Salsify Key Concepts: Data Model Components

This article is primarily focused on new Salsify implementations. If your organization is interested in updating an existing Salsify account with these modeling components, please reach out to your Customer Success Manager to discuss.

A strong data model is the key for a good governance policy. Your data model can:

- Make product navigation easier
- Help reduce data duplication and error
- Facilitate publishing data to retailers and other data partners
- Allow you to give the right team members access to the information they need to work with
- Support efficient collaboration on content development and improvement

Getting familiar with both your data and the Salsify tools available to model the information are keys to helping you find the best model to suit it. You'll use the following Salsify tools to build your data model. Note that building and modeling data in Salsify is an iterative process, and you can add to and change your model as you go. Starting with a model that best suits your data will also make it easier to make the changes you need to along the way.

Model Components

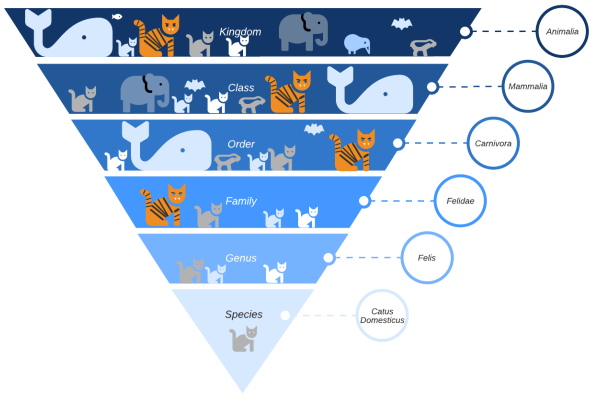
Your data model is made up of a few key components. How you put them together will determine what your products are made up of, and where they fit into your overall data structure.

Taxonomy

Taxonomy in Salsify is the structure that defines what a product is, and the pieces of data that make it up. Each product fits into the taxonomy once, and where it sits in the taxonomy determines what will be included with its record.

Taxonomy is a classification and a hierarchy that defines the things in its structure. The structure is made up of nodes, which are branches of the taxonomy. It's a lot like a biological taxonomy - every living thing has a specific place in the hierarchy, and the details get more specific as you move from the top to the bottom of the taxonomy.

You can place each product only one time in the hierarchy so your goal is to create a specific taxonomy that will allow you to place the right characteristics at node and level to accurately describe your products, and continues to be described by the hierarchy above it.



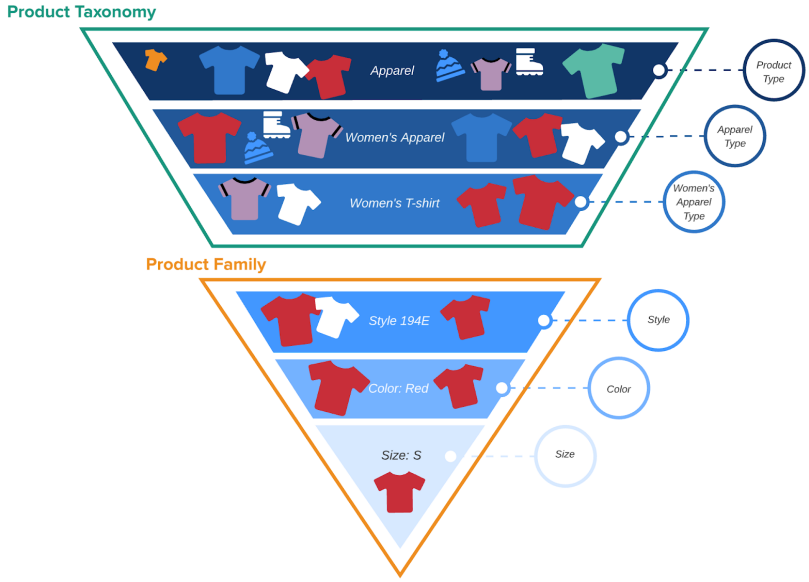
In this biological hierarchy, the top level characteristics broadly describes all the animals included in the hierarchy - they're animals, rather than plants, and that's really all you can say they have in common.

As you move down the hierarchy, the next node down describes a more narrow set of animals with more in common. They split off into other nodes, where the descriptions apply to them.

This taxonomy, like the Salsify taxonomy helps us gain efficiency, because where we can use terms to describe a broad range of animals in this case, and products in our case. And those same terms apply down through every level. But as the attributes get more specific, we'll define more specific traits at each level that define what that product is.

In Salsify, when we get to the extremely specific levels of our products, we have some other tools that more efficiently pull information together to apply to products with traits in common.

Once assigned a taxonomy value, Salsify knows what properties are applicable to the product and how the product family should be structured (if assigned). So, our product data model is defined implicitly through its placement in the taxonomy. If you change the taxonomy node, you change things about the product.



Salsify Taxonomy Rules

As you're thinking about your taxonomy, keep a few things in mind.

- There can only be one taxonomy property in an organization
- A product can only have one assignment in the taxonomy (it can only be in one place)

- A product family (see below) can only have the taxonomy assignment at the highest part of the family

For example, our company sells apparel and other items like exercise equipment and protein bars and shakes. We set up the taxonomy with separate nodes for those product types, because the details about them are very different. And we'll use validations to define and set rules for the details for the products in each of those nodes of the taxonomy.

Validations

Validations go hand-in-hand with taxonomy. Validations are how Salsify sets the rules about what makes up the components of a set of products, and what users are allowed to add to them.

You'll define the rules that make up the products in each node at a high level, and as products get more specific in that node, you can also use product families, which we'll talk about in the next section, to build out sets of details for the products.

So from our example, we add validations to describe our t-shirt:

- Every piece of apparel should have a name, an image, a description, 3 feature bullets, a SKU and style
- T-shirts should indicate whether they have a pocket
- T-shirts should have a material and that value should be one of: cotton, linen, polyester, or rayon
- T-shirts should have a sleeve length
- T-shirts should tell us whether they have a crew neck or a vee neck

These are validations we set throughout the taxonomy to define what makes up a t-shirt. Then as we add products to those taxonomy nodes, Salsify will know it needs the information that describes a t-shirt for those products.

So far, we've been requiring that there is a value, but that value will vary depending on what the product is - a shirt may or may not have a pocket, and the sleeve length will vary between styles. But at the style level, we have shirts that should all have the same description, the same material, etc.

Product Families

A product family is a set of products with inheritance-based relationships. They're sellable products that have many common properties, and the family further defines what makes up the products in it. For example, a women's t-shirt may come in a wide variety of colors and sizes which may result in dozens of sellable variations. They vary by two traits, size and color. But they all have a number of traits in common like bullet points, description, and other details.

Managing these traits at an individual SKU level would result in a lot of time spent editing and inconsistent values. You can use product families to define products that should have traits in common, and apply them across those products.

You'll define a product family for the category. In this case, we'll assign a three-level product family: Style, Color and SKU. Because you assigned properties to the node in the Taxonomy & Validations section, Salsify also knows to check for a Name (at the Style level), a Description (at the Style level), a Swatch (at the Color level), a size (at the SKU level), etc. We set up the style family to contain a style ID, material and other details about the style that are common across all the variations. We set up our family model as Style > Color > SKU (size). So all variations of color and SKU inherit the values associated with Style, and all variations of SKU also inherit the color level properties.

In this way we use the taxonomy and product families together to maintain consistency and gain efficiencies across the width of the style variations.

Product References

With product references, you can link independent products together to make up the components of a single product. For example, in our demo company, PXM Industries, we sell disguise kits for super heroes. The kit contains a trench coat, 3 pairs of sunglasses and 2 wigs, and we sell each component separately as well. So we'll use a Quantified Product Reference to store the components with our kit.

Bundle Components

3 References

Quantity		Product ID	Name
3		908040634	Women's Sunglasses - Tough Love
2		908040779	Women's Disguise Wigs - Auburn Diana Prince
1		908040147	Women's Stealth Trenchcoat - Khaki L

In this case, we use product references to pull together all the items included in the bundle. The bundle kit sits in the taxonomy where it should so that it has the details it needs, and the component products are stored in a property on the kit which shows what products the pieces are, and each product maintains all its characteristics.

Product References are one-way relationships between products, with no automatic inheritance. You can store product(s) in a reference property that relates them with each other. References can be used by parts of the platform, via formulas, to pull information about the products stored in the referenced property to products storing the reference. In this example, the products that make up this kit are stored in a property called Bundle Components. Storing the components this way also gives you access within this product to all the details for each of the components. So for example, when you're publishing this product to a retailer, you could include the short description of each component as a bullet point to describe what's included in the kit. Or you could use a formula to add up the shipping weights to send for the kit product. Whatever information is stored on each component is available to the product that has them stored in a property.

References are useful for use cases like the following:

- **Upsells / Cross Sells:** Cases that relate two products together, suggesting products related to the product they're associated with
- **Parts / Bills of Materials:** Identify the component parts of a product, indicate their quantities, and compute your costs
- **Collections / Brands:** Create non-product entities and link products to them
- **Kits / Bundles:** Create non-product entities, tie products and quantities to them
- **Packaging Hierarchy:** Represent levels of packaged units like eaches, cases and pallets.

References can be used by parts of the platform, via formulas, to pull information about the products stored in the referenced property to products storing the reference. Product References:

- Allow a single source of truth for each piece of information/product for maximum flexibility
- Avoids duplication of information and better governance of product details.
- Supports modeling more complex distribution packaging levels like case packs, pallets, etc.

Click here to learn more about product references.