**Sales Rank:** A large eCommerce 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" and the #24th best-selling product under "Safety." Describe how you would design this system.

**Solution:**

Create a database which lists the products and **index** based on the category.

Each product can have **multiple** category

List is going to be showing the entire list and their ranking. We could pre compute using the ranking system and then alter as the new products come in and existing is longer sold.

We could use the binary tree mechanism which ensures that search is based on log n and so when the product is removed we can alter that ranking in an efficient and easier manner. Create a self-balanced binary tree for each category. Example: Graph DB (neo4j)

Functions:

|  |  |  |
| --- | --- | --- |
| **Administrator** | **User** | **Ranking Algorithm** |
| Add Products  Remove Products  Add Offer  Remove Offer | Purchase  Return  Review/Feedback | Periodically update the ranking for the entire list (Pre calculate the ranking using a system independent of the other functions) |

**Tree** - BigO (log n) retrieval – re-sort the products frequently is easier

**HashMap** - BigO (1) retrieval – re-indexing will be difficult

**LinkedList** – retrieval is not easy – re-sort is not so easy compared to tree

Graph data structure to hold the products based on the sales ranking. Day of arrival and no. of copies sold help in determining the rank for each product

|  |
| --- |
| **Product** |
| id  name  List<Category>  dateOfArrival  sold  price  discount  seller  review  description |

1

2

3

4

N

**Clients**

**Load Balancer 1**

**Load Balancer 2**

**Active – Passive Load Balancer**

**1**

1

2

3

N

**Cluster of Application Server**

**Read Replicas**

**Master – Master**

**2**

**3**

**Sales Rank**

**Web Service System**

**Batch Jobs at regular interval to re-index based on sales**

1 – Client request to the web application for Purchase, Return, Review/Feedback actions

2- Application server will update the master – master database

3 – When read request is made the replicas are utilized

4 – When the product listing service is needed a separate web service system is invoked (Sales Rank)

5 - The ranking system is periodically updated with the batch job run at read replicas to reduce load on write intensive servers

6 – Administrator will have privilege to add, remove, update product and the details are used in the next ranking batch call

**5**

**4**