**Unleashing the Power of Design Attributes in Lightning Datatable for LWC**

* Introduction  
   In the dynamic world of Lightning Web Components (LWC), creating robust and visually appealing user interfaces is crucial. One powerful tool in the developer's arsenal is the Lightning Datatable, a versatile component that allows you to display data in a tabular format. To take your UI to the next level, let's explore the use of design attributes in Lightning Datatable within LWC, enabling you to customize the look and feel of your tables seamlessly.
* Understanding Design Attributes  
   Design attributes are a set of properties that developers can leverage to control the appearance and behavior of Lightning Web Components. These attributes offer a high degree of flexibility, allowing developers/admins to fine-tune the user experience according to their specific requirements. When applied to Lightning Datatable, design attributes can transform a simple table into a polished and user-friendly interface.
* Getting Started  
  To add design attributes in datatable, we need to set targetConfig in Meta.xml file.  
    
  1. Create properties of type Boolean in targetConfig tag which will be used to show/hide the columns in datatable.  
  refer syntax given below:   
    
  Meta.XML

<?xml version="1.0"?>

<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">

    <apiVersion>57.0</apiVersion>

    <isExposed>true</isExposed>

    <targets>

        <target>lightning\_\_HomePage</target>

    </targets>

    <targetConfigs>

        <targetConfig targets="lightning\_\_HomePage">

            <property name = "ShowAccountOwner"  
 description="Click to Show account Owner column in Table"   
 label="Account Owner"   
 type="Boolean">  
 </property>

            <property name = "ShowRelatedContacts"   
 description="Click to add Related Contacts column in Table"   
 label="File Sizes"   
 type="Boolean">  
 </property>

        </targetConfig>

    </targetConfigs>

</LightningComponentBundle>

2. Create lightning datatable and fetch record in columns

HTML file

<template>  
    <lightning-card title="DATATABLE">  
        <lightning-datatable  
             key-field="id"  
            data={accList}  
             show-row-number-column  
             columns={columns}>  
        </lightning-datatable>  
    </lightning-card>  
</template>

3. For columns,we will mention only those columns which we want to show by default. In this case, we have Account Name,ID.  
 - Using @api decorator, add the targetConfig properties in JS initialize with false.  
 - We will check if the property is checked or not in connected callback since component is loaded on DOM in connected callback.  
 - If the property returns true, add the column in datatable columns using spread operator.  
JS File

import { LightningElement,wire,api } from 'lwc';  
import getAccounts from '@salesforce/apex/DatatableLwcController.getAccounts';

const  COLUMNS = [

        {label: 'Account Name', fieldName: 'Name' },

        { label: 'ID', fieldName: 'Id' }]

export default class DatatableLWC extends LightningElement {

    columns = COLUMNS;

    accList;

    @api ShowAccountOwner = false;

    @api ShowRelatedContacts = false;

@wire(getAccounts)

    accountData({ error, data }) {

      if (data) {

        let accParsedData=JSON.parse(JSON.stringify(data));

        accParsedData.forEach(acc => {

            if(acc.Id){

                acc.Account\_Owner = acc.CreatedBy.Name;

            }

        });

        this.accList = accParsedData;

      } else if (error) {

        console.error('Error:', error);

      }

    }

    connectedCallback() {

      if(this.ShowAccountOwner){

        this.columns = [...this.columns,{ label: 'Related Contacts', fieldName: 'No\_of\_Contacts\_\_c' }]

      }

      if(this.ShowRelatedContacts){

        this.columns = [...this.columns,{ label: 'Created By', fieldName: 'Account\_Owner' }]

      }

    }

}

APEX

public class DatatableLwcController {

    @AuraEnabled(cacheable=true)

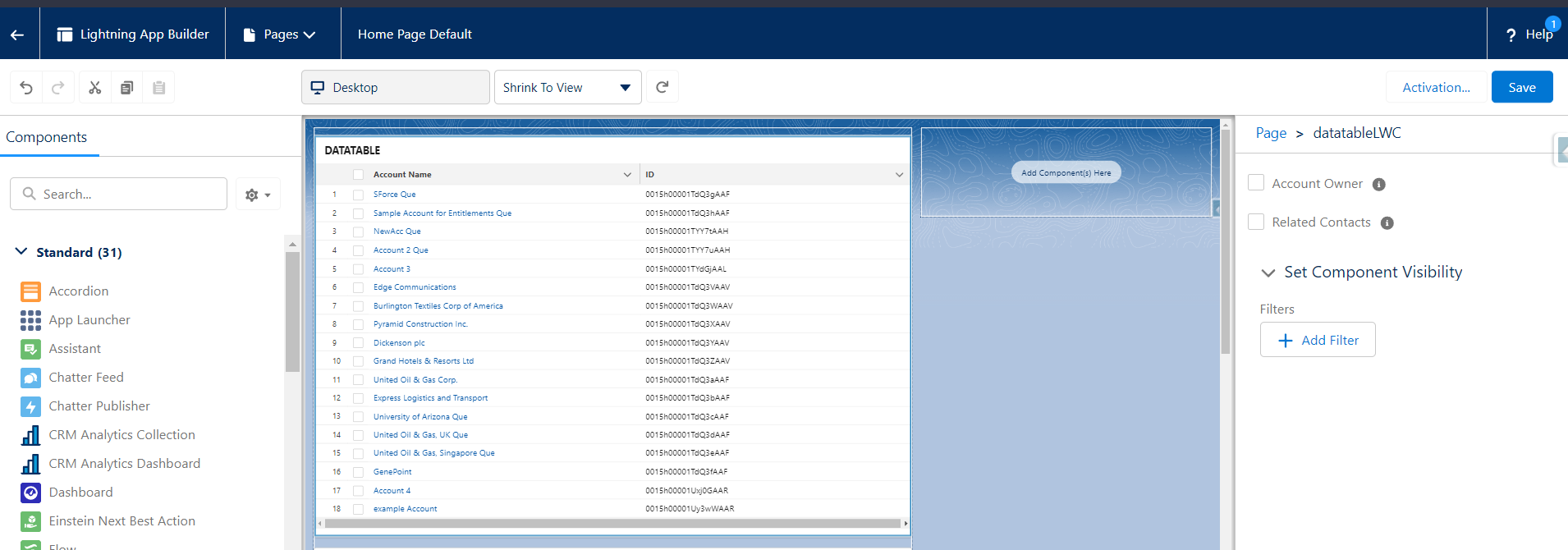
    public static List<Account>getAccounts(){

        List<Account> accList = [SELECT Id,Name,No\_of\_Contacts\_\_c, CreatedBy.Name FROM Account];

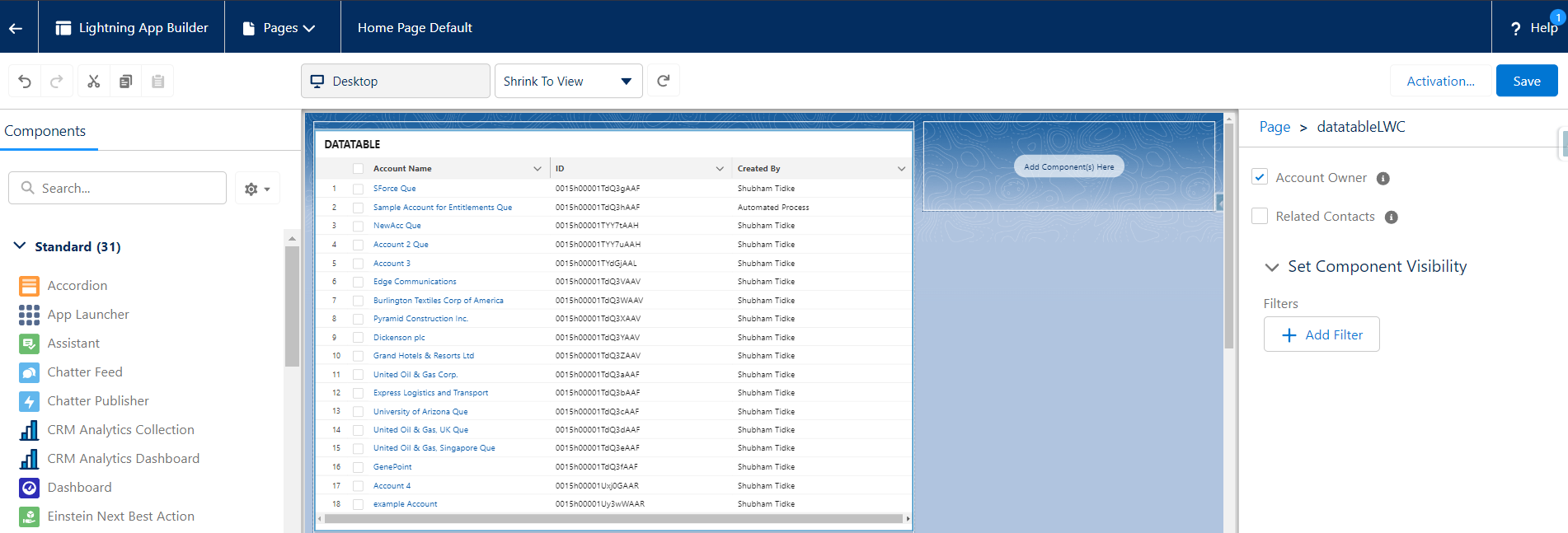
        return accList;

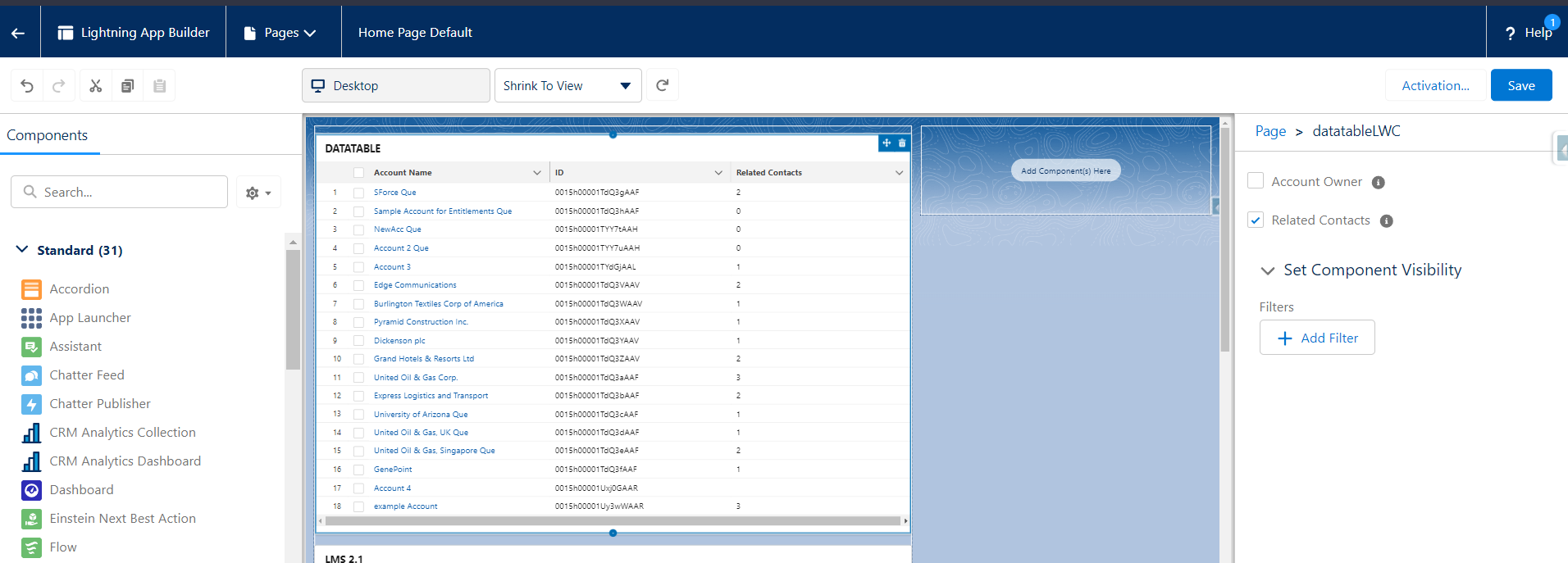
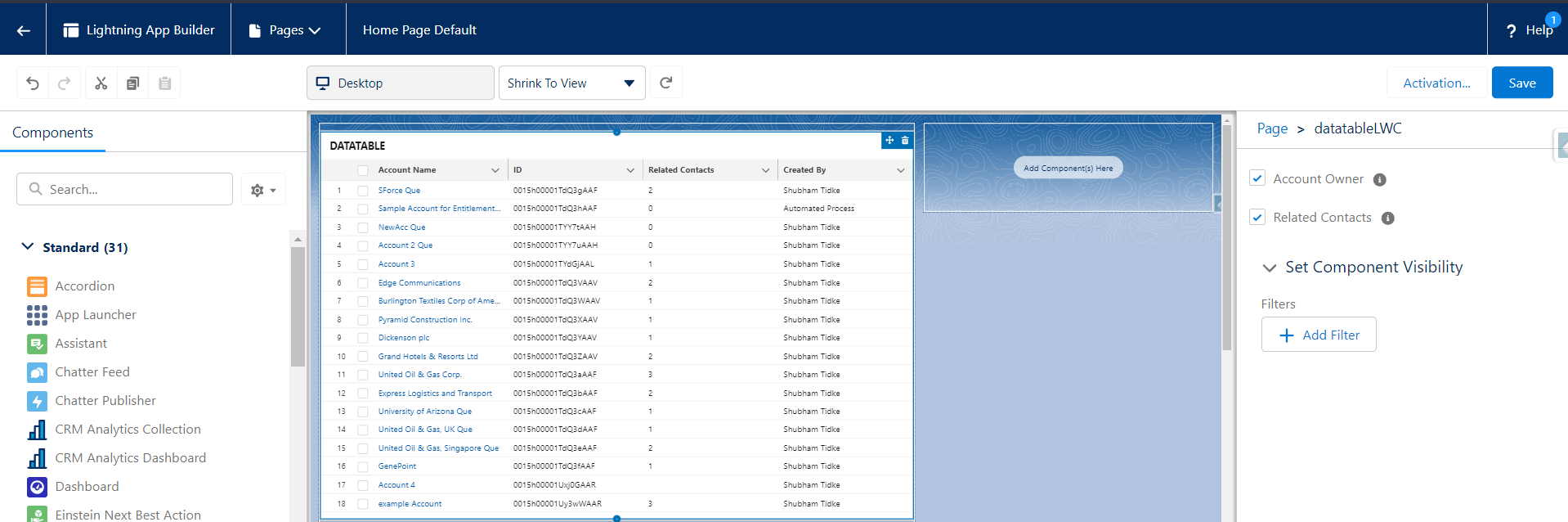
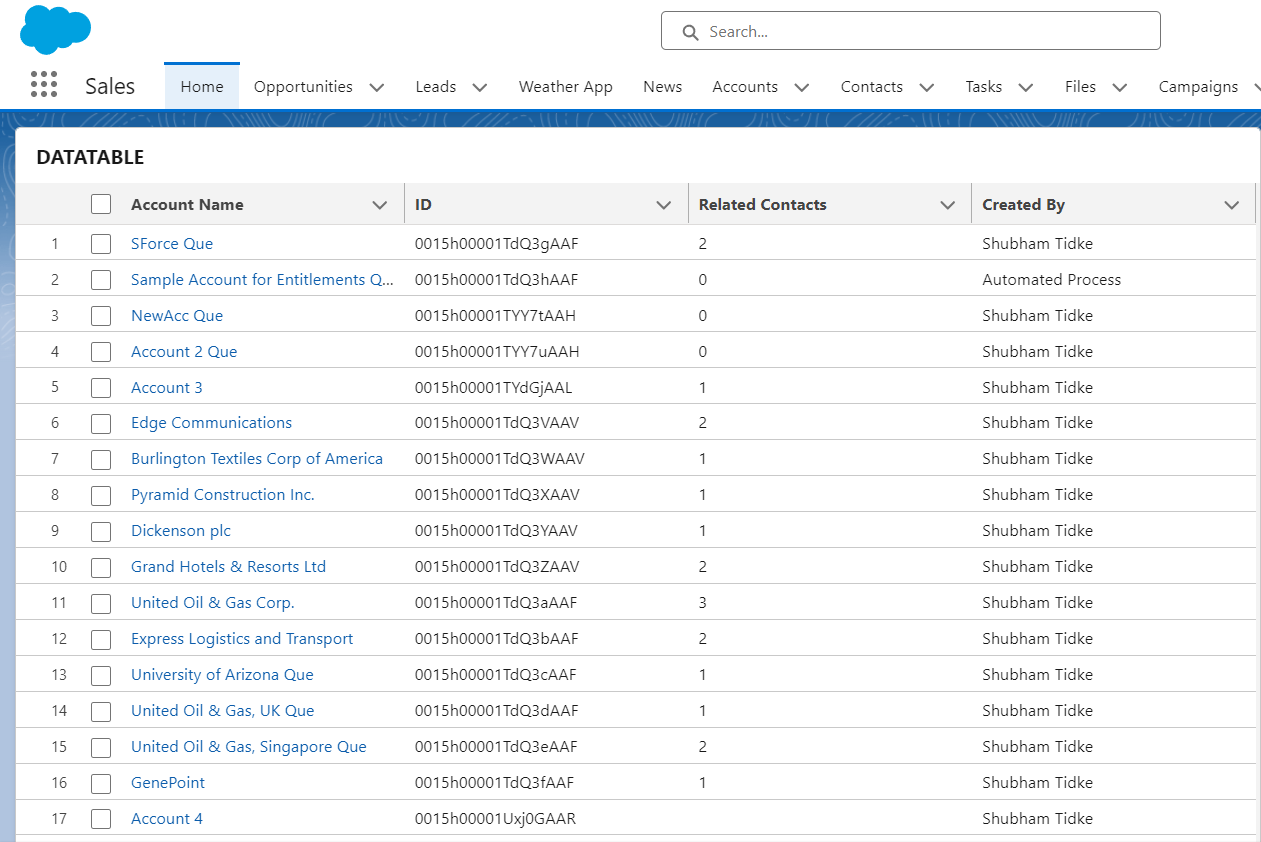
    }

}

4. Deploy the component and add it to home page,  
You will see checkboxes for the mentioned properties.  
  


5. Select the checkboxes for the columns you want to add in datatable,



  
  
  
  
Save the component and reload the page,  
  


* Conclusion  
  Harnessing the power of design attributes in Lightning Datatable for LWC opens up a world of possibilities for creating visually stunning and user-friendly interfaces. These attributes provide the fine-grained control you need to tailor your tables to perfection. Experiment with these attributes and elevate your LWC development to new heights.