

NAME: UJJWAL KUMAR

DATE: 20TH MAY' 22

E-MAIL: ujjwal.kumar.official076@gmail.com

CONTACT NO.: 7634900918

Q1. We have an online store with items organized in categories, sub-categories, and sub-sub-categories up to N levels. Draw the database structure for the online store with the category tree to store things. Each category has a name. The number and level of categories can go up to infinite.

- Write a query to fetch all the first-level children (categories+items) for a category.
- Write a query to fetch all the children (categories+items) at all category levels for a category.

Table = "category_details"

category_id	category_name	parent_category_id	asset_id	asset_path	category_path
1	category_1	NULL	1	aapname/category_1/	0y1
2	category_2	NULL	2	aapname/category_2/	0y2
3	category_3	NULL	3	aapname/category_3/	0y3
4	category_4	2	3	aapname/category_2/category_4/	0y2y4y
5	category_5	2	3	aapname/category_2/category_5/	0y2y5y
6	category_6	4	23	aapname/category_2/category_5/category_6	0y2y4y6y
7	category_7	4	23	aapname/category_2/category_5/category_7	0y2y4y7y
NULL	NULL	NULL	NULL	NULL	NULL

Query 1: SELECT * FROM testdb.category_details
WHERE parent_category_id = 2 OR category_id = 2;

```
26
27 # First Level Category + First Level Children Category
28 • SELECT * FROM testdb.category_details
29 WHERE parent_category_id = 2 OR category_id = 2;
```

category_id	category_name	parent_category_id	asset_id	asset_path	category_path
2	category_2	NULL	2	aapname/category_2/	0y2
4	category_4	2	3	aapname/category_2/category_4/	0y2y4y
5	category_5	2	3	aapname/category_2/category_5/	0y2y5y
NULL	NULL	NULL	NULL	NULL	NULL

QUERY 2: # First Level Category + All Level Children

```
SELECT * FROM testdb.category_details
```

```
WHERE parent_category_id = 2 OR category_id = 2
```

```
UNION
```

```
SELECT * FROM testdb.category_details
```

```
WHERE parent_category_id IN (SELECT category_id FROM testdb.category_details  
WHERE parent_category_id = 2);
```

```
26 # First Level Category + All Level Children
27 • SELECT * FROM testdb.category_details
28 WHERE parent_category_id = 2 OR category_id = 2
29 UNION
30 SELECT * FROM testdb.category_details
31 WHERE parent_category_id IN (SELECT category_id FROM testdb.category_details WHERE parent_category_id = 2);
32
```

Result Grid						
Filter Rows: <input type="text"/>						
Export: Wrap Cell Content:						
	category_id	category_name	parent_category_id	asset_id	asset_path	category_path
▶	2	category_2	NULL	2	aapname/category_2/	0y2
	4	category_4	2	3	aapname/category_2/category_4/	0y2y4y
	5	category_5	2	3	aapname/category_2/category_5/	0y2y5y
	6	category_6	4	23	aapname/category_2/category_5/category_6	0y2y4y6y
	7	category_7	4	23	aapname/category_2/category_5/category_7	0y2y4y7y

Q2. Write the REST API endpoints for products consisting of (id, name, price) where the price is in dollars.

The products now consist of (id, name, price, currency).

Define how will you apply this change, considering that

a) API is in production.

b) other clients are using the existing API.

If you choose to version the API, write at a high level how the change will affect controllers, services, entities, etc.

Solution: -

Assumption: Let go with one specific REST API for products will take below request body info as POST request and if that product is available then it will be placed into current logged-in user cart.

API type: POST

URL: /rest_api_base/app-name/products/add-to-cart

REQUEST BODY: JSON format data

```
{
    "product_id": xyz,
    "product_name": abc,
    "product_price": 100$
}
```

RESPONSE BODY: JSON format data

```
{
    "added_to_cart_status": ture
}
```

NEW CHANGES REQUESTED: Add a new property to product "Currency".

NOTE: The API is live.

Changes needs to be done.

1. Important changes need to added in the Model Class for Product.

Product class contain the new field CURRENCY and idea is to make an **Enum** for CURRENCY which will contains all the supported currency format e.g. (US Dollar, INR, EURO, etc).

By Default, value will be the Dollar, means if we don't have any value provided from the request DTO then by default \$ will be filled.

```

Products.java ×
1 package com.ujjwal_Learning.Games;
2
3 import java.util.Date;
4
5 public class Products {
6
7     // Mapped field for column name
8     public static final String PRODUCT_ID = "product_id";
9     public static final String CATEGORY_ID = "category_id";
10    public static final String NAME = "product_name";
11    public static final String PRICE = "product_price";
12    public static final String CTIME = "creation_time";
13    public static final String CURRENCY = "currency";
14
15    private long productId;
16    private long categoryId;
17    private String name;
18    private Date creationTime;
19    private long price;
20    private String currency;
21
22    // getters and setters
23    public long getProductId() {
24        return productId;
25    }

```

Changes Needs to be done at different level i.e., at Controller, Service, Component, Repository.

1. Controller Level.

Before getting into Controller level, we need to add new parameter Currency to Request DTO. Then Simply pass the DTO to service layer.

2. At **Service Layer** do the validation on the input fields present in the Request DTO, i.e., if the field currency value is NULL or Empty, then by default we will insert the value as Dollar as earlier we are supporting this.

3. At **Component Layer** we will implement our calculation logic and the currency value conversion based upon the actual price provided and set the field values into the respective variables.

4. At **Repository Level** need to pass one more parameter to update the Product details in DB and then also check the availability based upon the delivery location and hence sending the valid response.

For the above task best way is to implement a **Stored Procedure** which will do this task for us. This will increase our query runtime and also increase our service response time.

Q3. Create the REST API crud (create, read, update, delete) operations using flask for Q2.

Added GitHub Repository: <https://github.com/the-wolverine-coder-UjjwalKumar/Products-Details>