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

Ans: For performance I have considered following factors :

1. Indexing: Properly indexed the frequently queried columns and foreign keys

significantly improve performance. So that we don’t need to scan the whole tables.

1. Normalization: Properly normalized table for reducing data redundancy increase data integrity across tables. Ex. Separated order and order details table to reduce redundant order details data in order table.
2. Query Optimization : Regularly optimizing the database queries, analyzing query execution times can help identify performance issue.
3. Scalability: Scaling the application when there are high number of traffic can we catered using techniques like sharding , partitioning or replication to achieve scalability.

4. Mention about Indexing, Normalization and Denormalization.

Ans**: Inventory Module:**

For indexing, I have applied indexing on the following:

1. Index on the "product\_name" column in the "Product" table can speed up product searches by name.
2. Index on the "price" column in the "Product" table can even make fast retrieval of products based on price range queries.

For Normalization, I have separated the "Seller" table from the "Product" table, and avoided duplicating seller information for each product. Instead, the "Seller" table has a primary key (seller\_id), and the "Product" table references the seller\_id as a foreign key.

The "ProductSKU" table stores SKU details separately, so that we can avoid the need of repeating product information for each unique SKU.

**Cart/Order Module:**

For indexing, I have combinedly indexed order\_date and status column in order table for faster

retrieval of data on certain conditions.

For Normalization,

The Order table is normalized to store order-specific data and OrderDetails table is used to capture individual product details within an order.

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

Ans: Considering the case if there are high traffic in the application, then we can go with horizontal

scaling so that we distribute the loads across multiple machines. This approach will allow the

system to handle a higher number of concurrent requests by adding more resources in parallel.

When scaling horizontally, load balancers are used to distribute incoming requests across

multiple servers thereby ensuring no single server is overwhelmed with traffic and helps

maintain consistent performance.

We can implement caching mechanism to reduce the load on the database and application

servers. We can use In-memory caching to store frequently accessed data, to reduce the need

for repeated database queries.

Also, since we are breaking down the application into multiple decoupled microservices which

will allow for independent scaling of different applications.

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

Ans: 1) Assumption for Inventory Module:

* The "Inventory Module" is implemented using SQL Server, which handles product management, including adding products, their details, images, and SKUs.
* Products are stored in SQL Server tables, and product images and SKUs are stored in separate tables, connected through foreign key relationships.
* Reason for SQL Server is because it provides a robust relational database system well-suited for structured data like products and their details.

2) Assumption for Cart/Order Module:

* The "Cart/Order Module" is implemented using MongoDB, a NoSQL database, which handles cart and order management for buyers and sellers in the application.
* Buyers can have multiple carts, and each cart can contain multiple products with their quantities.
* The "Order" entity represents the final order after the buyer confirms the purchase, and the "OrderDetail" entity stores the details of each product in the order.
* The "Cart" and "Order" entities, along with their associated product information, are stored as documents in MongoDB collections.
* Reason for MongoDB is based on its ability to handle large volumes of rapidly changing data and its ability to scale horizontally as the traffic grows.

3) Assumption for Auth Module :

* The " Auth Module" for user authentication and authorization is implemented using SQL Server.
* User data, including login credentials, roles, and subscription preferences, are stored in SQL Server tables.
* The assumption for using SQL Server is based on its ability to manage structured user data, enforce constraints, and handle authentication efficiently.