PowerPlanner manual

Contents

Installation

Setup and prerequisites

To install PowerPlanner first download and run the setup from the Products\Downloads section of Power-Planner.com (you need to be logged in).

The installer will first try to install the prerequisites:

Microsoft Windows Installer 3.1

Microsoft Windows Installer 4.5

Microsoft Windows PowerShell 1.0

Microsoft .NET 3.5 sp1 full

Microsoft .NET 4.0 full

Microsoft Office Excel 2010 (Not installed by setup)

Microsoft Office Primary Interop Assemblies

Microsoft Visual Studio Tools for Office 2010 Runtime

Microsoft PowerPivot for Excel

If the installation of any prerequisite fails try to download and install it manually from the Downloads section of Microsofts' website (http://www.microsoft.com/download). Please make sure that you have at least 3 Gb of free disk space for the prerequisites (the most common cause of installation failure is the lack of disk space). If you install a prerequisite manually make sure to install the correct version based on the operating system version and office version.

After all the prerequisites are installed, you can select from the following install components:

Microsoft SQL 2008 R2 Express with Tools

PowerPlanner Examples

PowerPlanner for Excel 2010 / PowerPlanner for Excel 2013 (depending on Office version)

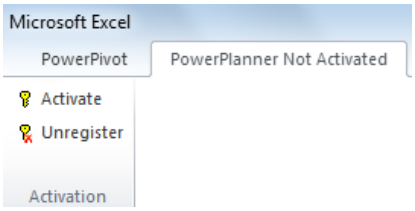
You only need to select SQL Express if it's not already installed.

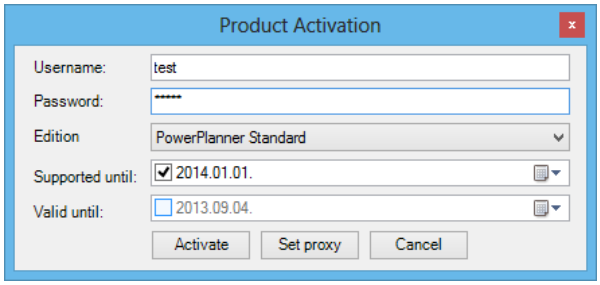
Finally you need to specify setup parameters for SQL Express and for restoring the Examples model.

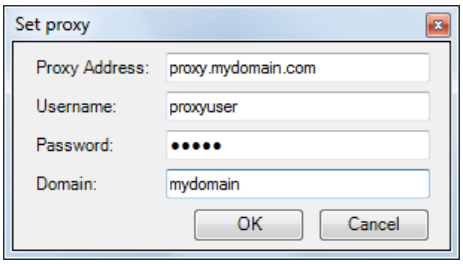
Licensing and activation

After installing PowerPlanner you need to activate it. For this open Excel, click on the PowerPlanner tab then click on Activation. In the activation dialog enter the username and password you registered on Power-Planner.com and select the correct product edition, supported and valid until dates (if applicable). For supported and valid until dates please refer to the Licences section of our website (requires login). Activation requires a working internet connection. If you access the internet through

a proxy set up the proxy details in the proxy section (if you don't know how to do this contact your network administrator).







IMPORTANT NOTE:

Before uninstalling PowerPlanner (e.g. re-using license on another computer or reinstalling operating system, or upgrading to another edition) always unregister your license at the activation group of the PowerPlanner tab in Excel. Otherwise you won't be able to re-use your license. If you forgot this you can unregister a licence at the Licences section of our website (requires login).

Restoring examples manually

You can also restore the sample database Budget.zip with the name Budget manually. (see Restoring databases section for details).

The examples contain a Budget.xlsx file which contains an example Budget PowerPivot model, with Sales, COGS, Margin, Salaries, P&L plan.

The Import from ERP folder contains csv source files for importing actual data into the model. The Account Import.xlsx file demonstrates importing the Account.csv file. To change it for other files / data sources edit the Data \ Connections \ Import from ERP Account.csv data source in the excel file. To use another csv file change the folder in the datasource from the installation folder to your own folder, and the 'Account.csv' to your own csv file name at all places in the connection string. You also need to change the schema.ini file according to the layout of your csv file in the import folder. To use other data sources (SQL, Oracle, Access, etc.) use Excel's data source editor to replace the data source with your own (Definition \ Browse \ New Data source).

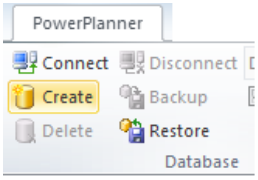
For more details on importing data check the import section.

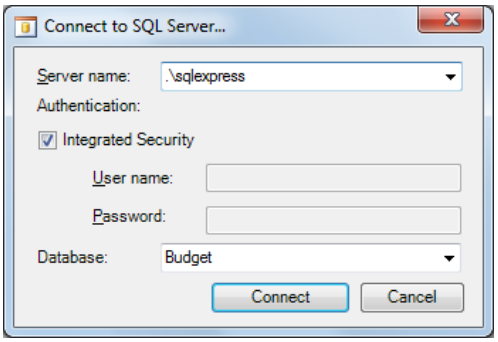
Building PowerPivot Budgeting models

Database

Creating database

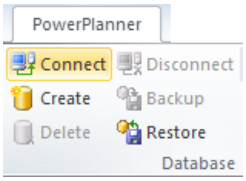
The first step in creating a PowerPivot budgeting model is to create the database. For this click on the create button in the Database tab of PowerPlanner. Then in the dialog box specify the name of the new database, the instance name of the SQL server (.\sqlexpress by default) and the authentication details for the SQL server (Integrated security or user: sa , password: Sa123456 by default)

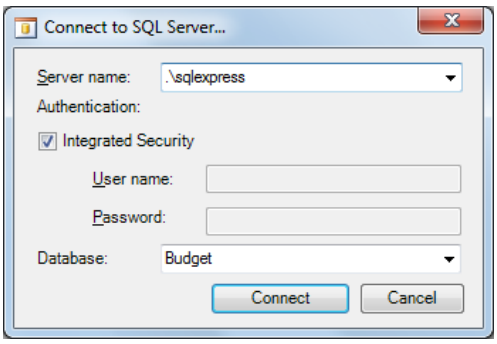




Connecting to existing database

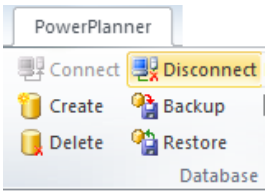
To connect to an existing database press the connect button (only active when no databases are connected), then specify the SQL connection details as above. Connecting to a database enables editing its tables, and connecting to models.





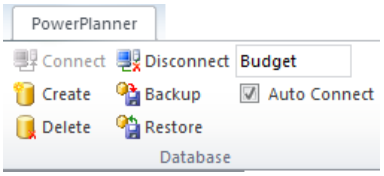
Disconnecting

To disconnect from an existing database press the disconnect button (you need to disconnect before you can connect to another database).



Auto-connect settings

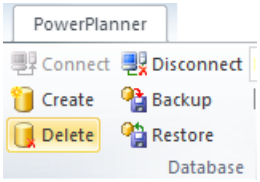
If the auto connect option is set, PowerPlanner automatically connects to the database and the model (if there is any model in the workbook), once a workbook is opened. **Important: Auto Connect is only supported using Integrated authentication (not supported using SQL authentication with user / password).**



Deleting database

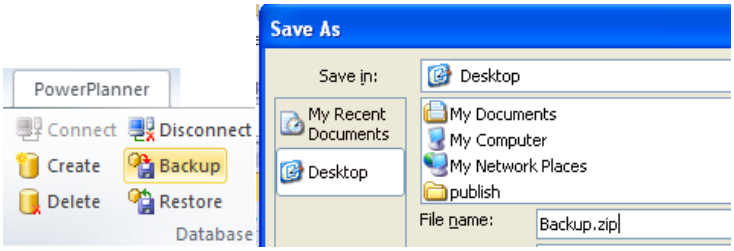
To delete the database click on the delete button (you need to be connected to the database first).

WARNING: The database will be permanently deleted.



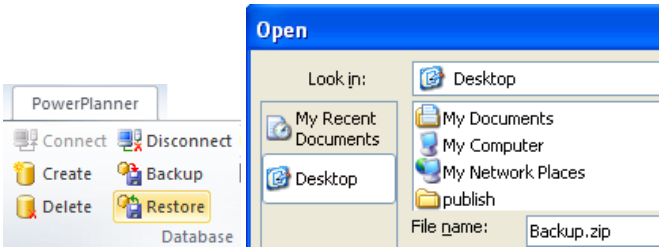
Backing up database

Databases can be backed up pressing the backup button (you need to be connected to the database first), and selecting the backup file name (.zip). It's important to make regular backups of your database to avoid any data loss. Backup operation creates a copy of the database in a compressed zip file.



Restoring database

To restore a previously backed up database press the restore button and then select the zip file containing the backup.



WARNING: Restoring permanently overwrites the existing database with its backup version.

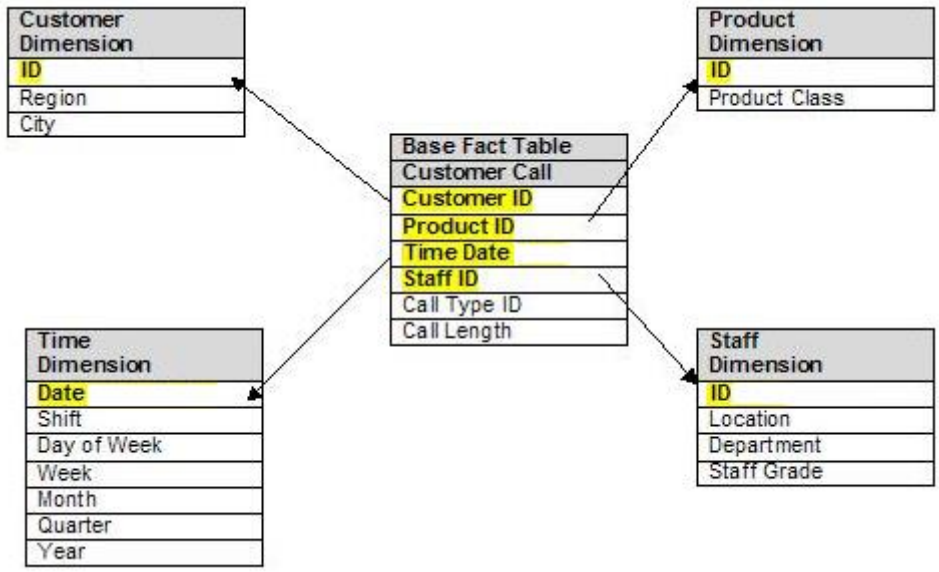
Tables

Star schema, Snowflake schema, Naming conventions

The next step in creating a PowerPivot Budgeting model is to create or import the base tables for the PowerPivot cube. To make a good cube, tables need to be in a special arrangement called the Star or the Snowflake schema.

Star Schema

Most business intelligence data warehouses use what is called a dimensional model, where a basic fact table of data e.g. sales or support calls is surrounded and linked with other tables holding the dimensions of the fact table.



This particular fact table has four main dimensions - Customer, Time, Product and Staff.

These dimensions are then linked to the fact table through indexes (highlighted in yellow)to enable tables to be joined to permit fast queries, reports and data consolidations to be carried out.

For example, how many transactions for product x have we had this quarter?

This data model or schema is simple, allows fast retrieval, can be readily extended without changing all the existing standard reports and queries. The disadvantage is that there is some data redundancy which could cause inconsistency if not all of the redundant data is kept up to date.

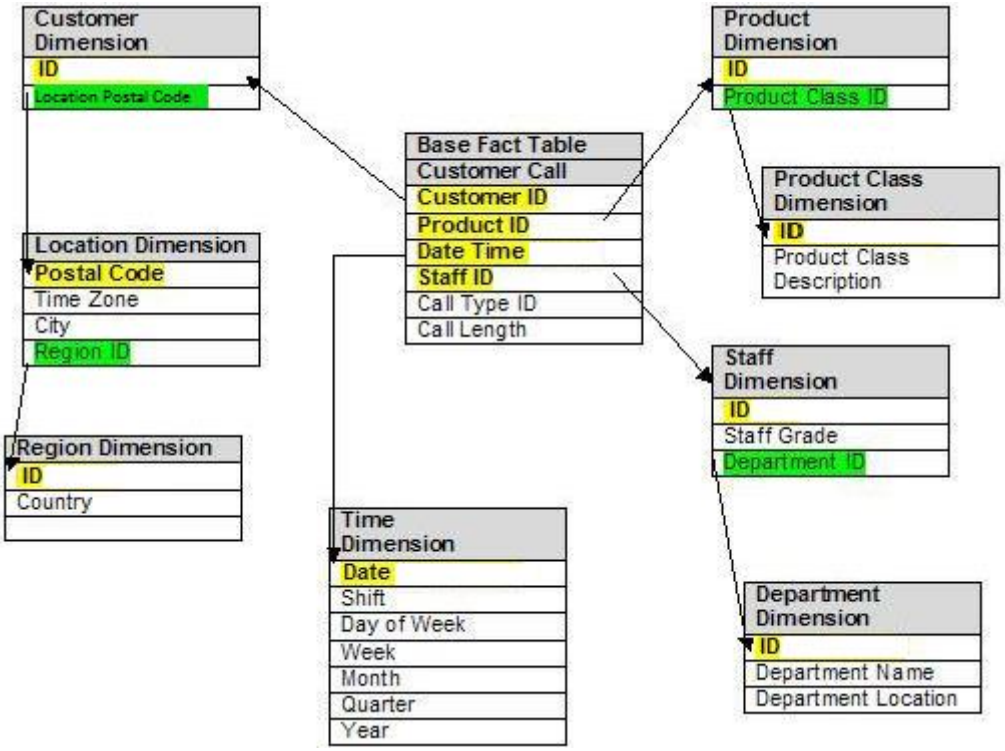
Snowflake Schema

The next kind or model is called a snowflake model and is very similar to the above schema except that some of the redundancy in the dimensions is removed by using what is called data normalised tables.

See the following example of a snowflake schema. Primary and foreign keys (primary keys are highlighted in yellow and foreign (linking) keys in green) are used to join up the tables to the central fact table and other dimension tables.

Remember, data normalisation is a three step process to ensure a that every piece of data is uniquely identified and there optimum data redundancy. For example, in the Location table the value of a country would be repeated many times. Now by introducing a Region dimension table the value of a particular country would be repeated less often.

To remove all country redundancy completely you would need a further table for Country. Often the Snowflake model does not go all this way to normalise the data fully.

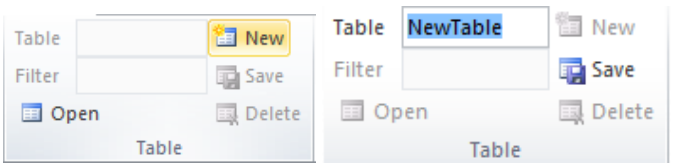


Naming conventions

As you can see from the above example there is a naming convention for naming columns in parent and child tables. The related key column in the parent table always starts with the child table name, then a space then the name of the child column. E.g. The column in the fact table connected to the key column in the Customer table (ID) is called Customer ID. This is important because PowerPlanner can automatically detect the relationships between parent and child tables based upon the naming convention. So you should use this naming convention in your PowerPivot Budgeting model so that PowerPlanner can automatically detect relations between your tables.

Creating table

Tables can be created pressing the New button in the Tables group in the PowerPlanner tab in Excel (the Table tab is only active when you're connected to a database, it always shows settings related to the table which is currently selected in Excel), then entering a name for the table in the textbox on the left of the new button. Tables are represented as Excel lists.

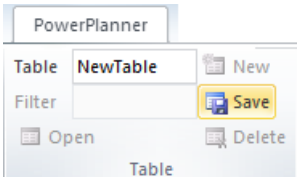


Renaming table

The table name can later be changed. If it's a new unsaved table, then the name can be changed without consequences. If it's an existing table the table will be renamed in the database as well, and it needs to be changed everywhere in the model as well, so be careful when renaming a table.

Saving table

Once a table is created and edited it can be saved into the database by pressing the Save button. If it's a new table the table will be created in the database and the data will be saved in it. If it's an existing table the changes made in columns will be saved in the database and the data will be saved. Saving clears all previous data (taking filters into account) and saves the new data from Excel. Lists created by Excel (new list / convert range to list) can also be saved as tables in the database (in this case the name needs to be specified after selecting the table.



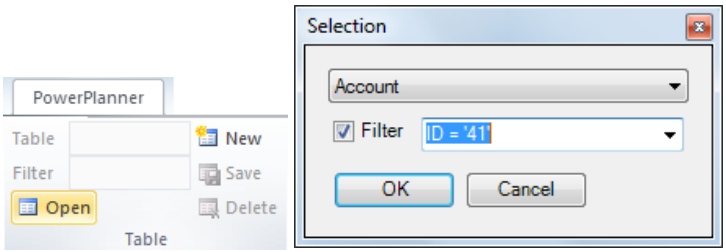
If the table contains references to other tables PowerPlanner will ask for refreshing external references and extending tables to the length of the longest external table so that all referenced values are in the table.

Editing table

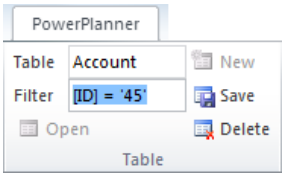
Users can edit both the columns and the data in existing or new tables.

Opening table, Using Filters

Existing tables can be opened by clicking on the Open button (it's only active when there is no table selected in Excel), then selecting the table from the database. Filters can also be set on the table in the standard SQL syntax. When using filters only the filtered part of the table data is loaded. The filter is also taken into account when saving the table (only the filtered part is cleared before saving new data).



Filters can be edited on an open table as well, by changing the text in the filter box on the table panel. In this case the table is reloaded with new filters.

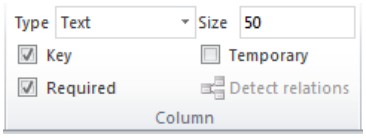


Editing columns

You can edit columns in the table by selecting the column in Excel. After this the columns tab will become active and you can change the column type, size and key, required, temporary options.

The column type can be of the following Text (varchar), Integer (int), Number (float), Date (datetime), True/False (bit). The size option is only available for the Text type and defines the maximum length of Text that can be saved in the column. The key option sets the column as a key in the table (must be

unique). The required option sets that the column must not be empty (key columns are also required). The temporary option means that the columns is not saved in the database table, but only used in Excel for temporary calculations descriptions etc.



Columns can also be created, deleted and renamed. This is done by simply renaming, deleting or inserting new columns in the Excel list that represents the table in Excel.

	A	B	C	D	E	F
1	ID	Account Name	Account Group	Account Group L3	Account Group L2	Account Group
2	41	Sales Revenue	Gross Profit	Gross Profit	Income from Operations	Net Profit

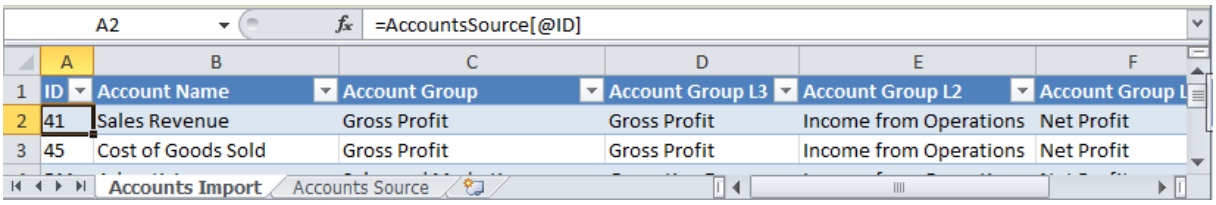
Editing table data

Data in the table can simply be edited by changing the values in the Excel list representing the table. Formulas and references to other tables can also be used.

Importing data

Importing data can be set up by referencing one or more source Excel list that have an external data source. External tables can be referenced using the =TableName[@[Field name]] formula.

For example in the examples the Account Import.xlsx file contains a two tables one called Account which is basically the account table opened, and AccountsSource an Excel list with an external csv file datasource (set up using the data connections wizard in Excel). And all columns of the Account table are referenced using the above formula from the AccountsSource table.



When saving the table PowerPlanner ask if the external tables referenced should be refreshed, and the table should be extended to the length of the longest referenced table to include all external values. Other Excel formulas can also be used for temporary calculations these are also copied down when extending the table. When the table is extended the formulas from the first row are copied down. Import also takes into account the filters set on the list, so you can apply filters to import parts of the full table separately.

Automated import (Modeler edition only)

Automated import can be setup for automatically executing the import procedure described in the import section on multiple Excel workbooks at a regularly scheduled interval. This option is only available in the modeler edition.

To set up the automated import first create a new text file.

Open the text file and enter the following:

Import.exe PathToImportFolder

Where the PathToImportFolder is the path of the folder where the Excel workbooks are which imported.

Save the text file as import.bat

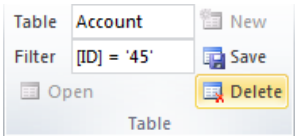
Open the windows task scheduler from Start menu \ All programs \ Accessories \ System tools \ Scheduled Tasks and add the import.bat file you just prepared as a new task with the required scheduling options.

When the import runs each of the workbooks in each of the subdirectories of this folder are opened and all PowerPlanner tables in the workbook are imported (saved with data source refresh and table extension turned on).

Problems during import are logged in the Import.log file, you can open it by opening a command prompt in windows (Start Menu \ Run \ cmd) and typing in Import.log and enter.

Deleting table

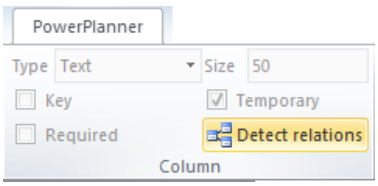
Tables can be deleted from the database using the Delete button in the Table group (the table needs to be selected first).



WARNING: This operation permanently deletes table from the database.

Detecting relationships

The Detect relationships button detects relationships based upon the naming convention specified in the naming conventions section. It saves the user the work of manually setting up relationships in the PowerPivot Model. Detecting relationships only works if there is a connected PowerPivot Budgeting Model.



PowerPivot Budgeting Model

Creating model

PowerPivot models define the PowerPivot cube. It defines which tables are used in the cube, what the relation between these tables is, and what formulas are used to calculate KPI-s.

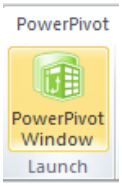
The model is saved in the Excel workbook itself (one model / workbook). So it's advised to first create a master Excel workbook with the model in it, and create all other workbooks based upon this. You should always make changes in the model in the master workbook, and copy the changes to your other workbooks either manually (described in publishing model manually section) or automatically

(described in publishing workbooks automatically section, this option only works with the modeler edition).

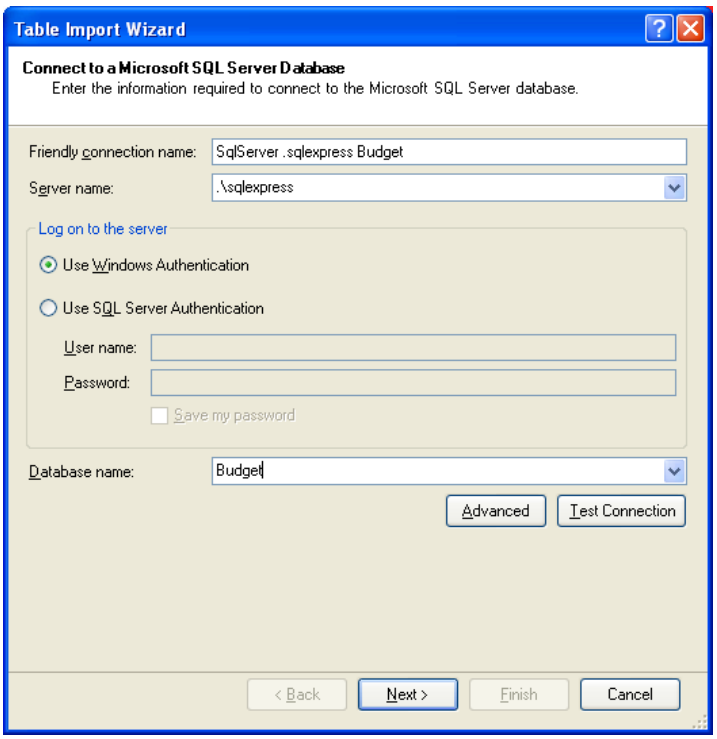
PowerPivot models can be created using the standard Microsoft PowerPivot tools in Excel.

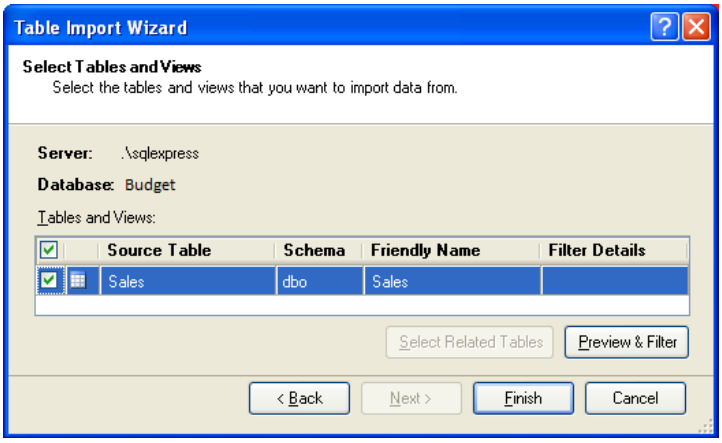
The first step for creating a new model you need to create a new workbook.

Then click on the PowerPivot window button in the PowerPivot tab in Excel.



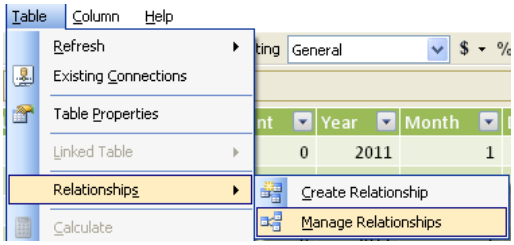
In the PowerPivot window connect to the SQL server (similarly to described in the connecting to database section), then select the tables to import (if you already have a datasource set up and you want to add new or adjust existing tables click on to Table\Existing connections\Open instead).





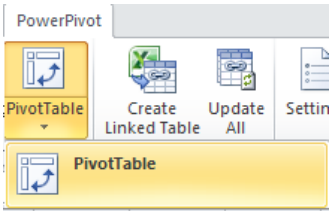
You can select columns required and set filters using the Preview & Filter button.

After this you can set up relations manually at the relations menu (if you have followed the naming conventions in the naming conventions sections this part is not necessary).

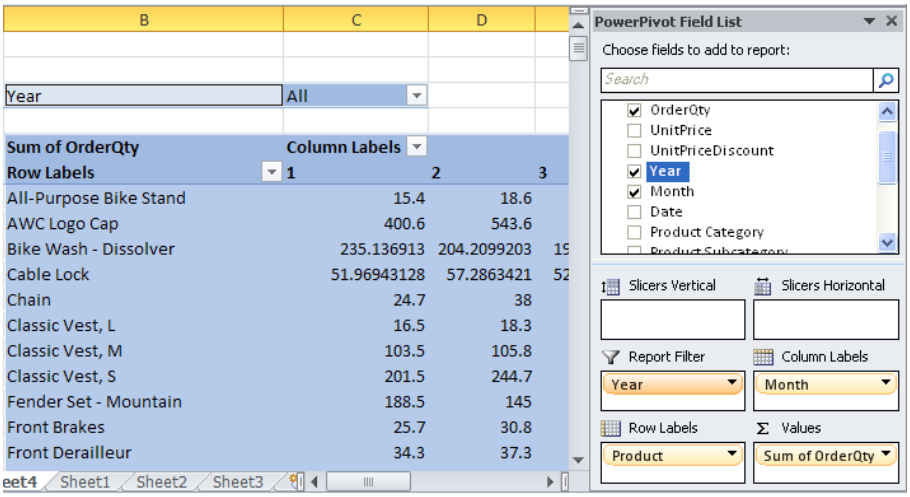


Once finished close the PowerPivot window. If you've followed naming conventions you can detect relations between tables automatically (as described in the detect relationships section).

After creating the model you can insert a PowerPivot PowerPivot PivotTable based upon the PowerPivot model created by clicking on the insert PowerPivot PivotTable button in the PowerPivot tab.



You can edit the layout of the PowerPivot PivotTable in Field List on the left side. Here can drag fields from the model to the appropriate rows, columns and filters of the PowerPivot PivotTable.

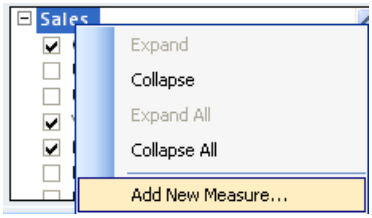


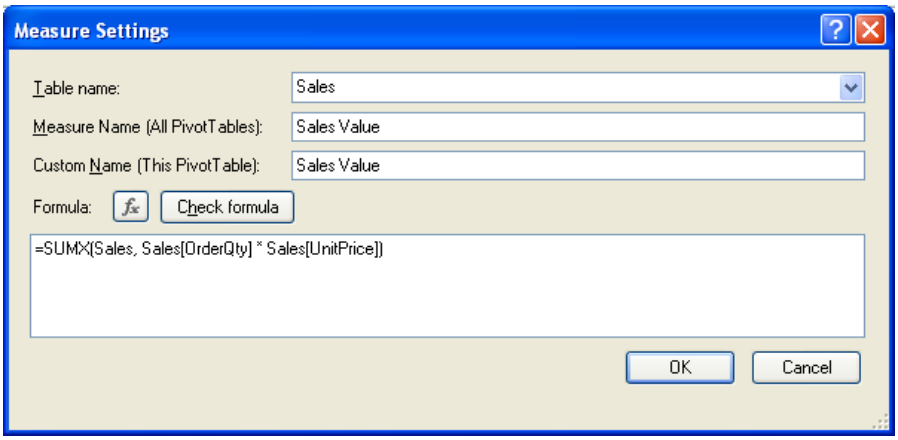
The Sales Plan video in the Demos section of power-planner.com demonstrates this process.

You can also add calculated KPI-s as a new measure column to any table. Formulas used for calculated measures are based on the standard Microsoft PowerPivot DAX syntax.

For example you can add a calculated measure to calculate the Sales Value the Sales table using following the formula:

=SUMX(Sales, Sales[OrderQty] * Sales[UnitPrice])





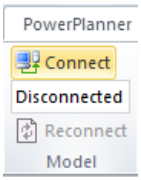
This means to summarize the ordered quantity multiplied by unit price in the sales table. The SUMX function is needed so that the product of these two measures is correct on all levels (if it would be left out the summarized quantity would by multiplied by the summarized price which would make no sense).

For a complete reference and examples of formulas using the DAX syntax refer to the appropriate Microsoft Websites (e.g. http://go.microsoft.com/fwlink/?LinkId=180638).

For more information on building PowerPivot models visit the appropriate Microsoft websites and forums.

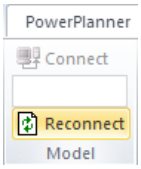
Connecting to existing model

Once the model is created using the PowerPivot tools you need to connect to it to be able to use the PowerPivot PivotTable editing features and to be able to publish the model on the SQL server (modeler edition only). Connecting to the model can be done by pressing the Connect button in the Model group of the PowerPlanner tab in Excel. Connecting connects to the PowerPivot model in the current workbook (if any), you don't need to specify any connection information. If there is no PowerPivot Model in the workbook connection will be unsuccessful. Once connected the features in the Model group, Data and Security (Modeler edition only) groups will become available, and the name of the Model (published models only) will be shown in the textbox beside the Connect button.



Reconnecting

You can reconnect the model using the Reconnect button in the Model group. This operation manually reloads the changes in the PowerPivot model. PowerPlanner automatically reloads changes before any operation referring to the model (e.g. writeback, goal-seek, detecting relationships etc.), so this operation only needs to be run to reflect changes immediately.



Publishing model

As mentioned before it's strongly advised to make a master Excel workbook with the master model in it and make all changes on the model in that. Reflecting changes in the model in other workbooks can be either done manually in the Standard edition or automatically using the Modeler edition.

Publishing manually

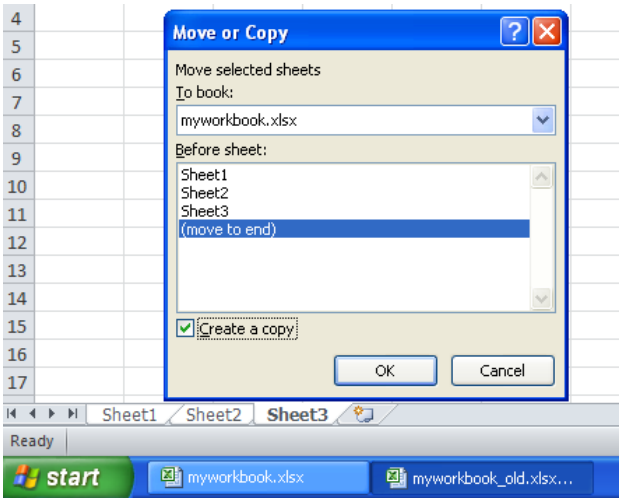
To update your Excel workbooks to your latest model manually:

First rename the workbook you want to update (e.g. rename myworkbook.xlsx to myworkbook_old.xlsx).

Then make a copy of the master workbook with the name of the workbook you want to update (e.g. copy masterworkbook.xlsx to myworkbook.xlsx).

Then open both workbooks and copy all sheets form your old workbook (e.g. myworkbook_old.xlsx) to the new copy of the master workbook (e.g. myworkbook.xlsx).

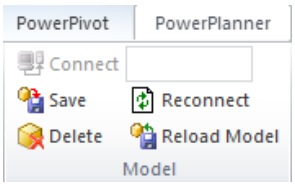
Finally delete any sheets remaining from the original master workbook.



What this does in short is that the model from the master workbook is copied by copying the Excel file, and then the data is copied from the old workbook by copying the sheets.

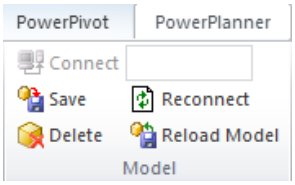
Publishing automatically (Modeler edition only)

If you have the modeler edition then you can save time with skipping this manual process, by publishing the model on the SQL server. Basically what this does it saves your current Excel workbook on the SQL server as a master workbook. You can perform this by pressing the Save button in the Model group of the PowerPlanner tab in Excel (you need to be connected to the model for this to be active). If it's a new model that has not yet been published to the server yet, you need to specify a name for the model.



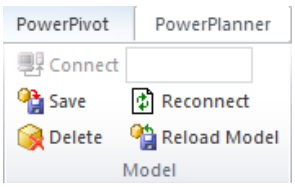
Reloading model (Modeler edition only)

Reloading the model refreshes the model in the current workbook with its latest published version. It does exactly what's described in the Publishing manually section with the model's master workbook downloaded from the server. You can perform this by pressing the Reload button in the Model group.



Deleting model (Modeler edition only)

Published models can be deleted from the SQL server using the Delete button in the Model group (the model needs to be connected for this button to be active).

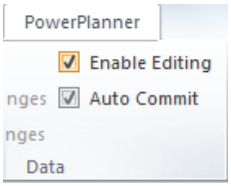


Data editing

After the model is connected the Data, Goal-Seek and Calculate groups become enabled and data in the PowerPivot PivotTables in the Excel workbooks can be edited.

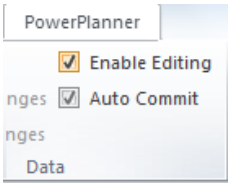
Enable pivot table editing

Editing of the currently selected PowerPivot PivotTable can be enabled using the Enable Editing checkbox in the Data group. Only non-calculated cells can be changed by normal editing, to change calculated cells use the Goal-seek feature (see in Goal-seeking section).



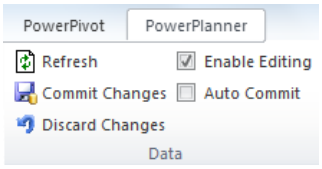
Auto-commit settings

If the Auto commit setting is enabled using the Auto commit checkbox, data edited in the PowerPivot PivotTable is automatically saved on the SQL server.



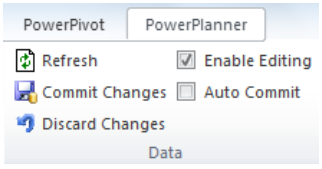
Committing changes

If the auto commit setting is disabled, changes made in the PowerPivot PivotTable are stored temporarily in the PowerPivot PivotTable until the Commit button is pressed. This enables faster data entry and pasting large sets of data before saving it on the server.



Discarding changes

If the auto commit setting is disabled, way changes made in the PowerPivot PivotTable can also be discarded, if the user decides to undo them.



Using the calculation panel

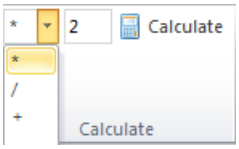
The calculation panel can be used for adjusting the currently selected cells in a PowerPivot PivotTable quickly using mathematical operators. The operators that can be used in the calculation panel are:

*	multiply the selected cells by *n*
/	divide the selected the cells by *n*

+	increase selected cell by *n*
-	decrease selected cell by *n*
%	increases the selected cells by *n*%
=	set the total of the selected cells to *n* (by multiplying them with the same factor)
<-	set each cell to *n*

Where *n* is the number in the entered in the textbox of the calculation panel (can be negative)

The calculation is executed by pressing the calculate button.



Using quick edit formulas

The quick edit formulas serve the same purpose as the calculation panel, but they can only be used to adjust a single cell (can be a total). If you enter a quick edit formula in a cell instead of a number it performs a calculation on the original number in the cell. The quick edit formulas that can be used are:

inc*n*	increase selected cell by *n*
inc*n*%	increase selected cell by *n*%
dec*n*	decrease selected cell by *n*
dec*n*%	decrease selected cell by *n*%
mul*n*	multiply selected cell by *n*
div*n*	divide selected cell by *n*

where *n* is an arbitrary number

Top-down planning

Using PowerPlanner not only detail cells can be changed, but also aggregated values (e.g. any total or average in the PowerPivot PivotTable). Top-down planning multiplies all the detail cells that add up to the total by the same number so that the total reaches the desired value. In other words it keeps original the distribution of the details. If the original total is 0 or empty then the details are equally allocated.

Year total	mul2
Jan	1
Feb	2
Mar	3

->

Year total	12
Jan	2
Feb	4
Mar	6

Average price	10
Product1	
Product2	
Product3	

->

Average price	10
Product1	10
Product2	10
Product3	10

Goal-seeking

Calculated cells can't be edited just like normal cells in the PowerPivot PivotTable, but they can be edited using Goal-seeking. What goal-seeking does is that is that it adjusts selected drivers of the given calculated KPI so that it reaches the desired value (it multiplies the details of all selected cells by the same number, to reach the goal).

For example in case of a Sales calculation, where:

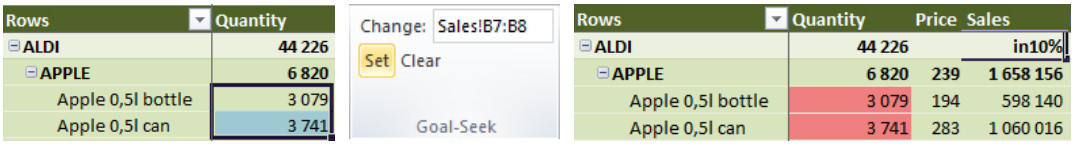
Sales Value = Ordered Quanity * Unit Price

either the Ordered Quantity or the Unit Price can be adjusted to reach a certain goal in the Sales Value.

Goal-seeking also works on aggregates (e.g. total cells). In this case top-down planning is used to adjust the drivers. When goal-seeking on a total cell, a selection of the driver details can also be adjusted only. For example increase the total sales value by 10% by increasing the sales quantities of only a certain product category (the others are left intact).

Multiple drives can also be selected. In this case all selected drivers are multiplied by the same number to reach the goal.

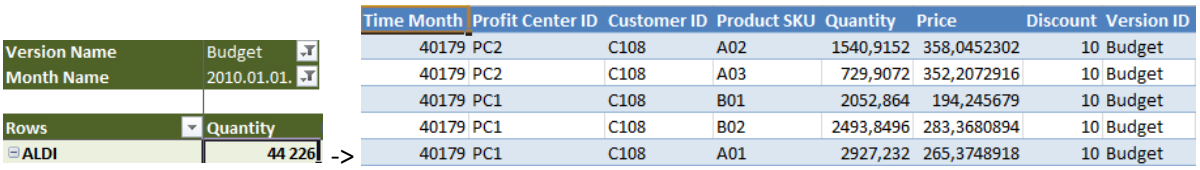
Goal-seeking can be performed by first selecting the source drivers to be changed in the PowerPivot PivotTable, then pressing the Set button in the Goal-Seek group of the PowerPlanner tab in Excel (if you want to clear the selection and select a new source, you can press the Clear button). After this the selection becomes highlighted, and you can change the target cell.



You can watch a complete example of Goal-seeking in the Sales Plan demo video in the Demos section of Power-Planner.com

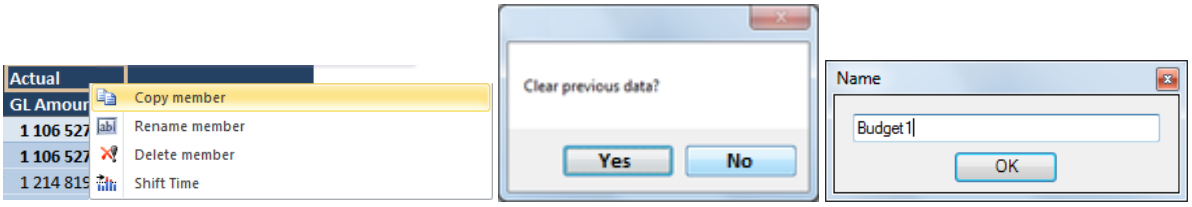
Drill-through

To find out where a figure in a cell originates from, you can use the drill-through function by double clicking on the cell. This opens a detailed list of rows from the base table which show where the figure originates from. This is usually interesting for aggregates, to check what the detail rows add up to the total. This operation only works on non-calculated cells.



Copying items

For making quick simulations and rolling forecasts, it's often required to copy versions (or any other item such as products, customers, etc.). This can be done by selecting the item in the PowerPivot PivotTable, then right click with the mouse and select Copy member from the context menu. Then PowerPlanner asks whether to Clear previous data. This option deletes previous data for the copied item. It can be used when data in the original item has changed and you want to re-copy it. Then you need to specify a name for the new item. For Example Actual data can be copied to the Budget1 version to start a new roll-forward plan based upon actual data.

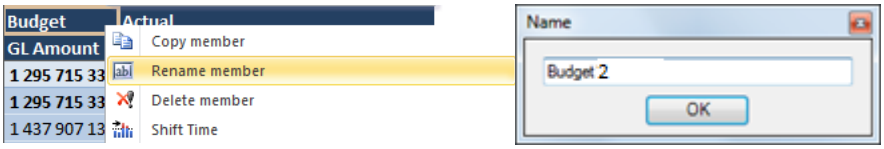


Finally PowerPlanner asks whether to Shift Time as well (this is described in the Shifting time Section).

Renaming, Deleting items

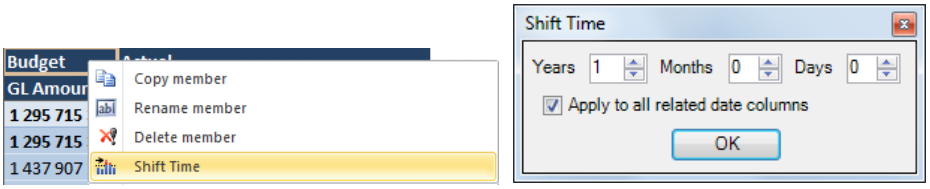
Items can also be renamed selecting the Rename member menu item, and specifying a new name for the item. This renames the item in the dimension table and all related fact tables. For example you can change the name of Budget1 to Budget2.

Items can also be deleted using the Delete member menu item. WARNING: This deletes the item from the dimension table and all related fact tables.



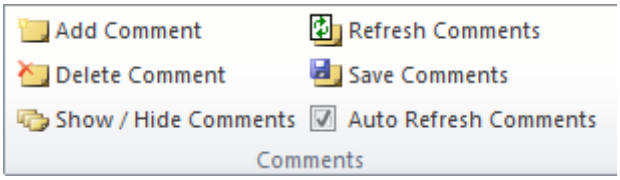
Shifting time

Often when copying an item (e.g. a Version) the Time needs to be shifted too, for example when you want to copy last years actual data as a base for this years budget. In this case after copying the Actual version to Budget, you need to shift time by 1 year too. This can be done by right clicking the copied item and selecting the Shift time menu item (it can also be performed together with the Copy operation, see Copying items section for details). After this you can select the time period to shift by (Years, Months, Days), and whether to apply the shift to any other columns in the table with date type. If the Apply to all related date columns option is not selected PowerPlanner will ask whether or not to apply the shift for each date column in the table one by one.



Comments

Comments can be used to mark changes and share them with other users. A comment can be added to the current PivotTable cell using the Add Comment Button in the Comments group. Coments in the current worksheet are saved in the Database after pressing the Save Comments button. Comments saved in the database can then be displayed in all PivotTable cells with the same attributes in the current worksheet using the Reload Comments button. Comments can also be automatically reloaded when reloading data by ticking the Auto Refresh Comments checkbox. A comment in the current PivotTable cell can be deleted using the Delete Comment button. The Show / Hide Comments Button displays or hides all Comments.

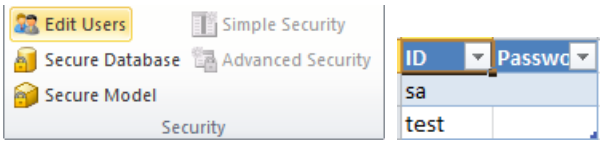


Security (modeler edition only)

Security settings enable the user to create users in the SQL Database and specify access rights for these users. Security settings are only available in the modeler edition (you need to be connected to a model for this group to be active).

Editing users

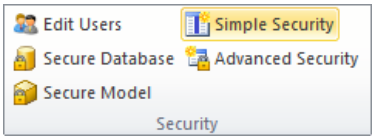
To edit users click on the Edit users button in the Security group of the PowerPivot tab in Excel. This opens a list of users in Excel that are assigned to the current database. The list contains the usernames and passwords. The passwords for existing users are shown as blank (if you change these the password will be reset to the supplied password upon saving). You can enter new users along with passwords (by adding new rows), or delete existing users (by deleting rows). To save users press the Save button in the Table group just like saving a table. These users are created in as sql users on the SQL server and added to the database with limited rights. These users can then be used to connect to the database with and to specify access rights for in the Simple and Advanced Security Settings. The sa user is the built in administrator user with full rights to all databases. You should use this user to connect to the database as the administrator user. The default password for the sa user in the SQL server instance installed with the software (.\sqlexpress) is Sa123456, you should change this after installation using the Edit users feature.



Simple security

When applying Simple security settings on a table, you can assign each row in the table one user that has access to read only / read-write access to the row. It can be used in simple scenarios, when each item has one single responsible user. For example when each region in the regions table has one sales manager, who can view / edit the data and there is an administrator user who can view / edit everything (sa). Simple security can be activated by pressing the Simple security button in the Security group. This option is only active if you're connected to a model, and there is a table selected in Excel. The table is extended with two Security columns: User ID, Write. The User ID columns contains name the user who can read data related to the current row (if it's a dimension table, read access is inherited to all related fact tables). If the user column is left empty only the administrator user will be able to access the row. If the Write column contains True then the user specified in the user column has read/write access to the row (and all related child rows), otherwise the user has read only access.

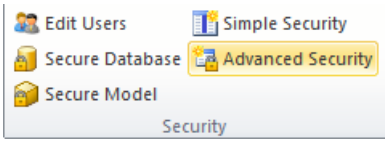
If you save a table with Security settings defined in it PowerPlanner asks if the model should be secured. This option Secures the Database, and Secures the model (see Securing database and Securing model section). Removing Simple security settings can be done by deleting the security columns.



Region	User ID	Write
East	jfreddy	
West	ksmith	TRUE

Advanced Security

Advanced Security settings allow multiple users with read only or read/write access to be assigned to the distinct values of a column of a table. You can apply Advanced Security settings by clicking on the Advanced Security button button in the Security group. This option is only active if you're connected to a model, and there is a column of the table selected in Excel (non-security table). A new security table is created (with the name Security.TableName.ColumnName where TableName and ColumnName correspond to the original table and column name), where you can assign multiple users with read only / read-write access for each distinct column value in the original table. If there is already an existing Security table PowePlanner ask whether to create a new Security table and overwrite the existing, or open the existing Security table. If a new security table is created, PowerPlanner asks if the original column values should be copied (later this needs to be maintained manually, values omitted will only be accessible by the administrator user). The Security table contains 3 columns: ColumnName, User ID, Password. ColumnName (corresponds to the original column name) contains the distinct column values from the original table. The User ID column contains name the user who can read data related to the current column value in the original table (if it's a dimension table, read access is inherited to all related fact tables). If the Write column contains True then the user specified in the user column has read/write access to read data related to the column value (and all related child rows), otherwise the user has read only access. For example if jfreddy has read access and ksmith has read-write access to the to the East region it would look like this:



PowerPlanner

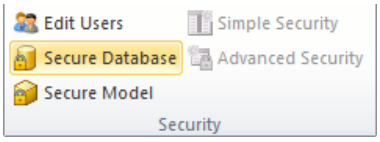
Table Security.Regions.Region

Region	User ID	Write
East	jfreddy	
East	ksmith	TRUE

If you save a table with Security settings defined in it PowerPlanner asks if the model should be secured. This option Secures the Database, and Secures the model (see Securing database and Securing model section). Removing Advanced Security Settings can be done by deleting the Security table.

Securing database

Securing the database applies the access rights specified in the Simple and Advanced security settings of tables to the model. Whenever the security settings are added or removed from a table, the database needs to be Secured again (otherwise changes in security settings will not be applied). When user rights are edited within an existing Security table or a table with Simple Security Settings, the database does not need to be secured again.



Securing model

Securing the model secures the current Excel workbook that contains the model. What it does is that it removes any trace of the administrative connection details (administrator user, password) from the

Excel file, to make sure that it can only be accessed securely. Each model published on the server or offered to be edited by contributor users securely, must be secured first (if the model is not secure contributor users will not be able to edit it). The model only needs to be secured once, regardless of changes in the Security settings.