

Creating a View

- [Overview](#)
- [Create](#)
- [Entity Relationship](#)
- [Selecting Fields](#)
- [Field Categories & Meta Data](#)
- [Field Format](#)
- [Calculated Fields](#)
 - [Calculated Metric](#)
 - [Date Hierarchy Fields](#)
- [Drill Down Hierarchy](#)
- [View Summary & Saving](#)
- [Further Information](#)

Overview

[top](#)
The View is a metadata layer used by Yellowfin to hide the complexity of database structures from Report Writers. The View is used to define which columns in your database you wish to make available for Report Building. These fields may come from multiple tables and therefore will require joins to be defined (the business logic that links rows in a table together).

The two major steps in creating a view include:

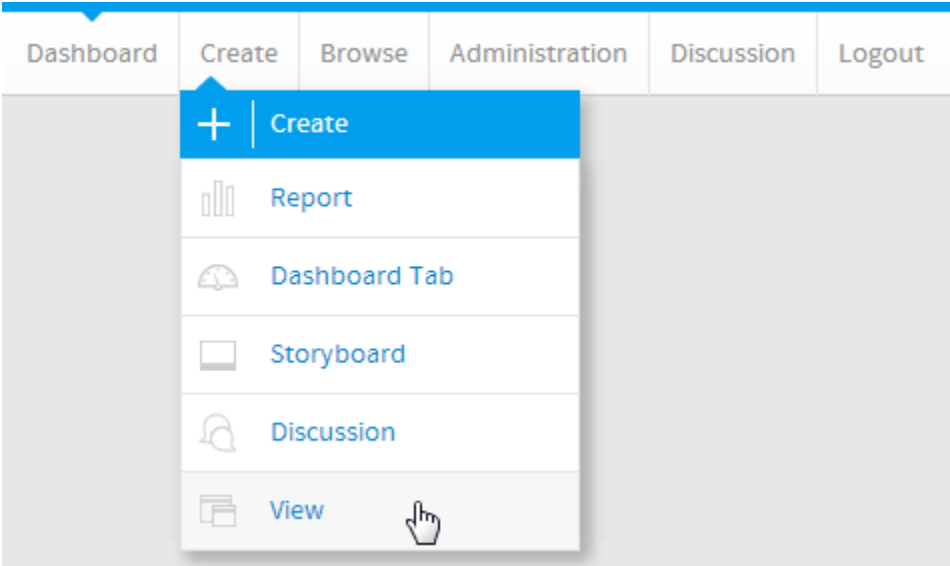
1. **Relationship Entity Diagram** - selecting the tables you need from your database and defining how data in these tables are joined
2. **View Field Selection** - defining which fields you wish to make available from these tables and providing metadata for them.

See [Views](#) for more information.

Create

[top](#)
This section will walk through the initial steps of creating the view, including selecting the data source connection to base it on and the builder type to use.

1. Click the **Create** link in the main navigation bar of the system, and select the **View** option.
Alternatively you can click **Add** from the **Views** section of the **Admin Console**.



2. Select the following options:

Author Type: **Drag and Drop Editor**
Data Source: **Ski Team**
Category List: **Ski Team**

3. Click the Next button to continue.

View Options
▼ View Options
Name:
<input type="text" value="Tutorial View"/>
Business Description:
<input type="text" value="This will be used to learn about Yellowfin."/>
Technical Description:
<input type="text"/>
Audience:
<input type="text"/>
View Category:
<input type="text" value="Ski Team"/>

Entity Relationship

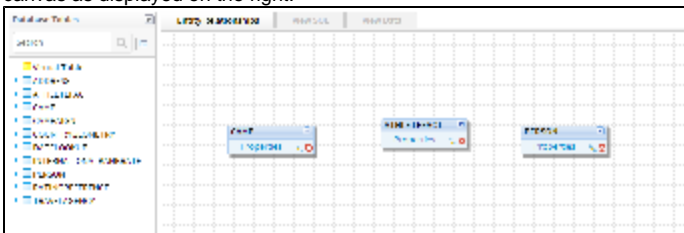
[top](#)

The Entity Relationship is one of the key components of the view builder. This allows you to define all the key relationships between your selected database tables.

1. From the table list on the left of the screen, drag the following tables onto your canvas:

AthleteFact, **Person**, and **Camp**

You should now have three tables on your canvas as displayed on the right.



2. On the AthleteFact table click the join link. This will open the join pop-up. Create a join between the AthleteFact Table and the Person Table.

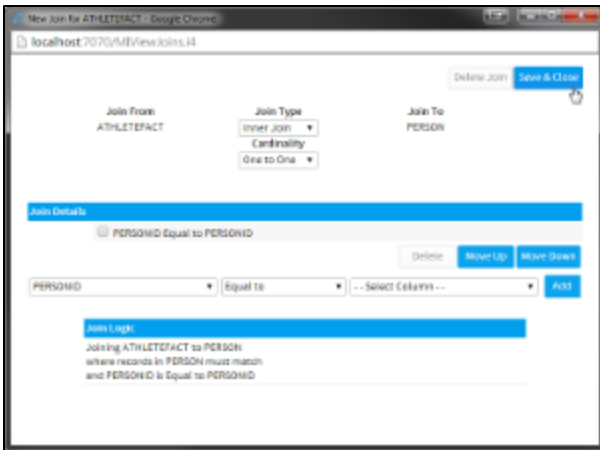
Join From: **AthleteFact**

Join Type: **Inner Join**

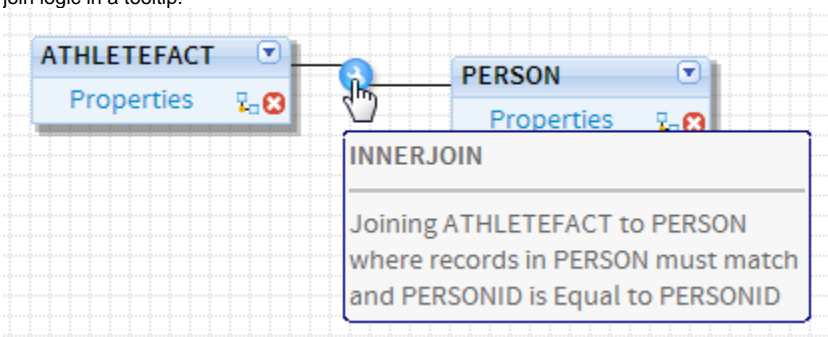
Join To: **Person**

Join logic: **PersonID Equal to PersonID**

3. Click the **Add** button to add to the join list. You should now see the join logic as depicted on the right.

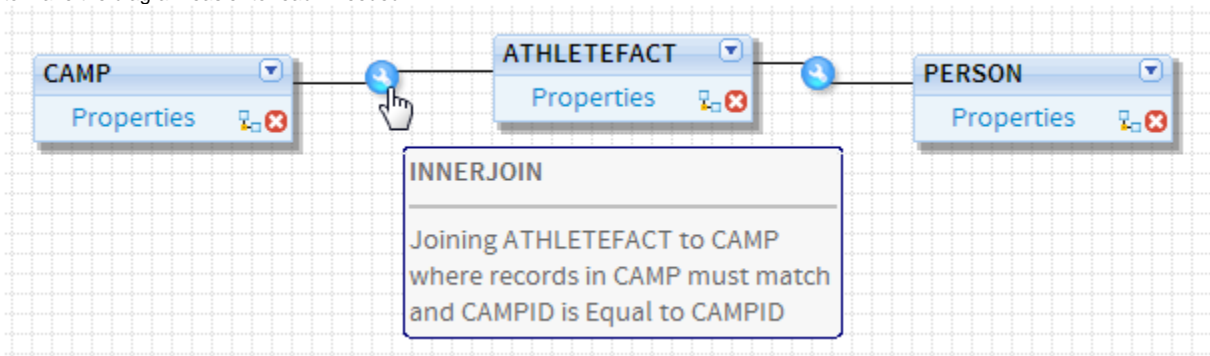


4. Click the **Save & Close** button to save your join. The join will now be displayed as a line between your AthleteFact and Person tables. Hovering over the join icon will display the join logic in a tooltip.



5. Create another **Inner Join** from **AthleteFact** to **Camp** where **CampID = CampID**

You can move your tables around the canvas to make the diagram easier to read if needed.



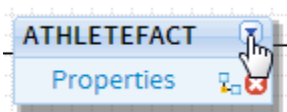
See [Drag & Drop View Builder](#) for more information.

Selecting Fields

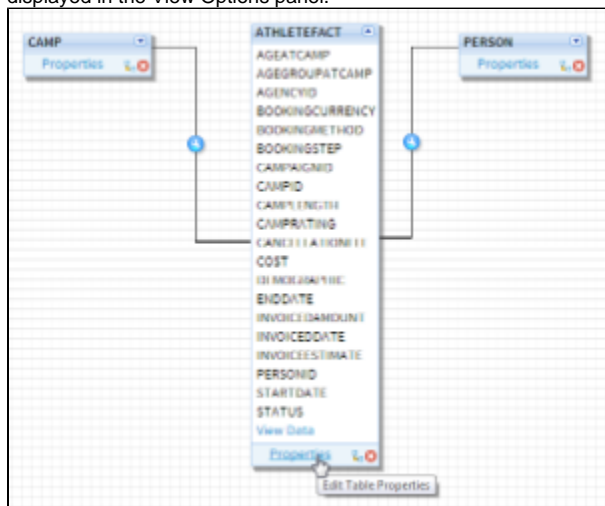
[top](#)

Select fields that you wish to make available to your end users for reporting. Only columns selected from each table in your Unattached list will be available for reporting.

1. Click on the expand icon next to a table name to expand it.

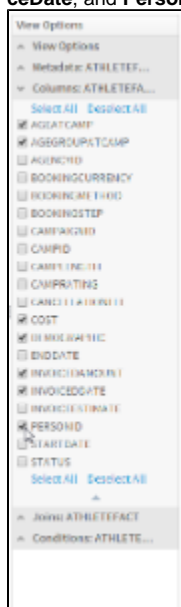


2. Click the **Properties** link on the AthleteFact table. The table properties will now be displayed in the View Options panel.



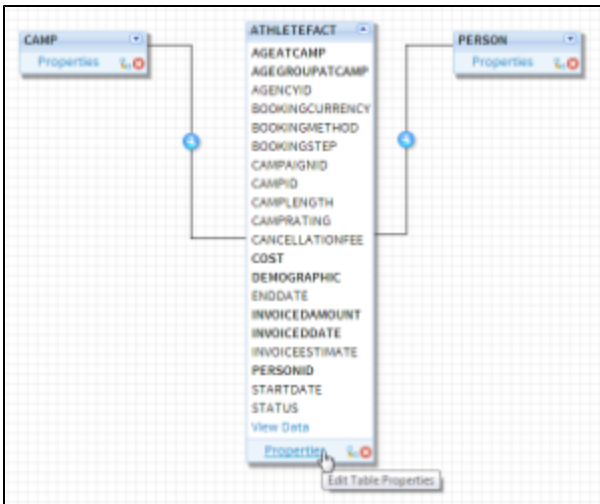
3. Click the Columns section link to open the options. A set of columns from the AthleteFact table will be displayed.

Select the **AgeAtCamp**, **AgeGroupAtCamp**, **Cost**, **Demographic**, **InvoiceEstimate**, **InvoiceDate**, and **PersonID** Columns.



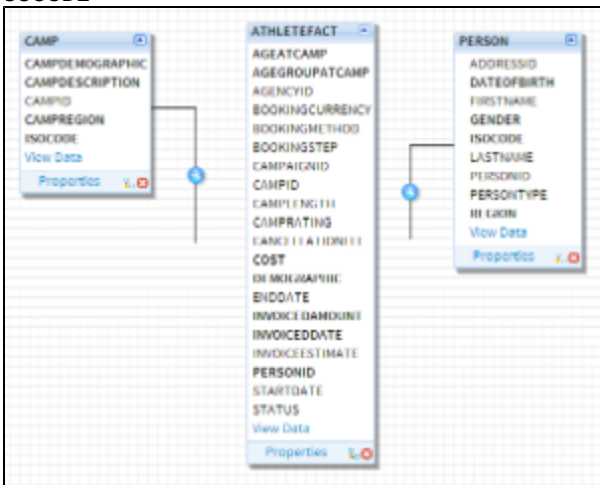
4. Once you have selected these, click on the Properties link again to update your diagram.

5. The columns you selected should now appear in **bold** on your table (as pictured).



6. Replicate the last step for each table.

Camp: **CampDemographic**, **CampDescription**, **CampRegion**, and **ISOCODE**
 Person: **DateOfBirth**, **Gender**, **Region**, and **ISOCODE**



7. Click on **Step 2** in the navigation bar to continue to the **View Fields** page.



See [Table Properties](#) for more information.

Field Categories & Meta Data

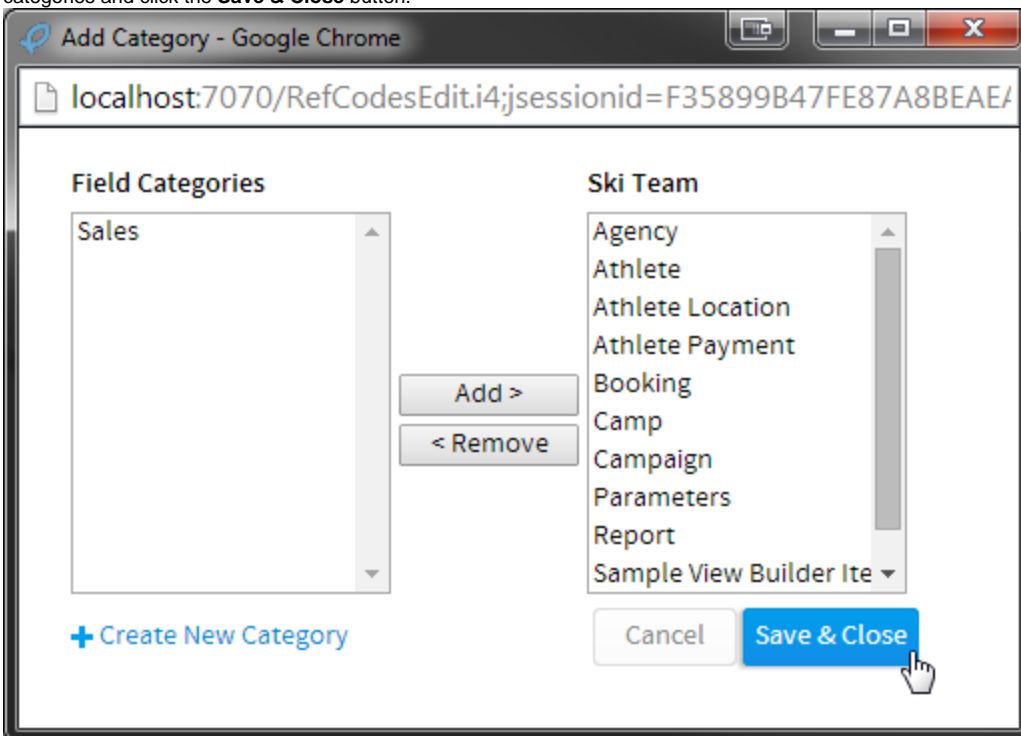
Initially, fields selected from the previous step will all be in the **Unattached** panel in folders that represent the tables that they originated from. These fields have not had meta data associated with them and **cannot** be used by your report writers. You must assign fields to categories (folders) in the **Available Fields** panel. The reason you do this is to organise you fields in a way that is logical for the Report Writer, giving you the chance to group them differently than the table structure in the database.



1. First of all, make sure you have the categories you want to use to divide your fields. Click on the **Edit Categories** link in the Available Fields section.



2. Add the **Athlete**, **Camp**, **Payment**, and **Time** categories and click the **Save & Close** button.



3. Click the expand arrow next to the folder names to see the columns you have selected from each table.

4. From the AthleteFact table either select and drag the fields individually or use shift to click on each of the **AgeAtCamp**, **AgeGroup AtCamp**, and **Demographic** fields and drag these into the **Athlete** folder.

Unattached

Search

+

-

ATHLETEFACT

AGEATCAMP

AGEGROUPATCAMP

COST

DEMOGRAPHIC

INVOICEDAMOUNT

INVOICEDDATE

PERSONID

CAMP

PERSON

Calculated Fields

Available Fields

Agency

Athlete

3 rows selected

Athlete Location

Athlete Payment

Booking

Camp

Campaign

Parameters

Report

Sample View Builder Items

Time

Edit Categories

On completion you will note that the column names are now in bold indicating that they have been added to the view.

Unattached

Search

+

-

ATHLETEFACT

AGEATCAMP

AGEGROUPATCAMP

COST

DEMOGRAPHIC

INVOICEDAMOUNT

INVOICEDDATE

PERSONID

CAMP

PERSON

Calculated Fields

Available Fields

Agency

Athlete

AGEATCAMP

AGEGROUPATCAMP

DEMOGRAPHIC

Athlete Location

Athlete Payment

Booking

Camp

Campaign

Parameters

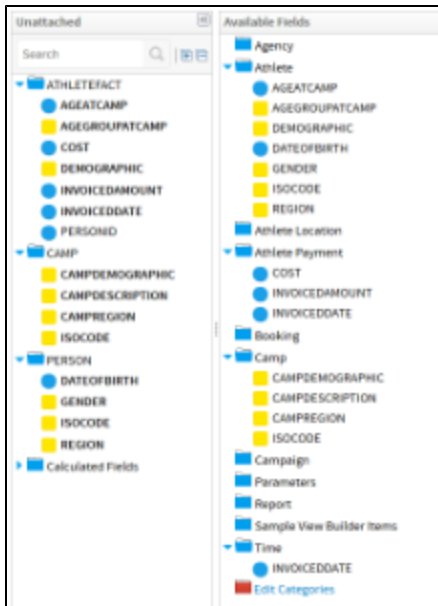
Report

Sample View Builder Items

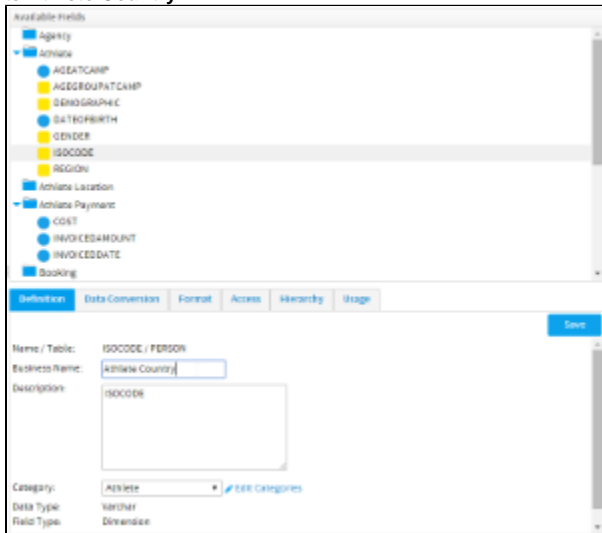
Time

Edit Categories

5. Now follow the same steps as above and put all the fields in their correct categories. **Cost**, **InvoicedAmount**, and **InvoicedDate** fields > **Payment** category. **InvoiceDate** field > **Time** category. **Camp** fields > **Camp** category. **Person** fields > **Athlete** category.



6. To update the field name to provide a more user friendly name – double click the **ISOCODE** field. Change the business name of the field to **Athlete Country**.



7. This field will use an Org Ref Code to convert ISO Country Codes to their respective names. To set this up, click on the Format tab.

8. Set the Format option to **Org Ref Code**. In this case you will use an existing type that we have set up for a sample.

See [Organisation Reference Codes](#) for more information.

9. Select **Country** from the drop down menu.

10. Click on **Save** button to commit your changes.

11. The ISOCODE column will now be updated in the Athlete folder. This method is used to provide business terms for all your attributes.

See [View Fields](#) for more information.

Field Format

[top](#)

The format options will provide the default for how the field will be used on the report. A user will still be able to change a format for a specific report.

1. Double click on the **Cost** field to open it in the edit window.
2. Click on the **Format** Tab
3. Add a prefix of **\$** and set Decimal Places to **0**
4. Select **SUM** as the default aggregation.
5. Apply a colour for chart display if required.
6. Click on **Save** button to commit your changes.

See [Field Format](#) for more information.

Calculated Fields

[top](#)

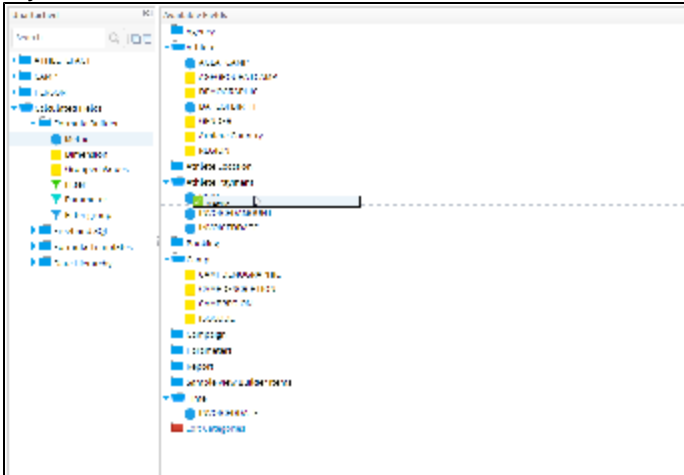
In addition to fields from your database you can create calculated fields, pre-defined filters, and date hierarchy fields.

Calculated Metric

This type of calculated field allows you to build a calculation that will return a numeric value as the result. In this example we will aim to calculate profit by subtracting cost from invoice figures.

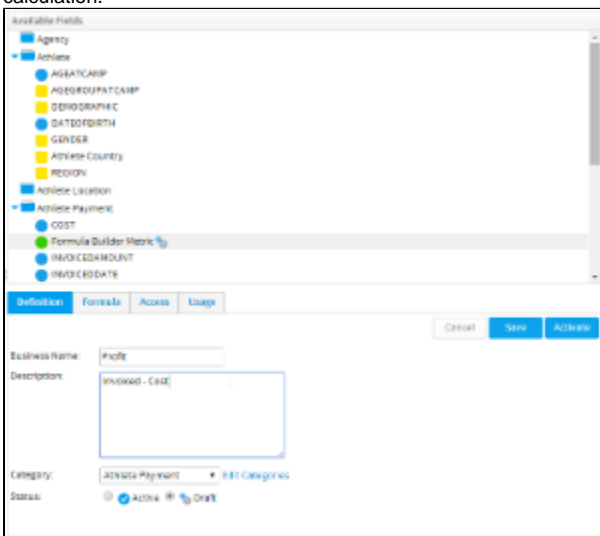
1. In the **Unattached** panel open the Calculated Fields folder and then the Formula Builder folder.

2. Drag the **Metric** field into the **Athlete Payment** folder.



3. You will now see the Definition tab. Set the Business Name to **Profit** and enter the calculation logic **Invoiced - Cost** into the Description - this is useful for users that want to know the formula used in a calculation.

4. Click the Formula tab to define the calculation.



5. From the Select Field drop down choose **INVOICEDAMOUNT** and click the **+ Add** button next to it to build it into the calculation.



6. Now click the - (minus) button directly below the Select Field drop down.

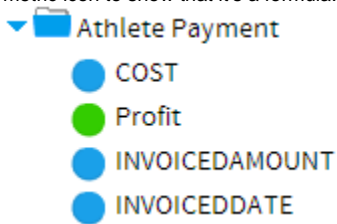
7. Select the **COST** field and click the **+ Add** button to finish this simple calculation.

8. Click the **Click here to test this formula** link in order to let Yellowfin validate your calculation. You should see a **Formula Successful** message displayed above the builder if successful.

9. Click **Activate** to save the field and make it available for use in reports.



11. You will now see the **Profit** calculated field in the **Athlete Payment** category and it will have a green icon instead of the usual metric icon to show that it's a formula.



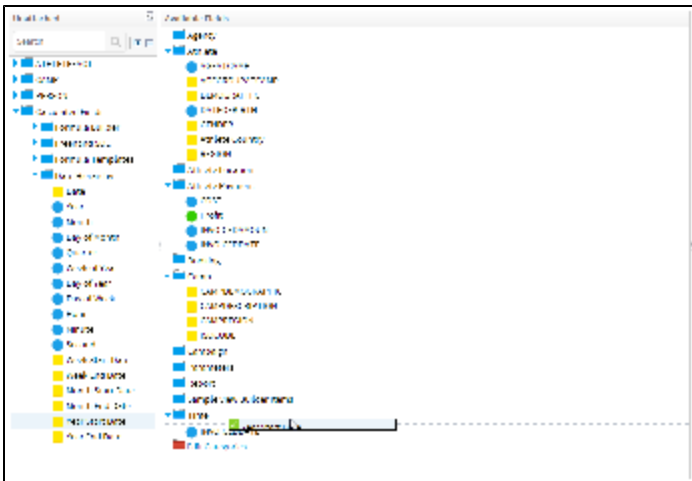
See [View Calculated Fields](#) for more information.

Date Hierarchy Fields

Date Hierarchy calculated fields allow you to build levels of a hierarchy based on a single date field in your database. This can then be used to define Drill Down hierarchies, or for other purposes in reports.

1. First you will need to ensure you have a date field to use with the hierarchy templates. We've already added the InvoiceDate field into the Time folder, so this has been taken care of.

2. Drag each of the levels one by one into the same folder as your date field - this is important for when you build the hierarchy later. Start with the **Year Start Date**. The reason we're using the Year Start Date is so that the field is still a date format, even though we can change the display to be just the Year component. This means we can use it for Time Series charts and other date related functionality.



3. Rename the field to **Year** and the Description to **Invoiced Year**. This lets the user know that the field is based on the Invoiced Date, avoiding incorrect use.

4. Set the field type to **Dimension** on the Definition tab - the field has to be a dimension in order to use it in a Drill Down hierarchy as metrics are not deemed compatible.

5. Set the field to base the Year on to **INVOICEDDATE**, found at the very bottom of the Definition tab.

6. **Activate** the field.

7. Repeat the same process, this time creating a **Month** field based on Month Start Date of Invoiced Date.

Definition	Format	Access	Hierarchy	Usage
<div style="text-align: right;"> Save Activate </div> <div style="border: 1px solid #ccc; height: 100px; margin-bottom: 10px;"></div> <div> <p>Category: Time Edit Categories</p> <p>Status: <input checked="" type="radio"/> Active <input type="radio"/> Draft</p> <p>Data Type: Date</p> <p>Field Type: <input checked="" type="radio"/> Dimension <input type="radio"/> Metric</p> <p>Please select the date field to base this field on.</p> <p>Field: INVOICEDDATE</p> </div>				

8. Double click your INVOICEDDATE field in order to rename it to **Invoiced Date** and change the field type to **Dimension** in preparation for building the hierarchy.

9. **Activate** the field.

Definition	Format	Access	Hierarchy	Usage
Save				
Name / Table:	INVOICEDDATE / ATHLETEFACT			
Business Name:	<input type="text" value="Invoiced Date"/>			
Description:	<div>INVOICEDDATE</div>			
Category:	Time Edit Categories			
Data Type:	Date			
Field Type:	<input checked="" type="radio"/> Dimension			

Drill Down Hierarchy

[top](#)

The hierarchy allows report users to drill down a dimensional hierarchy by limiting the result set as they select one level to the next. For example drill from Year (2014) to Month (August) etc.

When creating the hierarchy, you need to start from the top level and work your way down. In this example we are creating a Year > Month > Date hierarchy, so we will start with the Year field at the top.

1. Double click the **Year** and navigate to the **Hierarchy** tab.

2. Enable **Drill Down** and select the **Month** field as the **Drill To** option. **Activate** the field.

Definition	Format	Access	Hierarchy	Usage
Save Activate				
Drill Down:	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No			
Drill To:	<div> <div>++ Select Field ++</div> <div> <div>Select Month</div> <div>Month Start Date</div> <div>Invoiced Date</div> </div> </div>			

You will now notice that there is a link between the Year and Month fields. This lets you know there is a hierarchy link defined between the two fields.



3. Next repeat the process by double clicking on the **Month** field, navigating to the **Hierarchy** tab, enabling **Drill Down**, and specifying the **Invoiced Date** field as the Drill To option.

4. **Activate** the field.

5. You will now see there is a 3 level hierarchy defined. You won't have to define Drill Down options on the bottom level (Invoiced Date).



See [Drill Down Hierarchies](#) for more information.

View Summary & Saving

[top](#)

1. From the View Fields page click the last step to continue to the Summary page.



2. You can explore the tabs to see example data and the SQL that Yellowfin generates.

3. Click the **Activate** button to save your view and activate it for use.

4. You will now see on the general tab that your Tutorial view is **Active**. It is now ready to report off.

5. Click **Close** to close the view and return to your view list.

See [View Summary](#) for more information.

Further Information

[top](#)

For more information around the creation of Views in Yellowfin see the [Views](#) section of the wiki.

[top](#)