



VIASSIST

***VIAssist 2.6
Metadata Builder
User Guide***

December 14, 2011

Revision History

| Date | Revision | Paragraphs Affected | Description |
|------------|----------|---------------------|---|
| 12/14/2011 | 2.6 | All | User manual for VIAssist 2.6 Metadata Builder |

Table of Contents

| | | |
|-------|--|----|
| 1 | About..... | 3 |
| 2 | User Guide | 4 |
| 2.1 | Create a Metadata File..... | 4 |
| 2.2 | Add Basic Metadata Information | 4 |
| 2.3 | Add Table Information | 4 |
| 2.4 | Add Visualizations | 5 |
| 2.5 | Save the Metadata File..... | 6 |
| 3 | User Interface | 7 |
| 3.1 | Toolbar | 7 |
| 3.2 | Base Information..... | 8 |
| 3.3 | Tables | 9 |
| 3.3.1 | Information | 10 |
| 3.3.2 | Columns | 11 |
| 3.3.3 | Relationships..... | 13 |
| 3.3.4 | Join Paths | 15 |
| 3.3.5 | Virtual Fields | 17 |
| 3.4 | Default Select Fields..... | 19 |
| 3.4.1 | Vertical Market Settings | 20 |
| 3.4.2 | Vertical Market Field Setting Dialog | 21 |
| 3.5 | Visualizations..... | 22 |
| 4 | Advanced Topics | 23 |
| 4.1 | Data Tools..... | 23 |
| 4.1.1 | Data Tools | 23 |
| 4.2 | Inferring From Database | 24 |
| 4.2.1 | Inferring Information from Existing Database | 24 |
| 4.2.2 | Caveats..... | 27 |
| 4.3 | Using Join Paths..... | 27 |
| 4.3.1 | What are join paths?..... | 27 |
| 4.3.2 | Example..... | 28 |
| 4.4 | Virtual Fields..... | 32 |

1 About

The VIAssist product is a powerful tool with many powerful features. In order for each user to get the most out of VIAssist, the client can be configured to meet their specific needs for individual projects.

The configuration is stored in a metadata file and contains information ranging from what visualizations are available to what tables and information are available for querying in the database. Because of the wide range and complexity of the data, the Metadata Builder was created to get things up and running much more quickly.

This help file provides two sections depending on the current need:

1. User Interface
2. Advanced Topics

The User Interface section goes through every form and indicates all form elements that are user accessible and their purpose. Links to related topics are provided directly in the reading.

Advanced Topics provide a higher level understanding of the more complex functionality the Metadata Builder provides.

2 User Guide

The Metadata Builder provides all needed functionality to produce a client metadata file. This metadata defines the available functionality within VIAssist as well as what data is available to use. The Metadata Builder separates the different sections of metadata into different sections in the user interface. This user guide will describe the process of creating a metadata file and will give an overview of what the metadata is and which user interface elements to use.

2.1 Create a Metadata File

A metadata file can be created completely new or by editing an existing metadata file. To create or edit a metadata file, use either the create or open buttons on the toolbar. If a completely new file is created, no metadata will be present. If an existing metadata file is opened, metadata will be present and must be edited to reflect the desired changes.

2.2 Add Basic Metadata Information

The [base information](#) describes general project metadata, such as the project ID and its name. Fill this form with information that reflects the properties of this project.

Client metadata could belong in an existing vertical market. The Vertical Market Configuration section allows this client metadata to bring in more general vertical market data. Selecting a vertical market configuration will automatically populate tables, relationships, and join paths that pertain to that vertical market.

The SQL Manager defines the library that will be used to connect to the database from VIAssist, as well as from the Metadata Builder tool itself if certain database operations are selected. The SQL Manager selected must be appropriate for the type of database that will be connected to; for instance, if the database is a MySQL database, then a MySQL SQL Manager needs to be selected.

2.3 Add Table Information

Note: adding tables, deleting tables, or modifying tables only affects the metadata. No changes are made to any database.

Adding [tables](#) can be done in two different ways: manually, by adding individual tables and their fields, or through an inference process that connects to an existing database and pulls down all table information.

For more information on how to use the inference functionality, please see the Advanced Topic [Inferring From Database](#) .

If the inference process is not used, [table information](#) can be populated manually. Start by adding a new table and populating the basic table information. The dropdown list of tables will not contain a readable table name until the display name of the table information has been set.

For each table, [columns](#) must be defined. Without columns, no fields of information would be available. If the inference process was used, columns will already be present. If not, columns can be manually added. Each column added represents a field of information that can be exposed to the user for querying, or hidden from the user. VIAssist has many options for fields of information which the Metadata Builder exposes through the field details button. The [field details](#) form defines exactly how VIAssist should utilize this field. The field details form also allows specific data tools to be set for that field. These data tools are made available within VIAssist when interacting with that specific field. For more detailed information about data tools, please see the Advanced Topic [Data Tools](#) .

When more than one table is defined, [relationships](#) can then be defined. A relationship simply describes how two tables relate to each other by defining which columns can be used as keys into the related table.

When relationships are defined between tables, [join paths](#) can then be created. Join paths describe a route from one table to another that may go through several intermediate tables. For more detailed information about join paths, please see the Advanced Topic [Using Join Paths](#) .

Finally, the virtual fields functionality may be used to link a field into the table that exists in another table.

A table can be deleted at any time.

2.4 Add Visualizations

[Visualizations](#) define what visualizations will be available within VIAssist for use on this project. The visualizations section contains all visualizations that VIAssist supports. All of them can be

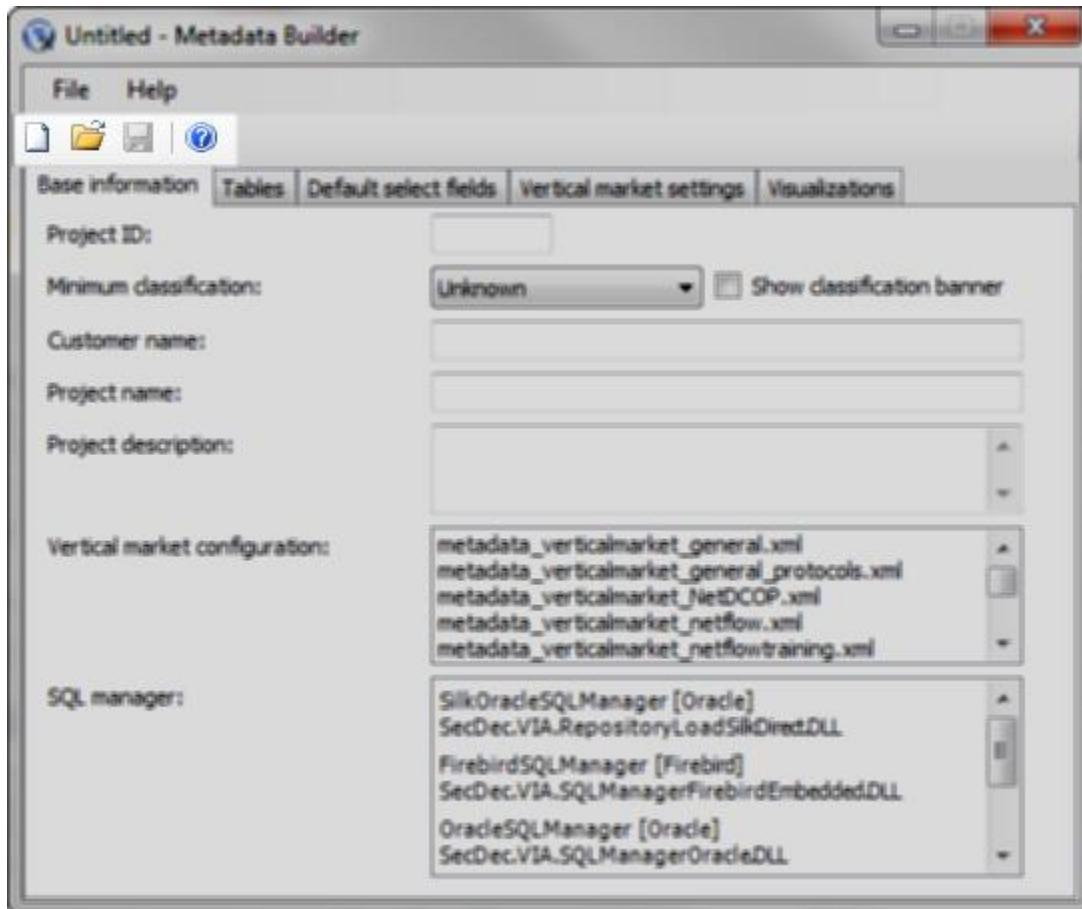
added if desired, but it is usually best practice to select visualizations that are most applicable to the type of data this project represents.

2.5 Save the Metadata File

After all metadata information has been populated, the metadata file can be saved. If the metadata was created from an existing metadata file, take care to only overwrite the existing file if the modifications are for that file. Save the metadata file.

3 User Interface

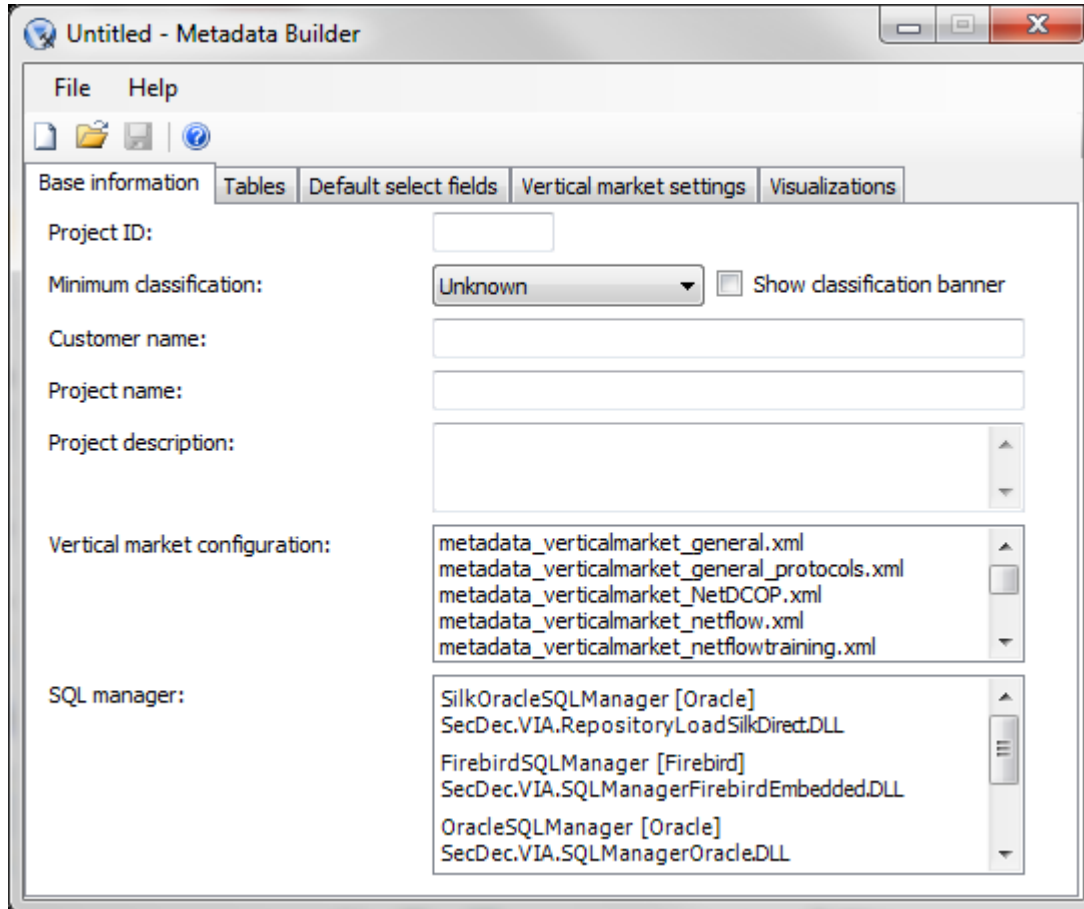
3.1 Toolbar



The main toolbar contains four buttons. From left to right:

- Create a new metadata file.
- Open an existing metadata file.
- Save the current metadata. This option will be disabled if there are no changes to save.
- Open this help document.

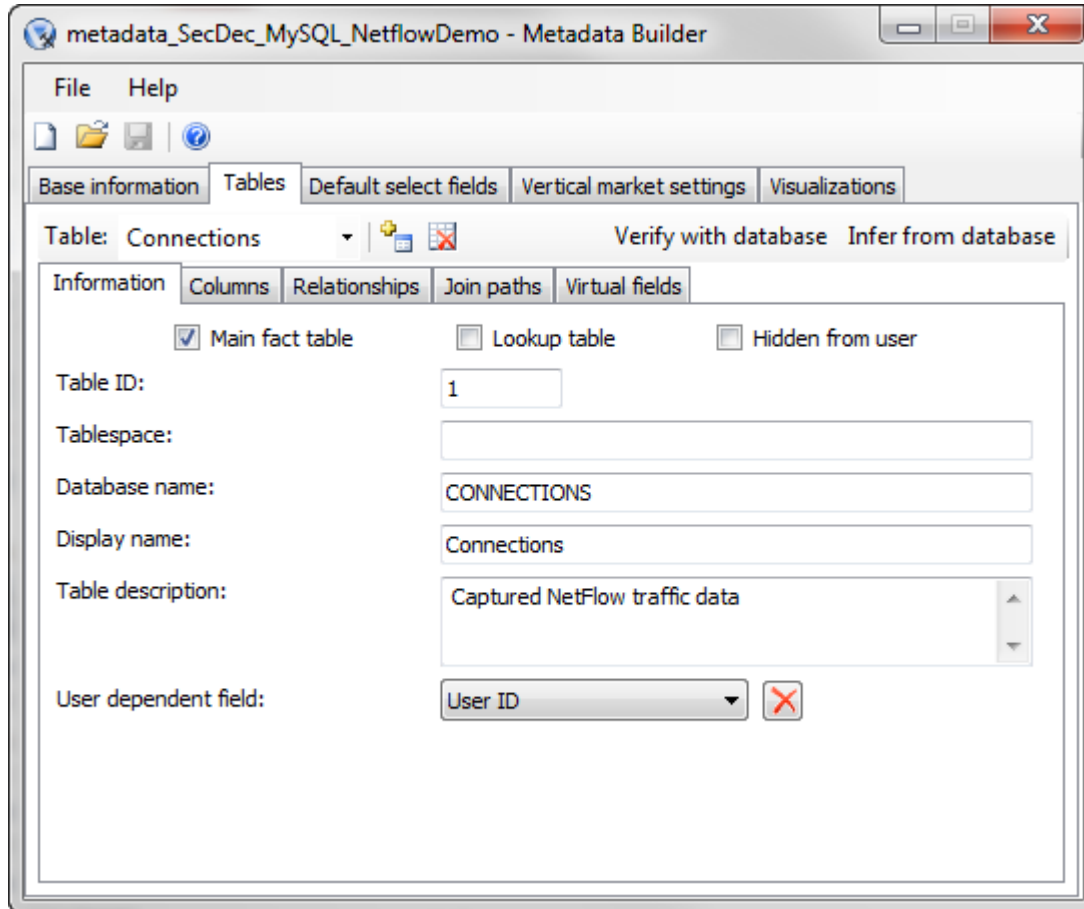
3.2 Base Information



The base information page defines basic information about the project.

- Project ID: a unique identification number for this metadata project.
- Minimum classification: the minimum required classification level to utilize this metadata.
- Show classification banner: whether or not the classification banner should be shown to the user in VIAssist.
- Customer name: the name of the customer for this metadata file.
- Project name: the name of the project the metadata file describes.
- Project description: an optional, short description of the project. This will be shown whenever the data source selection dialog is displayed.
- Vertical market configuration: the base vertical market configuration file to use.
- SQL Manager: the SQL manager to use to communicate with the database.

3.3 Tables

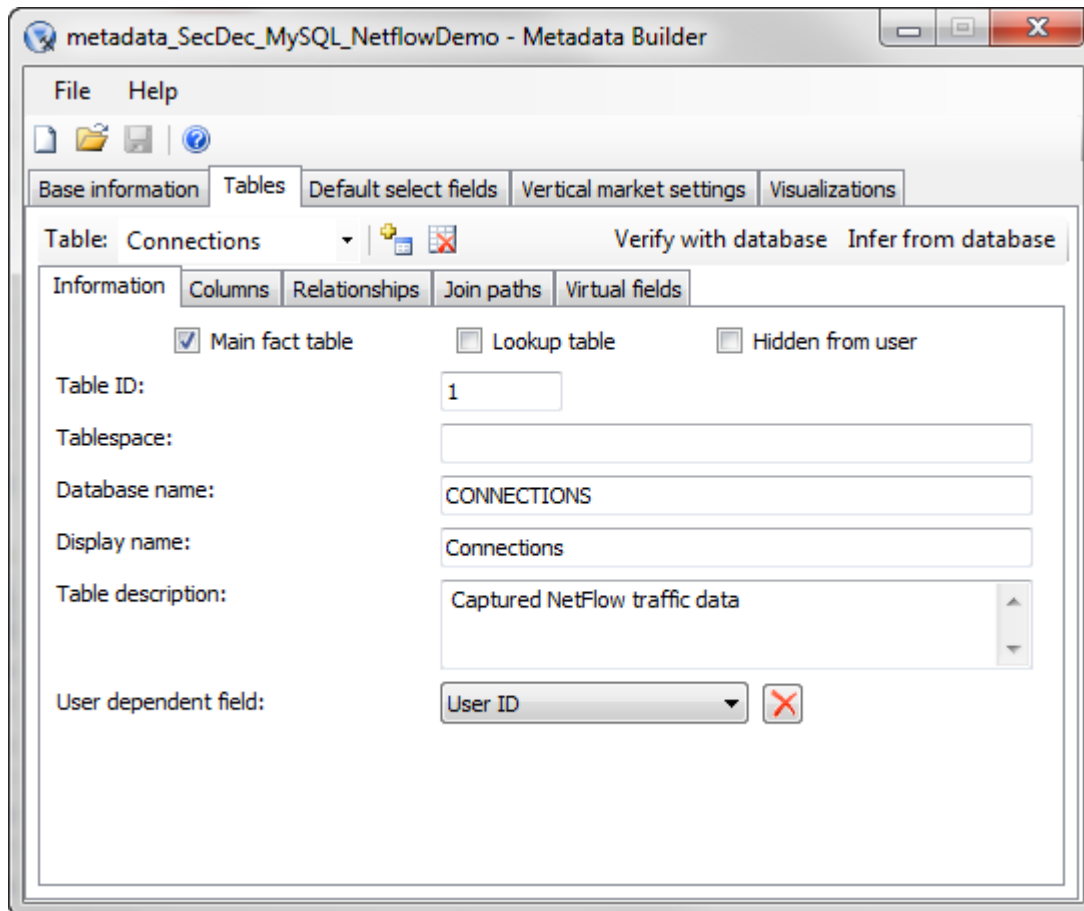


The tables page defines the data that is available in the database as well as how tables can relate to one another. Tables can be added either manually or through a process that infers information directly from a database.

- The table combo box provides a list of all tables available for modification.
- The two buttons next to the tables text box allow adding and removing tables.
- Verify with database: Verifies with the database that the table information created is valid.
- Infer from database: Connects to the database and infers as much data as possible. [Please read the "Inferring from a database" page for more information.](#)
- There are tabs for editing various aspects of the table:
 - The [Information](#) tab, for setting basic table information.
 - The [Columns](#) tab, for adding, editing, and removing fields.

- The [Relationships](#) tab, for adding, editing, and deleting relationships between tables.
- The [Join paths](#) tab, for adding, editing, or deleting join paths.
- The [Virtual fields](#) tab, for adding, editing, and deleting virtual fields.

3.3.1 Information

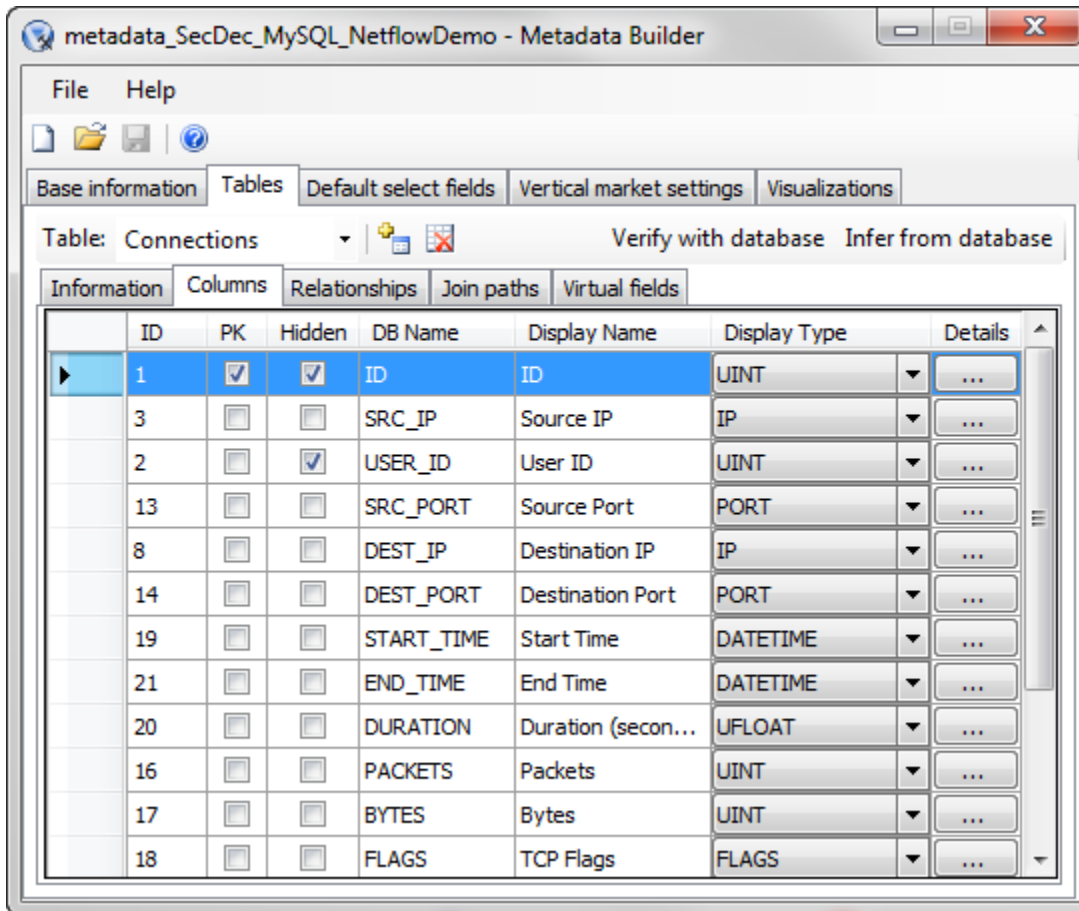


The table information page allows for setting basic table information.

- Main fact table: whether or not this table acts as a main fact table.
- Lookup table: whether or not this table acts as a lookup table.
- Hidden from user: whether or not this table is hidden from the user.
- Table ID: a unique identification number for this table.
- Tablespace: defines the tablespace this table belongs to. This is only required if the current user logging in does not have the table space where the table resides as their default tablespace. This is usually left blank.

- Database name: the name of the table, as defined in the database.
- Display name: the name of the table, as shown to the user.
- Table description: short description of the table, displayed in the field selection dialog.
- User dependent field: the field that depends on which user is using the table, if applicable.

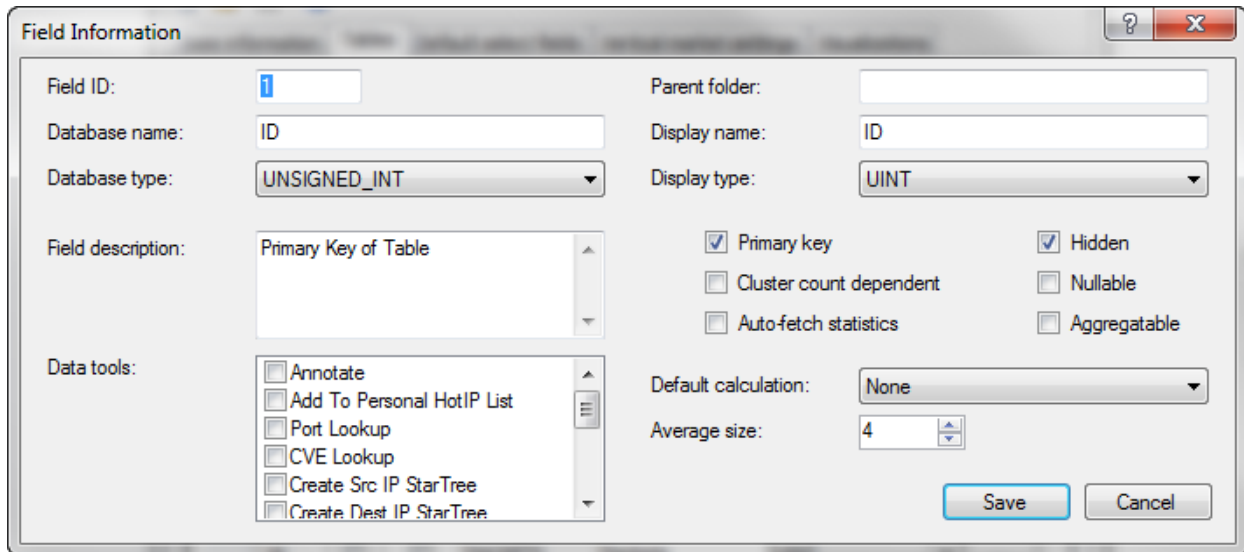
3.3.2 Columns



The table columns page allows for adding, editing, and removing fields in the table.

- PK column: whether or not this field is a primary key to the table.
- Hidden column: whether or not this field is hidden from the user.
- Display Type column: the data type of the field, as shown to the user.
- Details column: opens the detailed [field information form](#), which allows for more advanced editing options.

3.3.2.1 Field Details

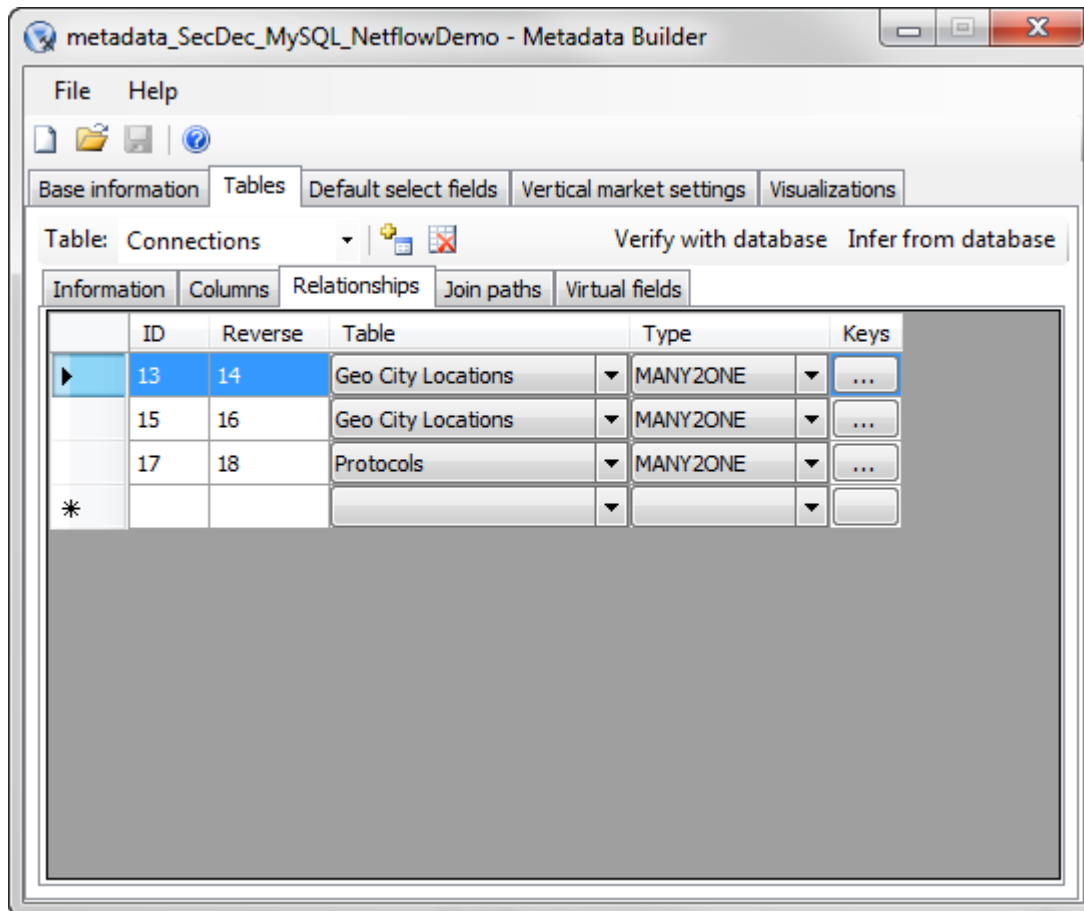


Note: Not every option on a field may be editable. For example, if a field was inferred from a database, only options that are not tied to the database can be modified.

- Field ID: a unique identification number for this field.
- Database name: the name of the field, as defined in the database.
- Database type: the data type of the field, as defined in the database.
- Field description: an optional, short description of this field.
- Data tools: the [data tools](#) that should be enabled for this field, dependent on the display type (8).
- Parent folder: the parent folder, if any, that this field should appear under in the Visual Query Builder.
- Display name: the name of the field, as shown to the user.
- Display type: the data type of the field, as shown to the user.
- Primary key: whether or not this field is a primary key for the table.
- Cluster count dependent: whether or not this field is dependent on cluster count. If checked, values are multiplied by the cluster count to create a total value (i.e., bytes).
- Auto-fetch statistics: whether or not statistics for the field should be automatically fetched for the Visual Query Builder, prior to the user referencing the field.
- Hidden: whether or not the field is hidden from the user.
- Nullable: whether or not the field may be null.
- Aggregatable: whether or not the field is aggregatable.
- Default calculation: the calculation that is applied to the field by default.

- Average size: indicates the average size of a value in the field. This value is used by the Smart Aggregator.
- Save: saves the modifications made to the field information.
- Cancel: cancels the modifications made to the field information.

3.3.3 Relationships



The table relationships page allows for adding, editing, and deleting relationships between tables.

Note: Relationships must be defined in order to create [join paths](#).

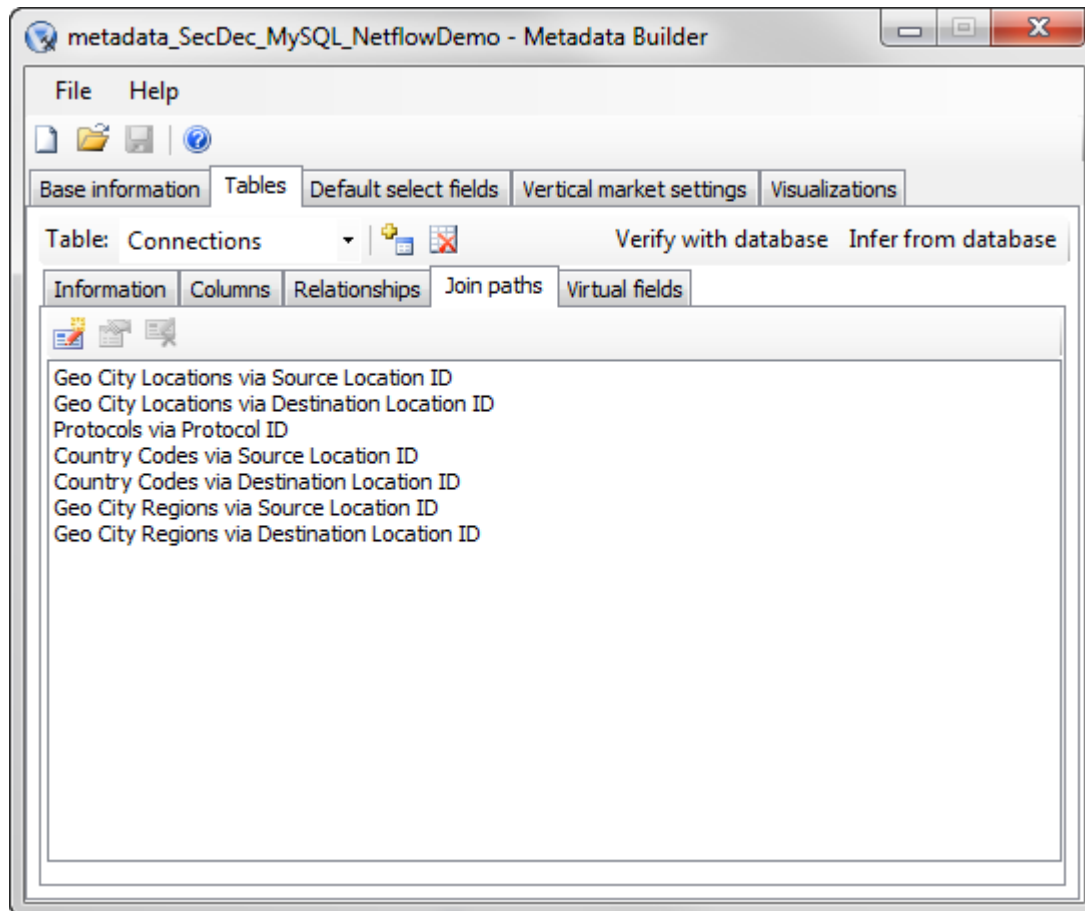
- Table column: the other table participating in this relationship.
- Type column: the cardinality of the relationship between these two tables.
- Keys column: displays the [Key Information form](#) for setting the primary and foreign keys that establish this relationship.

3.3.3.1 Key Information

| ID | Primary Key | Foreign Key |
|----|-------------------------|-------------|
| 13 | Destination Location ID | Location ID |
| | | |

- ID column: a unique identification number for this key.
- Primary Key column: the primary key for a relationship.
- Foreign Key column: the foreign key for a relationship.
- Save: saves the key information for this relationship.
- Cancel: cancels the modification to key information.

3.3.4 Join Paths



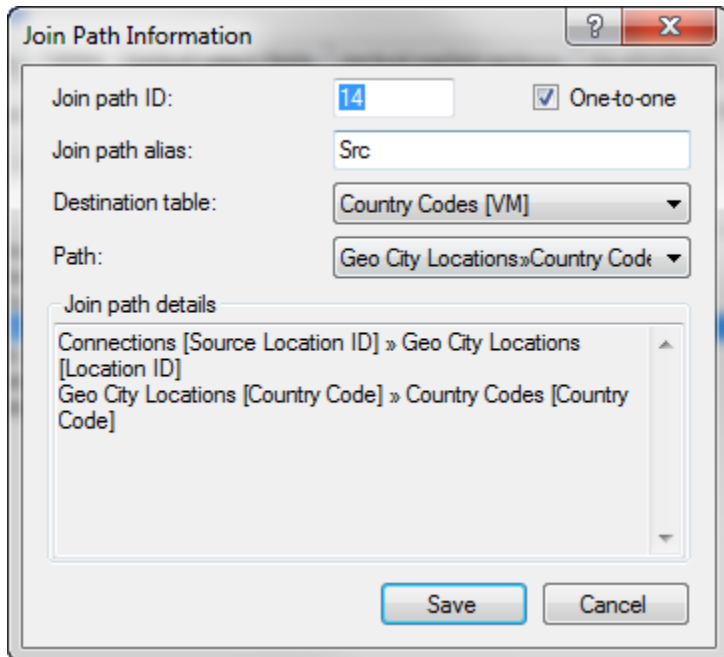
The table join paths page allows for adding, editing, and deleting join paths among tables.

Note: [Relationships](#) must be defined in order to create join paths.

Note: For more in-depth information on what join paths are and how to use them, please visit the [Advanced Topics page Using Join Paths](#).

- The tab contains a list of all join paths that have been added for the current table.
- The toolbar contains three buttons. From left to right:
 - The add button, which will activate the [Join Path Information](#) form for adding a new join path.
 - The edit button, which will activate the [Join Path Information](#) form for editing an existing join path.
 - The delete button, for removing a join path.

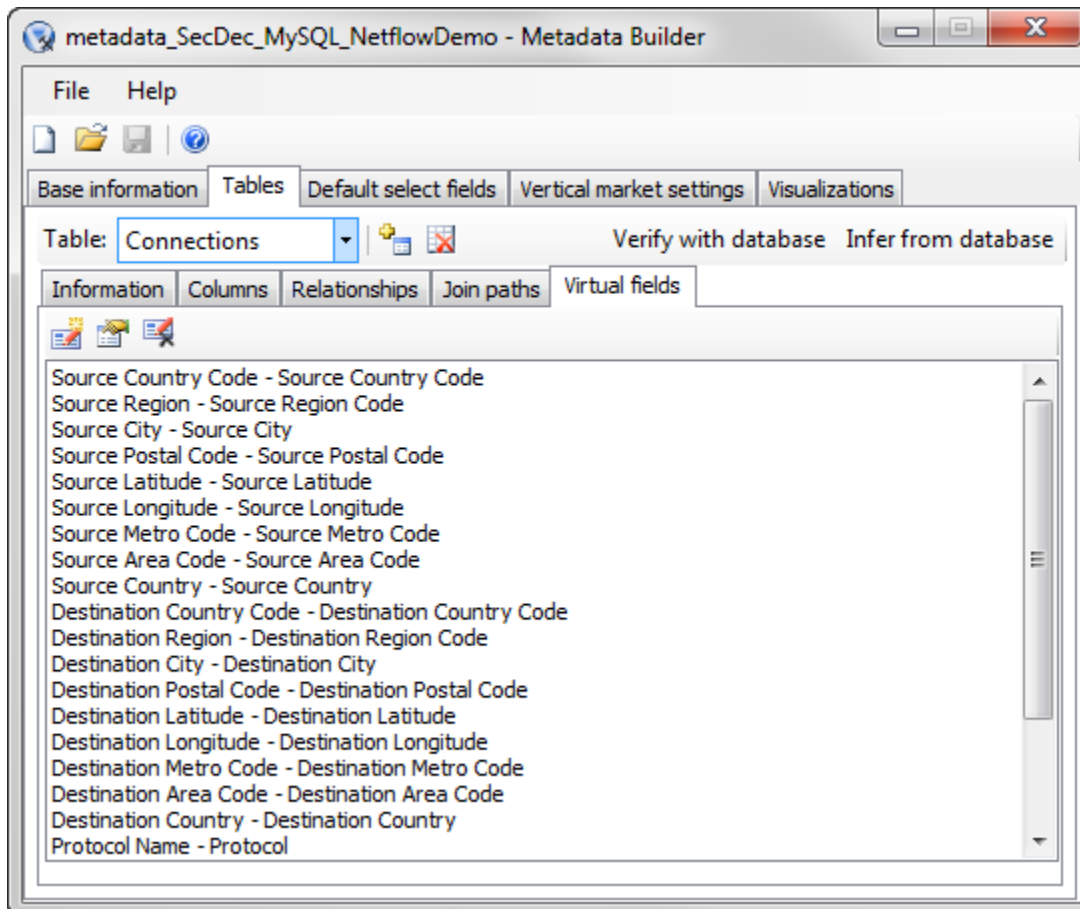
3.3.4.1 Join Path Information



Note: For more in-depth information on what join paths are and how to use them, please visit the [Advanced Topics page Using Join Paths](#).

- Join path ID: a unique identification number for this join path.
- One-to-one: whether or not this join path indicates a one-to-one relationship between the source and destination.
- Join path alias: an alias for the join path, shown with the field name. This value helps differentiate between multiple join paths, for example, when joining to a table based on source IP and destination IP.
- Destination table: the table to be joined to.
- Path: the path to take for the join. Join path choices are calculated based on table relationships.
- Join path details: the details of the actual join path, showing how the path traverses through the tables from source table to destination table.
- Save: saves this join path.
- Cancel: cancels the creation of this join path.

3.3.5 Virtual Fields

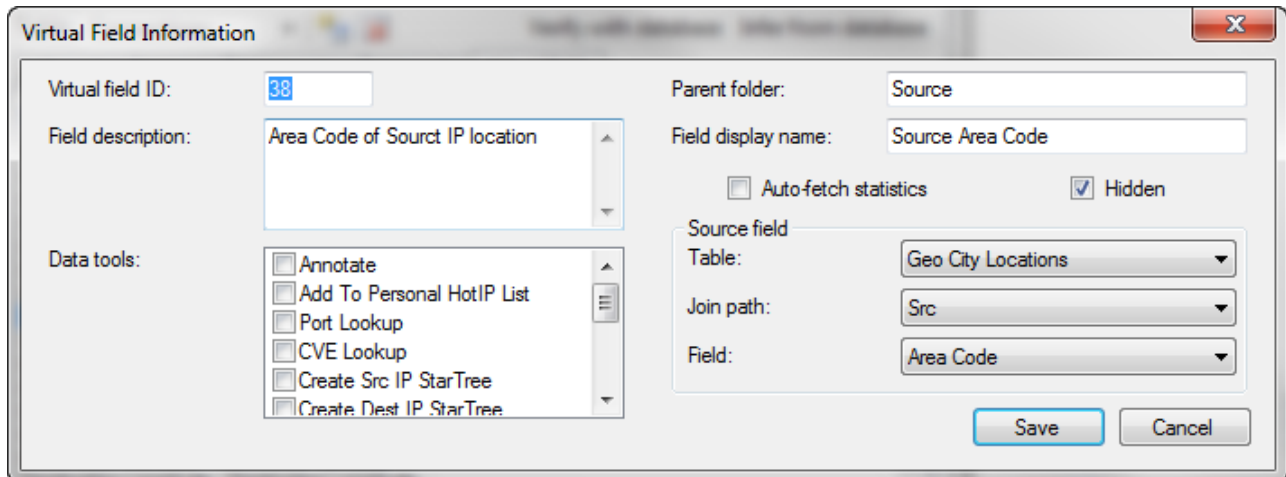


Note: For more in-depth information on what virtual fields are, please visit the [Advanced Topics page Virtual Fields](#).

The table virtual fields page allows for adding, editing, and deleting virtual fields for the current table.

- The tab contains a list of all virtual fields that have been added for the current table.
- The toolbar contains three buttons. From left to right:
 - The add button, which will activate the [Virtual Field Information](#) form for adding a new virtual field.
 - The edit button, which will activate the [Virtual Field Information](#) form for editing an existing virtual field.
 - The delete button, for removing a virtual field.

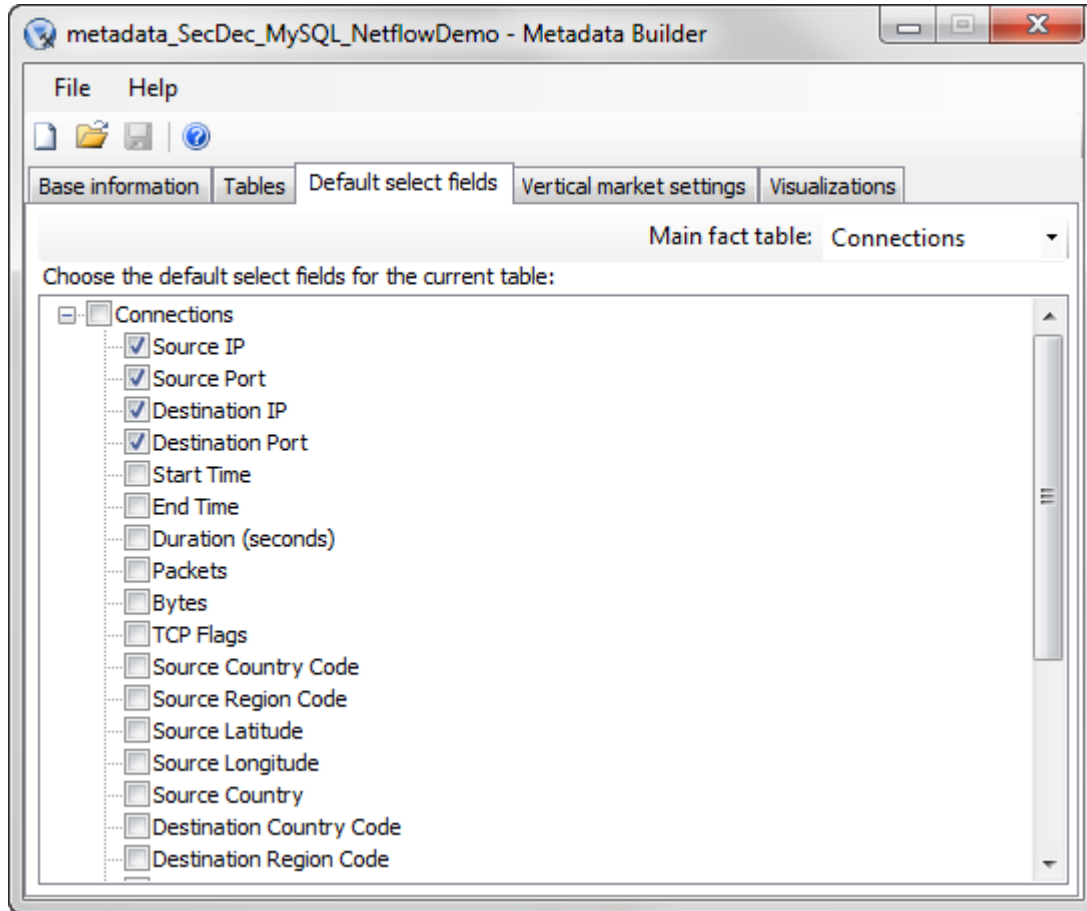
3.3.5.1 Virtual Field Information



Note: For more in-depth information on what virtual fields are, please visit the [Advanced Topics page Virtual Fields](#).

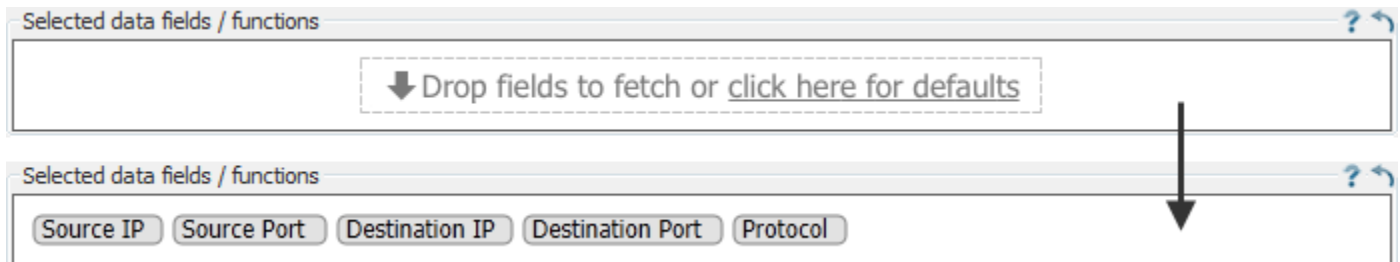
- Virtual field ID: a unique identification number for this field.
- Field description: an optional, short description of this field.
- Data tools: the [data tools](#) that should be enabled for this field, dependent on the display type (8).
- Parent folder: the parent folder, if any, that this field should appear under in the Visual Query Builder.
- Field display name: the name of the field, as shown to the user.
- Auto-fetch statistics: whether or not statistics for the field should be automatically fetched for the Visual Query Builder, prior to the user referencing the field.
- Hidden: whether or not the field is hidden from the user.
- Table: the table to copy the virtual field from.
- Join path: the join path to use to connect to the selected table.
- Field: the field to copy from the table.
- Save: saves the modifications made to the field information.
- Cancel: cancels the modifications made to the field information.

3.4 Default Select Fields

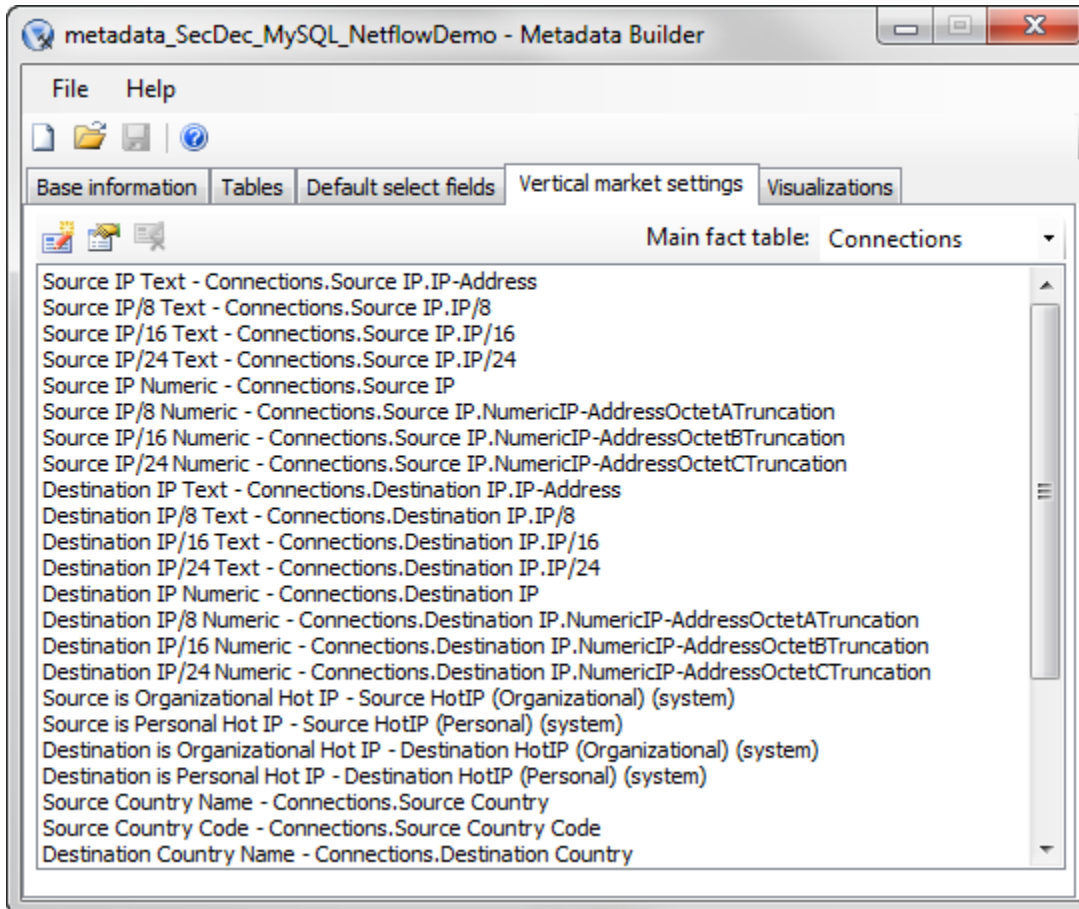


The default select fields tab allows selection of the default selection fields.

- Main fact table: the table to edit the default select fields for.
- The list on the tab shows which fields will be added when the user requests the default select fields in the Visual Query Builder when the corresponding main fact table is active.



3.4.1 Vertical Market Settings



The vertical market settings page presents any configurable settings for the selected vertical market file.

- The main fact table combo box provides a list of all tables available for configuration. Vertical market settings apply to main fact tables.
- There are three toolbar buttons beside the main fact table selection. From left to right:
 - The add button, which will activate the [Vertical Market Field Setting](#) form for adding a custom vertical market setting.
 - The edit button, which will activate the [Vertical Market Field Setting](#) form for editing an existing vertical market setting.
 - The delete button, for removing a custom vertical market setting.
- The list on the tab shows all vertical market settings. This includes any custom ones added, as well as ones pre-defined by the vertical market.

3.4.2 Vertical Market Field Setting Dialog

Vertical Market Field Setting

Source IP/24 Text - Field containing source IP field, truncated to third octet, in string format. Usually uses 'IP/24' calculation.

Constant name: SrcIP_TruncC

Constant field

Table: Connections

Join path: None

Field: Source IP

Calculation: IP/24

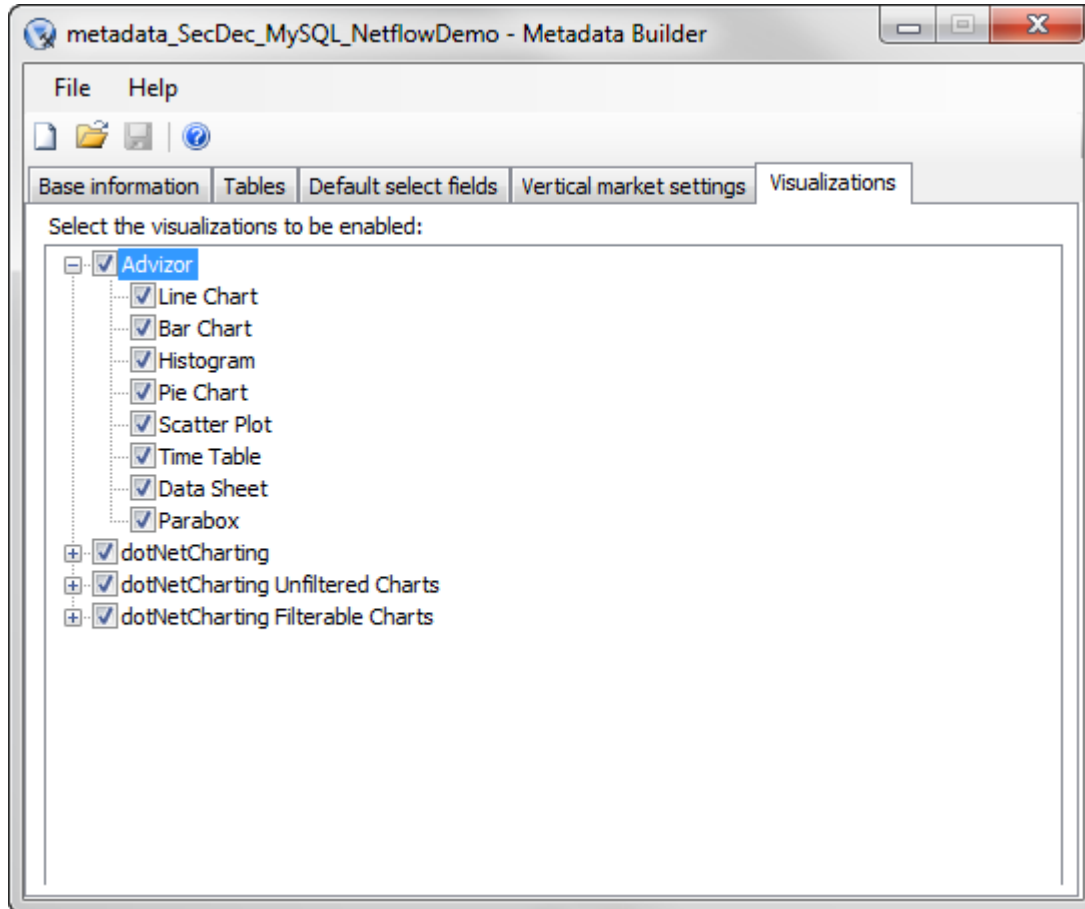
Save Cancel

Note: Not every option on a vertical market setting may be editable. If the setting was pre-defined by the vertical market, the name will be read-only, and the join path can.

Note: The table/field/calculation boxes only show compatible values. Only tables with compatible fields will be shown, and only fields with compatible calculations will be shown. Join path selection is only required when connecting to a separate table that has more than one possible join path.

- The name and description of the setting are displayed at the top of the dialog.
- Constant name: the name of the vertical market setting. This is read-only while editing pre-defined vertical market settings.
- Table: the table to use for the field setting.
- Join path: the join path to use to get to the selected table, if one is required.
- Field: the field from the table to use.
- Calculation: the calculation to use on the selected field.

3.5 Visualizations



The visualizations tab allows selection of which views will be available within VIAssist.

The list on the tab shows available visualizations. Those which are checked will be made available for the user within VIAssist while using this metadata project.

4 Advanced Topics

4.1 Data Tools

4.1.1 Data Tools

Data tools give extended functionality to the interaction of specific fields within a table. These tools can be wide-ranging in effect, from simply acting as a highlight to being the focus of a drill-in action. Other utility type of functionality may be exposed as well.

To set data tools on a field, access the [field details](#) form for the field of interest. This form contains many options, but there are only two parts of the form that pertain to the data tools: the "Display Type" drop down box and the "Data Tools" selection box.

The screenshot shows the "Field Information" dialog box. The "Display Type" dropdown is set to "TEXT". The "Data Tools" section is expanded, showing the following options:

- Show Data Details
- Drill-In
- Add as Display Filter
- Add as Highlight Filter
- Copy to Clipboard
- Show Full Field Contents

Other visible options in the dialog include:

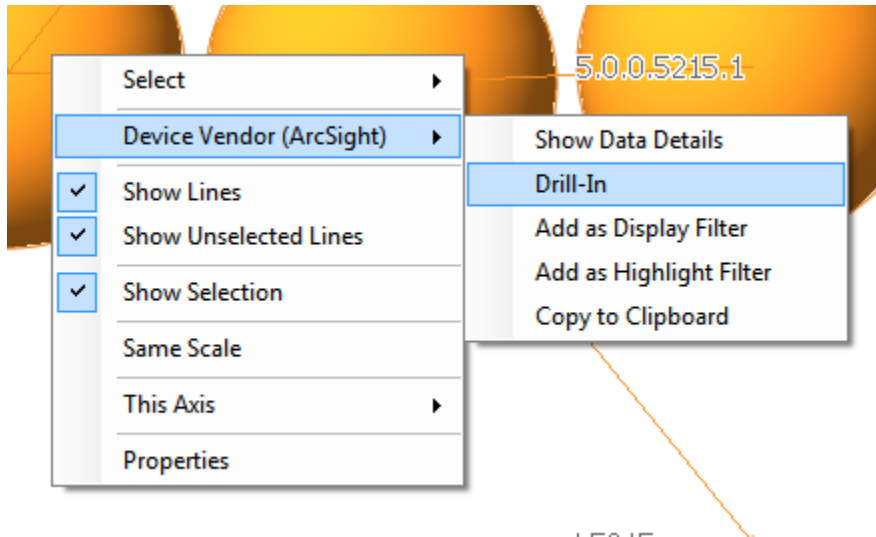
- Primary Key
- Cluster Count Dependent
- Auto-Fetch Statistics
- Hidden
- Nullable
- Aggregatable
- Default Calc: None
- Average Size: 5
- Buttons: Save, Cancel

The "Display Type" drop down box determines what data tools can be selected for this field. Not every data tool is compatible with every type, therefore the display type is important to consider. Once a display type is chosen, the data tools selection box will contain a list of data tools that could be applied to this field.

Select as many of the data tools as desired. When the selections have been made, save this modified field information.

In the example highlighted screenshot, the "Device Vendor" field has been modified so that the display type is TEXT and five data tools are selected: Show Data Details, Drill-In, Add as Display Filter, Add as Highlight Filter, and Copy to Clipboard.

These data tools will now be available within VIAssist when interacting with that particular field in a visualization. Right click the field inside of a visualization to bring up the context menu.



In this example, the right click context menu in VIAssist is shown for the "Device Vendor" field. A submenu is seen populated with the same data tools that were selected from the Metadata Builder.

4.2 Inferring From Database

4.2.1 Inferring Information from Existing Database

The Metadata Builder is a flexible tool that allows every aspect of the metadata to be defined manually. While ideally the metadata file would be created side-by-side as the database is developed, this is not always possible. Systems may already be in place and having to manually add this information could be a tremendous task.

To ease the creation of the metadata, the Metadata Builder can connect to an existing database to pull down all table information through the [Infer from Database \(6\)](#) functionality.

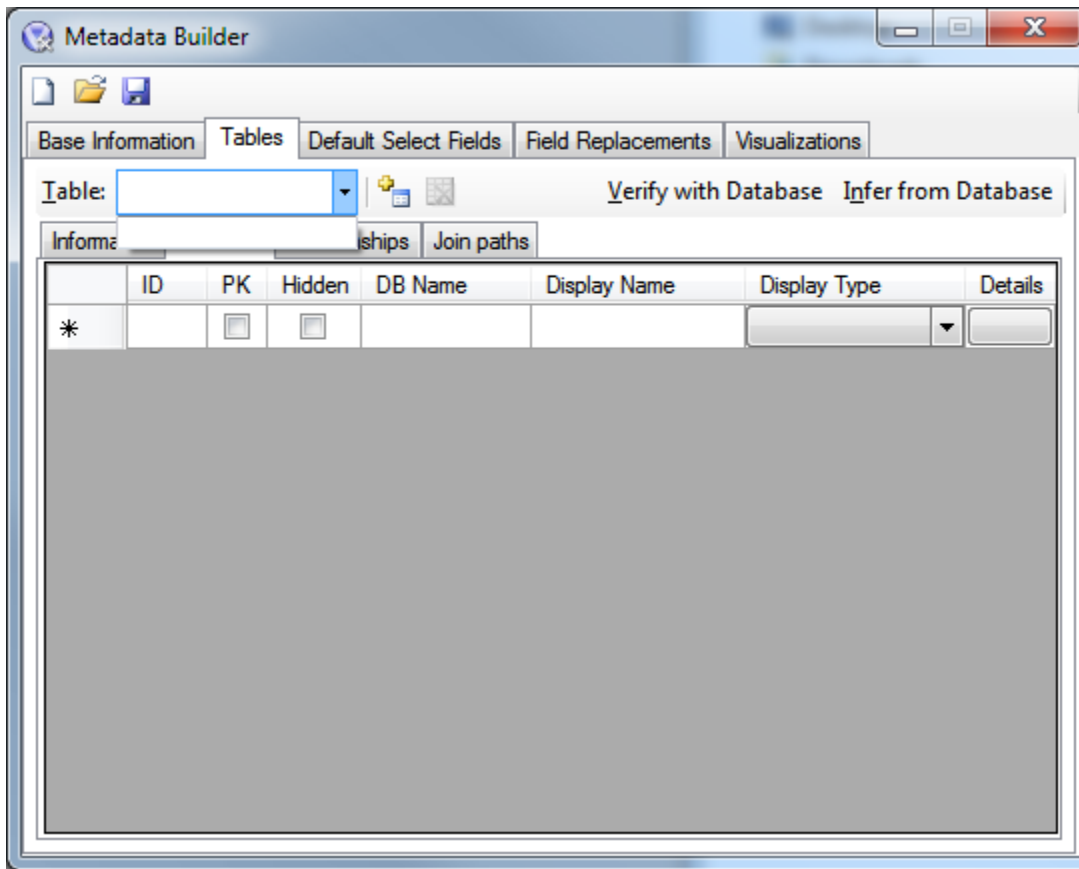
The inference process is largely automated. When first activating the Infer from Database functionality, a credentials login screen will be presented. The login screen will differ depending on the [SQL Manager chosen on the Base Information page](#) .

The image shows a 'Log In...' dialog box with the following fields and buttons:

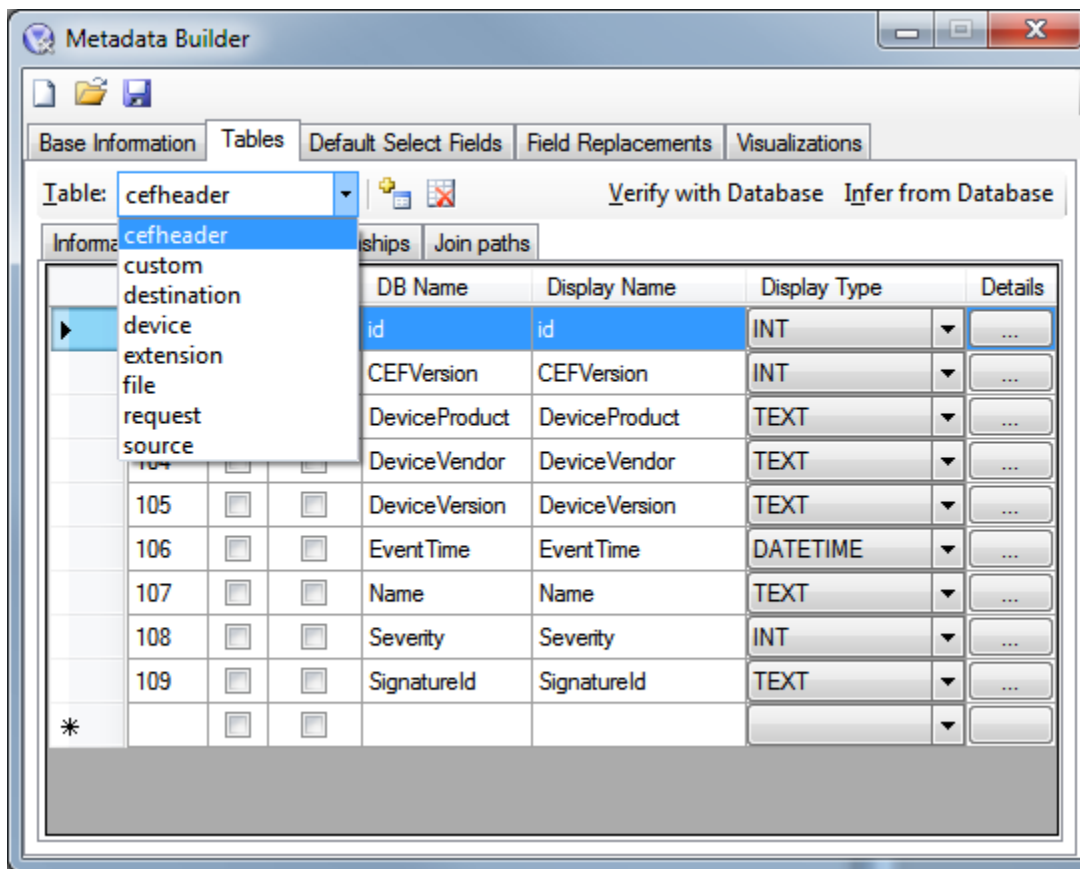
- Server name:
- Port:
- Database:
- Timeout (seconds):
- Credentials section:
 - Username:
 - Password:
- Buttons: Log In, Exit

Once valid credentials are accepted, the Metadata Builder will automatically populate all table information.

Note here how there are no tables available for selection and, subsequently, no field information.



After the inference process completes, many tables are available for viewing, as well as their fields.



4.2.2 Caveats

By automatically populating all of the table information stored in the database, the potentially large process of manually entering every table, field, type information, and more is avoided. However, relationships and join paths are not automatically created due to the subjective nature that they represent.

It is likely that unneeded tables will need to be removed, that needed [relationships](#) will have to be added, and that needed [join paths](#) will have to be created, but this should be significantly less of a burden than entering large amounts of table information.

4.3 Using Join Paths

4.3.1 What are join paths?

Tables are related to each other through any number of primary key to foreign key relationships. When rows from different tables have a matching pair of primary and foreign keys, the data contained in those rows can be associated together. However, complete

information is not always present within a single relationship. Data can be spread across any number of tables and there is a need for a well-defined path to get from one table to another in order to get desired information. Even further, there may be many different ways to go from one table to another.

Join paths help clarify what paths are available between different tables.

4.3.2 Example

4.3.2.1 Tables

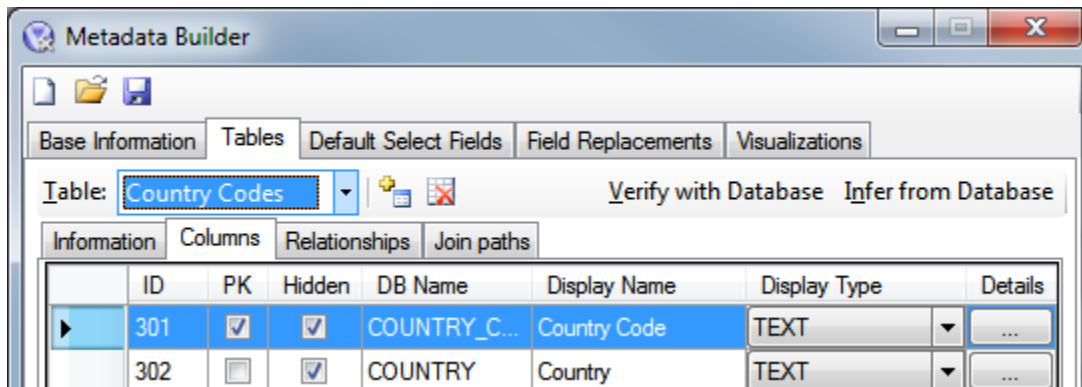
Three basic [tables](#) are defined:

The screenshot shows the 'Metadata Builder' application window with the 'Connections' table selected. The 'Columns' tab is active, displaying a table with the following data:

| ID | PK | Hidden | DB Name | Display Name | Display Type | Details |
|-----|--------------------------|-------------------------------------|---------------|-----------------------|--------------|---------|
| 101 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | SOURCE_LO... | Source Location ID | UINT | ... |
| 102 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | DESTINATIO... | Destination Locati... | UINT | ... |

The screenshot shows the 'Metadata Builder' application window with the 'Geo City Location' table selected. The 'Columns' tab is active, displaying a table with the following data:

| ID | PK | Hidden | DB Name | Display Name | Display Type | Details |
|-----|-------------------------------------|-------------------------------------|--------------|--------------|--------------|---------|
| 201 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | LOCATION_ID | Location ID | INT | ... |
| 202 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | COUNTRY_C... | Country Code | TEXT | ... |



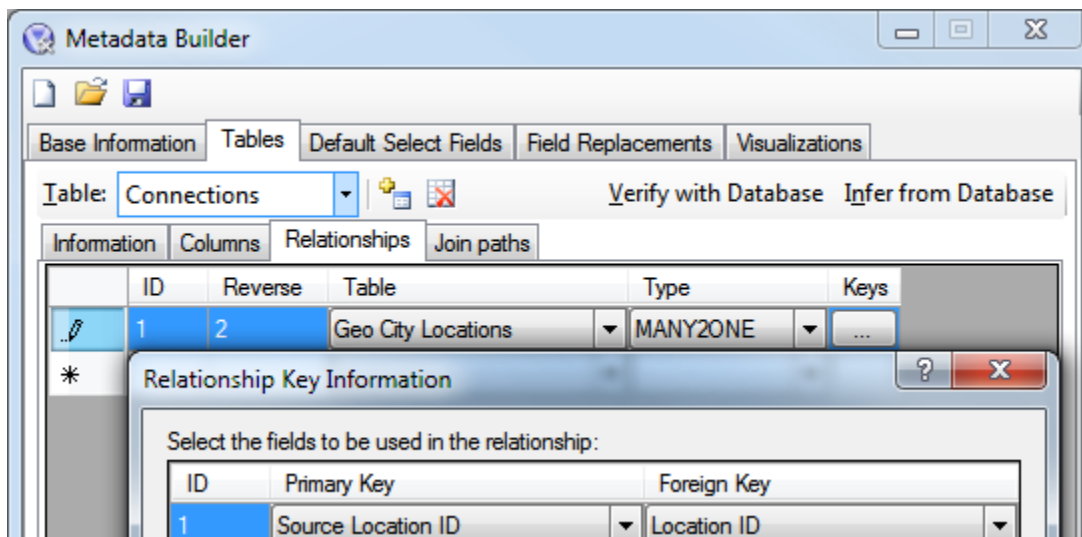
4.3.2.2 Problem

Given a Source Location ID in the Connections table, there is no immediate way of knowing the name of the country the location ID belongs to.

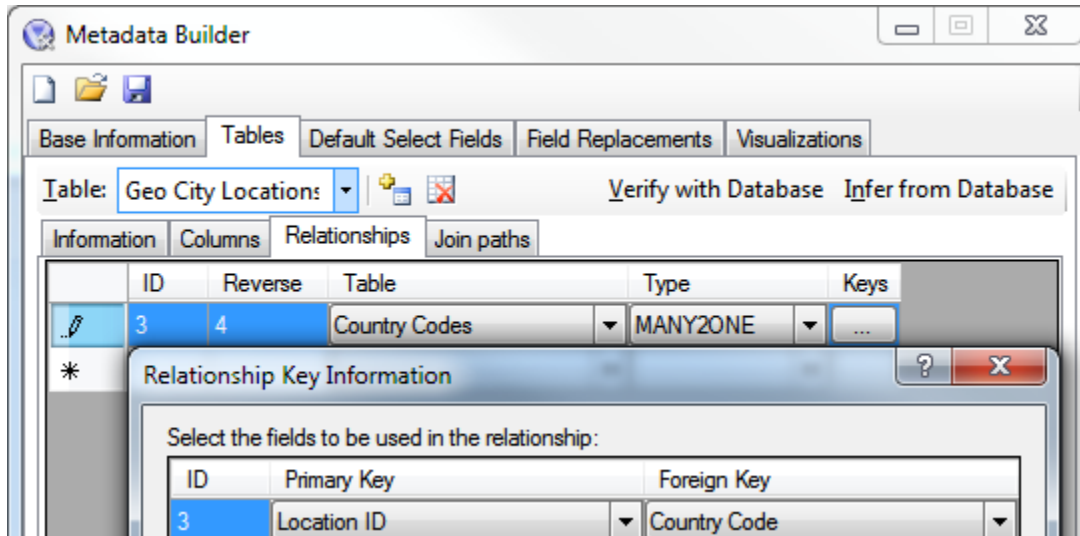
4.3.2.3 Solution

To solve this problem, the needed [relationships](#) are established between each table that will participate in the join path.

The first relationship created is between the Connections table and the Geo City Locations table. The relationship is defined through the Source Location ID field of the Connections table and the Location ID field of the Geo City Locations table.



The second relationship created is between the Geo City Locations table and the Country Codes table. The relationship is defined through the Location ID field of the Geo City Locations table and the Country Code field of the Country Codes table.

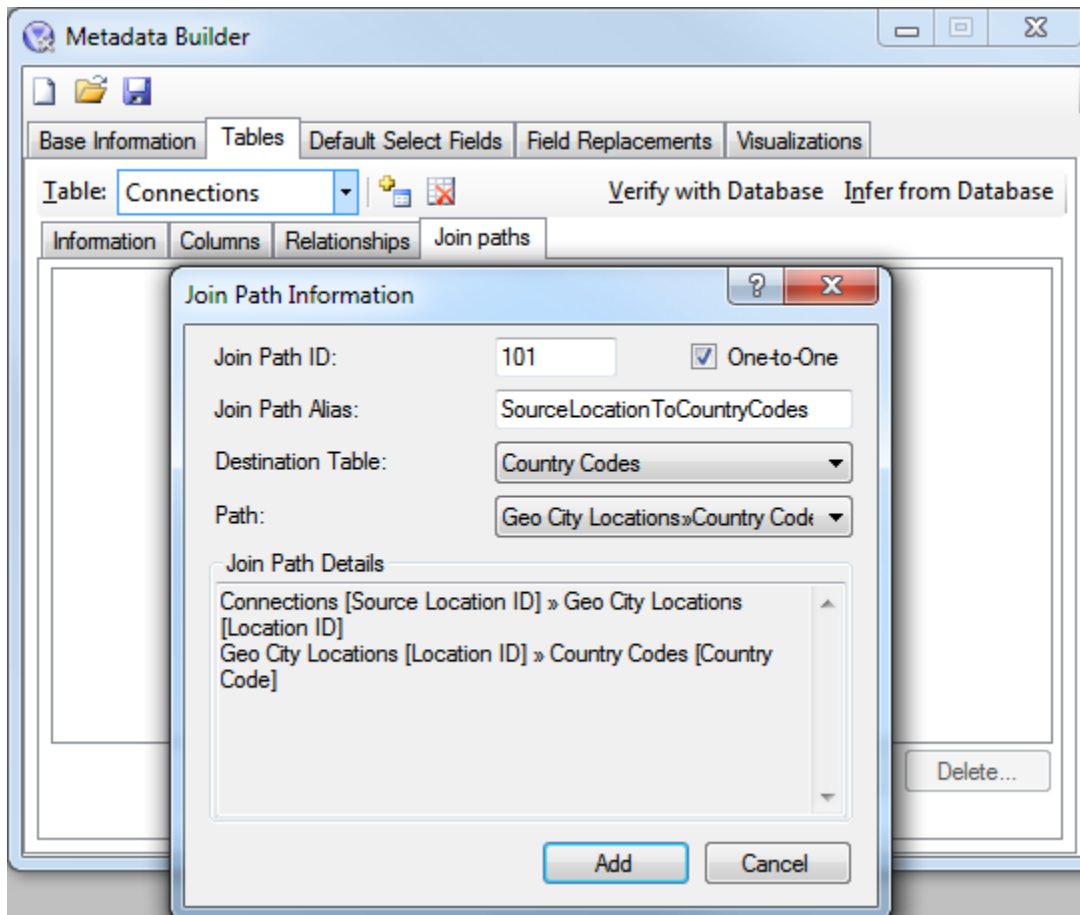


No other relationships need to be established.

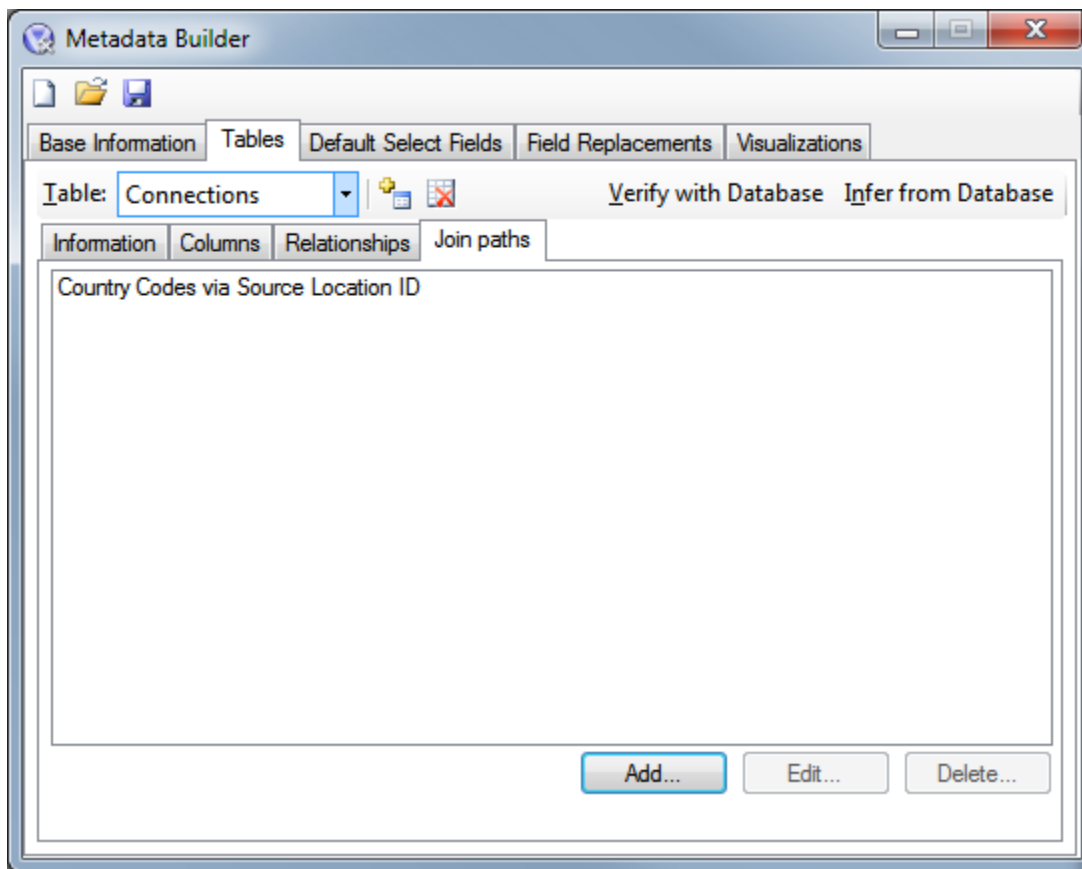
Now that the relationships are established, [join paths](#) must be set.

A join path is created first by selecting the destination table that needs to be reached. Once the destination table is selected, all potential paths to get to that table are listed in the Path drop down box. Simply select the desired path (in this example, only one such path exists) and then press the "Add" button.

Note: The Join Path Information form only allows valid selections, as defined by what relationships are available. This greatly simplifies the process of adding paths between tables.



After a join path has been added, it appears in the list of available join paths.



4.4 Virtual Fields

Virtual fields are fields that don't actually exist in the table. When a virtual field is requested by the user, it is fetched from an alternate table behind the scenes, and copied into the table. This allows fields from additional tables to be flattened and presented all as one table.

An example of this is geo location information. Typically, in a normalized database, the location ID would be stored in the main fact table, with joining through satellite tables required to fetch location name, country name, and other information. By the use of virtual fields, these location fields can be shown as part of the main fact table, allowing the user to easily see that these fields are available and select or use them.

These virtual fields may be added, edited, and removed from any table using the [virtual fields tab](#) while configuring that table. Limited configuration of the field, as exposed in the table, is possible while editing the virtual field, while all other information is copied from the source table.