本文主要内容来自 SpriCoder的博客，更换了更清晰的图片并对原文的疏漏做了补充和修正。

本文提供了 pdf 版，以供打印： 商务智能-05-多维建模 | EagleBear2002 的博客。

1. 维度建模中的基本概念

1. 事实表
2. 维度表
3. 事实与维度的融合
    1. 星型模型
    2. 雪花模型
    3. 数据立方体

1.1 事实表

事实表是维度建模的核心和基本表。

每一事实表都对应着一个或若干个“度量值”：

1. 度量值是事实表的核心，也是趋势分析的对象
2. 通过事实表来记录维度值与度量值之间的关系

事实表中的一行对应一个度量值：

1. 事实表中的所有度量值必须具有相同的粒度
2. 粒度划分模型（粒度由小到大）：事务，周期快照，累积快照

1.1.1 事务

1. 记录的事务层面的事实，保存的是最原子的数据，又称“原子事实表”，事务事实表中的数据在事务时间发生后产生。
2. 粒度是一条记录，比如银行转账 1 块钱。
3. 更新方式是增量更新，具有稀疏性质，因为很多的事实可能不同时发生，是稀疏表，只有当天发生了操作才有记录。

1.1.2 周期快照

1. 以具有规律性的、可预见的时间间隔来记录事实，统计的是间隔周期内的度量统计。时间间隔：年、月、日等。
2. 周期快照没有粒度的概念，是周期+状态度量的组合，其粒度是每个时间段一条记录。
3. 周期快照事实表维度少于事务事实表，但是记录的事实要多于事实事务表。
4. 更新方式是增量更新，是稠密表，哪怕当天没操作也会有记录
5. 用于记录重复的可预测时间间隔的事实，比如每月账单。

1.1.3 累积快照

1. 累积快照事实表存储的是不确定的周期的数据，他完全覆盖了一个事务或一个产品的生命周期的时间跨度，通常有多个日期字段来记录关键时间点，比如订单的付款时间、发货时间、收货时间等。
2. 累积快照事实表只会有一条记录，数据会一直更新到过程结束。
3. 通常包含很多日期字段，并且会有一个用户只是最后更新日期的附加日期字段。
4. 用于记录较短周期，有着明确开始和结束状态等多个状态的过程。

更多阅读： 事实表的分类：事务事实表，周期快照事实表，累积快照事实表

表 11.7　三种事实表的比较

	事务事实表	周期快照事实表	累积快照事实表
时期/时间	离散事务时间点	以有规律的、可预测的间隔产生快照	用于时间跨度不确定的不断变化的工作流
日期维度	事务日期	快照日期	相关业务过程涉及的多个日期
粒度	每行代表实体的一个事务	每行代表某时间周期的一个实体	每行代表一个实体的生命周期
事实	事务事实	累积事实	相关业务过程事实和时间间隔事实
事实表加载	插入	插入	插入与更新
事实表更新	不更新	不更新	业务过程变更时更新

1.1.4 事实表中的度量值

最常用的度量值：数值类型，方便处理

度量值通常是一个可以连续取值的量，很少采用文本形式的度量值，因为文本没有办法处理。

三种类型的度量值：

1. 可做加法运算
2. 可沿着某些维度做加法运算：比如每天剩下的零钱按照时间加。
3. 不能做加法运算
    1. 计数统计
    2. 计算平均值
    3. 取样统计

无法量化不是量化本身的问题，而是体系的问题。

1.1.5 事实表中的关键字

每个事实表都有两个或两个以上的外关键字（Foreign Key）

1. 通过外关键字建立事实表与维表之间的联系，从而可以通过维度表来存取事实表中的度量值
2. 可以由外关键字的组合构成事实表的主关键字（Primary Key）

日销售情况事实表

日期关键字（FK）
产品关键字（FK）
商场关键字（FK）
销售量
销售额

3. 销售量和销售额是度量值，可以体现出其关联关系。
4. 多少个维度就有多少个外关键字。
5. 事实表中单独的 Primary Key 是没有意义的，但有时候为了解决问题我们可能会引入新的关键字。

1.2 维度表

1. 维度表是事实表的入口，为用户提供了使用数据仓库的接口。
2. 维度表中的维度属性通常用于定义事实表上的查询条件，也可作为定义报表和统计查询的“列”。
3. 维度表的定义通常包括：
    1. 尽可能多的列：和事实表的差别
    2. 相对少的行（相对于事实表）

1.2.1 维度表的属性组成

操作型数据环境中不会有这么多数据：只有部分数据是有意义的。

产品关键字(PK)	重 量
产品描述	重量单位
SKU编号	储藏类型
商标描述	货架类型
分类描述	货架宽度
部门描述	货架高度
包装类型描述	货架深度
包装尺寸	……
含脂量描述	
食物类型描述	

1.2.2 维度属性

通常是文本数据，或者是离散数据

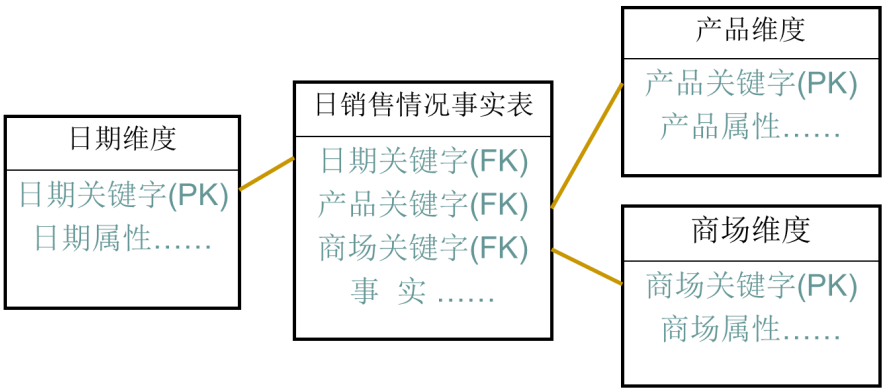
尽量减少使用编码属性：对于人而言不好理解

维度属性与度量值（属性）的区别：

1. 度量值属性：有许多取值可能并可以参与统计运算的属性
2. 维度属性：
    1. 离散的或取值可能不多的属性
    2. 取值不变或很少产生变化的属性
    3. 从不参与统计计算但经常用作查询条件的属性

1.3 事实与维度的融合

将事实表及其相关的维表通过关键字进行连接



1.4 维度建模案例

1. 维度建模案例之一：零售营销
2. 维度建模案例之二：库存管理
3. 维度建模案例之三：订单管理
4. 维度建模案例之四：客户关系管理

注：上述案例及其图、表均引自：“数据仓库生命周期工具箱：设计、开发和部署数据仓库的专家方法”一书

2. 维度建模案例之一：零售营销

2.1 维度建模的设计过程

1. 选取要建模的业务处理过程（分析型）：根据分析需要
2. 定义业务处理的粒度：确定事实表中每一行的度量值的取值粒度，和多维度相关。
3. 选择事实表中的维度（事先已经建立）：设计中一定是先设计好维度
4. 选择事实表中的度量值
    1. 以分析对象为依据
    2. 可以有多个度量值

2.2 零售营销的需求分析

1. 数据的入口（数据驱动）：前台 POS 机和后台的货物入库
2. 管理决策需要（面向主题）：定价和促销

2.3 维度建模的设计过程

选取业务处理：在什么促销条件下，在什么样的日子里，在什么商店，正在销售什么样的商品

定义粒度：

1. POS 事务的单个商品条目结构
2. 最初粒度的选择与可以执行的分析操作有关

威达软件

日期：2013-01-15 17:00:50
单据号：NO20130115000004
收银员：888

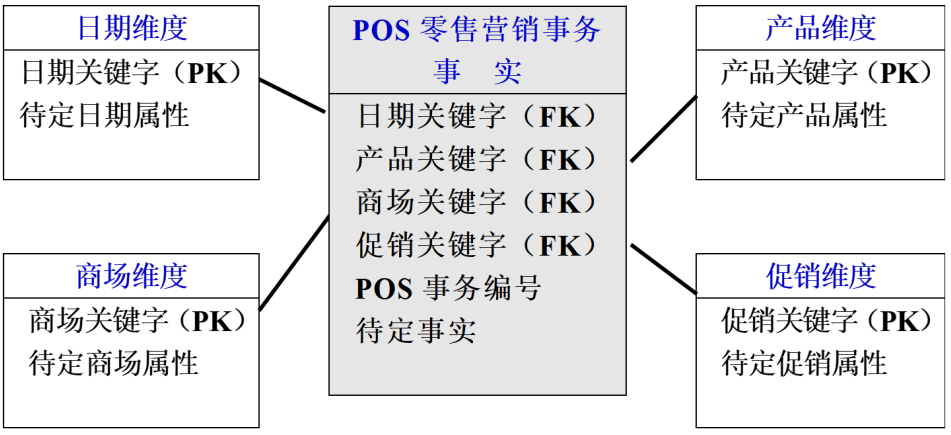
商品名称	条码	数量	金额
爱尚非蛋糕(草莓味)40g	6921682823487	1	3.00
阿尔卑斯棒棒糖(混合味)	6911316540309	1	0.60
阿尔卑斯双享棒棒糖(果味)	6911316375161	1	1.20
阿甘牦牛肉干麻辣味	6922898337621	1	15.60
棒棒娃牛肉粒118g	6920601702223	1	21.00
达利园蛋黄派250g	6911988006783	1	7.20

合计金额：42.80　优惠金额：5.80
实收金额：50.00　找零金额：7.20
合计件数：6.00
客户姓名：张金　卡号：100003

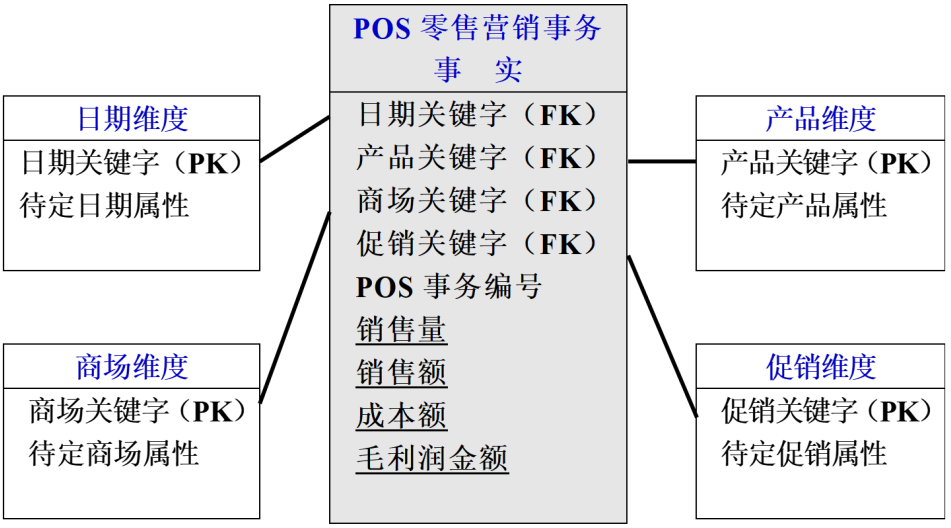
地址：福建莆田威达软件
电话：18250500838

请保留小票，产品凭小票及完好包装可在七天内换货！谢谢惠顾！

2.3.1 选定维度



2.3.2 确定事实



通过计算而获得的可加性度量值也可以物理存储在事实表中，如：毛利润金额（毛利润金额 = 销售额 - 成本额）

不具有可加性的计算结果则应该由分析展现工具在访问过程中进行计算，如：毛利润率，单价等

在不同环境下，成本可能是分散的。

原生性：要什么有什么，导出性数据考虑一下对时间和性能的改善。

POS 零售营销事务 事　实
日期关键字（FK）
产品关键字（FK）
商场关键字（FK）
促销关键字（FK）
POS 事务编号
销售量
销售额
成本额
毛利润金额

2.3.3 维度设计-日期维度

1. 日期维度是每个数据仓库必须具备的维度
2. 日期维度表可以事先建立：可以预先建立好 5 到 10 年的日期维度值

2.3.3.1 日期维度中的属性

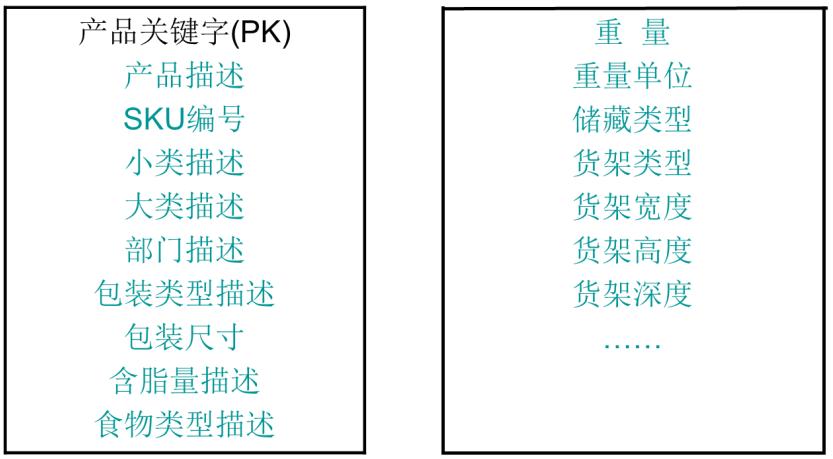
日期关键字（PK）	日历日期编号	日历周结束日期
日期完全描述	日历周编号	年度日历周数
星期	日历月编号	日历月名
纪元日编号	财政月日编号	年度日历月数
纪元周编号	周末指示符	日历年月（YYYY-MM）
纪元月编号	月末指示符	

日历季度	财政周	节假日指示符
日历年季度	年度财政周数	星期指示符
日历半年度	财政月	销售旺季
日历年	年度财政月数	重大事件
	财政年月	SQL日期标记
	财政季度	……
	财政年季度	
	财政半年度	
	财政年	

2.3.3.2 日期维度表

日期关键字	日期	日期完整描述	星期	日历月	日历年	财政年月	节假日指示符	周日指示符
1	01/01/2002	2002 年 1 月 1 日	星期二	1 月	2002 年	F2002-01	节日	平日
2	01/02/2002	2002 年 1 月 2 日	星期三	1 月	2002 年	F2002-01	非节日	平日
3	01/03/2002	2002 年 1 月 3 日	星期四	1 月	2002 年	F2002-01	非节日	平日
4	01/04/2002	2002 年 1 月 4 日	星期五	1 月	2002 年	F2002-01	非节日	平日
5	01/05/2002	2002 年 1 月 5 日	星期六	1 月	2002 年	F2002-01	非节日	周日
6	01/06/2002	2002 年 1 月 6 日	星期日	1 月	2002 年	F2002-01	非节日	周日
7	01/07/2002	2002 年 1 月 7 日	星期一	1 月	2002 年	F2002-01	非节日	平日
8	01/08/2002	2002 年 1 月 8 日	星期二	1 月	2002 年	F2002-01	非节日	平日

2.3.4 维度设计-产品维度



2.3.4.1 产品维度表（部分）

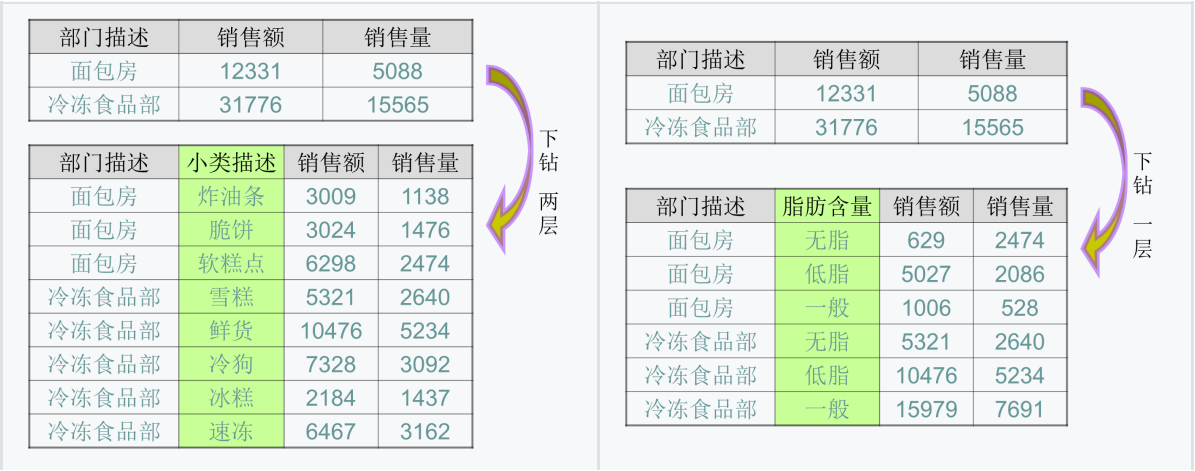
产品关键字	产品描述	小类描述	大类描述	部门描述	含脂量
1	低碱烤肉包	烧烤	面包	面包房	低脂
2	松脆全麦切片	松脆	面包	面包房	一般
3	松淡全麦切片	松脆	面包	面包房	低脂
4	脱脂小挂卷	松软	甜面包	面包房	无脂
5	2 加仑装美食香料	冷裹品	冷冻点心	冷冻食品部	无脂
6	1 品脱装黄油软奶桃	鲜类	冷冻点心	冷冻食品部	低脂
7	1/2 加仑装巧克力美食	冷冻	冷冻点心	冷冻食品部	一般
8	1 品脱装草莓冰淇淋	冰冻	冷冻点心	冷冻食品部	一般
9	冰淇淋三明治	冰冻	冷冻点心	冷冻食品部	一般

2.3.4.2 产品维度的属性

在产品维度表中存在着两类属性：

1. 产品的多级体系划分属性（构成属性体系结构）：
    1. SKU 编号→小类描述→大类描述→部门描述
    2. 从左到右，每一级都是“多对一”的对应关系，从而构成一个关于商品的分类体系
2. 其它描述属性：
    1. 包装类型，脂肪含量，……

2. 这类属性并不是产品体系的组成部分，但可以与产品的体系划分属性组合在一起进行有意义的分析应用



部门描述	销售额	销售量
面包房	12331	5088
冷冻食品部	31776	15565

下钻两层

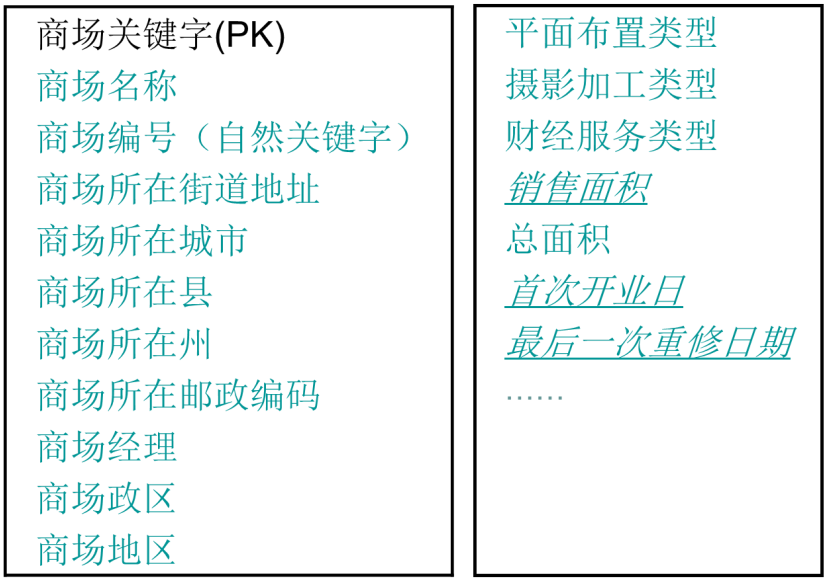
部门描述	小类描述	销售额	销售量
面包房	炸油条	3009	1138
面包房	脆饼	3024	1476
面包房	软糕点	6298	2474
冷冻食品部	雪糕	5321	2640
冷冻食品部	鲜货	10476	5234
冷冻食品部	冷狗	7328	3092
冷冻食品部	冰糕	2184	1437
冷冻食品部	速冻	6467	3162

部门描述	销售额	销售量
面包房	12331	5088
冷冻食品部	31776	15565

下钻一层

部门描述	脂肪含量	销售额	销售量
面包房	无脂	629	2474
面包房	低脂	5027	2086
面包房	一般	1006	528
冷冻食品部	无脂	5321	2640
冷冻食品部	低脂	10476	5234
冷冻食品部	一般	15979	7691

2.3.5 维度设计-商场维度



使用维度支架的方式连接，也就是首次开业日会作为 FK 连接到另一张表上。

2.3.5.1 商场维度的属性

1. 销售面积：
    1. 数值类型的字段，且是跨商场可相加的
    2. 但由于这是商场的一个不变属性，且大都用做查询分析报表的列标题，所以还是安排在商场维度表中
2. 首次开业日 与 最后一次重修日期
    1. 其取值来自于定义在前述的“日期维度”表上的视图
    2. 采用维度支架加以实现

2.3.6 维度设计-促销维度

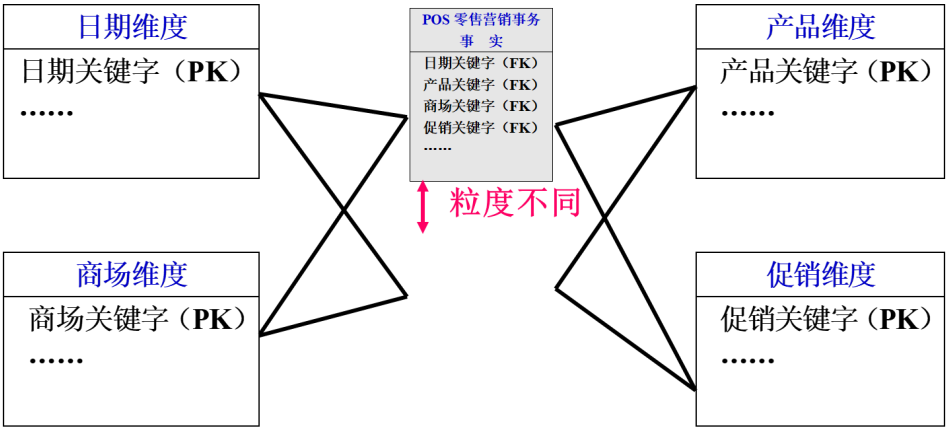
1. 对商品促销活动的评判因素：
    1. 促销商品的销售分析：
        1. 在促销期间是否出现增长?
        2. 在促销进行之前或随后是否减少?
    2. 相邻或同类的其它商品的销售是否出现相应的降低情况?

3. 与促销商品同类的所有商品的销售是否出现总体增长?
4. 促销是否赢利? （考虑促销活动自身的成本）

2. 存在多种不同的促销形式（降价，广告，展销，优惠券，...）
    1. 每一种类型的促销活动可以单独形成一个促销维度表
    2. 也可以将所有的促销活动揉合在一个促销维度表中（如下图）
    3. 促销关键字（PK）：促销名称、减价类型、促销媒体类型、广告类型、展览类型、优惠卷类型、广告媒体类型、展览提供者、促销价、促销起始日期、促销结束日期等
3. 维度的组合：
    1. 参与组合的维度高度相关，组合起来的维度就不会比分开的维度大许多
    2. 组合起来的维度能够高效地进行浏览
4. 维度的分散：
    1. 在用户分开考虑时，分开的维度更加容易理解
    2. 独立维度的管理对于组合维度来说，更加直截了当
5. 不在促销范围之内的商品销售事实如何在事实表中表示?
    1. 在促销维度表中定义一个特殊的“行”
    2. 在事实表中，所有没有参与促销活动的行（产品销售事实）都引用该特殊的“行”，以表示该维度值对事实表中的当前行不可用，也就是有为空的感觉。

POS 零售营销事务 事　实
日期关键字（FK）
产品关键字（FK）
商场关键字（FK）
促销关键字（FK）
POS 事务编号
销售量
销售额
成本额
毛利润金额

1. 在商品促销效果分析中，还有一类问题是上述的零售营销模型无法回答的：什么样的促销产品还没有卖出去?
2. 需要另外一个非事实型事实表来记录每天每件商品的促销活动
    1. 促销范围事实表
    2. 不存在度量指标（仅记录各个维成员之间的关系）
    3. 为每天中每个商场的每个促销产品创建一行



2.3.7 维度设计-POS 事务编号

1. 退化维度：维度表为空，具体的维度值直接存放在事实表中
2. 例如：
    1. 事务编号
    2. 订单编号
    3. 发票编号
    4. 提货单编号
3. 可能不同的部分关键字之间是存在内在关联的。
4. 关联数据分析：比较重要的就是购物篮问题，也就是购物篮中哪些商品是关联的。
5. 必须要有事务编号维度来保证维度 POS 事务编号是必要的。

POS 零售营销事务 事　实
日期关键字（FK）
产品关键字（FK）
商场关键字（FK）
促销关键字（FK）
POS 事务编号
销售量
销售额
成本额
毛利润金额

2.4 零售示例的多维模型

事实表：

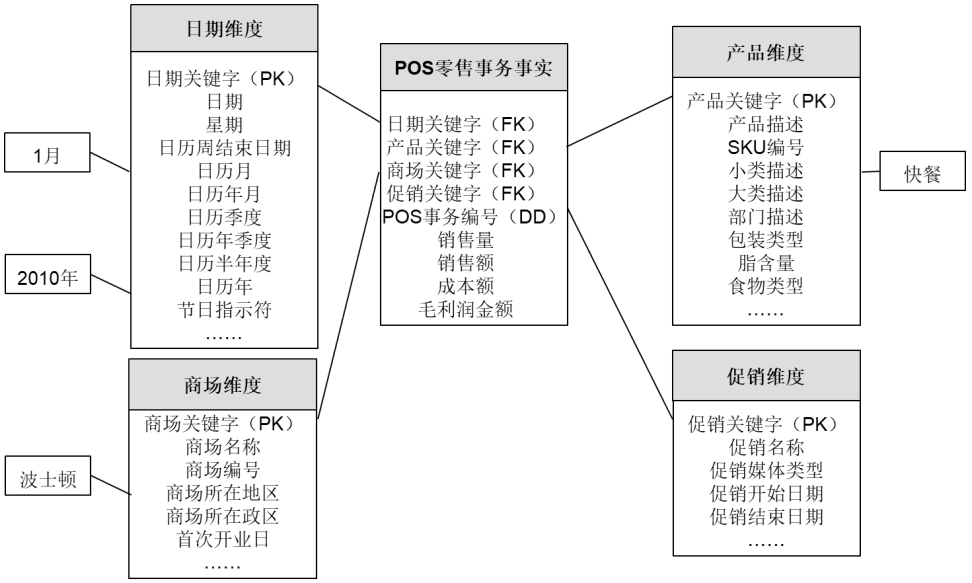
1. 销售量，销售额，成本额，毛利润
2. 促销记录

维度表：

1. 日期，商场，产品，促销
2. 退化维度：POS 事务编号

在零售多维模型上的数据访问：通过维度表中的维度属性访问事实表

2.5 多维模型访问方式

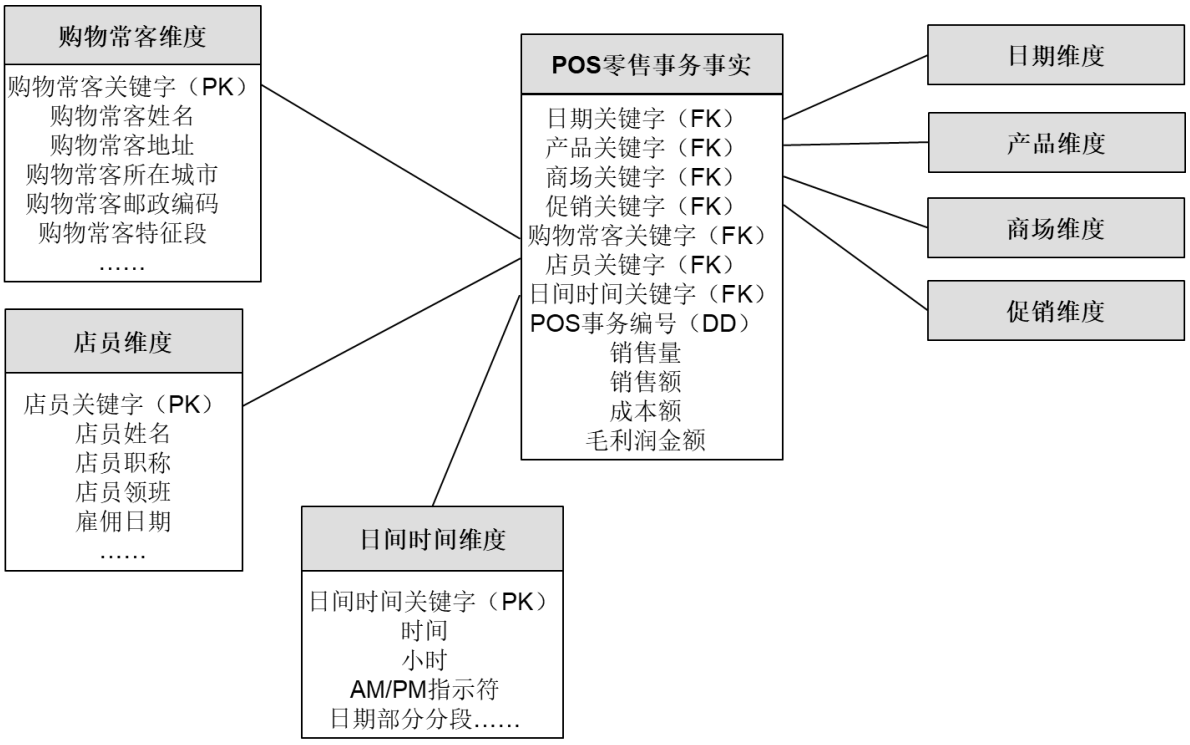


1. 首先对具体的维度的部分进行制定。
2. 使用二维数据模型来完成了多维数据模型的描述。

2.6 模型的演化

1. 新的维度属性（例如，产品的全新描述属性）：
    1. 加入时间点前的，使用“不可用”进行填充
    2. 比如添加了是否转基因的条目，那么之前为空的部分，就用“不可用”进行填充。
2. 新的维度（例如，会员、店员、日间时间等分析的新角度）：
    1. 新的维表
    2. 在事实表中填加新的外关键字
3. 新的度量值事实：
    1. 添加新的度量值属性
    2. 需要考虑事实表粒度
4. 维度变得具有更多的粒度性：
    1. 建立新粒度层次上的维度表
    2. 可能带来新粒度层次上的事实表，从而需要同时建立新的维度表和事实表，新的维度表会添加新的事实。
5. 全新的数据源的加入，会同时牵涉现存的维度和不能预见的新维度：
    1. 新数据源几乎总是拥有自己的粒度和维度
    2. 建立新的事实表和维度表

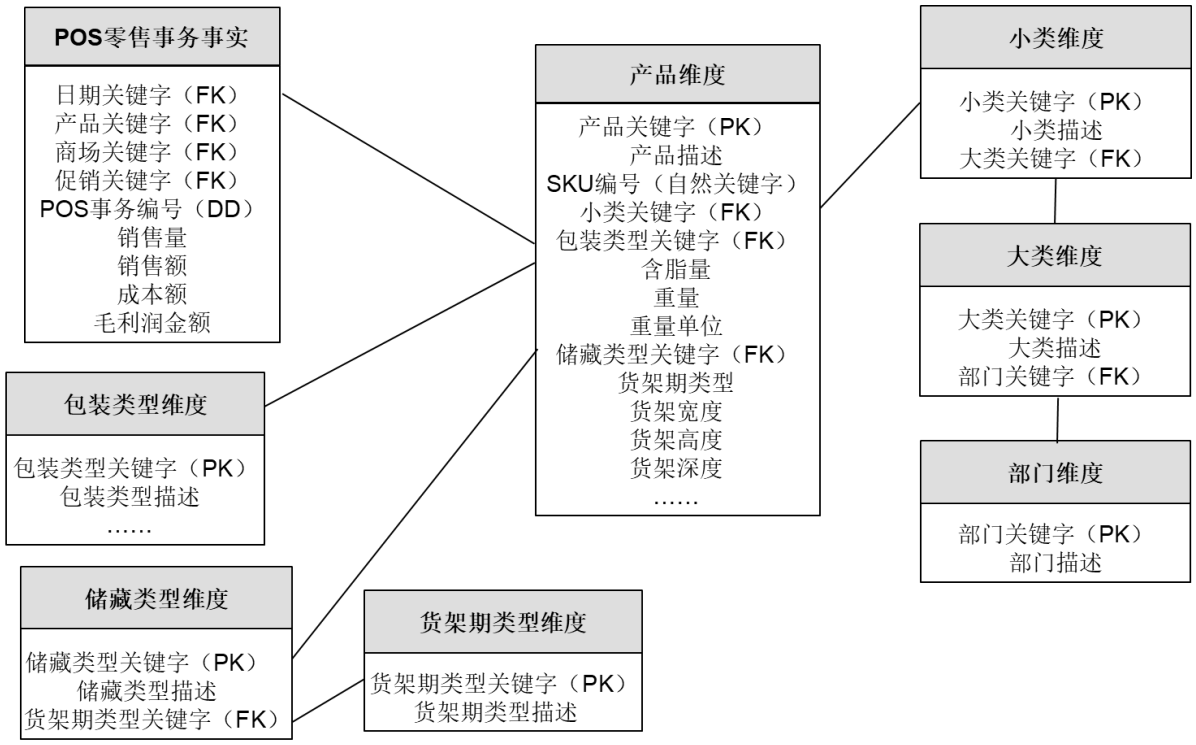
2.6.1 新加入的维度



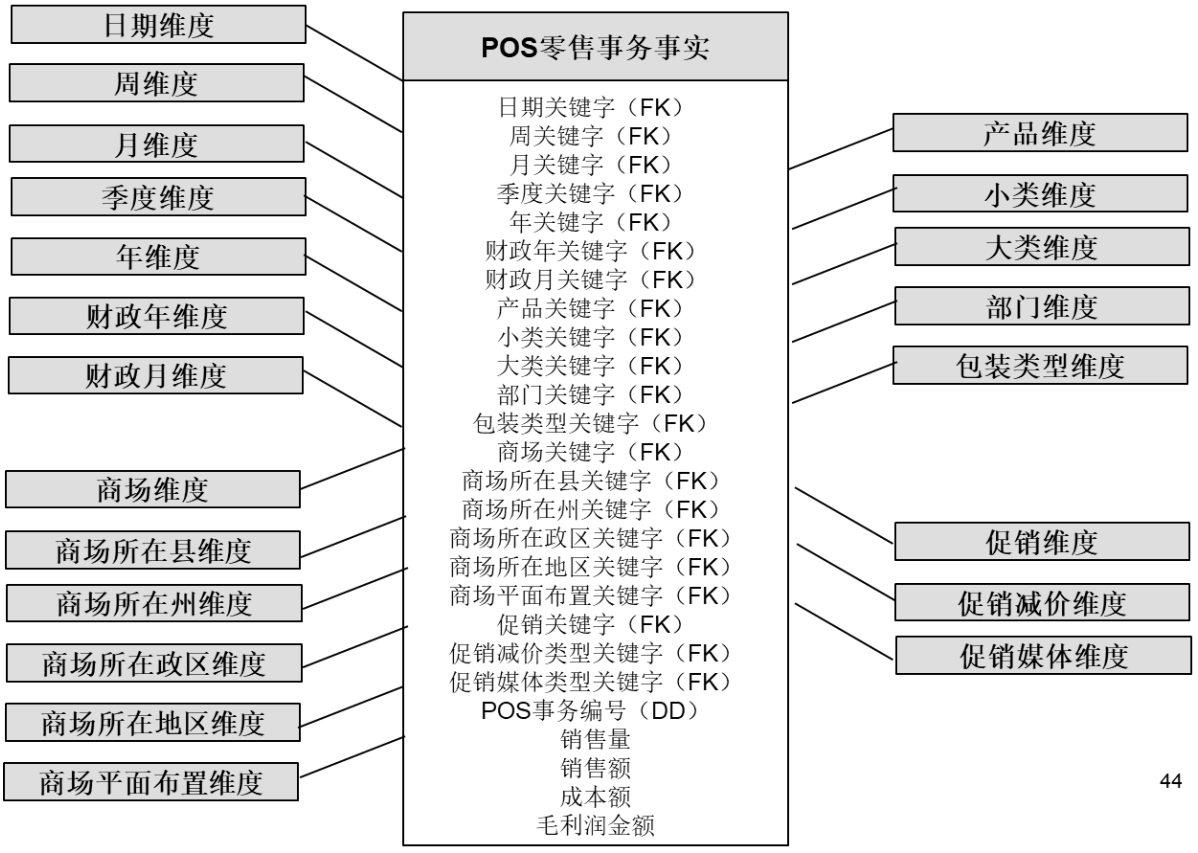
事实表的粒度设计将影响到是否易于加入新的维度。

2.6.2 维度的规范化处理

规范化	非规范化
雪花模型	星形模型
复杂的表关系	简单的表关系
节省存储空间	记录之间存在数据冗余
连接的复杂，高开销	连接简单，低开销
低维度浏览能力	高维度浏览能力
不支持物理加速技术	支持物理加速技术



2.6.3 避免维度使用过多（蜈蚣状事实表）



2.6.4 维度表中关键字的设计

代理关键字，避免直接使用操作型数据作为维度表和事实表的主关键字和外关键字

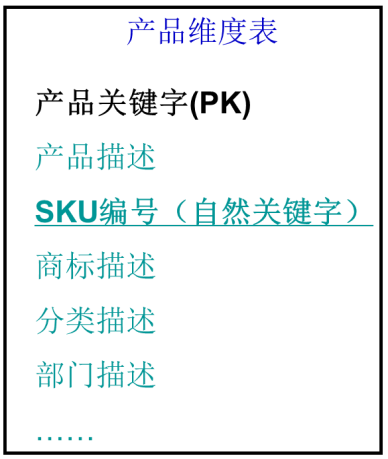
1. 可以缓冲操作型数据的变化对数据仓库数据的影响
2. 性能优势：自然关键字不一定为数值类型，而如果是文本类型，则容易导致效率较低。
3. 操作型数据可能无法作为关键字：数据库中自然关键字可以唯一确定，但是在数据仓库中则不一定，比如化学实验中的调单，需要纸质和电子同时保存，具有时效性 3 年，印刷一批单号为 1-100,000 的单子，可以使用 5 年，但是在操作型数据环境中可以保证唯一，但是进入到数据仓库中后，是无法按照单号唯一确定。

4. 日期维度的特殊要求：有一些日期在真实场景中是不存在的。

5. 历史一致性：两个关键字的两条记录对应同一实体在不同时间段的情况

修改事实表中的关键字，则同时需要修改维度表和事实表，代价比较大。

我们选择使用产品关键字（代理关键字），而不是自然关键字。



2.6.4.1 日期维度的特殊要求

SQL 日期不能为“日期待定”或“日期不可用”

1. 日期待定：事件会发生，但是不确定时间
2. 日期不可用：事件不会再发生了

日期维度的代理关键字应当按照有意义的连续次序进行分配：

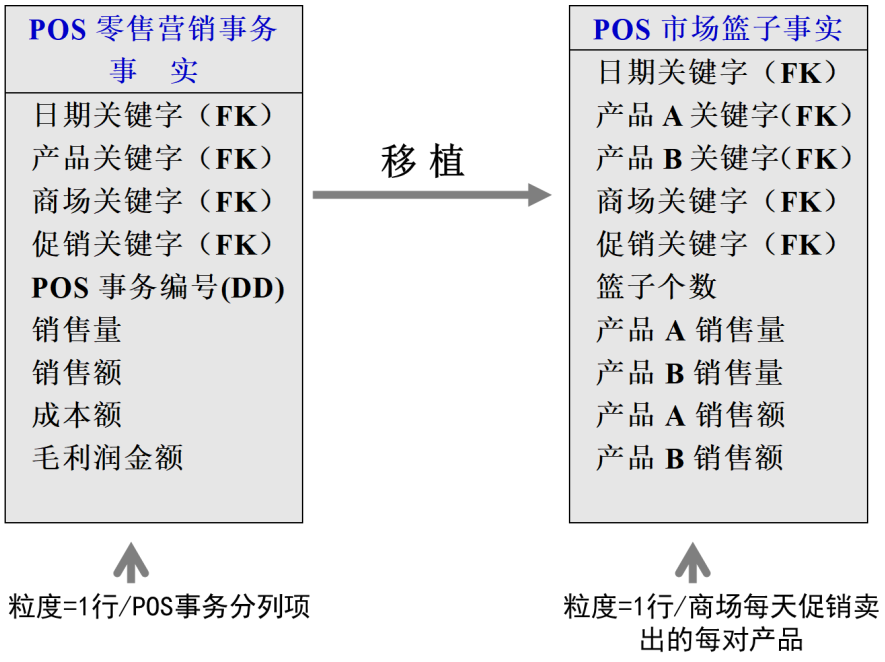
1. 允许在日期关键字基础上进行物理分区和索引
2. 1 月 1 日→1，1 月 2 日→2，2 月 1 日→32
3. YYYY-MM-DD，是不合适的做法

2.6.5 市场篮子分析

不使用 OLAP 或者数据挖掘工具。

事实表的抽取：

1. 从零售营销事实表中抽取形成新的事实表，以实现新的分析应用
2. 例：商品促销活动实施效果分析



比较 A 和 B 可以找到关联。

N 个产品，$N \times (N-1)$ 种组合。解决方式：使用粗过滤来过滤掉部分

1. 领域知识支持
2. 层次式的分析：
    1. 类别（25×25）
    2. 商标（500×500）
    3. 产品（10000×10000）

2.7 总结

1. 维度建模时的步骤
2. 找到最简单的模型进行构建

3. 维度建模案例之二：库存管理

3.1 库存管理维度模型

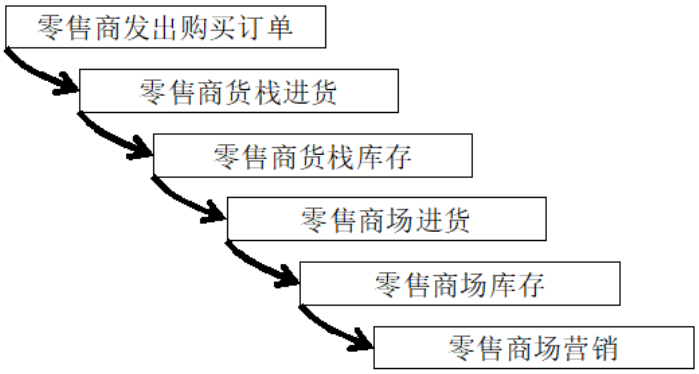
内容：用于大型杂货连锁店营销事务的维度模型

主要概念：

1. 值链
2. 三种事实表模型：事务，周期快照，累积快照
    1. 半加型事实
    2. 增强型库存事实
3. 数据仓库总线结构与矩阵
4. 一致性维度与事实

3.2 值链

1. 由企业的关键业务组成
2. 值链确定了企业主体活动的自然逻辑流程，并不是商业智能中特有的概念



零售商值链的子集

3. 其中的每一步业务处理都将产生大量的周期性事务记录（来自企业自身的业务处理系统）
4. 决策支持系统的首要目标是监控关键处理过程的性能结果
    1. 其分析的依据是来自于每一步业务处理过程的事实表
    2. 从每一步业务处理过程的业务数据库中可以衍生出一个或多个事实表
5. 操作型数据环境认为重要，则在数据仓库中被认为是重要的概率比较高。

3.3 事实表粒度模型

三种互补的库存事实表粒度模型：

1. 库存周期快照（粒度最粗）：
    1. 定期生成每种商品的库存水平（数量）
    2. 可以使用冗余的数据存储方式来解决
2. 库存事务：记录影响库存水平的主要因素，包含商品的进/出仓库等事务
3. 库存累积快照：记录每件商品的分发历史，直至其离开仓库为止

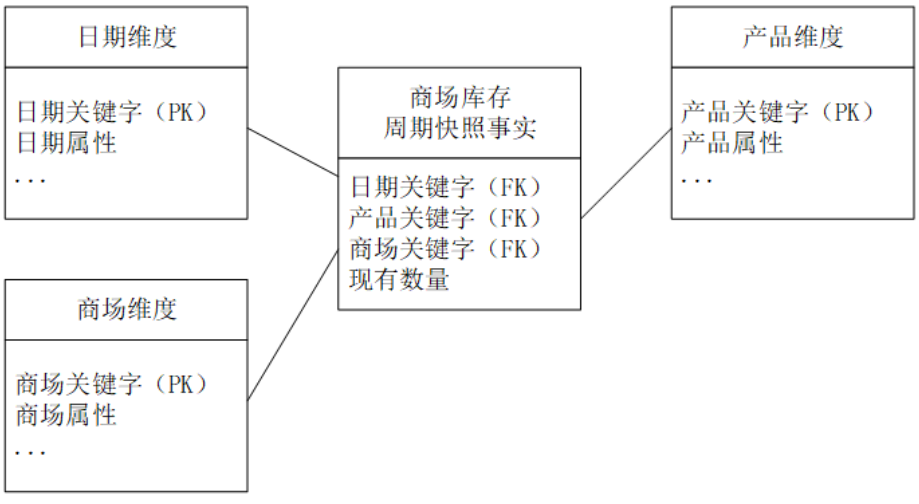
3.3.1 库存周期快照

3.3.1.1 目标

1. 确保合适的商场在合适的时间中存在合适的商品
2. 可最大限度地减少脱销现象，并减少存货维护的总体开销

3.3.1.2 四步维度建模

1. 零售商需要具备通过产品和商场分析出每天手头库存水平的能力
2. 四步维度建模
    1. 业务处理过程：零售商场的库存
    2. 粒度：每个商场每天每种商品的库存
    3. 维度：最初的维度选择
        1. 日期、商场、商品
        2. 促销：如果我们认为这是有用的，那么我们可以这样做。
    4. 事实（度量值）：库存数量

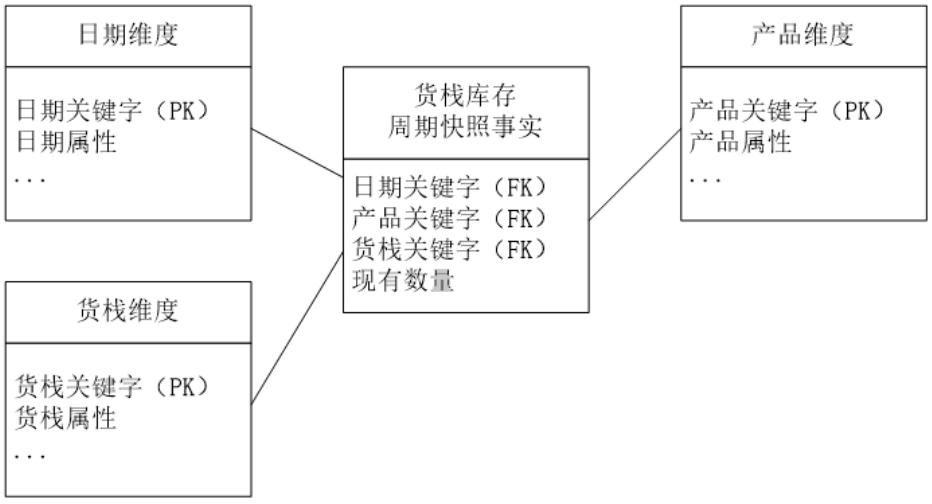


商场库存周期快照方案

3. 维表设计：
    1. 日期维度表同“案例一，零售营销”中的日期维度表保持一致（公共维度）
    2. 产品与商场维度也可以保持一致（公共维度）
4. 也可以根据实际的分析需求进一步引入其他属性（公共维度本应考虑）
    1. 产品维度：最小重购数量

2. 商场维度：冷冻、冷藏面积

5. 万一维度有缺失，则进行相应的演化即可



货栈库存周期快照方案

数据结构一样不代表数据一样。

3.3.1.3 库存周期快照事实表与销售事务事实表的区别

1. 销售事实表是稀疏的，而库存事实表则是稠密的
    1. 在销售事实表中记录每天实际发生的商品销售情况
    2. 而库存事实表则需要记录每天、每种商品、在每个商场的库存情况（不管库存是否发生了实际的变化）
2. 解决办法：
    1. 随着时间的推移可降低周期快照的频度
    2. 最近 60 天内的以天为粒度单位的周期快照
    3. 最近 3 年内的以周为粒度单位的周期快照

3.3.1.4 半加型事实

1. 只在部分维度上具有可加性的度量值被称为“半加型事实”
2. 在商品营销中，绝大部分的度量值在所有的维度范围内都具有极好的可加性
3. 在库存快照模型中，“库存量”可以跨“产品”或“商场”进行汇总（具有可加性），但不具有跨“日期”的可加性
4. 几种常见的半加型事实：
    1. 库存数量，银行帐户余额，温度，水位，含量.....
    2. 用于记录静态水平的度量值在跨日期维度以及可能的其它维度范围内都是不可加的
        1. 对于不可加的度量值，可用的常用聚集方法如平均、统计
        2. 不能简单地利用 SQL 中的 AVG 函数来完成这样的平均、统计计算工作

3.3.1.5 分析操作

如果扩充事实表，则可以提供更多的分析操作：

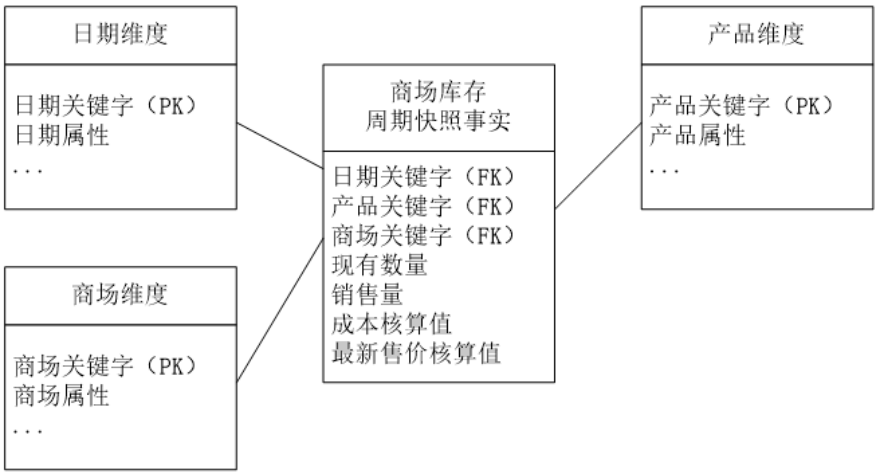
1. 周转次数
    1. 日周转次数：$\frac{\text{当日销售量}}{\text{当日持有量}}$

2. 年周转次数：$\frac{\text{年销售总量}}{\text{年平均持有量}}$

2. 日供给次数：$\frac{\text{平均持有量}}{\text{平均销售量}}$

3. 库存毛利润 GMROI：

$$GMROI = \frac{\text{总销售量} * (\text{最新售价核算值} - \text{成本核算值})}{\text{日平均持有量} * \text{平均售价核算值}}$$



支持GMROI的改进型库存周期快照方案

3.3.1.6 事实表扩充

1. 库存数量（持有量，现有量）
2. 销售量：在三个维度之间都是可加的
3. 成本核算值：在三个维度之间都是可加的
4. 最新售价核算值：在三个维度之间都是可加的

处于同一张事实表中的上述度量值需要具有统一的统计粒度。

如 GMROI 的计算分量处于不同的事实表，并拥有不同的粒度，则需要分析展现工具进行额外处理

3.3.2 库存事务

库存周期快照无法提供如下的分析操作，即没有办法获取以下的部分：粒度比较细的没有办法回答。

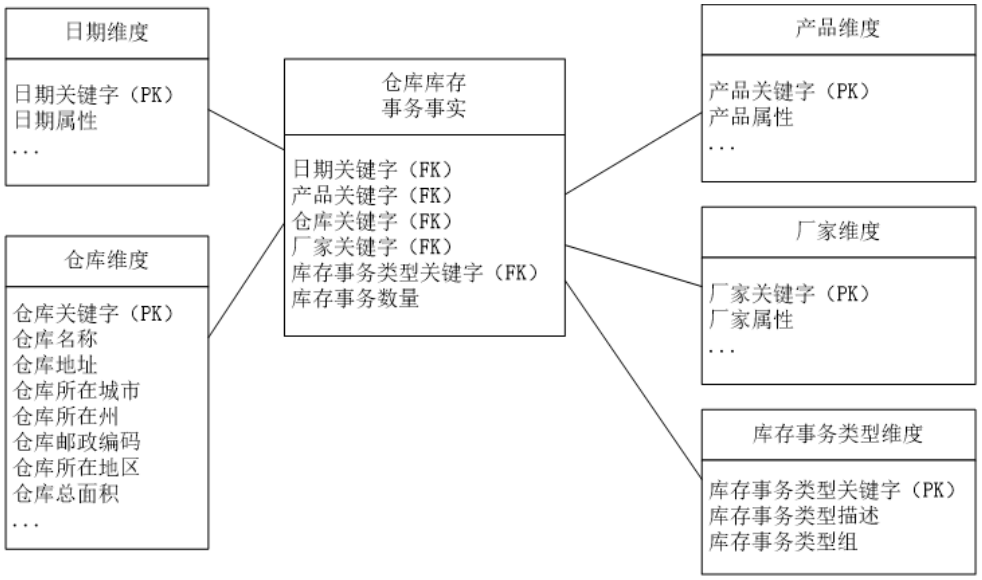
1. 发生过多少次产品入柜以后又在同一天的不同时间将它取出的情形？无法区分开是没有了，还是快速被取出了
2. 从某厂家那里接收过多少次分开装运的货，以及是什么时候收到的？
3. 哪些产品是由于出现多次检验不合格而导致向厂家退货的？

频度测算和具体事务类型的计算需要库存事务模型的支持。

常见的库存事务类型：产品接收、产品送检、对检验合格的产品进行分发、将检验不合格的产品退给厂商、产品入柜、产品销售审批、产品出柜、运输前的产品包装向顾客发货、从顾客那里回收产品、对回收产品进行封存、从库存中删除产品。

库存事务记录：日期，产品，仓库，厂家，事务类型，数量（影响库存总量的值）。

事实表的粒度：每个库存事务对应着事实表中的一行



仓库库存事务模型

3.3.3 库存累积快照

模型思想：

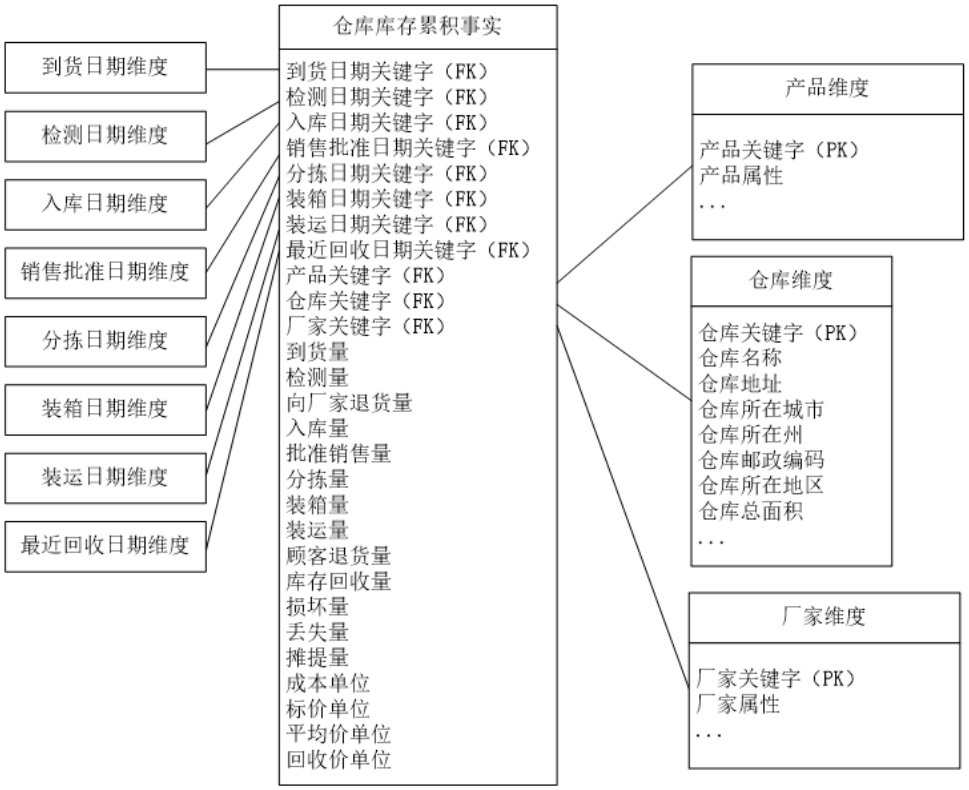
1. 在事实表中，为每种特定商品的入库装运给出相应的一行记录
2. 对产品装载处理情况的跟踪放在事实表的单行记录中，直到产品离开仓库为止，也可以实现对单件/批商品的处理情况进行跟踪（按照生命周期来开展）

特点：

1. 库存累积快照事实表中存在多个取值为日期的外关键字
2. 需要对事实表中的每一行进行多次的访问和修改操作
3. 很少用于长期运行而需要不断进行补充的库存处理

假定数据进入数据仓库的同时，伴随着一系列良好定义的活动或重要事件：接受、检验、入柜、销售批准、选取、装箱、起运。

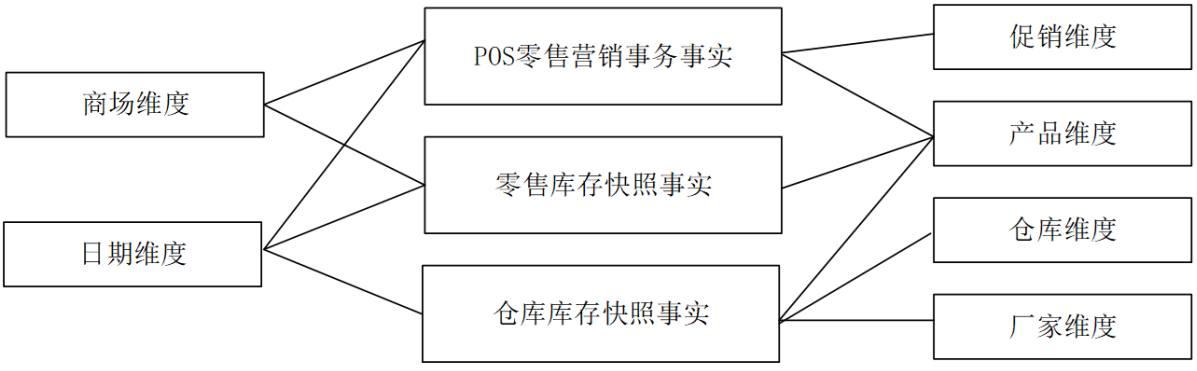
一般希望数据仓库中的数据是只读的。



仓库库存累积快照

3.4 主题的集成

1. 集成的目的：
    1. 跨主题范围的数据查看分析。
    2. 可以使得来自不同的主题的度量值可以被组合到单个分析任务中去。
2. 集成的方式：共享公共的维度设计。
3. 目前我们认为公共维度就是完全一样的，在第一个实例中我们已经完成了 POS 零售营销事务事实表。



主题之间的维度共享

3.5 数据仓库总线结构

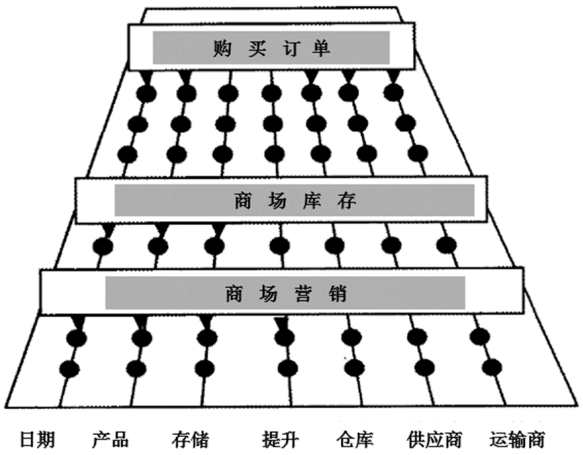
3.5.1 一种可以按增量开发方式分步建造企业数据仓库的方法

1. 计算机中的总线
2. 通过为数据仓库环境定义标准的总线接口，独立的数据集市就可以由不同的开发小组在不同的时间进行实现。只要遵循这个标准，独立的数据集市就可以插入到一起并有效地共享，比如数据集市：数据集市是不一定需要实施的，总线知识我们进行概念数据模型构建时的规范

3.5.2 一组综合的具有一致性的共用维度

1. 通过设计出一整套在企业范围内具有统一解释的标准化维度与事实，从而可以对企业数据仓库的建设任务进行合理的分解，由不同的开发小组分阶段，或并行地进行数据仓库的建设
2. 采用总线体系结构可以独立于技术手段和数据库平台

3.6 数据仓库总线矩阵



查看主题的上下文：

1. 如果这个主题并不是被公用的，那么问题不大
2. 如果这个主题会被很多的主题使用，那么需要仔细设计

3.7 数据仓库总线矩阵

	公共维度							
主题	日期	产品	商场	促销	仓库	厂家	合同	发货人
零售营销	x	x	x	x				
零售库存	x	x	x					
零售交货	x	x	x					
仓库库存	x	x			x	x		
仓库交货	x	x			x	x		
购买订单	x	x			x	x	x	x

矩阵的行：对应着主题

1. 如果数据来源不同，功能不同，或者矩阵行代表的内容无法在单个迭代过程中合理完成，就应当创建独立的矩阵行
2. 不能够再次进行细分的公共维度模型。
3. 每一个主题都会包含三部分：
    1. 事务是什么
    2. 累积快照
    3. 周期快照

矩阵的列：对应着共享的公共维度。

总线矩阵就是在规划阶段使用的。

规划主题过程中必然不可能一蹴而就，不可能第一次使用很多的主题全部完成，我们使用原型法，先使用起来。

我们倾向于将一个分析域的主题放在一起进行构建，当我们成功构建分析域后，则其分析这个局部是最优的（也就是将最相似的主题放在一起）

3.8 一致性维度

一致性维度是进一步开发总线结构数据仓库系统的基础

1. 要么是同一的，要么是具有最佳粒度与细节性的维度在严格数学意义上的子集
2. 一致的维度具有如下特征
    1. 一致的维度关键字
    2. 一致的属性列名字
    3. 一致的属性定义
    4. 一致的属性值

有表 A 和表 B，这时候有两张维表甲（生成自表 A）和乙（生成自表 B），如果甲和乙中有一致性维度，那么我们可以通过一个维度访问原表甲和乙。

一致的维度可能意味着是相同的维度表

1. 与它们相连的事实表具有完全相同的内容（不同的度量值）。例如：连接到销售事实表与库存事实表上的日期维度表是同一的，意味着销售事实表和库存事实表中的内容是相同的
2. 这样的维度表在物理上可能是同一张表，也可能是不同的表，但它们应该具有相同数目的行、相同的关键字值、相同的属性标签、相同的属性定义与相同的属性值

大多数一致的维度是在可能的最佳粒度层次（最细粒度）上定义的。

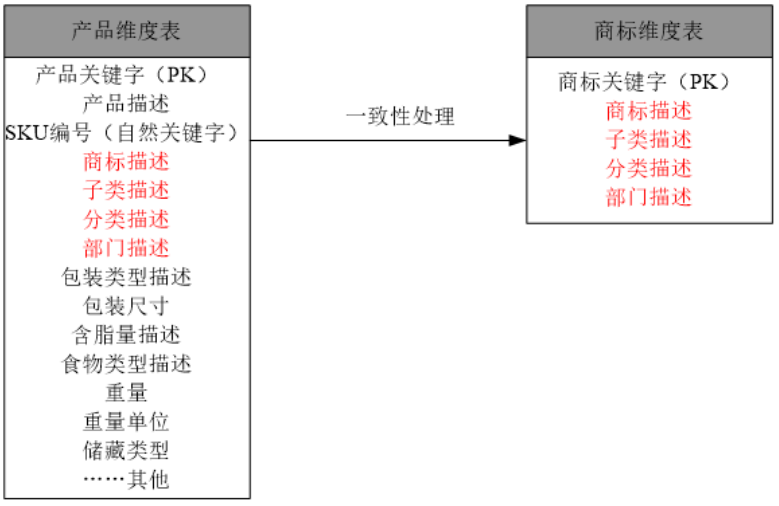
1. 顾客维：单个顾客
2. 产品维：用以对产品进行跟踪的最低层次
3. 日期维：天

3.8.1 不同粒度的维度

1. 原子型维度：在最佳粒度层次上的维度定义（最小的粒度）
2. 上钻维度（roll-up dimensions）
    1. 在较高层次上的维度定义（较大的粒度），用以连接较高层次的事实表：日期维表（连接每日快照）vs. 周维表（连接每周快照）
    2. 如果上钻维度是基本层次上原子型维度严格意义上的子集，则堆积维度与原子型维度保持一致

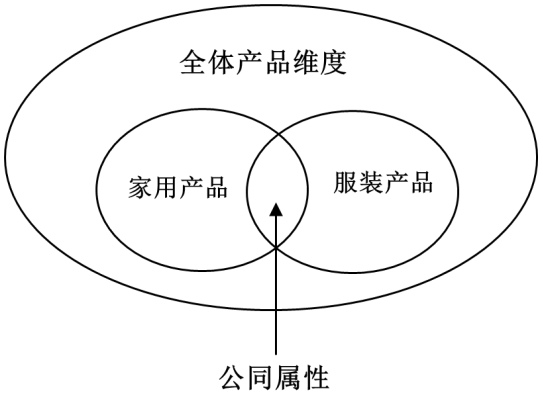
3.8.2 不同业务处理的事实粒度不同

产品维度 vs. 商标维度



1. 商标维度表，从数据结构、数据内容和元组而言都是产品维度表的一个子集，那么则商标维度表和产品维度表是保持一致的。
2. 我们将这个产品维表和商标维表合并后认为是一个公共维度，我们可以挑选出来部分属性，认为和原来的整体是保持一致的。
3. 上图的例子是指有映射保持一致。

两个处于相同细节层次上的维度表，如果它们均是另一个维表的子集，则它们也是一致的：



1. 没有映射关系，但是也保持一致的例子如上图所示。
2. 全体产品维度中有的维度对于家用产品是没有意义的，所以家用产品的维度是小于全体产品维度。而家用产品和服装产品之间存在有公同属性
3. 交集则会存在同时家用产品和服装产品的产品 A，产品 A 属于家用产品的属性和服装产品的属性具有不同和相似点，我们使用的是全体产品维度的值来连接两个事实表。
4. 不拿掉用不到的维度除了浪费空间以外也不影响，说明使用一个一致性维度（我们设计了一个大的表，但是使用的是部分的一致性维度表）

3.9 一致性事实

同样的事实在不同的数据备份中进行存储的一致性：

1. 取值单位的一致性
2. 值的一致性
3. 自然关键字的一致性

一般说来，事实表数据不在多个数据备份间明确的进行拷贝。

数据仓库中一般不允许使用编码格式，但是如果存在有共识，比如 1 代表男性，则可以使用编码格式。

如果事实表存在于多个数据备份，那么支撑这些事实的定义和方程必须都是相同的。

如果无法使事实完全保持一致，那么应该对不同的解释给予不同的名称。

4. 多维建模案例之三：订单管理

4.1 订单管理

1. 订单事务方案
2. 事实表规范化方面的考虑
3. 维度设计策略
    1. 日期维度的角色模仿
    2. 维度表的多属性体系结构
    3. 杂项维度
4. 事实表设计策略
    1. 多种货币与计量单位

2. 不同粒度层次上度量值的分配考虑
3. 赢利与亏损事实的票据处理事务方案
4. 订单处理流水线的累积快照方案

5. 三种不同类型事实表的比较
6. 数据仓库中的实时分区

4.2 订单管理的引入

4.2.1 订单管理

1. 所关注的业务处理流程：报价，生成订单，安排生产计划，组织货物的装运发送，票据处理
2. 所关注的分析对象：
    1. 数量：订购，生产，装运……
    2. 收入：订购额，贴现额，净订购额……

4.2.2 数据仓库的总线矩阵子集（订单管理部分）

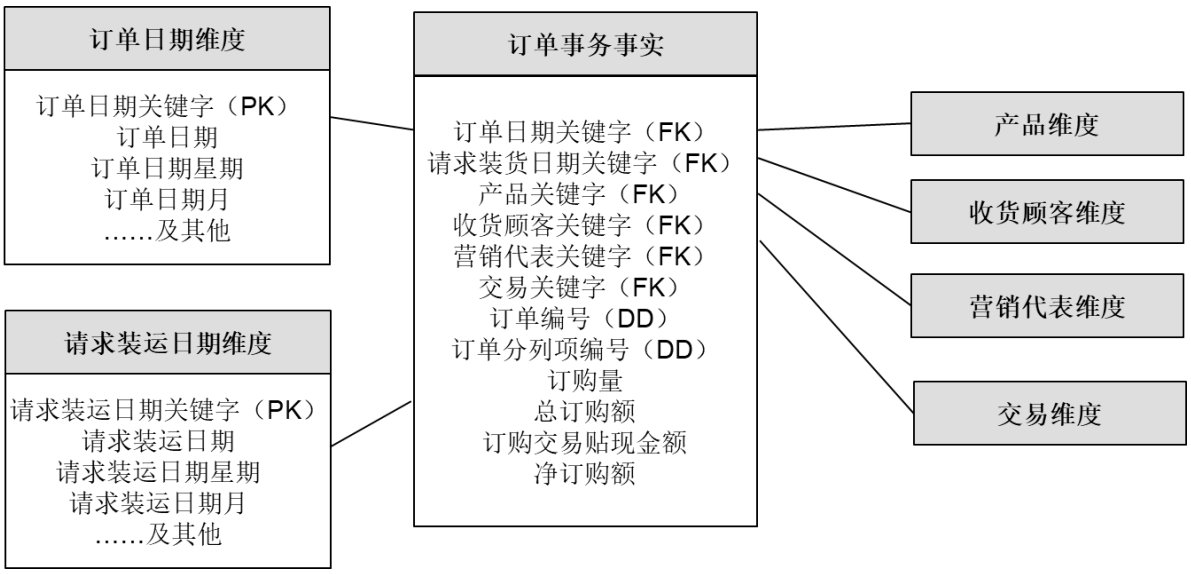
不打算多描述我们直接简写就可以过去，不用描述每一个项。

	日期	产品	顾客	让利	营销代表	发货	货主
报价	x	x	x	x	x		
订单	x	x	x	x	x		
装运	x	x	x	x	x	x	x
发票	x	x	x	x	x	x	x

4.2.3 最基本的订单事务事实表

1. 为订单的每个订单分列项建立一个记录行（元组）
2. 所包含的度量值有：
    1. 订购量
    2. 订单增值总额：导出性属性值
    3. 订单贴现金额
    4. 订单增值余额（订单增值总额-贴现）
    5. ……

4.3 订单事务事实表



1. 我们是将总订购额、订单交易贴现金额和净订购额要不要包含在事实表中
2. 在绝大多数情况下我们不进行事实表规范化，只有满足以下两个条件时才进行拆分。
    1. 事实行的事实设置比较稀疏
    2. 不在事实之间施加运算
3. 规范化
    1. 规范化拆分的时候往往后面的互斥属性会节约空间，但是前面的属性会冗余可能会导致浪费空间，需要进行考虑。
    2. 有上下文环境访问拆分后的表，需要用很多次查询才可以拿到原来应该得到的属性，也就是这种操作不一定节约空间，一定浪费时间
4. 约定了很多的日期，就应该有很多的维度用来表示日期

4.4 事实表的规范化考虑

事实表的规范化：将一张事实表中的多个度量值分解组装成多个事实表

4.4.1 事实表规范化的目的

在对事实表进行规范后，可以连同标识事实类型维度一起得到单一的一般性事实

4.4.2 规范化的时机

1. 事实行的事实设置比较稀疏
2. 不在事实之间施加运算

4.4.3 规范化的问题实例

1. 方案 A（反规范化）：假设有一个由 10 个外关键字属性，5 个度量值属性以及 100 万行元组所构成的事实表
2. 方案 B（规范化）：将上述的一张事实表分解为只记录单个度量值的 5 张事实表

	方案 A	**方案 B**
结构复杂性	1 张表，15 个属性	5 张表，55 个属性
数据量	1500 万个属性值	最多 5500 万个属性值

	方案 A	方案 B
数据访问	可以直接在 SQL 语句中进行数学计算，以获得新的度量值	首先需要执行表的联接操作，然后才能进行数学计算。而联接操作需要更多的时间开销

4.4.4 规范化情况

一般不考虑事实表的规范化。除非不同度量值的计算处于不同的粒度层次上，那么则需要将它们分解到不同的事实表中去

如果可以将“粗”粒度的度量值分配到“细”的粒度层次上，那么则可以在尽量细的粒度层次上通过统一粒度层次来建立一张统一的事实表：事实表中的粒度层次越“细”，则可以提供的分析操作就越多

反规范化节约空间又节约时间，所以我们乐意这样干，但是有时候反规范化会导致误导和粒度不兼容

1. 误导：部分属性或关键字的含义没有被用户恰当的理解到。

粒度无法统一的案例

订单的折扣：这个是按照订单作为单位

1. $购买总金额 \times 订单折扣 = 实际购买金额$
2. 按照摊派的思想，我们要将订单的折扣摊派给每一个元组，但是可能会导致分析误导：订单折扣和其他的部分关联非常大，订单折扣和谈判人员是有关的。

这是针对订单事实的度量值，我们不能将其细化到订单分列项事实上

1. 虽然不影响整个订单的购买金额的计算，但是：
2. 会影响到沿着商品维度（或其它订单分列项元组中的维度属性）的分析操作

4.5 维度设计策略

基于数据仓库总线的设计思想，订单管理维度模型可以与前述的其它维度模型共享一组公共的维度表，如日期维，产品维，顾客维.....

针对订单管理的特殊性，在维度表的设计过程中还需要考虑下列问题：

1. 维度的角色模仿
2. 维度表的多属性体系结构
3. 杂项维度

4.6 日期维度的角色模仿

在基于多事务的订单管理事实表中，存在着若干个日期类型的外关键字：

1. 每一个日期外关键字都对应着订单处理过程中的某一个业务处理步骤，如：订单的创建日期，产品加工日期，成品入库日期，请求装运日期，计划装运日期，实际装运日期，到货日期，发票日期，.....

实现方式：

1. 为每个日期类型的外关键字建立一个独立的日期维表
2. 所有日期类型的外关键字共享同一个物理的日期维表

单个维度同时在同一个事实表出现几次：

1. 建立多组合的维度：订单日期 × 装货日期（365×365）
2. 角色模仿

日期维度的角色模仿：

1. 后台只维持一个单一的日期维度表，类似数据库的 View
2. 为事实表中的每一个日期外关键字建立一个日期维表上的视图，类似的也可以用上角色模仿（比如员工的模仿）
3. 优点：降低存储空间开销，方便使用

例如：

```
CREATE VIEW ORDER_DATE(ORDER_DATE_KEY, ORDER_DAY_OF_WEEK,ORDER_MONTH,...)
 AS SELECT DATE_KEY, DAY_OF_WEEK,MONTH,......
      FROM DATE
CREATE VIEW REQ_SHIP_DATE(REQ_SHIP_DATE_KEY, REQ_SHIP_DAY_OF_WEEK, REQ_SHIP_MONTH,...)
   AS SELECT DATE_KEY, DAY_OF_WEEK,MONTH,......
      FROM DATE
```

1. 日期维度在单个事实表中承担不同角色
2. 映射一定是一对一的，可以用来连接若干个不同的度量值的。

4.7 维度表的属性体系结构

在一个维度表中：

1. 通常存在着若干组用于描述维度表中的元组在不同方面的描述属性
2. 在许多非体系属性之外，存在一个或而多个属性体系结构

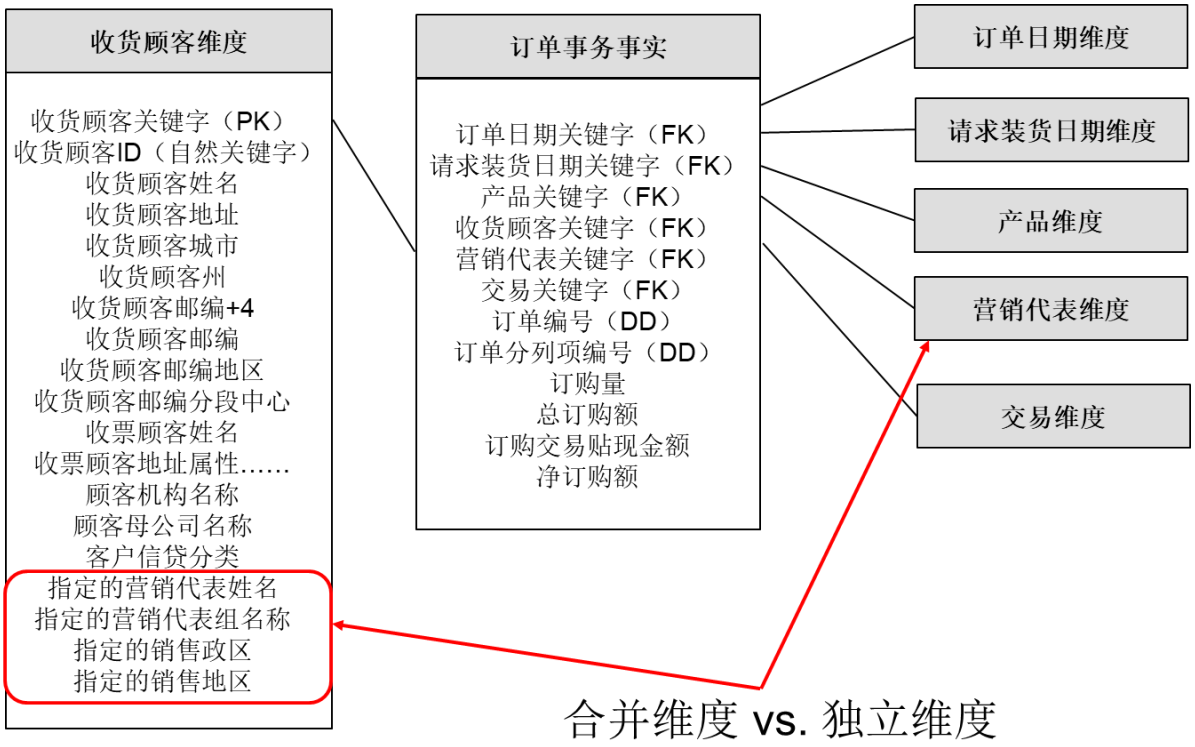
例如：商品维

1. 子类描述，分类描述，部门描述
2. 包装类型描述，包装尺寸
3. 含脂量描述，食物类型描述
4. 重量，重量单位，储藏类型描述
5. 货架期限描述，货架宽、高、深
6. ……

星形模型 → 雪花模型，浏览性能 → 存储空间。

在雪花/雪暴模型中，能够通过子维度作为公共维度连接多个多维模型的，应充分考虑维度的规范化。

4.8 收货顾客维度



上图中的情况类似淘宝的收货地址。

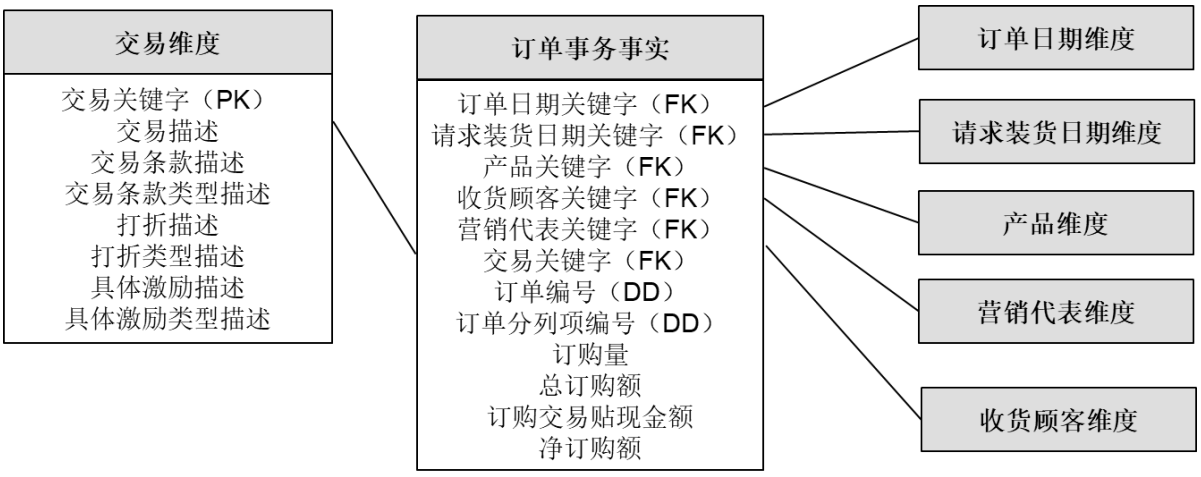
1. 粒度：为每个分离的收货地址包含一行内容
2. 收货顾客维表中的属性体系结构：

1. 收货地址
2. 收票地址
3. 顾客机构体系：地址 - 顾客单位与签收单
4. 营销机构

3. 营销机构（作为一个单独的维度还是放在顾客维度表）：
    1. 营销代表与收货地址之间的关系（同体系？）
        1. “一对一”或“多对一”高度相关：合并为一个收货顾客维度
        2. 随着时间或产品产生变化：两个独立的维度
    2. 如果营销代表与收货顾客独立地参与了其它的事实表，建立各自独立的维度表
4. 当实体之间存在固定的、不随时间变化的、强烈相关的关系时，需要作为单一维度进行建模，而其他情况下需要分割
5. 需要考虑维度过多的情况，如果方案已经确定维度数量（例如，25）则充分考虑维度组合的问题

4.9 交易维度

如果期限、打折等交易信息存在相关，则组合成一个维度



4.10 订单编号退化维度

1. 来源于操作型数据环境的订单细节已经从订单标题中剥离出来形成独立的维度：
    1. 订单日期
    2. 收货顾客地址
2. 订单编号用于对订单上的**分列项目**进行分组，因此仍然有效
3. 偶尔用于数据仓库反向连接操作型领域

4.11 杂项维度

从复杂的数据源中提取与事实、维度相关的字段后，往往还有大量在小范围内选取离散值的指示符与标志，这些维度和其他维度都没有关联，他们之间也只有少量的关联，由于没有足够的空间，则将他们压缩到一个维度中。

1. 将标志与指示符不加改变地留在事实表行中→事实表膨胀
2. 将每个标志与指示符放在本身的单独维度中→维度膨胀
3. 将所有标志与指示符从设计中剥离出来→删除难以理解的、杂乱的或者与分析操作无关的维度属性

可以将它们组装成一个或多个独立的杂项维度表（junk dimension）

1. 杂项维度使用非编码属性

2. 我们可以通过预先方式来确定杂项维度表：杂项维度中如果可以预先找到所有的可能属性组合，比如下图中的支付类型为信用卡和支付类型组是现金是不存在的。我们可以根据数据仓库分析是否有的元组没有被用到，如果用的比较常用则才使用杂项维度，不然可以用插入的方式。
3. 杂项维度可以用来处理自由注释字段，比如调查问卷中的选填问答。

订单指示符关键字	支付类型描述	支付类型组	订单出/入指示符	代办信用指示符	订单类型指示符
1	现金	现金	输入订单	可代办	一般
2	现金	现金	输入订单	非代办	展览
3	现金	现金	输入订单	非代办	示范
4	现金	现金	输出订单	可代办	一般
5	发现者信用卡	信用卡	输出订单	非代办	展览
6	发现者信用卡	信用卡	输入订单	可代办	一般
7	发现者信用卡	信用卡	输入订单	非代办	展览
8	发现者信用卡	信用卡	输入订单	非代办	示范
9	发现者信用卡	信用卡	输出订单	可代办	一般
10	发现者信用卡	信用卡	输出订单	非代办	展览
11	万事达信用卡	信用卡	输入订单	可代办	一般
12	万事达信用卡	信用卡	输入订单	非代办	展览
13	万事达信用卡	信用卡	输入订单	非代办	示范
14	万事达信用卡	信用卡	输出订单	可代办	一般

杂项维度例子：

1. 录入成绩的人的这种信息本身没有分析的价值，A 还是 B 录入都没有影响，这种属性就是杂项维度。
2. 用户是随心所欲的购买或有计划购买，可能这种信息本身只有很少的价值，但是本身也有一定的价值。

预先为所有组合创建杂项维度行 vs. 实际遇到的组合创建杂项维度行：组合可能大小 vs. 组合预计大小

杂项维度可以用以维护附在事实行上的自由注释字段

1. 参数化自由注释字段
2. 自由注释的数量远小于事实行的数量，需要引入“非注释行”的代理关键字
3. 自由注释字段可以有很多组

4.12 多种流通货币

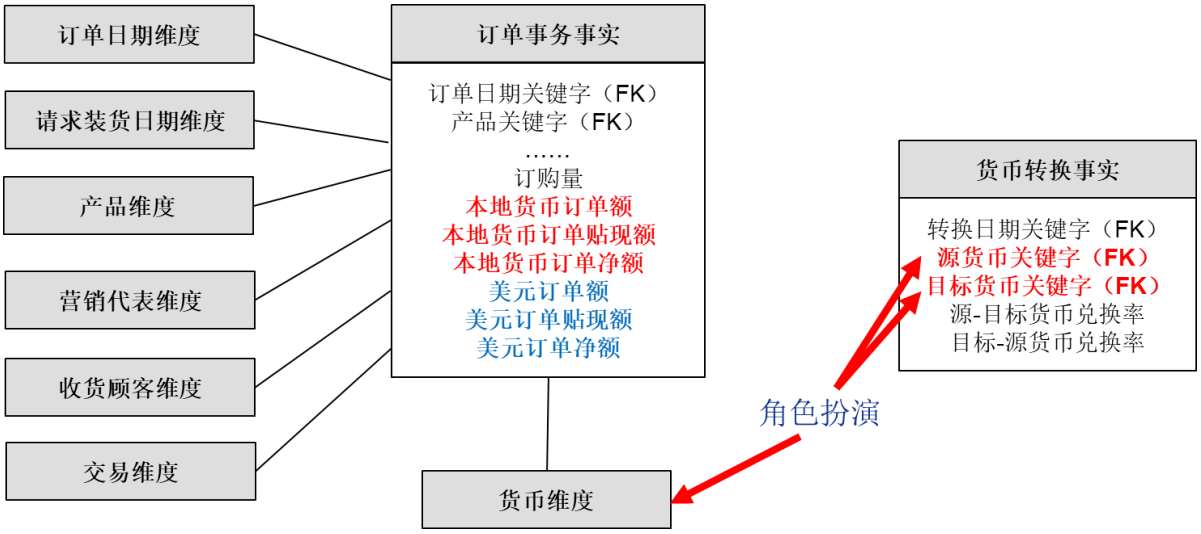
选择一个标准的通用货币，并建立其它货币与之转换关系：

1. 不同货币之间的汇率是随着时间变化的
2. 同时货币之间的互兑汇率也是不尽相同的

跨国企业的货币的结算方式是不同的，汇率也是在变化的，所以需要用存储汇率

建立货币转换事实表

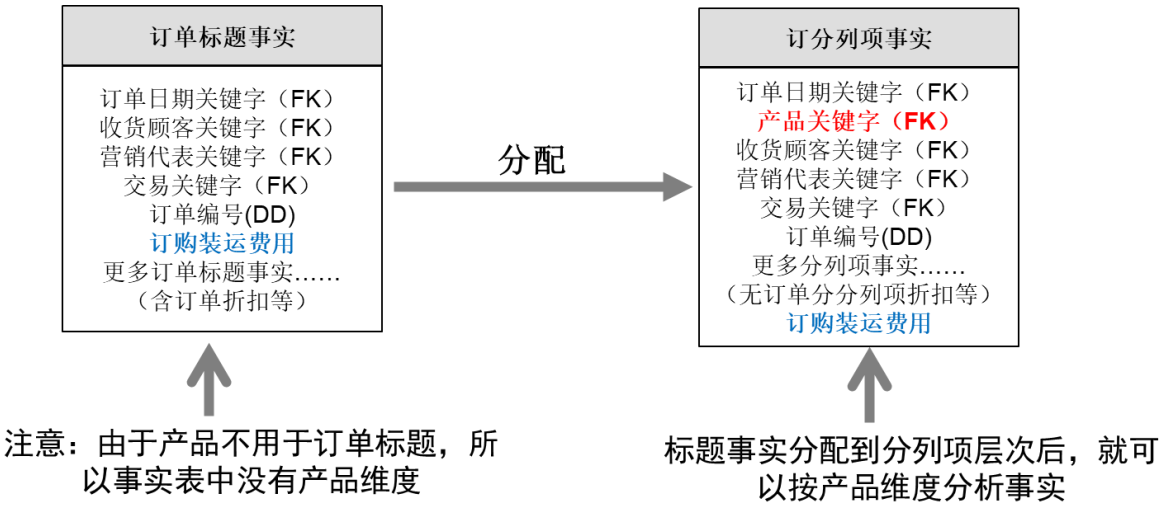
建立货币维表：货币和国家之间不一一对应



1. 红色货币计算由货币维度来存储计量
2. 源-目标货币兑换率和目标-源货币兑换率并不互为倒数

4.13 粒度不同的标题与分列项事实

1. 订单的运费：仅适用于整份订单
2. 描述：从订单到订购单中每一个商品
3. 处理方法：
    1. 在较低层次事实表中尽可能包含所有可用的高层事实表中的事实
        1. 但这样的实现方式并不能适用于所有的情况
        2. 不能在同一个事实表中混用不同粒度的事实，解决办法：向下分配事实
    2. 将运费与其他标题级事实展现在用于整个订购的聚集表中
    3. 高层事务进行分列来适应具体的分析需求
4. 订单标题事实到订单分列项的分配：订单标题事实由于包含订单折扣，所以是不可以替代的，设计到反规范化的问题。



4.14 装运发票事务

装运发票的内容：

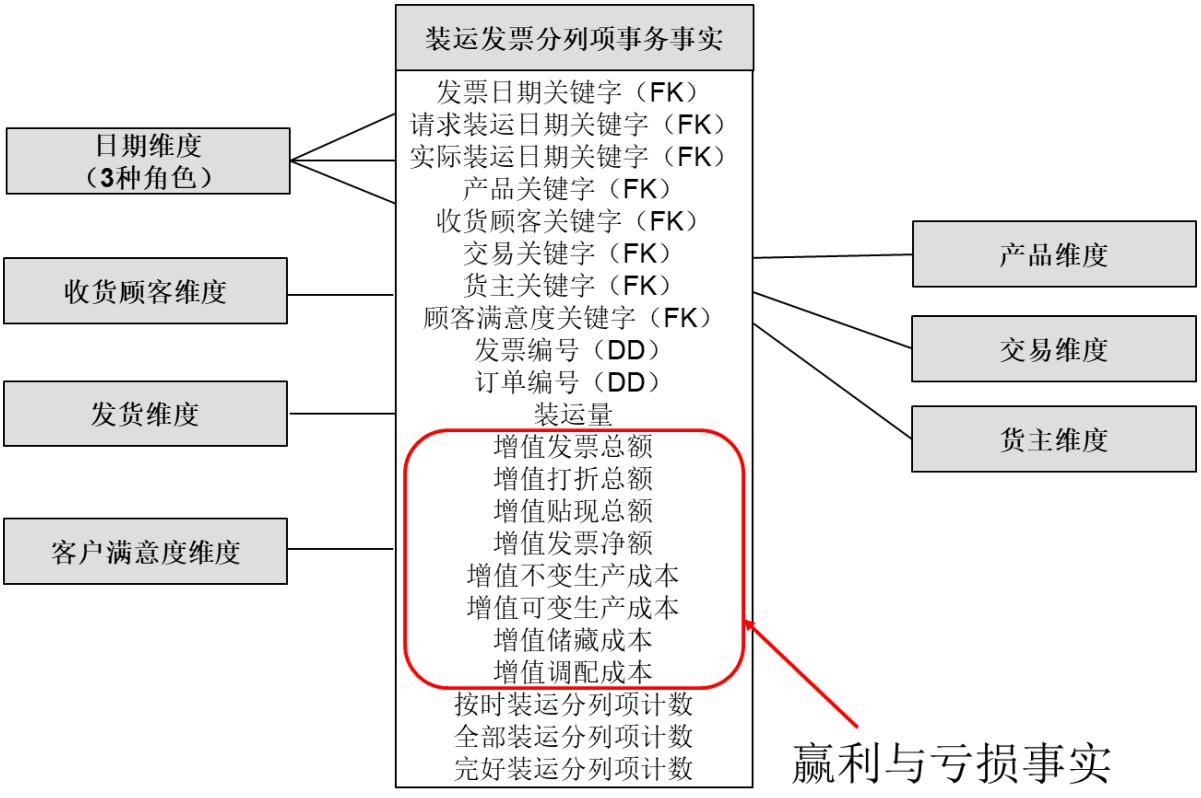
1. 发货日期，目的地，顾客
2. 具有多个分列项（对应着发送的不同商品）：不同的分列项有不同的数量、价格、贴现与打折等内容
3. 发票总额

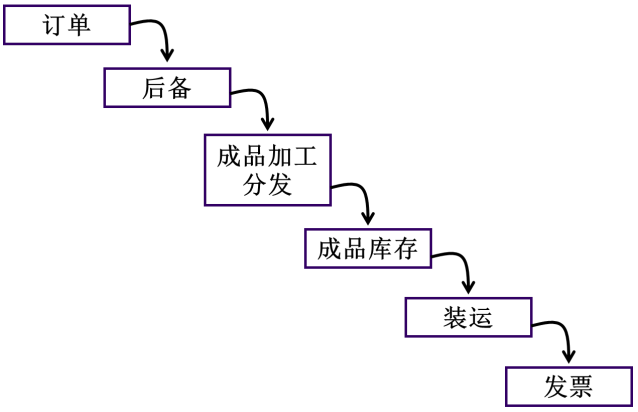
装运发票事实表的设计：

1. 建立对应各个分列项的事实
2. “新”的维度：发货，货运人，顾客满意度

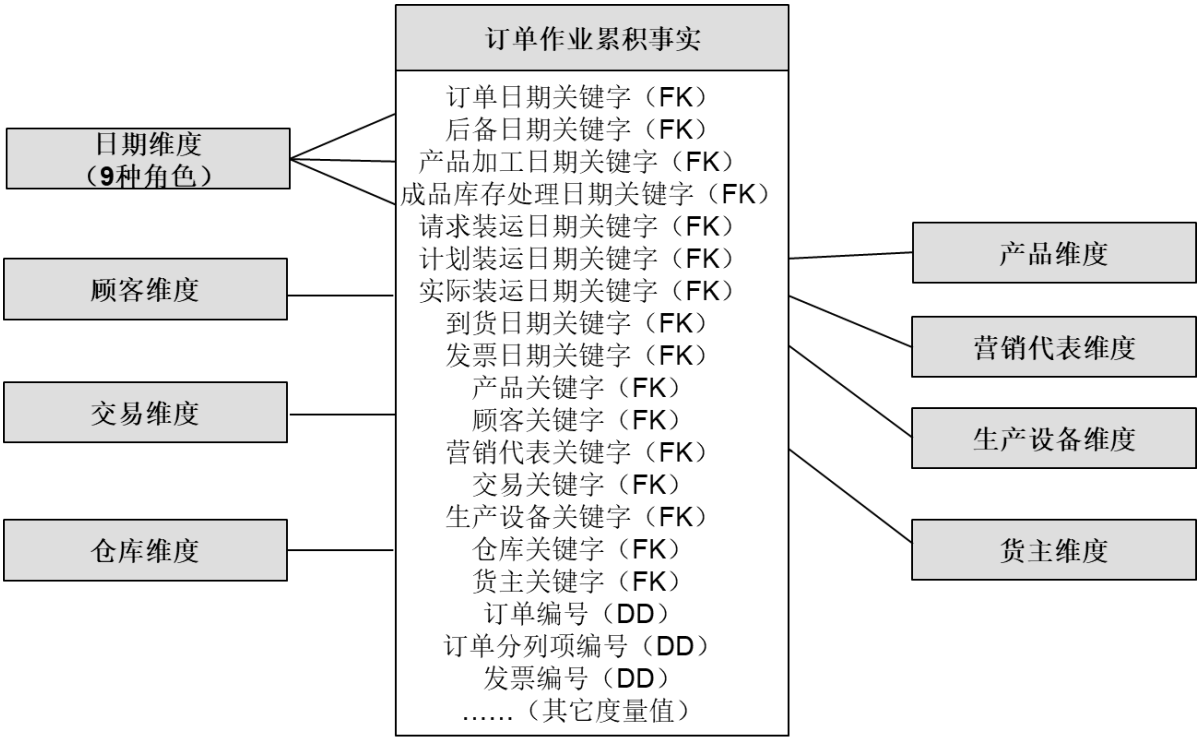
发货维度：为每个制造商货栈或装货点建立一个维度元组，包括：名字，地址，联系人，存储设施类型等维度属性

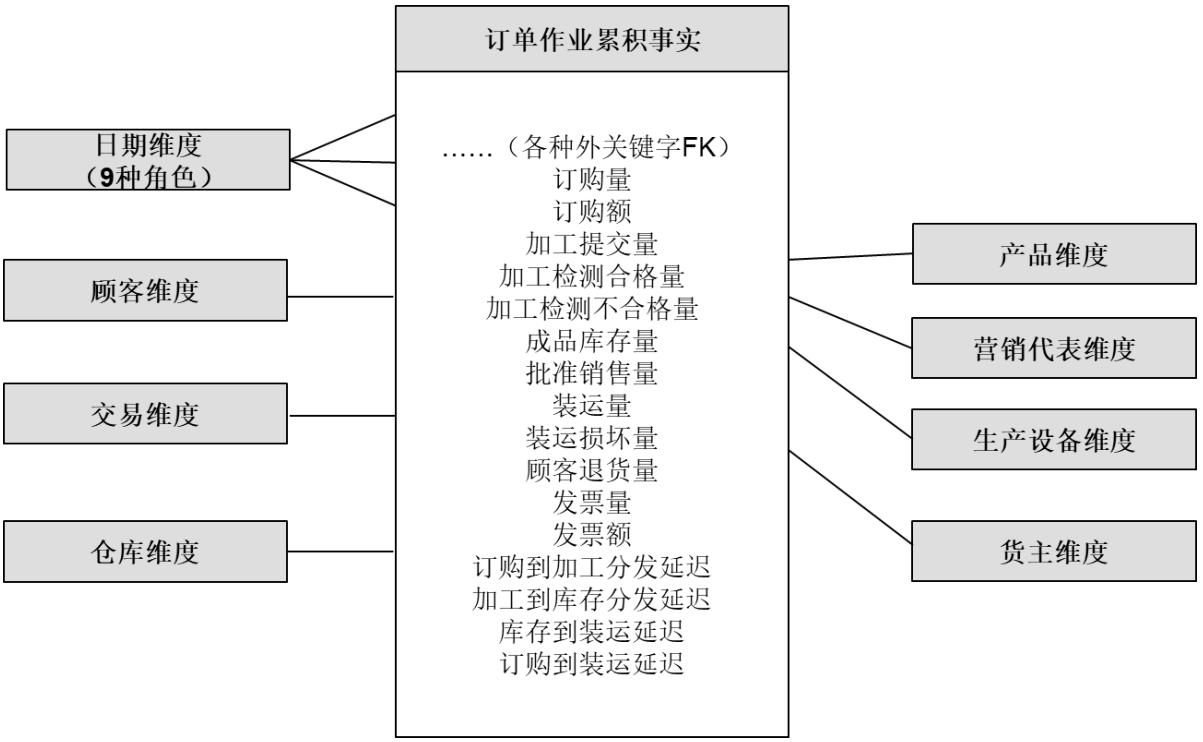
货运人维度：描述将产品从制造商运送给顾客所使用的方法与运载工具



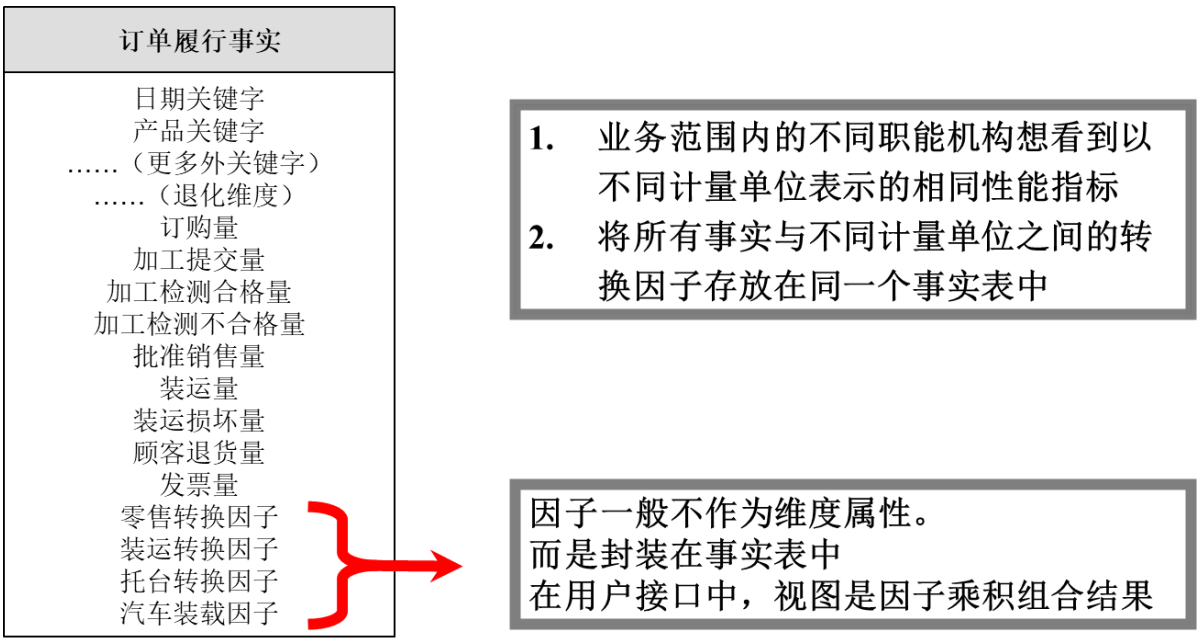


4.15 订单任务累积快照





4.16 支持多计量单位的事实表



1. 其中的转换关系不一定存在
2. 转换单位和比例因子并不由上下文来决定

4.17 三种类型事实表的比较

特征	事务粒度	周期快照粒度	累积快照粒度
代表的时间段	时间点	规律性可预见间隔	不确定时间跨度，一般是短期
粒度	每个事务事件一行	每段一行	每个生命期一行
事实表加载	插入	插入	插入与更新
事实行更新	不重新存取	不重新存取	行为发生任何时候都要重新存取
日期维度	事务发生日期	时间段终止日期	标准关键环节的多个日期
事实	事务活动	预定时间间隔的性能	给定生命期的性能

4.18 实时分区

在数据仓库环境中，对实时业务数据的访问需要：

1. 在常规静态数据仓库的前面建立一个物理的实时分区
2. 对实时分区的约束要求：
    1. 包括静态数据仓库最后一次更新以来出现的所有行为
    2. 尽可能无缝地连接到静态数据仓库事实表的粒度与内容上
    3. 能够轻松地建立索引，以致于总是可以不断吸纳新来的数据

三种不同类型的实时分区：

1. 事务粒度：当天的记录（并非统计结果）
2. 周期快照：最近一个周期内的统计结果，对非/半加性事实的考虑
3. 累积快照：只记录最近被更新的项

大多情况下，在原来的数据仓库中有什么数据模型，那么在实时分区中也会有同样的事实粒度和周期粒度

4.18.1 事实粒度

实时分区具有与它的支撑静态事实表具有完全相同的维度。

可能完全不建立索引：

1. 为加载操作维护一个持续打开的窗口
2. 没有时间系列可用

避免包含聚集值，提供很快的加载性能的同时，提供快速的查询性能

4.18.2 周期粒度

静态数据仓库事实表具有一个周期粒度，实时分区可以看作是当前的累积月

实时分区是当前正在开发的月份的映像，随着月份的推进不断更新。半加性或全加性事实随报表不断调整

月份结束时累计起来的实时分区，作为最新月份加载到静态仓库。

4.18.3 累积快照

静态数据仓库事实表采用累积快照时，实时分区仅仅包含当天更新的分列项。

当天结束时，实时分区包含了需要写到主要事实表上的记录的最新版本。

无需索引和聚集。

5. 多维建模案例之四，客户关系管理

客户维度又想要当事实表，又想当维度表，也就是可以分析自己，也可以分析其他人。

5.1 客户关系管理（CRM）

1. CRM 既包含操作处理，又包含分析处理
2. 目标：
    1. 将分析中心转向客户对象
    2. 支持以客户为中心的分析应用
3. 建设步骤：
    1. 客户数据集成
    2. 客户信息的分析与挖掘

5.2 客户数据的集成

1. 客户信息中的姓名和地址的集成：
    1. 对姓名和地址进行解析，分成更细小的片断
    2. 对姓名和地址进行标准化
2. 客户数据的更新：
    1. 新客户的识别
    2. 老客户的信息更新（可能需要使用数据仓库中已有的历史数据）

5.3 客户信息的分析与挖掘

1. 客户特征分类分析：新客户的发现
2. 客户盈利分析：优质客户挖掘
3. 客户行为分析：客户异常行为的发现
4. 客户需求分析

5. 客户反映分析：
    1. 提供一对一服务（one to one）
    2. 防止已有客户的流失

5.4 客户维度

1. CRM 的基础是一致性的公共客户维度
2. 客户维度的特点：
    1. 特别多的元组：通常具有数百万行以上的元组。
    2. 特别多的属性：客户所具有的属性是非常多的。
    3. 快速变化的维度：部分属性是在随时变更的。

5.5 姓名与地址的解析

1. 将姓名与地址属性尽可能地拆分成一些基本的部分。
2. 解析时的注意事项：
    1. 统一的表示形式
    2. 地址方面的差异
    3. 文化方面的差异

5.6 客户维度示例

5.6.1 过于一般的客户维度实例

维度属性	实例值
姓名	R. Jane Smith 律师小姐
地址-1	123 Main Rd. North West. Ste 100A
地址-2	2346 信箱
城市	康新敦
州	阿肯色
ZIP 编码	88887-2348
电话	888-555-3333 转 776 555-4444（传真）

具有解析姓名与地址的客户维度实例

维度属性	实例值
称呼	小姐
非正式问候称谓	Jane
正式问候称谓	Smith小姐
开头与中间称谓	R. Jane
姓	Smith
后缀称谓	初等（律师）
种族	英国人
头衔	律师
街道号	123
街道名称	干线
街道类型	道路
街道走向	西北

维度属性	实例值
信箱	2348
门牌号	100A
城市	康新敦
行政区	康沃尔
二级行政区	伯克利郡
州	阿肯色州
地区	南部
国家	美国
洲	北美洲
主邮政ZIP编码	8887
副邮政ZIP编码	2348
邮政编码类型	美国

注意发现具有区分度的属性：街道、头衔等

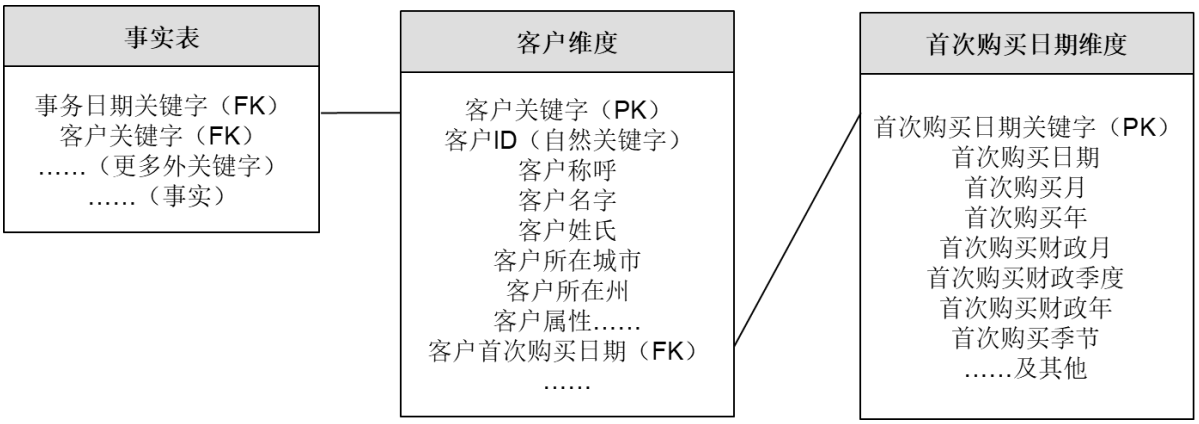
维度属性	**实例值**
办公电话国家编码	1
办公电话地区编码	888
办公电话号码	5553333
办公分机号	776
传真电话国家编码	1
传真电话地区编码	888
传真电话号码	5554444
电子信箱地址	RJSmith@ABCGenIntl.com
Web 站点	www.ABCGenIntl.com
唯一客户标号	7346531

唯一客户编号是用来反向连接操作型数据环境的。

5.6.2 常见的其他客户属性-日期

1. 客户维度中，包含类型为日期的维度属性
2. 为客户维度建立若干个日期维度表，如：
    1. 出生日期
    2. 首次购买日期
    3. 最后一次购买日期

5.6.3 日期维度支架



1. 首次购买日期维度连接过去是 FK
2. 后面可能本身对应后台一个物理表

5.6.4 常见的其他客户属性-客户分类属性

- 性别
- 种族
- 年龄（分级）
- 收入（等级）
- 状况（*如：新潮，活跃，不活跃，封闭*）
- 崭新程度（*最后一次购买日期*）

- 频率（*购买事务总数目*）与强度（*净购买总量*），以及据此挖掘获得的客户分类属性
- 客户的统计特征（*如：购买行为，支付行为，产品喜好、挑剔度以及违约概率等*）

根据客户的分类属性进行分类，来进行商品推荐等行为

5.6.5 常见的其他客户属性-聚集事实属性

用户经常需要执行基于聚集性能度量指标的客户信息查询，我们可以将用户最关心的聚集性能度量指标设计为客户维度表中的属性

1. 消费总额
2. 平均消费额
3. 单笔消费的最高额

就是用来担任度量值的属性，有顾客消费总额、平均消费额和单笔消费的最高额，这个作为核心可以构建多维模型的

5.7 低基数属性集的维数支架

低基数属性：

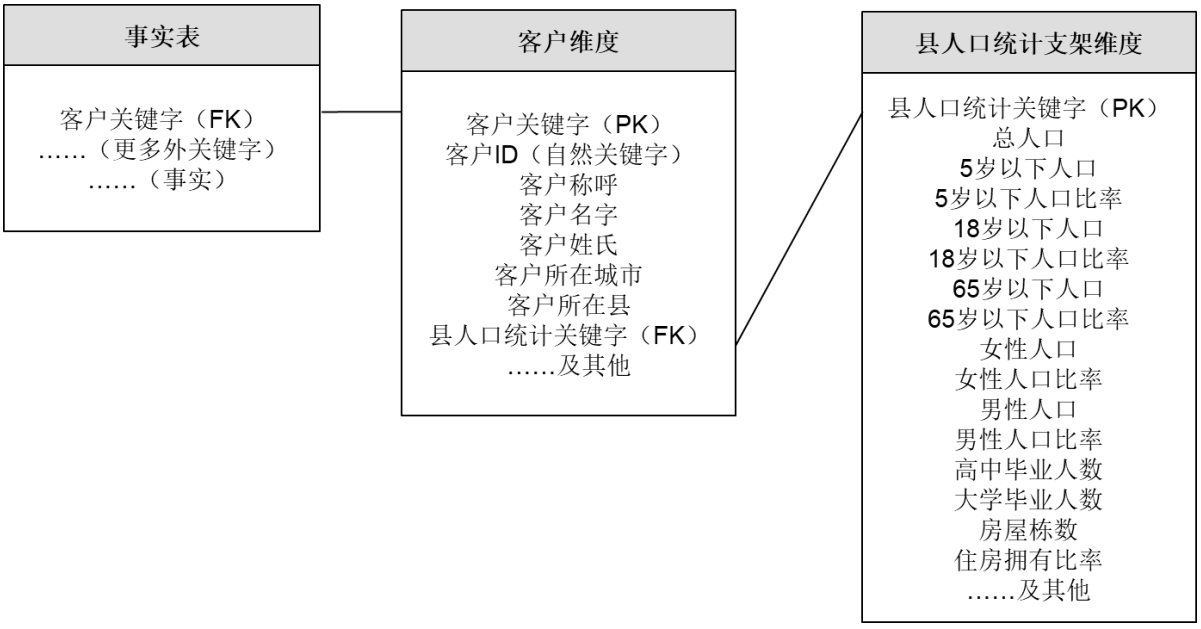
1. 只有少量几个取值的属性
2. 大量元组共享了相同的属性和属性组，这很可能就是我们看到的上下文信息

支架维度：

1. 将一组低基数属性单独构成客户维度的一个维度（称为支架维度），从而使整个模型呈雪花状
2. 支架维度中的数据一般是从外部数据提供者那里获得的，如：县人口统计支架维度
3. 如果用户的查询工具坚持使用星型结构，那么可以通过视图定义来隐藏维度支架

客户维度和支架维度（使用支架维度的好处）：

1. 客户维度往往和支架维度有相差悬殊的粒度
2. 具有不同的管理与加载次数
3. 可以节省客户维度表的存储空间
4. 如果用户的查询工具坚持使用星型模型，那么可以通过视图定义来隐藏维度支架



雪花模型中的一部分就是用来完成这部分，单独放大看就是星型模型。

客户维度和县人口统计支架维度的变化速度也是不一样的。

数据仓库刷新频率可以不一样，维表中如果有一个属性发生了变化，为了更新数据历史完整化，那么我们必须要复制一个元组，如果没有维度支架，那么就要复制整体，而如果有维度支架则只需要复制部分。

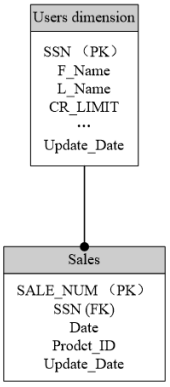
5.8 大型变化客户维度（重要）

在数据仓库的维度模型中，部分维度属性是会随时间而发生变化的。若只是将这些变化的维度属性值作简单的修正，即在维度表中只保留该维度属性的当前值，这会直接影响到对事实表中该维度属性所对应的事实数据元组的访问，特别是无法根据维度属性值的变化情况来进行分析处理。

维度表的划分（根据稳定性划分）

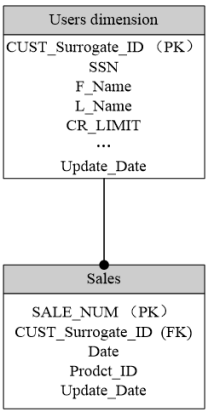
1. 稳定维度（done）
2. 渐变维度
3. 快变维度

5.8.1 Recall：历史完整性



Dimension.SSN	CR_LIMIT	FACT.SSN	SALE_NUM	DATE	TOTAL
123-12-1234	$2500	123-12-1234	1001	01/01/02	$1200
123-12-1234	$2500	123-12-1234	2310	02/25/02	$400

Dimension.SSN	CR_LIMIT	FACT.SSN	SALE_NUM	DATE	TOTAL
123-12-1234	$5000	123-12-1234	1001	01/01/02	$1200
123-12-1234	$5000	123-12-1234	2310	02/25/02	$400
123-12-1234	$5000	123-12-1234	4594	03/13/02	$2100



DIMENSION CUST_SURROGATE ID	SSN	CR_LIMIT
91101	123-12-1234	$2500
91101	123-12-1234	$2500
111211	123-12-1234	$5000

DIMENSION CUST_SURROGATE ID	SALE_NUM	DATE	TOTAL
91101	1001	01/01/02	$1200
91101	2301	02/25/02	$400
111211	4594	03/13/02	$1200

5.8.2 渐变维度

渐变维度的处理办法：

1. 类型 1：改写属性值
2. 类型 2：添加维度行
3. 类型 3：添加维度列
4. 类型 6：1+2+3

为了保证维度完整性

5.8.2.1 类型 1：改写属性值

容易实现，但不能对旧属性值的任何历史数据进行维护（无法保证历史一致性）。

不做任何处理，只需要保有最新的数据即可。

产品关键字	产品描述	部门	SKU（自然关键字）
12345	IntelliKidz1.0	教育	ABC922-Z

↓

产品关键字	产品描述	部门	SKU（自然关键字）
12345	IntelliKidz1.0	策略	ABC922-Z

5.8.2.2 类型 2：添加维度行

用两个元组来代表不同的情况：

1. 添加维度行是准确跟踪渐变属性的主要方法，也是跟踪历史变化的最常用的方式。
2. 通过引入新的行用来反映新的属性值
    1. 往往会导致维度行的膨胀
3. 可以引入生效或截止日期

产品关键字	产品描述	部门	SKU（自然关键字）
12345	IntelliKidz1.0	教育	ABC922-Z
25984	IntelliKidz1.0	策略	ABC922-Z

5.8.2.3 类型 3：添加维度列

使用维度列保存旧的属性值，但不适合跟踪维度属性大量变化。

设计维表的时候就设计预留位置来保存变更信息。有的时候是没有问题的，因为在业务中可能是有限制的，有上下文环境的，比如最多变化几次。

但是不可能所有属性都能确定有几次。

产品关键字	产品描述	部门	前部门	SKU（自然关键字）
12345	IntelliKidz1.0	策略	教育	ABC922-Z

5.8.2.4 类型 6（2+3+1）

产品关键字	产品描述	部门	前部门	SKU（自然关键字）
12345	IntelliKidz1.0	教育	教育	ABC922-Z

产品关键字	产品描述	部门	前部门	SKU（自然关键字）
12345	IntelliKidz1.0	策略	教育	ABC922-Z
25984	IntelliKidz1.0	策略	策略	ABC922-Z

产品关键字	产品描述	部门	前部门	SKU（自然关键字）
12345	IntelliKidz1.0	金点子	教育	ABC922-Z
25984	IntelliKidz1.0	金点子	策略	ABC922-Z
31726	IntelliKidz1.0	金点子	金点子	ABC922-Z

效率是最高的：

1. 列不能做连接操作，但是查看起来快
2. 跟踪近期变化，我们聚焦行就可以
3. 跟踪历史变化，我们聚焦列就可以

5.8.3 快变维度

5.8.3.1 快变维度的处理办法

微型维度

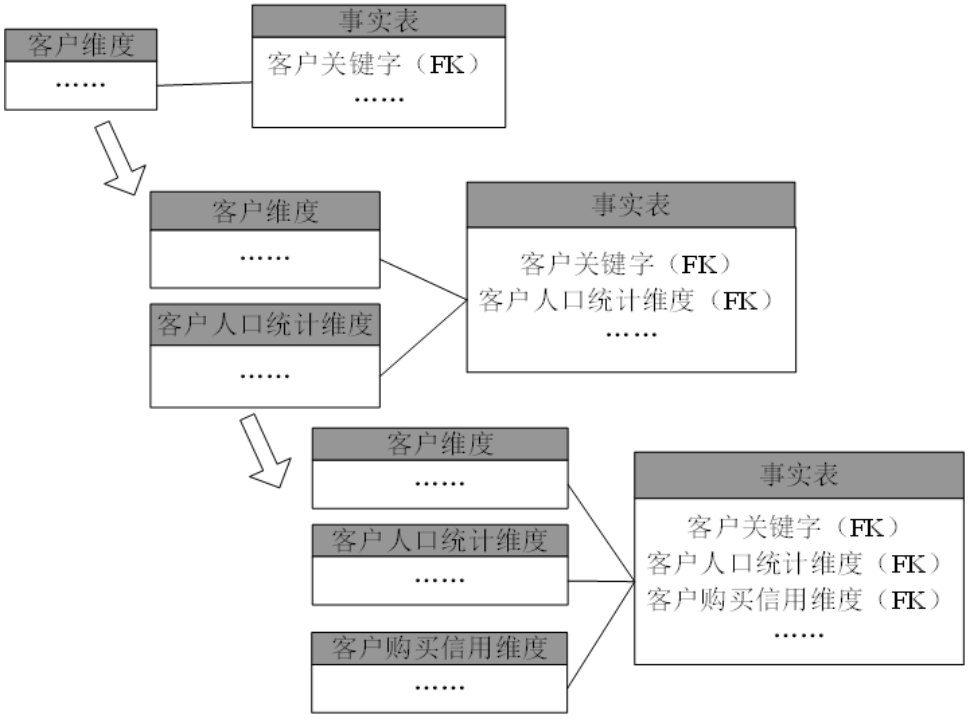
1. 将分析频率高或变化频率大的属性拆成为独立的微型维度（维度支架?），裂解后我们使用 FK 来连接上下文，如果选择类型 2 或类型 6 来处理，都需要来复制某些元组，而微型维度使得我们需要复制存储的元组大小变小。
2. 例如：客户维度中的年龄，性别，收入水平等属性，它们的每一种取值组合构成微型维度表中的一行

预设波段

1. 对于诸如收入与购买总额等不断变化的属性，应该被转换成呈波段分布的范围，即进行**离散化**处理，使其只能在数目相当小的离散值中取值，以减少维度表中的数据量
2. 将里面的值变化为区间，使用区间标签来作为实际的值

5.8.3.2 人口微型维度的样本行-微型维度

可以**将微型维度表独立于客户维度之外**，直接附加在事实表上，从而在基于微型维度属性进行分析操作时，可以**避开数量庞大的客户维度**，提高分析效率。



5.8.3.3 人口微型维度的样本行-预设波段

我们可以用预设波段来存储，比如划分为低中高三个维度。

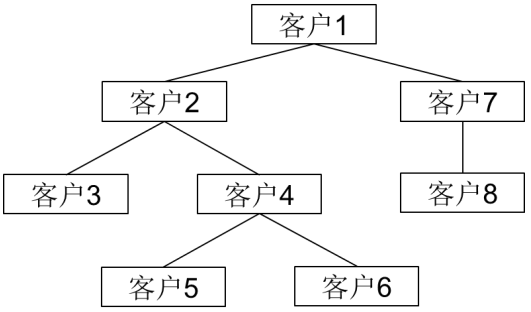
人口关键字	年龄	性别	收入水平
1	20~24	男	小于$20000
2	20~24	男	$20000~$24999
3	20~24	男	$25000~$29999
18	25~29	男	$20000~$24999
19	25~29	男	$25000~$29999

5.9 商务客户体系结构

深度不变体系：层数固定、可以预见的客户维度

深度可变体系：层数不固定的客户维度

如果我们客户不是自然人，而是法人，那么可能会有层次不固定的树状结构，不想要修改客户维度（保持一致性维度、保持公共总线）。

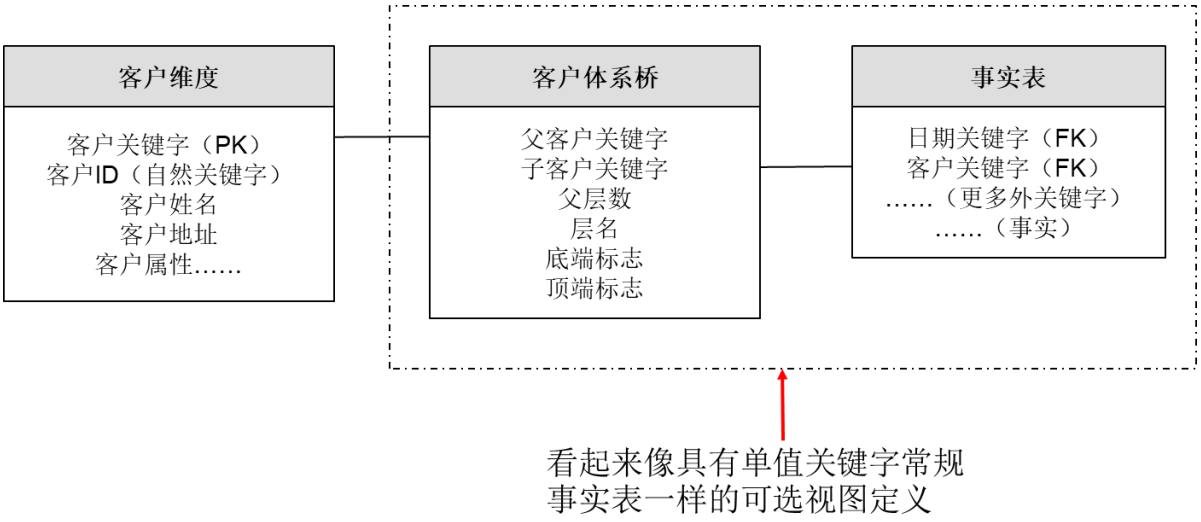


比如图上客户 1-8 所有客户所关联的某个度量值，或者是客户 4-6 的所有客户所关联的某个度量值

5.9.1 深度可变体系客户维度

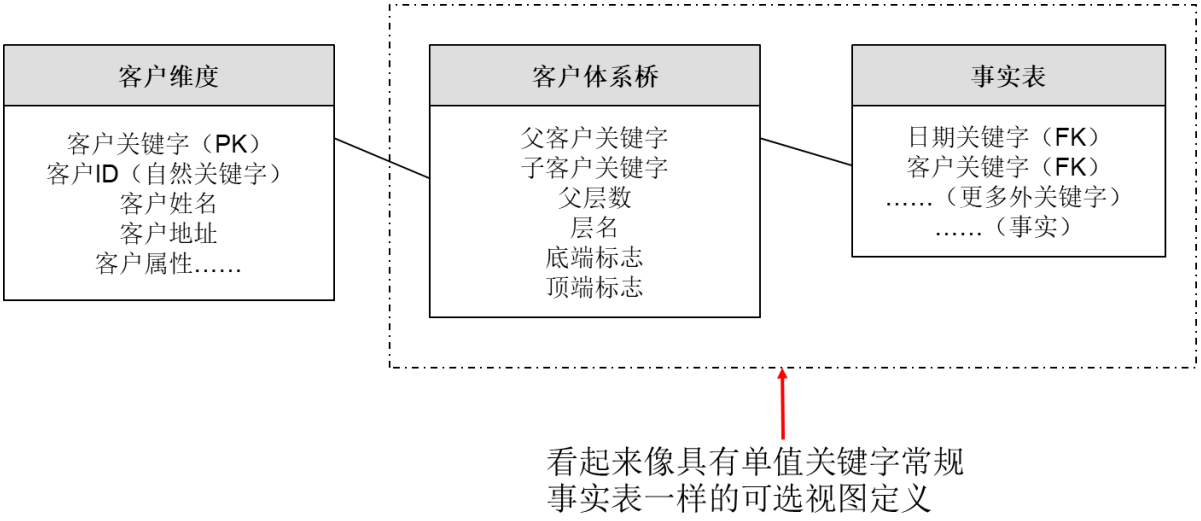
5.9.1.1 桥接表 - 自顶向下

1. 客户维度对应客户体系桥中的父客户关键字
2. 虚线框根据客户维度而言就是一张事实表
3. 在桥中间我们给定了父子客户关系



5.9.1.2 桥接表 - 自下而上

客户维度对应客户体系桥中的子客户关键字。



5.9.2 体系桥接连表样本行

体系桥接连表样本行数计算公式：

1. 用每层的取值数目乘以层深度（从顶层开始计数），然后将乘积相加起来
2. 体系桥接表样本行数 $= \sum(\text{层取值数目} \times \text{层深度})$

例：$1 \times 1 + 2 \times 2 + 3 \times 3 + 2 \times 4 = 22$

下图是树的二维化表结构，我们可以写出一个查询找到所要求的的目标结果。

父客户关键字	子客户关键字	父实体以下层数	底层标志	顶层标志
1	1	0	N	Y
1	2	1	N	N
1	3	2	Y	N
1	4	2	N	N
1	5	3	Y	N
1	6	3	Y	N
1	7	1	N	N
1	8	2	Y	N
2	2	0	N	N
2	3	1	Y	N
2	4	1	N	N

父客户关键字	子客户关键字	父实体以下层数	底层标志	顶层标志
2	5	2	Y	N
2	6	2	Y	N
3	3	0	Y	N
4	4	0	N	N
4	5	1	Y	N
4	6	1	Y	N
5	5	0	Y	N
6	6	0	Y	N
7	7	0	N	N
7	8	1	Y	N
8	8	0	Y	N