**Product Collection Analysis**

**Introduction**

This document is for only product database. It includes various information or details of products. There are 2 categories in this documents that is Products and Product Variants.

1. **Products Collection:** It contains various information of products which includes shopID, title, vendor, price, etc. This all information is very useful for the customer perspective.
2. **Products Variants:** Product Variants are nothing but in simple terms we can say that it’s a same product with different types or different color so that customer can get more options to select product of their choice by creating product Variants. This category is little bit different that Products. It includes fields like taxable, barcode, CompareAtPrice. It means products price can be varying.

**For Example:** Suppose if customer wants to buy ‘Toshiba External HDD’, product will be same but by creating product variants, customer will get different colors, size, types option within same product.

some products can have ID, title, vendor, description, type, price, etc.

**As we can see below table contains several rows and columns:**

Columns consist of Fields, type, Optional. These 3 categories are important for product analysis. Rows consists of Product information like product ID, product description, vendor, price, etc. This database is helpful in describing a particular products and its type and its categories.

1. **Fields:** It is very useful part of database. Each records contains several data fields. Fields are individual parts which contains information of particular record.

**For Example:** record having field name price which means that price is a one piece of data which contains information about price of a particular record.

1. **Data Types:** Generally, data types are useful to rectify the relation between particular data field is of which type. There are several data types in MongoDB can be used and among them we used few data types for product analysis which are described below:
2. **String:** String in mongoDB must be UTF-8 valid. UTF-8 is a 8 bit code and this name came from universal Coded character set + Transformation Format.
3. **Integer:** It can be used to store numerical value. Value can be 32 bit or 64 bit depends on your system.
4. **Boolean:** This value is just used for True/False value.
5. **Object:** It is used to store embedded documents
6. **Date:** It is used to store date or time as UNIX format and you can create your own Date or Time by creating object.

**For Example:** Number is one of the data type which is used to store information numerically. Another example is Date/Time which is used to store value in date/time format. No other character can be used if we define a particular data type to a field.

1. **Optional:** Optional fields means if a particular field must enter value or it can be left blank or not necessary to input value. In database if optional value is set to true that the particular field can be left blank or it’s not necessary to input value. If optional value is set to false it means that particular field must be require and it cannot be left blank.

**For Example:** Description field has optional value set to true for a particular product, it means it’s not compulsory to fill that field with values.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No.** | **Fields** | **Data Type** | **Optional** |
|  | \_id | String |  |
|  | ancestors | String |  |
|  | shopId | String |  |
|  | title | String |  |
|  | pageTitle | String | True |
|  | description | String | True |
|  | type | String |  |
|  | vendor | String | True |
|  | metaFields | ReactionCore.Schemas.Metafield | True |
|  | positions | Object | True |
|  | price | ReactionCore.Schemas.PriceRange |  |
|  | isLowQuantity | Boolean | True |
|  | isSoldOut | Boolean | True |
|  | IsBackOrder | Boolean | True |
|  | requiresShipping | Boolean | True |
|  | parcel | Boolean | True |
|  | hashtags | String | True |
|  | twitterMsg | String | True |
|  | facebookMsg | String | True |
|  | googleplusMsg | String | True |
|  | metaDescription | String | True |
|  | handle | String | True |
|  | isVisible | Boolean |  |
|  | templateSuffix | String | True |
|  | createdAt | Date |  |
|  | updatedAt | Date | True |
|  | publishedAt | Date | True |
|  | publishedScope | String | True |
|  | workFlow | ReactionCore.Schemas.Workflow | true |

**Product Variant Schemas Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No.** | **Fields** | **Type** | **Optional** |
|  | \_id | String |  |
|  | ancestors | String |  |
|  | index | Number | true |
|  | barcode | String | true |
|  | compareAtPrice | Number | true |
|  | fulfillmentService | String | true |
|  | inventoryManagement | Boolean | true |
|  | inventoryPolicy | Boolean | true |
|  | lowInventoryWarningThreshold | Number | true |
|  | inventoryQuantity | Number | true |
|  | minOrderQuantity | Number | true |
|  | price | Number | true |
|  | shopId | String |  |
|  | sku | String | true |
|  | type | String |  |
|  | taxable | Boolean | true |
|  | title | String | true |
|  | optionTitle | String | true |
|  | metaFields | ReactionCore.Schemas.Metafield | true |
|  | createdAt | Date | true |
|  | updatedAt | Date | true |