过逻辑模型中的详细类别定义, 实现一一对应的关系。在星型图中, 每个维度实体通过最底层的详细类别实体和指标实体进行连接。由此构建得到星型图模型, 如图 5 所示。

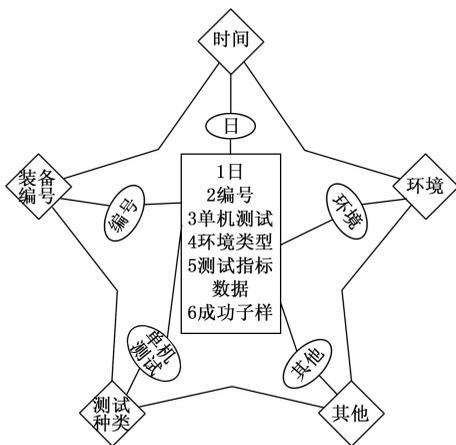


图 5 武器装备测试星型图

2.4 物理模型设计

物理模型是根据逻辑模型的设计内容, 在实际的物理具体介质上实现出来。物理模型阶段不仅需要建立物理结构模型, 而且需要规定数据标准, 明确具体特征, 以便于在物理介质上进行具体操作。定义数据标准包括命名约定、数据类型、约束、索引等方面; 实体特征包括值的类型、存储长度等方面信息。

在得到的星型图基础上, 要综合考虑数据立方体模型中事实表与维度表的实体构建, 以及它们之间的各种约束关系。根据逻辑设计的星型图, 将指标实体转化为中心事实表, 将维度实体转化为维度表。事实表中不仅仅包含数据的指标量, 还要包含维度实体中约定的主码值, 通过包含的主码与维表的外码构建事实表与维表之间的连接, 因此主码与外码键值关系必须一一对应。即是说事实表中包含两部分, 一部分是与之连接的维表主码值, 另一部分是事实数据的量值。维表包含定义的外码以及对应的一组维属性。

在武器装备测试数据分析主题中, 事实表包含时间标识、编号标识、测试标识以及环境标识共同构成的组合维码, 以及测试指标数据和成功子样的度量值信息。同时, 事实表中还含有数据类型等约定规范信息, 其具体的表结构如表 1 所示。

每个维都有与之对应的维表, 它展现了维度的外码键值、粒度级别信息以及约束信息。以武器装备编号维度为例, 维表包含了编号标识、{型号、批次、编号}等层次信息以及储存类型、数值长度等约束信息, 具体表结构如表 2 所示。

3 应用案例

本文实验数据来源于历史储存的装备测试数据。该数据集中每一条数据样本都有测试装备信息, 该装备测试的时间、环境等辅助信息, 反映不同测试类别的测试种类信息,

表 1 武器装备测试事实表

列名	字段名	数据类型	长度	是否为空	约束
时间标识	时间维码	字符型	8	不可为空值	主码外码一致
编号标识	装备维码	字符型	8	不可为空值	主码外码一致
测试标识	测试维码	字符型	8	不可为空值	主码外码一致
环境标识	环境维码	字符型	8	不可为空值	主码外码一致
测试指标数据	测试维码	浮点型	32	可为空值	
成功子样	产品状态	浮点型	32	可为空值	

表 2 武器装备编号维度表

列名	字段名	数据类型	长度	是否为空	约束
编号标识	装备维码	字符型	8	不可为空值	主码外码一致
装备型号	装备型号	字符型	8	不可为空值	
装备批次	装备批次	字符型	8	不可为空值	
装备编号	装备编号	字符型	8	不可为空值	

以及反映装备状态的测试数据组成。在武器装备测试数据管理平台中, 数据立方体配合联机分析方法采用多维分析技术, 实现数据的高效整合分析, 并可在前端界面进行表格化展示。

数据立方体的多维分析技术是在构建的数据立方体框架内对数据进行转换、整合分析, 其具体的操作主要包括切块、切片、下钻、上卷以及旋转。切片以及切块操作是选择特定目标数据进行查询分析, 下钻与上卷操作是依据层次结构进行数据整合分析, 旋转操作是从不同方向研究指标数据在维上的分布。

图 6 (a) 展示了装备编号维在下钻操作下的前端显示, 图 6 (b) 展示了数据立方体在切块操作下对特定区域数据的查询显示。可以看出, 装备编号维的下钻操作是依照设定的层次结构依次向下进行的, 体现了数据的从属关系, 能够有效地在总体与细节上观测数据; 对数据立方体进行切块查询操作, 选择出需要的数据对象, 能够精细地整合数据, 聚合同一类型的数据对象, 为数据分析提供了现实数据基础。

试验结果分析发现, 相较于表格存储, 对数据进行重构得到的数据立方体模型拥有更好的数据组织整合能力, 能够体现设计要求, 提供多维度观测分析的视角, 实现了数据灵活精细地组织整合, 满足了数据查询与分析的需求, 为之后的数据挖掘提供了良好的数据基础平台。

4 结束语

庞大的数据规模以及精细化管控目标不仅要求采用合适的数据分析方法得到有价值的知识, 而且需要对数据进行更深入的挖掘, 更全面的分析。本文采用数据立方体模

