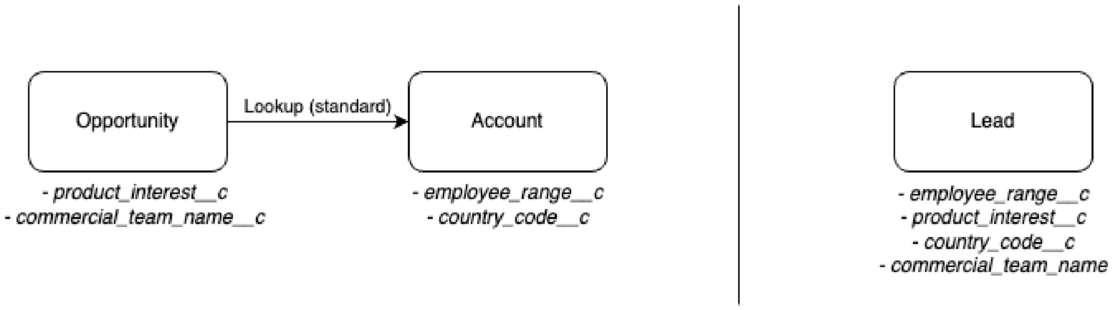
Salesforce Use Case - Swile

Base setup



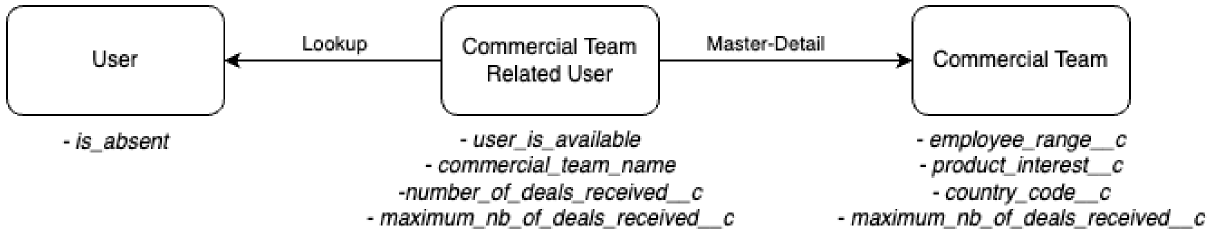
Lead :

- ***employee range, product interest & country code :*** fields to be able to identify which record should go to which team
- ***commercial_team_name :***
    - Formula field
    - Concatenation of the ***employee range, product interest & country code*** to generate a unique commercial_team_name

Opportunity :

- ***commercial_team_name :***
    - Formula field
    - Concatenation of the ***product interest, ACCOUNT employee range, & ACCOUNT country code*** to generate a unique commercial_team_name

Additional setup for user assignment



Commercial Team :

- ***employee range, product interest & country code :*** fields to be able to identify which record should go to which team
- ***maximum_nb_of_deals_received :***
  - Rollup summary field
  - → to retrieve the MAX number of deals associated to a user in this commercial team (will be useful when a user is no longer *absent*)

Commercial Team Related User :

This is a junction object between the Commercial Team and the Users, to allow one user to belong to several teams

- ***user_is_available :***
  - Formula field
  - Will check if the user is active and not absent : then user_is_available = true
- ***commercial_team_name :***
  - Formula field : Concatenation of the Commercial Team's product_interest + country_code + employee_range
  - This will generate a **unique** commercial team name that will be useful to assign the user when a new deal arrives
- ***number_of_deals_received :***
  - Number field, that will be incremented each time the associated user receives a new deal
- ***maximum_nb_of_deals_received :***
  - Formula field to retrieve the ***maximum_nb_of_deals_received*** field of the commercial team

User :

- ***is_absent :***
  - Checkbox field to specify if a user is absent
  - This field will be editable only for the users with a specific permission set : ***Manager***

Logic behavior

Lead & Opportunity Trigger :

- When a new lead / opportunity is created, a trigger is fired
    - This will retrieve the formula field : **commercial_team_name** of the triggering records
    - We can then query the ***Commercial Team Users*** related to these teams, we will only retrieve the users with is_user_available = true
    - We loop on each of them to retrieve the one with the ***least number_of_deals_received***
    - We ***assign*** the new lead / opportunity to this ***specified user***
    - We ***increment the number_of_deals_received*** of this user in this team
- This is the most efficient and optimal way I found to make the least DML statements & queries

'Reassign' button :

- For this need, I have created an LWC which contains a button to reassign
- This lwc will then check if the record to reassign is a **Lead** or an **Opportunity** and call a different apex functions based on the record
- The apex function will :
    - Retrieve an **available user** from the commercial team different than the current Owner
    - Assign him the deal and **increment** his **counter**
    - If no available user is found, it will return an exception to the LWC

UserTrigger :

- On this trigger, we first check if the user value ***is_absent*** was updated from 'true' to 'false'
- If it was then we call the apex function ***Users.updateUserNbOfDeals()***
    - This function will retrieve the field ***Maximum number of deal***s which is a ***roll-up summary*** of the **MAX** number_of_deals_received
    - It will then **update the current user value** to this MAX value

Reset the counter of each team member every month :

- I have created an apex Batch
- It will query all of the ***Commercial Team Users*** records and reset their ***number_of_deals_received*** to 0.
- This batch is scheduled to run every first day of the month