HANDS-ON ACTIVITIES

APPLICATION DESIGN

FEBRUARY, 2018

1. OPPORTUNITY LOCKING

Describe the use of the workflow capabilities (rules, tasks, emails, field updates, and approvals) within a solution.

USE CASE

The customer, Universal Containers, would like to lock all opportunity fields and the products related list after an opportunity is closed to ensure that opportunity data and reports are as accurate as possible.

DETAILED REQUIREMENTS

- Lock all opportunity fields immediately upon close (Stage is either "Closed Won" or "Closed Lost") for all users except the opportunity owner and the administrator.
- No products can be added, deleted, or updated after it is moved to closed and no opportunity line item fields (e.g., Quantity, Sales Price) can change either.
- The opportunity owner can change the Description field for seven days but cannot change any other field in that time period.
- After seven days, all fields become locked.
- Administrator profile users have no restrictions to editing.

PREREQUISITE SETUP STEPS

N/A

CONSIDERATIONS

Once an opportunity's stage is set to "Closed Won" or "Closed Lost," consider the following:

1. Can the stage be changed back to an open stage, which would allow fields to be edited?
2. Can the Close Date be changed after moving the stage to "Closed Won"?
3. Can products be deleted or edited?
4. Can the amount field on the product be changed?
5. Can the opportunity owner still edit the description field, but no other fields?

6. Can a person who does not own the opportunity and is not an administrator update the description field?
7. When a user adds a new opportunity, can the “Locked Opportunity” record type be selected?
8. When logging in as an Administrator, are there any restrictions?

SOLUTION

- Overall, a good solution is to use a “Locked” checkbox field that is set by a Workflow Rule & Field Update when an opportunity is moved to “Closed Won” or “Closed Lost.” Validation Rules are also used to prevent a user from saving a record after a user has updated opportunity fields or Opportunity Product records or fields.
- A combination of Record Type and Page Layout can also be used to enhance the user experience by not allowing a user to change a field in the first place, but this does not work with some standard fields (e.g., Name, Currency, Closed Date, Stage, Forecast Category) and related lists (e.g., Opportunity Products).

---

1. Why use a custom Close Date (Locked Date) instead of the standard Close Date?
    a. While the opportunity is open, the user can set the opportunity Close Date to any date. When setting the opportunity Stage to Closed Won, the Close Date gets set to the current date. When setting the Stage to Closed Lost, it is not changed to the current date, which will affect the 7-day requirement.

2. Why use an opportunity “Locked” field?
    a. The other option is to have validation rules for each field that is editable. By using a “Locked” field, only one validation rule is needed per object (Opportunity, Opportunity Product).

3. Can Workflow, Record Types, and Page Layouts alone solve the solution?
    a. No, because there are some standard fields on the opportunity record (e.g., Name, Currency, Closed Date, Stage, Forecast Category) that cannot be set to Read Only on a page layout.
    b. This will not work for the Opportunity Products related list.

4. What are some impacts to the User experience that should be considered?
   a. When using validation rules alone, a user will be able to update a field, and then they will get an error message after the record is saved.
   b. When using validation rules with Record Types and Page Layouts, some fields will not be updateable, which is ideal, but there will still be some fields that can be updated and prevented from change only after saving.

5. Why not just remove the “Add Product” and “Edit All” buttons from the Opportunity Product related list in the Locked Opportunity page layout?
   a. By removing these buttons, an administrator will not be able to update products after the record has been locked.

SOLUTION DESCRIPTION:

1. Custom fields: Opportunity Locked field (checkbox), Opportunity Locked Date field
2. Validation Rules: One for Opportunity and one for Opportunity Product; it should evaluate who the user is (System Admin vs. Owner vs. anyone else), the locked date and the field being updated
3. Workflow Rules: When Opportunity becomes Closed Won or Closed Lost, then trigger Field Updates
4. Workflow Field Updates: Locked checkbox gets checked, Opportunity Locked Date is set to present date (may not be the same as the actual Close Date if the close is done several days later) and Opportunity Record Type
5. Record types: One for standard, one for “Locked” set by a Workflow Field Update
6. Page layouts: One for standard, one for “Locked” (all fields marked as read-only where allowed)

DETAILED SOLUTION:

1. Add opportunity Record Type.

- Name = e.g., Opportunity Locked
- Do not Enable for Profiles (we do not want users to be able to select this RecordType when creating new Opportunities)

2. Add opportunity Field.

- Name = e.g., Locked

- Type = Checkbox
- Read Only

3. Add another opportunity Field.

- Name = e.g., Locked Date
- Type = Date
- Read Only

4. Add opportunity Page Layout.

- Copy Opportunity Layout
- Name = e.g., Opportunity Locked
- Make all fields Read-only:
  - ▫ Except for Description
  - ▫ Use validation rules for all other exceptions
- Associate Layout to the "Locked Opportunity" Record Type

5. Add Workflow Rule (Opportunity Object).

- Evaluation Criteria:
  - ▫ Evaluate the rule when a record is “created, and any time it’s edited to subsequently meet the criteria"
  - ▫ Note: if an administrator re-opens an Opportunity, we want the rule to be re-evaluated if it is set to Closed again
- Run this rule if the following "criteria are met"
  - ▫ Rule Criteria: Opportunity: StageEQUALSClosed Won,Closed Lost
- Workflow Actions (Opportunity Object)
  - ▫ Field To Update: Opportunity Record Type; field value "Opportunity Locked"
  - ▫ Field To Update: Locked; field value "true"
  - ▫ Field To Update: Locked Date; field value formula: Today()

6. Validation Rule (Opportunity).

- Rule Name = e.g., Opportunity_Locked
- Error Condition Formula (example):
  - ▫ /*1. If any field except Description is changed, then only Admin can update the field*/
    (NOT(ISCHANGED(Description)) && Locked__c = true && $Profile.Name <> "System Administrator") ||
    /*2. If Description is changed and Locked Date is less than 7, then only allow opportunity owner and Admin to change all fields*/

(ISCHANGED(Description) && (Today() - Locked_Date__c < 7) && Owner.Id <> $User.Id && $Profile.Name <> "System Administrator")||
/*3. If Description is changed and Locked Date is greater than 7, then only allow Admin to change the Description field*/
(ISCHANGED(Description) && (Today() - Locked_Date__c > 7) && $Profile.Name <> "System Administrator")

- Error Message = e.g., Record is locked when the opportunity is closed.

7. Validation Rule (Opportunity Products).

- Rule Name = e.g., Opportunity_Product_Locked
- Opportunity.Locked__c = true && $Profile.Name <> 'System Administrator'
    - Error Message = e.g., Record is locked when the opportunity is closed.

2. PROCESS AUTOMATION USING LIGHTNING PROCESS BUILDER

Describe the use of Lightning Process Builder in a solution that previously required code, along with the limitations.

USE CASE

Universal Containers has a sales operations team that monitors high-value deals and handles various follow-ups, including quoting follow-up, contract creation, internal notifications, scheduling "lessons learned" meetings, and creating follow-up tasks. As their business has grown exponentially, they have found it extremely challenging to keep up with their backlog and are in desperate need of a more automated approach.

DETAILED REQUIREMENTS

1. If a high-value deal becomes Closed and Won:

a. Unless the associated account is already a customer, update a checkbox to indicate the account is a customer and indicate the Year Started is the current year.
b. Create a draft contract record with the same name as the opportunity that's associated with the opportunity's account. It should start 30 days after the opportunity close date.
c. Congratulate the opportunity owner for closing and winning the opportunity by posting to the entire (company-wide) sales team.
d. Schedule a follow-up task ("Contract Follow-up") for the associated account's owner to be created 7 days after the opportunity's Close Date, due one week later.

2. If a high-value deal becomes Closed and Lost:

a. Set the Reason Lost to Price.
b. Notify the Sales Rep's manager's manager on the deal, including the opportunity name and amount that was lost. It's important that others with access to the Opportunity be able to comment on this as well.

3. If a Quote is given for a high-value deal:

   c. a. Create a follow-up task ("Quote Follow-up") due 10 days from the current day. It should be assigned to the opportunity owner.

ASSUMPTIONS

a. A high-value deal is considered to be an opportunity whose amount is >= $5,000,000.
b. Quote given means the Opportunity Stage = Proposal/Price Quote.
c. If the first condition (1) isn't met, go to the next one (2), and so on.
d. The actions above should only occur (once) when the appropriate conditions are met (and previously weren't met). However, there may be other automated workflows in place that could make the deal meet any of the requirements indicated above.

PREREQUISITE SETUP STEPS

1. Ensure there is a Sales Process assigned to several Opportunity Stage values, including "Proposal/Price Quote" (this should not be the default picklist value).
2. Create a custom Account field called IsCustomer with a default value of False.

CONSIDERATIONS

1. Will workflow rules potentially conflict with the Lightning process and, if so, how would you mitigate this?
2. What features are necessary to meet the requirements that a Lightning Process supports that Workflow Rules do not?
3. What if you or your customer is just more comfortable using Workflow Rules?
4. What is the difference between Criteria on an Action Group versus individual Action?
5. When do you use Immediate versus Scheduled Actions?
6. How can you monitor the execution of the Process?

SOLUTION

Using the Process Builder, we've combined three criteria nodes and associated actions into a single, automated process. To automate the same business process with workflow, you would have to create three different workflow rules and use Apex triggers to create the contract record and post to the Sales Chatter group.

---

1. Will workflow rules potentially conflict with the Lightning process and, if so, how would you mitigate this?
    a. If you create processes to replace any workflow rules, make sure you delete or deactivate those workflow rules when you activate the equivalent processes. Otherwise, both workflow rules and processes will fire and cause unexpected results, such as overwritten records or redundant email messages.
    b. Consider using Recursion in the Lightning process (Advanced option on the object). This will help to ensure that the process can execute up to 5 times within a single transaction (due to some other process, workflow rule, or flow updating the record in the same transaction).

2. What features are necessary to meet the requirements that a Lightning Process supports that Workflow Rules do not?
    a. Create your whole process in one place. Create a record, post to Chatter.

3. What if you or your customer are just more comfortable using Workflow Rules?
    a. Workflow Rules are supported for now, but will eventually be replaced by Lightning Processes (refer to last paragraph): [Lightning Process Builder]

4. What is the difference between criteria on an Action Group versus individual Action?
    a. The criteria on an Action Group determines when to fire all actions in the group, while an individual Action can have criteria only applicable to the single Action.

5. When do you use Immediate versus Scheduled Actions?
   a. Immediate Actions should be used any time they must be triggered as soon as its parent Action Group conditions are met.
   b. Scheduled Actions should be used any time the conditions are met for an Action Group but a Time Lapse (delay) is required until the desired Action is triggered.

6. How can you monitor the execution of the Process?
   Troubleshoot Processes

SOLUTION DESCRIPTION:

1. Create a Process using Lightning Process Builder on the Opportunity Object.
2. Add three Action Groups.
3. Add Update and Insert Actions to the Action Group as appropriate.
4. Set Actions as Immediate or Scheduled as appropriate.

DETAILED SOLUTION:

1. Create a Process using Lightning Process Builder on the Opportunity Object.

   a. Start the process when a record is created or edited.
   b. Select the Recursion option under Advanced.

2. Add an Action Group (Won).

   a. Criteria for Executing Actions: Conditions are met
   b. Condition #1
      - Field: [Opportunity].StageName
      - Operator: Equals
      - Type: Picklist
      - Value: Closed Won

   c. Condition #2
      - Field: [Opportunity].Amount
      - Operator: Greater than or equal
      - Value: $5,000,000.00

   d. Conditions: All of the conditions are met (AND)
   e. Do you want to execute the actions only when the specified changes are made to the record? Yes

3. Add an Action Group (Lost).

a. Criteria for Executing Actions: Conditions are met
b. Condition #1
    - Field: [Opportunity].StageName
    - Operator: Equals
    - Type: Picklist
    - Value: Closed Lost
c. Condition #2
    - Field: [Opportunity].Amount
    - Operator: Greater than or equal
    - Type: Currency
    - Value: $5,000,000.00
d. Conditions: All of the conditions are met (AND)
e. Do you want to execute the actions only when the specified changes are made to the record? Yes

4. Add an Action Group (Quote Given).

a. Criteria for Executing Actions: Conditions are met
b. Condition #1
    - Field: [Opportunity].StageName
    - Operator: Equals
    - Type: Picklist
    - Value: Proposal/Quote
c. Conditions: All of the conditions are met (AND)
d. Do you want to execute the actions only when the specified changes are made to the record? Yes

5. Add Actions to the Won Action Group.

1. Immediate Actions

- Action #1 (Update Account to Customer)
    - Action Type: Update a Record
    - Action Name: Update Account to Customer
    - Record Type: Select the Opportunity record that started your process
    - Criteria for Updating Records: Updated records meet all conditions
    - Condition #1
        - Field: IsCustomer
        - Operator: Equals
        - Type: Boolean

    - Value: False
  - New Field Value #1
    - Field: IsCustomer
    - Type: Boolean
    - Value: True
  - New Field Value #2
    - Field: Year Started
    - Type: Formula
    - Value: TEXT(YEAR(TODAY()))

- Action #2 (Insert Draft Contract)
  - Action Type: Create a Record
  - Action Name: Insert Draft Contract
  - Record Type: Contract
  - New Field Value #1
    - Field: Account ID
    - Type: Reference
    - Value: [Opportunity].AccountId
  - New Field Value #2
    - Field: Description
    - Type: Reference
    - Value: [Opportunity].Name
  - New Field Value #3
    - Field: Contract Start Date
    - Type: Formula
    - Value: [Opportunity].CloseDate + 30
  - New Field Value #4
    - Field: Status
    - Type: Picklist
    - Value: Picklist

- Action #3 (Post to Sales group)
  - Action Type: Post to Chatter
  - Action Name: Post to Sales Group
  - Post to: Chatter Group
  - Group: Sales Strategy (or the relevant group name)
  - Message: Congratulations {![Opportunity].OwnerId} for closing and winning the following opportunity: {![Opportunity].Name}!!!

2. Scheduled Actions

- Set Time for Actions to Execute: 7 Days After [Opportunity].CloseDate
- Action #1 (Create Follow-up Task)
  - Action Type: Quick Actions

- Action Name: Create Follow-up Task
- Filter Search By: Global actions
- Type: Create a Record
- Action: NewTask
- New Quick Action Field Value #1
  - Field: Assigned To ID
  - Type: Reference
  - Value: [Opportunity].OwnerId
- New Quick Action Field Value #2
  - Field: Due Date Only
  - Type: Formula
  - Value: [Opportunity].CloseDate + 14
- New Quick Action Field Value #3
  - Field: Subject
  - Type: String
  - Value: Contract Follow-up

6. Add Actions to the Lost Action Group.

1. Immediate Actions

- Action #1 **(Add Reason Lost)**
  - Action Type: Update a Record
  - Action Name: Add Reason Lost
  - Record Type: Select the Opportunity record that started your process
  - Criteria for Updating Records: No criteria-just update the records!
  - New Field Value #1
    - Field: Reason_Lost__c
    - Type: Picklist
    - Value: Price
- Action #2 **(Post to Sales group)**
  - Action Type: Post to Chatter
  - Action Name: Post to This Record
  - Message: {![Opportunity].Owner.ManagerId} The following opportunity has been lost:

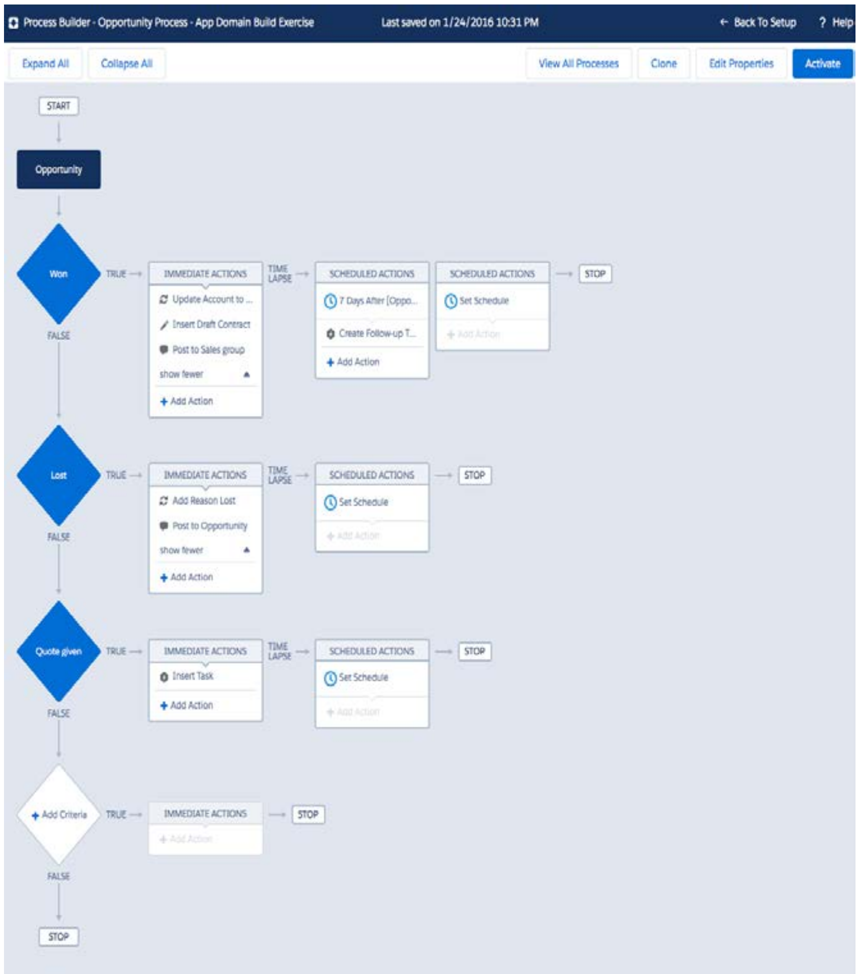
    Name: {![Opportunity].Name}
    Amount: {![Opportunity].Amount}

7. Add Actions to the Quote Given Action Group.

1. Immediate Actions

- Action #1 **(Insert Task)**
    - Action Type: Quick Actions
    - Action Name: Insert Task
    - Filter Search By: Global actions
    - Type: Create a Record
    - Action: NewTask
    - New Quick Action Field Value #1
        - Field: Assigned To ID
        - Type: Reference
        - Value: [Opportunity].OwnerId
    - New Quick Action Field Value #2
        - Field: Subject
        - Type: String
        - Value: Quote Follow-up
    - New Quick Action Field Value #3
        - Field: Due Date Only
        - Type: Formula
        - Value: TODAY() + 10

Solution Process (Visual):



3. CUSTOM LINKS

Utilize advanced formula features (VLOOKUP, rollup summary, image, x-object) and describe capabilities and limitations.

USE CASE

Universal Containers would like to easily be able to see a summary of opportunities from any account record for all accounts in an account hierarchy. Build a solution to meet these requirements.

DETAILED REQUIREMENTS

- When user is on Account Details page, provide a new link.
- This link should be called Opportunities Summary.
- Clicking on this link should show users a summary of opportunities.
- Summary should be for all accounts in that account hierarchy.
- This includes all accounts with the same parent, plus the parent itself.
- User should be able to pull this summary from any account record.

PREREQUISITE SETUP STEPS

N/A

CONSIDERATIONS

Once the user is on the Account details consider the following:

1 Can this be a button or a custom link?
2 When a user clicks, does the solution work dynamically based on the record they are on?
3 Does it work from child records as well as parent records?
4 Will this perform well and be able to be maintained?
5 Is it possible to avoid a lot of customizations?

	Notes	Limitations / Caveats	Recommendations / Alternatives
Design/ Build	**Assumption:** Account depth will not exceed 5	**Limitations:** Hard-coded depth of account hierarchy traversal (currently supports 5 levels)	Use Apex and Recursion
Ops	Consider security settings for access to Report Folders	▪ Initial Amount may be entered in error, and would be stored by Workflow – would need to manually edit stored value to override ▪ Workflow Rule Field Updates occur after Validation Rules are consistent with Field Updates ▪ Deals created prior to Workflow creation will not have an initial amount set	Workflow Rules can be bypassed by Bulk Import

SOLUTION

Overall, a good solution is to use configuration only options.

A combination of Custom Formula Field and Report can be used to enhance the user experience by allowing quick access from record detail standard page layout.

---

1. While Account is open, use a button versus a link to call a report with Account as parameter?
2. Add the button on the Opportunity related list in the page layout?
3. Replace current standard page layout with a VF page to be able calculate and render Opportunities Summary inline?

SOLUTION DESCRIPTION

1. Custom fields: Opportunity Summary field (link)
2. Page layouts: standard page layout
3. Report

Account formula field: You need to build a field on the account that is populated either with a workflow or potentially a formula (but including this field in other reports might cause the formula to be slow.) This field would calculate the top level parent (e.g., Account Root ID, Name). It would be populated with the parent Account name and the ID (if populated) or if not populated, it would contain the account information itself.

You would use this field in the report criteria and pass it into the report. The link should be on the account record.

NOTE:

Note:
It won't work if you use an opportunity report with two blank criteria, for example, account name =, parent account = "OR'd" together and you pass the account name and the parent account name into the report. The reason is that for any account with the parent account blank, the report will return all opportunities for accounts whose parent account is blank. This isn't correct, since that could include completely unrelated opportunities where the parent account is blank.

Other considerations: Place a limit on the number of levels possible in a formula. You also should include the report link format, for example, with the pv0={your formula field) and explain how to create the report (accounts with oppty report), get the ID (also note that this ID will change when pushed to production), and lastly that report links currently don't work with the Lightning experience and format the report link.

DETAILED SOLUTION

1. Add custom formula field(s) to the Account for Root level parent Name and ID.
2. Add custom formula fields to the Account page layout.
3. Create / clone Opportunities summary report.
4. Use Account parameter passed from the custom formula link as filter in the report.

FORMULA DETAILS

Here are the details of the two custom fields needed on Account: Account Root ID and Account Root (e.g., name):

1. Calculate Root Account ID:

```
IF(
  LEN(Parent.Name) < 1, Id,
  IF(
        LEN(Parent.Parent.Name) < 1, Parent.Id,
        IF(
              LEN(Parent.Parent.Parent.Name) < 1, Parent.Parent.Id,
              IF(
                    LEN(Parent.Parent.Parent.Parent.Name) < 1,
  Parent.Parent.Parent.Id,
                    IF(

        LEN(Parent.Parent.Parent.Parent.Parent.Name < 1,
  Parent.Parent.Parent.Parent.Id,
                                NULL
                          )
                    )
              )
        )
)
```

2. This calculates the Root Account name (For display), and renders it as a hyperlink to a parameterized report, passing the root account id as the parameter:

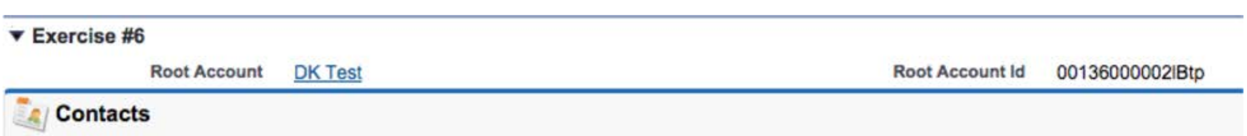
SAMPLE CODE:

```
IF(
LEN(Parent.Name) < 1, HYPERLINK("/00O36000000kNsj?pv0=" & Id,
  Name,"_parent"),
IF(
LEN(Parent.Parent.Name) <1, HYPERLINK("/00O36000000kNsj?pv0=" &
  Parent.Id,Parent.Name,"_parent"),
IF(
LEN(Parent.Parent.Parent.Name) < 1, HYPERLINK("/00O36000000kNsj?pv0=" &
  Parent.Parent.Id,Parent.Parent.Name,"_parent"),
IF(
LEN(Parent.Parent.Parent.Parent.Name) < 1,
  HYPERLINK("/00O36000000kNsj?pv0="
  &Parent.Parent.Parent.Id,Parent.Parent.Parent.Name,"_parent"),
IF(
 LEN(Parent.Parent.Parent.Parent.Parent.Name) < 1,
  HYPERLINK("/00O36000000kNsj?pv0=" & Parent.Parent.Parent.Parent.Id,
  Parent.Parent.Parent.Parent.Name,"_parent"),
 "Parent beyond 5 levels"
 )
 )
 )
 )
 )
```

NOTE:

NOTE:
You need to replace "00O36000000kNsj" with right Report ID specific to your org / sandboxes. Best practice would be to not hard code such IDs but that's beyond the scope of this exercise.

Here is the output expected including the link to the report with parameter:



ADDITIONAL ACTIVITIES

CROSS OBJECT FORMULA FIELD

Currently account status is displayed on the account with a custom field that indicates a status. In order to bring visibility to the account's status to the sales person working an opportunity, it's necessary to show the account status on the Opportunity field.

SOLUTION

1. Create a custom pick list field "Account Status" on the Account object with following values: Active, At Risk, Renewal Negotiations.
2. Load three images: one for green, yellow, and green traffic light.
3. Add a text formula field on Opportunity – "Account Status" to return the value of the account status in a green, yellow, or red traffic signal image depending for account status " as shown below:

```
1 IF( TEXT(Account.Status__c) = 'Active',
  IMAGE("/servlet/servlet.FileDownload?file=01570000000Q6El", "Green"), IF (
  TEXT(Account.Status__c) = 'Renewal Negotiations',
  IMAGE("/servlet/servlet.FileDownload?file=01570000000Q6Ef", "Yellow"),
  IMAGE("/servlet/servlet.FileDownload?file=01570000000Q6Ep", "Red")))
```

4. Add the new formula field to the Opportunity page layout.

ROLL-UP SUMMARY FIELDS

The account executive would like to display the sum of all open opportunity values for the current calendar year on the Account page.

SOLUTION

1. Create a roll-up summary custom field on the Account object called “Current Year Open Opportunities.”
2. Set the field to summarize “Opportunities” and choose “Sum” for Roll-up type and use “Amount” as the field to aggregate. Use criteria of “Stage” not equal to “Close Won” and not equal to “Close Lost,” Close Date equal to or greater than 1/1/2016 and Close Date less than 1/1/2017 to only return sum of open opportunity amounts for the calendar year.

This is a paragraph of placeholder text. It is only here to help show the layout of the page and how the text will flow. Replace this placeholder text with your own meaningful content.

USE OF A CUSTOM SETTING IN A FORMULA FIELD

Sales users would like to view their annual sales quota on the Opportunity page and what percentage of the quota will be attained by each opportunity. The sales quota changes on the user's position in the organizational hierarchy and should be able to be set for each individual sales user.

SOLUTION

1. Add a custom setting "Sales Quota" of type Hierarchy and with "Public" visibility.
2. Add a field "Annual Sales Quota" of type currency to this setting.
3. Click on manage and set the organization value of 0 to be the default value.
4. Set higher values for specific users depending upon the organizational structure.
5. Create a custom formula field of type "percent" on opportunity to display the percent of annual sales quota attained by each opportunity for each sales user:

    1 If ($Setup.Sales_Quota__c.Annual_Sales_Quota__c > 0, Amount /$Setup.Sales_Quota__c.Annual_Sales_Quota__c,0)

RELATED LISTS LOADING SETTINGS

Some of the users are complaining that the Account and Opportunity pages take a relatively long time to load especially when there are a lot of items related to the Account or Opportunity records. The team would like to investigate ways to improve the performance.

SOLUTION

1. Open an Account record with 30 Contacts, 20 Opportunities, and 100 Activity History related records and notice lag in opening up the record.
2. Update the Related Lists Loading Settings by following the menu path: **Setup > App Setup > Customize > User Interface** and click Enable Separate Loading of Related Lists of External Objects.
3. Open the same Account record before the change to the loading of related lists. You should notice performance improvement after settings are updated.

HYPERLINK FORMULA FIELD

The sales users would like to have a URL on the Opportunity page to an on-premise web application. The URL should pass the account name and the user's first and last name. The page layout shouldn't display the URL on screen, but instead should display the text “Account Report” for the URL.

SOLUTION

1. Create a custom formula field to return the hyperlink:
   Hyperlink('http://acme.com/accountreport.jsp?account='+ Name + '&FirstName='+ $User.FirstName + '&LastName=' + $User.LastName, 'View Account Report')

2. Add the new formula field to the Opportunity page layout.

ADDITIONAL ACTIVITIES

VISUAL WORKFLOW

1. Before you dive into this activity, make sure that you have:

- Signed up for a Developer Edition organization
- Installed the enhanced Warehouse data model.

This exercise will give you hands-on training for creating a basic Visual Workflow.

Step 1: Add a screen element to capture user's region.

Step 2: Add and configure a Decision Element.

Step 3: Add a screen element for the Americas option.

Step 4: Add a screen element for the EMEA option.

Step 5: Collect commute information.

Step 6: Collect email address.

Step 7: Thank the respondents.

Step 8: Save the flow.

Step 9: Run the flow.

For reference, please refer to the Trailhead Projects that correspond to flow exercises:

Flow Tutorial	Trailhead Project
Measure Customer Satisfaction	Create a Satisfaction Survey
Calculate Discounts on Opportunities	Build a Discount Calculator
Reassign Accounts	Build an Account Reassignment Wizard

CREATING A SUGGESTION BOX APP

In this Trailhead project, you learn how to build an application on App Cloud from start to finish without code.