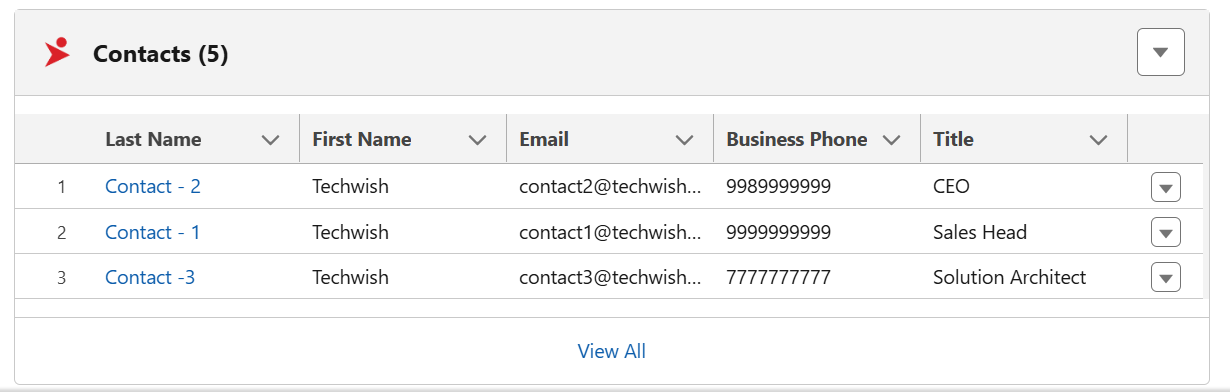
1. TW\_GetRelationshipApiNameClass
   1. methodTogetChildRelationShip



In the image above, when the user clicks the “View All” button, they will be navigated to the related list. Please refer to the code below.

**Navigation to Contact related list of account**

    navigateToContactRelatedList() {

        this[NavigationMixin.Navigate]({

            type: 'standard\_\_recordRelationshipPage',

            attributes: {

                recordId: this.recordId,

                objectApiName: 'Account',

                relationshipApiName: 'Contacts',

                actionName: 'view'

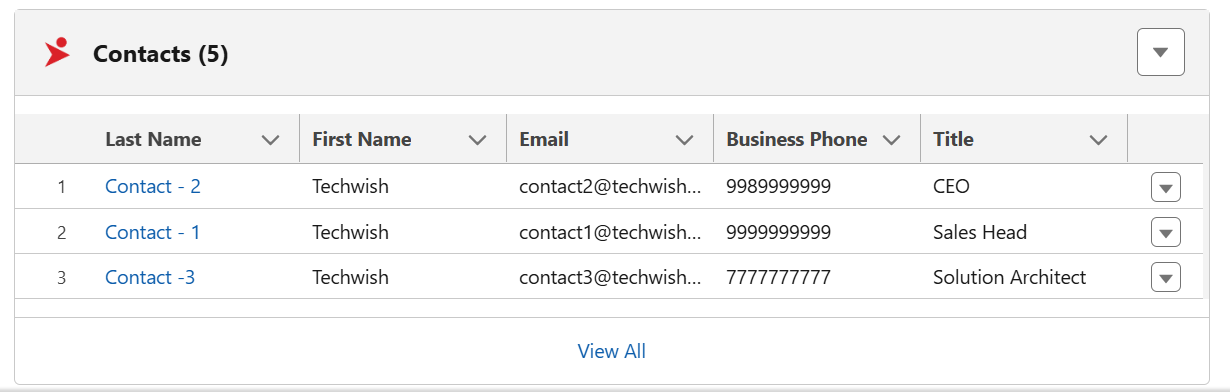
            },

        });

    }

Our Multi-Record Create and Edit component is designed as a generic solution. To enable it to dynamically retrieve relationship metadata, we developed a supporting Apex method, “methodToGetChildRelationship,” which fetches the relationshipApiName metadata.

* 1. getPluralNameofsObject



In the image above, we display the plural name of the child object, following the format used by standard related list components. To achieve this, we developed an Apex method that retrieves the plural name of the related list object from the backend, based on the related list name specified by the user during component configuration.

* 1. **sendObjectLabelForDeleteModal**

This Apex method retrieves the label of the related list object by passing the related list object's API name.

NB : We are not currently using the label of the child object, which refers to the related list object label.

--------------------------------------------------------------------------------------------------------------------------------------

1. TW\_DynamicQueryClass
   1. sendData

A screenshot of a email

Description automatically generated

* During page loading, our LWC component displays the related list records. To retrieve these records from the backend, we developed an Apex method.

A screenshot of a computer

Description automatically generated

In the two images above, we retrieve the values for child object API name, relationship field API name, and field API names from the component configuration. We then use these values to make a query to obtain the data. While showing the data we are handling user permission by using the “Security” class.

The **Security** class in Salesforce provides mechanisms to enforce field-level security and object-level permissions, ensuring that users can only access data they are authorized to view. In our class, we utilized this functionality to perform a security check on the retrieved records before returning them to the calling function.

* 1. sendLabels

During component configuration, the admin or developer provides the field API names. To display the corresponding field labels wherever needed within the component, we created this method to retrieve those labels.

A screenshot of a computer

Description automatically generated