Database Design Assignment

Inventory Module

1. The seller can add products, images, and details of product ex: Price, discount, specification, Product SKUs etc.

2. At a time, there can be 100K users (active in application) searching for a product.

ER Diagram

A screenshot of a computer screen

Description automatically generated

**Order/Cart Module**

1. Buyers can add products to cart and buy them.

2. Seller can accept order or reject order. 3. Buyers/Sellers can check their order details based on order date.

ER Diagram

A screenshot of a computer screen

Description automatically generated

**Notification Module**

1. Buyer will get notification for product availability if he/she subscribed to same.

2. The buyer will get notification about product orders and order status when ready to ship, packed and delivered etc.

3. Application should support push notifications for promotional offers and sale discounts.

ER Diagram

A screenshot of a computer

Description automatically generated

**Authentication & Authorization**

1. Buyer and Seller should be able to login via email/password or Phone number with OTP.

2. Application should support Forget Password feature.

ER Diagram

A screenshot of a computer

Description automatically generated

Searching Performance:

There are many ways to improve the searching performance of a database, the following ways are possible :-

* **Indexes :** Indexes are a way of organizing data so that it can be searched more quickly. When you create an index, the database stores a copy of the data in a way that makes it easier to find. This can significantly improve the performance of queries that search for specific data. Ex: In the given assignment to improve the searching performance we can use SKU CODE index to improve search performance of the product table.
* **Caching :** Implementing a caching layer can significantly improve search Performance. It will help to avoid repetitive databases queries.
* **Optimize Search queries**: The way you write your queries can have a big impact on their performance. There are a number of things you can do to optimize your queries, such as using the correct operators, avoiding unnecessary joins, and using the correct data types.
* There are many more methods which can be applied based on the use case and scale and performance targets in mind and some of them are **Data partitioning, Elastic Search.**

**Major factors taken into consideration for performance:**

We can also use **Normalization** means breaking down data into smaller, related tables to eliminate data duplication. For example, in the notification module there could be a single table for notification of order and promo offers, but to remove duplicates preferred for separate tables to increase the performance of the application. For normalization there must not be a partial or transitive dependency.

Following are some of the ways in which we can optimize the performance: -

* Optimized query execution plans and tuning of SQL queries.
* Proper indexing of frequently queried columns.
* Effective caching mechanisms for frequently accessed data.
* Utilization of database connection pooling to manage connections efficiently.
* Appropriate hardware infrastructure and resource allocation.
* Load balancing techniques to distribute the load across multiple servers.
* Efficient database design and schema optimization.

**Normalization:**

The normalization process is divided into several normal forms. Each normal form represents a set of rules that define a specific level of normalization. The most used normal forms are:

* First Normal Form (1NF): This level requires that each attribute within a table should be atomic, meaning it cannot be further divided. Additionally, each attribute should contain only a single value, and there should be a primary key to uniquely identify each record.
* Second Normal Form (2NF): In addition to meeting the requirements of 1NF, this level addresses partial dependencies. It means that no non-key attribute should depend on only a part of the primary key. If such a dependency exists, the table should be split into two or more separate tables.
* Third Normal Form (3NF): 3NF builds upon 2NF and deals with transitive dependencies. It ensures that no non-key attribute depends on another non-key attribute. If such dependencies exist, they should be eliminated by creating separate tables.
* The entities and attributes have been normalized to avoid redundancy, such as separating ProductImage and SKU into separate tables.

**Handle Scaling:**

There are different ways to handle scale following are some:

* Horizontal Scaling
* Vertical Scaling
* Database Caching
* Load Balancing
* Database Replication

**Assumptions**

* The User entity includes attributes for email, phone, and password to support authentication and authorization. It also stores the last login information.
* The entities and attributes have been normalized to avoid redundancy, such as separating Product Image and SKU into separate tables.
* The Inventory entity tracks the available quantity of each product and is associated with the Product entity using a foreign key.
* The Notification entity stores notifications related to product availability, order status, and promotional offers. It has a foreign key reference to the User entity.
* The authentication process involves email/phone and password-based login, as well as OTP (One-Time Password) verification for phone-based login.
* The application provides a "Forgot Password" feature for users to reset their password
* Buyers and sellers are treated as separate entities with their own authentication information.
* Scaling and Normalization are also done to increase overall performance of the database.
* Each product can have multiple images, so there is a one-to-many relationship between Product and Product Image.
* Each product can have multiple SKUs, so there is a one-to-many relationship between Product and SKU.
* The assumption is made that an order can have multiple order items, which represent the individual products added to the cart.
* The order status is represented as a separate entity to track the various stages of order processing.
* Notifications are associated with buyers and can be of different types (e.g., product availability, order status, promotions).