From Poorna Shekar Reddy Puli

1. **DB Schema for an online store which has N Level sub-categories**

There are two ways to do this:

1. We can store reference Id of sub categories in each category,
    a. By storing in this way we will have to iterate through the subcategories which is going to increase the traffic on the database, by making too many calls from service.
2. Taxonomy: We can generate a string and use it to differentiate the categories and subcategories.
    a. Category code (XX.XX.XX.XX) (where each dot differentiates levels and X = 0,1..9,a,b,...y,z,A,B..Y,Z)
        i. By storing in this way we can fetch all the records of a product in a single query.

Preferred approach is the Second approach.

Models / Schemas:

Category Model:

__id: (Autogenerated Category Id)
Category_name: (String)
Category_code: (String)

Product Model:

__id: (Autogenerated Product Id)
Product Name: (String)
Category_code: (String)
Category_name: (String) (Not required in Products table)

Query to fetch categories in the first level:

```
select * from products where length(categorycode) - length(replace(categorycode, '.', '')) = 0 and categorycode REGEXP "^nn";
```

Query to fetch categories and items at all levels for a category:

```
select * from products where length(categorycode) - length(replace(categorycode, '.', '')) >= 0 and categorycode REGEXP "^nn";
```

2. REST API design change

Solution:
I will use a weblab method through which I will allow traffic to invoke new definitions.
Explanation:
Currently we have an API which customers are using with 3 fields. I will add another method with 4 fields and route a small amount of traffic to the new method by using a weblab method. Once the new method is working properly, we will confirm from metrics, we will increase the traffic to the new method.
Using DevOps:
Usually when we deploy it through AWS API Gateway or AWS Code Deployment we can route the traffic step wise. And the above weblab feature is available.