

# Challenge Case Study

First, if you want access to the dev org to test, please email me and I grant you credentials.

All the detailed steps will be seen below:

## 1- Component to display tables with values.

The final output will be shown first and then I will explain how I did it.

These are 2 accounts that can have 1 or more contacts. The Contact will, according to fields Product\_\_c and Home\_Country\_\_c. These are picklist fields that can have values according to the given table.

The screenshot shows the Salesforce interface for the account "Burlington Textiles Corp of America". The left sidebar contains account details: Type (Customer - Direct), Phone ((336) 222-7000), Website (www.burlington.com), Account Owner (redacted), Account Site, and Industry (Apparel). The main content area displays a "Contact Datatable" with one contact: Jack Rogers (Identifier: 0038X00003b7...). Below the table is an "Activity" section with tabs for Chatter, Details, and News. It includes buttons for "New Task", "Log a Call", "New Event", and "Email". A filter bar shows "All time", "All activities", and "All types". The "Upcoming & Overdue" section indicates "No activities to show." The right sidebar shows "Contacts (1)" with details for Jack Rogers (Title: VP, Facilities; Email: jrogers@burlington.com; Phone: (336) 222-7000) and "Opportunities (1)" with details for Burlington Textiles Weaving Plant Gen... (Stage: Closed Lost; Amount: R\$ 235,000.00; Close Date: 26/05/2022).

and another example:

The screenshot shows the Salesforce interface for the account "Jiban Mercados". The left sidebar contains account details: Type, Phone, Website, Account Owner (redacted), Account Site, and Industry. The main content area displays a "Contact Datatable" with one contact: Joao Jiban (Identifier: 0038X00003aM...). Below the table is an "Activity" section with tabs for Chatter, Details, and News. It includes buttons for "New Task", "Log a Call", "New Event", and "Email". A filter bar shows "All time", "All activities", and "All types". The "Upcoming & Overdue" section indicates "No activities to show." The right sidebar shows "Contacts (1)" with details for Joao Jiban (Title: ; Email: ; Phone: (336) 222-7000), "Opportunities (0)", "Cases (0)", and "Notes & Attachments (0)".

How was done:

1 - created fields Product\_\_c and Home\_Country\_\_c. Both are picklists.

Product\_\_c

The screenshot shows the Salesforce Setup interface for the 'Contact' object. The left sidebar lists navigation options: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, and Object Limits. The main content area is titled 'Validation Rules' and 'Values'. Under 'Validation Rules', there is a 'New' button and a message 'No validation rules defined.' Below this, the 'Values' section has buttons for 'New', 'Reorder', 'Replace', 'Printable View', and 'Chart Colors'. It also includes 'Delete Selected', 'Deactivate Selected', and 'Replace Selected' buttons. A table lists the values for the 'Product\_\_c' field:

Action	Values	API Name	Default	Chart Colors	Modified By
<input type="checkbox"/>   Edit   Del   Deactivate	Standard	Standard	<input type="checkbox"/>	Assigned dynamically	[Redacted] 03/03/2023 18:22
<input type="checkbox"/>   Edit   Del   Deactivate	Black	Black	<input type="checkbox"/>	Assigned dynamically	[Redacted] 03/03/2023 18:22
<input type="checkbox"/>   Edit   Del   Deactivate	Metal	Metal	<input type="checkbox"/>	Assigned dynamically	[Redacted] 03/03/2023 18:22

At the bottom, there is an 'Inactive Values' section with a 'Delete Unused Values' button and a link to 'Inactive Values Help'.

Home\_Country\_\_c

The screenshot shows the Salesforce Setup interface for the 'Contact' object. The left sidebar lists navigation options: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, and Record Types. The main content area is titled 'Validation Rules' and 'Values'. Under 'Validation Rules', there is a 'New' button and a message 'No validation rules defined.' Below this, the 'Values' section has buttons for 'New', 'Reorder', 'Replace', and 'Printable View'. It also includes 'Delete Selected', 'Deactivate Selected', and 'Replace Selected' buttons. A table lists the values for the 'Home\_Country\_\_c' field:

Action	Values	API Name	Default	Modified By
<input type="checkbox"/>   Edit   Del   Deactivate	DE	DE	<input type="checkbox"/>	[Redacted] 03/03/2023 18:24
<input type="checkbox"/>   Edit   Del   Deactivate	FR	FR	<input type="checkbox"/>	[Redacted] 03/03/2023 18:24
<input type="checkbox"/>   Edit   Del   Deactivate	ES	ES	<input type="checkbox"/>	[Redacted] 03/03/2023 18:24
<input type="checkbox"/>   Edit   Del   Deactivate	IT	IT	<input type="checkbox"/>	[Redacted] 03/03/2023 18:24
<input type="checkbox"/>   Edit   Del   Deactivate	UK	UK	<input type="checkbox"/>	[Redacted] 03/03/2023 18:24

At the top of the main content area, there is a note: 'Active and inactive picklist values 5 (500 max)'.

## Metadata

I decided to create 3 metadata objects to store the table:


Cost per Calendar Month	DE	FR	ES	IT	UK
Standard	€ 0	€ 0	€ 0	€ 0	£ 0
Black	€ 9,90	€ 9,90	€ 9,90	€ 9,90	N/a
Metal	€ 16,90	€ 16,90	€ 16,90	€ 16,90	N/a

ATM Fee in other currencies	DE	FR	ES	IT	UK
Standard	1.7%	1.7%	1.7%	1.7%	1.7%
Black	Free	Free	Free	Free	Free
Metal	Free	Free	Free	Free	Free

Card Replacement Cost	DE	FR	ES	IT	UK
Standard	€ 6	€ 6	€ 6	€ 6	£ 6
Black	€ 6	€ 6	€ 6	€ 6	£ 6
Metal	€ 45	€ 45	€ 45	€ 45	£ 45

The reason to use metadata is that changing 1 value in one of its instances will automatically have different values when accessing an Account and having displayed the fees and values for each Contact. And 3 metadata objects is better cause you avoid having a single point of failure. To refresh the values, an Admin just need to access one of these 3 objects and replace values.

So, here is the result for the 3 metadatas:



**SETUP**  
**Custom Metadata Types**


View: [My View](#) [Edit](#) [Create New View](#)

New

Action	Label *	ATM Fee in Other Currencies Name	Type	Country	Value
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">DE_Black</a>	DE_Black	Black	DE	Free
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">DE_Metal</a>	DE_Metal	Metal	DE	Free
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">DE_Standard</a>	DE_Standard	Standard	DE	1.7%
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">ES_Black</a>	ES_Black	Black	ES	Free
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">ES_Metal</a>	ES_Metal	Metal	ES	Free
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">ES_Standard</a>	ES_Standard	Standard	ES	1.7%
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">FR_Black</a>	FR_Black	Black	FR	Free
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">FR_Metal</a>	FR_Metal	Metal	FR	Free
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">FR_Standard</a>	FR_Standard	Standard	FR	1.7%
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">IT_Black</a>	IT_Black	Black	IT	Free
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">IT_Metal</a>	IT_Metal	Metal	IT	Free
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">IT_Standard</a>	IT_Standard	Standard	IT	1.7%
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">UK_Black</a>	UK_Black	Black	UK	Free
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">UK_Metal</a>	UK_Metal	Metal	UK	Free
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">UK_Standard</a>	UK_Standard	Standard	UK	1.7%

Show me [fewer](#) ▲ records per list page

<div>  <div> <div>SETUP</div> <div>Custom Metadata Types</div> </div> </div>					
<div> <div>view:   My View ▾   Edit   Create New View</div> <div>New</div> </div>					
Action	Label	Card Replacement Cost Name	Type ▴	Country	Value
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">DE Metal</a>	DE_Metal	Metal	DE	€ 16,90
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">ES Metal</a>	ES_Metal	Metal	ES	€ 16,90
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">UK Metal</a>	UK_Metal	Metal	UK	N/a
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">FR Metal</a>	FR_Metal	Metal	FR	€ 16,90
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">IT Metal</a>	IT_Metal	Metal	IT	€ 16,90
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">UK Black</a>	UK_Black	Black	UK	N/a
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">DE Black</a>	DE_Black	Black	DE	€ 9,90
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">FR Black</a>	FR_Black	Black	FR	€ 9,90
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">IT Black</a>	IT_Black	Black	IT	€ 9,90
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">ES Standard</a>	ES_Standard	Standard	ES	€ 0
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">ES Black</a>	ES_Black	Standard	ES	€ 9,90
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">UK Standard</a>	UK_Standard	Standard	UK	£ 0
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">IT Standard</a>	IT_Standard	Standard	IT	€ 0
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">DE Standard</a>	DE_Standard	Standard	DE	€ 0
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">FR Standard</a>	FR_Standard	Standard	FR	€ 0
Show me <a href="#">fewer</a> ▲ records per list page					

<div>  <div> <div>SETUP</div> <div>Custom Metadata Types</div> </div> </div>					
<div> <div>view:   My View ▾   Edit   Create New View</div> <div>New</div> </div>					
Action	Label ▴	Cost Per Calendar Month Name	Type	Country	Value
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">DE Black</a>	DE_Black	Black	DE	€ 6
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">DE Metal</a>	DE_Metal	Metal	DE	€ 45
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">DE Standard</a>	DE_Standard	Standard	DE	€ 6
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">ES Black</a>	ES_Black	Black	ES	€ 6
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">ES Metal</a>	ES_Metal	Metal	ES	€ 45
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">ES Standard</a>	ES_Standard	Standard	ES	€ 6
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">FR Black</a>	FR_Black	Black	FR	€ 6
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">FR Metal</a>	FR_Metal	Metal	FR	€ 45
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">FR Standard</a>	FR_Standard	Standard	FR	€ 6
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">IT Black</a>	IT_Black	Black	IT	€ 6
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">IT Metal</a>	IT_Metal	Metal	IT	€ 45
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">IT Standard</a>	IT_Standard	Standard	IT	€ 6
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">UK Black</a>	UK_Black	Black	UK	£ 6
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">UK Metal</a>	UK_Metal	Metal	UK	£ 45
<a href="#">Edit</a>   <a href="#">Del</a>	<a href="#">UK Standard</a>	UK_Standard	Standard	UK	£ 6
Show me <a href="#">fewer</a> ▲ records per list page					

## APEX and LWC

To fetch an Account, its Contacts and these Metadatas, I created an Apex Class:

ContactFeeService:

Will receive the current Account Id, get its Contacts and for each of them, if fields Product\_\_c and Home\_Country\_\_c are populated, then the Account page will display the data for its Id, Name and respective values in the table.

And we are using an internal MetaContactDetailsObject wrapper class to group the values and send to the LWC component.

## Apex

ContactFeeService.cls

```

public with sharing class ContactFeeService {

    @AuraEnabled(cacheable=true)
    public static List<MetaContactDetailsObject> getContactCostsByAccount(String accountId){
        List<MetaContactDetailsObject> records = new List<MetaContactDetailsObject>();
        Contact contact = [SELECT Id, Name, Product__c, Home_Country__c FROM Contact WHERE AccountId = :accountId LIMIT 1];
        MetaContactDetailsObject record = new MetaContactDetailsObject();
        record.Id = contact.Id;
        record.name = contact.Name;
        record.calendar = getCalendarMonthValue(contact.Product__c, contact.Home_Country__c);
        record.fee = getFeeInOtherCurrenciesValue(contact.Product__c, contact.Home_Country__c);
        record.card = getCardReplacementValue(contact.Product__c, contact.Home_Country__c);
        records.add(record);
        return records;
    }

    //btw: querying against metadata without textArea fields won't bother to SOQL limits but only rows counting

    public static String getCalendarMonthValue(String product, String homeCountry){
        List<Cost_Per_Calendar_Month__mdt> records = [SELECT Id, Value__c FROM Cost_Per_Calendar_Month__mdt WHERE Type__c = :product AND Country__c = :homeCountry];
        return records[0].Value__c;
    }

    public static String getFeeInOtherCurrenciesValue(String product, String homeCountry){
        List<ATM_Fee_in_Other_Currencies__mdt> records = [SELECT Id, Value__c FROM ATM_Fee_in_Other_Currencies__mdt WHERE Type__c = :product AND Country__c = :homeCountry];
        return records[0].Value__c;
    }

    public static String getCardReplacementValue(String product, String homeCountry){
        List<Card_Replacement_Cost__mdt> records = [SELECT Id, Value__c FROM Card_Replacement_Cost__mdt WHERE Type__c = :product AND Country__c = :homeCountry];
        return records[0].Value__c;
    }

    public class MetaContactDetailsObject{
        @AuraEnabled
        public String Id;
        @AuraEnabled
        public String name;
        @AuraEnabled
        public String calendar;
        @AuraEnabled
        public String fee;
        @AuraEnabled
        public String card;
    }
}

```

## LWC

We will have to fetch the data and to display the following files:

contactFees.js

```

import { LightningElement, api, wire, track } from 'lwc';
import getContactCostsByAccount from '@salesforce/apex/ContactFeeService.getContactCostsByAccount';

export default class ContactFees extends LightningElement {

    @api recordId;
    error;
    contact;

    @track columns = [{
        label: 'Identifier',
        fieldName: 'Id',
        type: 'text',
        sortable: true
    },
    {
        label: 'Name',
        fieldName: 'name',
        type: 'text',
        sortable: true
    },
    {
        label: 'Cost per Calendar Month',
        fieldName: 'calendar',
        type: 'text',
        sortable: true
    },
    {
        label: 'ATM Fee in other currencies',
        fieldName: 'fee',
        type: 'text',
        sortable: true
    },
    {
        label: 'Card Replacement Cost',

```

```

        fieldName: 'card',
        type: 'text',
        sortable: true
    }
];

@wire(getContactCostsByAccount, {accountId: '$recordId'})
wireContact({error, data}){
    if(data){
        this.contact = data;
        this.error = undefined;
    }else{
        this.contact = undefined;
        this.error = error;
    }
}
}
}

```

contactFees.html

```

<template>
    <h2> Contact Datatable</h2>
    <template if:true={contact}>
        <lightning-datatable data={contact} columns={columns} key-field="Id">
        </lightning-datatable>
    </template>
    <template if:true={error}>
        {error}
    </template>
</template>

```

## PART 2

### Rest Controller

I have developed class ApexQueryAccountDetailsRestController as following.

It used the previous constructed class in part 1 and you need to pass the Account's Id on the URL.

ApexQueryAccountDetailsRestController .cls

```

@RestResource(urlMapping='/Account/*')
global with sharing class ApexQueryAccountDetailsRestController {

    @HttpGet
    global static String doGet(){
        RestRequest req = RestContext.request;
        RestResponse res = RestContext.response;
        String accountId = req.requestURI.substring(req.requestURI.lastIndexOf('/') + 1);
        List<ContactFeeService.MetaContactDetailsObject> result = ContactFeeService.getContactCostsByAccount(accountId);
        return JSON.serialize(result);
    }
}

```

The output will be a JSON:

```
[{"name":"Jack Rogers","Id":"0038X00003b77aLQAQ","fee":"1.7%","card":"€ 0","calendar":"€ 6"}]
```