**Question**

Sales Rank:

A large e-commerce company wishes to list the best-selling products, overall and by category. For example, one product might be the #1056th best-selling product overall but the 13th best-selling product under “Sports Equipment” category and #24th best-selling product under “Safety”. Describe how you would design this system.

**Assumption**

1. There is no limit to the product and everything has to be ranked. It is not like top 100 overall or a particular category.
2. No need to update the rank real time. It gets updated after a particular interval 5 hours. (We could leverage Cache mechanism to achieve this)
3. Best-selling is purely based on the quantity of product sold.
4. Category of the product will not alter too often.

**Solution**

1. It won’t be a real time ranking system. It will be refreshed after a particular interval (5 hours)
2. We would cache the ranking results on a web cache so that every request will be responded with the same detail throughout the time period until next refresh.
3. Before the time interval ends the process to rank occurs (i.e., it runs parallel such a way that the query is made to sort the commodities by overall/category). Since we are serving from the cache the ranking mechanism would not be disturbed.
4. We could use two servers one for **Cache** mechanism and **updating the product** bought in the Database and another server which purely concentrates on **ranking** and providing it to the cache. Here since the same Database (Resource) is being used by two servers we might need to implement it asynchronously (Thread)
5. Ranking Mechanism:
   1. Merge Sorting (worst case is n log n). Since the whole list of product has to be taken it would be useful if we query the database and get the whole list and then sort them to rank overall.
   2. Many lists are created as many as the no. of category and we are iterating through the overall rating and copying the Products to each list as needed. Suppose a Product is of the #1056th best-selling product overall but the 13th best-selling product under “Sports Equipment” category and #24th best-selling product under “Safety” when we come to this product there will be already 12 Sports Equipment Category present and 23 Safety Category present. Now we add the Product to both the lists.
   3. Once everything is done, we save them in the cache with their ranking as the key. Assumption here is that the e-commerce website only will be categorizing using rank.
   4. Big O will be obviously less than n2 since Big O (n) will be required for moving through the products. Worst case of the sorting process is n log n
6. Saving Details/ Fetching Details from Database

We could use the thread concept to toggle between the two servers to use the same Database ensuring each gets sufficient time and the query does not time out. If this process is not achievable we need to use the normal manner but it will have a little time lag.

**Cache Web Server**

**Ranking Web Server**

**Database**

**Users**

**2**

**1**

**3**

**4**

**2**

Check for availability to save the product

**3**

Saving Details (only if the database resource is available)

**4**

Fetching details for ranking (if this happens the saving in not)

**1**

Saving the Ranked Products in the Cache Web Server

**Thread based implementation if it is non-thread then point 2 is not present and there is no availability concept involved**

**Modification (Optimal Solution)**

Here one thing when we are working on the ranking system, we could do it using the **Database** table and then join them based on the product id and so duplicates of the same product will obtained and then sort them according to the category.

If the database is considered an overhead we could leverage **MapReduce** technique to gather the details in a text file (only the details of a particular week for each product) and then process the ranking mechanism also based on the category.

In the database approach we could additionally think of storing the in-cache for some time and then bulk update to the database. We need to take care the updates and ranking occurs one after the other not simultaneously.