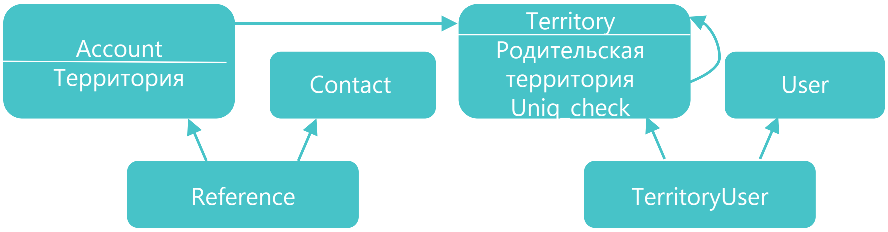
CUSTOMERTIMES

INDIVIDUAL PROJECT

Custom Territory Management Framework

Schema

Standard sObjects: User, Account, Contact
Custom sObjects: Territory, TerritoryUser, Reference
Account: A company
Contact: An organization contact person
Reference: A junction sObject to define many-to-many relationship – a person can work for multiple organizations
Territory: Self-referenced sObject to define a hierarchy of territories
User/Salesforce user: administrator or Sales Rep (Field Agent)
TerritoryUser: A junction sObject to define a user responsibility – on which territories a particular user work



Overview

The System Administrator is responsible to assign a particular user (Sales Rep) for a territory using TerritoryUser custom sObject record. An action automatically grants access for the user to a related information (Accounts and all related Contacts – referenced by the junction sObject). If the TerritoryUser junction record has been removed from the system – the access should be restricted to appropriate information.

Acceptance Criteria

1. As a System Administrator I should be able to assign a Sales Rep User for a particular territory (TerrUser sObject record).
2. As a Sales Rep User, I should be able to see and edit all Account and Contact records of the territory and it's children territories that I have been assigned for.
    a. When a user assigned to Ukraine, they should be able to see all Accounts and Contacts from Ukraine and related territories.
3. As a Sales Rep User, I should be able to see and edit relevant Accounts and Contacts if I have been reassigned from one territory to another
4. The system should not contain territory records with the same name on the same hierarchy and parent territory (there can't be two or more Kyiv cities in Ukraine, but Kyiv can be a village in one of regions of Ukraine)
5. As a minimum, the system should support a territory hierarchy for up to 3 levels. Nevertheless, it should provide enough scalability to be able to perform sharing actions for N – levels.

Setup

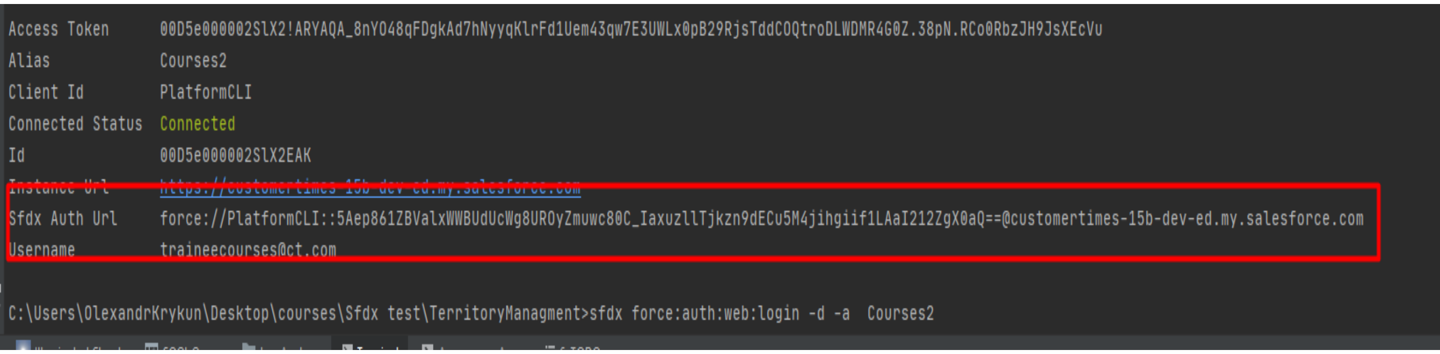
1. Create new clean dev org for this project.
2. Install package with data model https://login.salesforce.com/packaging/installPackage.apexp?p0=04t5w000000av9cAAA

3. Assign territoryManagment(shipped with package) permission set to sales rep user.

4. Clone standard user profile and name it **TerritoryUser.**

5. After project is completed send your 'login' 'password' and 'oauth url':

a. sfdx force:auth:web:login -d -a 'youralias';
b. sfdx force:org:display -u 'youralias' --verbose.



```
Access Token      00D5e000002SlX2!ARYAQA_8nYO48qFDgkAd7hNyyqKlrFd1Uem43qw7E3UWLx0pB29RjsTddCOQtroDLWDMR4G0Z.38pN.RCo0RbzJH9JsXEcVu
Alias             Courses2
Client Id         PlatformCLI
Connected Status  Connected
Id                00D5e000002SlX2EAK
Instance Url      https://customertimes-15b-dev-ed.my.salesforce.com
Sfdx Auth Url     force://PlatformCLI::5Aep861ZBValxWWBUdUcWg8UROyZmuwc80C_IaxuzllTjkzn9dECu5M4jihgiif1LAaI212ZgX0aQ==@customertimes-15b-dev-ed.my.salesforce.com
Username          traineecourses@ct.com

C:\Users\OlexandrKrykun\Desktop\courses\Sfdx test\TerritoryManagment>sfdx force:auth:web:login -d -a  Courses2
```

Notes

1. Use standard Salesforce UI – do not create custom components.
2. When create an architecture of a solution you should think about scalability of the system – how the sharing and re-sharing actions will work with thousands of territories, accounts, and contacts
3. Do not use the standard Contact – Account lookup relationship. Every contact must have a related Account using this reference – use a dummy account record for this.

4. **Following actions are forbidden to do with data model: Create new object, change field api name, change object api name**