

View Creation

- [Overview](#)
 - [View Builders](#)
- [Single Table View](#)
- [Multiple Table View \(Drag and Drop Builder\)](#)
- [Freehand SQL View](#)
- [OLAP Cube View](#)
- [Stored Procedure](#)
- [Composite View](#)

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.

To explore a basic introduction to creating views, visit the [Creating a View](#) page.

View Builders

When building a view, you have several methods available to you, including;

- [Single Table View](#)
- [Multiple Table View \(Drag & Drop Builder\)](#)
- [Freehand SQL View](#)
- [OLAP Cube](#)
- [Stored Procedure](#)

Single Table View

[top](#)

This option allow you to automatically create a view based on single table in the selected database. Yellowfin will automatically group metrics together, made up of numeric and date fields, and dimensions of text fields.

Create New View

Ski Team Create View Analyse

Select your view type
Create a view based on a single table within your selected data source connection.

Single Table ☒
Multiple Tables ☐
Freehand SQL ☐

Select Table:
Choose the table you want to analyse from your selected data source

Search

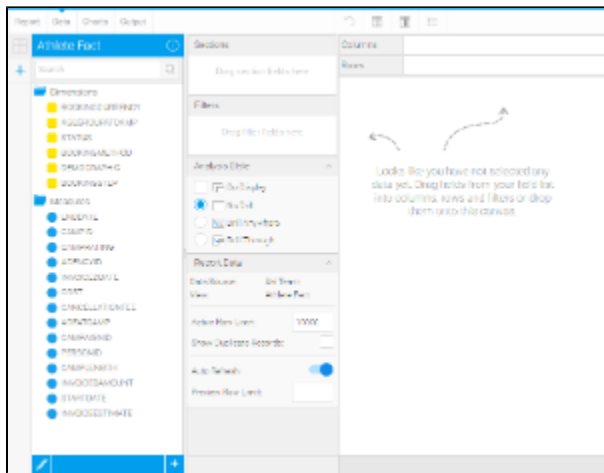
- ADDRESS
- ATHLETEFACT
- BONEBREAKS
- CAMP
- CAMPAIGN
- COUNTRYGEOMETRY
- DATELOOKUP

View Name
Give your view a great name to help others find it

View Description
Give your view a great description to help others find it

Create Analysis

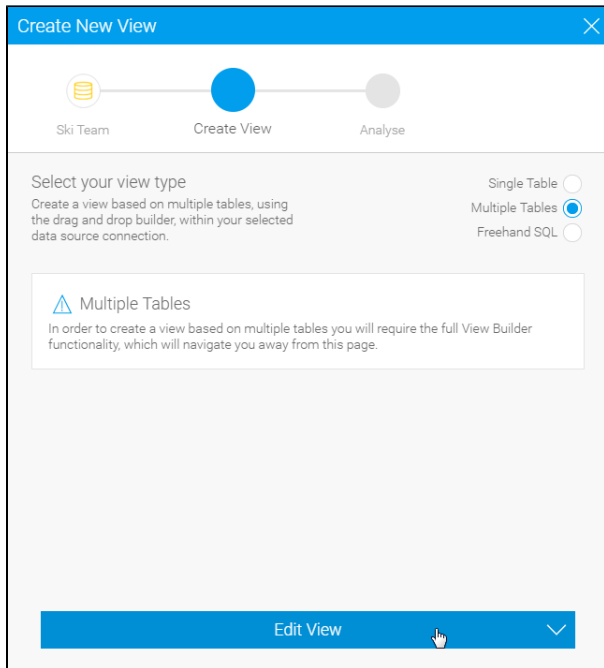
From here you can proceed directly to the Report Builder.



Multiple Table View (Drag and Drop Builder)

[top](#)

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

[top](#)

The freehand SQL option allows you to write your own query to generate a view. From here Yellowfin will automatically create the view, grouping metrics and dimensions together, much the same as the [Single Table View](#).

Create New View

Ski Team Create View Analyse

Select your view type
Create a view based on the results of a custom defined SQL query run on your selected data source connection.

Single Table ☐
Multiple Tables ☐
Freehand SQL ☒

select * from athletefact

Validate

✓ SQL is valid

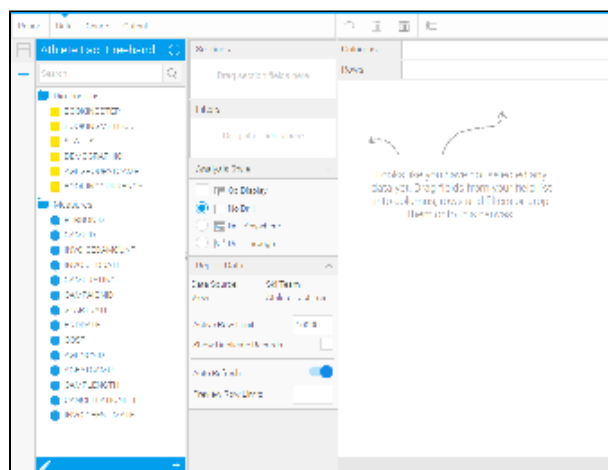
View Name
Give your view a great name to help others find it

Athlete Fact Freehand SQL

View Description
Give your view a great description to help others find it

Create Analysis

From here you can proceed directly to the Report Builder.



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.

OLAP Cube View

[top](#)

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 an OLAP cube source from the Create New View window.

2. Choose the cube from a list of available cubes.

Create New View

OLAP - SkiTeam **Create View** Analyse

Select Cube:
Choose the cube you want to analyse from your selected data source

Search: SkiTeamOlapDemo

View Name
Give your view a great name to help others find it
Ski Team Cube

View Description
Give your view a great description to help others find it

Create Analysis

3. You can then proceed straight to the report building process.

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

[top](#)

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.

Create New View

Stored Procedures **Create View** Analyse

Select your view type
Create a view based on the results of a stored procedure within your selected data source connection.

Single Table ☐
Stored Procedure ☒
Multiple Tables ☐
Freehand SQL ☐

Select Stored Procedure
Choose the stored procedure you want to analyse from your selected data source

Search: sp_500Day
sp_Payments
sp_SalesByState
sp_SalesByYearMonth
sp_StockAlbumName
sp_StoreSale

View Name
Give your view a great name to help others find it
Monthly Sales

View Description
Give your view a great description to help others find it

Edit View

3. Click the **Edit View** button 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).

- When you have completed the parameter section click accept to return the list of fields in the result set.

The screenshot shows the 'Current Result Set' section of the Yellowfin report builder. It features a table with two columns: 'Column Name' and 'Type'. The table contains the following data:

Column Name	Type
\$ Total Amount	DECIMAL
Month	VARCHAR
Total	BIGINT
Year	VARCHAR

Below the table is a 'Done' button. To the right of the table is a 'View Options' button and a 'View Options' link.

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

Composite View

[top](#)

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.



Note

We strongly recommend the use of [Advanced Sub Queries](#) over Composite Views as the functionality is more flexible and robust.

[top](#)