**Design Task**

**Description**

Currently we receive a full customer catalog on an hourly basis. Each time we run over all products in the catalog. For every product we run our algorithm on product image to fetch the deeptags for the image. Then we write the catalog data, with the new calculated deeptags into the Elastic search database. So each run creates a new index which is a production index. Meaning this is an index that we use for searching customers' products.

**Problem**

The process as described above is not ideal, and certainly not efficient. Usually not all products change from one indexing to another. Some stay as they were, some get small changes - like change in price, in\_stock count. Other products have change in the images themselves. And of course there are new items added and items that no longer exist in the catalog, but did exist in the previous version of the catalog.

**---------------**

**Possible Solution:**

A possible solution may be to calculate a kinde of checksum (hash) value of the main fields for each product at the time of receiving the catalog, especially for images(using metadata, timestamp, checksum..). Store this checksum in some key-value storage, or in a relational database.

Next time resieving a catalog, calculate checksum of a product and compare it with the one that was saved during the previous catalog analysis.

If the checksum of the product has not changed, use pre-calculated deep tags.

If the checksum is different or the product is new, run the deep tags generation algorithm.

Products that are not present in the new catalog are not included in the updated index.