数据仓库的设计（二）

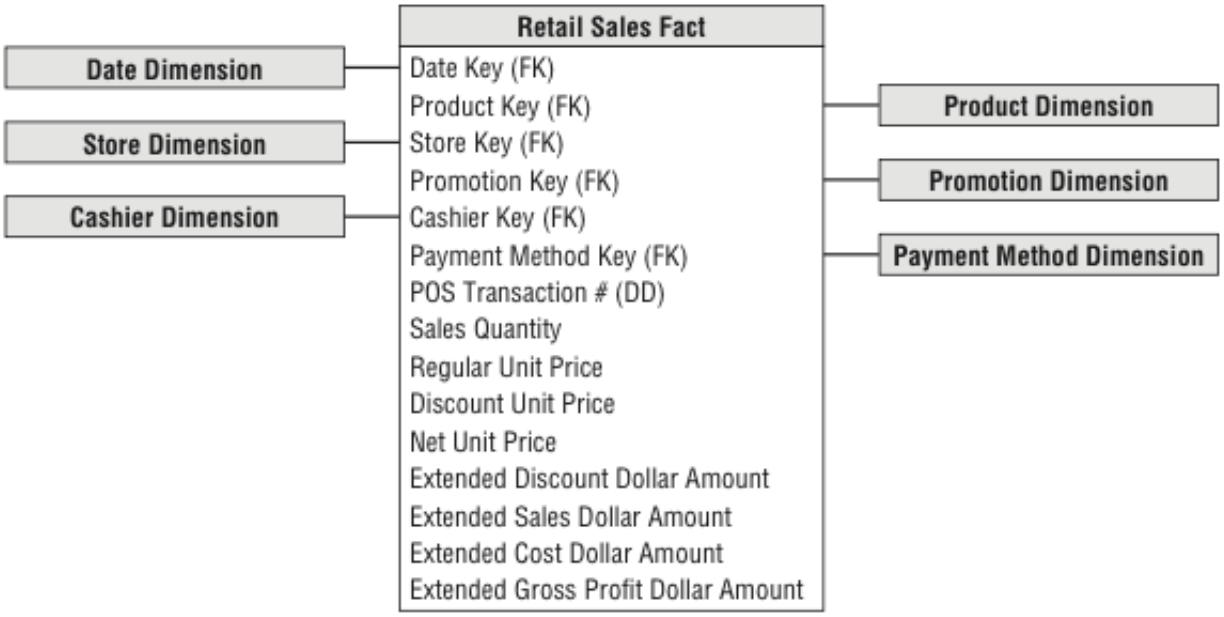
书中以不同行业为背景，举了不同的例子来说明数据仓库设计中的各种问题。而且强烈建议读者把所有的内容都阅读一遍，无论是不是你感兴趣的行业。因为行业只是背景，在不同行业背景里的数据仓库，用到了不同的设计方法，只有全部阅读一遍，才能全面了解数据仓库的设计。

这一章，是零售行业的背景。

维度设计的过程

开始数据仓库的维度设计，需要进行以下四个步骤：

1. 选择业务过程：样例中管理层想要了解POS系统中顾客的购买情况，因此业务过程就是POS零售交易，它的数据可以使得业务人员分析在哪天哪个卖场的哪个商品在哪种促销情况下售出。
2. 定义粒度：样例中就是粒度定义为POS上发生的第一笔交易，这样可以最大程度的从不同角度进行数据的分析。
3. 识别维度：日期、商品、卖场、促销、出纳员、支付方式等都是该系统的维度
4. 识别事实：销售数量、单价、折扣等



零售模型

事实表的设计

衍生事实（**Derived Fact**）

可由其它事实数据计算得出的数据，称为**衍生事实**。如事实表中有收入数据与成本数据，那么可以依据收入-成本，就可以计算出利润的数值。

这种数据可以在事实表中增加一列，直接计算出结果存储下来，或者在后面用到的时候再临时计算得出。书中给出的建议是直接计算出结果，物理存储下来。这样的好处是减少错误的可能性，虽然只是增加了一点存储的容量。

数值型度量数据的三种类型

事实表中，数值型的度量值一般分为三种类型：
全加型(Additive)：在任何维度条件下，数据都是有实际意义的可加性的。
半加型(Semi-Additive)：在部分维度条件下，数据可加；
非加型(Non-Additive)：在任何维度条件下，数据都是不可加的。如利润率、单价、温度等，不同的值之间相加是没有意义的。

维度表的设计

日期维度的设计

日期维度是数据仓库中特殊的维度，几乎所有的数据仓库都有日期的维度，其中每一列都是一个日期的各种属性。而且，不像其它的维度表，日期维度一般是预先建好的。一般会在表中存放10~20年的数据。

Date Key	Date	Full Date Description	Day of Week	Calendar Month	Calendar Quarter	Calendar Year	Fiscal Year-Month	Holiday Indicator	Weekday Indicator
20130101	01/01/2013	January 1, 2013	Tuesday	January	Q1	2013	F2013-01	Holiday	Weekday
20130102	01/02/2013	January 2, 2013	Wednesday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130103	01/03/2013	January 3, 2013	Thursday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130104	01/04/2013	January 4, 2013	Friday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130105	01/05/2013	January 5, 2013	Saturday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130106	01/06/2013	January 6, 2013	Sunday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130107	01/07/2013	January 7, 2013	Monday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130108	01/08/2013	January 8, 2013	Tuesday	January	Q1	2013	F2013-01	Non-Holiday	Weekday

日期维度表

为什么要有一张这样的日期维度表？有的人可能会问，直接用一个日期列不就可以表示了么？为什么还要再去join另一张日期维度表，增加负担呢？
书中的解析有两个原因：

1. 日期表的join是十分高效的，这种join基本可以忽略；
2. SQL的日期功能不是很强，将一些固定的值事先存放在表中，在使用的时候可以直接利用，而且这种日期的逻辑在维度表中就可以解决掉，不用再到应用层的代码里再进行处理了。

需要注意一点，上面图中的日期维度表的主键最好直接用年月日拼出的整数，而不要用自增列，否则在事实表中查看时，无法直接看出是哪个日期。

用文本代替**Flag**类型属性

日期维度表中，可能存在是否为周末的属性，这种属性一般都是用Y或N，0或1之类来区分。这种值可以减少存储量，但难以阅读。所以，为了方便用户理解，而且免去应用层再用代码去处理，最好直接用文本来代替，如“周末”，“非周末”。

当天与相对日期的属性

日期属性一般不会更新，但有的时候，我们可能需要一些随时间变化的属性，如IsCurrentDay,IsCurrentMonth,IsPrior60Days等。IsCurrentDay属性每天都需要进行更新，用来生成今天的报表。

有的日期维度表包括一个日期间隔的属性(lag attribute)，lag列当天为0，昨天是-1，明天是+1。Lag 列可以做为一个计算列，而不是物理列。

时间维度

如果日期的维度过大，需要考虑到小时，分钟，甚至秒级的时候，需要一个维度来统计。如果将秒级的数据都放入日期维度，会造成数据量爆增。如果需要进行上钻操作，可以另设一个时间维度，精确到秒级，也就24 60 60=1440行数据。如果不需要上钻操作，则可以直接在 fact 表中保存时间。

产品维度

产品有很多的描述型属性，有些属性可以按层级分组，如多个产品属于一个品牌，多个品牌属于一个分类，多个分类属于一个部分等。产品维度表可以允许有数据冗余，保存大量的重复数据。

有产品有一些高重复的属性值时，可以将他们重复的记录在维度表中，不需要完全按范式要求拆成多张表。

Product Key	Product Description	Brand Description	Subcategory Description	Category Description	Department Description	Fat Content
1	Baked Well Light Sourdough Fresh Bread	Baked Well	Fresh	Bread	Bakery	Reduced Fat
2	Fluffy Sliced Whole Wheat	Fluffy	Pre-Packaged	Bread	Bakery	Regular Fat
3	Fluffy Light Sliced Whole Wheat	Fluffy	Pre-Packaged	Bread	Bakery	Reduced Fat
4	Light Mini Cinnamon Rolls	Light	Pre-Packaged	Sweeten Bread	Bakery	Non-Fat
5	Diet Lovers Vanilla 2 Gallon	Coldpack	Ice Cream	Frozen Desserts	Frozen Foods	Non-Fat
6	Light and Creamy Butter Pecan 1 Pint	Freshlike	Ice Cream	Frozen Desserts	Frozen Foods	Reduced Fat
7	Chocolate Lovers 1/2 Gallon	Frigid	Ice Cream	Frozen Desserts	Frozen Foods	Regular Fat
8	Strawberry Ice Creamy 1 Pint	Icy	Ice Cream	Frozen Desserts	Frozen Foods	Regular Fat
9	Icy Ice Cream Sandwiches	Icy	Novelties	Frozen Desserts	Frozen Foods	Regular Fat

产品维度样例

数值作维度还是事实

有一些数值型的值，作为维度还是事实不是太清楚。典型的例子就是产品的单价。一般来说，如果：

- 数值作为计算使用，则存在事实表中
- 数值作为过滤或分组，则可以作为维度表中的一个属性

如果该数值同时要进行使用和分组使用，则应事实和维度表中都存放。

维度属性的下钻（**Drill Down**）

Drill down 是指从维度表中增加维度，把现有的数据按增加的维度进行细分。 上钻（Drilling up）是指除去维度，并将该维度的数据合并。

如下图所示，按照 brand name 下钻：

Department Name	Sales Dollar Amount
Bakery	12,331
Frozen Foods	31,776

Drill down by brand name:

Department Name	Brand Name	Sales Dollar Amount
Bakery	Baked Well	3,009
Bakery	Fluffy	3,024
Bakery	Light	6,298
Frozen Foods	Coldpack	5,321
Frozen Foods	Freshlike	10,476
Frozen Foods	Frigid	7,328
Frozen Foods	Icy	2,184
Frozen Foods	QuickFreeze	6,467

下钻

商店维度

商店维度表中，包括了地理的位置信息，每个商店属于一个地点。我们可以上钻到任何一层的位置，如县，市，州等。因为美国有很多名字相同的城市，所以在维度表中，使用了一个city-state的组合属性。

促销维度

促销维度描述了产品以何种形式进行促销的活动，包括降价、优惠券等。这个维度通常称为causal diemension。

Promotion Dimension
Promotion Key (PK)
Promotion Code
Promotion Name
Price Reduction Type
Promotion Media Type
Ad Type
Display Type
Coupon Type
Ad Media Name
Display Provider
Promotion Cost
Promotion Begin Date
Promotion End Date
...

促销维度

Null外键

有的产品没有促销的活动，那如何与促销维度表进行关联？一般是在促销维度表中有一条主键为0或-1的记录，用来标注无促销的产品销售记录。这样，就可以保证产品销售fact表中没有null值的存在。

退化维度

Degenrate diemension是指在业务系统中的旧主键，在转到数据仓库后，并没有相对应的维度表进行绑定，已经退化，但建议对其进行保留。因为利用该维度，还是可以对原始的事务进行还原。

无事实（**factless**）的事实表

对于没有度量值的事实表，我们称之为无事实的事实表，一般通过利用count，distinct等统计记录个数的方式，得出所需的度量指标。

维度表与事实表的键

维度表的代理键（**surrogate key**）

业务系统中表的主键，一般称为自然键natural key。维度表的主键一般不使用自然键，而是使用无意义的数字作为主键。维度表中记录的主键，称为代理键（Surrogate Key）。一般从1开始，2，3这样顺序排下去，它没有实际的意义，主要是用来与事实表进行联接。

维度表的**Natural and Durable Supernatural Keys**

超自然键

退化维度代理键

日期维度

日期维度代理键一般使用yyyymmdd格式的整形数字

事件表的代理键

事件表的代理键，是由整形数构成，并无实际的意义。一般只用于ETL操作。

抵御一些诱惑

维度表的规范化（雪花模型）

雪花模型符合3NF范式，但带来的性能提升却不如普通维度表的易用与高性能的特点。

##维度使用过多
过多的维度表，会导致事实表过多地连接维度表，对于可用性和查询性能来说，都是一个大问题。

如果维度过多，就要考虑将相关维度组合成单个维度。