Database Design Assignment

**2. Explain about searching performance. How will you handle replication in SQL for searching & Reporting?**

Searching performance is the speed and accuracy with which the application can return search results. This can improve the searching performance by using a search engine that is optimized for ecommerce applications.

We are using Elastic search Engine which is a powerful search engine that can be used to improve the searching and reporting capabilities of an ecommerce application.

Elasticsearch can be used in an ecommerce application as follows:

* **Product search**: Elasticsearch can be used to index the product catalog and make it searchable by product name, description, attributes, and other criteria. This can help customers to find the products that they are looking for more quickly and easily.
* **Customer search**: Elasticsearch can be used to index customer data, such as name, email address, and purchase history. This can help customer service representatives to find the information that they need to help customers more quickly.
* **Reporting**: Elasticsearch can be used to generate reports on a variety of metrics, such as sales, customer behavior, and product popularity. This can help businesses to track their performance and make informed decisions.
* **Personalization**: Elasticsearch can be used to personalize the search results for each customer. This can be done by using factors such as the customer's purchase history, browsing behavior, and location to rank the search results.
* **Analytics**: Elasticsearch can be used to analyze customer behavior and website performance. This can help businesses to understand how customers are using their website and make changes to improve the user experience.

**To handle replication in SQL for searching and reporting in an e-commerce application, we can consider implementing database replication.**

Database replication involves creating multiple copies of the database on different servers to distribute the load and improve performance.

Load balancing and load distribution are crucial components in enhancing the searching and reporting performance of an e-commerce application. They ensure that the incoming requests are efficiently distributed across multiple servers, preventing any single server from being overwhelmed and leading to improved overall system performance.  
  
One common approach is to set up a **master-slave** replication configuration. In this setup, all write operations (inserts, updates, deletes) are performed on the master database, and the changes are replicated to one or more slave databases. The read operations (searching and reporting) can then be distributed among the slave databases, allowing for faster and more efficient searching.  
  
Another option is to use **sharding**, which involves dividing the database into smaller, manageable partitions called shards. Each shard can be hosted on a separate server, and specific data is stored in a particular shard based on a predefined rule (e.g., product categories). This approach can help in scaling the system horizontally and improving searching performance.

By combining load balancing and load distribution with database replication, indexing, sharding, and caching strategies, we can design a database that can efficiently handle searching and reporting tasks with improved performance.

**3. Explain what major factors are taken into consideration for performance.**

**Major Factors Considered for Performance:**

**Data volume:** The amount of data that will be stored in the database will have a significant impact on its performance. As the amount of data increases, so too will the demands on the database.

**Database Indexing:** Properly indexing the frequently queried columns improves search and retrieval performance.

**Query Optimization:** Writing efficient queries and avoiding complex joins or subqueries can significantly impact performance.

**Caching:** Implementing caching mechanisms (e.g., in-memory caching or using caching servers like Redis) can reduce database load and response times.

**Replication:** Replication is the process of creating copies of the database. This can help to improve performance by distributing the load across multiple servers.

**Denormalization:** In some cases, denormalizing data (reducing the number of joins required) can improve read performance, though it may introduce data redundancy

**Hardware and Infrastructure:** Ensuring sufficient hardware resources and a well-optimized infrastructure play a crucial role in performance.

**Load Balancing:** Distributing incoming traffic across multiple servers can help evenly distribute the workload.

**4. Mention about Indexing, Normalization and Denormalization.**

**Indexing** is used to improve the performance of database queries. An index is a data structure that stores the location of data in a database table. When a query is executed, the database engine uses the index to quickly find the data that is needed.

**We will create following indexes for fast searching-**

|  |  |
| --- | --- |
| **Tables** | **Columns** |
| Product | id, name, price, discount |
| Customer | id, name |
| Seller | id, name |
| Order | id, orderDate, totalAmount, customerId |

**Normalization** used to improve the data integrity of a database. It involves dividing the data into multiple tables and ensuring that each table contains data related to single entity only. It reduces data redundancy and inconsistencies.

All tables are normalized except Seller and Customer. **Example-** Product table has been normalized by creating separate tables for the product category and product image.

**Denormalization** used to improve the performance of database queries at the expense of data integrity. It helps to speed up queries that retrieve data from multiple tables.

**Example-**

Seller and Customer table are denormalized as we have not created separate tables for address which can increase the complexity of the design.

**5. How will you handle scaling, if required at any point of time.**

The best scaling for an e-commerce application database design depends on the specific needs of the application.

Since we assume that there are large number of users so we will use Horizontal scaling which will add more servers to the cluster as needed. This is done by distributing the database load across multiple nodes, which can help to improve query response times and reduce the risk of bottlenecks.

There are several ways to implement horizontal scaling like **Sharding** technique which involves diving the database into multiple partitions, **Replication** which involves creating multiple copies of the database, etc.

**6. Mention all the assumptions you are taking for solutions.**

* The following are the assumptions that are being made for this assignment:  
    
  The e-commerce application will be used by a large number of users searching for products simultaneously.
* The system is expected to handle a large number of concurrent users and transactions.
* The choice of databases for different modules is flexible based on specific requirements and can be subject to change as the application evolves  
  The e-commerce application will sell many products.
* We assume that one customer has only one cart to add products.
* We assume that we have multiple products in one orderId.