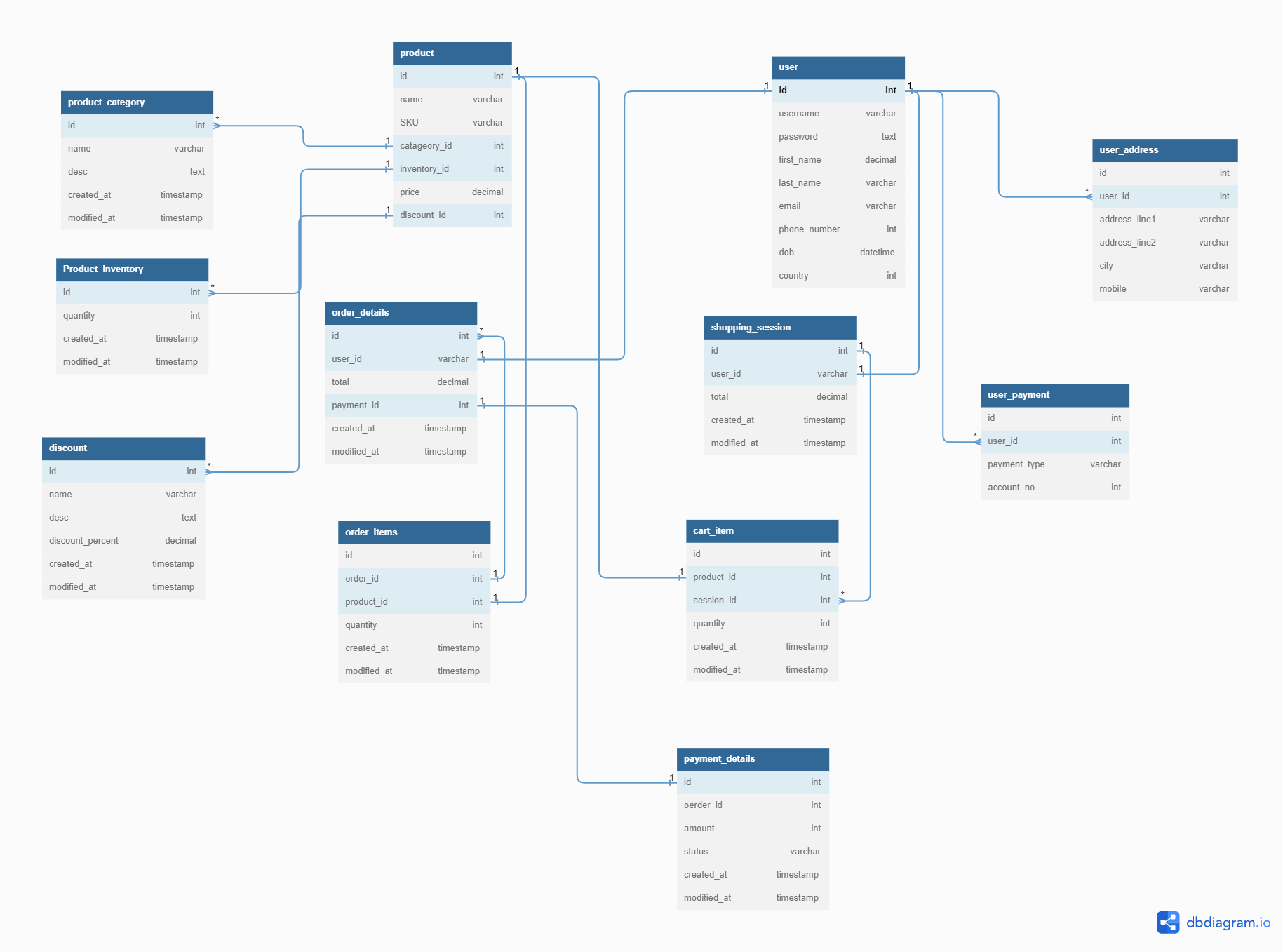
Database Design Assignment (E-commerce Application)

1. Physical Entity Relationship diagram of database: -

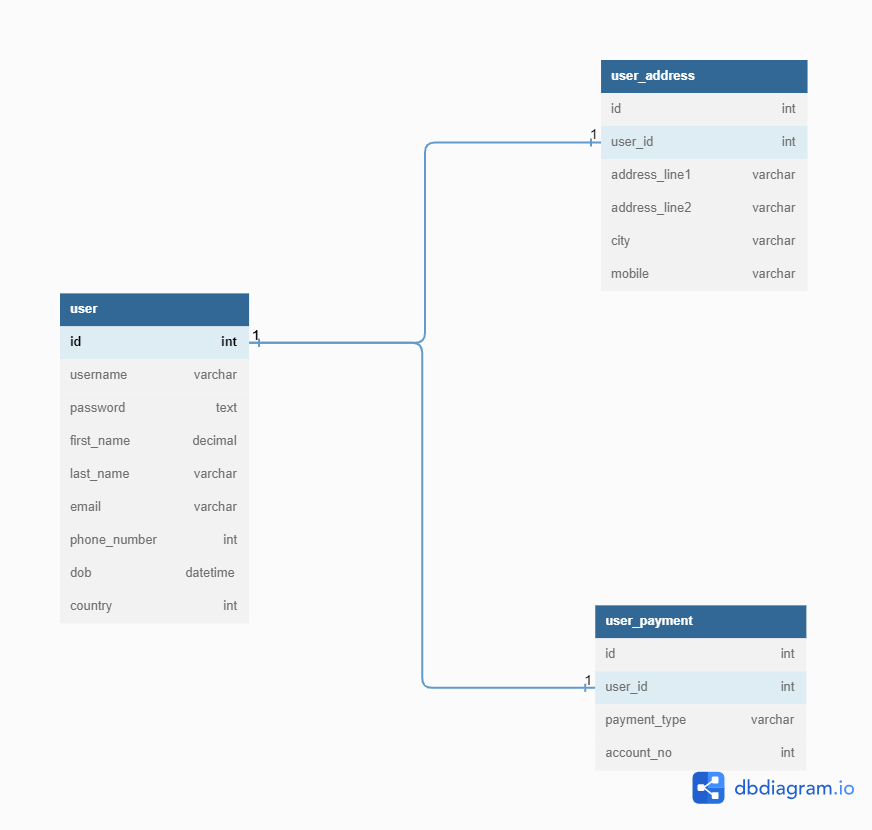
* The following table structure is an example of a database design that covers the e-commerce application platform.



* Here is link for this ER diagram: - <https://dbdiagram.io/d/649ea25202bd1c4a5e4c96ee>
* The table fields and indexes depending on the design of the overall platform.
* It contains three separate sections for user management, product management, and shopping process.

For Example: -

1.User management: -



I have created a user table that contains all the user details along with user\_payment and user\_address tables to store multiple addresses and payment details of users.

.

#### Product management: -

#### 

#### There are two other separate tables called discount, product\_inventory, and product\_category that are connected to it through database relationships.

#### Shopping process: -

#### 

#### The shopping process will guide a user to search the products, add the desired products to the shopping cart, and finally complete the transaction using a payment provider.

#### 2. Explain about searching performance. How will you handle replication in SQL for searching & Reporting: -

#### a. searching performance: -

#### Searching performance in an eCommerce application is crucial for providing a fast and efficient user experience. When users search for products or content within the application, they expect quick and accurate results. Optimizing searching performance involves several factors:

#### Indexing: - Indexing involves creating data structures that organize and store information in a way that allows for quick searching. it is typically applied to relevant fields such as product names, descriptions, categories, and attributes. By creating indexes on these fields improving search performance.

#### Caching: - Caching involves storing frequently accessed search results in memory or a separate caching layer. When a user performs a search query, the application first checks if the results are available in the cache. Caching is particularly effective for search queries that are repetitive or similar, improving overall search performance.

#### Scalability: - The search infrastructure is scalable helps maintain fast search performance as the application and user base expand.

#### b. replication for searching & Reporting: -

#### I. Read Replicas for Searching: - By setting up read replicas, this helps improve the speed and responsiveness of search queries, enhancing the overall searching experience for users.

#### II. Determine the Replication Strategy: - The two common approaches are master-slave replication and multi-master replication:

#### Master-Slave Replication: - This setup is suitable when there is a clear separation between write-intensive and read-intensive operations, such as in searching and reporting scenarios.

#### Multi-Master Replication: Allow multiple databases to accept both read and write operations. Changes made in one database are replicated to other databases. This approach is useful when searching and reporting involve frequent updates and changes.

#### III. Scaling: - Replication enables scaling and high availability for searching and reporting. As the application grows, add more replicas to handle increased traffic and ensure responsiveness.

#### Explain what major factors are taken into consideration for performance: -

#### In an e-commerce application, several major factors are taken into consideration for performance: -

#### Scalability: The application should be able to handle an increasing number of users, transactions, and data without significant degradation in performance. This requires a scalable architecture that can adapt to growing demands and efficiently utilize available resources.

#### Response Time: Users expect fast response times when interacting with an e-commerce application. The system should be optimized to minimize latency, ensuring that page’s load quickly, search results are delivered promptly, and transactions are processed without unnecessary delays.

#### Availability: High availability is crucial for an e-commerce application as any downtime can result in lost sales and customer dissatisfaction.

#### Security: E-commerce applications handle sensitive customer information such as personal data and payment details. Security measures like encryption, secure connections (HTTPS), and robust authentication mechanisms are essential to protect user data and prevent unauthorized access or data breaches.

#### Database Performance: The performance of the underlying database is crucial for an e-commerce application. Efficient query design, appropriate indexing, and proper data caching strategies should be employed to ensure fast and reliable retrieval of product information, user profiles, and transactional data.

#### Caching: Implementing caching mechanisms for frequently accessed data, such as product and static content, can significantly improve performance by reducing the load on backend systems.

#### Mobile Optimization: The application should be responsive, have a mobile-friendly user interface, and minimize data transfer to accommodate limited bandwidth and device capabilities.

#### Load Testing: Regular load testing and performance monitoring help identify performance, scalability issues, and areas for optimization. This enables proactive measures to be taken to enhance application performance, such as optimizing code, adjusting infrastructure resources, or employing caching strategies.

#### User Experience: Performance considerations go beyond technical aspects. The application's overall user experience, including intuitive navigation, fast checkout processes, and responsive.

1. Mention about Indexing, Normalization and Denormalization: -

**Indexing, normalization, and denormalization are techniques used to organize and optimize data storage, retrieval, and performance.**

* **Indexing: In the context of an e-commerce application, indexing can be used to improve the performance of search queries by creating indexes on frequently searched attributes such as product names, categories, or prices. Indexes speed up search operations by reducing the amount of data that needs to be scanned during query execution.**
* **Normalization: Normalization is a process used to improve data integrity in a database. Normalization follows a set of rules (known as normal forms) to organize data and minimize data duplication. In the context of an e-commerce application, normalization can be applied to ensure efficient storage of product data, user information, and transactional details.**
* **Denormalization: Denormalization is the process of selectively reintroducing redundancy into a normalized database to improve query performance. In the e-commerce application, denormalization can be used to create aggregated tables or add redundant data in certain scenarios, such as generating product listings or generating reports. Denormalization can speed up data retrieval but requires careful consideration to ensure data consistency, as redundant data increases the risk of anomalies during data modifications.**

1. How will you handle scaling, if required at any point of time: -

Handling scaling in an e-commerce application requires a proactive approach to ensure the system can handle increased traffic, transaction volumes, and data growth. Here's a step-by-step process for handling scaling when required:

* Set Scalability Goals: Define specific scalability goals and metrics based on expected growth patterns, user traffic, and transaction volumes
* Caching: Implementing caching mechanisms can significantly improve performance and scalability. Caching frequently accessed data, such as product catalogs, user profiles, and static content, reduces the load on backend systems and improves response times.
* Database Scaling: As the database plays a critical role in an e-commerce application, scaling the database is often necessary to handle increased data volumes and transactional demands. Explore techniques like sharing or partitioning to distribute the database across multiple servers or adopt scalable database solutions like NoSQL databases (e.g., MongoDB) or distributed SQL databases (e.g., Cockroach DB).
* Horizontal Scaling: One common approach is to horizontally scale the application by adding more servers or instances to the system. Horizontal scaling can be further enhanced by adopting technologies such as containerization (e.g., Docker) and orchestration frameworks (e.g., Kubernetes) to automate the deployment and management of application instances.
* Monitor and Adjust: Continuously monitor the application's performance, scalability, and user feedback. Regularly review performance metrics, user experience, and business growth to make necessary adjustments and optimizations to the scaling strategies.

1. Mention all the assumptions you are taking for solutions: -

* Database Technology: It is assumed that the e-commerce application utilizes a relational or NoSQL database to store and manage data. The solutions provide recommendations based on common database scaling techniques such as partitioning or adopting scalable database solutions.
* Load Testing: The solutions assume that load testing is performed regularly to assess the application's behaviour under high traffic and transaction volumes. This helps in identifying capacity limits and optimizing performance.
* Internet Connectivity: It is assumed that the e-commerce application operates in an environment with reliable internet connectivity to ensure seamless access for users and facilitate communication between different components of the application.
* Mobile Optimization: The solutions consider the need for mobile optimization and assume that the e-commerce application is designed to provide a responsive and mobile-friendly user experience.
* Modern Architecture: The solutions assume that the e-commerce application follows a modern architecture, such as a microservices-based architecture or a service-oriented architecture (SOA). This allows for independent scaling of different components and facilitates modular development and deployment.
* Data Caching: It is assumed that the e-commerce application employs data caching mechanisms using technologies like Redis or Memcached to improve performance and reduce the load on backend systems.
* Security Considerations: While scaling an e-commerce application, it is assumed that appropriate security measures are in place to protect user data, such as encryption, secure connections (HTTPS), and robust authentication mechanisms.