



链滴

# 数据仓库维度建模10大基本原则

作者: [cxuan](#)

原文链接: <https://ld246.com/article/1471881417061>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h3 id="原则1-载入详细的原子数据到维度结构中">原则 1 载入详细的原子数据到维度结构中</h3>  
<blockquote>  
<p>维度建模应该使用最基础的原子数据进行填充，以支持不可预知的来自用户查询的过滤和分组请求，用户通常不希望每次只看到一个单一的记录，但是你无法预测 用户想要掩盖哪些数据，想要显示哪数据，如果只有汇总数据，那么你已经设定了数据的使用模式，当用户想要深入挖掘数据时他们就会到障碍。当然，原子数据也可以通过概要维度建模进行补充，但企业用户无法只在汇总数据上工作，他们需要原始数据回答不断变化的问题。</p>  
</blockquote>  
<h3 id="原则2-围绕业务流程构建维度模型">原则 2 围绕业务流程构建维度模型</h3>  
<blockquote>  
<p>业务流程是组织执行的活动，它们代表可测量的事件，如下一个订单或做一次结算，业务流程通常会捕获或生成唯一的与某个事件相关的性能指标，这些数据转换成事实后，每个业务流程都用一个原事实表表示，除了单个流程事实表外，有时会从多个流程事实表合并成一个事实表，而且合并事实表对单一流程事实表的一个很好的补充，并不能代替它们。</p>  
</blockquote>  
<h3 id="原则3-确保每个事实表都有一个与之关联的日期维度表">原则 3 确保每个事实表都有一个与之关联的日期维度表</h3>  
<blockquote>  
<p>原则 2 中描述的可测量事件总有一个日期戳信息，每个事实表至少都有一个外键，关联到一个日维度表，它的粒度就是一天，使用日历属性和非标准的关于测量事件日期的特性，如财务月和公司假指示符，有时一个事实表中有多个日期外键。</p>  
</blockquote>  
<h3 id="原则4-确保每个事实表中的事实具有相同的粒度或同级的详细程度">原则 4 确保每个事实中的事实具有相同的粒度或同级的详细程度</h3>  
<blockquote>  
<p>在组织事实表时粒度上有三个基本原则：事务，周期快照或累加快照。无论粒度类型如何，事实中的度量单位都必须达到相同水平的详细程度，如果事实表中的事实表现的粒度不一样，企业用户会搞晕，BI 应用程序会很脆弱，或者返回的结果根本就不对。</p>  
</blockquote>  
<h3 id="原则5-解决事实表中的多对多关系">原则 5 解决事实表中的多对多关系</h3>  
<blockquote>  
<p>由于事实表存储的是业务流程事件的结果，因此在它们的外键之间存在多对多(M:M)的关系，如个仓库中的多个产品在多天销售，这些外键字段不能为空，有时一个维度可以为单个测量事件赋予多值，如一个保健对应多个诊断，或多个客户有一个银行账号，在这些情况下，它的不合理直接解决了实表中多值维度，这可能违反了测量事件的天然粒度，因此我们使用多对多，双键桥接表连接事实表</p>  
</blockquote>  
<h3 id="原则6-解决维度表中多对一的关系">原则 6 解决维度表中多对一的关系</h3>  
<blockquote>  
<p>属性之间分层的、多对一(M:1)的关系通常未规范化，或者被收缩到扁平型维度表中，如果你曾有过为事务型系统设计实体关系模型的经历，那你一定要抵抗住旧有的思维模式，要将其规范化或将:1 关系拆分成更小的子维度，维度反向规范化是维度建模中常用的词汇。在单个维度表中多对一(M:1)的关系非常常见，一对一的关系，如一个产品描述对应一个产品代码，也可以在维度表中处理，在事实表中偶尔也有多对一关系，如详细当维度表中有上百万条记录时，它推出的属性又经常发生变化。不怎样，在事实表中要慎用 M:1 关系。</p>  
</blockquote>  
<h3 id="原则7-存储报告标记和过滤维度表中的范围值">原则 7 存储报告标记和过滤维度表中的范围值</h3>  
<blockquote>  
<p>更重要的是，编码和关联的解码及用于标记和查询过滤的描述符应该被捕获到维度表中，避免在实表中存储神秘的编码字段或庞大的描述符字段，同样，不要只在维度表中存储编码，假定用户不需描述性的解码，或它们将在 BI 应用程序中得到解决。如果它是一个行/列标记或下拉菜单过滤器，那它应该当作一个维度属性处理。尽管我们在原则 5 中已经陈述过，事实表外键不应该为空，同时在维表的属性字段中使用“NA”或另一个默认值替换空值来避免空值也是明智的，这样可以减少用户的

惑。 </p>

</blockquote>

<h3 id="原则8-确定维度表使用了代理键">原则 8 确定维度表使用了代理键</h3>

<blockquote>

<p>按顺序分配代理键(除了日期维度)可以获得一系列的操作优势,包括更小的事实表、索引以及性能改善,如果你正在跟踪维度属性的变化,为每个变化使用一个 新的维度记录,那么确实需要代理键,使你的商业用户没有初始化跟踪属性改变的设想值,使用代理也会使下游策略变化更宽松,代理也允许你使用多个业务键映射到一个普通的配置文件,有利于你缓冲意想不到的业务活动,如废弃产品编号回收或收购另一家公司的编码方案。 </p>

</blockquote>

<h3 id="原则9-创建一致的维度集成整个企业的数据">原则 9 创建一致的维度集成整个企业的数据</h3>

<blockquote>

<p>对于企业数据仓库一致的维度(也叫做通用维度、标准或参考维度)是最基本的原则,在 ETL 系统管理一次,然后在所有事实表中都可以重用,一致的维度在 整个维度模型中可以获得一致的描述属性可以支持从多个业务流程中整合数据,企业数据仓库总线矩阵是最关键的架构蓝图,它展现了组织的心业务流程和关联的维度,重用一致的维度可以缩短产品的上市时间,也消除了冗余设计和开发过程但一致的维度需要在数据管理和治理方面有较大的投入。 </p>

</blockquote>

<h3 id="原则10-不断平衡需求和现实-提供用户可接受的并能够支持他们决策的DW-BI解决方案">

则 10 不断平衡需求和现实,提供用户可接受的并能够支持他们决策的 DW/BI 解决方案</h3>

<blockquote>

<p>维度建模需要不断在用户需求和数据源事实之间进行平衡,才能够提交可执行性好的设计,更重要的是,要符合业务的需要,需求和事实之间的平衡是 DW/BI 从业人员必须面对的事实,无论是你集在维度建模,还是项目策略、技术/ETL/BI 架构或开发/维护规划都要面对这一事实。 </p>

</blockquote>

<p>转自: <a href="https://ld246.com/forward?goto=http%3A%2F%2Fwww.cnblogs.com%2Fhadoopdev" target="\_blank" rel="nofollow ugc">http://www.cnblogs.com/hadoopdev</a> </p>