**Visibility Problem**

In a system like Local Line where buyers of a specific region should only see the products of the

suppliers in the same cities where the suppliers deliver, a search functionality is needed to show

the correct products in a fast, reliable, accurate, and scalable way.

- Would you use a relational or a non-relational database architecture? Why?

Non-Relational database is most suitable i.e. MongoDB as it is most suitable for string unstructured data as well it has support for querying based on geo location. Even it has text searching capabilities so I will choose MongoDB storing all the information related to store and products.

- Explain what technologies would be needed to implement search functionality.

For fast searching I can use elastic search and using that I can implement search engine for querying product based on requested user region.

So, when user request for query for to list out the product I will get location of user and based on location I will reverse lookup in and find out the request is based on which region. Once I will get region then will do filter operation on database based on region and will return all the products belongs to the region.

**Multiple Logins per Business**

We’re planning to make a feature which allows each account to create multiple logins. Each

account has a business name property. Each login will be associated with a different role and

each role will have different permissions. The roles are flexible and can contain any set of

permissions and should be managed only by the root login(s) in a business.

What would be the best DB structure to store the hierarchy of the accounts in a RDBMS?

**Ans :** I will prefer to use Adjacency List to maintain the hierarchy and use Nested Sets to query the hierarchy.

**Product pricing, Categories and Discounts/Markups**

On the Local Line platform, every supplier should be able to show a different price list for every

customer (or group of customers) they may have. How would you structure the DB to implement

this functionality in a way that can serve tens of thousands of products per day?

Bonus: Some suppliers on Local Line resell and/or repackage other suppliers’ items. How would

you fit this into your model?

Ans : We crate the group then assign it to list of customers so based on that we can identify the customer belong to which group. and refer to that group ID to supplier so based on that group id he can show price list for that group of customer or individual customer.

We can use indexing and partition concept for optimization so can get optimizes way search result.

Suppliers refer to same model so it’s called self-referencing.

**MVC Framework :**

You have a route that needs to do the following (given a user\_id, buyer\_id, seller\_id, and

order\_id): For each process, would you put it in a model or a controller? Why

1. Check for user authentication check\_auth(user\_id)

Ans : I will do authentication in controller as controller responds user request by authenticating use. If user is not authenticated, then controller will not allow user to do any further action.

1. Query the database for the required order

Ans : I will do it in model using model manager by customizing model manager.

1. Calculate the grand total for that order calculate\_grandtotal(...)

Ans : I can do Calculate the grand total for that order in model or Controller.

If I will implement logic at model side then when query will be executed after that I will call calculate\_grandtotal() function on that query set result which will to operation on query set result and return result to called controller.

If I will implement logic at controller side in that case my logic will called on the return result from my model query set API.

1. Return the result to the user

Ans : I will do in controller side as controller response user request. So If user request the view detail of particular product then controller return the result to user by calling model query set.