# 一、商品库存系统

## 1 商品库存的设计

### 1.1 商品模型设计

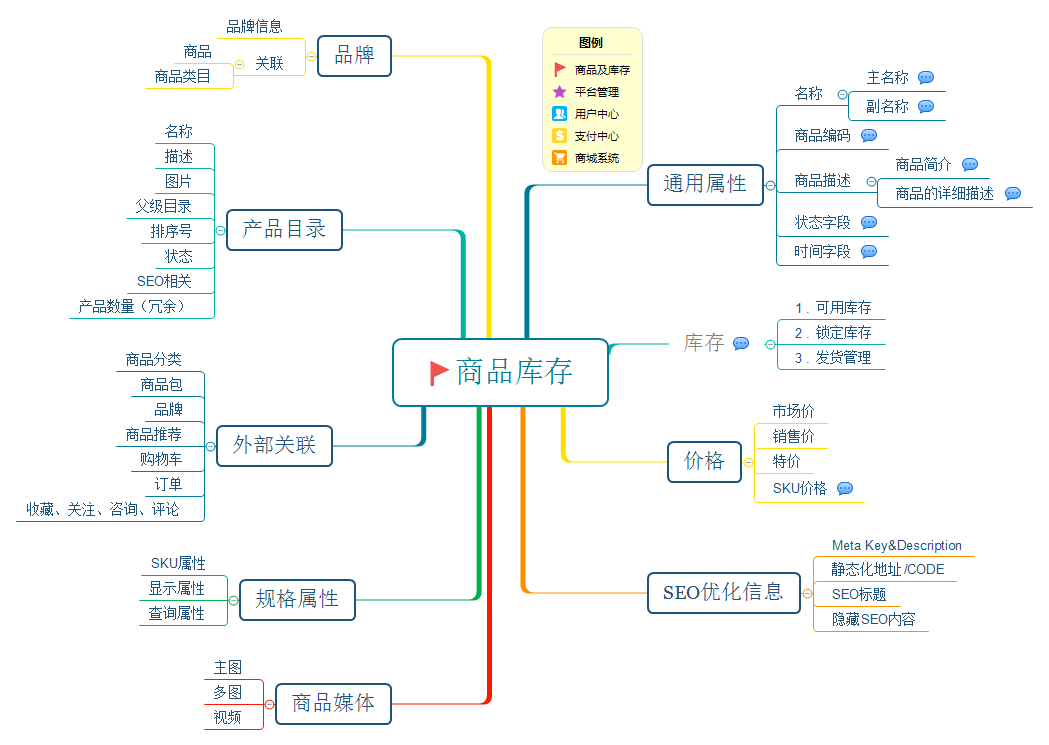


图 商品模块模型图

**通用属性**

指设计在商品表的一些基础字段。

**# 副名称**：由于商品名称经常要加上一些促销信息，如本商品参与什么活动之类。但经常改动主名称

容易导致出错，所以增加此字段来专门管理促销信息。显示时连接到主名称后即可。

**# 产品描述**：产品描述建议另设计一表存放，对提高产品搜索、产品列表显示有帮助。

**# 状态**：常见的状态有未发布、发布、下架等，如果是逻辑删除的，还有“已删除”状态。

**库存**

简化设计，只设计可用库存和锁定库存数量

**价格**

如果系统支持产品SKU，那么实际价格是在产品SKU实体中管理的。

促销价格不在这里管理，在营销管理模块统一管理。

**SEO优化信息**

集中管理各类SEO相关的信息。

**商品媒体**

**#主图**：由于显示频繁，会直接设计在产品表中（或是冗余）。

**#多图**（即附图）：开发中会提供多种查看方式。

**规格属性**

这是设计最困难的部分，也是商品是否可以灵活扩展的关键。

**外部关联**

商品的外部关联非常的多，这里列出了大部分，但随着系统的扩展，肯定会有新的外部关联实体。

所以商品模块的开发，需要提供大量的外部接口或者Tag封装（如商品选取器等。）

### 1.2 商品模块ER图设计

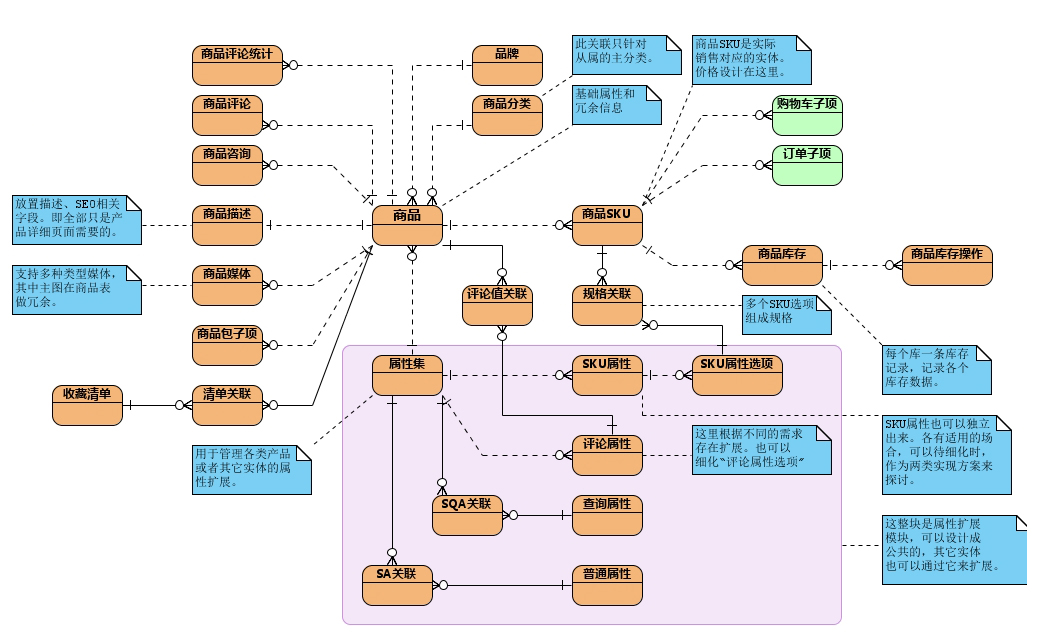


图 商品模块的E-R关系图

**商品**

模块的核心实体之一。承担和内部、外部的关联。该表内设计基础属性和冗余信息。

前台商品详细页面，已本实体的记录作为单元，一条记录一个详细页面。

**商品SKU**

模块的另一个核心实体，从属于商品。每一个商品SKU是商品关联的规格的一种组合。

比如 [颜色SKU-红色] + [尺码SKU-42码] 形成一种组和。这个组合构成一个商品SKU。

价格、库存和关联购物车、订单等，都通过此实体完成。

**商品描述**

和商品是一对一关系，将只会在商品详细页面使用的SEO、描述等相关字段分离出来，对提高商品列表的检索效率会有帮助。

**商品媒体**

通过媒体类型来区分图片、视频和文档等。

**属性扩展模块**

上面粉红色框住的部分是属性扩展模块，通过各类关联为商品模块实现SKU、评论项、查询属性和普通描述性

属性的扩展。从设计上考虑，属性扩展模块并不从属商品模块，它可以为其它的实体（如分类、订单、客户）

提供属性扩展服务。当然SKU属性则是商品独有的。

**商品库存**

这块是可选的设计，需要专门的库存进销存系统支撑，而在我们的项目中，可以先直接在商品SKU实体中设计一个“可用库存”和“锁定库存”字段，留待以后扩展也可以。

### 1.3 SKU分析

在设计商品SKU之前，首先让我们熟悉一下SKU和相关的一些概念。

① 什么是SKU

SKU=Stock Keeping Unit(库存量单位)

同一型号的商品，或者说是同一个产品项目（商品条形码是针对企业的产品

项目来进行定义的），因为产品与产品之间有某些属性不同，用以区别开这些

不同商品的属性即商品变异属性，又称作SKU属性，因为它决定了SKU的绝对数量。

② SKU属性和选项

比如某件衣服有多种颜色、多种尺码，这些属性会直接关联价格和库存的，系统会根据该商品关联的SKU属性的某个组合生成SKU。

比如某个款式的衬衫，有XL/L/XXL三种大小，有红黄蓝三种颜色。

对应这里例子，尺码和颜色都是是SKU属性。

对应尺码的XL/L/XXL等，都是SKU属性选项。

【注】上述的属性不一定在任何时候都是SKU属性，看实际的商品情况和设置。

比如对于尺码，某种商品是均码的。那么就不需要创建尺码这个SKU属性了，

而是设置为普通属性，仅作为显示用。

③ 什么是商品SKU

商品SKU实际上就是SKU，为了避免误解和SKU属性混淆，我用商品SKU来命名，表示从属于商品的、实际销售和存储的子实体。

一个商品SKU，表示该商品关联的若干SKU属性的的属性值的某个组合所形成的子实体。

如对应上面的例子，其中的一种组合 XL + 红色 就会形成一个商品SKU。然后，我们可以在该实体上管理价格、库存、专门的图片等信息。

④ 商品变种（变异）

商品变异其实就是商品SKU，只不过在某些技术文章中这样定义了。即以“变异”来表达商品SKU的生成。

⑤ 属性集

用于管理各类扩展属性的集合，其中SKU属性也是在管理范畴之内。

商品通过关联属性集而获得该属性集设置好的SKU属性，然后才可以根据这些SKU属性生成商品SKU。

属性集也成为产品类型。

常见的属性集有：服装、PC、家具、图书等。

### 1.4 本系统对SKU相关概念的统一定义

1）SKU（或称商品SKU）指的是商品子实体。

2）商品和商品SKU是主次关系，一个商品包含若干个商品SKU子实体，商品SKU从属于商品。

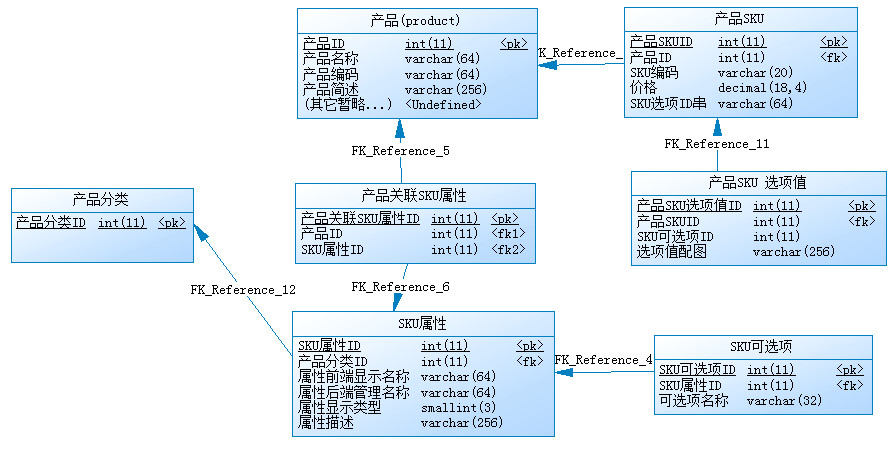
3）SKU不是编码，每个SKU包含一个唯一编码，即SKU Code，用于管理。

4）商品本身也有一个编码，即Product Code，但不作为直接库存管理使用。

有时为了方便管理，会通过商品的Product Code作为前缀生成SKU Code。

### 1.5 SKU属性集管理

在我们的系统中主要考虑2种模式：

① 商品独立关联

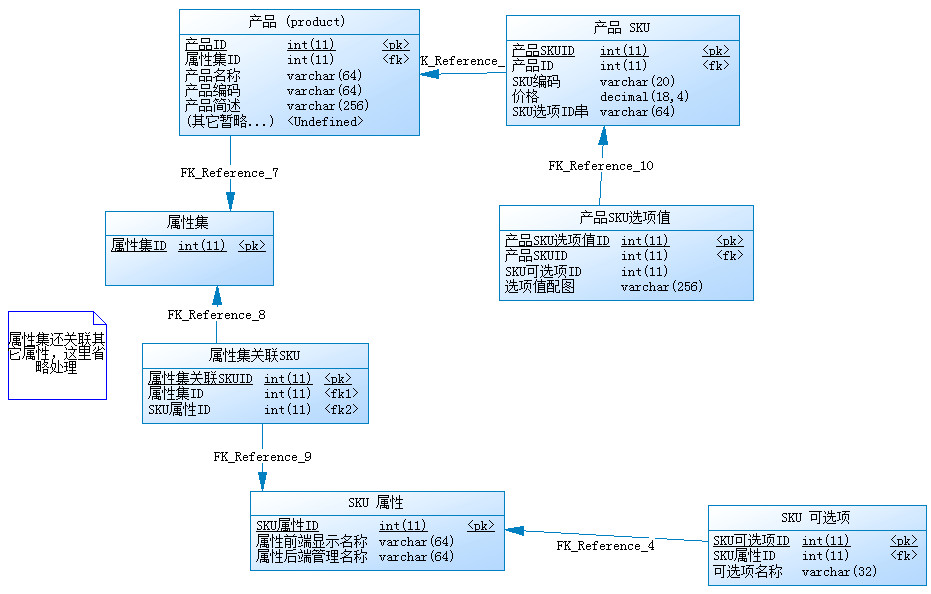
即SKU属性是公共的，每个商品根据自己的实际情况来关联若干个SKU属性，

然后选择若干选项生成商品SKU。

优点：灵活，设置好了若干SKU属性和相关选项后，由商品自行选择相关的组合。

缺点：太灵活了，容易出错。比如对于颜色这个SKU属性而言，由于是公共的，所以会定义数十个，甚至更多，但是对于ipad而言，其实只需要2个即可。这种管理方式，无法从源头控制商品添加时SKU属性和选项的范围。

② 通过属性集关联



即SKU属性是公共的，通过属性集关联。

优点：这种方式重用性不错，对于SKU选项较多的，会简化工作量。

缺点：由于SKU属性是公共的，所以会存在相同属性名的选项值差异较大且不能统一的问题。但我们也可以通过设置多个同名、但编码不同的SKU属性，如多个“颜色”SKU属性，但是选项不同。

### 1.6 商品类目设计

在电子商务系统中，产品目录是组织网店产品的一种分类方式。

产品目录一般为树形结构，结构上支持无限分级，但是最常见的是2～3级，很少4级或者以上的（B2B类网站除外）。

比如有类似京东、卓越一样的百货式商场，目录很丰富，图书/家电/手机数码等，每个一级目录下，还划分二级/三级子目录等，也是三级就满足了业务需求。

而对于一些专业类的网站，如草莓派/麦包包等，两级就可以了，其它的细分通过扩展属性来满足。

产品目录是用户定位产品最为常见的方式，所以网站设置产品目录需要仔细斟酌，根据网站运营特点来做设置。一般而言，建议能够两层解决的不必用三层，实在需要三层的，可以将第三层隐含在搜索条件中，总之让顾客尽快的到达产品目录对应的产品列表页面乃至找到对应的产品。

## 2 产品关联推荐

### 2.1 交叉销售 Cross Sell

定义

　　商家根据客户已经选择的产品推荐其它产品。例如在客户浏览苹果iPad时（或者加入购物车后），推荐适合iPad配套的"ESK ipad2鳄鱼纹皮套"。

原则

　　交叉销售的原则是基于客户在购买当前产品的前提下购买更多产品。这些产品有可能是相关的（如附件类产品/配套产品），也有可能是基于统计或者从众的购买建议（如Also Bought，即购买了该产品的顾客也购买了...。We guess you would like...我们猜你可能会喜欢...）

交叉不会推荐同类和相似的产品，而是推荐一些完全不同的产品和服务（如周边产品）。

**常见类型**

Also Bought：购买了当前产品的顾客也购买了...。通过分析订单信息获得。

Also View：浏览过当前产品的顾客也浏览了。

View and Bought：浏览了该产品最终购买了。通过分析浏览数据和订单数据获得的数据，并显示百分比。（见京东）

配件型产品：如在单反相机下面显示电池/三脚架/镜头等。

互补型产品：如在外套产品下显示裤子/鞋子等凑成合适的一套（需要较高的搭配水平）

经常一起购买的/最佳拍档：关系很紧密的两个或者多个产品，通过分析订单获得。并非产品包，只是一种推荐。（见卓越/当当）

和你兴趣相似的顾客还关注了：当当的一种类型。我觉得倒不如改成关注了当前产品的顾客还还关注了。

### 2.2 向上销售 Up Sell

定义

　　商家基于客户选定的产品推荐更好的产品（对客户或者对商家自己）。例如客户浏览某款平板电视时，推荐更高型号/进口品牌的产品给顾客。

原则

　　向上销售的原则是替换式，即建议购买我们推荐的，而不是当前浏览的。向上销售如果不是基于足够的统计数据，一般不好处理。只有在熟悉当前客户的各种消费喜好的前提下，这样的推荐才比较精准和不会引起客户的不快。

向上推荐会推荐同产品分类下的或相似的产品，但更贵的或者对商家更有价值（推广/利润高）的产品。

**常见类型**

同品牌的更高档产品：由于是同品牌的推荐，兼顾了顾客的品牌忠诚的可能性。

不同品牌的更高档产品：一般是推荐更好的品牌，但如何界定“更好”需要考虑。

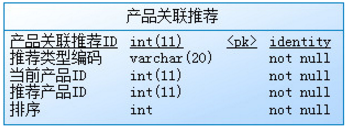
价格相似：这里不一定是向上了，但由于满足替换原则，所以也归入向上销售类。

替换推荐：当当前产品缺货时，推荐其它有货的相似产品。（减少因为缺货导致顾客流失）

### 2.3 实现关联推荐常见有三种模式

① 人工模式

这是最为基础的模式，也是其它模式都必然会支持的功能。即使是一些看上去可以通过算法获得真实的数据（如Also Buy）的交叉销售，在实际运营中也有人工维护数据的需要。一般而言，人工维护的数据总是在最上面/最前面的，后面才是自动搜索/统计获得的数据。当然，如果结果集是定时搜索出并将关联持久化的，那也可以再维护两类数据（人工维护+自动生成）的数据位置。



人工维护的表设计最为简单，如果关联推荐类型也是固化的，甚至只需要一张“产品关联推荐表”即可。



② 人工+自动搜索

如果只提供人工维护的方式，对于产品数据量大的网站，这类数据维护的工作量也会很大，而且在人工未维护到的情况下，在前台的产品详细页面的这些关联推荐栏目下将没有显示（或者数量不足）推荐的产品，这必然导致用户体验的不佳和营销的失当了。

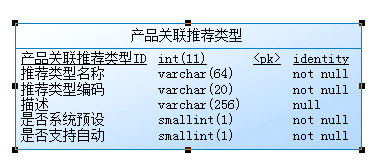
对于这种情况，则有必要引入自动搜索作为补充。

比如某个栏目下显示4个产品（页面模板设计决定的），如果没有人工维护，那么会根据该推荐类型对应的算法搜索出4个产品来显示。如果人工关联了2个产品，则剩余的2个产品由自动的来补充。

产品关联推荐的不同类设计的数据来源比较复杂，搜索算法也各有不同。比如Also Buy

是根据订单+订单子项+当前产品关联查询得出，而Also View却根据产品浏览统计数据。所以固化常见的若干条算法成了一种不错的选择，而且也能够满足大部分的业务。

      基于前面第二节的表设计，在产品关联推荐类型表增加一个字段：是否支持自动。



 当该字段=1时，表示该推荐类型支持自动算法，只有系统预设的推荐类型该字段才可能是1（系统预设的不一定支持自动）。