View Creation

Overview

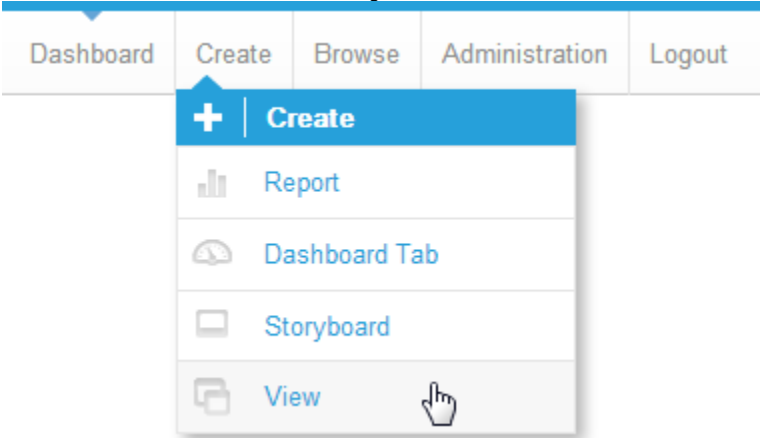top

Views within Yellowfin provide the link between users, their reports, and the database. The purpose of the view is to simplify the knowledge required by end users of the source database. This section describes some of the basic view management options available to you.

1. To create a new view:
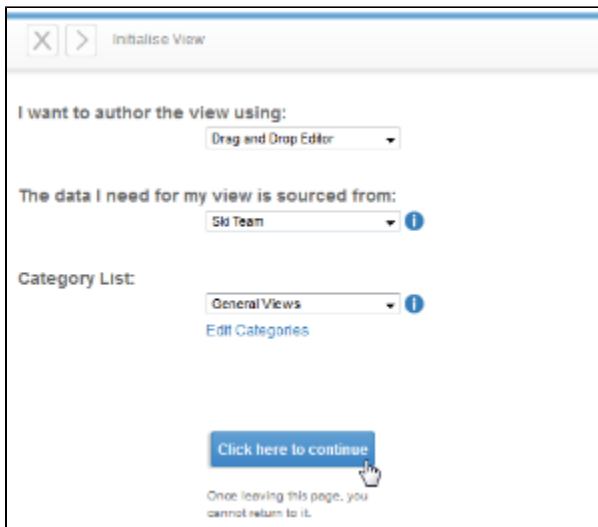   a. click the add view link from within the source record, OR

    b. click on the Create menu in the main navigation bar and select View

The view initiate page will prompt a select of the type of view you wish to create and the data source connection to use. The choices you have are:

Builder	Description
Drag and Drop	Use the Yellowfin builder to create a view on your database.
Freehand SQL	Use an SQL statement to create your view
Composite View	Use to create a view which connect to multiple separate data sources and views.
OLAP	Select an existing OLAP cube – you will not be able to add additional attributes to a cube.
Stored Procedure	The ability to create a view from existing stored procedures on you database.

2. Select a category for the view. This is used to limit the field item categories to an appropriate set. For example only Sales Related field categories will be displayed for a sales view.
3. To create a new category click the create category link. This will open a popup window in which you can add a new category.
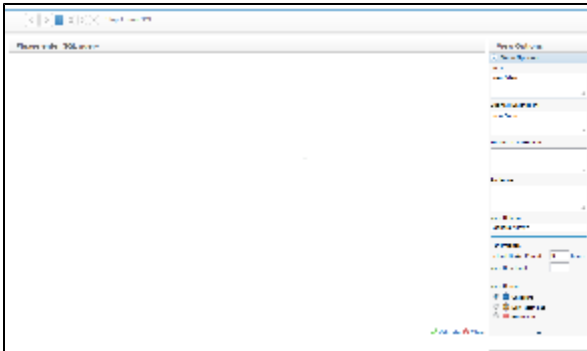
Drag and Drop Builder

This option allows you to use Yellowfin to simply join multiple tables from a single selected database through a web interface.

> See *Drag & Drop View Builder* for more information.

Freehand SQL View

The freehand SQL option allows you write your own view directly into the Yellowfin repository. The tables available from the source system are displayed in the table list on the left hand side.

Note: When writing an SQL view it is very important that you do not include an order by – since results are sorted by the Java application not the database. The order by will cause your view to fail.

Composite View

A composite view allows you to create a view which joins multiple data sources together. This allows users to create a report which includes data from these multiple databases.

The view builder for a composite view differs in that rather than displaying tables from a selected database it allows you to join existing Yellowfin views together and create virtual tables from any defined database.

1. To create a composite view select the composite view option from the author drop down.

2. You will also need to select a database in which you wish to store the view results. This is needed since composite views can be highly complex and result in slow query performance by your users. As a result Yellowfin schedules a query to generate the view and all its data on a regular basis to be stored in a specific warehouse database.

> ⚠ **Note**
>
> We strongly recommend the use of Advanced Sub Queries over Composite Views as the functionality is more flexible and robust.

OLAP Cube View

An OLAP cube is a pre-aggregated structure on your database. Like a data mart it already contains a set of dimensions and measures which have been pre-aggregated to support rapid reporting.

1. To create an OLAP cube view select OLAP cube from the author drop down menu.
2. Choose the source connection – Yellowfin will only provide OLAP source connections for you to choose from.
   If no source exists you will have to create a specific OLAP source connection record.
3. Choose the cube from a list of available cubes.

Note: When creating an OLAP cube as a view you will have limited options for updating field information. This is because most of the metadata is already contained in the cube itself.

Stored Procedure

With Yellowfin you can connect to stored procedures running on your application database.

1. To connect to a stored procedure select the stored procedure options from the create view drop down list.

2. Select the Database and then the stored procedure you wish to connect to.

3. Click the next arrow to continue to the stores procedure parameters page. Yellowfin will automatically detect parameters which exist in your stored procedure and display this list on the page.
4. You must fill in a value for each parameter. The value is only used to return column at this stage of the view builder. (The non-return fields will become automatic filters on the reports created from this stored procedure).
5. When you have completed the parameter section click accept to return the list of fields in the result set.

6. Complete the metadata as you would for a standard view. Note that the non-return parameters exist as filter fields.

7. Filters will be added into each report created from the stored procedure automatically. They will not be visible on the report data page but will be displayed on the report filter page.