atrium + tableau

Tableau and CRM Analytics

Choosing the Tool That's Right for Your Organization and Use Case

The Evolution of Tableau Over the Years

When Salesforce acquired Tableau years ago, the speculation began as to how that product might complement (or compete with) Salesforce's existing analytics tool, Einstein Analytics. Fast forward some years, throw in some name changes for Einstein Analytics (now CRM Analytics) and a somewhat shifting roadmap for the total Salesforce analytics suite of products, and we are finding many customers still have questions when it comes to which tool is right for them.

It should be noted that searching for a one-size-fits-all tool is generally not a recommended approach. The truth is often that you will frequently need multiple tools, and with Tableau and CRM Analytics being among the best-in-class available options, the odds are your organization may leverage both. So, on a use-case-by-use-case basis, how do you decide which is the best fit?

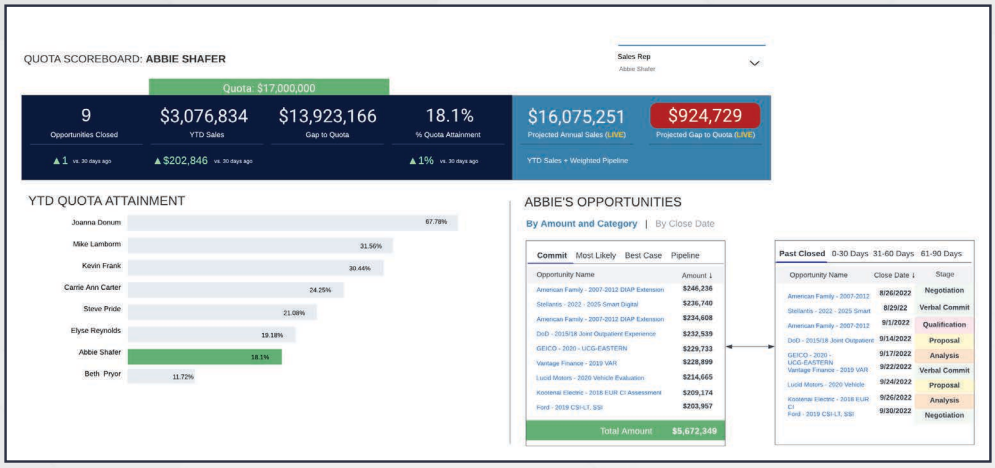
Tableau or CRM Analytics: Some of the Decision Points are Straightforward

If you're dealing with back office data, or if your primary use case is deep self-service data analysis within the tool, Tableau is the clear choice. It has significantly more pre-built connectors to data sources of all kinds, and the depth of options when it comes to visualizing data and enabling self exploration of data are unmatched. But as you examine CRM-based use cases, specifically those leveraging Salesforce as the primary source of data, the decision gets murkier. It's no accident that Einstein Analytics became first "Tableau CRM" and now "CRM Analytics."

Salesforce, intentionally or not, was clearly telling us that this is the analytics tool that best fits needs when working with Salesforce CRM data and use cases. But since the Tableau acquisition, Salesforce and the Tableau product team have made strides to close the gap between Tableau and CRM Analytics for use cases involving Salesforce. Instead of a blanket statement of "If your use case is in Salesforce, CRM Analytics is going to be the preferred choice," we now need to look a layer deeper.

The goal of this whitepaper is exactly that: Take a common CRM use case, build it in both CRM Analytics and Tableau, and pinpoint the next layer of decision points that may steer you in one direction or the other.

The Use Case



In order to go a layer deeper, we chose perhaps the most common CRM-focused use case we encounter at Atrium with our customers: a quota attainment dashboard, embedded on the Salesforce Home Page. Ask any salesperson what question they care about most, and you most certainly will get “Am I going to hit my number?” To explore this, we created this working wireframe to align our efforts.

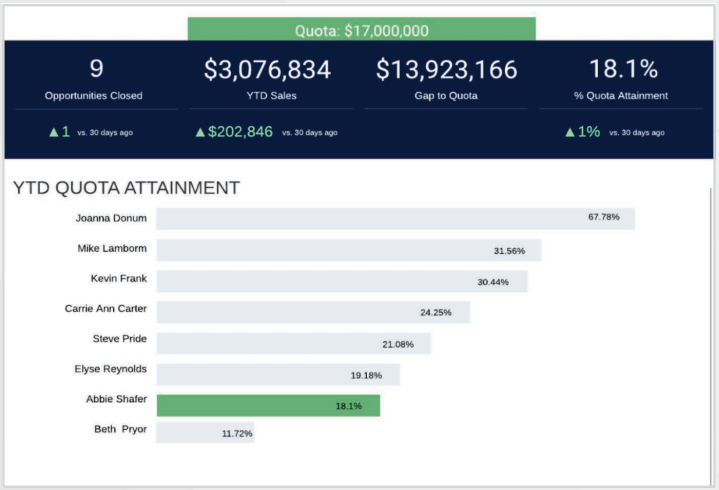
A few key features to point out with the above mockup:

Ability to switch between several reps:

As part of the POC, we’ll also explore defaulting to the logged-in user for this dashboard (which would remove the selector).

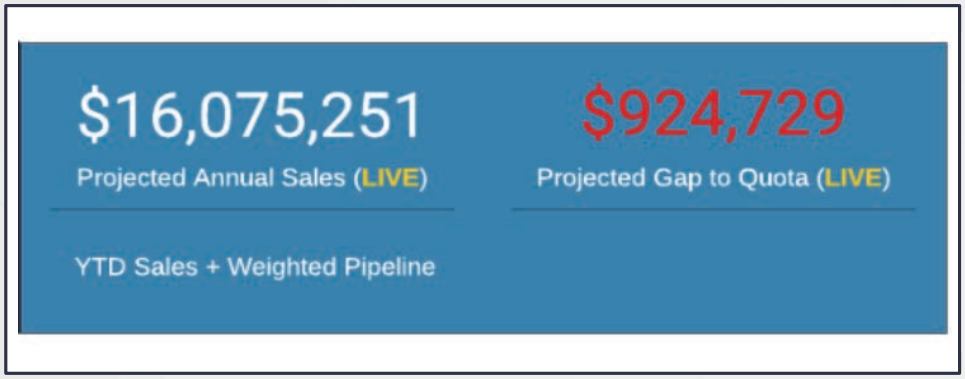
YTD Metrics (based on actuals):

On the left hand side of the dashboard, we track sales and quota attainment year-to-date, including how data is trending over the previous 30 days. This data is calculated using recipes to create datasets (CRMA) or Tableau to create Output files and is static until refreshed.



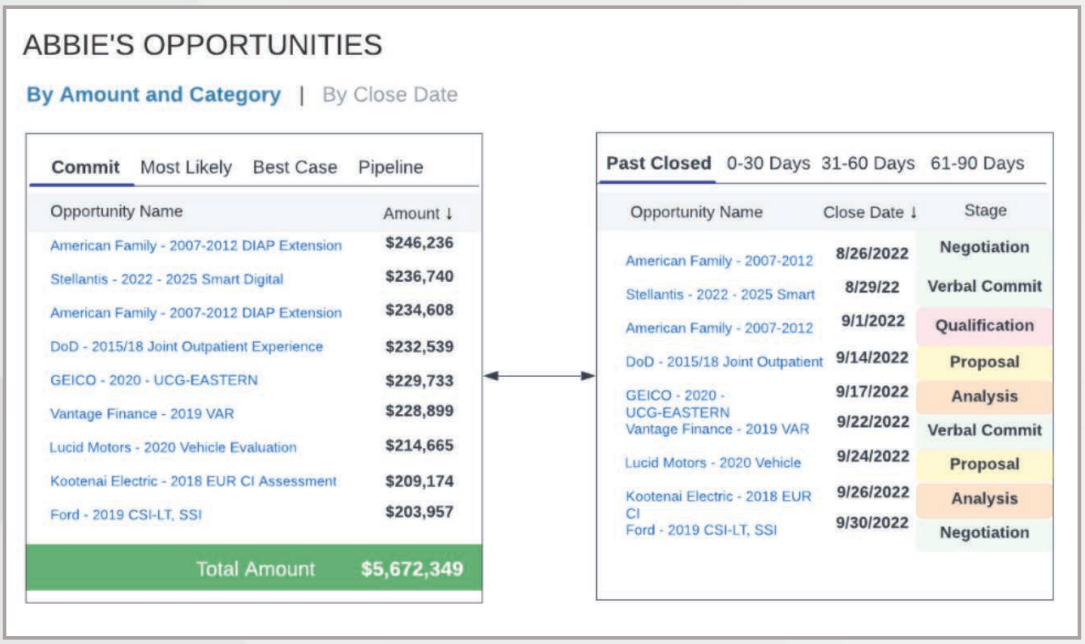
Live Projected Metrics:

On the right hand side of the metrics bar, we have live calculated values, based on the current value of any opportunities. This allows reps to see how they're tracking, and see changes they make to opportunity amounts or forecast probabilities reflected immediately in their numbers.



Actionable Opportunities:

If a rep isn't where they want to be, and if they want to close their gap to quota, this swap-pable component presents all opportunities owned by the selected (or logged-in) user. Clicking the "By Amount and Category" or "By Close Date" links switches the view as shown to see lists of sortable opportunities to allow the user to quickly find what they should work on. Best of all, this allows a rep to drill directly into an opportunity to work on it (e.g., update the amount, update forecast category, update close date, etc.). They'll see those changes immediately reflected in the live metrics on this dashboard.



Preparing the Environment with Data

With our use case solidified, it was time to get to work. Step 1 was to create and load the data we needed for our POC into Salesforce. Leveraging a handy data generation tool called — wait for it — Data Generator (available on the AppExchange), we created Accounts and Opportunities in a Salesforce.com environment. We also manually created users in Salesforce, and loaded quotas into a custom field on each user record.

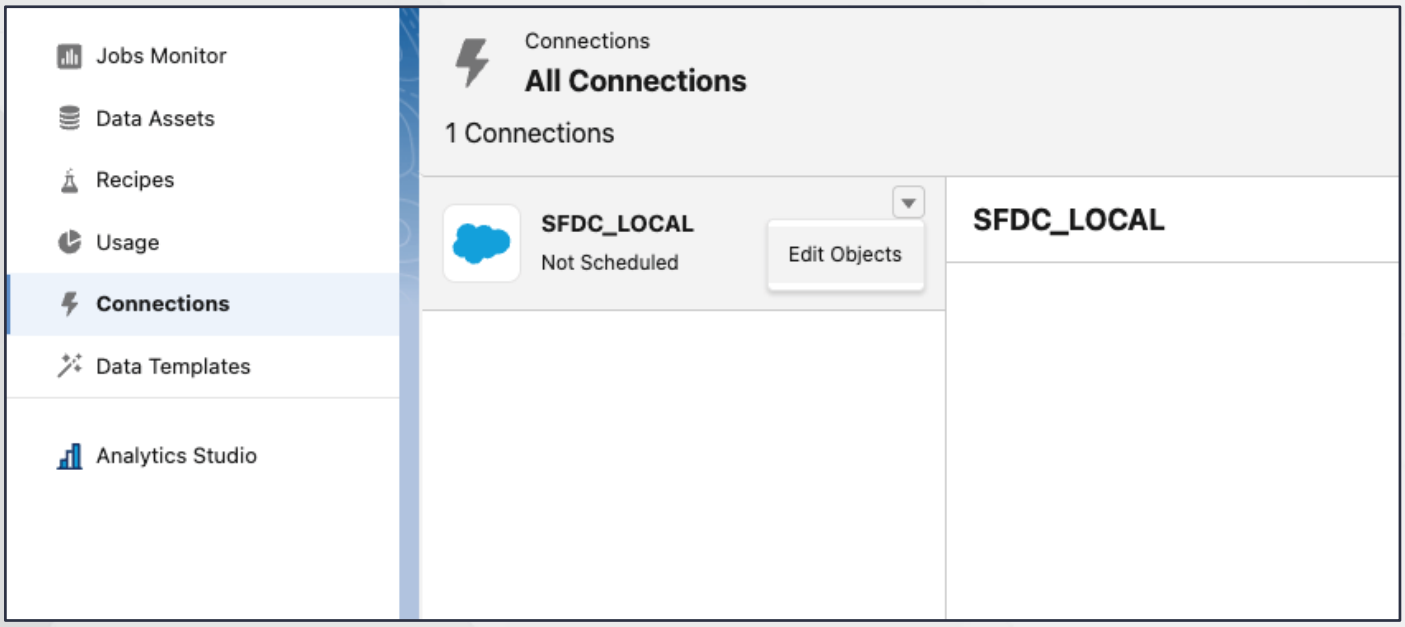
One slight limitation of Data Generator is that it isn't great at creating relationships between generated records. For example, if youcreate 10,000 Opportunities using the tool, you will be inserted as the Owner of all 10,000 records, and there's no way to override that or set conditions on percentages of records owned by an array of users. To solve for this, we did a quick extract, update, and reload of the opportunity data to modify Owner Id and Account Id on those records.

Comparison #1: Data Sync and Shaping

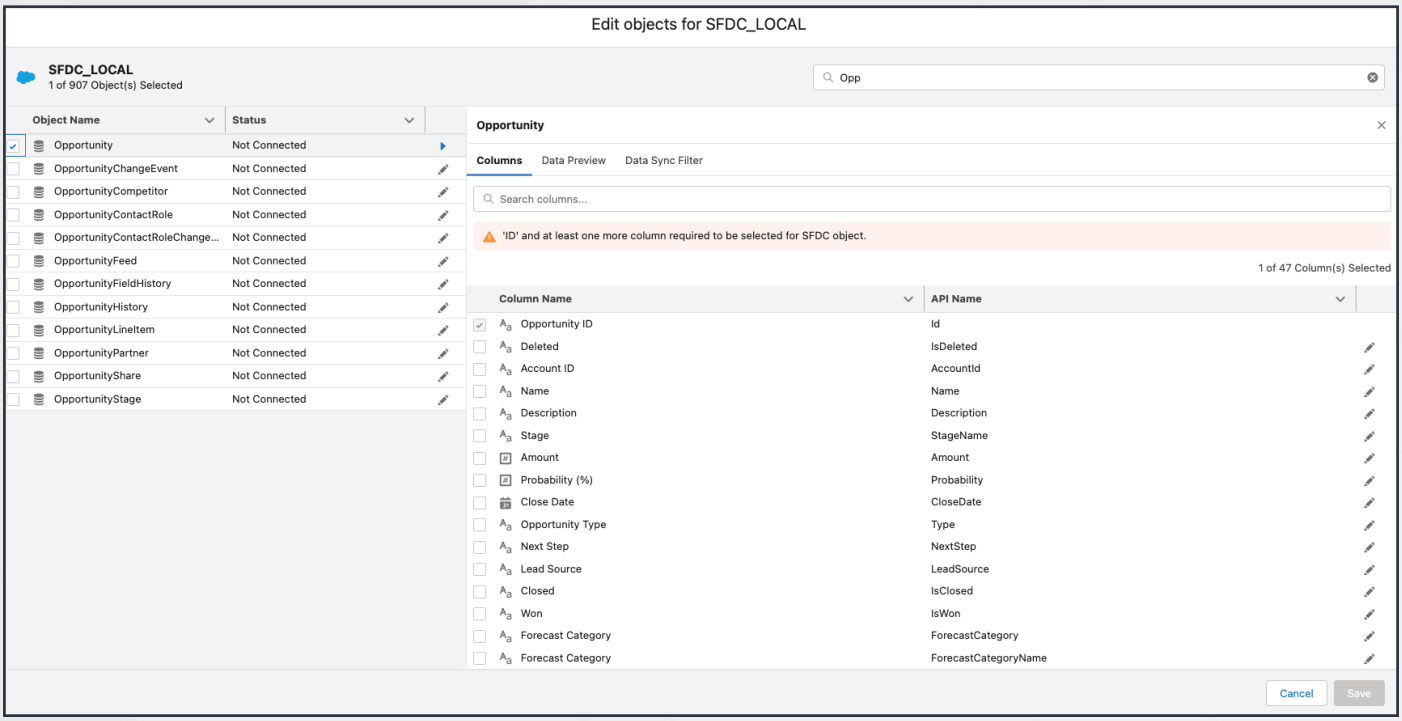
With our base data in Salesforce, it was now time for our first head-to-head comparison of CRM Analytics and Tableau. To start, we'll examine the process of accessing Salesforce data via both tools, and the features at our disposal within both tools to schedule regular syncs of data and to shape it for use within our dashboard.

CRM Analytics

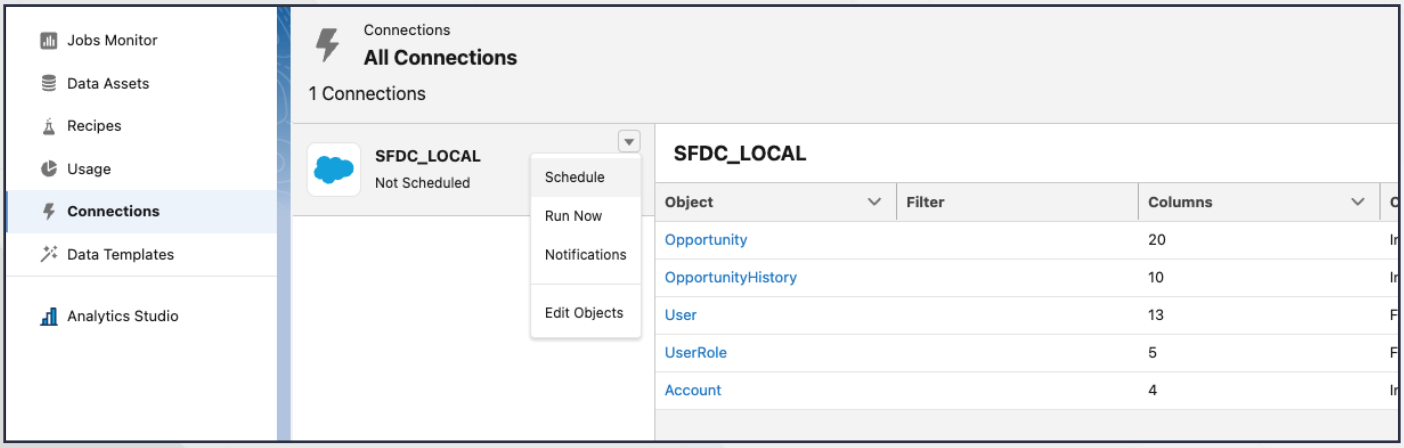
One great advantage of CRM Analytics is that Salesforce has made it so easy to get at data that lives in Salesforce. We started by modifying the default "SFDC_LOCAL" connection, which gives access to all objects in the instance in which you're running CRM Analytics.



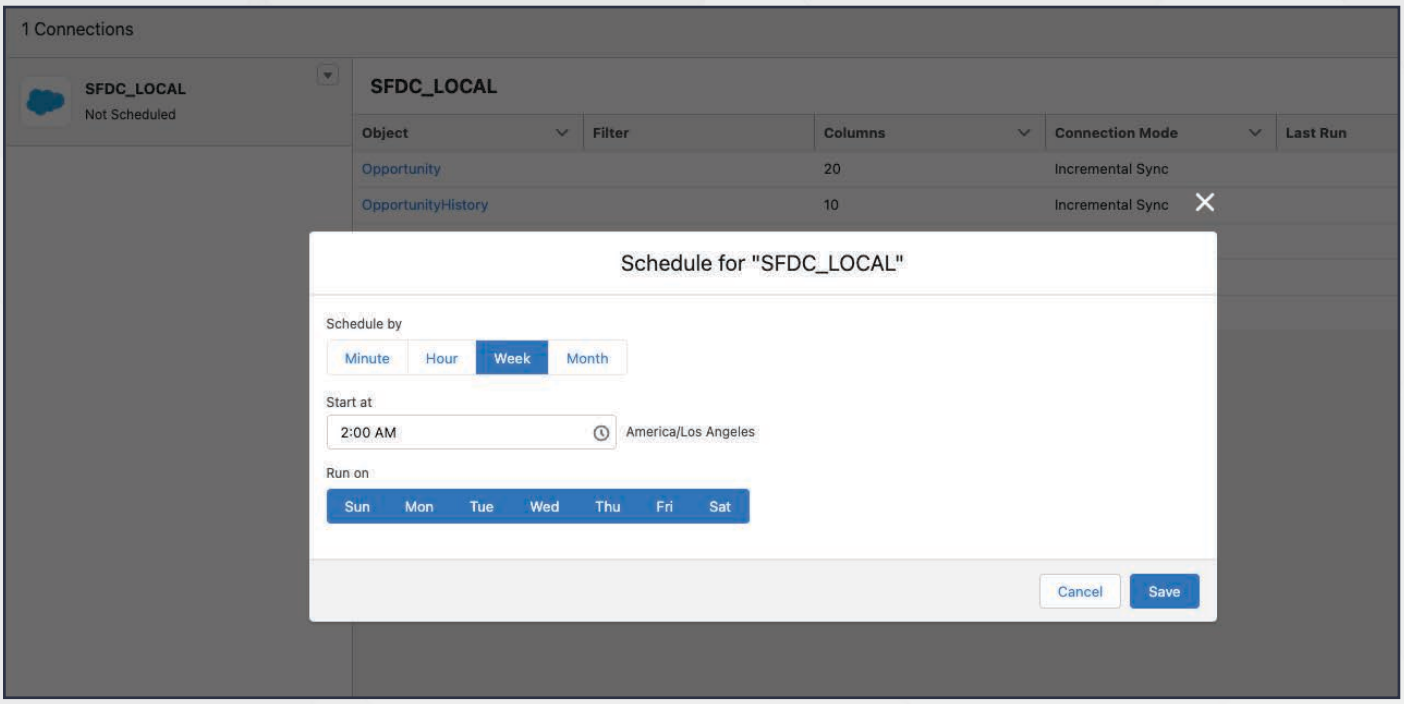
Using clicks, we could go through, find the objects that we needed (in our case, Account, Opportunity, OpportunityHistory, User, and UserRole) and select the columns we needed from each object. While the temptation is to "just bring over all the columns," Salesforce has intentionally not created a "select all" feature here. It's in your best interest to only grab what you need in order to maximize performance of your data sync and ensuing recipes.



Once the objects and fields are selected, you're able to set up a schedule from within the same screen, by which we could regularly refresh the data from Salesforce into CRM Analytics.

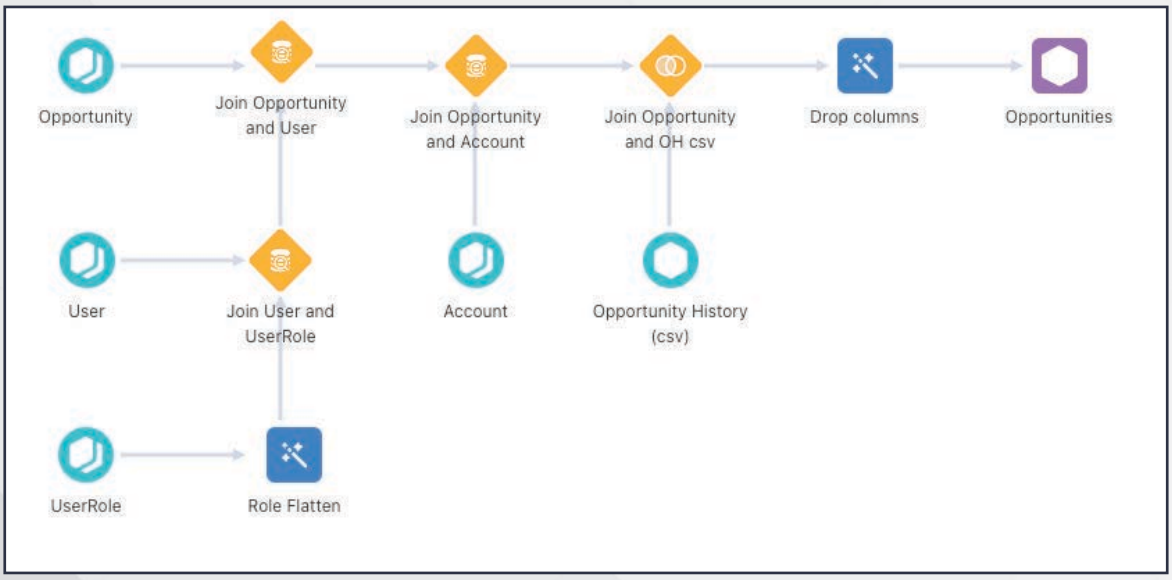


While data can be synced as frequently as every five minutes, we chose to schedule this refresh to happen once per day.



Now that we have access to replicated data in CRM Analytics, we created a simple recipe to join the appropriate tables and prepare datasets for use on the dashboard. For the purposes of this exercise, we're using some mocked data in a csv file for Opportunity History (since we just loaded temporary Opportunity records into the environment).

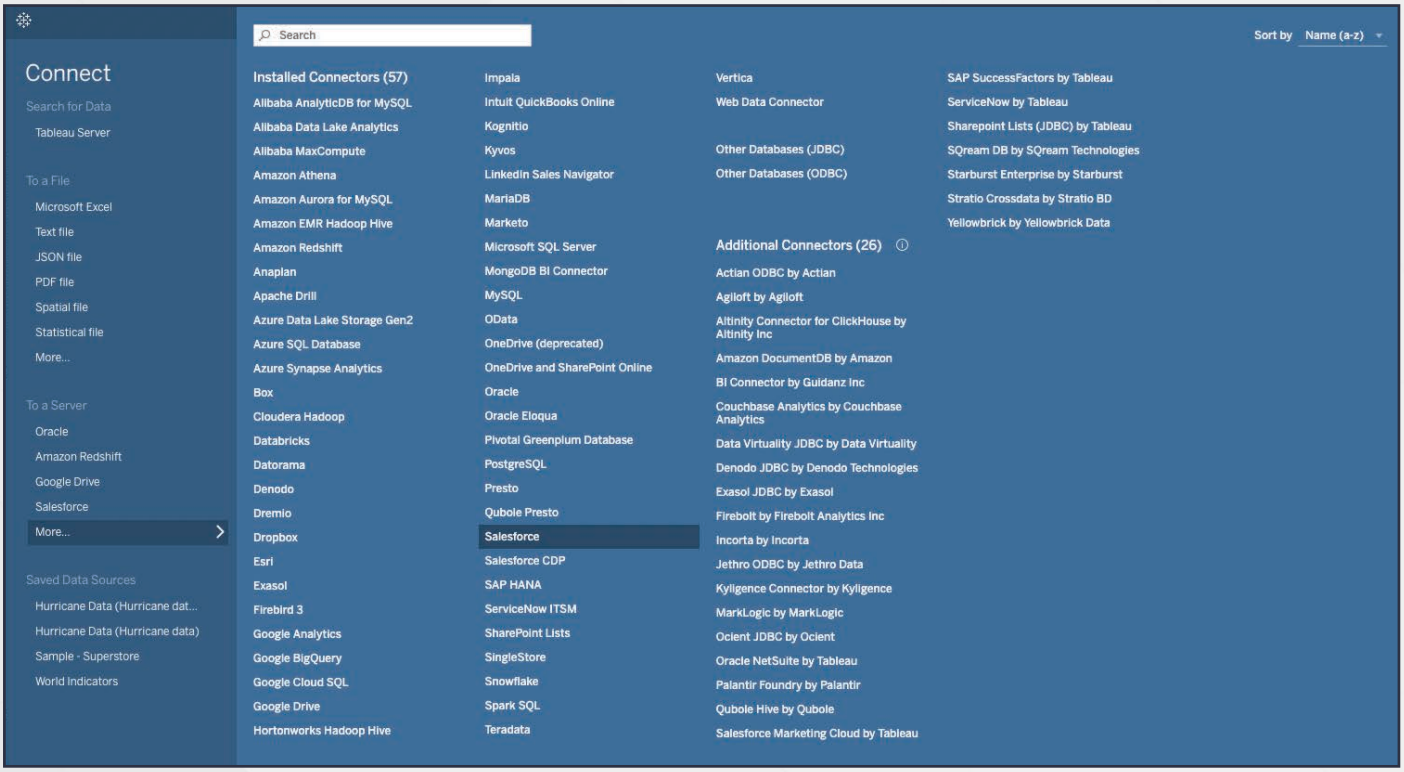
Here's how the recipe looks:



The end result is a dataset that has all of the information we need in order to create our dashboard in CRM Analytics.

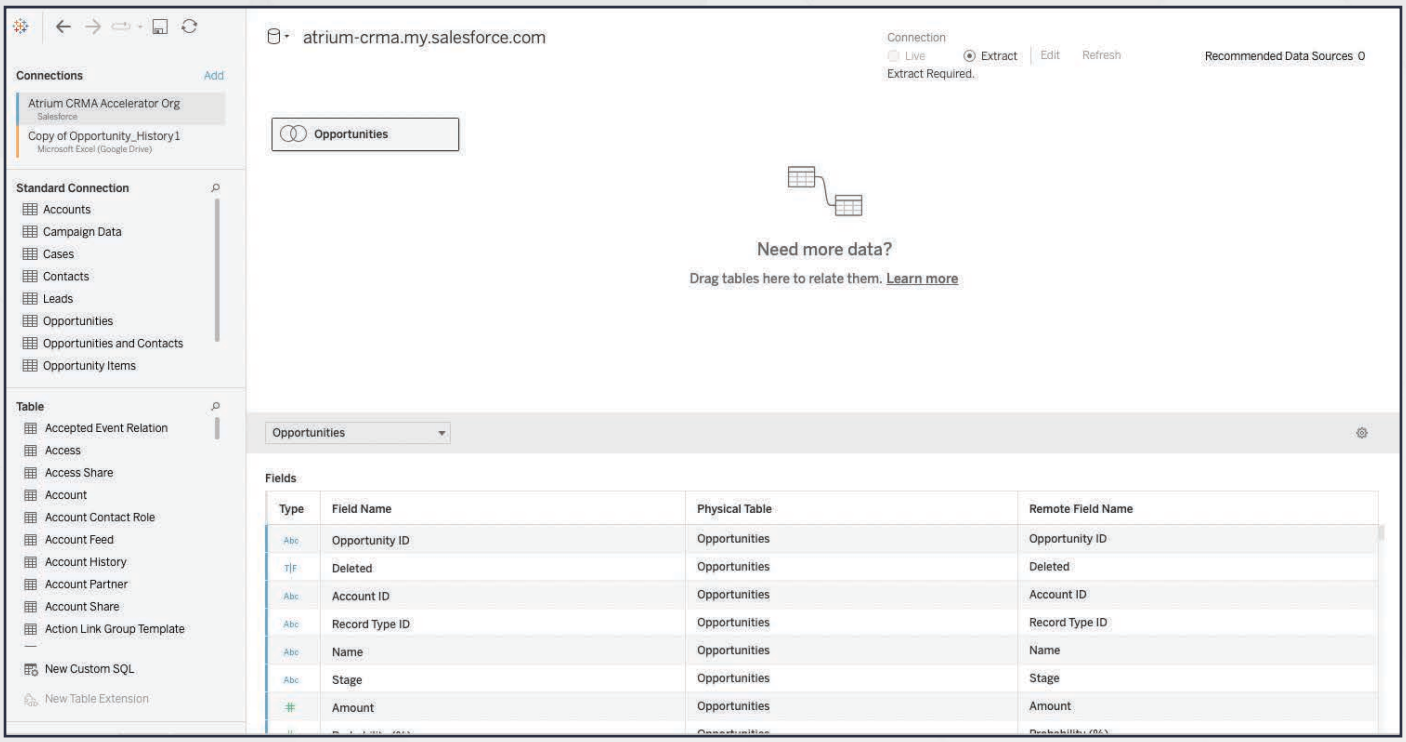
Tableau

Similar to CRM Analytics, Tableau can connect directly to our Salesforce data, allowing us to get to work in selecting the tables we need for our analysis. We started by selecting a direct connection to our Salesforce org data (one of 57 native connectors within Tableau) using Salesforce authentication. We can now start exploring our data.



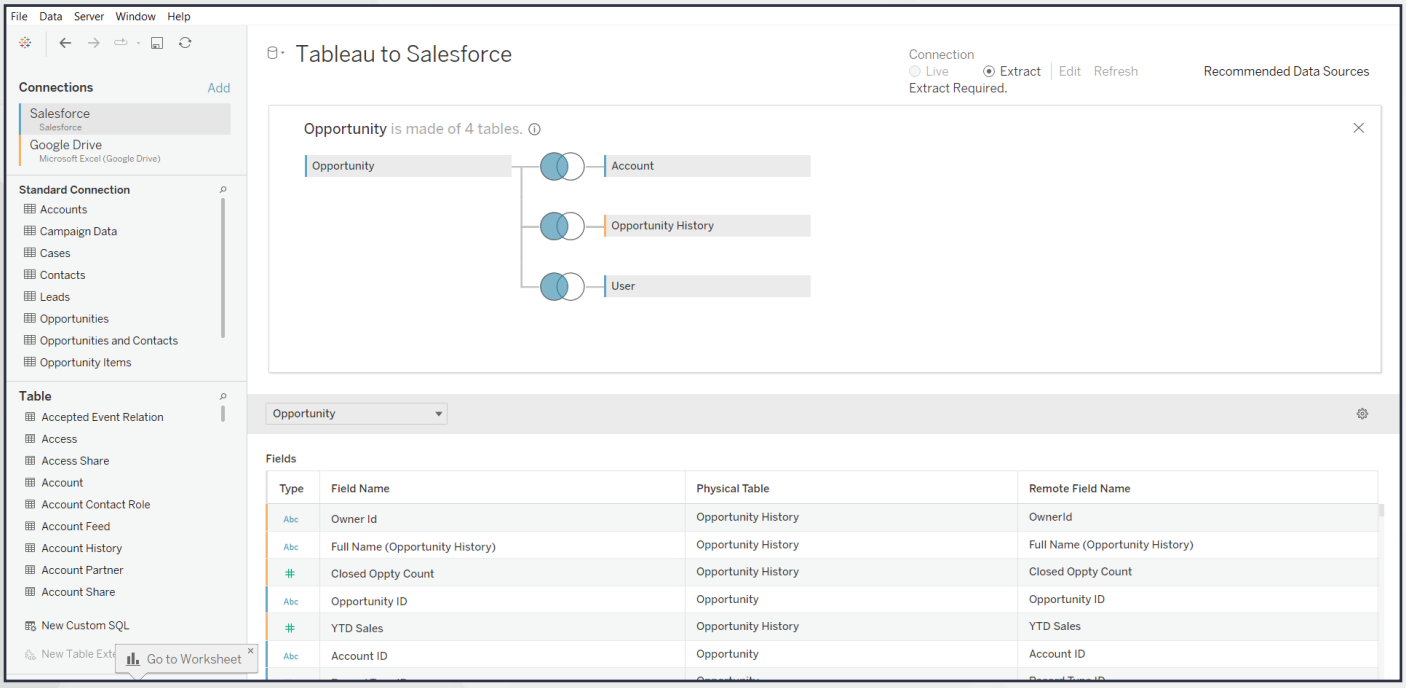
Similar to the CRM Analytics solution, we used mocked data for Opportunity History (since we just loaded temporary Opportunity records into the environment). With Tableau we converted the csv file to Google Sheets and connected directly to Google Drive for the data source. We also had the option to connect to a standalone csv file, but connecting directly to Google Drive allows for connection to a live/editable file.

Once connected to the data, we can start to create our data source, identifying which tables and fields we need to do our analysis.



Directly in Tableau, we can drag and drop the tables we need in our analysis and with a few clicks create the appropriate joins and relationships between tables. We can also prepare the data fields we need within the field section of the Data Source pane. No coding needed! You can easily execute joins with only the tables that are needed through simple, user-friendly drags and editing directly in the data pain. In order to remove fields in either the standard connections or manually joined tables, hide them within the data pane by selecting the field name, right clicking, and selecting “Hide.”

Also, while the data extract can be set to as frequently as you need, we will set our data extract to refresh once a day, which will allow for a performant experience.



The end result is also a dataset that has the key components we need in order to create our dashboard in Tableau.

Conclusion

For static, replicated data (datasets or extracts), both tools provide everything that we need. CRM Analytics recipes are slightly more advanced for data manipulation, but many of those features have been ported over to Tableau recently and the gap is essentially nonexistent.

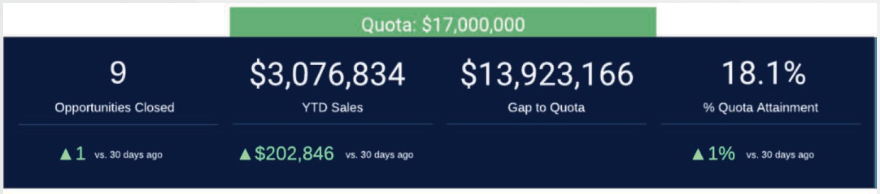
For this specific use case, there is no clear need for one tool over the other for the replicated data. Live data, however, is a different story, which we'll cover soon.

Comparison #2: Creating the Dashboard

With a clear wireframe to work from, creating the dashboard should be a fairly straightforward process on both platforms. We'll tackle YTD Metrics first, which are fairly straightforward calculations, but also involve joining to our Opportunity History data to calculate the trend. From there, we'll build what appears to be a simple YTD Quota Attainment bar chart, but with the added feature of highlighting the selected user in the bar chart. We'll also need to set up the Sales Rep filter to test this. Lastly, we'll grab the list of opportunities owned by the selected user and build our toggle to switch between "By Amount and Category" and "By Close Date" for our display and sort.

But wait — what about the live data calculations? Those are a bit trickier, and so we'll handle those last and break out the comparison for that piece of the dashboard, since it's a significant differentiator for one of our products.

CRM Analytics



YTD Metrics:

There are 5 straightforward calcs we created in CRM Analytics to cover the 4 leftmost measures as well as the Quota for the selected user. Since all of these can be pulled from the Opportunity dataset we created, we are able to calculate them in a single query, and reference that query in parts of the dashboard:

Query

Switch to SQL | Run Query

```
q = load "Whitepaper";
q = filter q by 'StageName' == "Closed Won";
q = filter q by date('CloseDate_Year', 'CloseDate_Month', 'CloseDate_Day') in ["current fiscal_year".."current day"];
q = group q by 'OpptyOwner.Name';
q = foreach q generate q.'OpptyOwner.Name' as 'OpptyOwner.Name', sum(q.'Amount') as 'YTD Sales', first(q.'OpptyOwner.Quota__c') as 'Quota', unique(Id) as 'Opp Count';
q = foreach q generate 'OpptyOwner.Name','Opp Count', 'YTD Sales', 'Quota', 'Quota'-'YTD Sales' as 'Gap to Quota', 'YTD Sales'/'Quota' as '% Quota Attainment';
q = order q by 'OpptyOwner.Name' asc;
```

Full Name ↑	Opp Count	YTD Sales	Quota	Gap to Quota	% Quota Attainment
Abby Shafer	3	3,305,097.78	17,500,000	14,194,902.22	0.19
Barbara Del Dotto	2	2,746,571.04	24,500,000	21,753,428.96	0.11
Beth Pryor	3	3,747,455.5	20,500,000	16,752,544.5	0.18
Elyse Reynolds	2	2,373,231.4	20,500,000	18,126,768.6	0.12
Hakeem Turnbull	1	311,774.08	22,500,000	22,188,225.92	0.01
Joanna Donum	1	1,013,244.93	22,500,000	21,486,755.07	0.05
Mike Lamborn	1	100,000	19,500,000	19,400,000	0.01
Randy Dunn	7	7,471,520.36	24,500,000	17,028,479.64	0.3

Lastly, we need to calculate the trend over the last 30 days by joining to our mock Opportunity History dataset. To do this, we'll filter that dataset to the snapshot date 30 days ago, run the same query against it that we did for the top line metrics, and then join (cogroup) those results with our top line calc to determine the difference.

YTD Quota Attainment:

Our next task is to create a standard horizontal bar chart displaying all sales reps in descending order of Quota Attainment. There are a couple of things that make this trickier than it appears. First, Quota Attainment will need to be calculated on the fly (as we did in the previous section). Second, we want to change the conditional formatting of the chart to highlight the selected user.

The calculation for Quota Attainment is sum(YTD Sales)/Quota. Since we have these measures in different datasets, we'll want to join them as part of the query, right? Not necessarily. One shortcut that makes this a bit easier is to also stamp the quota amount for the opportunity owner on the opportunity record. This can be confusing for those working with the data, so do it judiciously, but it's a shortcut that can speed up query time and reduce SAQL complexity. That's not to say this is the only place it should live. We'll want both versions of Quota in our data — a standalone dataset that's used for many queries, as well as a stamped reference point on the opportunity for scenarios like this calculation.

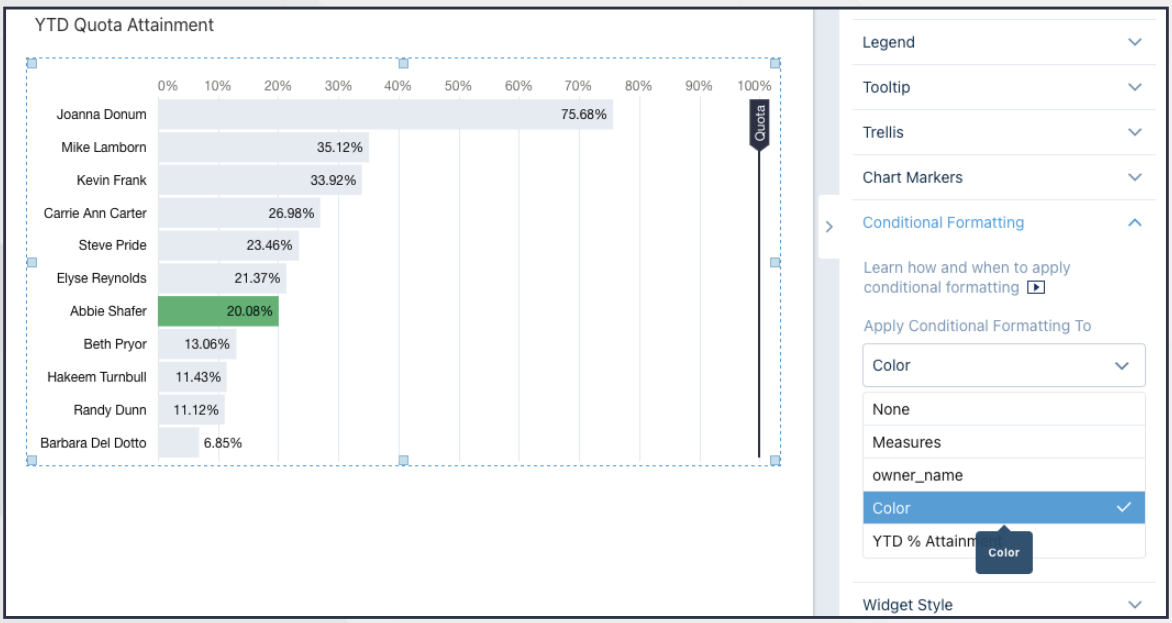
To achieve our changing highlighted color, we essentially create our query twice. Once where the Opportunity Owner does not equal the selected user, and once where it does. We assign a different value to a derived dimension "Color," which will then allow us to set conditional formatting on the chart based on that value. Here's how it looks in SAQL:

```
other = filter q by 'owner_name' != "{{cell(owner_name_6.selection, 0, "owner_name").asString()}}";
other = group other by 'owner_name';
other = foreach other generate 'owner_name', sum('amount')/first('owner_quota') as 'YTD % Attainment', "Team" as 'Color';

user = filter q by 'owner_name' == "{{cell(owner_name_6.selection, 0, "owner_name").asString()}}";
user = group user by 'owner_name';
user = foreach user generate 'owner_name', sum('amount')/first('owner_quota') as 'YTD % Attainment', "User" as 'Color';

u = union other, user;
u = group u by ('owner_name', 'Color');
u = foreach u generate 'owner_name',  'Color', sum('YTD % Attainment') as 'YTD % Attainment';
u = order u by 'YTD % Attainment' desc;
```

owner_name	Color	YTD % Attainment ↓
Joanna Donum	Team	0.76
Mike Lamborn	Team	0.35
Kevin Frank	Team	0.34
Carrie Ann Carter	Team	0.27
Steve Pride	Team	0.23
Elyse Reynolds	Team	0.21
Abbie Shafer	User	0.2

You'll notice two areas highlighted. In the code, you'll see that we have included a binding to our Sales Rep list selector, which allows us to dynamically change the value based on the rep that is selected (in this case, it's Abbie Shafer). We will cover how to default to the logged-in user later in this whitepaper, but for testing purposes, it's easier to drop this into the dashboard now. Secondly, in the bottom output table for the query, you'll see our derived dimension "Color" and the associated values of "Team" and "User."

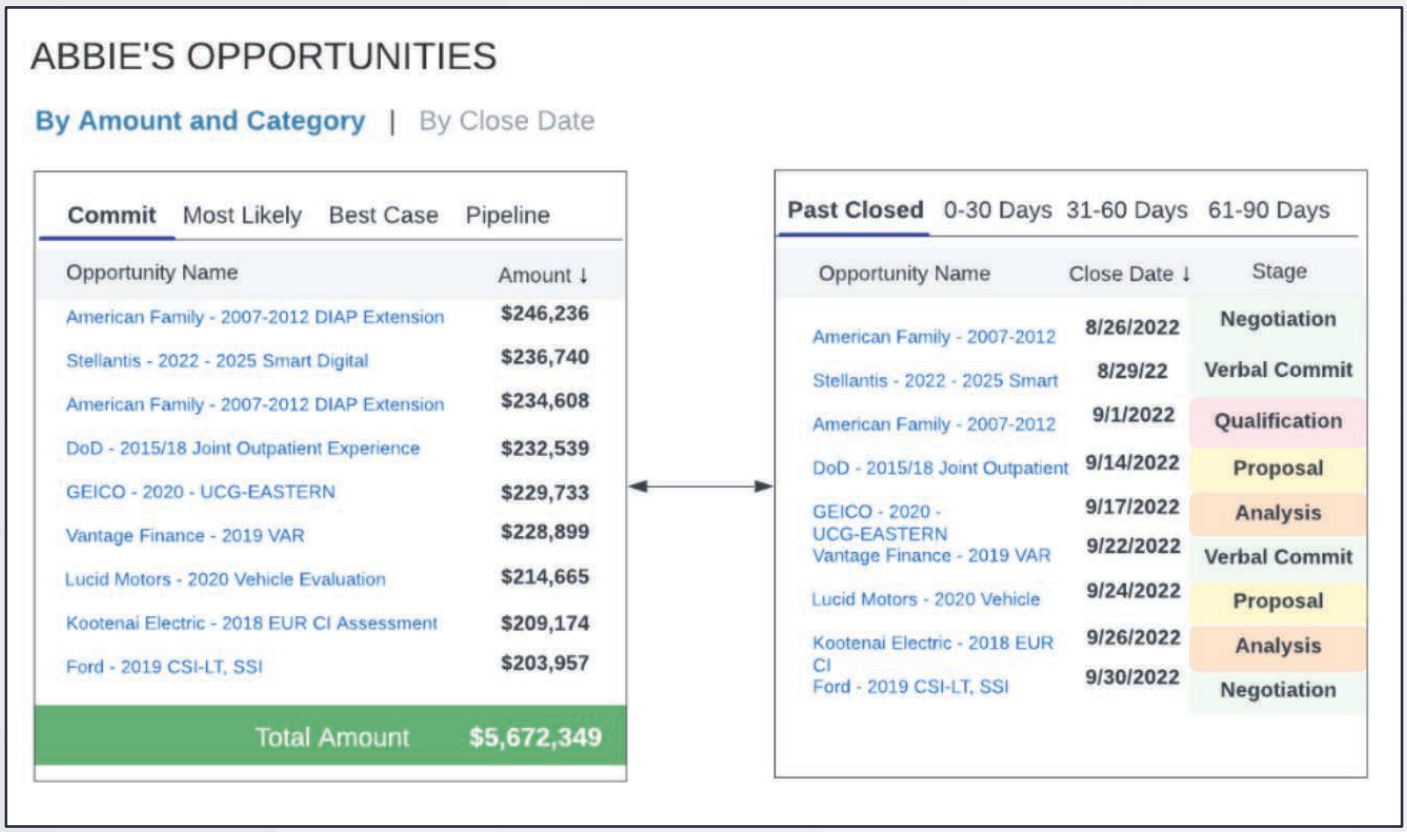
Our last step to complete this chart will be to modify its conditional formatting using that derived dimension. To do so, highlight the chart and navigate to the widget sidebar panel. Expand Conditional Formatting, and select the dimension/measure (in our case, "Color") off of which you want to base the conditional formatting.



Pick your values (in our case, “Team” and “User”), and select an appropriate color for each.

Opportunity Lists and Quick Filters:

To complete the dashboard, we need a way to create an actionable list of Opportunities owned by the selected user, with the ability to quickly toggle between presenting them “By Amount and Category” and “By Close Date.”



To handle this in CRM Analytics, we have a few options. In the Salesforce Revenue Insights app, there is a similar feature built as a series of pages within a component. But because we’re going to embed this in Salesforce, and we need to bind to the logged-in user to filter the dashboard properly, components aren’t an option (you can’t bind to them). That leaves us with 2 options: create separate dashboard pages for each of the 8 scenarios (the 4 Amount and Category filters, plus the 4 Close Date groupings) or use a toggle and a binding to switch the filter on the table.

Both options leave us with some potential technical debt. If we go thepages route, we may increase our work if we add additional widgets to the dashboard. If we do, we’ll need to copy those to every page, and position them appropriately. And beware! Any changes in the position of widgets on the dashboard are not replicated across pages, and so you’ll be duplicating effort there.

The binding route leaves us with some relatively complex code to maintain, and it means we’re binding to every query that needs to operate differently between pages. That can exponentially increase if those pages become more complex.

The decision depends a bit on the long-term plan for the dashboard as well as the level of expertise of the team supporting the solution. Because we built this dashboard specifically for the purposes of the whitepaper, we took the simpler and faster option and chose the pages approach.

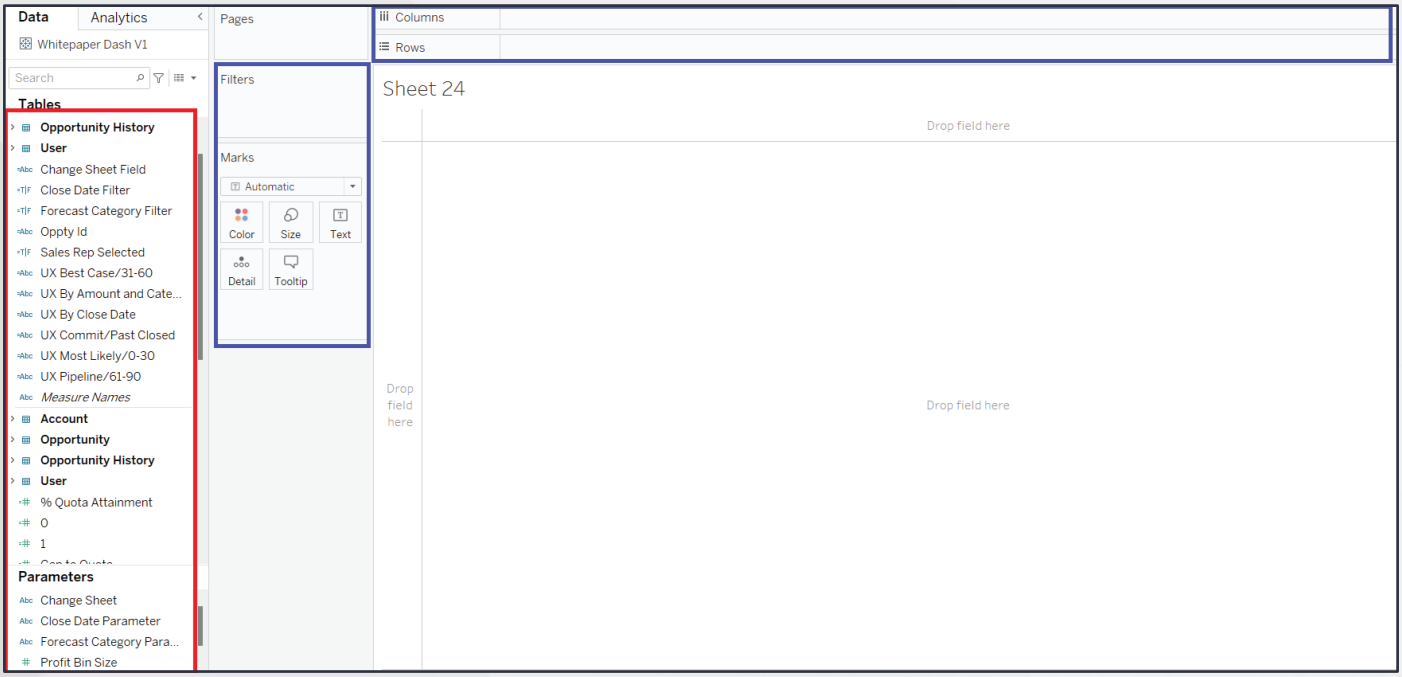
Tableau

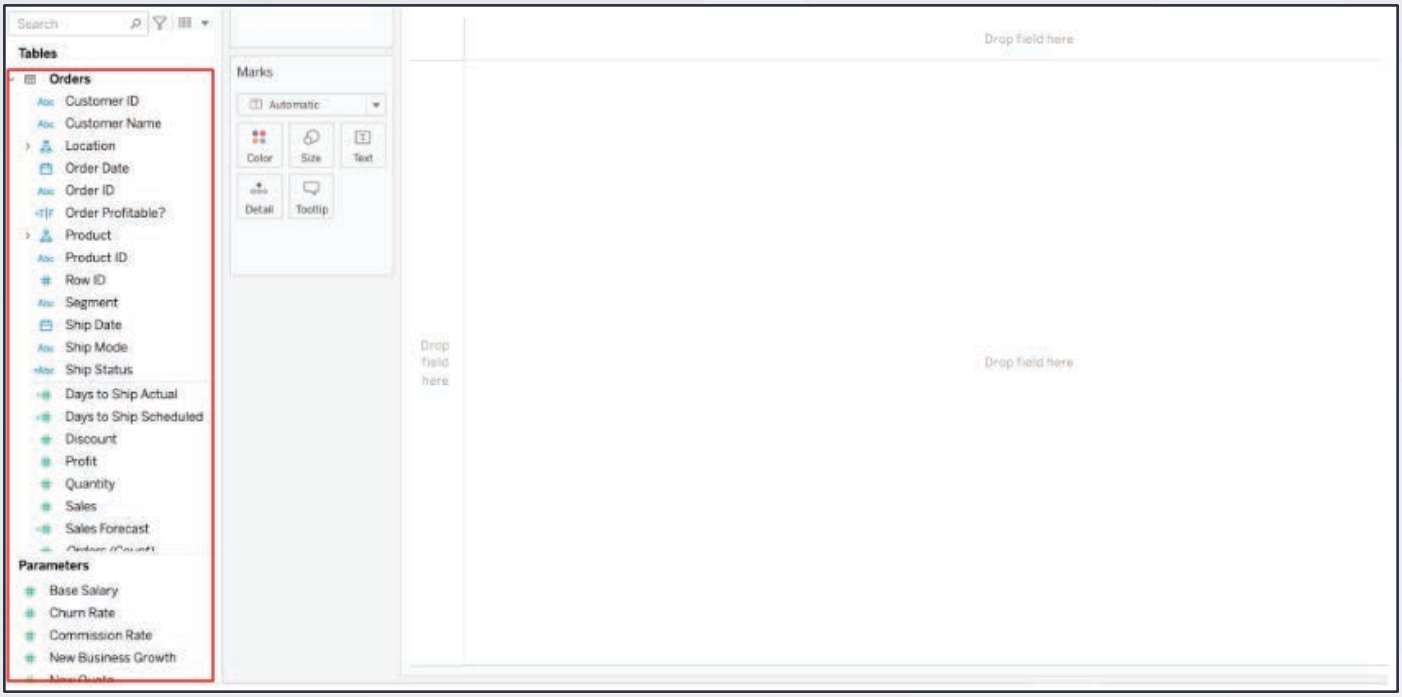
With clear requirements and a wireframe designed to incorporate visual best practices, we are ready to begin building our Views (i.e., worksheets) and Dashboard. You will start by creating views (using worksheets) in Tableau off of the data source we've created. We will also start with the YTD Metrics.

YTD Metrics:

There are 5 metrics we will include in the dashboard to address key KPIs tracked for Sales Quota. We will start by creating the 5 metrics within sheets by simply dragging and dropping the fields from the Data Pane into the shelf of the sheet.

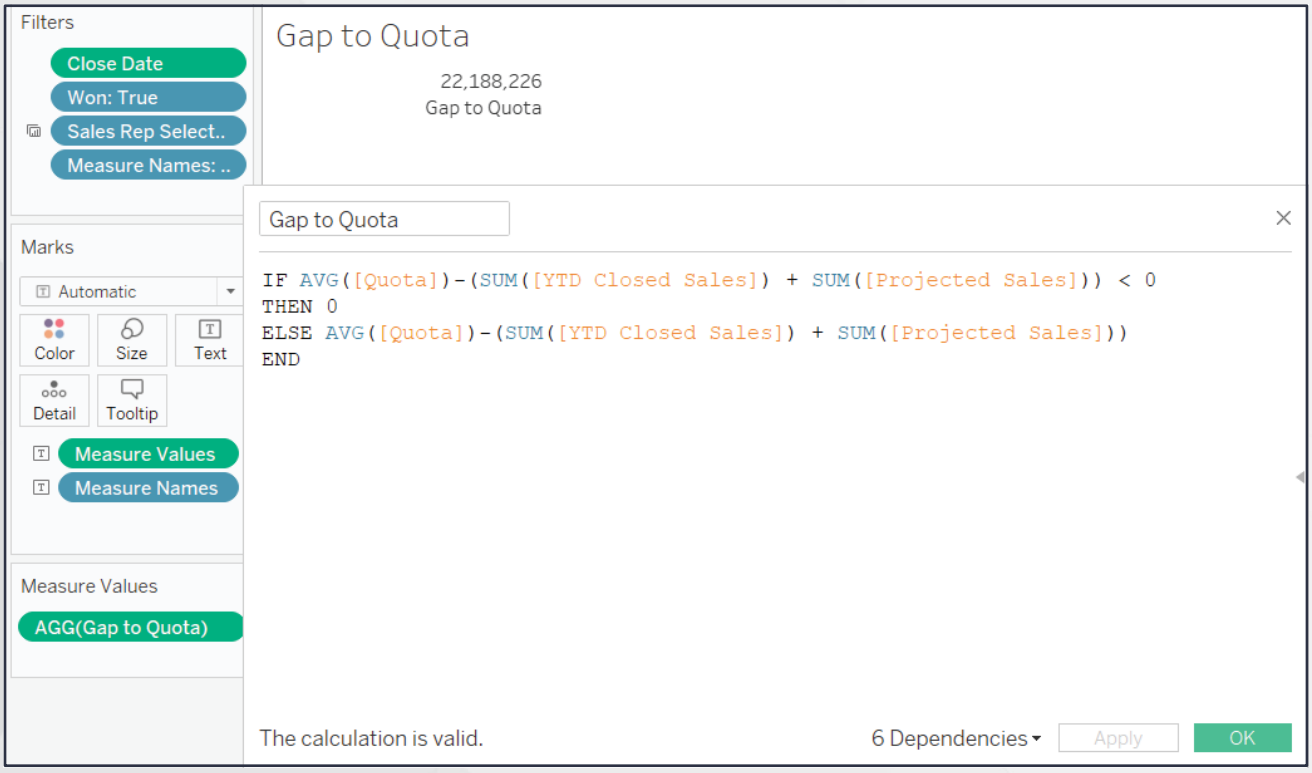
The data tables and fields will automatically show up in the Data Pane of the worksheet on the left side of the worksheet, allowing for easy organization, as well as configuration of the fields we want to build and analyze.





We will then create the calculated fields for our trend over the last 30 days by using our historical data (which is in the data source and Data Pane). We can also do this directly in the sheets within the workbook. Calculations can be made on one sheet or individual sheets, but we've created our calculations in individual sheets for easier formatting when we create our dashboard.

Below is an example of one of the calculations, in which we were able to create through building directly in the sheet by right clicking and editing in the calculation pane. This allows for customization of fields, calculation, parameters, and analysis all while staying within the sheet we are building our view.

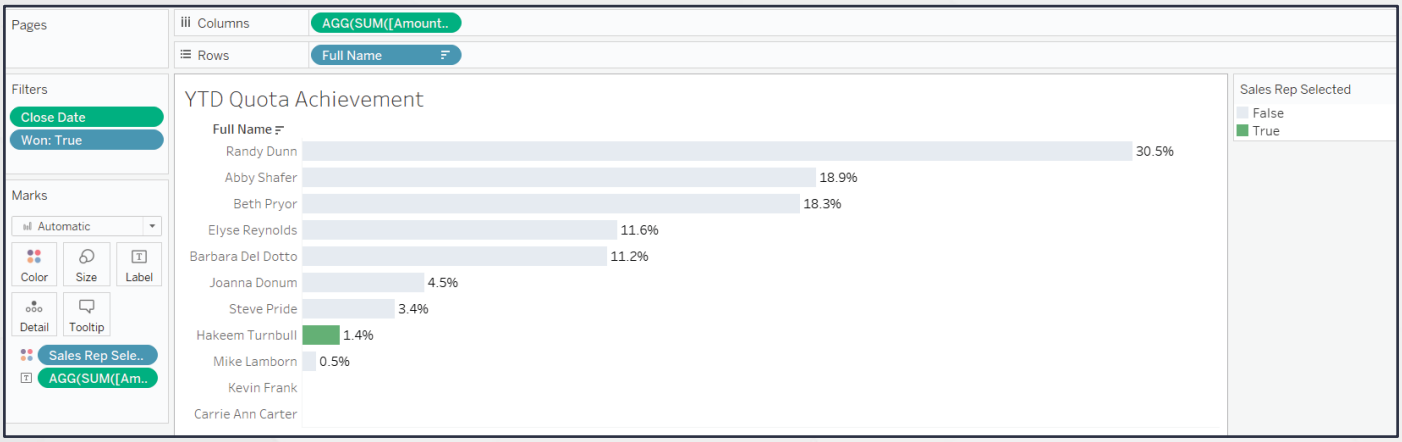


We built views for each of the 5 key metrics and the gap to quota and format accordingly within the view shelves so they will be ready for inclusion into the Dashboard.

YTD Quota Attainment:

Next, we will create our bar chart view for YTD Quota Attainment for each sales team member. A huge positive of Tableau is that the amount of coding needed is significantly lower than in CRM Analytics in this view. This can make upkeep, alterations, and iterations much easier. For example, where CRM Analytics needs SAQL coding to create this chart, Tableau simply requires an in-line calculation and dragging the appropriate pills from the data pane to the correct columns, rows, and marks within the Tableau shelf.

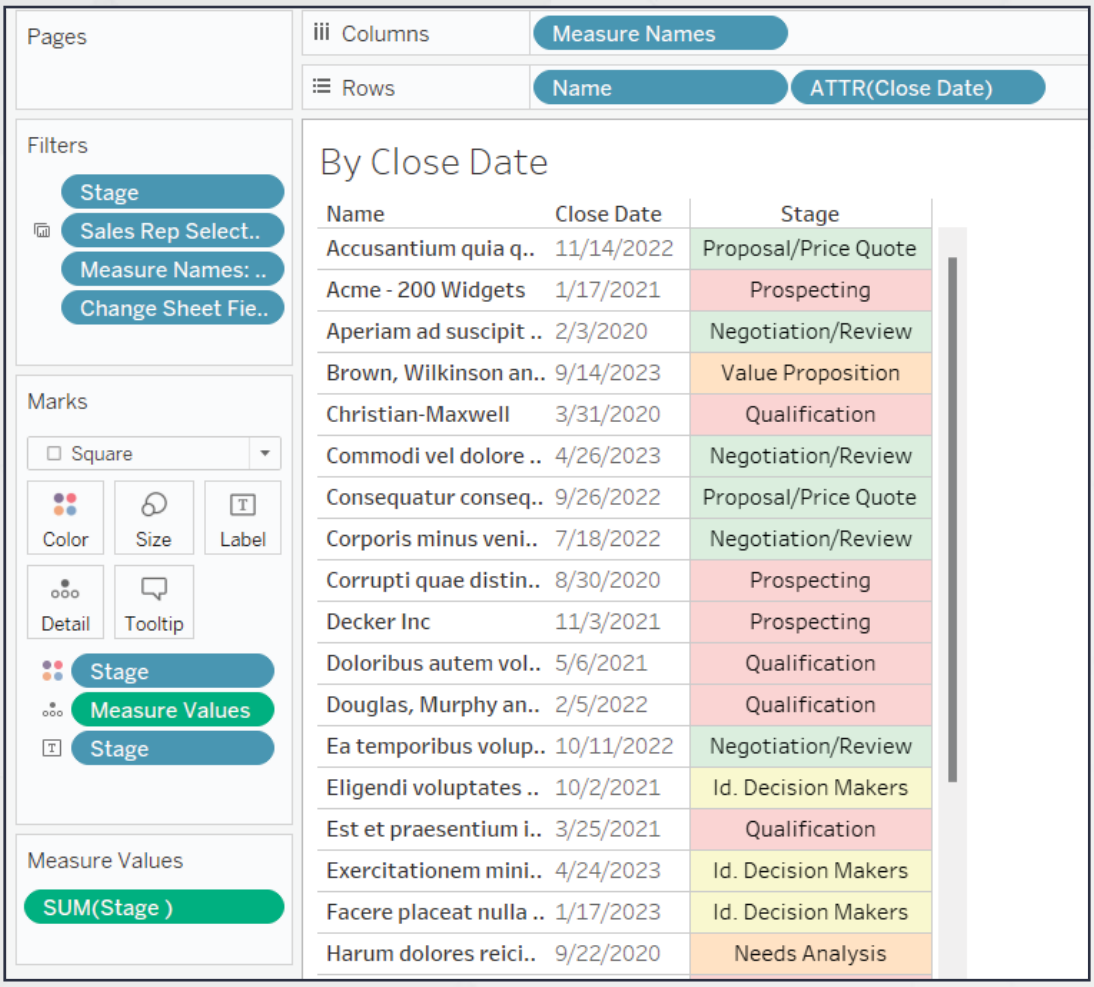
The calculation for Quota Attainment is sum(YTD Sales)/Quota. Although these measures are in different datasets, they will be available on the Data Pane for a simple calculation. We can change the formatting of marks directly on the shelf. We will cover the filters to show how you can easily filter data based on user parameters later on in this section.



As you can see in the screenshot above, filtering of the data and creation of the view can be done through drag, drop, and direct editing within the shelf. You can use “Show Me” within the shelf for Tableau to provide a recommended chart type or can manually choose through the marks section of the sheet. We manually created the bar chart below to format so that this view is ready for inclusion into the dashboard.

Opportunity Lists and Quick Filters:

Opportunity Lists and Quick Filters: Now on to the Opportunity Lists, where we will build out the dynamic charts and filters. Similar to how we built the Quota Attainment bar chart, we will build this view through dragging the pills in the rows, columns, and marks on the shelf. Text Tables in Tableau with one dimension require a bit of formatting to show the appropriate stage name. In order to name a column that is treated as a dimension in a table, create a calculated field simply titled that name (i.e., Stage) and a value of 0. The Measure Values and SUM(Stage) was created to name that column “Stage.” In order to shade the stages, we dropped “Stage” on the colors, changed the mark type to “Square,” and moved the size slider to the largest size. We then edited the colors based on the business requirements.



In order to create the dynamic swapping of how the user will view opportunities within Tableau, we set up sheet-swapping (i.e., we create the “By Amount and Category” and “By Close Date” tables in separate sheets), then will build in the functionality within dashboard actions to set dynamic swapping of sheets and filtering of the opportunities by the defined categories for each table.

Sales Rep Filter:

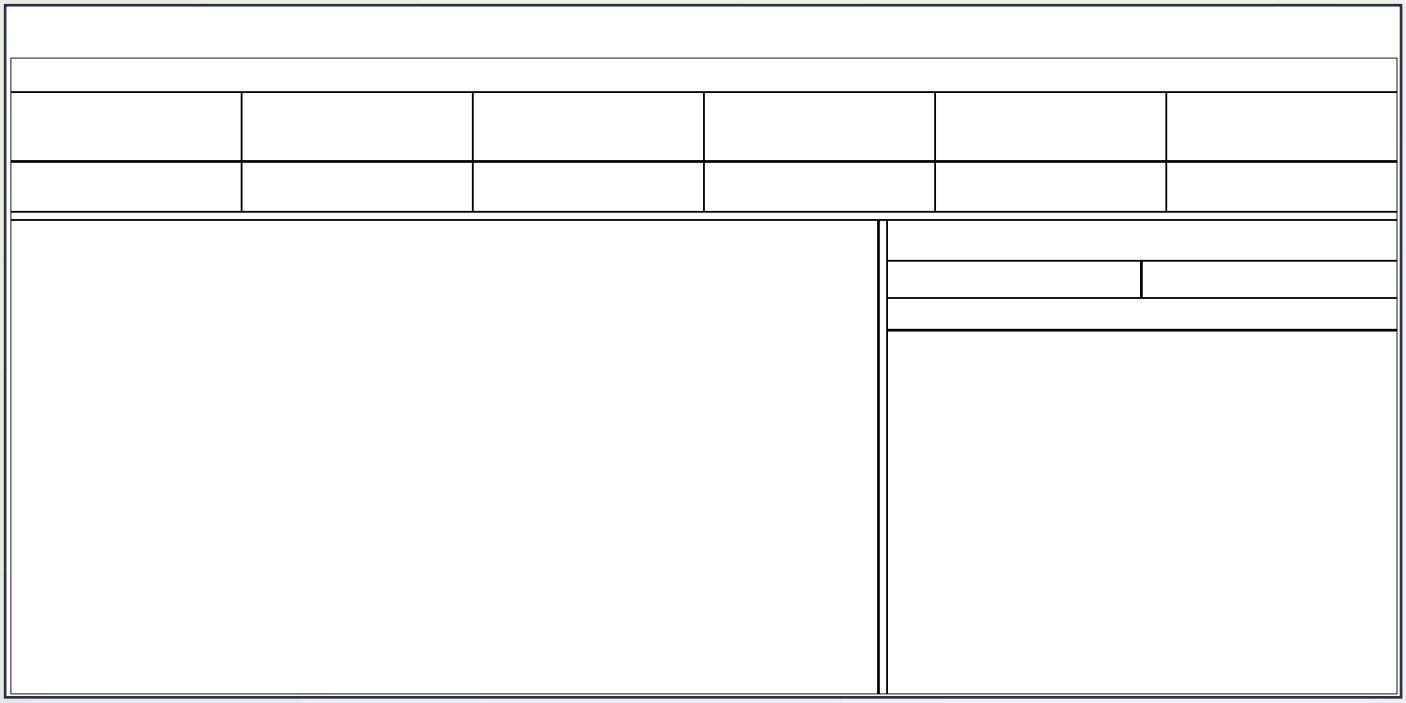
Now that we have our views created, we will create a filter for Sales Rep that will be linked to each of the views within the Dashboard. In Tableau, filters can be applied to data before it is visualized, single sheets, a selection of sheets/views, or within a dashboard containing multiple views.

Creating the filter was just as easy as it was on CRM Analytics. Where Tableau differs from CRM Analytics is in how we highlight the bars on the YTD Quota Achievement chart. We have to first create a parameter. In this scenario, we first created a parameter that holds the names of the sales reps, then created a calculated field to determine if the rep is the value of the parameter or not (T/F). The calculated field will be used to filter each of the sections. But instead of filtering in the bar chart, it will change the field to color accordingly.

Putting the Dashboard Together:

Now that we have all of our views set, we will bring them all together in one dashboard to create a user-friendly interface for viewing the Quota KPIs. Dashboard creation is slightly different in Tableau. CRM Analytics uses drag-and-drop widgets to create dashboards, where best practice in Tableau is to build dashboards within container layouts. Containers and dashboard formatting can seem challenging at first, but once you figure out the drop areas, they are relatively simple.

Below is a screenshot with borders of the containers created so that we can quickly add in the sheets we need to the dashboard containers for creation of the Quota Attainment dashboard.



Conclusion

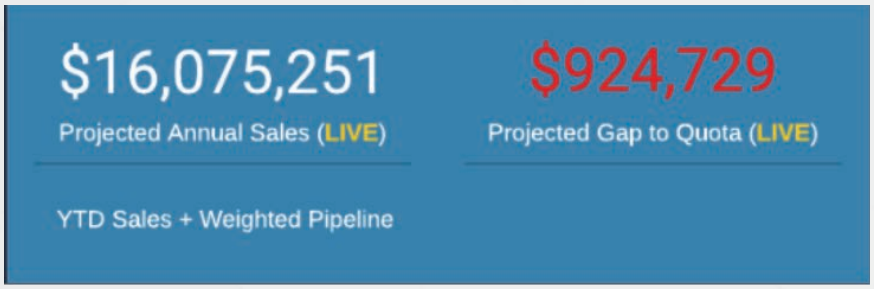
We were able to create all of the static data elements on the dashboard according to the spec using both tools with limited challenges. However, one will notice that more could be done in Tableau using drag-and-drop functionality than was possible in CRM Analytics. To summarize:

- CRM Analytics requires more “coding” knowledge for SAQL but is very easy to format and set up.
- Tableau vizzes can be simpler to build since it has a more powerful visualization layer and more can be done without writing lines of code, but it has more of a learning curve in formatting actions.
- Tableau focuses on as much in “pane” and “shelf” development as possible to limit the amount of coding and allow users to flexibly analyze and build off of their published data sources.

Overall, for the static portions of the dashboard, the process to get from wireframe to functioning dashboard was simpler on Tableau, solidifying its place as the market leader for visualizations.

Comparison #3: Live Data

With most of our dashboard created, we have one more tricky section to deal with: live data. Remember that section of the dashboard that updated immediately after a user made a change to an opportunity amount or forecast category? To refresh your memory, it's this:



While it's all seamlessly on one dashboard (or at least, that is our goal), the approach to work with live data takes a different form.

CRM Analytics

One strong advantage of using CRM Analytics with Salesforce is the ability to write queries directly against Salesforce data. Live data is available through Salesforce Direct queries (point and click) or SOQL queries (custom). In most cases, creating standard Salesforce Direct queries is going to give you what you need to do basic aggregations and filtering.

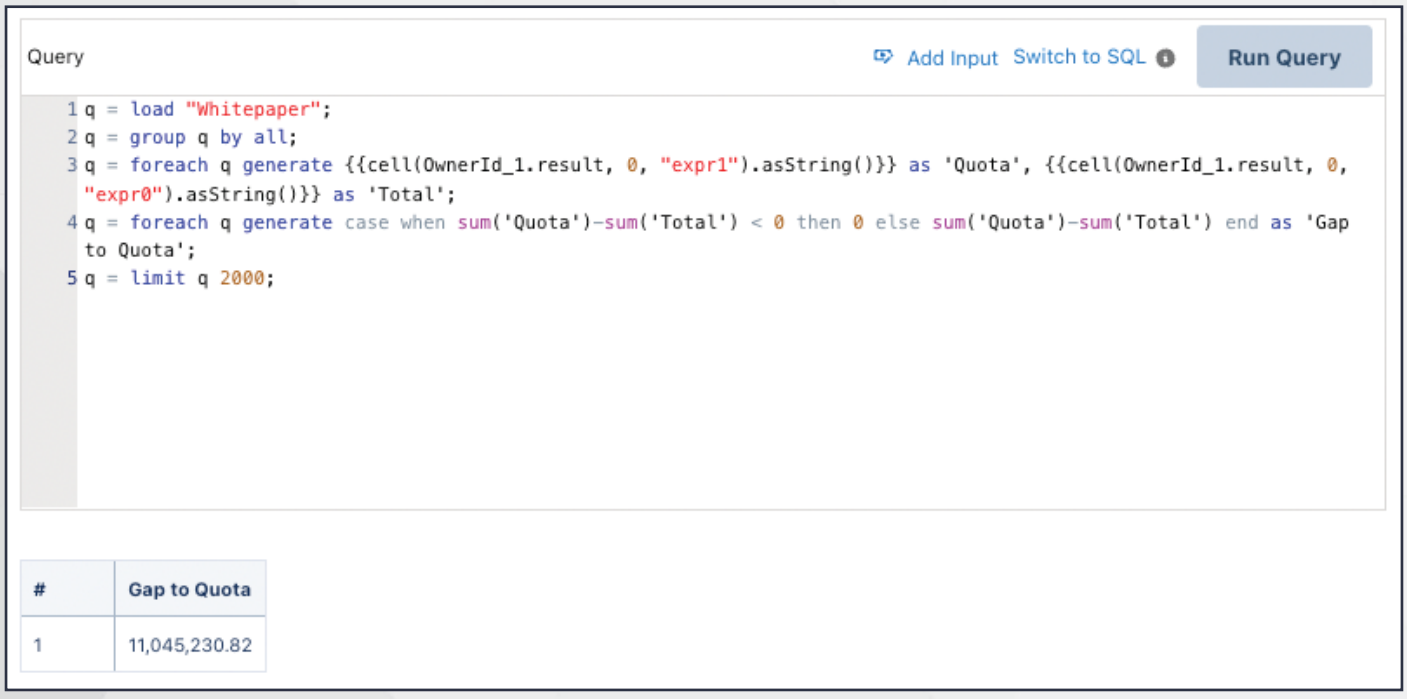
In order to do the calculation for Projected Annual Sales, we need to do a summation of weighted amount (Amount*Probability) on the opportunity level of granularity. Unfortunately, because we can't use arithmetic operators to do calculations in SOQL or reproject and group by all after our first projection to do this calculation, we needed to create a new calculated field in Salesforce that has (Amount*Probability).

After creating this field on the Opportunity object we can now select it in our SOQL query. We also need to get Quota information for our query, except this resides on the User object, and we're querying Opportunity. How do we solve for that?

You'll notice in our query below that this is listed as "Owner.Quota__c." The prefix of "Owner" lets us query data from the User record of the Opportunity Owner — in essence, allowing us to access two Salesforce objects in one single query.

Lastly, we will add a WHERE clause to the query to filter out all of the opportunities that do not have a close date this year, do not have a stage name that is “Closed Lost,” and where the Owner Id is equal to the selection in the Sales Rep filter.

Now that we’ve calculated Projected Annual Sales, we can use the aggregations from the last query we created to do the calculation for Projected Gap to Quota. In order to do this, we’ll use a SAQL query that loads the Whitepaper dataset and then substitutes in the values for Quota and Project Annual Sales as fields. After adding those values in, we will project again and generate the calculation for Projected Gap to Quota.



Tableau

One thing that Tableau cannot yet do is connect to a live Salesforce data feed. Currently, Tableau Desktop only has the ability to connect to Salesforce through extracts. These extracts can be set to update hourly, so while it replicates frequently, there is no real-time data for this specific Salesforce connection.

One thing worth noting: A majority of Tableau data connections do have live data connection options. Given this exists for other data connections, it seems likely that this capability will find its way to Tableau in the near future.

Conclusion

There's only one option at this time to stream data live from Salesforce objects, and it's CRM Analytics. With the evolution of Data Cloud, and the existence of live streams from other Tableau connections, we believe this gap will close, but if you need your Salesforce core objects visualized in real time, CRM Analytics is your choice.

Comparison #4: Embedding and Pre-filtering the Dashboard in Salesforce.com

Our dashboards are built; now it's time to embed them within our Salesforce instance. To fully understand the implications of embedding, we'll analyze the following scenarios for each tool:

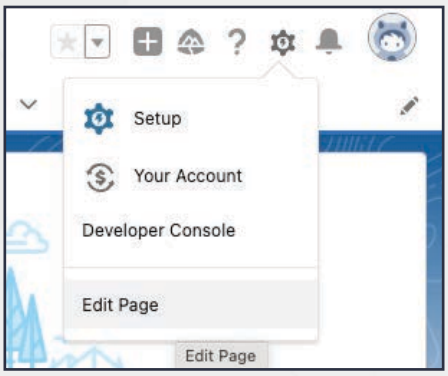
- Embedding
- Filtering to the logged-in user
- Licensing

CRM Analytics

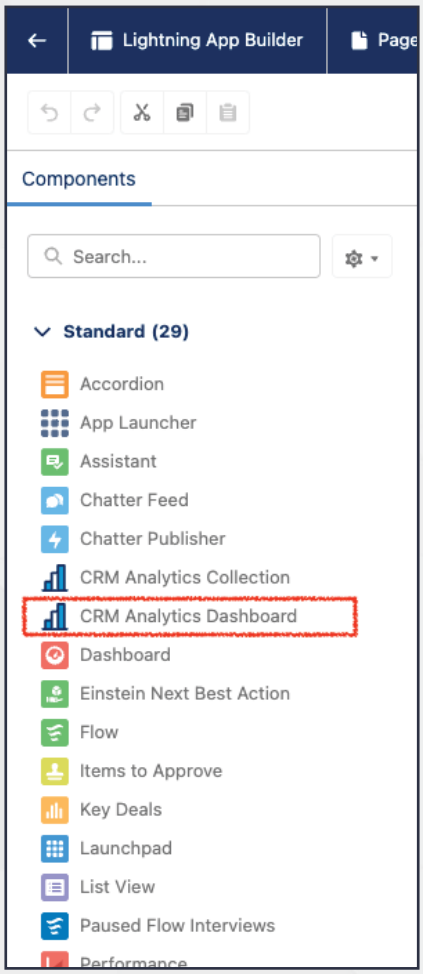
Embedding

One of the advantages of using CRM Analytics is that Salesforce has spent the last several years fully integrating CRM Analytics with Salesforce Lightning. As such, the steps to embed our dashboard on a Lightning page are quite straightforward.

To keep things clean, we create a new tab in Salesforce. After navigating to that tab, we select the gear icon and choose “Edit Page.”



From there, we find a list of Standard Lightning Components that can be added to our Lightning page. And look at that! Right in the list, we have the ability to select “CRM Analytics Dashboard.”



Once we drag the component onto the page, we have a number of pre-built configuration options to choose from as well. This is where we’ll tackle our second task of this section, filtering the dashboard to the logged-in user.

Filtering to the Logged-In User

Filtering the dashboard to the logged-in user is a relatively simple task using one SOQL query and two bindings to the Sales Rep filter. To get started, we created a Salesforce Direct query on our dashboard that selects the Id, Name, and User Role Id from the User object. After creating that, we entered into Query Mode and added in the WHERE statement. This WHERE statement allows you to pull those fields from the user object based on whoever is logged in.

Now that we've set up the live query that tells us all the pertinent information about the logged-in user, we can bind to the Sales Rep filter! In this use case, we only want Sales Reps to see their own data when looking at the dashboard or managers to see the data of all of the roles that roll up to them. To do this, we add in a filter statement to the SAQL query powering the Sales Rep filter. In the filter statement, we'll filter out all selections in the filter that don't match either of the following criteria:

- Opportunity Owner's Id is equal to the Id of the logged-in user.
- The role path for the Opportunity Owner that is calculated in the recipe (OpptyOwner. UserRole.Role_ID_Path) contains the logged-in user's User Role Id.

Query

Switch to SQL | Run Query

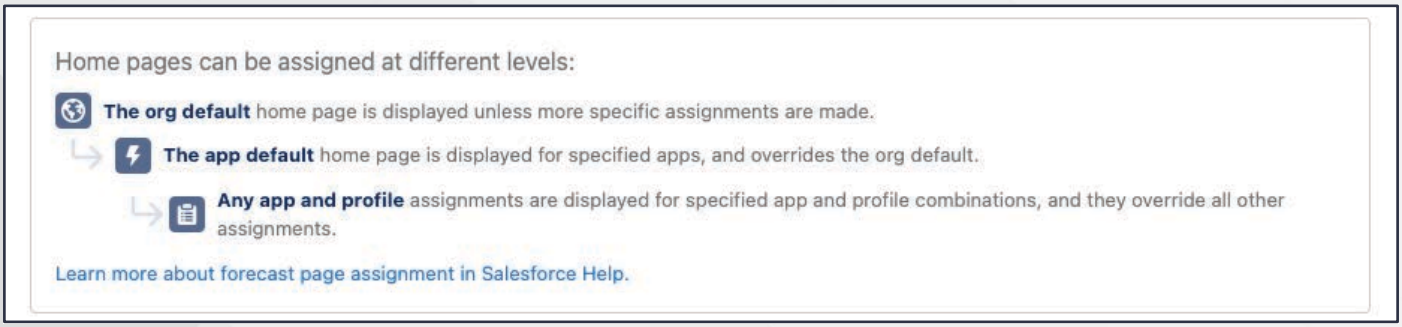
```
q = load "Whitepaper";
q = filter q by 'StageName' == "Closed Won";
q = filter q by date('CloseDate_Year', 'CloseDate_Month', 'CloseDate_Day') in ["current fiscal_year".."current day"];
q = group q by 'OpptyOwner.Name';
q = foreach q generate q.'OpptyOwner.Name' as 'OpptyOwner.Name', sum(q.'Amount') as 'YTD Sales', first(q.'OpptyOwner.Quota__c') as 'Quota', unique(Id) as 'Opp Count';
q = foreach q generate 'OpptyOwner.Name','Opp Count', 'YTD Sales', 'Quota', 'Quota'-'YTD Sales' as 'Gap to Quota', 'YTD Sales'/'Quota' as '% Quota Attainment';
q = order q by 'OpptyOwner.Name' asc;
```

Full Name ↑	Opp Count	YTD Sales	Quota	Gap to Quota	% Quota Attainment
Abby Shafer	3	3,305,097.78	17,500,000	14,194,902.22	0.19
Barbara Del Dotto	2	2,746,571.04	24,500,000	21,753,428.96	0.11
Beth Pryor	3	3,747,455.5	20,500,000	16,752,544.5	0.18
Elyse Reynolds	2	2,373,231.4	20,500,000	18,126,768.6	0.12
Hakeem Turnbull	1	311,774.08	22,500,000	22,188,225.92	0.01
Joanna Donum	1	1,013,244.93	22,500,000	21,486,755.07	0.05
Mike Lamborn	1	100,000	19,500,000	19,400,000	0.01
Randy Dunn	7	7,471,520.36	24,500,000	17,028,479.64	0.3

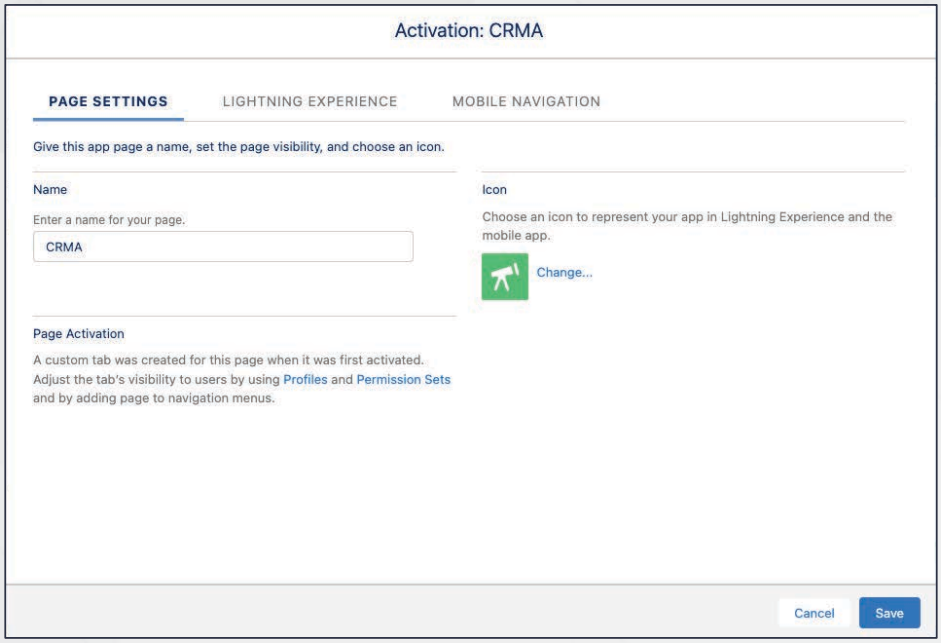
Licensing

While we can configure the dashboard to work for any user, in order to view information on the dashboard, the viewing user must be assigned a CRM Analytics license and permission set. In the event that all users in an organization don't have licenses, the user experience can be managed by only making the dashboard visible to certain users. The Salesforce platform provides extensive options for handling this.

If you were embedding your dashboard on a Home page, Lightning Page modifications can be applied at 3 different levels. All orgs will have a default preset at the org level. However, you can make changes and save those so that they only apply to certain applications, or you can make those changes apply only to certain profiles within each app.

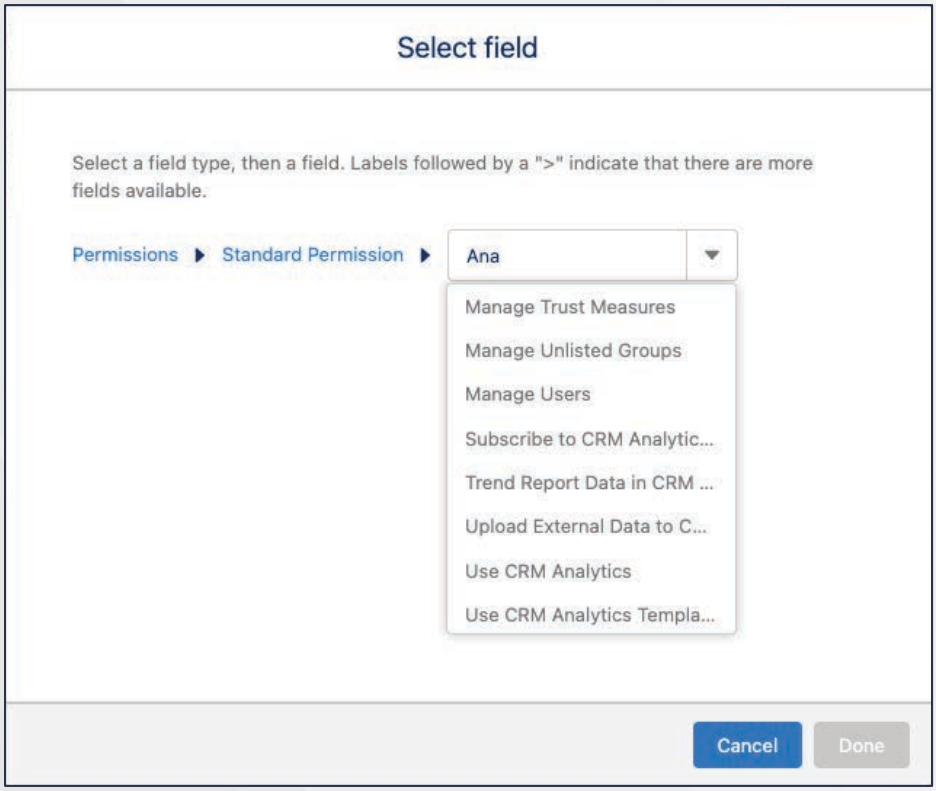


In our example, we chose to embed the dashboard on a "CRMA" page we created in the environment (just so we could quickly toggle between CRM Analytics and Tableau pages in the org). For a "non-Home" page in Salesforce, you have the ability to manage visibility using Profiles and Permission Sets.



What happens if we want to have different types of users see different dashboards on the same page? Salesforce can handle that! There is a section at the bottom of the CRM Analytics Dashboard component configuration panel called “Set Component Visibility.” This allows a given component to be filtered for visibility based on a field on the user record, a specific permission, or a device form factor (e.g., one dashboard for Desktop users; a different dashboard for Phone users).

How might this be applied? For starters, we probably don’t want to show this component to anyone who doesn’t have the CRM Analytics permission. To only show this to users who have been enabled, we’d Edit the Page, click the Add Filter button within Set Component Visibility, and choose Permissions > Standard Permission, and “Use CRM Analytics.”



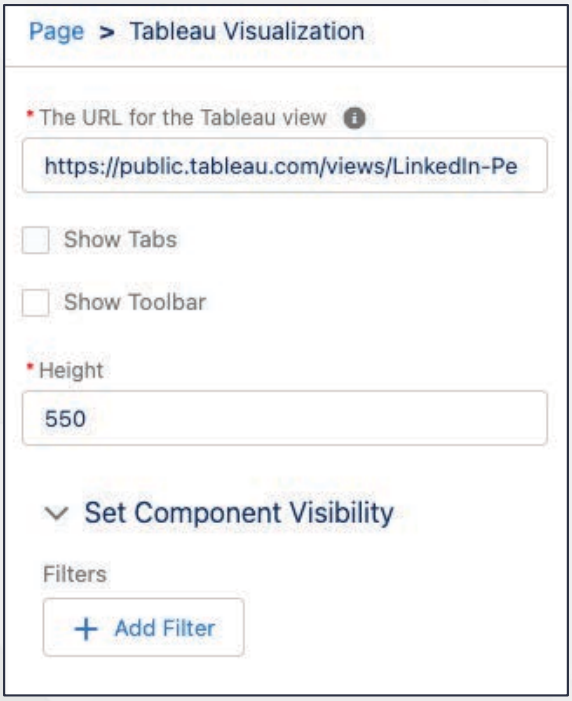
In another example, what if I had two users with the same profile, but who I wanted to see different dashboards? Let's say they were part of different departments, and each department had specific metrics they individually cared about. I could take a similar approach: click the Add Filter button within Set Component Visibility, and choose User > Department. In the dialog box to complete the filter, I'd enter in the name of the Department for that component. For our purposes, let's say it was "Department 1." Then, to show something different for those users not in that department, I'd drag a second CRM Analytics Dashboard component onto the page, and create the binary opposite filter (Not Equal to "Department 1") on that component. The result is that users that are part of Department 1 see the dashboard in the first Lightning Page component; users not part of it will see the second.

Tableau

Like CRM Analytics, we'll want to modify a Lightning Page to embed our Tableau dashboard within Salesforce. Unlike CRM Analytics, Tableau does not have a standard pre-built component. However, one is only a few clicks away.

You'll notice at the bottom of the list of standard components there is a blue backgrounded link that says "Get more on the AppExchange."Clicking that link and searching the Component Store on the AppExchange for "Tableau" will direct you to the Tableau Viz Lightning Web Component. Click the Get It Now button, and install for all users in your organization (or at least those who have permissions to modify Lightning pages.

Like with CRM Analytics, we'll drag this custom component onto our page. When we look at configuration options, though, we're presented with a more limited list:

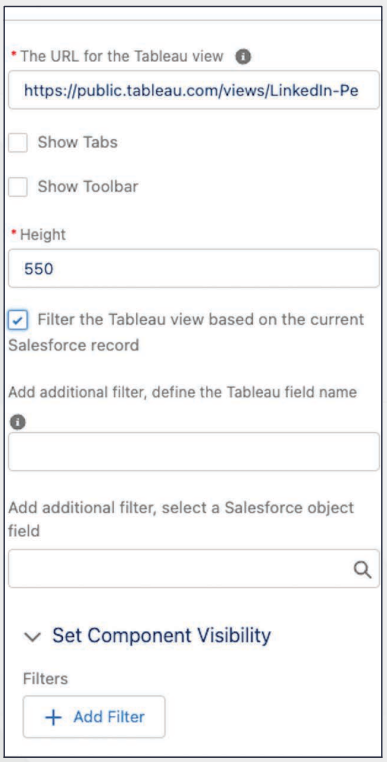


We're assuming you're working on Tableau Cloud or Tableau Server versus Tableau Public for an example like this. Fear not, you can still connect to your viz. To find the URL for a view on Tableau Server or Tableau Cloud, click Share on the toolbar and select Copy Link from the Share View dialog box.

Before we concern ourselves with filtering based on the user, we'll need to first make sure they can access the Tableau dashboard in Salesforce. Tableau has the ability to leverage SSO through Salesforce, so that when a user signs into Salesforce they do not need to reauthenticate to Tableau through another method. It should be noted that there are some restrictions in the SSO method if using Salesforce as your vehicle for authentication (e.g., requires SAML), but in general this can make for a seamless user experience. In Tableau, we will need to make sure that each user (e.g., Sales Rep) in Salesforce has a corresponding Tableau license (viewer at minimum) within our respective Tableau site.

In a case where you'd like the embedded Tableau dashboard to filter based on the logged-in user, Tableau is more limited at this time than CRM Analytics. As you saw above, filtering based on logged-in user requires a direct query against Salesforce standard objects (the User object, to be precise). Because SOQL/direct queries are not available in Tableau, we can't replicate this functionality exactly.

However, Tableau does have the ability to filter the dashboard based on a single record page. For example, if we were to embed the dashboard on the user's profile page, we could restrict the data only to that user. The same would hold true if we wanted insights around a specific opportunity. For use cases where this is sufficient, once you've embedded the Tableau Dashboard onto a record page, you will simply need to enable the context filter (i.e., check the box under "Height"). In order for this simple context filter to work, we made sure to include the User ID field in our Tableau view that corresponds to the record ID in Salesforce.



For more advanced filtering on Tableau or Salesforce fields, you can create those directly in the Lightning editor pane (as seen above in the screenshot). In order to build more advanced filtering you must have a corresponding field or parameter in Tableau that can be mapped to the field in Salesforce.

In cases where we need to limit access to the data based on roles, we could build row-level security directly in Tableau so that we can filter the dashboard so that only the user can see their data (which would be perpetuated in the embedded view in Salesforce). With live connections, this can be done by simply going to the Server on the menu bar and selecting "Create User Filter." You can then manually set up filters based on multiple fields. Since we do not have a live connection, in order to create user-based filtering, we would create an entitlements (i.e., lookup) table that maps user information and security groups (e.g., by manager). We then create a calculated field to filter off based on the information in the security table. That will allow for the appropriate user filtering. Now, with SSO setup in Salesforce and Tableau, and the user filter added, when a logged in Salesforce user accesses the embedded Tableau dashboard, the view will be filtered to see their data as referenced by the user filter.

Conclusion

Because it's natively integrated, you have more options for how to handle the embedded CRM Analytics dashboard, and there is no configuration for SSO since the user is already logged in to both platforms through their Salesforce ID. CRM Analytics gets the nod here, since in a CRM system like Salesforce the most common use case will require insights in context of the logged-in user. However, considering that you're a simple filter click away from achieving essentially the same result in Tableau, the advantage is somewhat minimal.

Comparison #5: Actionability

The final item to compare is the ability to trigger Salesforce actions from within the embedded dashboard. When we're designing visualizations, we're asking three primary questions:

1. What questions does the user most want to answer?
2. What is the order of importance of those questions?
3. What actions are they going to take once they have the answers to those questions?

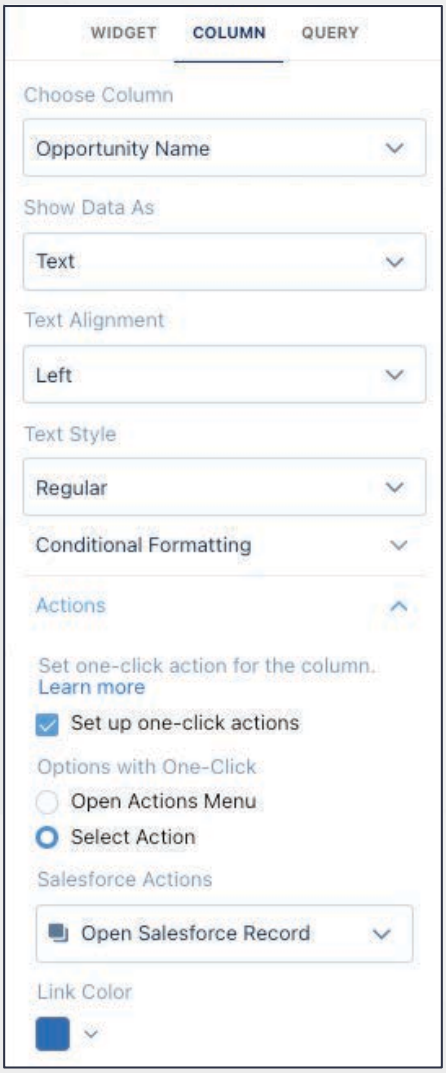
If we create a beautiful dashboard but we forget the last piece (enablement of those actions), we've missed a big chunk of what takes a dashboard from good to great. For the purposes of this example, we chose the simplest action of all: opening an Opportunity record in Salesforce.

CRM Analytics

Adding actions to dashboards and lenses in CRM Analytics is incredibly easy! There are two ways to configure standard actions on a CRM Analytics dashboard: the first is to configure your actions directly on the widget itself and the second is to configure actions directly on the dataset. In this instance, we added the actions directly to the dataset which will let us deploy the selected actions across all of our dashboards that use this dataset.

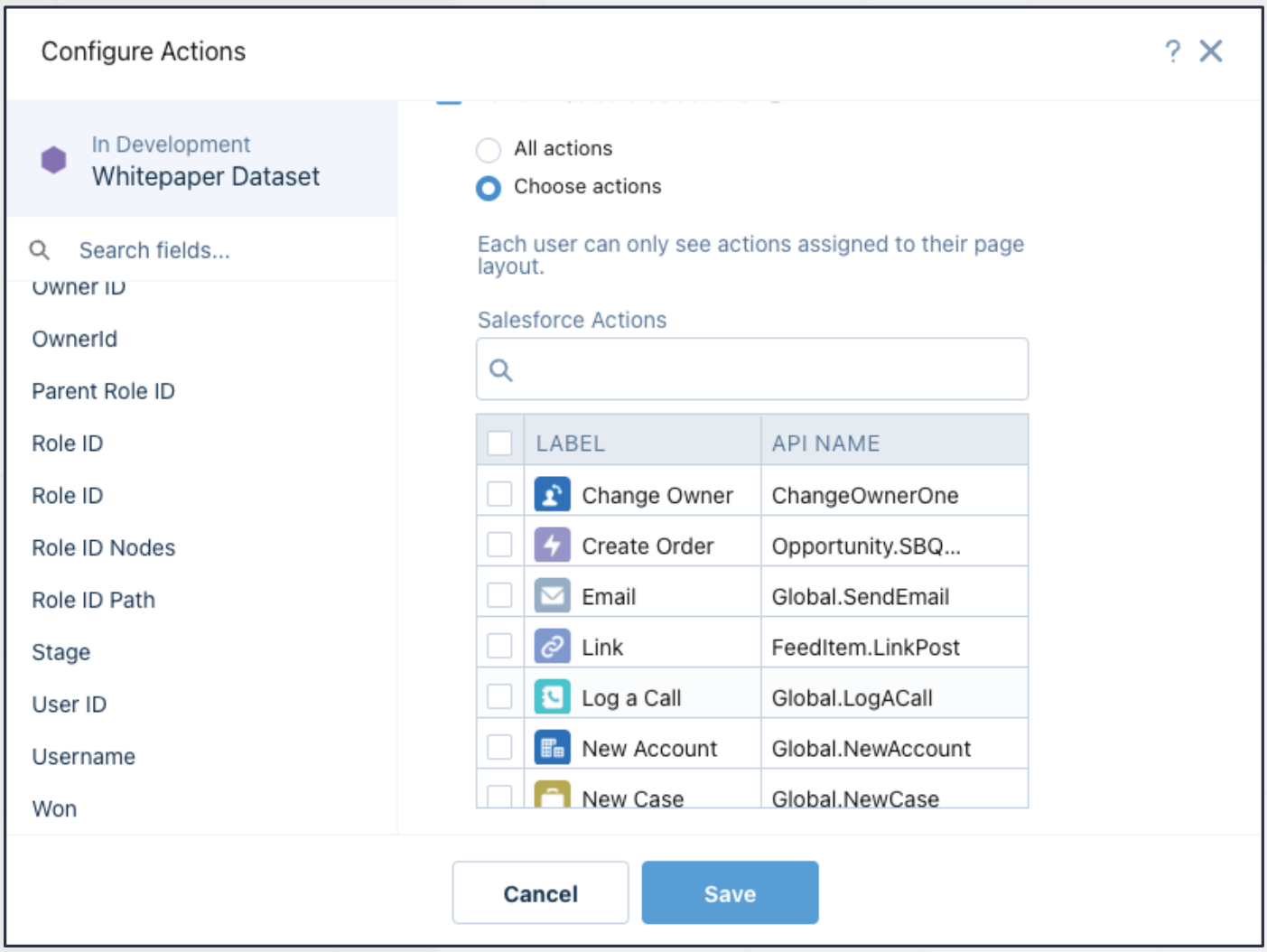
To set up actions directly on your dataset, navigate to the analytics studio and edit the dataset. Click the Configure Actions button. In the left pane, select the field you want to configure actions on. In this case, we'll use the Opportunity Id field in the dataset as the Record ID Field and the Opportunity Name as the Display Field and select the box for Open Salesforce record.

To make these actions even more accessible, we'll also set up one-click actions on our tables. To complete this step, navigate to the dashboard, select a table, navigate to the column tab on the right pane, choose the opportunity name column, and configure the one-click actions:



And just like that, we've configured actions that will show up on our tables as hyperlinks!

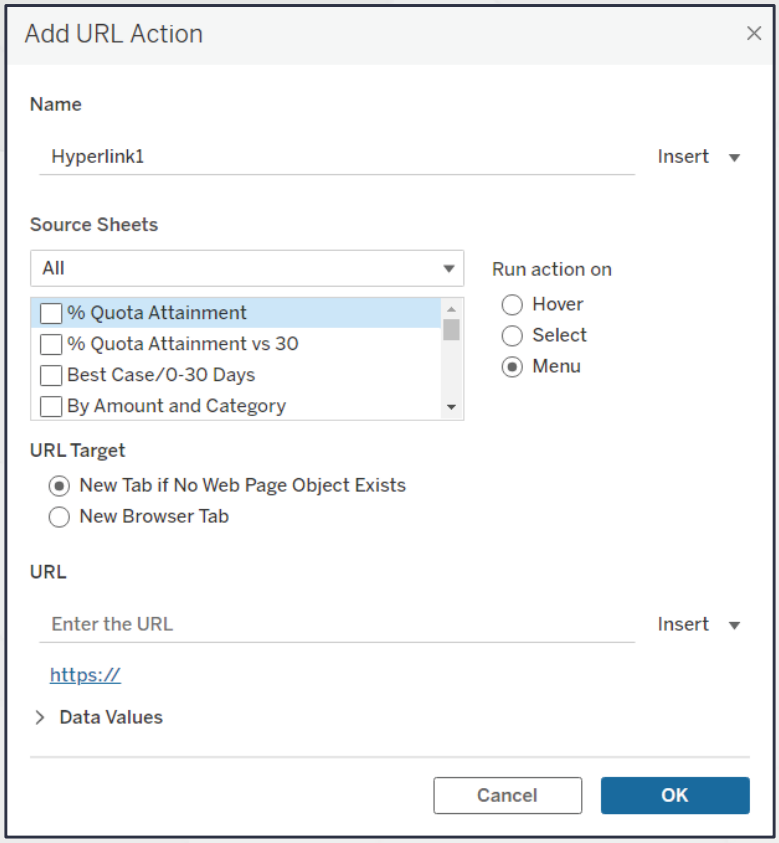
Note that you can preset a specific action, or you can open the entire action menu. Salesforce has 15+ prebuilt actions to leverage, plus it's also easy to create new actions, both through configuration (nothing but clicks!) and, if something significantly complex is desired, through code. What that allows for is not just better insights for system users, but also enhanced efficiency for those users, and increased usefulness and adoption for those dashboards.



Tableau

Tableau offers many of the same actionable features CRM Analytics offers, such as creating "actions" and "workflows" directly from Tableau workbooks (e.g., going to the opportunity page and launching a Salesforce Flow).

Setting up a "Go to Opportunity" action in Tableau is simple. We started by creating an action on the worksheet, by selecting URL Action, in the worksheet drop-down on the toolbar as seen in the screenshot below:



From there we select what source sheets contain the action and how to run the action. You can run actions via three options: Hover, Select, and Menu. Hover is good for highlighting certain data. Select is good for changing parameters, which is what we did earlier to change forecast categories. Menu actions work particularly well for URL actions because they allow the user to click the information, review it, and choose to go to the URL rather than immediately being redirected on click.

The key to making URLs dynamically change is in understanding the structure of URLs that are being linked. For example, a Salesforce Opportunity page is structured as follows: https://[Salesforce org name]/lightning/r/Opportunity/[Opportunity ID]/view. Start by copying an example into the URL section, then edit it based on what will change. Because the org name will not change, we can leave it hard coded. However, the Opportunity ID will change based on what Opportunity is selected. First delete the Opportunity ID section of the URL, then select “Insert” and choose the Opportunity ID field. One of the requirements is that the referenced field is included in the view.

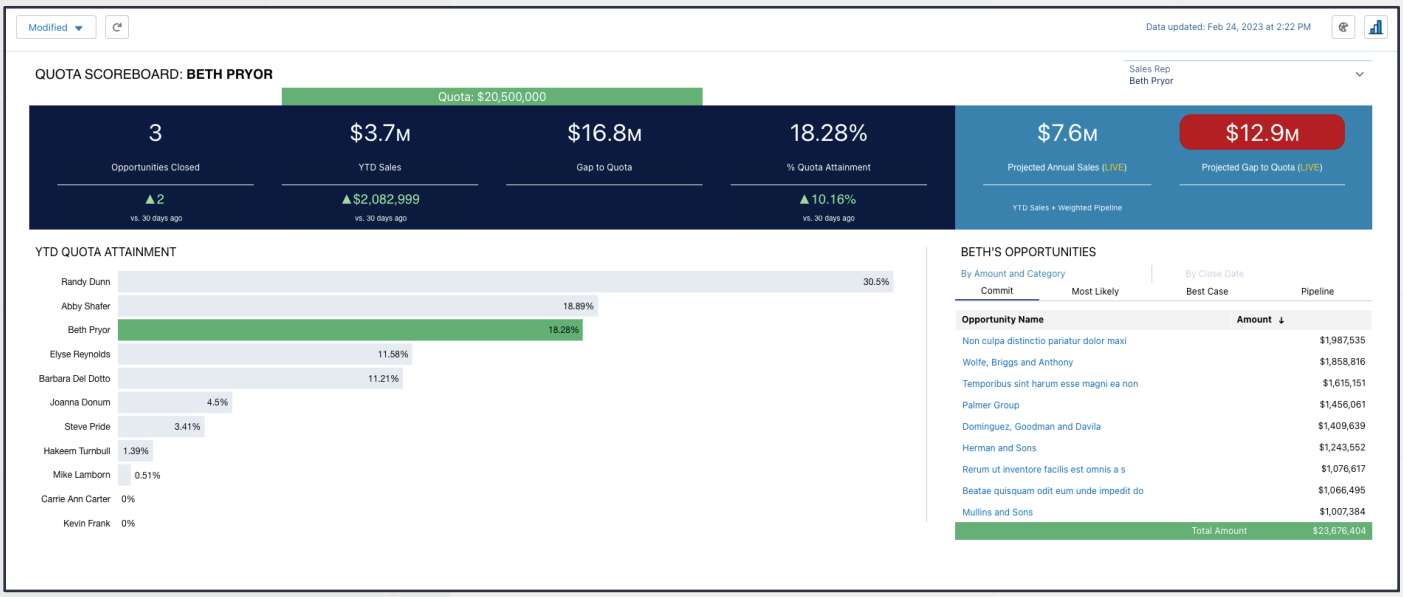
As captured above, CRM Analytics allows for updating/creating Salesforce records relatively simply through point-and-click setup. Tableau also allows for updating/creating Salesforce records, albeit through a few extra steps. Tableau requires the use of auto-launched Salesforce Flows, which must be configured inside of Salesforce before adding the functionality to the dashboard. The external actions using flows work by setting up a Salesforce Flow, dragging the “Workflow” object onto the dashboard (from the object pane), and selecting the Flow that was created. Once the workflow has been configured, you can change the appearance of the workflow button, make a selection, and click the button. More detail on some of the advances in Salesforce Actionability can be found here from Tableau.

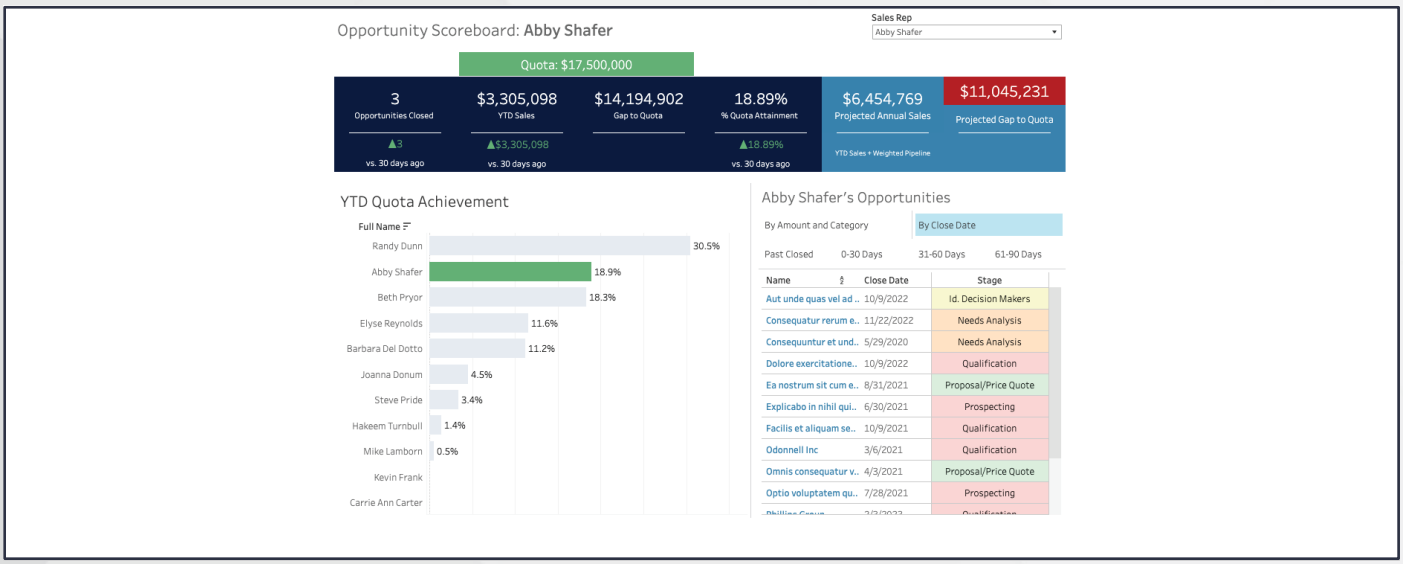
Conclusion

Like the process of embedding, actionability is just a bit easier and more natively part of CRM Analytics than it is with Tableau. However, as was the case on embedding, with a little extra effort and code, one can accomplish similar results in Tableau — most specifically through the recent enhancement to allow external actions on a Tableau dashboard, opening the window for Salesforce Flows to be called. While this requires having someone on hand who knows how to create Flows (versus some of the standard actions in Salesforce), in general the gap between the products is relatively small.

The Benefits of Having the Right Partner Help You with Your Analytics Tooling

Look at those beautiful, embedded, actionable dashboards!





What Did We Learn?

Compared to where things were at the time of the Tableau acquisition, several areas where Tableau lagged behind CRM Analytics have been reduced. The most significant factor at the moment is access to real-time data and querying the Salesforce objects directly. CRM Analytics can do it; Tableau cannot. However, with the launch of Data Cloud and some of the focus on Tableau's role with that product, that gap may be closed swiftly. In the other areas that we compared, there were no significant holes in either tool. Tableau won for ease with which some complex visualizations were created, while CRM Analytics won for some of its more tightly coupled integration with Salesforce around embedding, filtering, and interacting with insights.

CRM Analytics and Tableau: Which Should You Choose?

As we stated at the beginning, it is rarely a "one tool fits all" situation for the vast majority of organizations. You should select the right product based on your specific use case, and for use cases within Salesforce, both tools have scenarios where they can be argued to be the preferred choice.

In general, CRM Analytics is a purpose-built tool for use cases in Salesforce, and with the exception of requiring additional code to build some dashboard features that can be done in Tableau with clicks, it is the safest choice for maximum functionality for those Salesforce use cases. But if CRM Analytics is not an option, or you want to test drive Tableau in Salesforce to see if it can work, we demonstrated that the gap between the products has narrowed, and there are only a few limited scenarios that can't be accomplished using Tableau.

At Atrium, we live and breathe this stuff. It helps to have an expert partner that knows the ins and outs (not to mention the ongoing innovation, product updates, and enhancements) of both CRM Analytics and Tableau. We can help you understand the best approach for your organization or use case — and ensure you get the hands-on training and outcomes you need to reap the benefits of powerful analytics.

About Atrium

Atrium is a consulting services partner that helps advise, implement, optimize, and manage analytics, AI, and machine learning solutions. We combine industry expertise, leading cloud platforms, and data science to help our customers unlock the power of their data and solve their most important challenges. [Learn more at atrium.ai.](atrium.ai)

info@atrium.ai