数据仓库设计10原则

***Rule #1: Load detailed atomic data into dimensional structures.***

Dimensional models should be populated with bedrock atomic details to support the unpredictable filtering and grouping required by business user queries. Users typically don't need to see a single record at a time, but you can't predict the somewhat arbitrary ways they'll want to screen and roll up the details. If only summarized data is available, then you've already made assumptions about data usage patterns that will cause users to run into a brick wall when they want to dig deeper into the details. Of course, atomic details can be complemented by summary dimensional models that provide performance advantages for common queries of aggregated data, but business users cannot live on summary data alone; they need the gory details to answer their ever-changing questions.

原则1：维度表包含最细粒度:

维度模型需要最基础的原子性的细节信息来支持不可预测的筛选与汇总的商业需求。客户无需去看每个细节的记录，但是你不能武断预测他们将会只需要汇总 表。汇总性的数据，是你已经对数据使用的方式进行了预测才产生的，这样则会使得用户不能去挖掘更深层次的细节信息。当然，汇总性的摘要可以补充原子性细节 信息，汇总摘要型的模型设计的具有性能优势，但是商业用户不能仅仅使用汇总摘要信息，他们需要细节信息来回答他们永远变化的追问。

***Rule #2: Structure dimensional models around business processes.***

Business processes are the activities performed by your organization; they represent measurement events, like taking an order or billing a customer. Business processes typically capture or generate unique performance metrics associated with each event. These metrics translate into facts, with each business process represented by a single atomic fact table. In addition to single process fact tables, consolidated fact tables are sometimes created that combine metrics from multiple processes into one fact table at a common level of detail. Again, consolidated fact tables are a complement to the detailed single-process fact tables, not a substitute for them.

原则2：围绕商业流程来建立维度模型：

商业流程是你们组织机构产生的活动。他们代表了可度量的事件，例如下订单或者给客户签单。商业过程中通常捕捉到并产生与事件相关的度量性指标。这些度量将转变成为事实表，每一个商业过程将会被转变成原子的事实表。除了单个流程的事实表，整合事实表有时候在某个粒度上将多个度量，多个流程整合在一个事实表中。并且，整合的事实表是最细单个流程事实表的补充，而不是替代。

***Rule #3: Ensure that every fact table has an associated date dimension table.***

The measurement events described in Rule #2 always have a date stamp of some variety associated with them, whether it's a monthly balance snapshot or a monetary transfer captured to the hundredth of a second. Every fact table should have at least one foreign key to an associated date dimension table, whose grain is a single day, with calendar attributes and nonstandard characteristics about the measurement event date, such as the fiscal month and corporate holiday indicator. Sometimes multiple date foreign keys are represented in a fact table.

原则3：保证每个事实表与日期的维度表相联系

在原则2描述过的度量事件永远有一个日期戳的信息与之相关。无论它是每月结算的快照还是百分之一秒就会发生一次的金融交易。每个事实表至少有一个外键能够与日期维度表相连，它的粒度是衡量事件时间的属性或者非标准属性，比如是财务月或者组织的假日。有些事实表具有多个日期的外键。

***Rule #4: Ensure that all facts in a single fact table are at the same grain or level of detail.***

There are three fundamental grains to categorize all fact tables: transactional, periodic snapshot, or accumulating snapshot. Regardless of its grain type, every measurement within a fact table must be at the exact same level of detail. When you mix facts representing multiple levels of granularity in the same fact table, you are setting yourself up for business user confusion and making the BI applications vulnerable to overstated or otherwise erroneous results.

原则4：保证单个事实表的所有事实度量的粒度层级一致

所有的事实表中有三种基本粒度类型：事务型，一段时间的快照，汇总累计型快照。不管它们的粒度类型，事实表中的每个度量必须在细节水平上完全一致。当你将多层级的度量混合到一起进入同一事实表时，你将会使得商业用户迷惑，且让BI的应用程序非常脆弱并导致错误的结果。

***Rule #5: Resove many-to-many relationships in fact tables.***

Since a fact table stores the results of a business process event, there's inherently a many-to-many (M:M) relationship between its foreign keys, such as multiple products being sold in multiple stores on multiple days. These foreign key fields should never be null. Sometimes dimensions can take on multiple values for a single measurement event, such as the multiple diagnoses associated with a health care encounter or multiple customers with a bank account. In these cases, it's unreasonable to resolve the many-valued dimensions directly in the fact table, as this would violate the natural grain of the measurement event. Thus, we use a many-to-many, dual-keyed bridge table in conjunction with the fact table.

原则5：解决事实表中的多对多关系。

既 然事实表存放的是商业流程的结果。那么在事实表的外键属性中肯定存在多对多关系，例如多种产品将会在多天内被存放在多个仓库。这些外键字段将不能为空。有 时候事实表的单个事件度量对应维度表多个值，比如与一个医疗帐号与多个诊断相关或者一个银行帐号与多个客户相关。在这种情况下，将具有多个值的维度表与事 实表直接相连是没有道理的，因为，会破坏事实度量本身的自然粒度。因此，我们使用多对多，双键桥表来链接事实表。

***Rule #6: Resolve many-to-one relationships in dimension tables.***

Hierarchical, fixed-depth many-to-one (M:1) relationships between attributes are typically denormalized or collapsed into a flattened dimension table. If you've spent most of your career designing entity-relationship models for transaction processing systems, you'll need to resist your instinctive tendency to normalize or snowflake a M:1 relationship into smaller subdimensions; dimension denormalization is the name of the game in dimensional modeling.

It is relatively common to have multiple M:1 relationships represented in a single dimension table. One-to-one relationships, like a unique product description associated with a product code, are also handled in a dimension table. Occasionally many-to-one relationships are resolved in the fact table, such as the case when the detailed dimension table has millions of rows and its roll-up attributes are frequently changing. However, using the fact table to resolve M:1 relationships should be done sparingly.

原则6：解决维度表中的多对一关系

分层的，固定深度的多对一关系通常存在于一个扁平的维度表中。如果你将你生涯的大部分用于为事务处理型系统设计实体关系模型，你需要抵抗你的本能的冲动去将雪花模型的多对一的关系转化为更小的子维度；在维度模型中里单个维度表中的属性存在相对的M：1关系非常普遍，而1：1的关系，比如每个产品码对应的唯一的产品描述，也存在的。时常M:1的关系在事实表中存在，例如当一个细节的维度表中存在数百万行记录并且它的汇总的属性经常变化。然而，使用事实表解决M:1问题则是小心地进行。

***Rule #7: Store report labels and filter domain values in dimension tables.***

The codes and, more importantly, associated decodes and descriptors used for labeling and query filtering should be captured in dimension tables. Avoid storing cryptic code fields or bulky descriptive fields in the fact table itself; likewise, don't just store the code in the dimension table and assume that users don't need descriptive decodes or that they'll be handled in the BI application. If it's a row/column label or pull-down menu filter, then it should be handled as a dimension attribute.

Though we stated in Rule #5 that fact table foreign keys should never be null, it's also advisable to avoid nulls in the dimension tables' attribute fields by replacing the null value with "NA" (not applicable) or another default value, determined by the data steward, to reduce user confusion if possible.

原则7：将报表的标签与过滤器等信息存放在维度表。

这段话的意思是将描述标签或者查询的代码或符号存放在维度表里面。而避免将含糊的代码或庞大的描述字段存入事实表中；相应的，不要仅仅存放代码到维度表便假设用户不需要了解对应的描述信息，或者假设他们能够自己在BI程序中解决这些问题。如果它是行/列的标签或者下拉菜单的过滤器，则它需要被作为维度的一个属性来对待。虽然我们在原则5中阐述了事实表的外键不能为空，我依然建议尽量避免维度表的属性字段为空，或者使用”NA”这样的默认字段来填补空字段。以减少客户的困惑。

***Rule #8: Make certain that dimension tables use a surrogate key.***

Meaningless, sequentially assigned surrogate keys (except for the date dimension, where chronologically assigned and even more meaningful keys are acceptable) deliver a number of operational benefits, including smaller keys which mean smaller fact tables, smaller indexes, and improved performance. Surrogate keys are absolutely required if you're tracking dimension attribute changes with a new dimension record for each profile change. Even if your business users don't initially visualize the value of tracking attribute changes, using surrogates will make a downstream policy change less onerous. The surrogates also allow you to map multiple operational keys to a common profile, plus buffer you from unexpected operational activities, like the recycling of an obsolete product number or acquisition of another company with its own coding schemes.

原则8：让维度表使用代理键

使用无意义的，顺序赋值的代理键(除非是日期维度，这时候可接受按时间的赋予的，或者更加有意义的主键)能 够产生一系列的操作上的便利，包括更小的键以为着事实表可以更小，更小的索引，改善的性能。代理键在你需要跟踪维度属性变化的时候是必须的，每次变化将会 新建立一个维度记录。甚至商业用户无需一开始就意识到所跟踪属性是变化的，使用代理键将会使得下游的政策不那么繁琐。代理键也会允许你将多个操作主键做成 一个，从而让你免得遇到不能预料的操作难题，例如需要重利用使用的过时的产品代码键或者遇到使用自身的独特的编码体系的另外一个公司的请求。

***Rule #9: Create conformed dimensions to integrate data across the enterprise.***

Conformed dimensions (otherwise known as common, master, standard or reference dimensions) are essential for enterprise data warehousing. Managed once in the ETL system and then reused across multiple fact tables, conformed dimensions deliver consistent descriptive attributes across dimensional models and support the ability to drill across and integrate data from multiple business processes. The Enterprise Data Warehouse Bus Matrix is the key architecture blueprint for representing the organization's core business processes and associated dimensionality. Reusing conformed dimensions ultimately shortens the time-to-market by eliminating redundant design and development efforts; however, conformed dimensions require a commitment and investment in data stewardship and governance, even if you don't need everyone to agree on every dimension attribute to leverage conformity.

原则9：为整合企业数据创建一致的维度表

一致维度(或者被称为通用，主，标准，参考维度)对于建立企业级别的数据仓库具有关键性意义。通过ETL系统创建一次，然后再被多个事实表重复使用。一致维度表在维度模型里面能够提供一致性的描述属性信息，以及支持多范围的钻取，从多个商业流程中整合数据的作用。企业数据仓库总线矩阵图是关键性的蓝图架构用来表示组织的核心商业流程以及相关维度信息。重复使用一致维度极大缩短了推向市场的时间，因为减少了重复设计和维护工作。然而，即使你不需每个人在每个维度属性上达成一致，但一致性维度表需要一定的承诺以及调研数据的管理与治理策略。

***Rule #10: Continuously balance requirements and realities to deliver a DW/BI solution that's accepted by business users and that supports their decision-making.***

Dimensional modelers must constantly straddle business user requirements along with the underlying realities of the associated source data to deliver a design that can be implemented and that, more importantly, stands a reasonable chance of business adoption. The requirements-versus-realities balancing act is a fact of life for DW/BI practitioners, whether you're focused on the dimensional model, project strategy, technical/ETL/BI architectures or deployment/maintenance plan.

If you've read our [Intelligent Enterprise articles](http://www.intelligententerprise.com/experts/kimball/), [Toolkit books](http://www.kimballgroup.com/html/books.html?TrkID=IE200905_BOOK) or monthly [Design Tips](http://www.kimballgroup.com/html/designtips.html?TrkID=IE200905_DT) regularly, these rules shouldn't be news to you, but here we've consolidated our rules into a single rulebook that you can refer to when you are gathered to design or review your models.

原则10：一直去平衡需求和现实来完成可被用户接受的支持用户决策的DW/BI解决方案。

维度建模必须要持续地平衡商业用户需求以及与基于数据源所能够实现的设计的现实情况。更加重要的是，这之间存在一个可以接受的有道理的商业方案的机会。需求与现实的平衡艺术是DW/BI人的现实生活状况写照，无论你是否专注在维度建模，还是项目策略，技术/ETL/BI的架构或部署/维护方案上。