

# **PLANNING APPLICATION**

**with Microsoft Excel  
and SQL Server**

Sergey Vaselenko

# **PLANNING APPLICATION**

## **with Microsoft Excel and SQL Server**

Written by Sergey Vaselenko

This e-book shows how to configure a planning application built with Microsoft Excel and SQL Server.

You may download the source codes, install the application into your database, and repeat all the steps.

If your company uses Microsoft Excel to get and share business data, you may save a lot of time and money.

# Contents

---

<b>Contents</b> .....	<b>1</b>
<b>Introduction</b> .....	<b>3</b>
<b>Chapter 1. Scenario</b> .....	<b>4</b>
Business Case.....	4
Model .....	5
<b>Chapter 2. Members</b> .....	<b>6</b>
Units .....	6
Dimensions .....	7
Accounts.....	8
Times.....	14
Categories.....	17
Entities .....	18
Regions.....	18
Hierarchies.....	19
Relations.....	20
<b>Chapter 3. Forms</b> .....	<b>22</b>
User Forms .....	22
Offline User Forms.....	27
Web Forms .....	29
Form List.....	30
Form Settings.....	31
Form Rowsets .....	36
<b>Chapter 4. Permissions</b> .....	<b>39</b>
Database Permissions .....	39
Form Permissions.....	40
Member Permissions .....	40
Closing Periods.....	44
<b>Chapter 5. Integration</b> .....	<b>45</b>
Copying Data .....	45
Importing Data.....	45

<b>Chapter 6. Data Management</b> .....	<b>46</b>
<b>Chapter 7. Distribution</b> .....	<b>47</b>
Distributing Workbooks .....	47
Distributing using Connect Credentials.....	49
<b>Conclusion</b> .....	<b>56</b>
<b>FAQ</b> .....	<b>57</b>
<b>Appendix A. Installation</b> .....	<b>60</b>
Download .....	60
Download Package.....	60
Installation.....	60
Test Users .....	62
Uninstalling.....	62
<b>Appendix B. Connections</b> .....	<b>63</b>
<b>About the Author</b> .....	<b>66</b>

# Introduction

If your company uses Microsoft Excel to get and share business data, you have to continue reading.

You may get great benefits and save a lot of time and money.

This e-book shows how to configure a planning application built with Microsoft Excel and SQL Server.

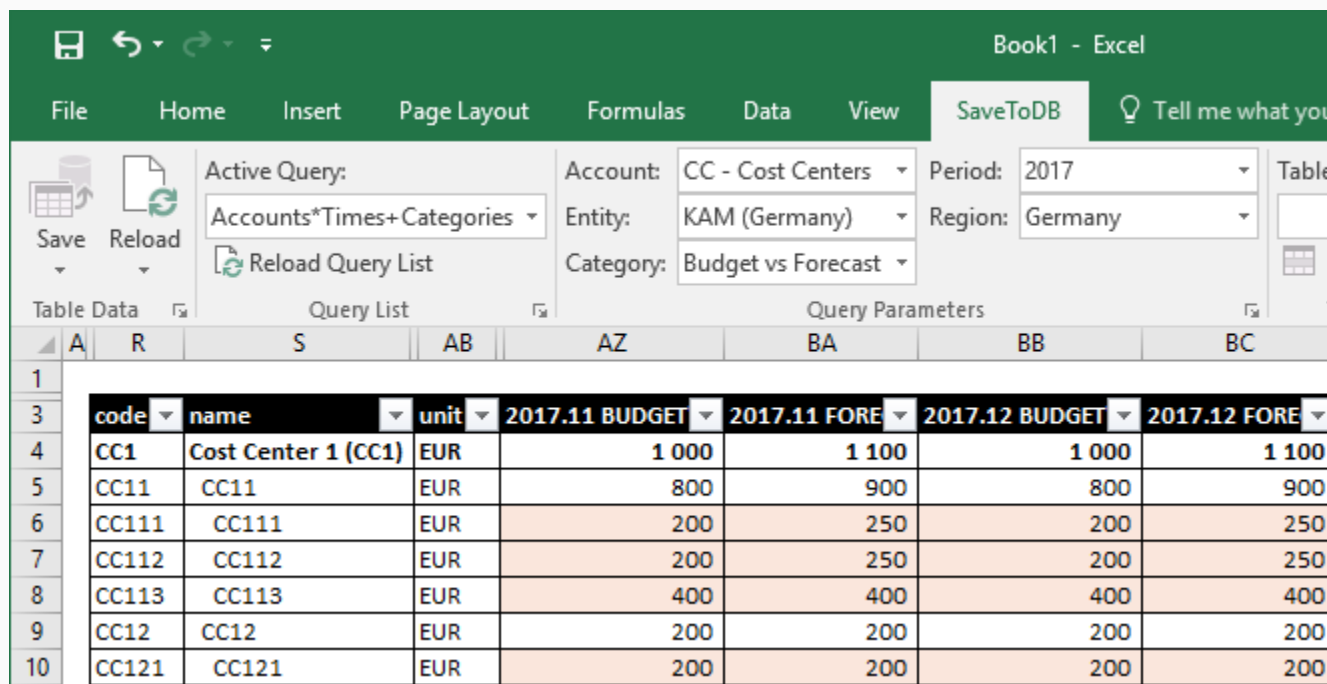
The application uses the SaveToDB Enterprise add-in for Microsoft Excel to work with SQL Server.

You may download the trial version of the SaveToDB add-in and the source code of the planning application to install it in your environment.

We will do typical steps for such class applications:

1. Configure dimensions.
2. Configure members.
3. Configure forms.
4. Configure permissions.
5. Import budget and actual data.
6. Copy budget and actual data to forecast.
7. Distribute workbooks to users.

As a result, your business users may use forms like this to work with source and consolidated data:



The screenshot shows the Microsoft Excel interface with the SaveToDB add-in. The ribbon includes File, Home, Insert, Page Layout, Formulas, Data, View, SaveToDB, and Tell me what you want to do. The SaveToDB ribbon has a 'Save' button, a 'Reload' button, and a 'Reload Query List' button. The 'Active Query' is 'Accounts\*Times+Categories'. The 'Account' is 'CC - Cost Centers', 'Entity' is 'KAM (Germany)', and 'Category' is 'Budget vs Forecast'. The 'Period' is '2017' and 'Region' is 'Germany'. Below the ribbon is a table with columns for code, name, unit, and budget/forecast for 2017.11 and 2017.12.

code	name	unit	2017.11 BUDGET	2017.11 FORE	2017.12 BUDGET	2017.12 FORE
CC1	Cost Center 1 (CC1)	EUR	1 000	1 100	1 000	1 100
CC11	CC11	EUR	800	900	800	900
CC111	CC111	EUR	200	250	200	250
CC112	CC112	EUR	200	250	200	250
CC113	CC113	EUR	400	400	400	400
CC12	CC12	EUR	200	200	200	200
CC121	CC121	EUR	200	200	200	200

I hope you will enjoy reading and become a fan.

Best regards,

Sergey Vaselenko

November 3, 2017

# Chapter 1. Scenario

## Business Case

We will configure the application for the following business case.

A company has a budget for the current year in a Microsoft Excel workbook.

The company has actual data for January – October in an accounting system.

CFO needs an updated cost forecast for November – December and the entire year.

CFO wants to see detailed and consolidated data, actuals versus forecast and budget versus forecast.

The company has three cost centers with hierarchical structures.

The company operates in Germany, Austria, Switzerland, and France.

The company has four key account managers (KAM) responsible for each country.

The desired workbook should look like this:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y		
1		Jan 2017		Feb 2017		Mar 2017		Apr 2017		May 2017		Jun 2017		Jul 2017		Aug 2017		Sep 2017		Oct 2017		Nov 2017		Dec 2017			
2		Budget	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget	Forecast	Budget	Forecast		
3	Cost Center 1 (CC1)	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1100	1000	1100	1000	1100	1000	1100	1000	1100	
4	CC11	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	900	800	900	800	900	800	900	800	900	
5	CC111	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	250	200	250	200	250	200	250	200	250	
6	CC112	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	250	200	250	200	250	200	250	200	250
7	CC113	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	
8	CC12	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	
9	CC121	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	
10	Cost Center 2 (CC2)	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	
11	CC21	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	
12	CC211	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	
13	CC212	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	
14	CC213	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	
15	CC22	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	
16	CC221	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	
17	Cost Center 3 (CC3)	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	900	1000	900	1000	900	1000	900	1000	
18	CC31	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	800	700	800	700	800	700	800	700	800	
19	CC311	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	150	200	150	200	150	200	150	200	
20	CC312	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	150	200	150	200	150	200	150	200	
21	CC313	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	400	
22	CC32	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	
23	CC321	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	

The table contains Budget vs. Actual for January – October, and Budget vs. Forecast for November – December.

Key account managers must have the possibility to see the budget and actual data by countries in a read-only mode and edit forecast data for November – December.

This example is based on the model created by MultiBase GmbH ([www.multibase.de](http://www.multibase.de)).

MultiBase GmbH has a great team of experts in reporting and analytics, ETL, and database management.

I appreciate the permission to use their model.

## Model

We may define the following model to implement business case requirements.

The target database must contain the following dimensions:

1. Accounts
2. Times
3. Categories
4. Entities
5. Regions

The **Accounts** dimension must contain members like these for each cost center:

- **Cost Center 1 (CC1)**
- CC11
- CC111
- CC112
- CC113
- CC12
- CC121

The **Times** dimension must contain months and a year total like

- **2017**
- 2017.01 ...
- 2017.12

The **Categories** dimension must contain the following categories:

- Budget
- Actual
- Forecast

The **Entities** dimension must contain the following members:

- **All KAMs**
- KAM (Germany)
- KAM (Austria)
- KAM (Switzerland)
- KAM (France)

The **Regions** dimension must contain the following members:

- **All Countries**
- Germany
- Austria
- Switzerland
- France

At this time we would combine the Entities and Countries dimensions. However, these are different business entities. So, we have two dimensions to have the possibility to extend the model in the future.

## Chapter 2. Members

We will use the [planning-app-analyst.xlsx](#) workbook from the [downloaded package](#) to configure the application. Open the workbook and [change connections](#) to your database first.

### Units

The application uses USD as the base currency (id 0). We need to change it to EURO.

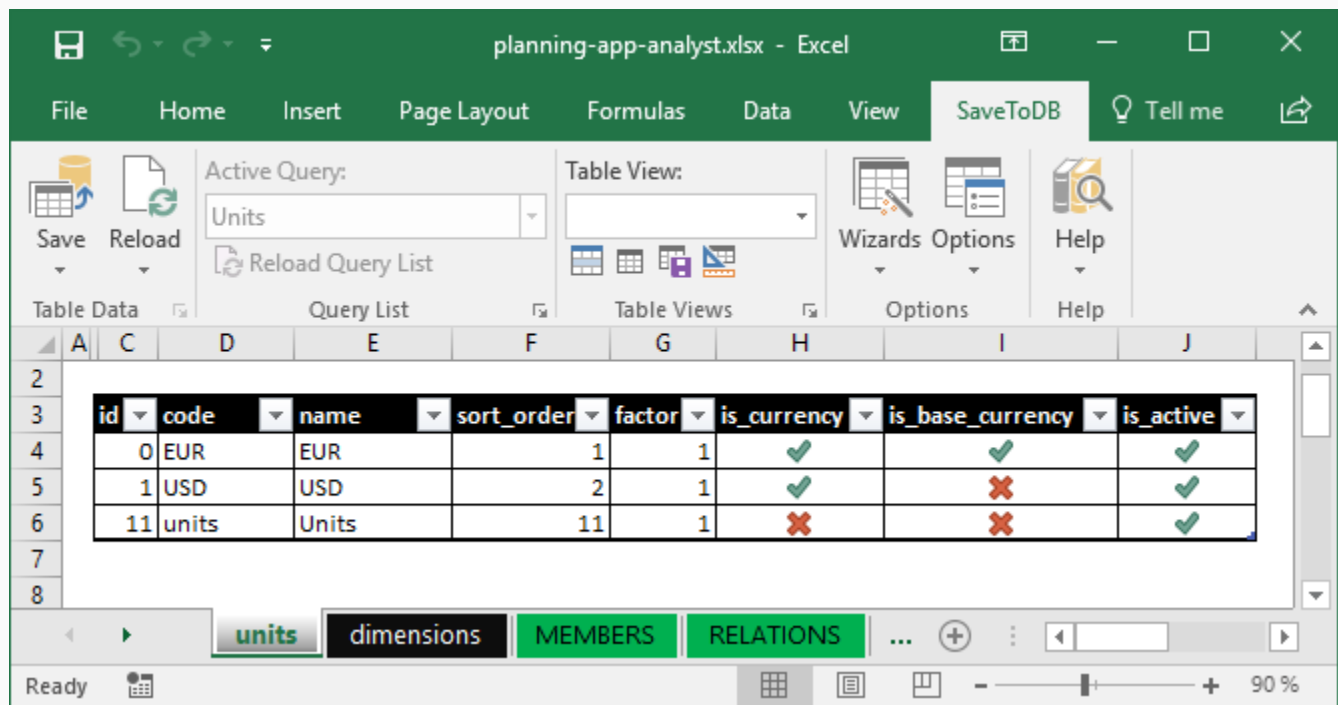
id	code	name	sort_order	factor	is_currency	is_base_currency	is_active
0	USD	USD	1	1	✓	✗	✓
1	EUR	EUR	2	1	✓	✓	✓
11	units	Units	11	1	✗	✗	✓

Right-click on EUR and execute “Set as the base currency.”

id	code	name	sort_order	factor	is_currency	is_base_currency	is_active
0	USD	USD	1	1	✓	✗	✓
1	EUR	EUR	2	1	✓	✓	✓
11	units	Units	11	1	✗	✗	✓



Now we have EURO as the base currency:



The screenshot shows the Excel interface with the 'SaveToDB' tab selected. The active query is 'Units'. The table below shows the data for the 'Units' query.

id	code	name	sort_order	factor	is_currency	is_base_currency	is_active
0	EUR	EUR		1	1	✓	✓
1	USD	USD		2	1	✗	✓
11	units	Units		11	1	✗	✓

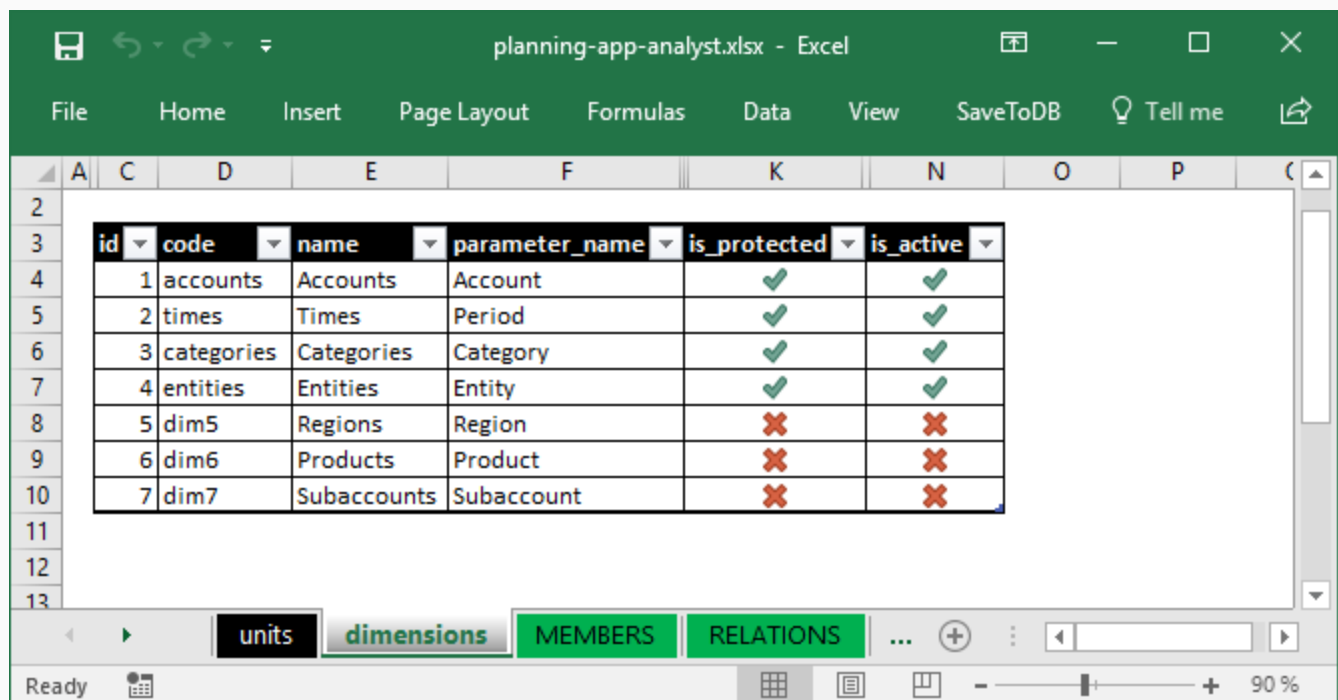
You may add other currencies and units later. Set `is_currency = 1` for currencies.

Set `is_base_currency = 1` for base currencies (the first currency in pairs like EUR/USD).

Do not forget to click the **Save** button on the **SaveToDB** tab to save changes.

## Dimensions

The application has the following default dimension configuration:



The screenshot shows the Excel interface with the 'SaveToDB' tab selected. The active query is 'Dimensions'. The table below shows the data for the 'Dimensions' query.

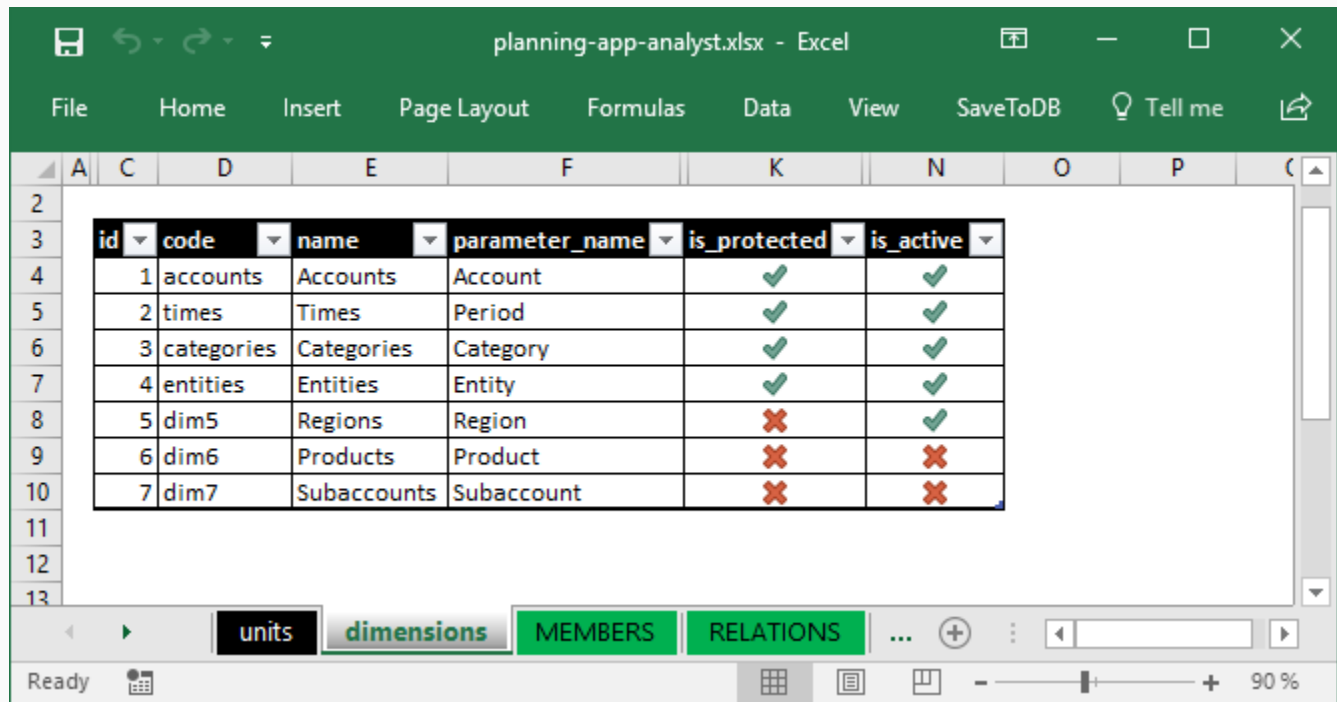
id	code	name	parameter_name	is_protected	is_active
1	accounts	Accounts	Account	✓	✓
2	times	Times	Period	✓	✓
3	categories	Categories	Category	✓	✓
4	entities	Entities	Entity	✓	✓
5	dim5	Regions	Region	✗	✗
6	dim6	Products	Product	✗	✗
7	dim7	Subaccounts	Subaccount	✗	✗

The application has seven dimensions. The first four are mandatory. Dimension 5-7 are optional.

You may change codes, names, and other parameters. However, do not change id values.

Also, do not change the meaning of the Accounts, Times, Categories, and Entities dimensions.

In this step, let's activate the Regions dimension and click the **Save** button to save the changes.

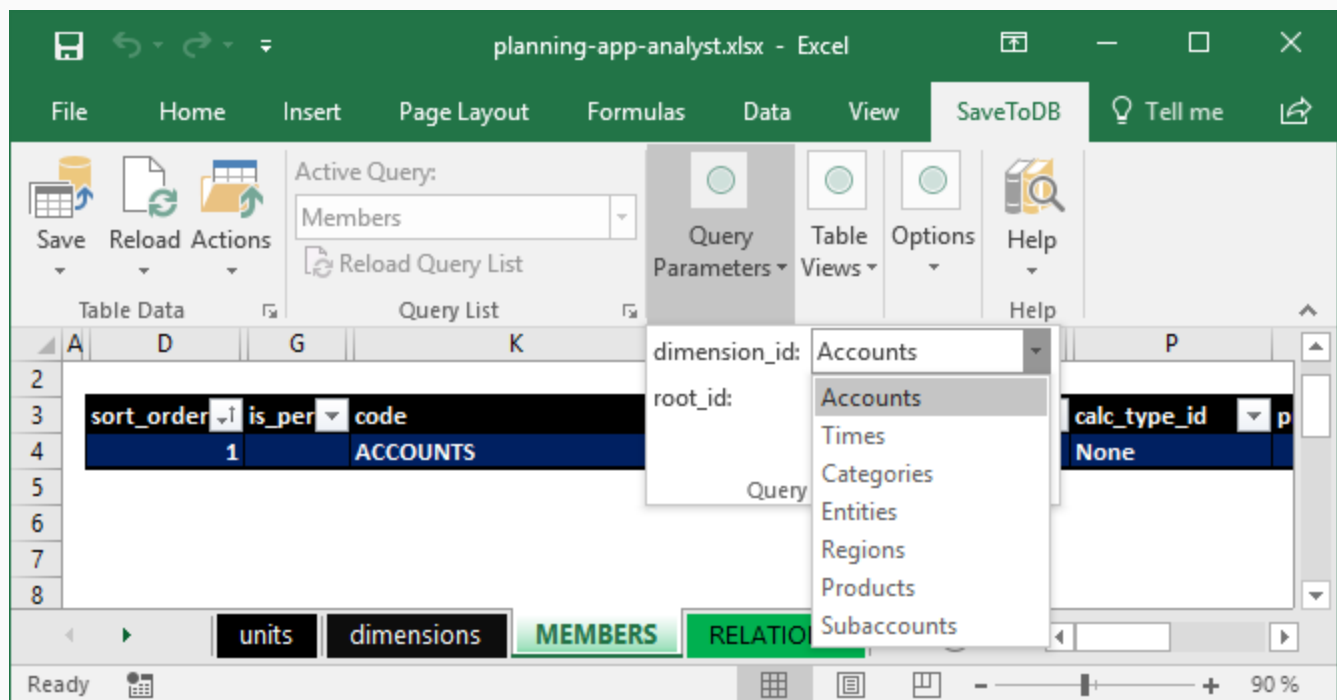


id	code	name	parameter_name	is_protected	is_active
1	accounts	Accounts	Account	✓	✓
2	times	Times	Period	✓	✓
3	categories	Categories	Category	✓	✓
4	entities	Entities	Entity	✓	✓
5	dim5	Regions	Region	✗	✓
6	dim6	Products	Product	✗	✗
7	dim7	Subaccounts	Subaccount	✗	✗

The table contains several parameters related to permissions. We will [configure](#) this later.

## Accounts

Let's open the **MEMBERS** worksheet and select the Accounts dimension:



sort_order	is_per	code
1		ACCOUNTS

The Accounts dimension contains a single member at the beginning.

Let's add the first members:

sort_order	code	name	tax_rate_id	unit_id	calc_type
1	ACCOUNTS	Accounts			None
2	CC	Cost Centers			None
3	CC1	Cost Center 1 (CC1)			None
4	CC11	CC11			None
5	CC111	CC111			None
6	CC112	CC112			None
7	CC113	CC113			None
8	CC12	CC12			None
9	CC121	CC121			None

First of all, we have added the root member for our reports, CC – Cost Centers.

Also, we have added items to the first cost center only (we will use these rows for other centers as templates).

We have filled the **sort\_order**, **code**, and **name** columns.

The code field values must be unique. The application uses sort\_order values to sort members.

Click the **Save** button to save new members.

In the next step, let's customize **row\_color**, **row\_bold**, and **row\_indent** values to get the desired format:

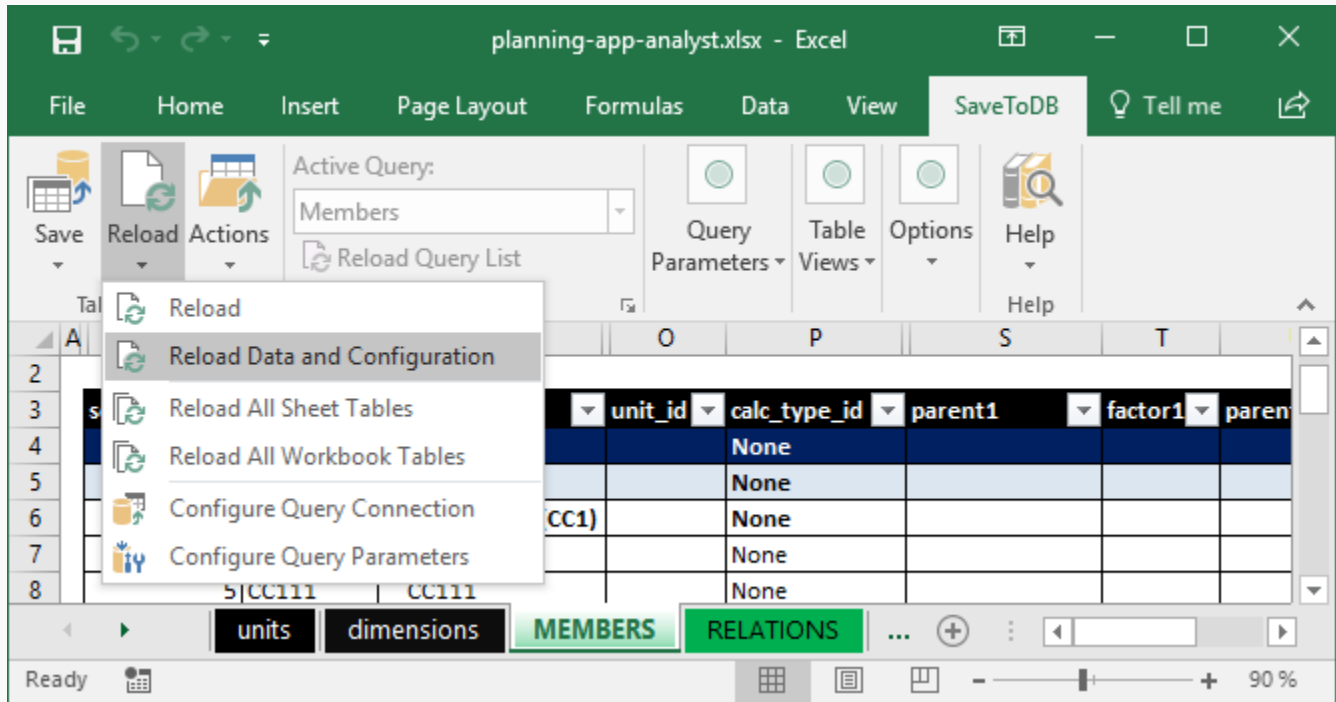
sort_order	row_color	row_bold	row_indent	code	name	unit_id	calc_type
1	11			ACCOUNTS	Accounts		None
2	12			CC	Cost Centers		None
3		1		CC1	Cost Center 1 (CC1)		None
4			2	CC11	CC11		None
5			4	CC111	CC111		None
6			4	CC112	CC112		None
7			4	CC113	CC113		None
8			2	CC12	CC12		None
9			4	CC121	CC121		None

The application outputs these values into hidden columns of forms and reports.

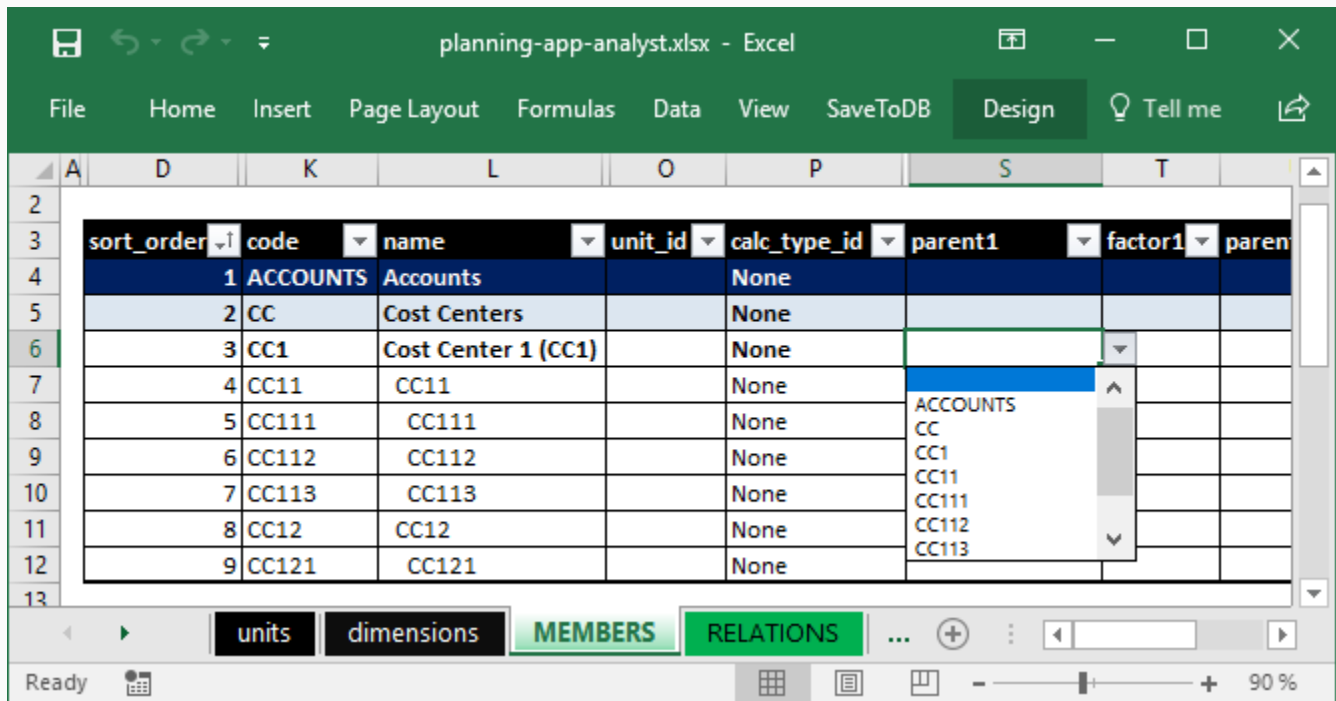
The forms use standard Excel conditional formatting to format rows, columns, and cells.

You may change the default conditional formatting using **Home, Conditional Formatting, Manage Rules...**

Before the next step we need to click **Reload Data and Configuration**:



This step is required to update validation lists with new members:



Use **Reload Data and Configuration** whenever you need to update lists or parameters.

Let's configure next properties: units, calculation types, parents, and factors:

sort_order	code	name	unit_id	calc_type_id	parent1	factor1	parent2
1	ACCOUNTS	Accounts		None			
2	CC	Cost Centers		None	ACCOUNTS	0	
3	CC1	Cost Center 1 (CC1)	EUR	Total	CC	0	
4	CC11	CC11	EUR	Total	CC1	1	
5	CC111	CC111	EUR	Total	CC11	1	
6	CC112	CC112	EUR	Total	CC11	1	
7	CC113	CC113	EUR	Total	CC11	1	
8	CC12	CC12	EUR	Total	CC1	1	
9	CC121	CC121	EUR	Total	CC12	1	

You may select values from drop-down lists or just copy and paste values.

In this example, we assign EURO and the Total calculation type to all members. You may use any value later.

Take a look at the parents and factors. These values are used to calculate aggregates and to create hierarchies.

The example has simple rules for members like CC11 or CC111.

We also need to set parents for upper-level members like CC with factor 0 to create hierarchies.

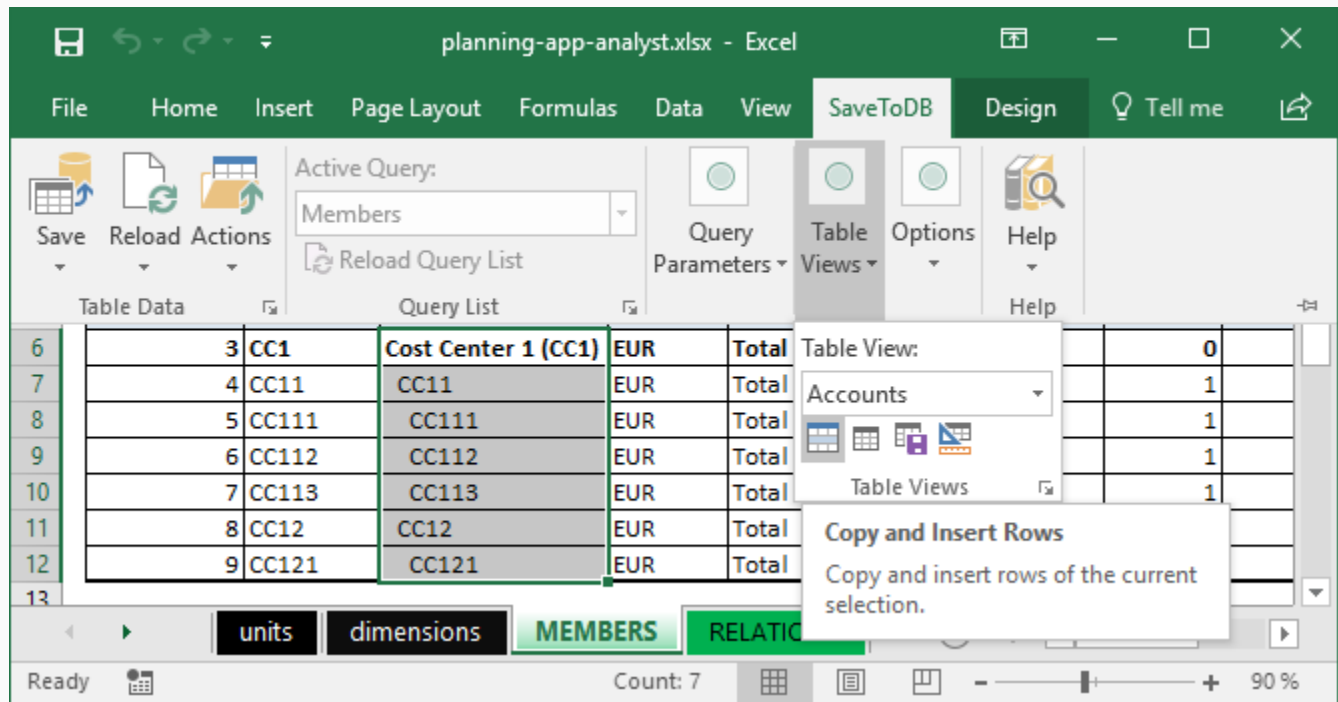
Click **Save** to save changes and then click **Actions, Update hierarchies** to update required rules for new members.

sort_order	code	name	unit_id	calc_type_id	parent1	factor1	parent2
3	CC1	Cost Center 1 (CC1)	EUR	Total	CC	0	
4	CC11	CC11	EUR	Total	CC1	1	
5	CC111	CC111	EUR	Total	CC11	1	

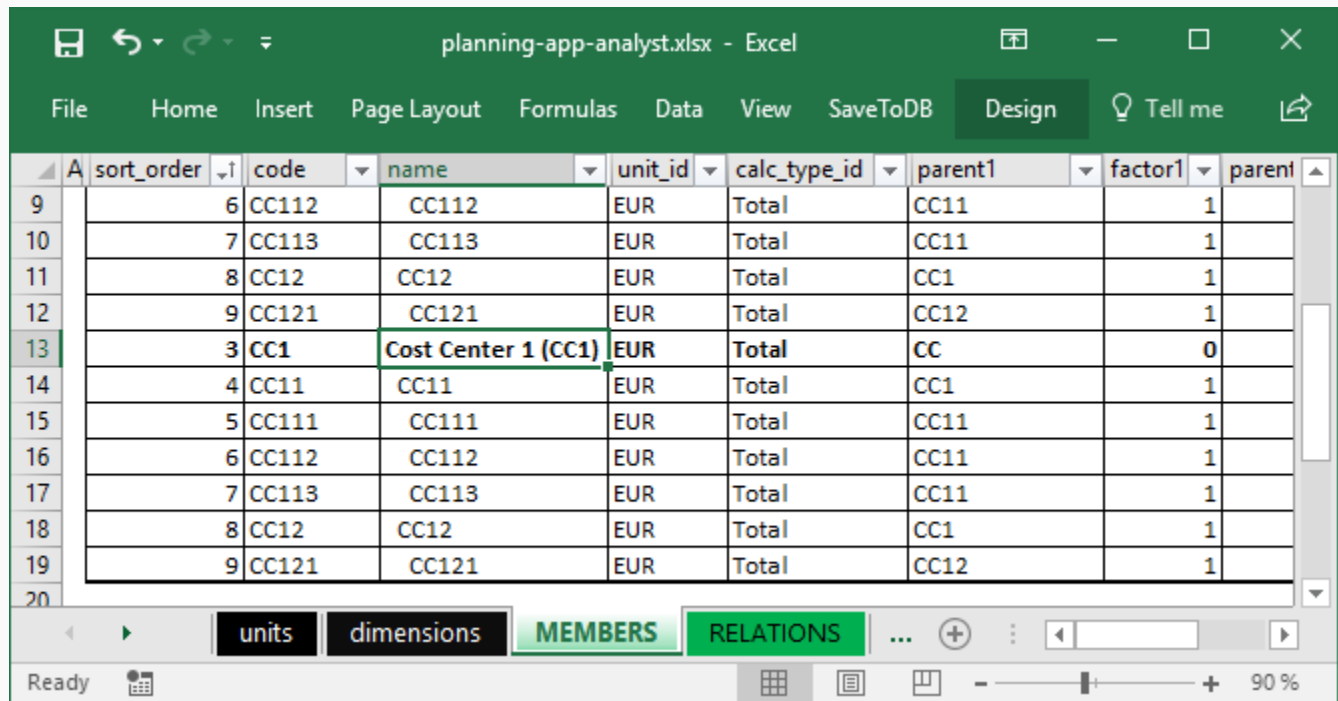
Do not forget to update hierarchies after changes in members. Try this first if you have any issue with members.

Now we will add members for cost centers 2 and 3 using a trick.

Select cells in rows to copy and click **Copy and Insert Rows**:



You will have a copy of selected rows:



Change the **sort\_order** values using the auto-fill feature and change CC1 to CC2 using search and replace:

	sort_order	code	name	unit_id	calc_type_id	parent1	factor1	parent
9	6	CC112	CC112	EUR	Total	CC11	1	
10	7	CC113	CC113	EUR	Total	CC11	1	
11	8	CC12	CC12	EUR	Total	CC1	1	
12	9	CC121	CC121	EUR	Total	CC12	1	
13	10	CC2	Cost Center 1 (CC2)	EUR	Total	CC	0	
14	11	CC21	CC21	EUR	Total	CC1	1	
15	12	CC211	CC211	EUR	Total	CC11	1	
16	13	CC212	CC212	EUR	Total	CC11	1	
17	14	CC213	CC213	EUR	Total	CC11	1	
18	15	CC22	CC22	EUR	Total	CC1	1	
19	16	CC221	CC221	EUR	Total	CC12	1	

You may use this trick with the SaveToDB add-in anywhere.

Click **Save** and then **Reload Data and Configuration** to update validation lists in the parent columns.

Now you may use search and replace in the parent column to change parents:

	sort_order	code	name	unit_id	calc_type_id	parent1	factor1	parent
9	6	CC112	CC112	EUR	Total	CC11	1	
10	7	CC113	CC113	EUR	Total	CC11	1	
11	8	CC12	CC12	EUR	Total	CC1	1	
12	9	CC121	CC121	EUR	Total	CC12	1	
13	10	CC2	Cost Center 2 (CC2)	EUR	Total	CC	0	
14	11	CC21	CC21	EUR	Total	CC2	1	
15	12	CC211	CC211	EUR	Total	CC21	1	
16	13	CC212	CC212	EUR	Total	CC21	1	
17	14	CC213	CC213	EUR	Total	CC21	1	
18	15	CC22	CC22	EUR	Total	CC2	1	
19	16	CC221	CC221	EUR	Total	CC22	1	

Click **Save** and then click **Actions, Update hierarchies**.

Now you may repeat the steps to the cost center 3. Just copy rows of the cost center 2, change sort orders and replace CC2 to CC3.

The final configuration of account members looks like this:

sort_order	row_color	row_bold	row_indent	code	name	unit_id	calc_type_id	parent1	factor1
1	11			ACCOUNTS	Accounts		None		
2	12			CC	Cost Centers		None	ACCOUNTS	0
3		1		CC1	Cost Center 1 (CC1)	EUR	Total	CC	0
4			2	CC11	CC11	EUR	Total	CC1	1
5			4	CC111	CC111	EUR	Total	CC11	1
6			4	CC112	CC112	EUR	Total	CC11	1
7			4	CC113	CC113	EUR	Total	CC11	1
8			2	CC12	CC12	EUR	Total	CC1	1
9			4	CC121	CC121	EUR	Total	CC12	1
10		1		CC2	Cost Center 2 (CC2)	EUR	Total	CC	0
11			2	CC21	CC21	EUR	Total	CC2	1
12			4	CC211	CC211	EUR	Total	CC21	1
13			4	CC212	CC212	EUR	Total	CC21	1
14			4	CC213	CC213	EUR	Total	CC21	1
15			2	CC22	CC22	EUR	Total	CC2	1
16			4	CC221	CC221	EUR	Total	CC22	1
17		1		CC3	Cost Center 3 (CC3)	EUR	Total	CC	0
18			2	CC31	CC31	EUR	Total	CC3	1
19			4	CC311	CC311	EUR	Total	CC31	1
20			4	CC312	CC312	EUR	Total	CC31	1
21			4	CC313	CC313	EUR	Total	CC31	1
22			2	CC32	CC32	EUR	Total	CC3	1
23			4	CC321	CC321	EUR	Total	CC32	1

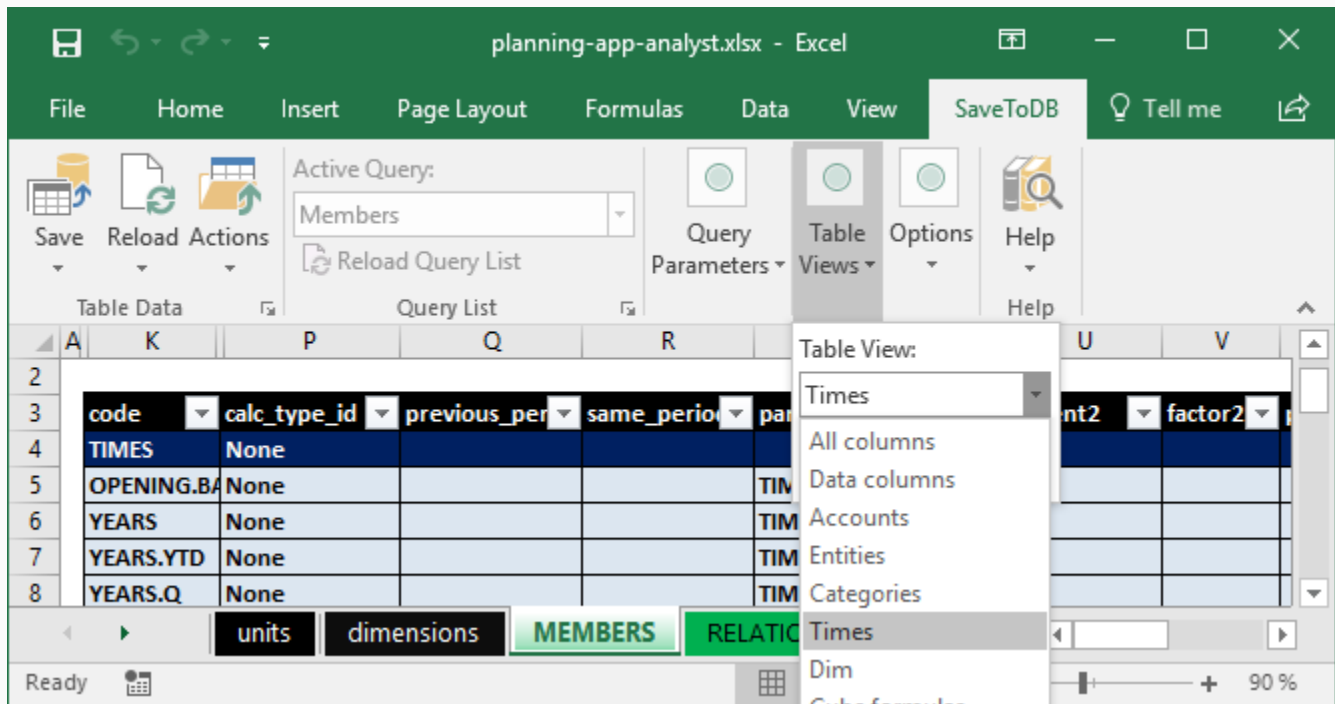
## Times

Let's open members of the Times dimension. Select Times in the **dimension\_id** parameter at the ribbon:

The screenshot shows the Microsoft Excel interface with the 'SaveToDB' ribbon active. The 'Active Query' is 'Members'. A dropdown menu for 'dimension\_id' is open, showing a list of dimensions: Accounts, Times, Categories, Entities, Regions, Products, and Subaccounts. The 'Times' option is selected. The background shows a table with columns 'code', 'calc\_type\_id', 'previous\_per', and 'same'. The 'MEMBERS' tab is selected in the bottom navigation bar.



Also, apply the Times view using the Table Views drop-down menu:



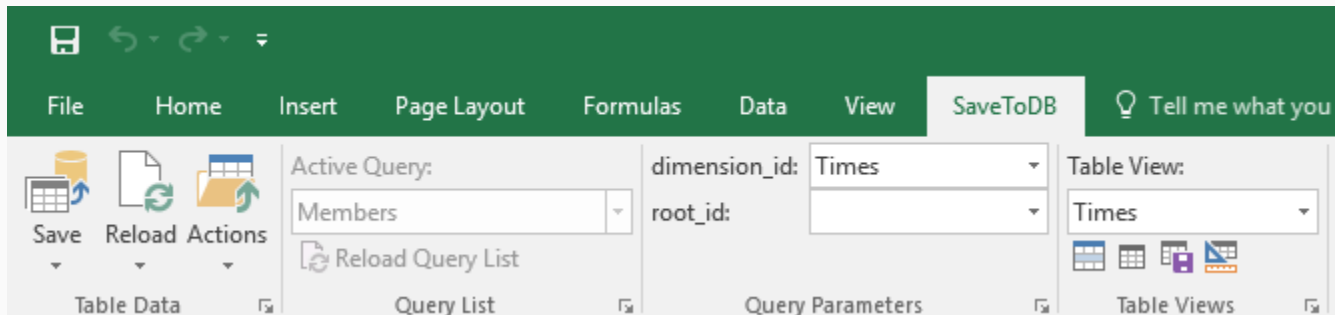
This view shows columns related to the Times dimension and hides others.

You may use this feature on other worksheets also.

Moreover, you may create your personal views. Just click the **Save Table View** button in the **Table Views** group.

The screenshots above have a compacted ribbon.

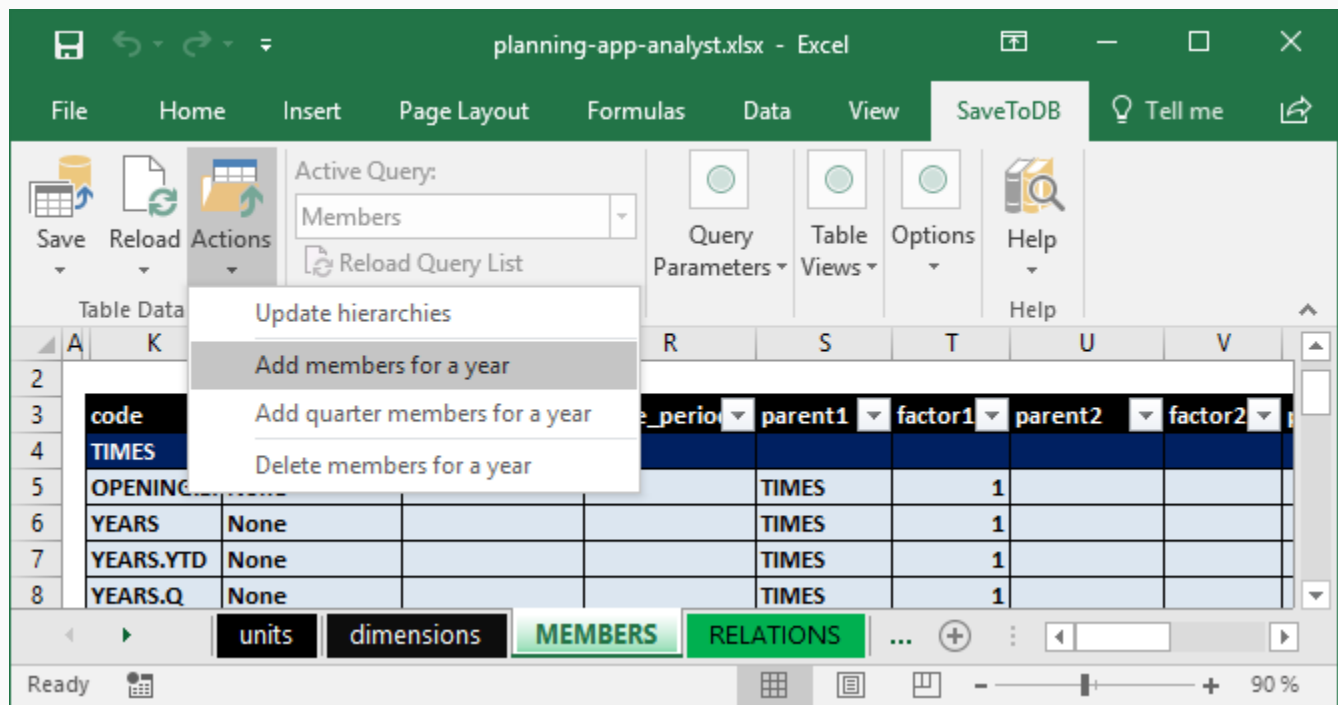
Usually, you will see a complete ribbon like this:



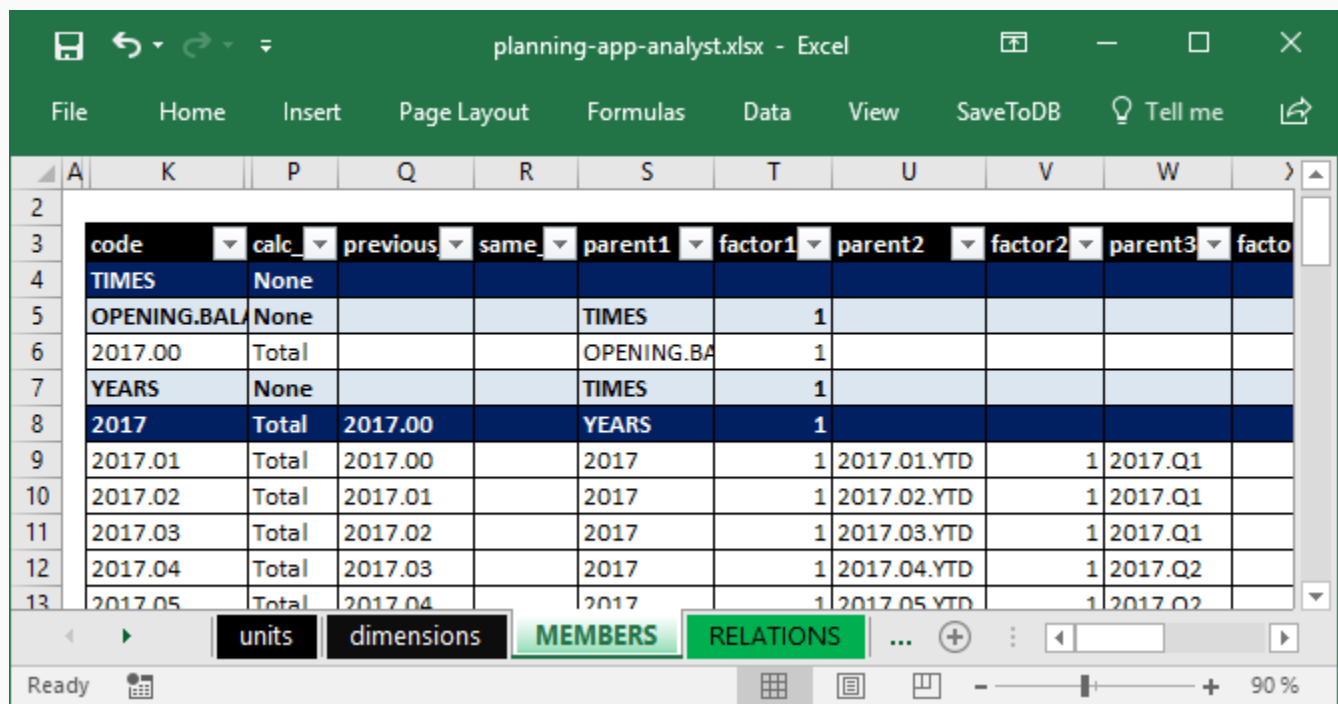
Initially, the Times dimension contains only root members:

- TIMES
- OPENING.BALANCE
- YEARS
- YEARS.YTD
- YEARS.Q

Click **Actions, Add members for a year** to add time members on monthly bases:



The application adds the complete set of configured members to get data for months, YTD, quarters, and years.



Use members like 2017.00 to input opening balance values in your applications. We do not use it in this example.

## Categories

Select the Categories in the dimension\_id parameter at the ribbon and apply the Categories table view.

You will see the preconfigured members of the Categories dimension:

sort_order	code	name	calc_type_id	previous	same_peri
1	CATEGORIES	CATEGORIES	None		
2	CATEGORIES.ALL	All Categories	None		
3	SAME_PERIOD	Same Period	Same Period		
4	BUDGET	Budget	Total		
5	FORECAST	Forecast	Total		
7	ACTUAL	Actual	Total		Same Period
8	BUDGET-SAME_PERIOD	Budget vs Same Period	Difference	Same Period	Budget
9	BUDGET/SAME_PERIOD,%	Budget vs Same Period,%	Percent	Same Period	Budget
10	FORECAST-BUDGET	Budget vs Forecast	Difference	Budget	Forecast
11	FORECAST/BUDGET,%	Budget vs Forecast, %	Percent	Budget	Forecast
12	FORECAST-SAME_PERIOD	Budget vs Forecast	Difference	Same Period	Forecast
13	FORECAST/SAME_PERIOD,%	Budget vs Forecast, %	Percent	Same Period	Forecast
14	ACTUAL-SAME_PERIOD	Actual vs Same Period	Difference	Same Period	Actual
15	ACTUAL-SAME_PERIOD,%	Actual vs Same Period,%	Percent	Same Period	Actual
16	ACTUAL-BUDGET	Budget vs Actual	Difference	Budget	Actual
17	ACTUAL-BUDGET,%	Budget vs Actual, %	Percent	Budget	Actual
18	ACTUAL-FORECAST	Actual vs Forecast	Difference	Forecast	Actual
19	ACTUAL-FORECAST,%	Actual vs Forecast,%	Percent	Forecast	Actual

The dimension contains three basic members that contain data: **Budget**, **Forecast**, and **Actual**.

The **Same Period** is a special member that shows actual data for the same period of the previous year.

Also, the dimension contains several calculated members that show differences in comparison reports.

You may add and configure additional members in your applications (for example, for different versions).

In this example, we will use the predefined members.

## Entities

Select the Entities in the dimension\_id parameter at the ribbon and apply the Entities table view.

You will see the preconfigured members of the Entities dimension:

sort_order	row_color	code	name	calc_type_id	parent1	factor1
1	11	ENTITIES	ENTITIES	None		
2		ENTITIES.ALL	All Entities	None	ENTITIES	1
3	11	TOTAL	Company Total	Total	ENTITIES.ALL	1

Let's add the required members:

sort_order	row_color	code	name	calc_type_id	parent1	factor1
1	11	ENTITIES	ENTITIES	None		
2		ENTITIES.ALL	All Entities	None	ENTITIES	1
3	11	TOTAL	Company Total	Total	ENTITIES.ALL	1
4		KAM.DE	KAM (Germany)	Total	TOTAL	1
5		KAM.AT	KAM (Austria)	Total	TOTAL	1
6		KAM.CH	KAM (Switzerland)	Total	TOTAL	1
7		KAM.FR	KAM (France)	Total	TOTAL	1

We have used the existing member **TOTAL** as a parent instead of creating a new member All KAMs.

We may create it later if we need a separate aggregate for it.

Do not forget to update hierarchies after saving changes.

## Regions

Select the Regions in the dimension\_id parameter at the ribbon. You will see default members:

sort_order	code	name	calc_type_id	parent1	factor1
1	DIM5	DIM5	None		
2	DIM5.ALL	DIM5.ALL	None	DIM5	1
3	DIM5.NONE		Total	DIM5.ALL	1

Let's add the desired members:

sort_order	code	name	calc_type_id	parent1	factor1
1	DIM5	DIM5	None		
2	DIM5.ALL	DIM5.ALL	None	DIM5	1
3	DIM5.NONE		Total	DIM5.ALL	1
4	COUNTRIES	All Countries	Total	DIM5	1
5	COUNTRY.DE	Germany	Total	COUNTRIES	1
6	COUNTRY.AT	Austria	Total	COUNTRIES	1
7	COUNTRY.CH	Switzerland	Total	COUNTRIES	1
8	COUNTRY.FR	France	Total	COUNTRIES	1

In this case, we have created a separate member, **All Countries**, to get totals.

Do not forget to save the changes and to update hierarchies.

## Hierarchies

You may check generated application rules in the **hierarchies** worksheet.

Use parameters to change dimensions. Here are the Entities dimension members:

table_name	member	parent	calc_type	factor	level
parents	All Entities	ENTITIES	None	1	
parents	Company Total	All Entities	None	1	
parents	KAM (Germany)	Company Total	Total	1	
parents	KAM (Austria)	Company Total	Total	1	
parents	KAM (Switzerland)	Company Total	Total	1	
parents	KAM (France)	Company Total	Total	1	
hierarchies	ENTITIES	ENTITIES			0
hierarchies	All Entities	ENTITIES			1
hierarchies	All Entities	All Entities			0
hierarchies	Company Total	ENTITIES			2
hierarchies	Company Total	All Entities			1
hierarchies	Company Total	Company Total			0
hierarchies	KAM (Germany)	ENTITIES			3
hierarchies	KAM (Germany)	All Entities			2
hierarchies	KAM (Germany)	Company Total			1
hierarchies	KAM (Germany)	KAM (Germany)			0
hierarchies	KAM (Austria)	ENTITIES			3
hierarchies	KAM (Austria)	All Entities			2
hierarchies	KAM (Austria)	Company Total			1
hierarchies	KAM (Austria)	KAM (Austria)			0
hierarchies	KAM (Switzerland)	ENTITIES			3
hierarchies	KAM (Switzerland)	All Entities			2
hierarchies	KAM (Switzerland)	Company Total			1
hierarchies	KAM (Switzerland)	KAM (Switzerland)			0
hierarchies	KAM (France)	ENTITIES			3
hierarchies	KAM (France)	All Entities			2
hierarchies	KAM (France)	Company Total			1
hierarchies	KAM (France)	KAM (France)			0
factors	KAM (Germany)	Company Total	Total	1	
factors	KAM (Germany)	KAM (Germany)	Total	1	
factors	KAM (Austria)	Company Total	Total	1	
factors	KAM (Austria)	KAM (Austria)	Total	1	
factors	KAM (Switzerland)	Company Total	Total	1	
factors	KAM (Switzerland)	KAM (Switzerland)	Total	1	
factors	KAM (France)	Company Total	Total	1	
factors	KAM (France)	KAM (France)	Total	1	

You may see source parents from the **MEMBERS** worksheet, created hierarchies, and calculation factors.

The application uses these data to filter members by parents and hierarchy levels and to calculate aggregates.

You may update rules using the **Update hierarchies** item in the **Actions** menu.

## Relations

Configuring member relations solves two tasks:

1. It prevents incorrect input.
2. It helps to show meaningful rows in forms with multiple dimensions in rows.

To edit member relations open the **RELATIONS** worksheet.

For example, we may configure the following relations between cost centers and key account managers:

code	name	KAM.DE	KAM.AT	KAM.CH	KAM.FR
<b>CC1</b>	<b>Cost Center 1 (CC1)</b>				
CC11	CC11				
CC111	CC111	✓	✓	✓	✓
CC112	CC112	✓	✓	✓	✓
CC113	CC113	✓	✓	✓	✓
CC12	CC12				
CC121	CC121	✓	✓	✓	✓
<b>CC2</b>	<b>Cost Center 2 (CC2)</b>				
CC21	CC21				
CC211	CC211	✓	✓		
CC212	CC212	✓	✓		
CC213	CC213	✓	✓		
CC22	CC22				
CC221	CC221	✓	✓		
<b>CC3</b>	<b>Cost Center 3 (CC3)</b>				
CC31	CC31				
CC311	CC311			✓	✓
CC312	CC312			✓	✓
CC313	CC313			✓	✓
CC32	CC32				
CC321	CC321			✓	✓

In this example, all managers may input data for the cost center 1, only managers from Germany and Austria may input data for the cost center 2, and only managers from Switzerland and France may work with the cost center 3.

Note that these are not permissions. These are business rules.

That means that the company does not have cost center 2 operations in Switzerland and France, and cost center 3 operations in Germany and Austria.

To set the relation, type 1. To clear the relation, delete the value. You may copy and paste values.

The application changes the configuration immediately after the change. You do not need to click the **Save** button.

Below is the same table for relations between accounts and regions:

code	name	COUNTRY.DE	COUNTRY.AT	COUNTRY.CH	COUNTRY.FR
<b>CC1</b>	<b>Cost Center 1 (CC1)</b>				
CC11	CC11				
CC111	CC111	✓	✓	✓	✓
CC112	CC112	✓	✓	✓	✓
CC113	CC113	✓	✓	✓	✓
CC12	CC12				
CC121	CC121	✓	✓	✓	✓
<b>CC2</b>	<b>Cost Center 2 (CC2)</b>				
CC21	CC21				
CC211	CC211	✓	✓		
CC212	CC212	✓	✓		
CC213	CC213	✓	✓		
CC22	CC22				
CC221	CC221	✓	✓		
<b>CC3</b>	<b>Cost Center 3 (CC3)</b>				
CC31	CC31				
CC311	CC311			✓	✓
CC312	CC312			✓	✓
CC313	CC313			✓	✓
CC32	CC32				
CC321	CC321			✓	✓

In this example, the tables are similar.

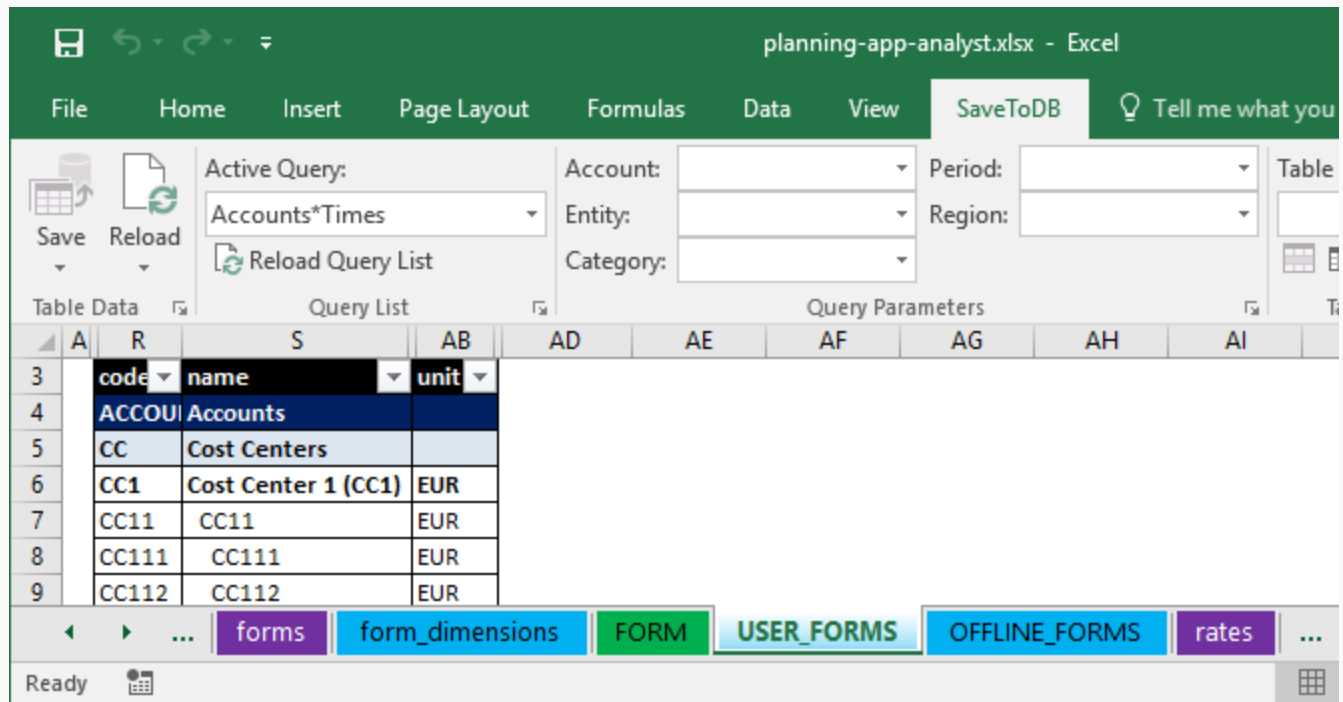
# Chapter 3. Forms

In this chapter, we will discuss configuring forms for business users.

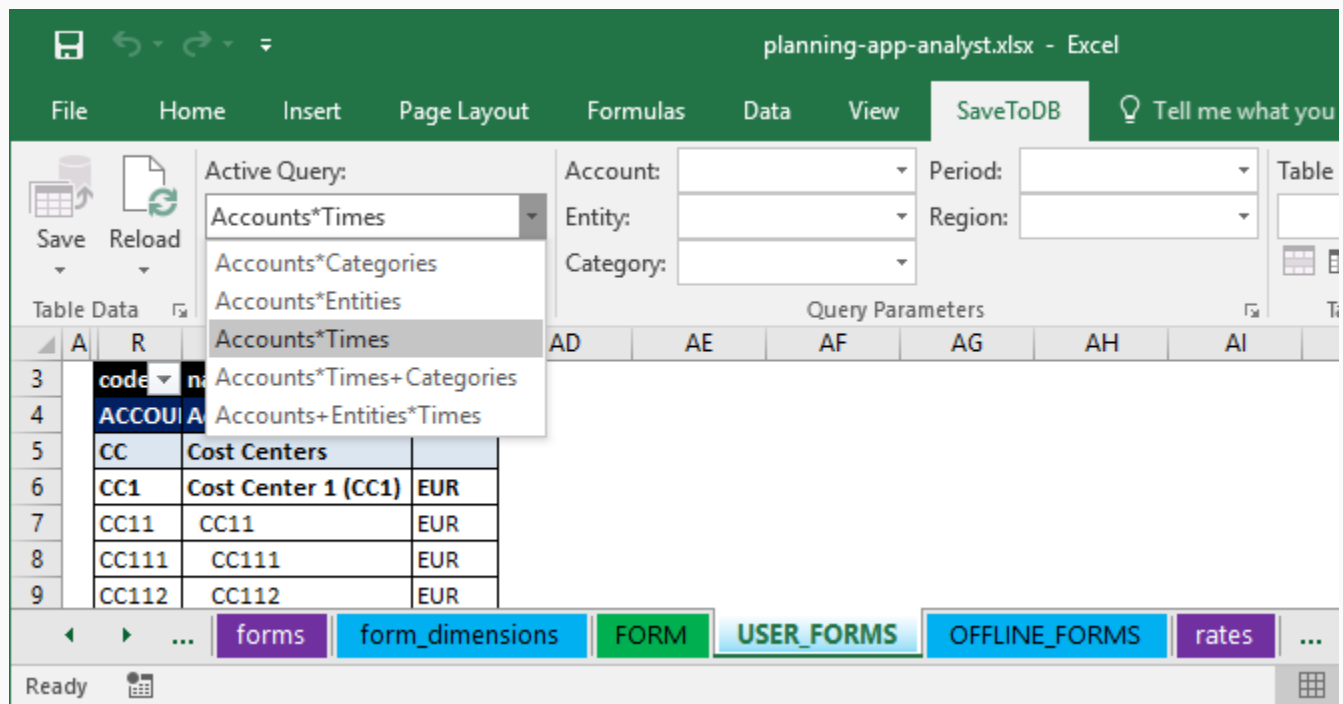
## User Forms

Business users have personal Excel workbooks to work with forms. We will open an example later.

Business analysts may test forms on the **USER\_FORMS** and **OFFLINE\_FORMS** worksheets.



First of all, click the **Reload Query List** button and select a report from the drop-down **Active Query** list:

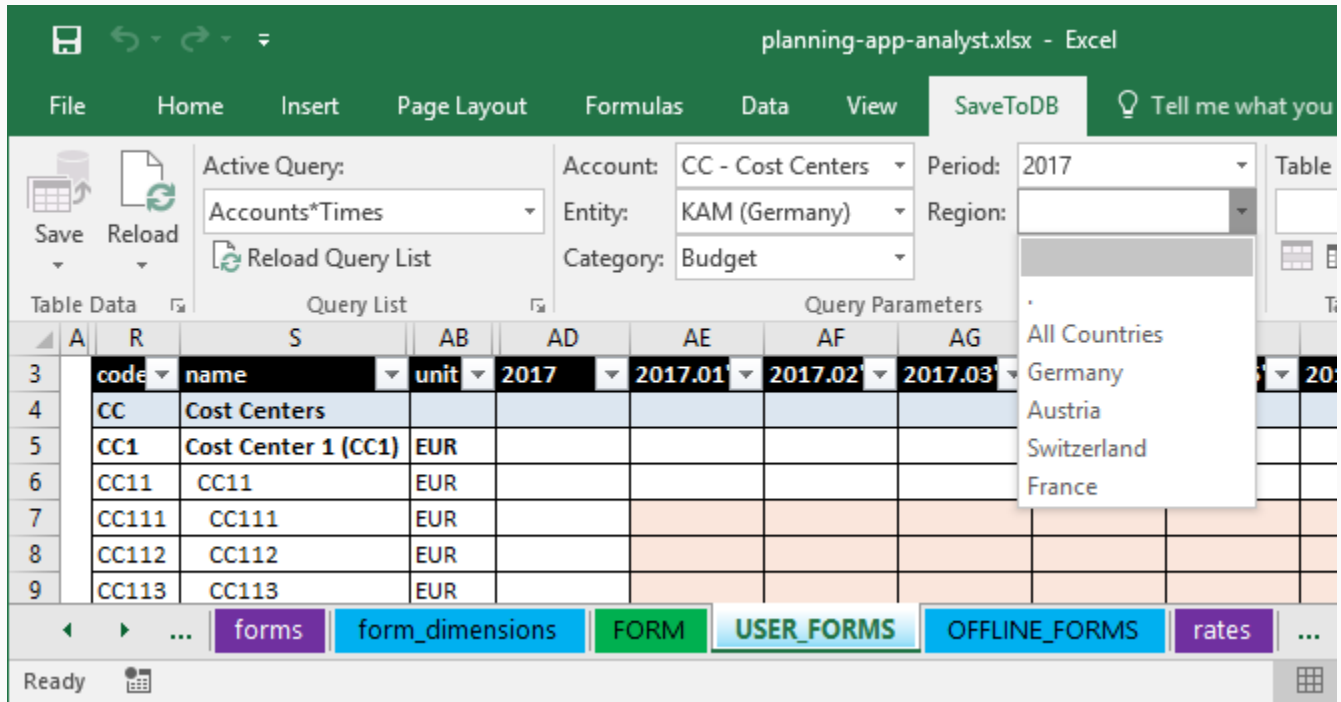




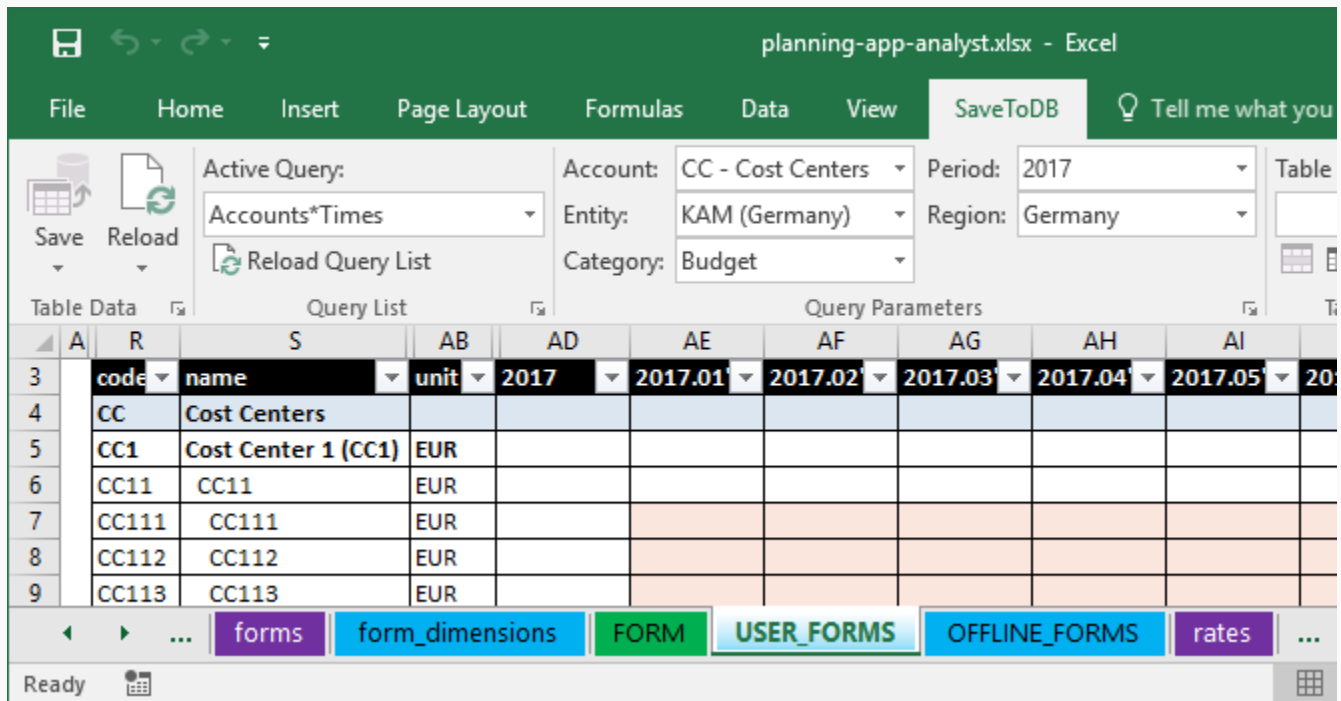
The **Active Query** list is dynamic and contains preconfigured active forms with the actual select permission.

So, different business users may have different lists.

Let's select parameter values using drop-down lists at the ribbon:



Now, we see an empty form:

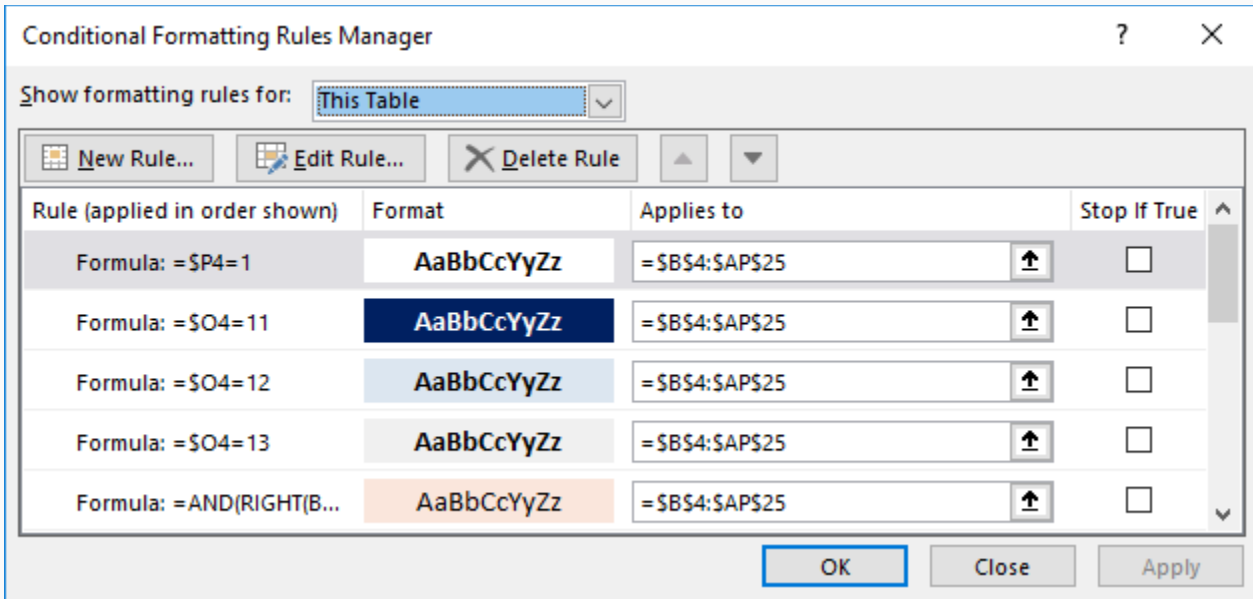


We may see accounts in form rows and time periods in columns. This is a preconfigured layout for the Accounts\*Times form. You may configure layouts with the desired row and column dimensions.

## Conditional Formatting

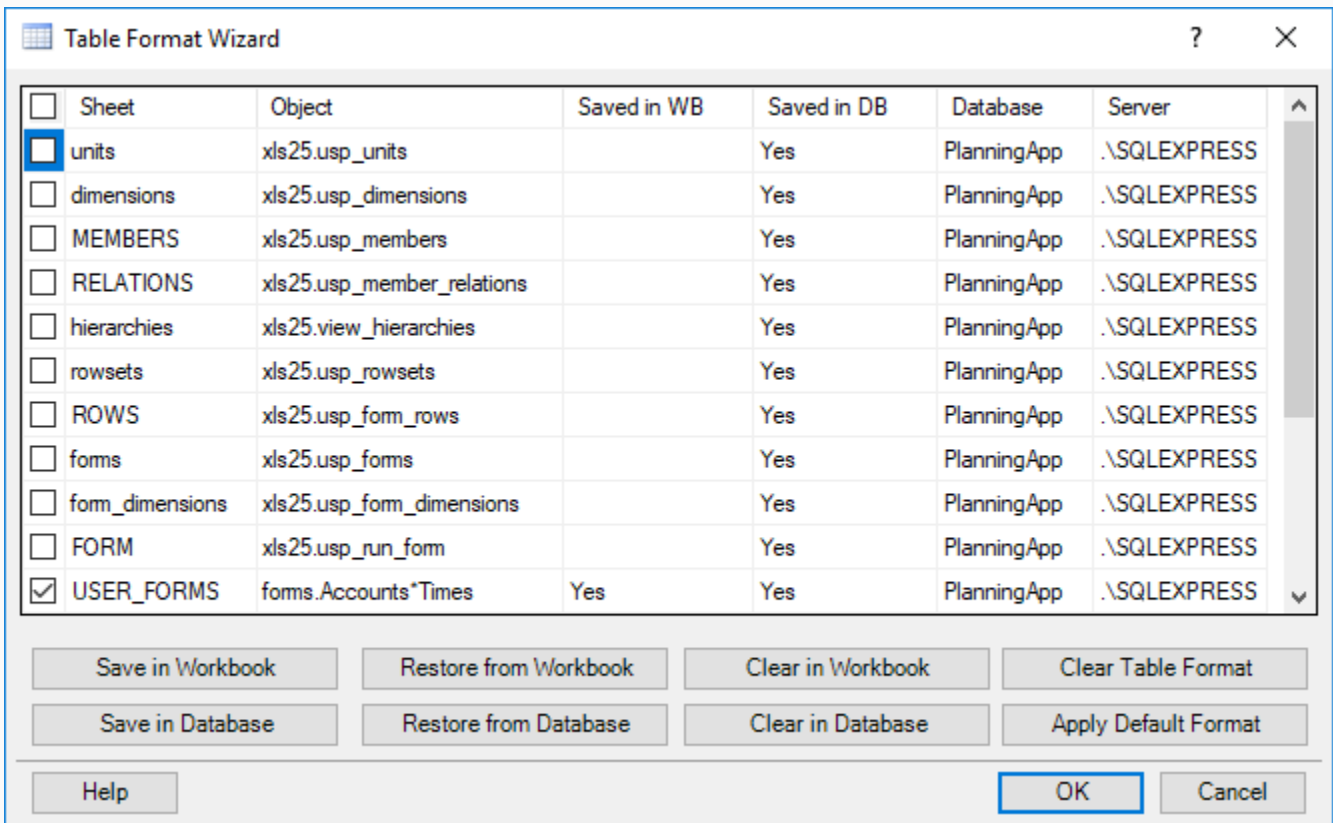
All application forms use standard Excel formatting features.

For example, the current form has the following formatting rules:



You may change rules including fonts, colors, sizes, borders, and number formats.

If you want to set new formatting as default for other users, click **SaveToDB, Wizards, Table Format Wizard**:

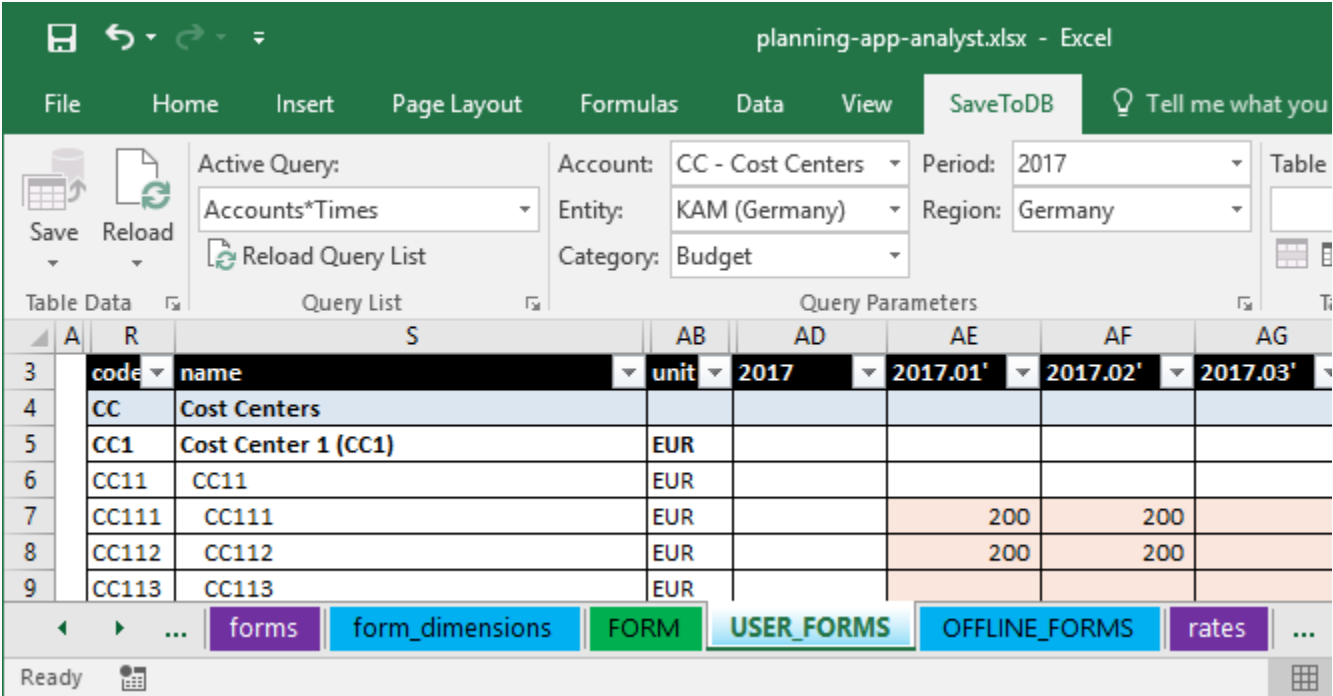


Check the forms you want to save and click **Save in Database**.

Note that you and other users may also restore formats from a database using the **Restore from Database** button.

## Editing Data

Let's edit several highlighted (editable) cells:



The screenshot shows an Excel spreadsheet titled "planning-app-analyst.xlsx". The interface includes a ribbon with tabs: File, Home, Insert, Page Layout, Formulas, Data, View, SaveToDB, and Tell me what you... The "Active Query" is "Accounts\*Times". The "Account" is "CC - Cost Centers", "Entity" is "KAM (Germany)", and "Category" is "Budget". The "Period" is "2017" and "Region" is "Germany". The table data is as follows:

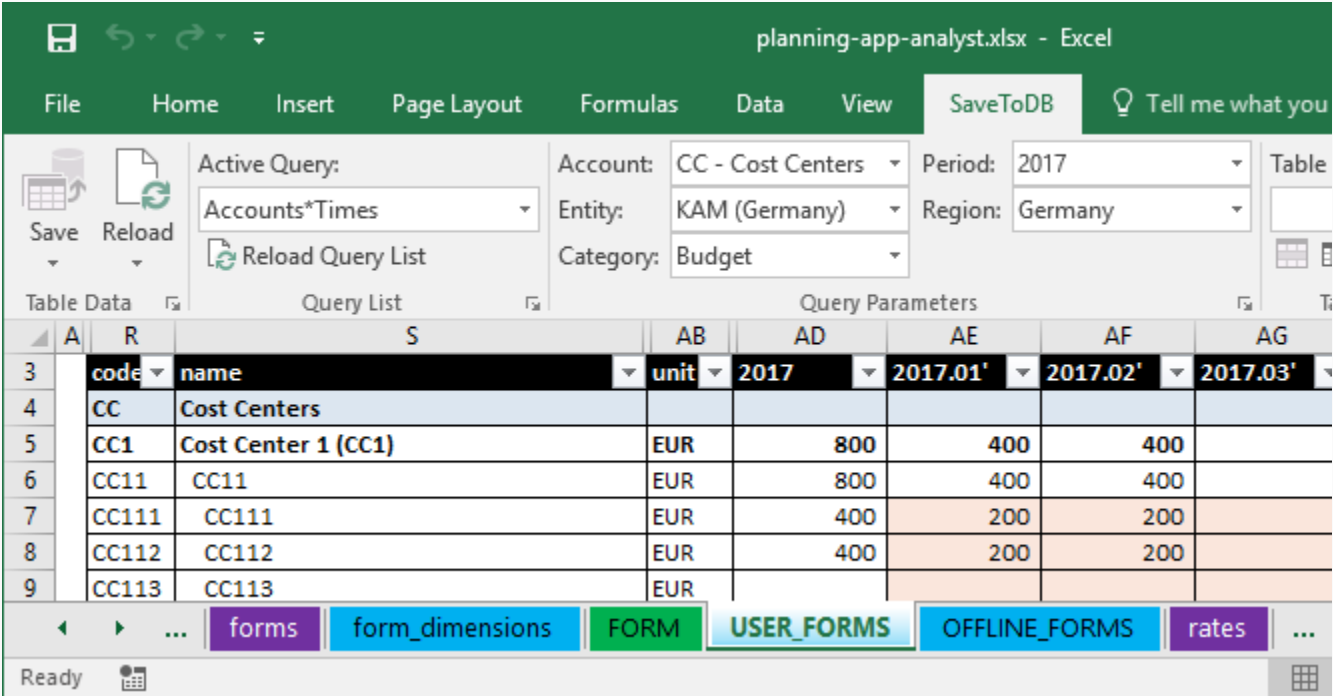
code	name	unit	2017	2017.01'	2017.02'	2017.03'
CC	Cost Centers					
CC1	Cost Center 1 (CC1)	EUR				
CC11	CC11	EUR				
CC111	CC111	EUR		200	200	
CC112	CC112	EUR		200	200	
CC113	CC113	EUR				

The application updates the source cube data immediately after changes, however, it leaves the form as is.

You may paste values using formulas. However, the application replaces formulas with values later.

Note that you may use **Undo** (Ctrl-Z) to revert changes.

Click the **Reload** button to reload data:

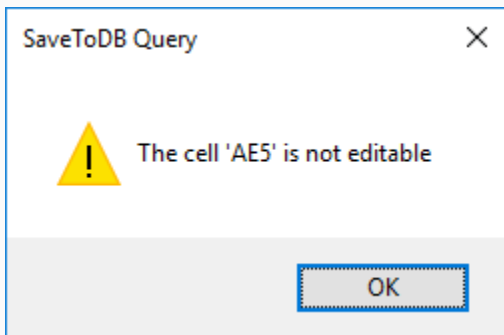


The screenshot shows the same Excel spreadsheet as above, but with updated data. The "Reload" button in the "Active Query" section is highlighted. The table data is now:

code	name	unit	2017	2017.01'	2017.02'	2017.03'
CC	Cost Centers					
CC1	Cost Center 1 (CC1)	EUR	800	400	400	
CC11	CC11	EUR	800	400	400	
CC111	CC111	EUR	400	200	200	
CC112	CC112	EUR	400	200	200	
CC113	CC113	EUR				

As you may see, the application updates the aggregates.

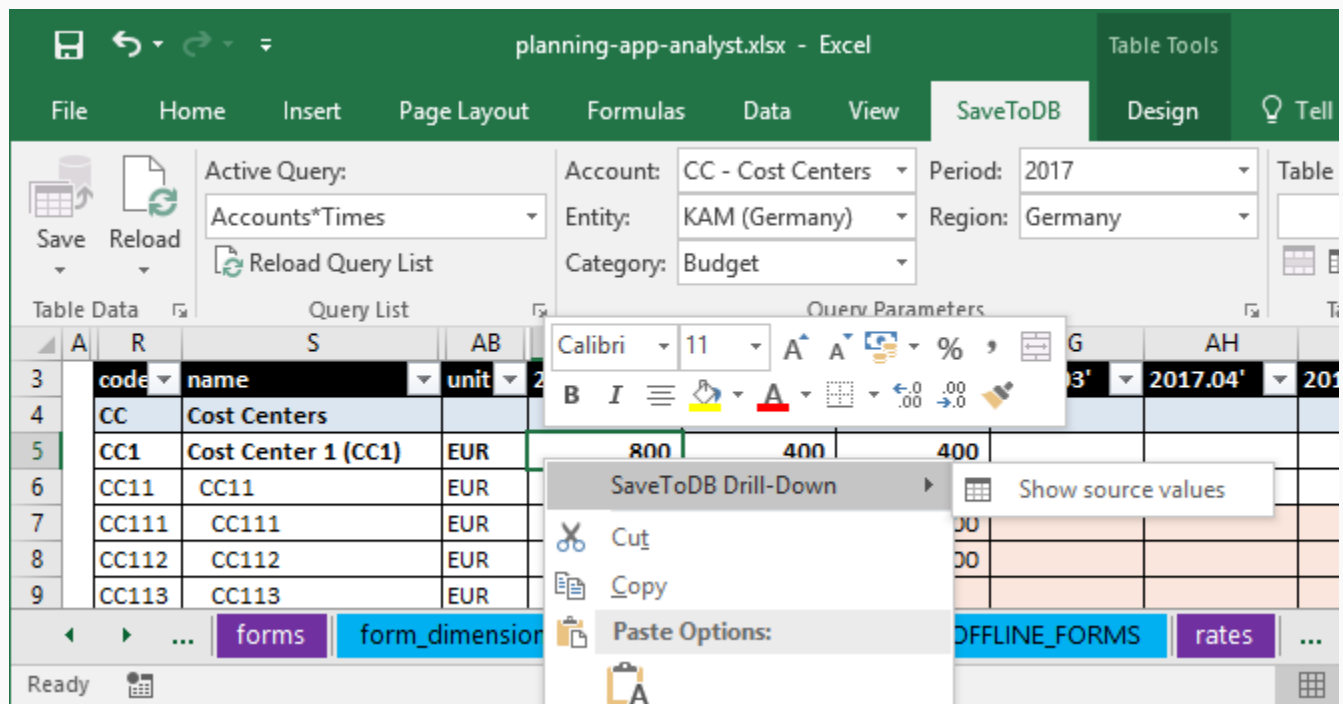
Try to edit any cell with the aggregated value. You will have an error message:



The application controls business logic rules and permissions and may reject user's changes.

## Drill-Down

You may double-click on any cell or right-click and open on **Show source values**:



The application shows the underlying values:

Show source values										
id	dim_1	dim_2	dim_3	dim_4	dim_5	unit	source_val	rate	factor	value
1	CC111 - CC111	2017.01	Budget	KAM (Germany)	Germany	EUR	200.00		1	200.00
5	CC111 - CC111	2017.02	Budget	KAM (Germany)	Germany	EUR	200.00		1	200.00
2	CC112 - CC112	2017.01	Budget	KAM (Germany)	Germany	EUR	200.00		1	200.00
6	CC112 - CC112	2017.02	Budget	KAM (Germany)	Germany	EUR	200.00		1	200.00

EXEC [xls25].[xl\_actions\_run\_form\_cell\_data] @column\_name = N'2017', @table\_name = N'[forms].[Accounts\*Times]', @m...

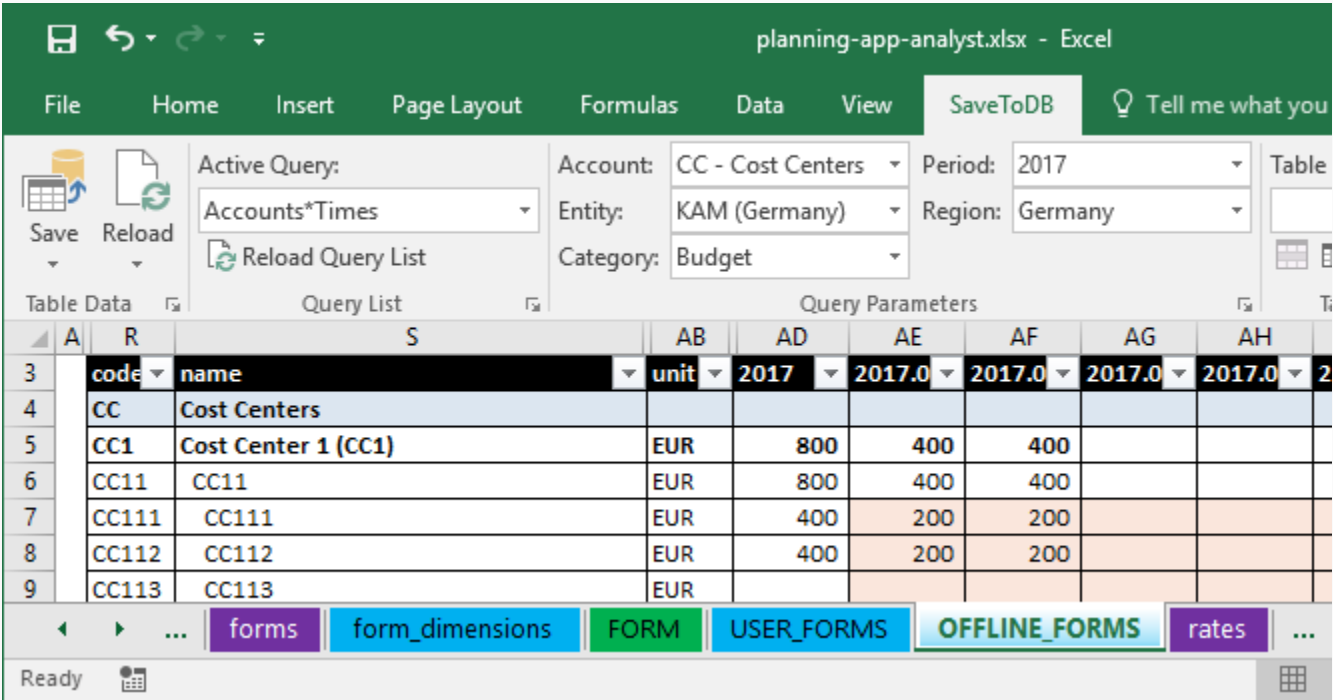
## Offline User Forms

In the previous topic, we have discussed forms that update the underlying database immediately after changes. Also, forms control user values using business rules and permissions.

However, sometimes we need the possibility to work outside of the corporate network.

In this case, we may use offline forms.

Just load the required data and edit it. Later, you may click the **Save** button to save the changes.



The screenshot shows an Excel spreadsheet with the following data:

code	name	unit	2017	2017.0	2017.0	2017.0	2017.0
CC	Cost Centers						
CC1	Cost Center 1 (CC1)	EUR	800	400	400		
CC11	CC11	EUR	800	400	400		
CC111	CC111	EUR	400	200	200		
CC112	CC112	EUR	400	200	200		
CC113	CC113	EUR					

This is a useful scenario. However, note that the form ignores the most of the business rules during editing.

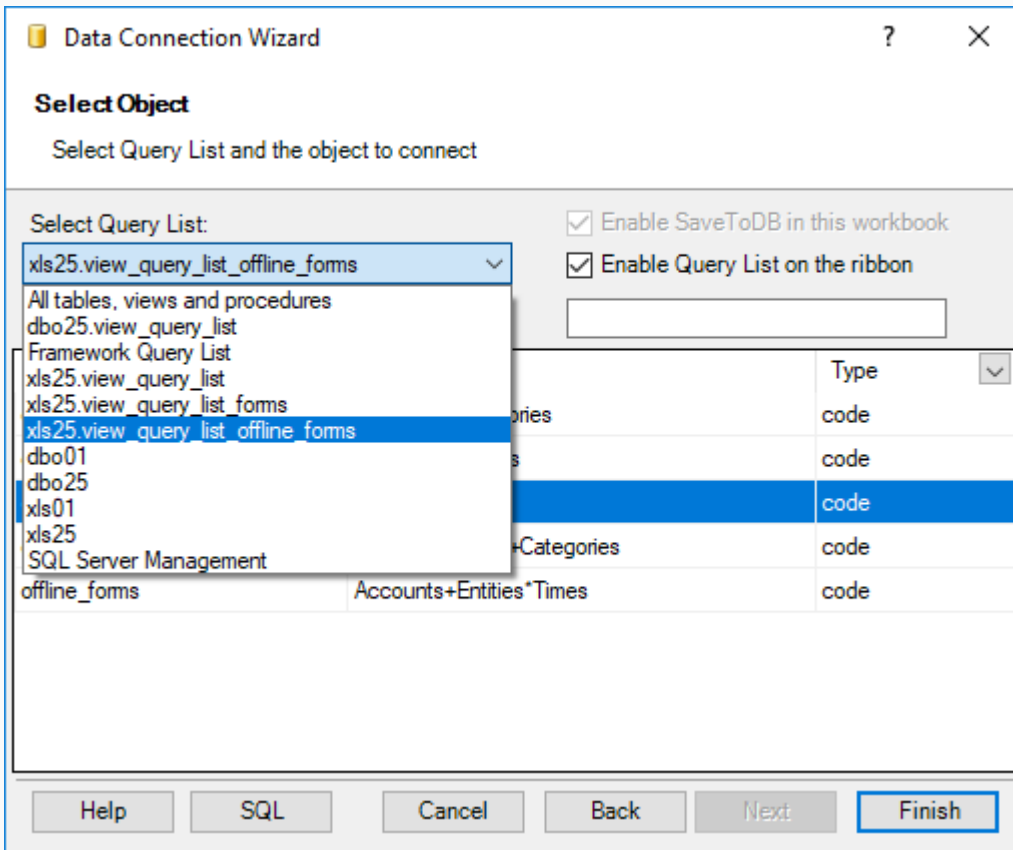
For example, you edit data offline. CFO has closed the period for changes.

When you click **Save**, the application will not save your data into closed periods.

Note that you have to use SaveToDB 7.13 or higher to work with offline forms.

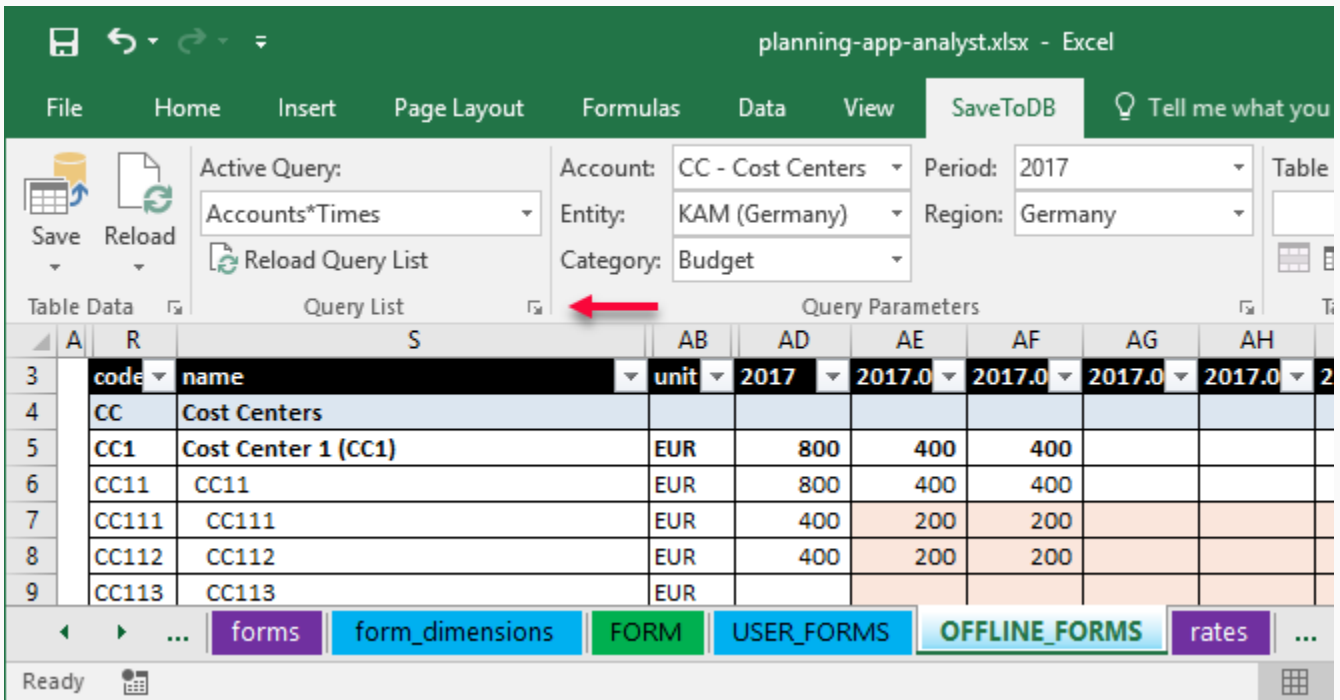
## Online and Offline Forms

You select the form behavior in the **SaveToDB Data Connection Wizard** using the appropriate query list view:



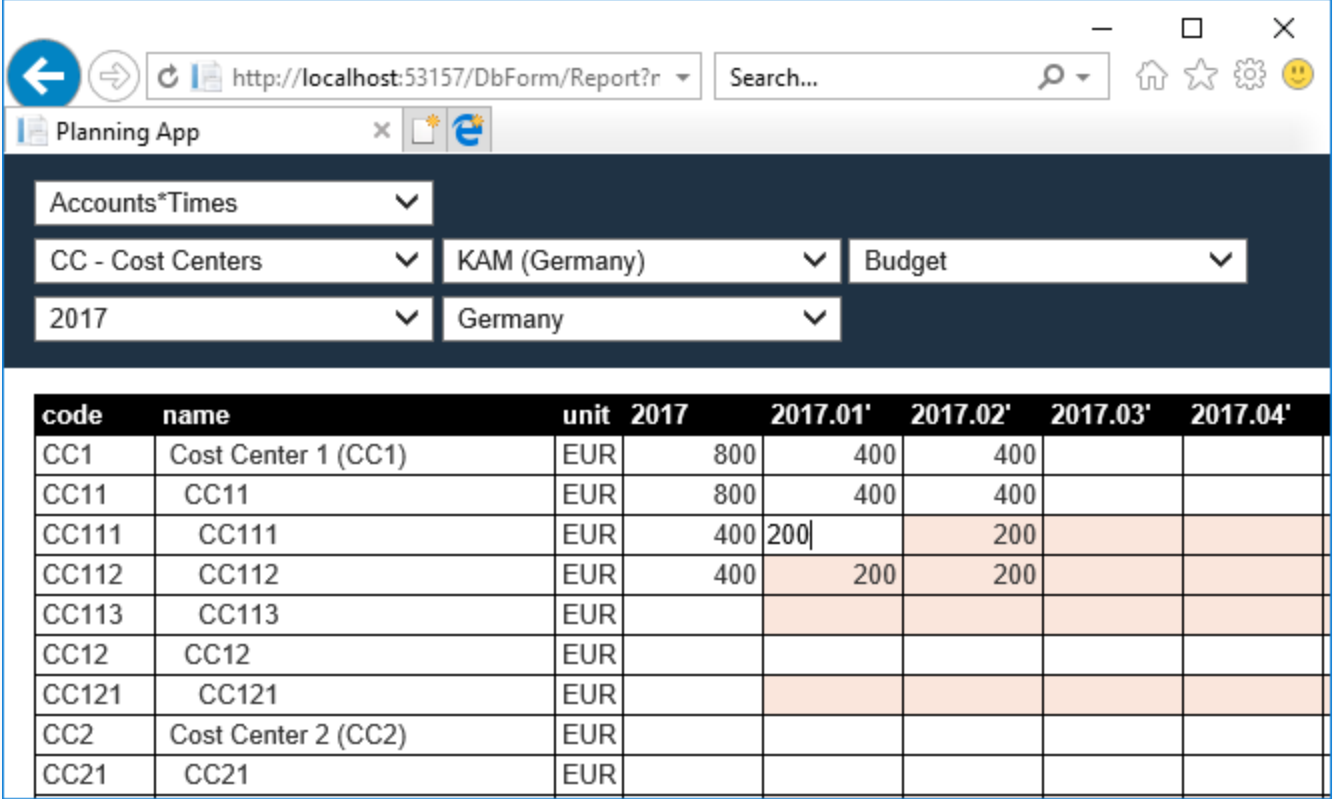
You may distribute application workbooks to end users with any variant or even for both.

Business users may change query lists quickly using the **Query List** group button that opens the same dialog box.



## Web Forms

An additional ASP.NET application allows getting and editing data online on desktops, tablets, and smartphones.



code	name	unit	2017	2017.01'	2017.02'	2017.03'	2017.04'
CC1	Cost Center 1 (CC1)	EUR	800	400	400		
CC11	CC11	EUR	800	400	400		
CC111	CC111	EUR	400	200	200		
CC112	CC112	EUR	400	200	200		
CC113	CC113	EUR					
CC12	CC12	EUR					
CC121	CC121	EUR					
CC2	Cost Center 2 (CC2)	EUR					
CC21	CC21	EUR					

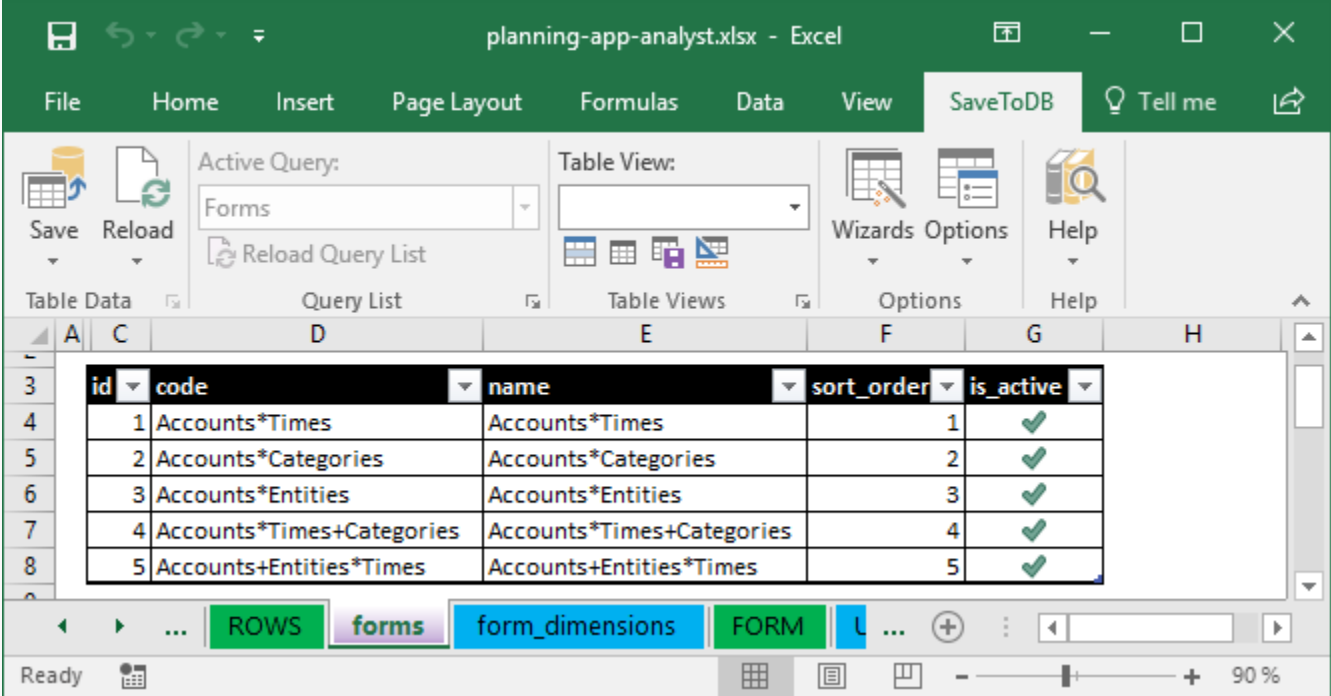
The web application uses the same settings as the Excel application with the SaveToDB add-in, except the conditional formatting. You may customize formatting using CSS styles.

## Form List

We have discussed working with forms above. Below we will discuss configuring forms.

As a business analyst, you may create and configure as many forms as your company need.

First of all, you need to define a form in the **Forms** table. The default app includes the following forms:



id	code	name	sort_order	is_active
1	Accounts*Times	Accounts*Times	1	✓
2	Accounts*Categories	Accounts*Categories	2	✓
3	Accounts*Entities	Accounts*Entities	3	✓
4	Accounts*Times+Categories	Accounts*Times+Categories	4	✓
5	Accounts+Entities*Times	Accounts+Entities*Times	5	✓

Each form must have a unique code and name.

You may change form names anytime. However, try to define form codes correctly before activating.

Do not use spaces in codes.

Default forms use the \* and + signs in the codes and names. This is not necessary but useful.

The \* sign splits row and column dimensions. The + sign splits dimensions in the same axis.

Use friendly codes and names for business forms. For example, CostCenters\*Times or just CostCenters.

In this example, we need a form to show Budget vs. Actual and Budget vs. Forecast by months.

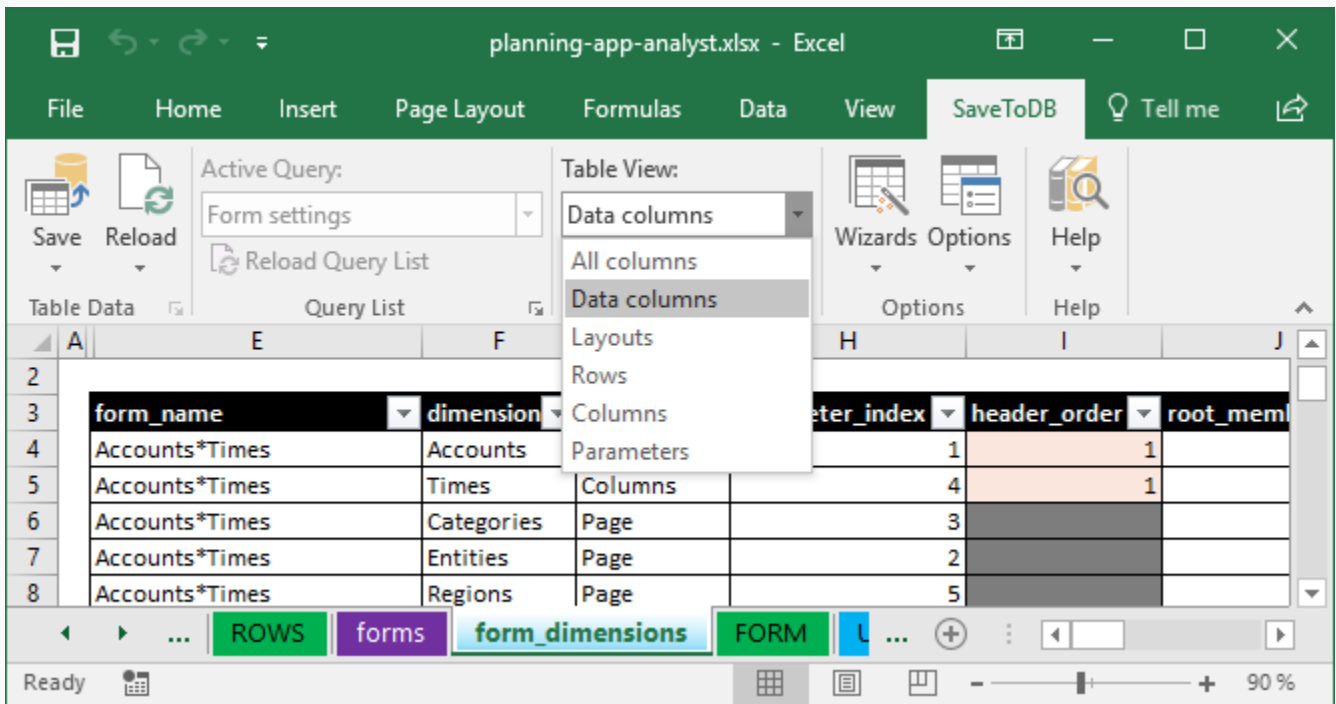
We may use the default Accounts\*Times+Categories form for this.



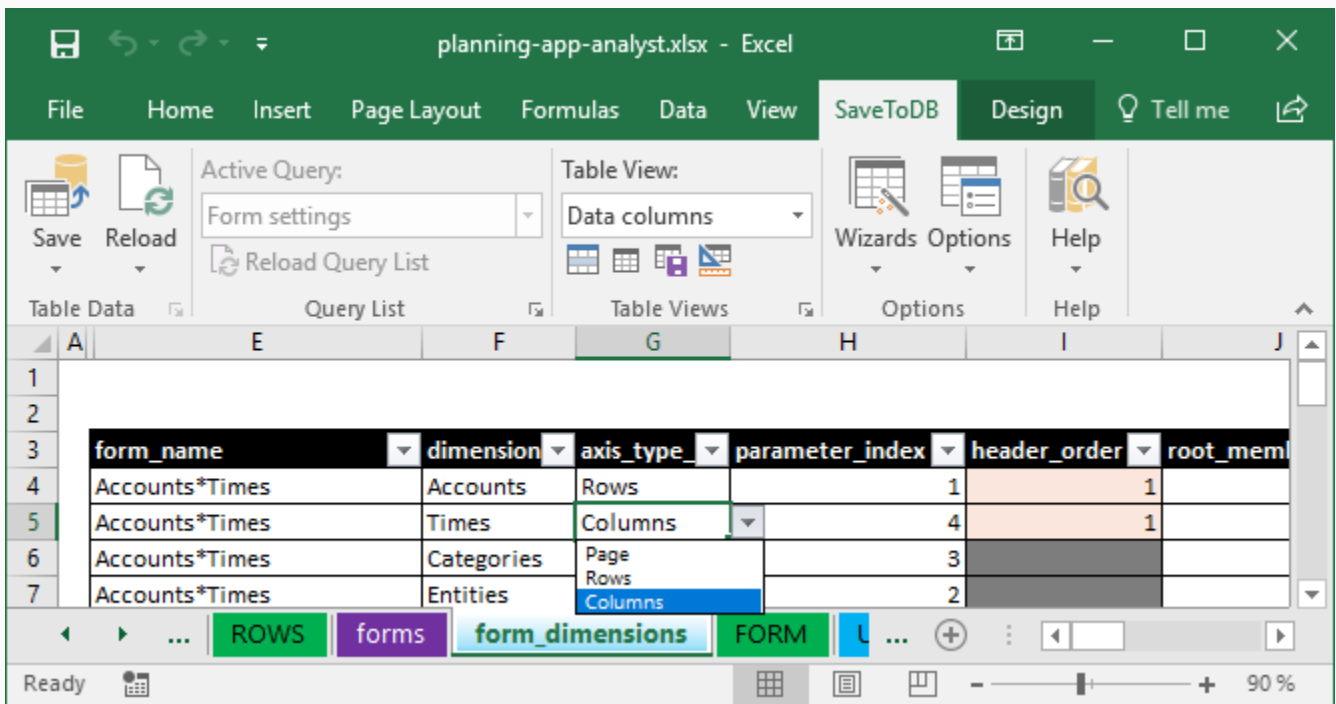
## Form Settings

We will use the **form\_dimensions** worksheet to configure form layouts and parameters.

You may use the built-in table views to show required columns for every mode:



The form uses Excel validation lists to suggest correct values and the conditional formatting to highlight mandatory and disabled cells:



Use **Reload**, **Reload Data** and **Configuration** to reload validation lists if you add or change dimension members.

## Form Layouts

You have to specify the axis for every dimension of each form:

form_name	dimension	axis_type	header_order
Accounts*Times	Accounts	Rows	1
Accounts*Times	Times	Columns	1
Accounts*Times	Categories	Page	
Accounts*Times	Entities	Page	
Accounts*Times	Regions	Page	
Accounts*Categories	Accounts	Rows	1
Accounts*Categories	Times	Page	
Accounts*Categories	Categories	Columns	1
Accounts*Categories	Entities	Page	
Accounts*Categories	Regions	Page	
Accounts*Entities	Accounts	Rows	1
Accounts*Entities	Times	Page	
Accounts*Entities	Categories	Page	
Accounts*Entities	Entities	Columns	1
Accounts*Entities	Regions	Page	
Accounts*Times+Categories	Accounts	Rows	1
Accounts*Times+Categories	Times	Columns	1
Accounts*Times+Categories	Categories	Columns	2
Accounts*Times+Categories	Entities	Page	
Accounts*Times+Categories	Regions	Page	
Accounts+Entities*Times	Accounts	Rows	1
Accounts+Entities*Times	Times	Columns	1
Accounts+Entities*Times	Categories	Page	
Accounts+Entities*Times	Entities	Rows	2
Accounts+Entities*Times	Regions	Page	

The form shows only active dimensions. You activate or deactivate dimensions in the **dimensions** worksheet.

You have to specify the header order for dimensions used in rows and columns.

For example, the Accounts\*Times+Categories form shows columns with times followed by categories like 2017.01 Budget, 2017.01 Actual, 2017.02 Budget, 2017.02 Actual.

Spaces separate member codes in column headers. That because of you must not use spaces in member codes.

## Form Rows

This table view shows columns used to configure form rows.

By default, a user may select a parent member for the row axis. The application will include all children.

This is a good solution for universal forms.

However, if we need to create a specific form with the predefined row set, we may specify the parent directly.

For example, we may set the CC member (Cost Centers) as a root member:

form_name	dimension	header_order	root_member	form_start_level	form_end_level
Accounts*Times	Accounts	1			
Accounts*Categories	Accounts	1			
Accounts*Entities	Accounts	1			
Accounts*Times+Categories	Accounts	1	CC		
Accounts+Entities*Times	Accounts	1			
Accounts+Entities*Times	Entities	2			

In this case, the form would contain only the CC member and its children (see [Hierarchies](#)).

If we want to exclude the CC member, we may set 1 in the **form\_start\_level** column (to exclude level 0).

If we want to show children of certain levels only, we may set the end level in the **form\_end\_level** column.

Another way to configure row sets is using [Rowsets](#). We will discuss this below.

However, we specify the row set member in the same field, **root\_member\_id**.

In this example, we reject the change for the default form. Just click **Reload** and do not save changes.

## Form Columns

This table view shows columns used to configure form columns.

It has the same logic as described above for rows.

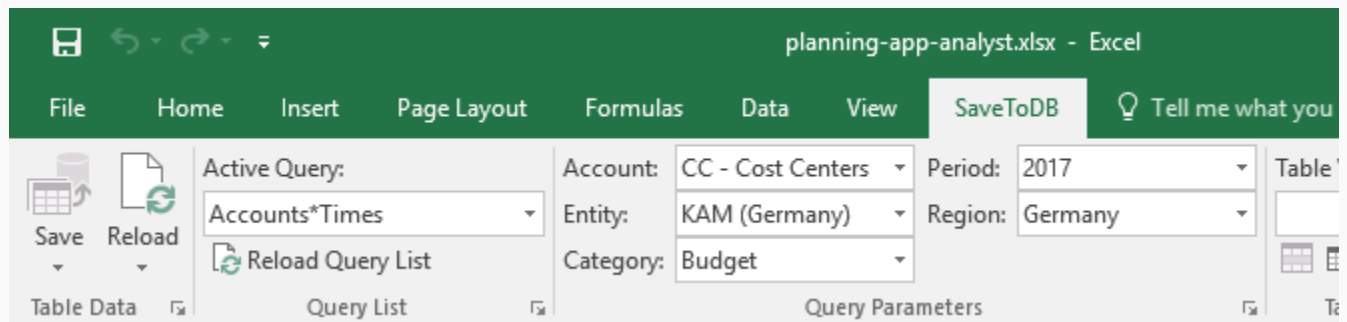
For example, to show only year totals without monthly details, we may set the years\* row set as a root:

form_name	dimension	header_order	root_member	form_start_level	form_end_level
Accounts*Times	Times	1	years*		
Accounts*Categories	Categories	1			
Accounts*Entities	Entities	1			
Accounts*Times+Categories	Times	1			
Accounts*Times+Categories	Categories	2			
Accounts+Entities*Times	Times	1			

We dismiss this change in this example.

## Form Parameters

The application shows form parameters at the ribbon:



You may configure what parameters in what order to show and what members to use as values.

Activate the Parameters table view to show related columns:

form_name	dimension	axis_ty	param	head	root	parameter_start	parameter_end
Accounts*Times	Accounts	Rows	1	1			
Accounts*Times	Times	Columns	4	1	years*		
Accounts*Times	Categories	Page	3				
Accounts*Times	Entities	Page	2				
Accounts*Times	Regions	Page	5				
Accounts*Categories	Accounts	Rows	1	1			
Accounts*Categories	Times	Page	4				
Accounts*Categories	Categories	Columns	3	1			
Accounts*Categories	Entities	Page	2				
Accounts*Categories	Regions	Page	5				
Accounts*Entities	Accounts	Rows	1	1			
Accounts*Entities	Times	Page	4				
Accounts*Entities	Categories	Page	3				
Accounts*Entities	Entities	Columns	2	1			
Accounts*Entities	Regions	Page	5				
Accounts*Times+Categories	Accounts	Rows	1	1			
Accounts*Times+Categories	Times	Columns	4	1			
Accounts*Times+Categories	Categories	Columns	3	2			
Accounts*Times+Categories	Entities	Page	2				
Accounts*Times+Categories	Regions	Page	5				
Accounts+Entities*Times	Accounts	Rows	1	1			
Accounts+Entities*Times	Times	Columns	4	1			
Accounts+Entities*Times	Categories	Page	3				
Accounts+Entities*Times	Entities	Rows	2	2			
Accounts+Entities*Times	Regions	Page	5				

You may reorder parameters using the **parameter\_index** column (shown as param in column 4).

Also, you may remove the parameter from the ribbon clearing the **parameter\_index** value.

In this case, you must specify the root member (column 6 shows as root).

Also, you may change the root member, **parameter\_start\_level**, and **parameter\_end\_level** to show children of the specified parent with required levels in hierarchies.

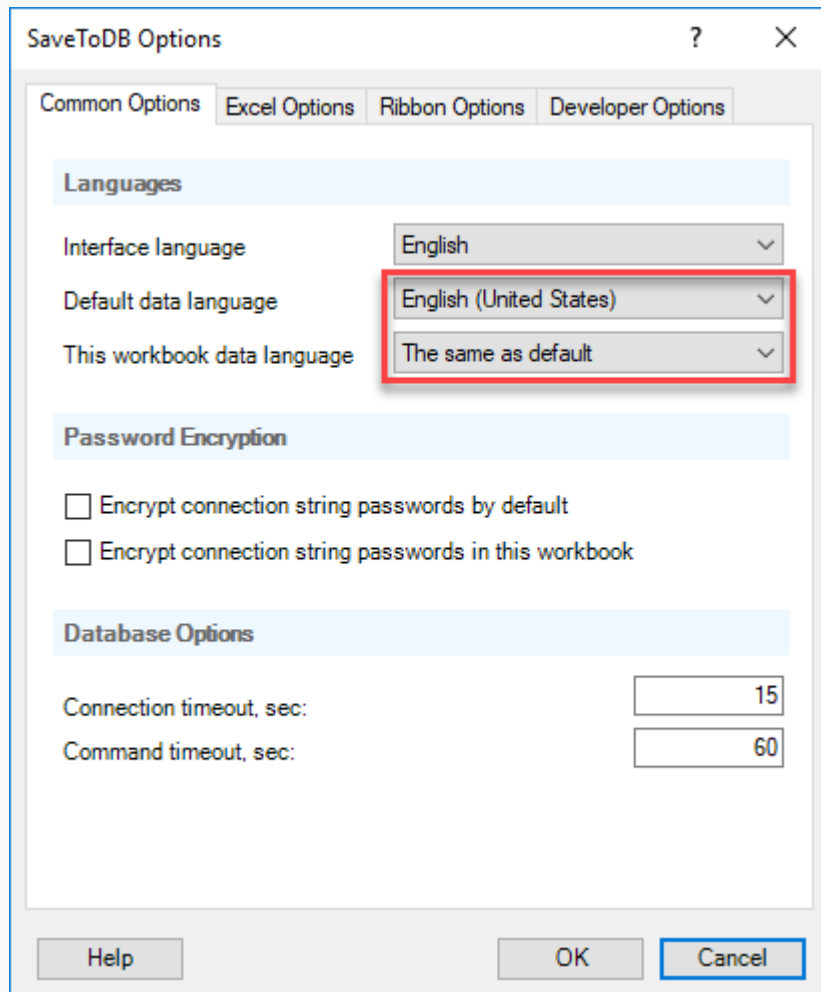
The example shows using the years\* row set as a root member (we dismiss this change).

The application shows parameters using business names.

You may change these names in the **parameter\_name** field of the [dimension configuration](#) discussed above.

id	code	name	parameter_name	is_protected	is_active
1	accounts	Accounts	Account	✓	✓
2	times	Times	Period	✓	✓
3	categories	Categories	Category	✓	✓
4	entities	Entities	Entity	✓	✓
5	dim5	Regions	Region	✗	✓
6	dim6	Products	Product	✗	✗
7	dim7	Subaccounts	Subaccount	✗	✗

Note that to activate translation, users must select data translation in the SaveToDB add-in options:



## Form Rowsets

As shown above, by default, the application uses hierarchies to filter members in rows, columns, and parameters.

For example, when we select 2017 as a parent, the application includes 2017 and all months of 2017.

If we select the YEARS member as a parent, the application includes all year totals and all months.

Row sets allow defining sets of members to use in forms.

For example, the application contains the built-in years\* row set that includes only year totals.

The second reason to use row sets is creating sets of members from different hierarchies.

For example, we have parents: Cost Centers, Sales, Cash Flow, and HR.

To have a form that uses members from these hierarchies, we may create a set and assign members to its rows.

You may check and create row sets for any dimension in the **rowsets** worksheet:

code	name	dimension_id
years*	Years*	101
quarters*	Quarters*	102

You may see the asterisk at the end of codes and names.

This is just a trick used to distinguish hierarchy members and row sets in parameters and validation lists.

You may create your rule.

The application automatically updates these row sets when you add new time members using the **Actions** menu.

Below are the predefined row sets for the Categories dimension:

code	name	sort_order
BUDGET_VS_SAME_PERIOD*	Budget vs Same Period	101
BUDGET_VS_SAME_PERIOD**	Budget vs Same Period+	102
BUDJET_VS_FORECAST*	Budget vs Forecast	103
BUDJET_VS_FORECAST**	Budget vs Forecast+	104
BUDJET_VS_ACTUAL*	Budget vs Actual	105
BUDJET_VS_ACTUAL**	Budget vs Actual+	106
ACTUAL_VS_SAME_PERIOD*	Actual vs Same Period	107
ACTUAL_VS_SAME_PERIOD**	Actual vs Same Period+	108
ACTUAL_VS_FORECAST*	Actual vs Forecast	109
ACTUAL_VS_FORECAST**	Actual vs Forecast+	110
FORECAST_VS_SAME_PERIOD*	Forecast vs Same Period	111
FORECAST_VS_SAME_PERIOD**	Forecast vs Same Period+	112

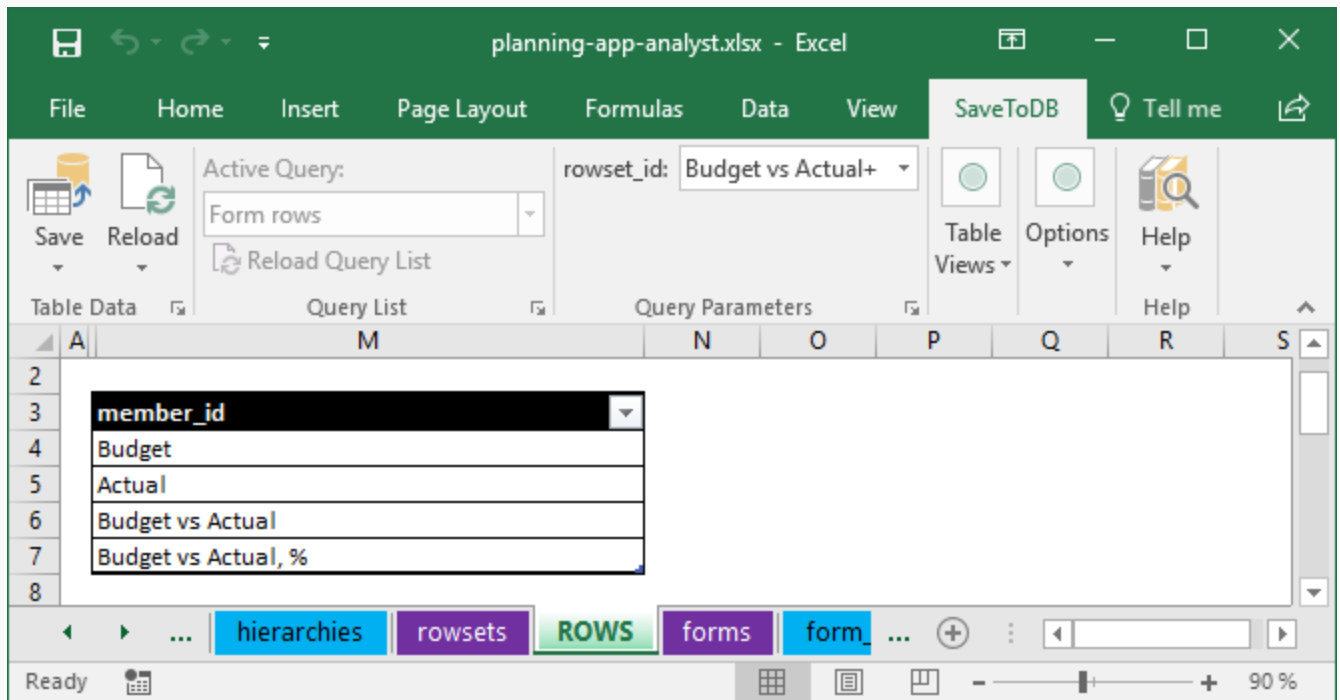
Row sets with one asterisk contain two columns only. For example, Budget and Actual.

Row sets with two asterisks contain base columns and differences.

For example, BUDGET\_VS\_ACTUALS\*\* includes the following columns:

code	name	unit	BUDGET'	ACTUAL'	ACTUAL-BUDGET	ACTUAL-BUDGET,%
CC	Cost Centers					
CC1	Cost Center 1 (CC1)	EUR	400		-400	-100.0%
CC11	CC11	EUR	400		-400	-100.0%
CC111	CC111	EUR	200		-200	-100.0%
CC112	CC112	EUR	200		-200	-100.0%
CC113	CC113	EUR				

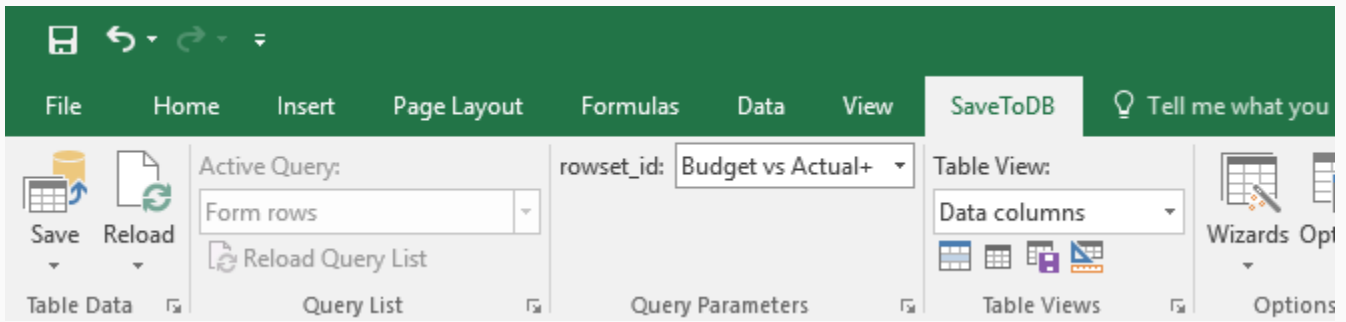
You may check and configure row set rows in the **ROWS** worksheet:



To add a member, copy the line in the desired position using the **Copy and Insert Rows** in the **Table Views** group, change the member, and save the changes.

You may do even more.

Select the **Data columns** table view on the ribbon:



You will see additional columns:

decimal_places	is_percent	row_color	row_bold	row_indent	code	name	member_id
							Budget
							Actual
							Budget vs Actual
							Budget vs Actual, %

You may rewrite any property of the underlying member.

For example, you may change row codes, row colors, and indents of form rows.

Note that you may add empty lines here (do not add empty members to dimensions).

To change row orders, cut and paste lines to the desired position.



## Chapter 4. Permissions

---

Microsoft Excel tables connect directly to a Microsoft SQL Server database using user credentials.

Microsoft SQL Server controls user permissions.

So, a database administrator must add application users to a database and assign application roles.

The planning application has four built-in roles:

1. **planning\_app\_users** - Business users
2. **planning\_app\_analysts** - Business analysts
3. **planning\_app\_administrators** - Application administrators
4. **planning\_app\_developers** - Application developers

Each role has specific predefined permissions for application objects.

In addition to Microsoft SQL Server permissions, the planning application checks additional internal permissions on the forms and dimension members for business users (members of the **planning\_app\_users** role).

Application administrators manage such permissions using the **planning-app-admin.xlsx** workbook.

It is easy. So, business managers may manage application permissions with no requests to IT.

### Database Permissions

This topic is for database administrators.

To add an application user:

1. Create a login.
2. Create a user in the application database.
3. Assign an application role to the user.

For example, you may use the following code to create an SQL Server login (in the master database):

```
CREATE LOGIN pa_user_01 WITH PASSWORD=N'Dev_2011#_Xls4168'
```

You may use the following code to create a user (in the application database):

```
CREATE USER pa_user_01 FOR LOGIN pa_user_01
```

You may use the **sp\_addrolemember** procedure to assign a role:

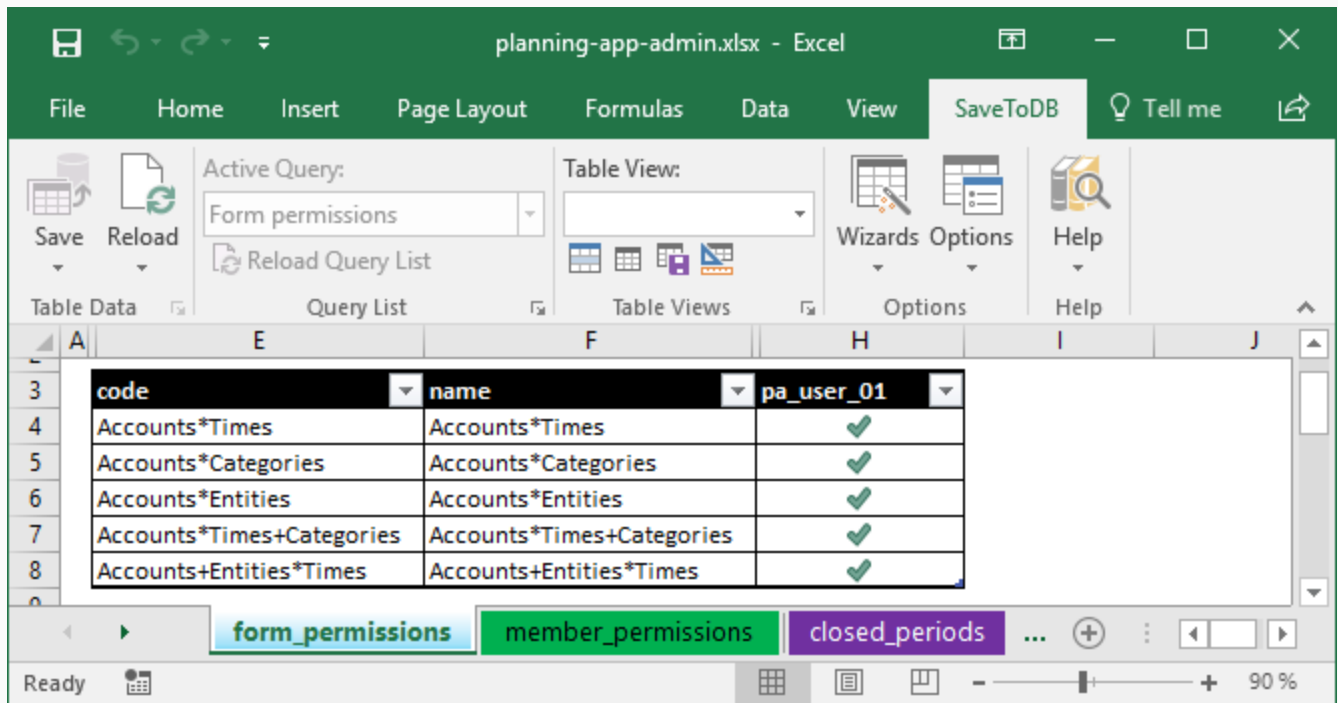
```
EXEC sp_addrolemember 'planning_app_users', 'pa_user_01'
```

As talked above, you may add a user to one of the application roles:

**planning\_app\_users**, **planning\_app\_analysts**, **planning\_app\_administrators**, and **planning\_app\_developers**.

## Form Permissions

To manage form permissions, open the **planning-app-admin.xlsx** workbook and the **form\_permissions** worksheet:

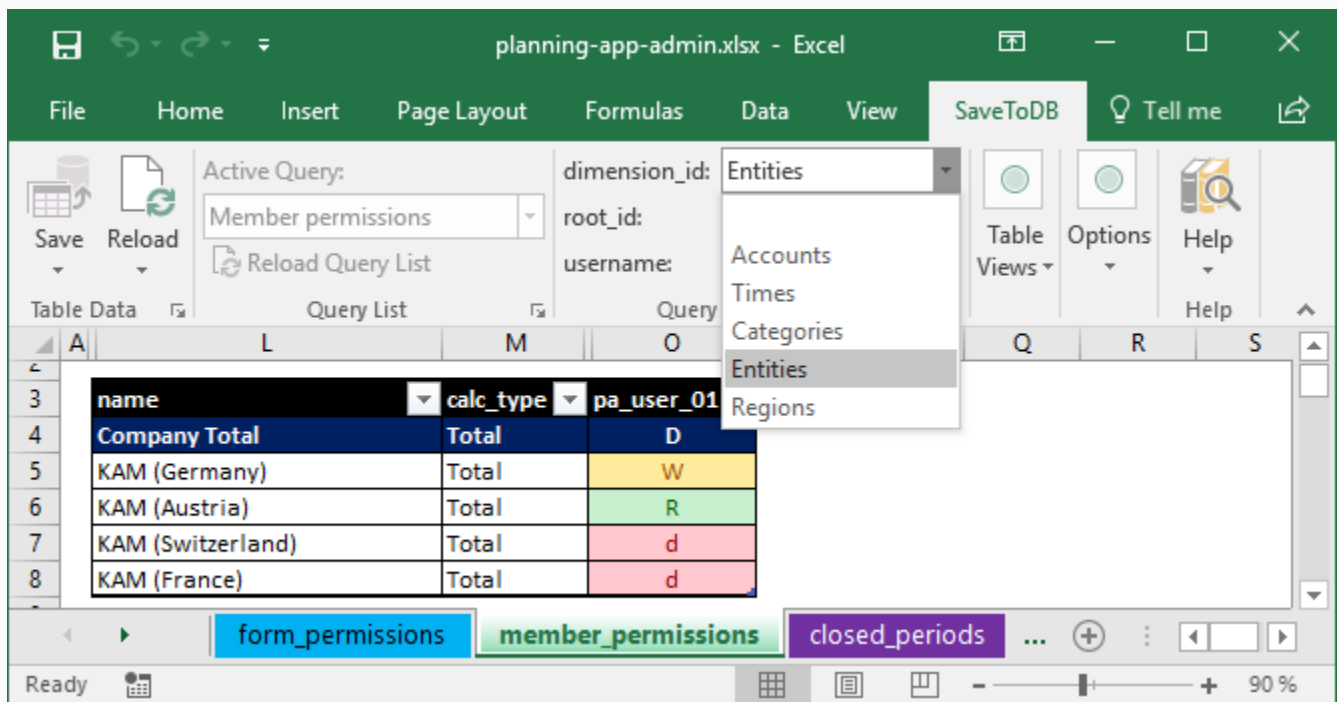


You see a form list configured by business analysts in rows and usernames in columns.

Just set 1 to allow executing the form and 0 to deny executing.

## Member Permissions

To manage member permissions, open the **member\_permissions** sheet and select a dimension on the ribbon:



Here is a legend of used symbols:

- R** - select (read), specified for this member;
- r** - select (read), inherited from a parent member;
- W** - select and update (read and write), specified for this member;
- w** - select and update (read and write), inherited from a parent member;
- D** - deny, specified for this member;
- d** - deny, inherited from a parent member.

In the screenshot above, we may see:

The administrator denies read and write operations for Company Total directly and for its children by inheritance.

However, he allows reading and writing for KAM (Germany) and reading for KAM (Austria).

As a result, the user may see only two entities and may edit data for the first one only.

Type “**r**”, “**w**”, or “**d**” to change permissions.

The application applies for new permissions immediately after changes.

## Base Settings

The application verifies member permissions in accordance with the base [dimension settings](#):

id	code	name	is_protected	default_select_permission	default_update_permission	is_active
1	accounts	Accounts	✓	✗	✗	✓
2	times	Times	✓	✓	✓	✓
3	categories	Categories	✓	✓	✓	✓
4	entities	Entities	✓	✗	✗	✓
5	dim5	Regions	✗	✓	✓	✓
6	dim6	Products	✗	✓	✓	✗
7	dim7	Subaccounts	✗	✓	✓	✗

In this example, we see five active dimensions.

The first four are protected. The application checks permissions for these dimensions only.

The Times and Categories dimensions allow selecting and updating any member by default.

So, you need to deny access to such dimension members if required.

The Accounts and Entities dimensions deny access by default.

So, you need to allow access for such dimension members.

You may find this form in the **dimensions** worksheet in the **planning-app-analyst.xlsx** workbook.

## Account Permissions

Here is an example of account permissions:

name	calc_type	pa_user_01
<b>Cost Center 1 (CC1)</b>	<b>Total</b>	<b>W</b>
CC11	Total	w
CC111	Total	w
CC112	Total	w
CC113	Total	w
CC12	Total	w
CC121	Total	w
<b>Cost Center 2 (CC2)</b>	<b>Total</b>	<b>R</b>
CC21	Total	r
CC211	Total	r
CC212	Total	r
CC213	Total	r
CC22	Total	r
CC221	Total	r
<b>Cost Center 3 (CC3)</b>	<b>Total</b>	<b>D</b>
CC31	Total	d
CC311	Total	d
CC312	Total	d
CC313	Total	d
CC32	Total	d
CC321	Total	d

Typically, you set permissions for upper-level members, and the application applies permissions to child members.

You may tune permission for any child member. Just change its permissions.

## Entity Permissions

Here is an example of entity permissions:

name	calc_type	pa_user_01
<b>Company Total</b>	<b>Total</b>	<b>D</b>
KAM (Germany)	Total	W
KAM (Austria)	Total	R
KAM (Switzerland)	Total	d
KAM (France)	Total	d

Typically, this is a protected dimension, and you have to allow access directly.

## Time Permissions

Here is an example of time dimension permissions:

name	calc_type	pa_user_01
2017.00	Total	
<b>2017</b>	<b>Total</b>	
2017.01	Total	
2017.02	Total	
2017.03	Total	
2017.04	Total	
2017.05	Total	
2017.06	Total	
2017.07	Total	
2017.08	Total	
2017.09	Total	
2017.10	Total	
2017.11	Total	
2017.12	Total	
<b>2017.Q</b>	<b>Total</b>	
2017.Q1	Total	
2017.Q2	Total	
2017.Q3	Total	
2017.Q4	Total	

Typically, this is a protected dimension that allows reading and writing by default. So, you may leave this as is.

## Category Permissions

Here is an example of category dimension permissions:

name	calc_type	pa_user_01
Budget	Total	
Forecast	Total	
Actual	Total	

Typically, this is a protected dimension that allows reading and writing by default. So, you may leave this as is.

Note that you see dimension members that may contain input data. You do see calculated members.

## Region Permissions

Here is an example of region dimension permissions:

name	calc_type	pa_user_01
.	Total	
All Countries	Total	
Germany	Total	
Austria	Total	
Switzerland	Total	
France	Total	

In this example, the region dimension is not protected. So, the application does not control access, and you do not need to change its permissions.

## Closing Periods

Use the **closed\_periods** worksheet to prevent changes in closed periods:

Set 0 to close periods for changes

code	BUDGET	FORECAST	ACTUAL
2017.09	×	×	×
2017.10	×	×	×
2017.11	×	✓	×
2017.12	×	✓	×

For example, in this case, we closed all periods for all categories except the November and December forecast. The application checks these settings for all users including business analysts and application developers. The **planning-app-analyst.xlsx** workbook also contains this form, and business analysts may close periods too.

## Chapter 5. Integration

---

You may use two basic methods to import data into the planning application:

1. Copy and paste data from existing Excel workbooks.
2. Import data using SQL commands.

If you have a lot of data, you may contact us to discuss other methods.

### Copying Data

It is a simple method as it requires time only.

You may use a trick. Create a row set with accounts to import and a form that uses this row set.

In this case, you may import data for all periods and any entity and category at once.

Also, you may create an export form in the source workbook and paste data into existing forms.

Note that you have to open the periods for changes to insert data.

### Importing Data

You may use the **dbo25.usp\_import\_fact** procedure to import data easily.

For example, you may use commands like this:

```
EXEC dbo25.usp_import_fact 200, 'CC111', '2017.01', 'ACTUAL', 'KAM.DE', 'COUNTRY.DE';  
EXEC dbo25.usp_import_fact 200, 'CC111', '2017.02', 'ACTUAL', 'KAM.DE', 'COUNTRY.DE';  
EXEC dbo25.usp_import_fact 200, 'CC112', '2017.01', 'ACTUAL', 'KAM.DE', 'COUNTRY.DE';  
EXEC dbo25.usp_import_fact 200, 'CC112', '2017.02', 'ACTUAL', 'KAM.DE', 'COUNTRY.DE';
```

You may export such commands from your accounting system and import using the sqlcmd utility.

Note that the procedure accepts member codes as parameters.

You may contact us to discuss other methods.

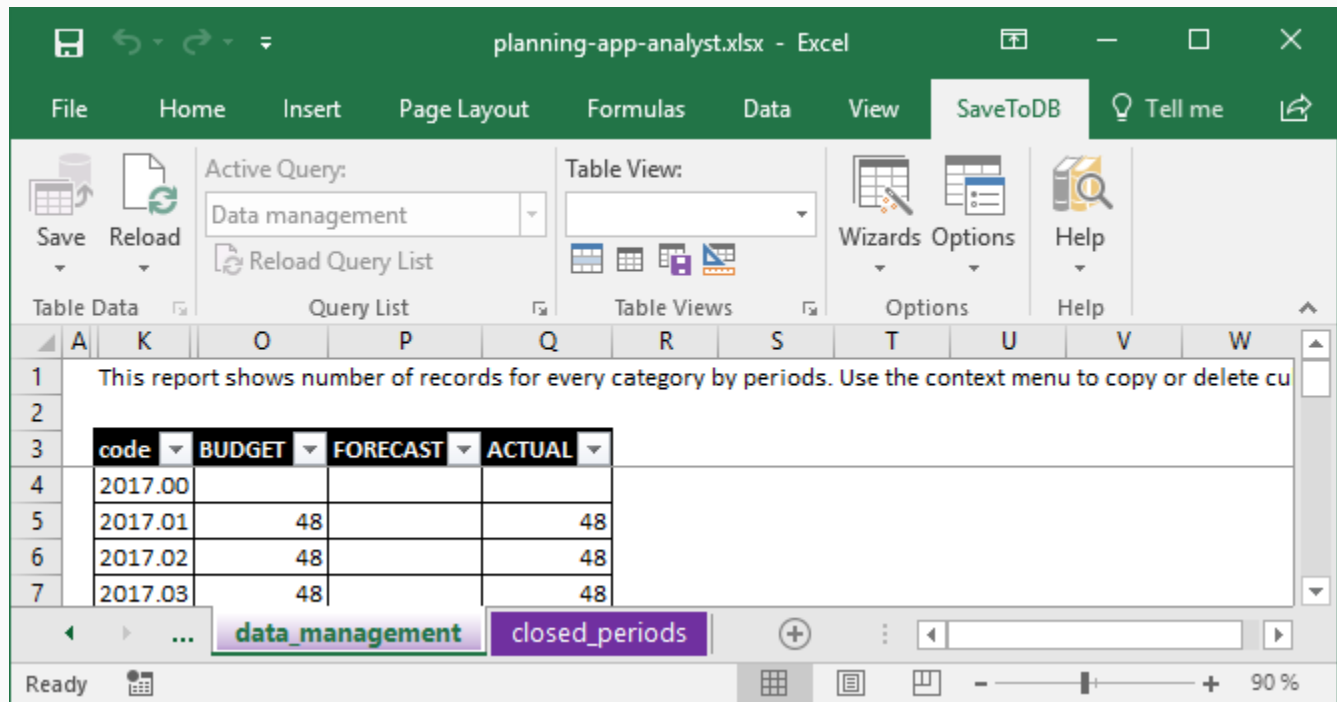
## Chapter 6. Data Management

Periodically, we need to create a copy of the current data.

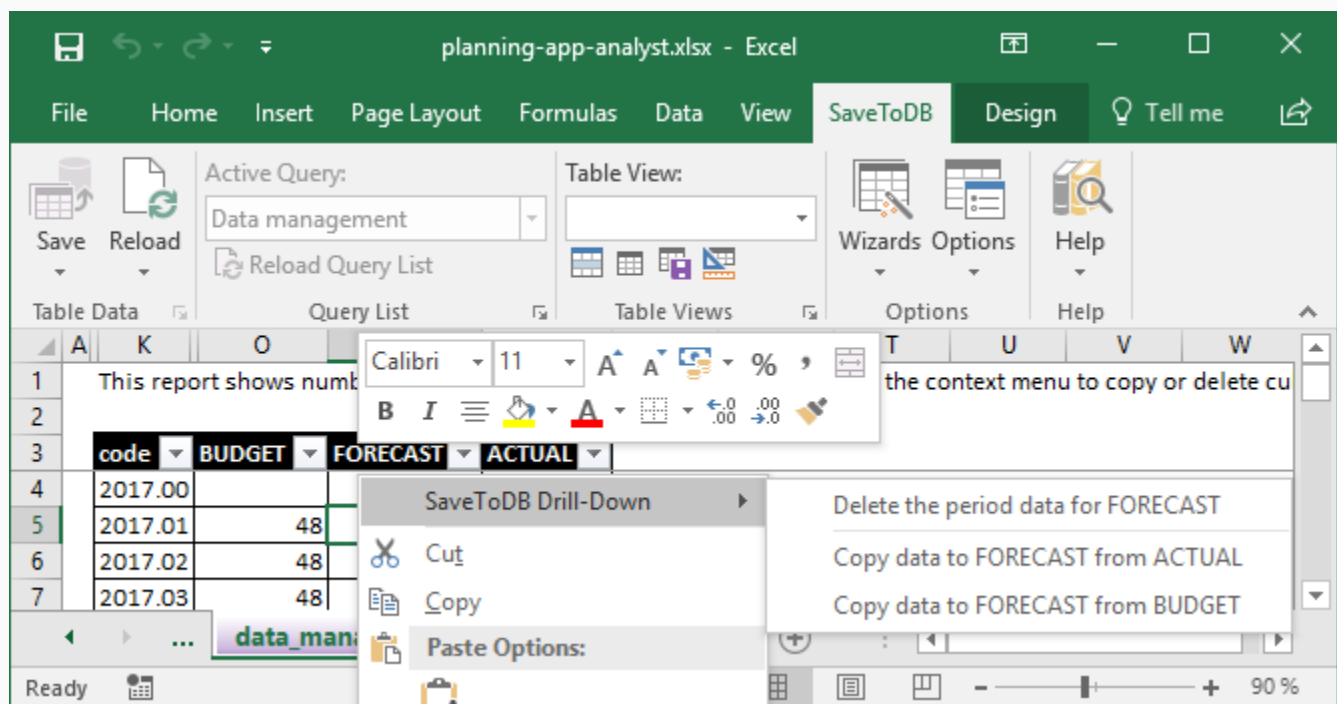
For example, we may fill the forecast data with actual data for past months and with budget data for future ones.

Or, we may create several budget and forecast versions to make a decision about right numbers.

You may copy data between periods and categories using the **data\_management** worksheet:



Just right-click on the desired cell and choose the operation (the menu is dynamic):



For example, we may use these operations to prepare forecast data for this case.



## Chapter 7. Distribution

---

In the previous steps, we have created dimension members, configured forms and permissions, imported data.

Now we may distribute the workbooks to our business users.

We have two choices:

1. We may send or publish a ready-to-use workbook.
2. We may send an email with database connection credentials.

In any case, the application uses the configuration stored in a database. So, both ways have the same features.

Moreover, you may change settings later. Business users just update data and configuration.

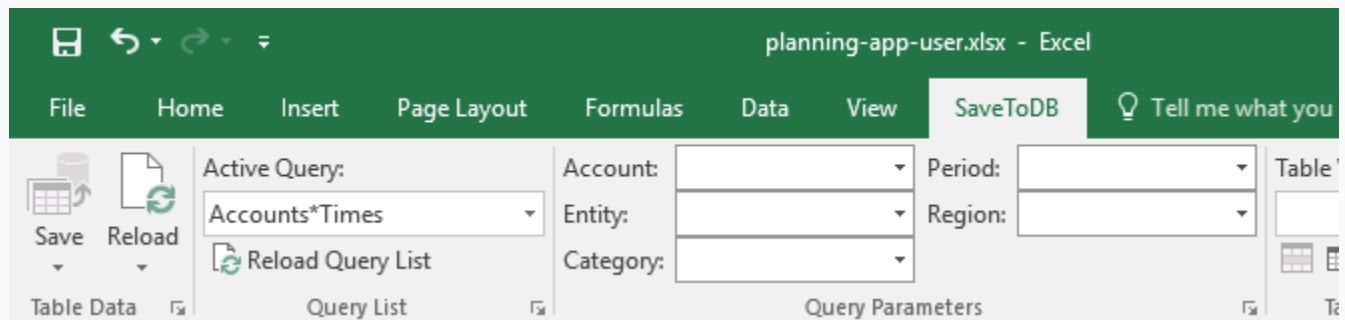
So, you distribute workbooks once.

Note that your users must have the SaveToDB Enterprise 7.13 or higher installed.

### Distributing Workbooks

You may create a workbook for business users, create worksheets, and connect to tables.

You may leave parameters empty. So, users will not see sensitive data.



When a user opens the workbook, he or she must click **Reload, Reload Data and Configuration**.

This action updates parameter values actual for the user.

Then, a user selects parameter values and get the form with actual data, and may input data.

When you create a new form, you may notify your users. They just click **Reload Query List** and get new forms.

### Protecting Connect Credentials

Usually, you create workbooks using your credentials.

If you use the Windows authentication, you may send the workbook as is.

When a user reloads data, the application connects with user's credentials, not yours.

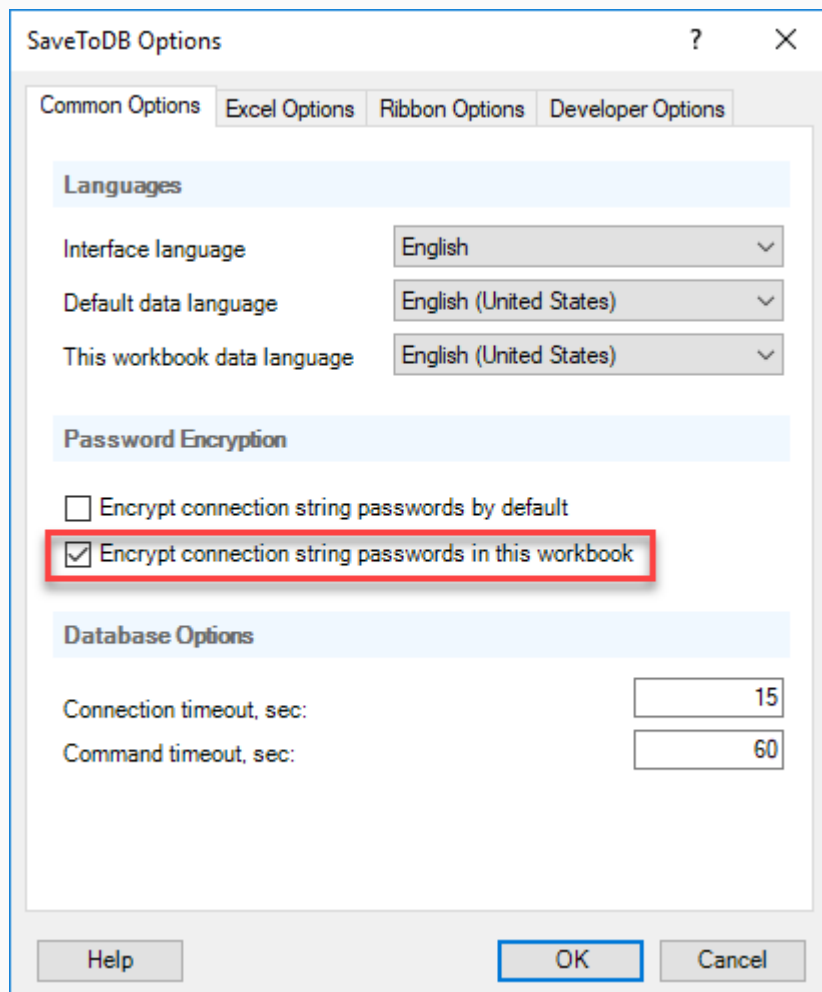
If you use the SQL Server authentication, you must protect your password.

You may clear it. This is a standard Excel feature.

In this case, Excel prompts a user for a password, and the user uses his login and password.

However, you may use the SaveToDB add-in feature designed for this case specially.

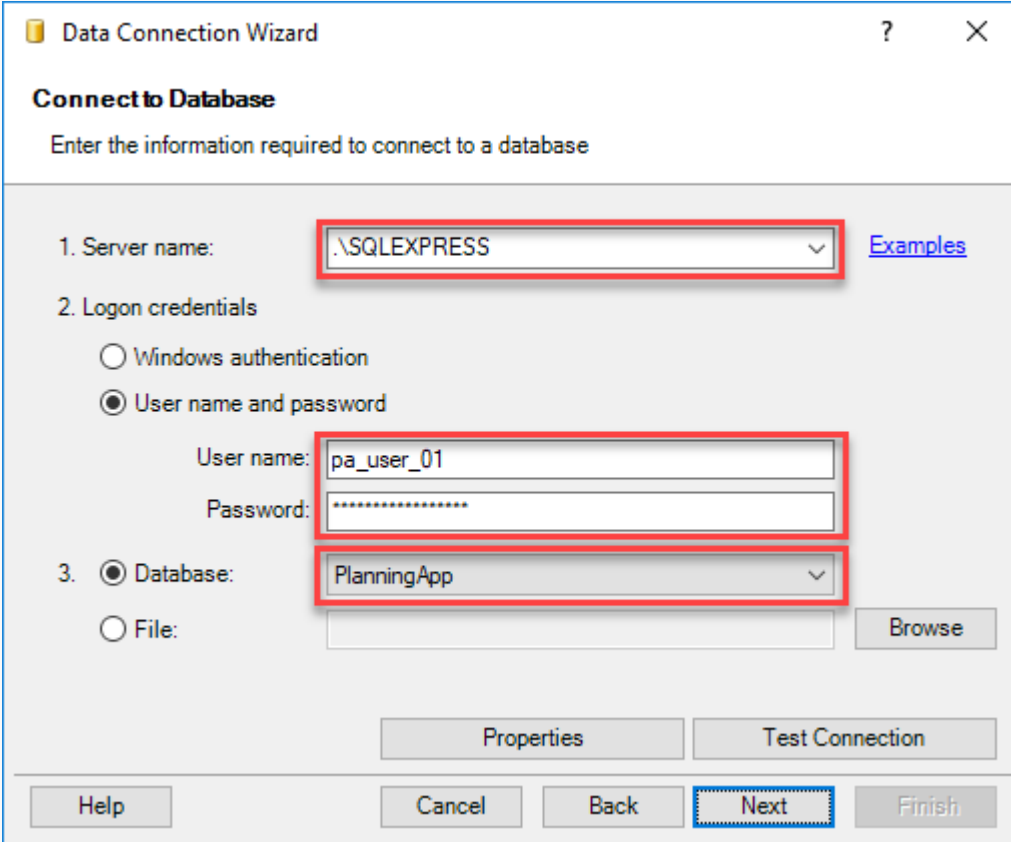
You may open SaveToDB options and check **Encrypt connection string passwords in this workbook**:



In this case, the SaveToDB add-in encrypts passwords using Windows encryption features, and no one can see or use your passwords.

## Distributing using Connect Credentials

The planning application loads application settings from a database. This is a SaveToDB add-in feature. So, you may send to your users connect credentials shown below, and they may connect themselves.



The screenshot shows the 'Data Connection Wizard' dialog box, titled 'Connect to Database'. The instructions prompt the user to 'Enter the information required to connect to a database'. The wizard is divided into three numbered steps:

- 1. Server name:** A dropdown menu is set to '.\SQLEXPRESS'. A blue 'Examples' link is visible to the right.
- 2. Logon credentials:** Two radio buttons are present: 'Windows authentication' (unselected) and 'User name and password' (selected). Below these, the 'User name' field contains 'pa\_user\_01' and the 'Password' field is filled with asterisks.
- 3. Database:** A radio button for 'Database:' is selected, with a dropdown menu set to 'PlanningApp'. An unselected 'File:' radio button is also present, accompanied by a 'Browse' button.

At the bottom of the dialog, there are several buttons: 'Help', 'Cancel', 'Back', 'Next' (highlighted with a blue dashed border), and 'Finish'. Additionally, 'Properties' and 'Test Connection' buttons are located above the 'Next' button.

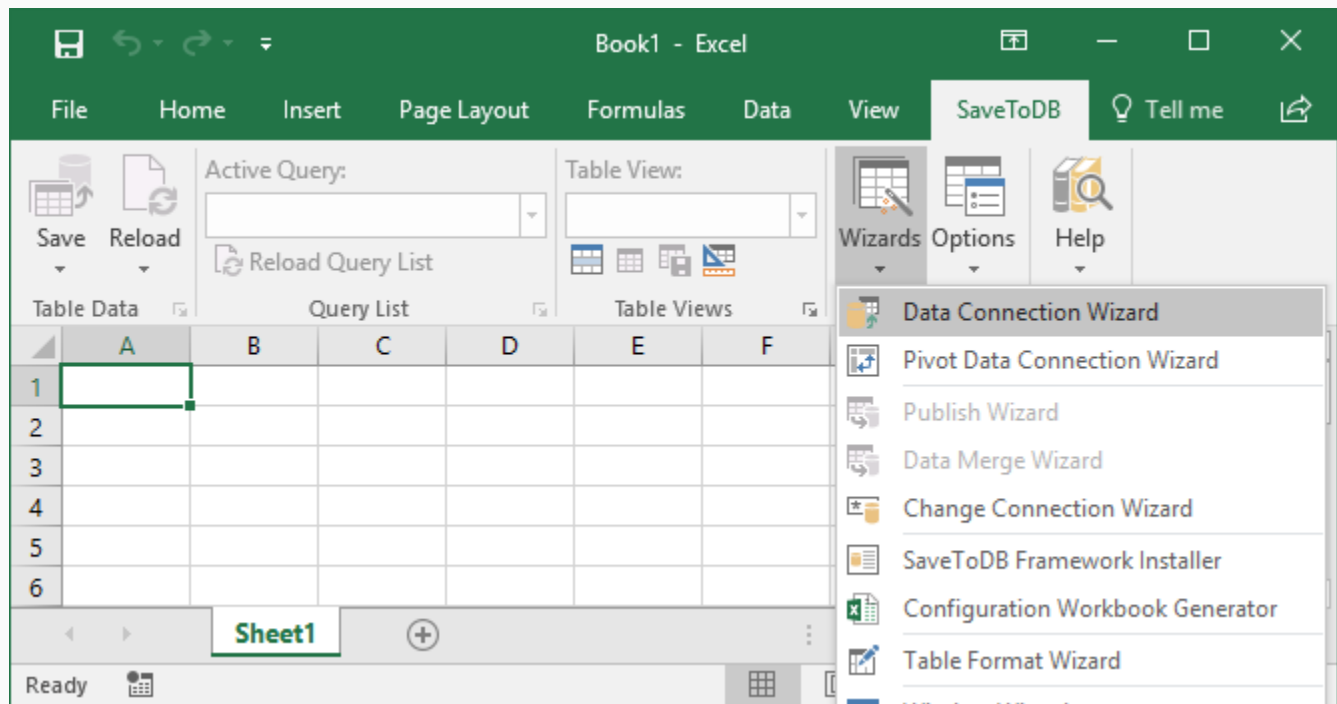
Below is a complete instruction. You may send a link to the online version also.

<https://www.savetodb.com/connecting-to-planning-app.htm>

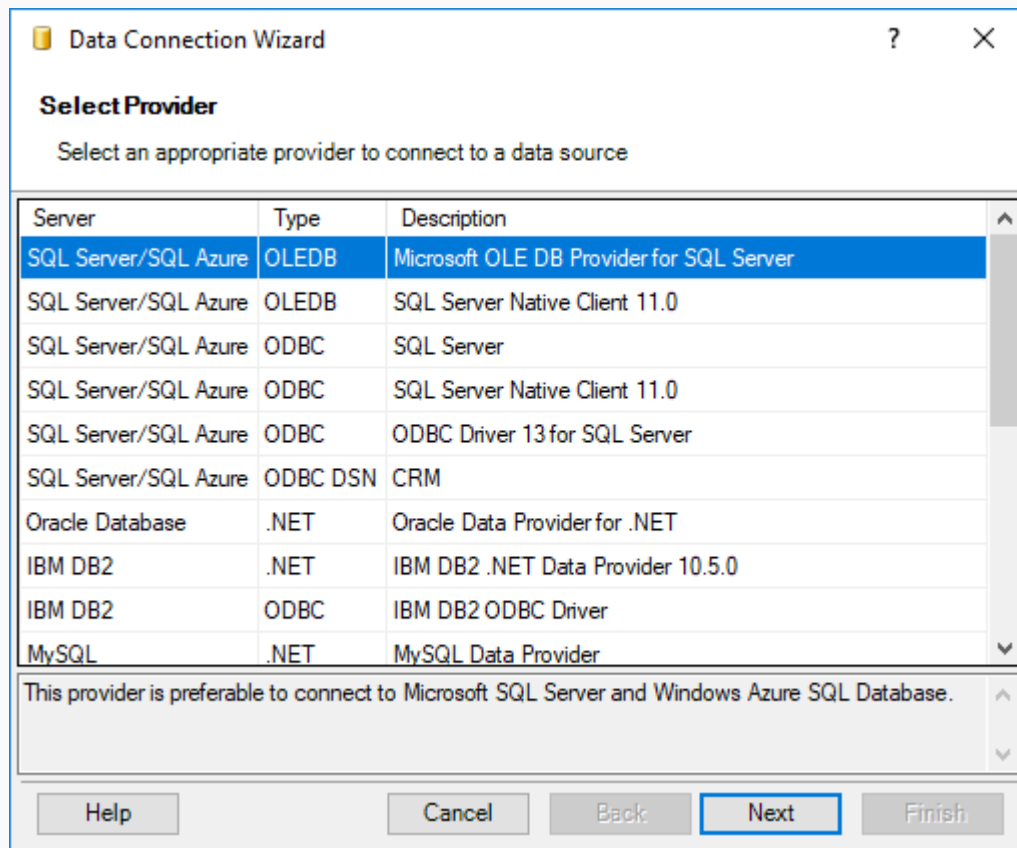
Users do this operation once. They may save a workbook and use it further.

## Connecting to Planning Application

To connect to the application, run **SaveToDB, Wizards, Data Connection Wizard**:



Select **Microsoft OLE DB Provider for SQL Server** and click **Next**:



Fill-in the server name, your username and password, and select a database. Click **Next**.

The screenshot shows the 'Data Connection Wizard' dialog box, titled 'Connect to Database'. The instruction is 'Enter the information required to connect to a database'. The dialog is divided into three numbered steps:

- 1. Server name:** A dropdown menu is set to '.\SQLEXPRESS', which is highlighted with a red box. A blue 'Examples' link is to the right.
- 2. Logon credentials:** The 'User name and password' radio button is selected. The 'User name' field contains 'pa\_user\_01' and the 'Password' field is masked with dots, both highlighted with a red box.
- 3. Database:** The 'Database' radio button is selected, and the dropdown menu is set to 'PlanningApp', highlighted with a red box. A 'File:' radio button and a 'Browse' button are also present.

At the bottom, there are buttons for 'Properties', 'Test Connection', 'Help', 'Cancel', 'Back', 'Next' (highlighted with a blue dashed border), and 'Finish'.

Select **xls23.view\_query\_list\_form** in the **Query List** and leave all check-boxes checked. Click **Finish**.

The screenshot shows the 'Data Connection Wizard' dialog box, titled 'Select Object'. The instruction is 'Select Query List and the object to connect'. The dialog is divided into two main sections:

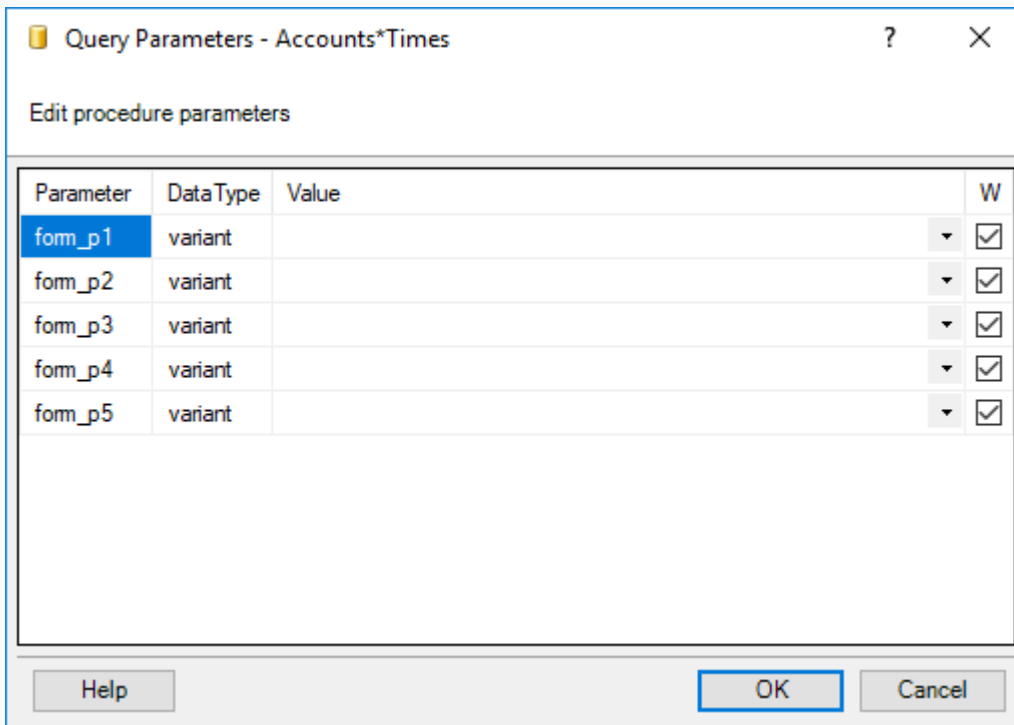
- Select Query List:** A dropdown menu is open, showing a list of query lists. 'xls23.view\_query\_list\_form' is selected and highlighted in blue. Other items include 'All tables, views and procedures', 'xls01.viewQueryList', 'xls25.view\_query\_list', 'xls25.view\_query\_list\_forms', 'xls25.view\_query\_list\_offline\_forms', and 'SQL Server Management'.
- Options:** Two checkboxes are checked: 'Enable SaveToDB in this workbook' and 'Enable Query List on the ribbon'. There is an empty text input field below them.

Below the options is a table with columns for 'Name', 'Description', and 'Type':

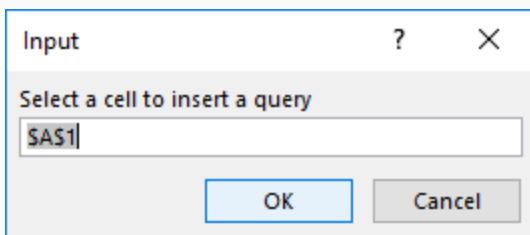
Name	Description	Type
forms	Accounts*Entities	code
forms	Accounts*Times	code
forms	Accounts*Times+Categories	code
forms	Accounts+Entities*Times	code

At the bottom, there are buttons for 'Help', 'SQL', 'Cancel', 'Back', 'Next', and 'Finish' (highlighted with a blue border).

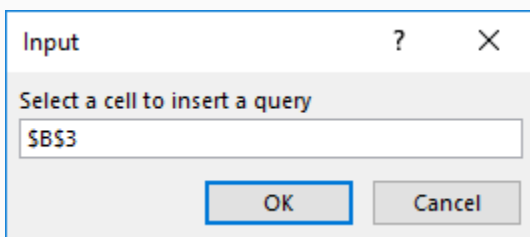
You may select parameter values here. Or just click **OK**.



You will see a dialog box to choose a cell to insert a table.



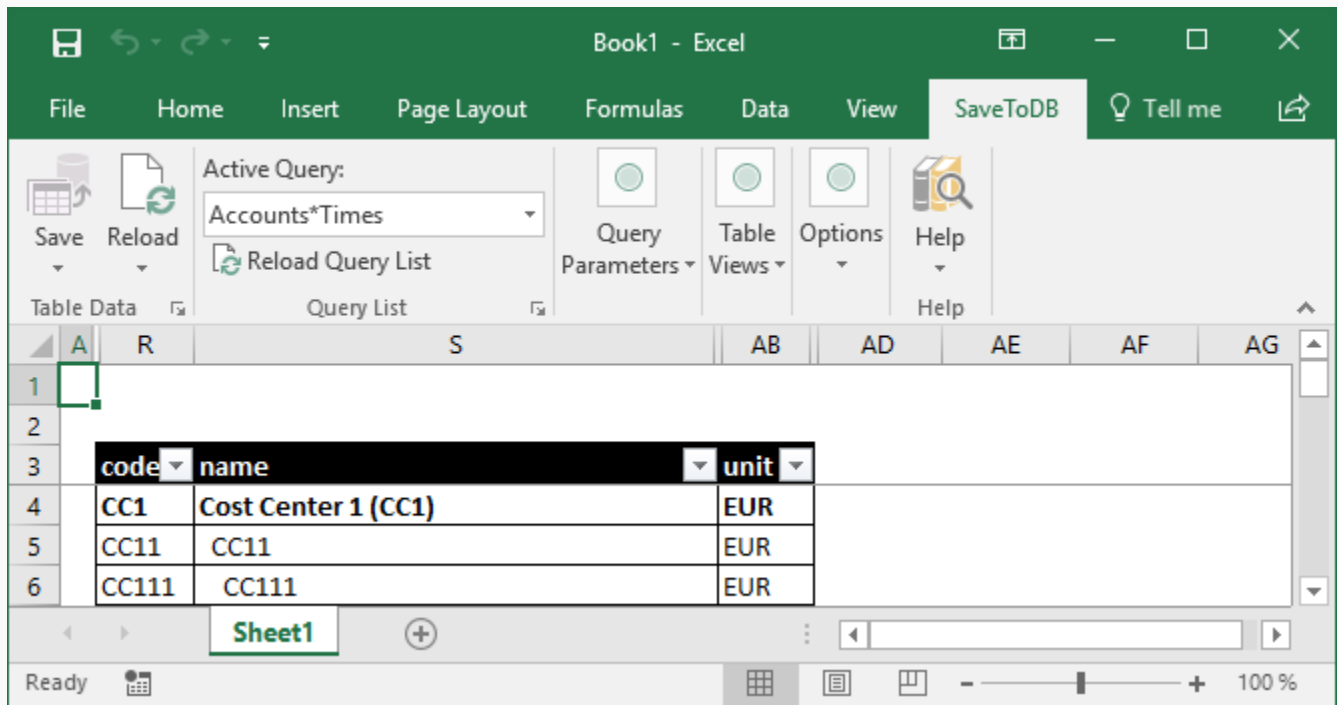
Select cell B3 in the worksheet:



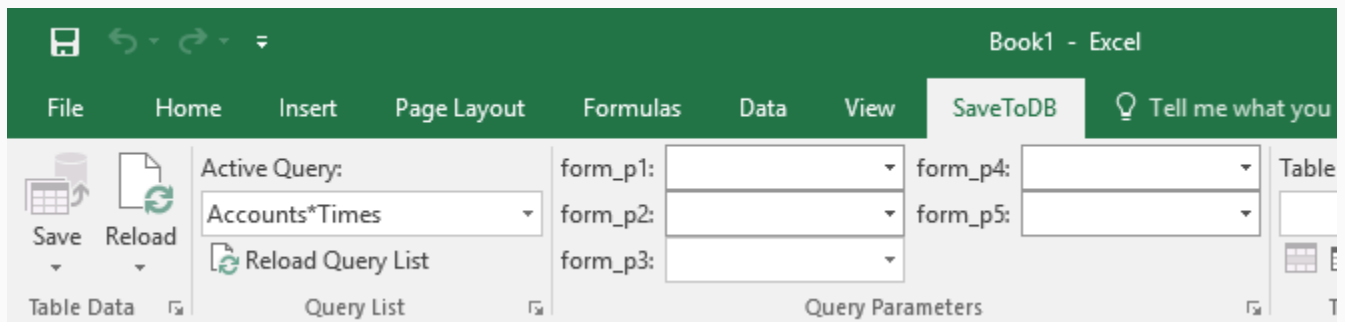
When you insert a table into cell B3, you have the following benefits:

1. You may use row 1 to place formulas and comments.
2. You may use row 2 to filter rows using auto-filters. Just type filter values like >10000 or \*CC11\*. Delete the cell value to clear the auto-filter.
3. You have an empty column A. The table looks better.

The wizard inserts a table, with empty data columns if you do not specify the parameter values.

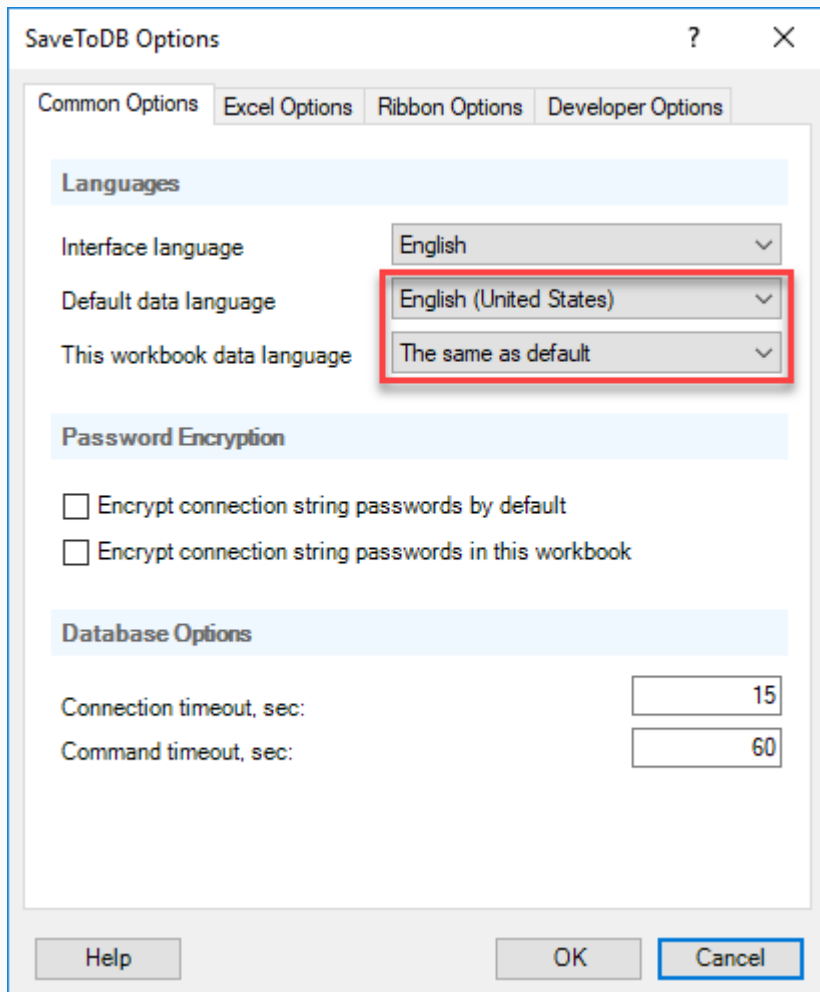


Increase the window width to see the complete ribbon:



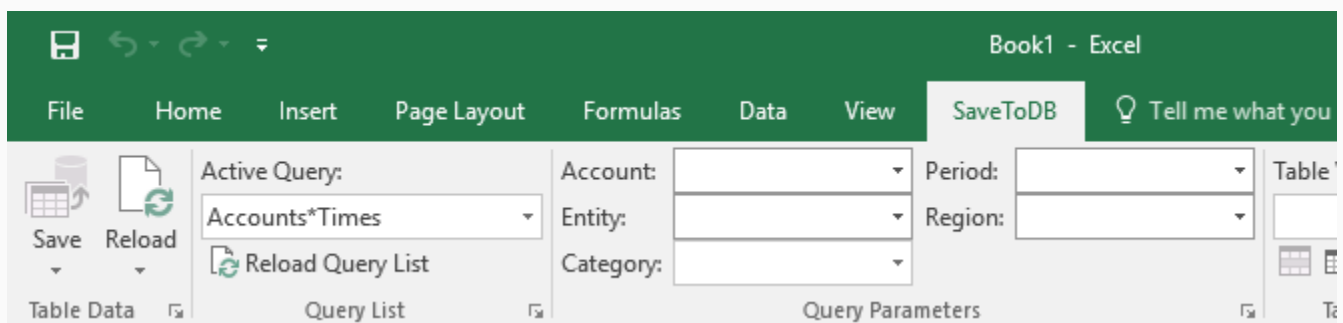
You see parameter names used in a database if you do not turn on the translation.

To turn on the translation, open SaveToDB **Options** and select the desired language:



Click **OK**. Then click **Reload, Reload Data and Configuration**.

You will see the translated parameter names:





Select parameter values and voilà!

The screenshot shows the Microsoft Excel interface with the 'SaveToDB' ribbon tab active. The 'Active Query' is set to 'Accounts\*Times'. The 'Account' is 'CC - Cost Centers', 'Entity' is 'KAM (Germany)', and 'Category' is 'Forecast'. The 'Period' is '2017' and 'Region' is 'Germany'. The 'Table Data' section shows a table with the following data:

code	name	unit	2017	2017.01	2017.02	2017.03
CC1	Cost Center 1 (CC1)	EUR	12 400	1 000	1 000	1 000
CC11	CC11	EUR	10 000	800	800	800
CC111	CC111	EUR	2 600	200	200	200

You may edit highlighted cells.

You may paste formulas into form cells. However, the application replaces them with values while reloading data.

Place your formulas under the table. In this case, you may copy and paste values if the source data changed.

You may use **Undo** (Ctrl-Z) to undo changes.

Use **Reload** to reload data with newly calculated aggregates.

You may select new parameter values using the ribbon.

You may change forms using the **Active Query** drop-down list.

Use **Reload Query List** to reload the list and get new forms configured in a database.

You may copy worksheets. Just copy the worksheet and click **Reload, Reload Data and Configuration**.

Use **Reload, Reload All Workbook Tables** to reload all tables at once.

Sometimes, you may see incorrect column orders because of Excel adds new columns at the end.

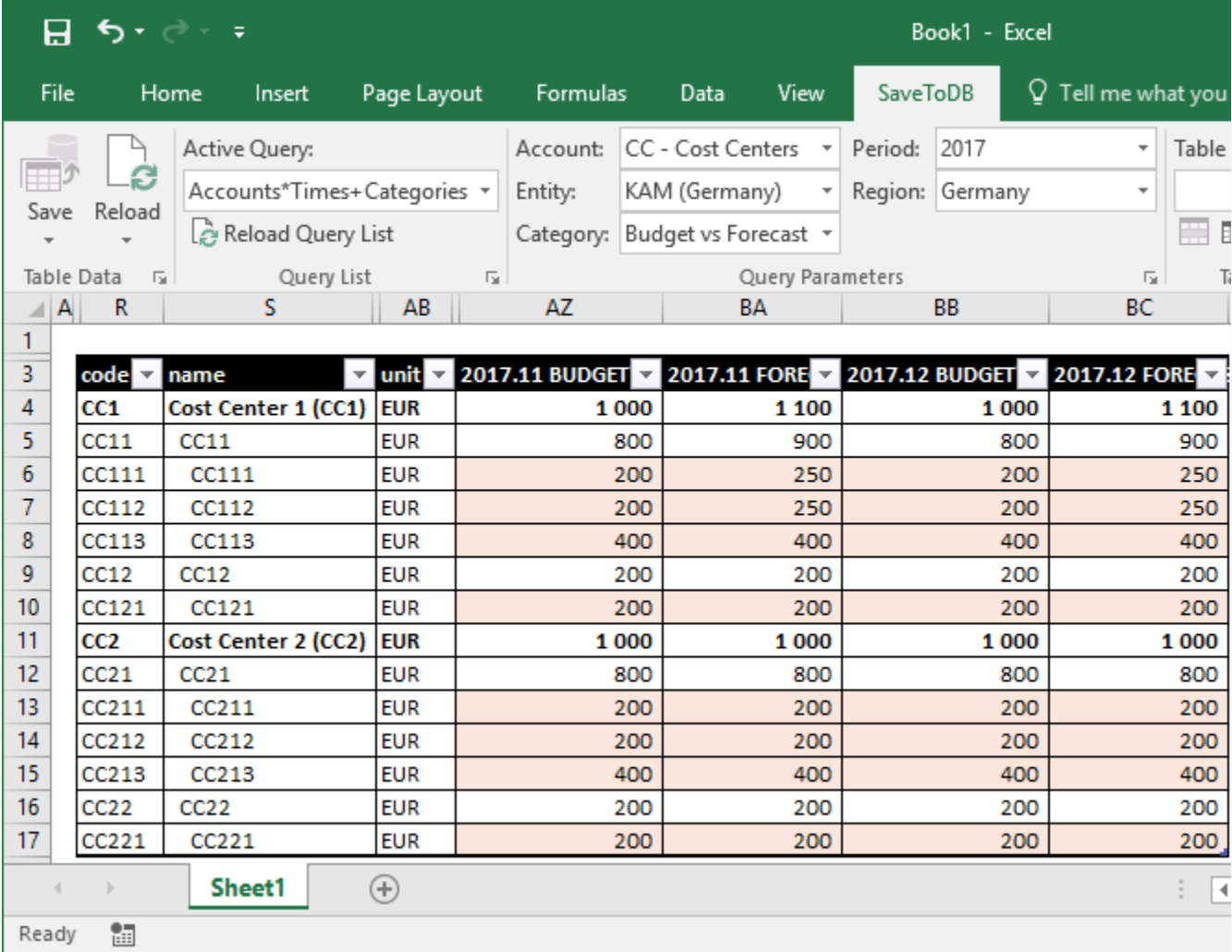
Just select an empty value in any parameter to clear columns and then select the desired value.

You may change the default table formatting. Use regular and conditional formatting Excel features.

You may restore the default formatting from a database using **Wizards, Table Format Wizard**.

# Conclusion

We have configured the planning application. Our users may work in Microsoft Excel and online.



The screenshot shows the Microsoft Excel interface with a table of financial data. The table has columns for 'code', 'name', 'unit', '2017.11 BUDGET', '2017.11 FORE', '2017.12 BUDGET', and '2017.12 FORE'. The data is organized into two main sections: 'Cost Center 1 (CC1)' and 'Cost Center 2 (CC2)'. Each section includes a summary row and several sub-rows for individual cost centers and their components. The values are in EUR.

code	name	unit	2017.11 BUDGET	2017.11 FORE	2017.12 BUDGET	2017.12 FORE
<b>CC1</b>	<b>Cost Center 1 (CC1)</b>	<b>EUR</b>	<b>1 000</b>	<b>1 100</b>	<b>1 000</b>	<b>1 100</b>
CC11	CC11	EUR	800	900	800	900
CC111	CC111	EUR	200	250	200	250
CC112	CC112	EUR	200	250	200	250
CC113	CC113	EUR	400	400	400	400
CC12	CC12	EUR	200	200	200	200
CC121	CC121	EUR	200	200	200	200
<b>CC2</b>	<b>Cost Center 2 (CC2)</b>	<b>EUR</b>	<b>1 000</b>	<b>1 000</b>	<b>1 000</b>	<b>1 000</b>
CC21	CC21	EUR	800	800	800	800
CC211	CC211	EUR	200	200	200	200
CC212	CC212	EUR	200	200	200	200
CC213	CC213	EUR	400	400	400	400
CC22	CC22	EUR	200	200	200	200
CC221	CC221	EUR	200	200	200	200

We have made the following steps:

8. Configure dimensions.
9. Configure members.
10. Configure forms.
11. Configure permissions.
12. Import budget and actual data.
13. Copy budget and actual data to forecast.
14. Distribute workbooks to users.

We have made these steps in Microsoft Excel. And you may repeat these steps for your case.

You may have great results and satisfied users spending less money and time.

I hope to see you as our valuable customers.

Best regards,

Sergey Vaselenko

## FAQ

---

### **We need more than seven dimensions. How to add a dimension?**

Adding a dimension requires changes to the application database. See a developer guide for this.

### **We need change or add features. How to do this?**

The application has open source codes. So, database developers may do this.

Moreover, you may add a lot of features just configuring the SaveToDB add-in.

Please do not hesitate to ask us about your demands.

### **We have a lot of accounts. Will your application work fast?**

The application uses standard joins with linear relations to calculate aggregates. So, it should work fast.

### **We have a lot of business users. Will your application work fast?**

Business users execute stored procedures to select data on demand and to update the source cube data on cell changes within Excel. These operations are short.

Microsoft SQL Server works with thousands of users. The application should work fast.

### **Can business users work offline?**

Yes. See [Offline User Forms](#).

### **Can we use Azure Active Directory integrated authentication?**

Yes.

### **Can you port your application to MySQL or PostgreSQL?**

This is possible. However, we do not have such plans. Note you may use Microsoft SQL Server Express for free.

### **Can we operate with prices and amounts?**

Yes. Just use the Average calculation type for such accounts.

### **Can we operate with the product of prices and amounts?**

In the current version, no. The application uses the linear model to calculate aggregates.

However, you may add formula lines below the forms in Excel and just copy results into forms.

**Can we operate with conditional formulas like IF THEN ELSE? To calculate taxes, for example.**

In the current version, no. The application uses the linear model to calculate aggregates.

However, you may add formula lines below the forms in Excel and just copy results into forms.

**How to calculate VAT?**

Just set the VAT account as a parent and set the factor value as the tax rate in the usp\_members form.

**How to use different calculation models for budgets and actuals? For example, to calculate VAT.**

You may use three account members like these:

VAT	- set this member as a parent for VAT planned and VAT actual
VAT planned	- set this member as a parent for underlying budget accounts
VAT actual	- import the actual data to this account

**We need to have the same editable account in several forms. How to do this?**

You have two options:

1. Split the account into multiple accounts for different forms and set the master account as a parent.
2. Create different row sets for different forms and link the rows to a single underlying account.

**A business user does not see the required form in the Query List. How to fix this?**

First of all, click the Reload Query List button at the ribbon.

If the required form is absent, set the permission to execute the form in the usp\_form\_permissions form (in the planning-app-admin.xlsx workbook) and then reload the query list.

**A business user does not see the required rows in a form. How to fix this?**

Check and change permissions in the usp\_member\_permissions form (in the planning-app-admin.xlsx workbook).

**Sales managers must have cost accounts from production in the same form. How to do this?**

A typical structure and permissions for this case are:

TOTAL	- no permissions
SALES	- read and write permissions
PRODUCTION	- read permissions
Accounts of sales	- read and write permissions
Accounts of productions	- read permissions

In this case, sales managers may see data from production.

Also, place the Accounts and Entities dimensions in the row axis of the forms.

Otherwise, if the Entities dimension is on the page axis, the user can work with the sales or production data separately, but not in the same screen form.

### **How to change form formatting for all users?**

You may change the default form formatting in Excel.

The conditional formatting rules use values from the following columns:

is\_editable, decimal\_places, is\_percent, row\_color, row\_bold, and row\_indent.

To save formats to a database, use SaveToDB Table Format Wizard.

Users may run the same wizard and load new formats from a database.

See details at <https://www.savetodb.com/savetodb/wizard-table-formats.htm>.

# Appendix A. Installation

---

## Download

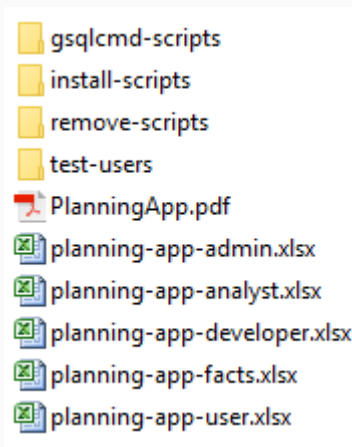
You may download the planning app at <https://www.savetodb.com/download.htm>.

The application also requires SaveToDB Enterprise 7.13 or higher. You may download the add-in here too.

## Download Package

Unzip the downloaded package to a local drive.

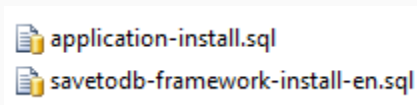
Open the PlanningApp folder. It has the following content:



The folder contains application workbooks, this e-book, and folders with SQL scripts.

## Installation

You may find the installation scripts in the **install-scripts** folder:



To install the application:

1. Create a database like **PlanningApp**.
2. Open and execute (F5) the **savetodb-framework-install-en.sql** script.
3. Open and execute (F5) the **application-install.sql** script.

You may do these steps using SQL Server Management Studio (SSMS).

If you do not have SSMS installed, you may use gsqlcmd scripts discussed below.

## Installing Examples

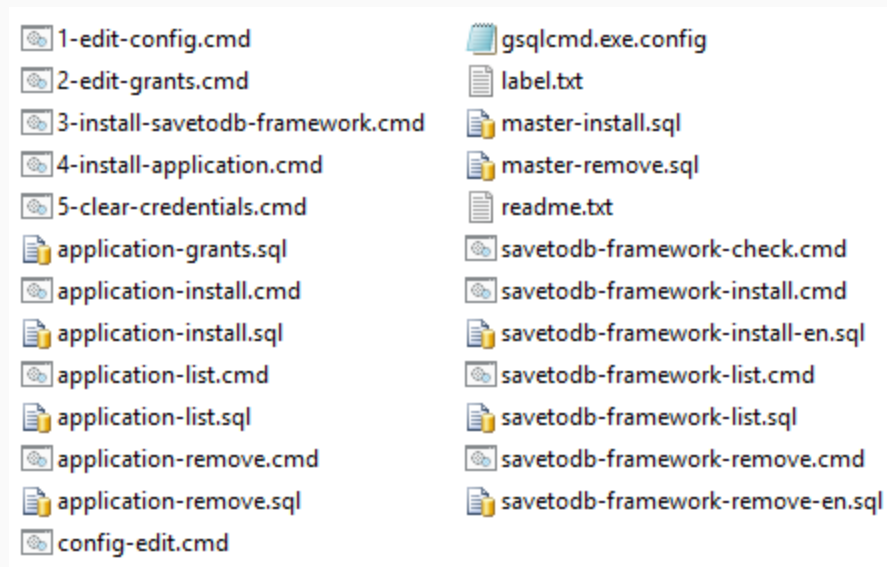
You may install the **Cash Flow** and **Cost Centers** examples using the **Actions** menu in the **MEMBER** worksheet in the **planning-app-analyst.xlsx** workbook.

## Installation using gsqlcmd

You may visit <https://www.gsqlcmd.com>, download and install the **gsqlcmd** utility.

This free utility allows executing SQL scripts using connections configured in a visual mode.

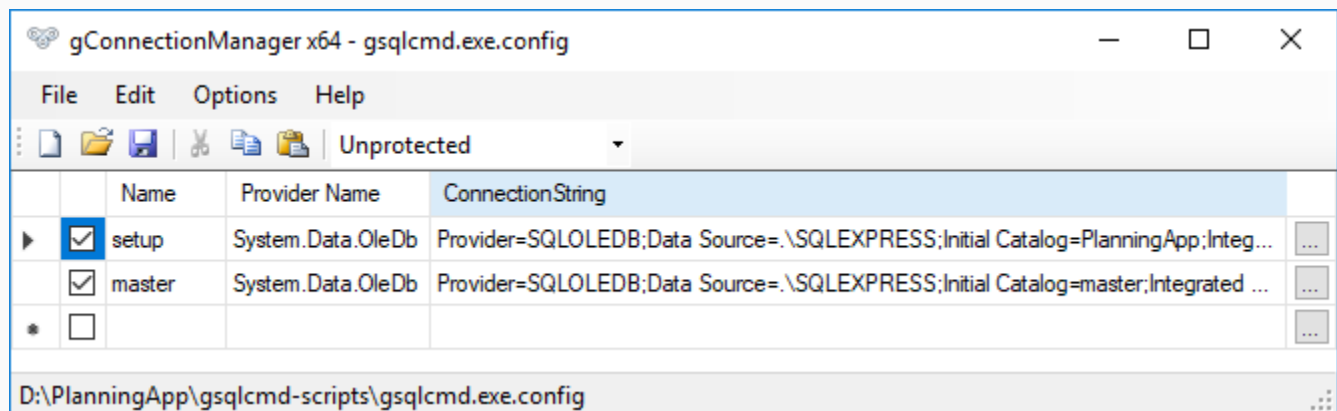
Open the **gsqlcmd-scripts** folder:



And execute the following batch files one-by-one:

1. 1-edit-config.cmd
2. 2-edit-grants.cmd
3. 3-install-savetodb-framework.cmd
4. 4-install-application.cmd
5. 5-clear-credentials.cmd

The **1-edit-config.cmd** open a visual connection string editor, **gConnectionManager**:



Click on the ... button and edit connection strings. Then click the **Save** button and close the editor.

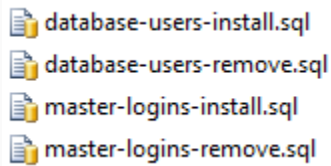
You may use gsqlcmd scripts to uninstall the application also.

Just use **application-remove.cmd** instead of **4-install-application.cmd** and **savetodb-framework-remove.cmd** instead of **3-install-savetodb-framework.cmd**.

## Test Users

If you install the application for test purposes, you may create test users used in the application workbooks.

Open the **test-users** folder:



You have to open and execute **master-logins-install.sql** first against the **master** database.

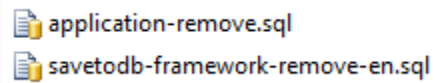
Then open and execute **database-users-install.sql** against the planning application database.

The gsqlcmd scripts discussed above include these operations.

You may use **database-users-remove.sql** and **master-logins-remove.sql** files to remove test users.

## Uninstalling

You may use the following files from the **remove-scripts** folder to uninstall the application:



Open and execute **application-remove.sql** first. Then open and execute **savetodb-framework-remove-en.sql**.

If you do not have SQL Server Management Studio (SSMS) installed, you may use gsqlcmd scripts discussed above.



## Appendix B. Connections

---

The download package includes configured workbooks:

- planning-app-admin.xlsx
- planning-app-analyst.xlsx
- planning-app-developer.xlsx
- planning-app-facts.xlsx
- planning-app-user.xlsx

Workbooks use connections to a local SQL Server Express instance with the test usernames and passwords.

When you open a workbook for the first time, you have to change connections to your database.

To change connections, run **SaveToDB, Wizards, Change Connection Wizard**.

You will see the first screen like this:

**Change Connection Wizard**

**Connect to Database**  
Enter the information required to connect to a database

1. Server name:  [Examples](#)

2. Logon credentials

Windows authentication

User name and password

User name:

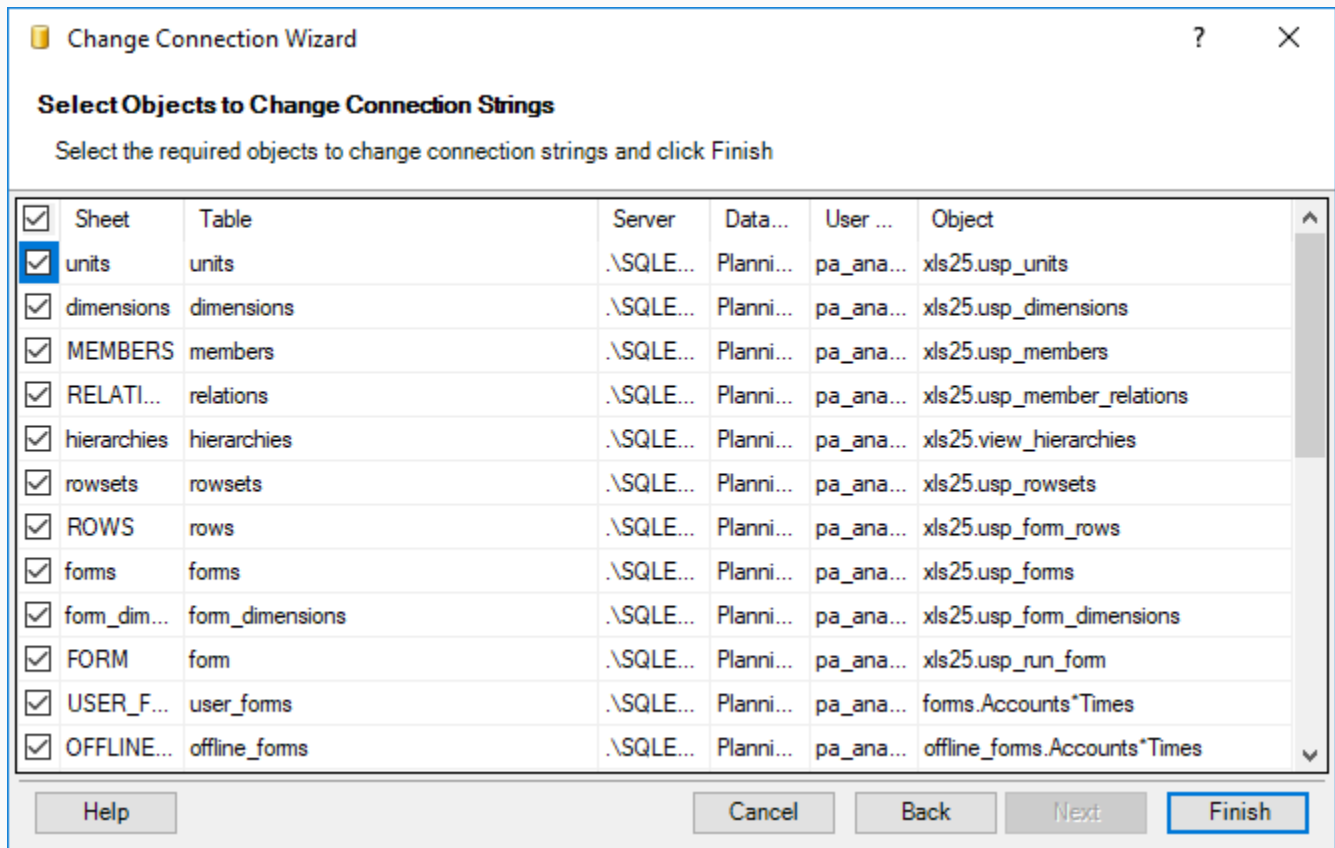
Password:

3.  Database:

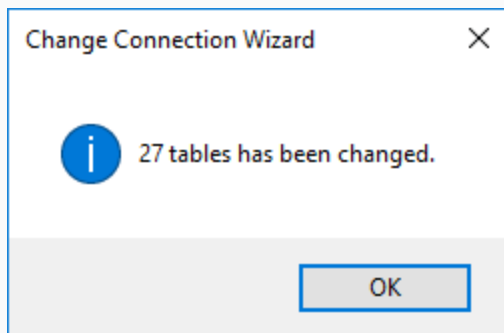
File:

Change the server name and your user credentials. Click **Next**.

In this step, leave all tables checked, and click **Finish**:



You will see the message like this:

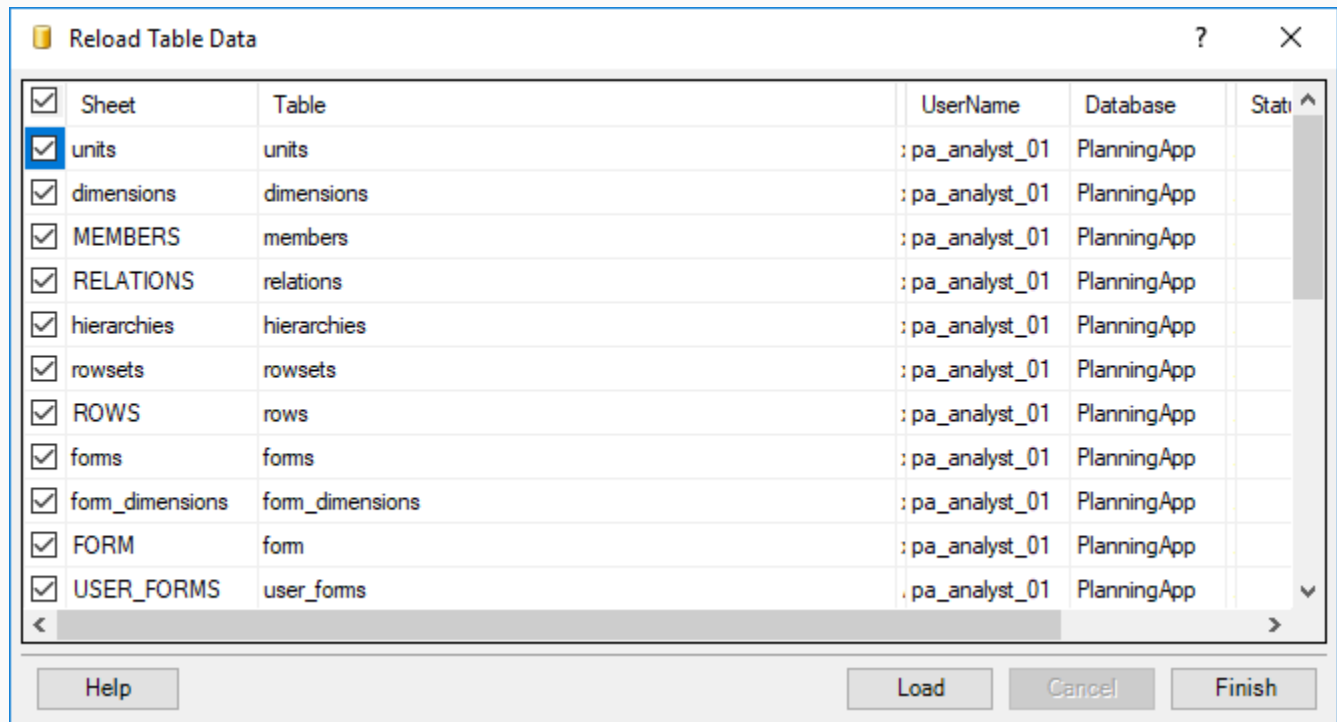


Click **OK**.

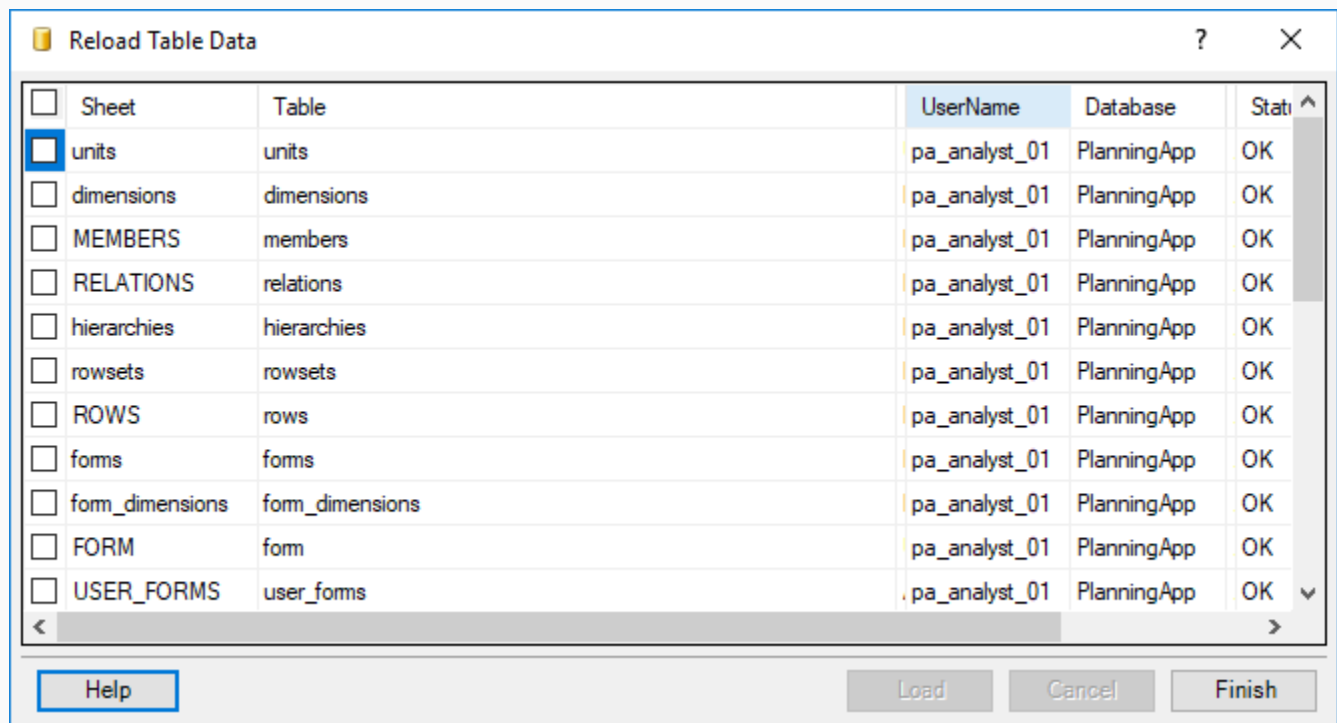
In these steps, we have changed connections.

Now you have to reload all data from your database using new credentials.

Click **SaveToDB**, **Reload**, **Reload All Workbook Tables**. You will see the wizard window like this:



Leave all tables checked, and click **Load**. The wizard will reload all tables and show the reload status:



Click **Finish**.

That's all. You may save the workbook and continue work.

## About the Author

---



My name is Sergey Vaselenko.

I am from Russia, Moscow.

My passion is creating software.

I am a founder and CEO of Gartle Technology Corporation and a leading developer of the SaveToDB add-in.

You are welcome to contact me at

[www.facebook.com/sergey.vaselenko](http://www.facebook.com/sergey.vaselenko)

[www.linkedin.com/in/vaselenko/](http://www.linkedin.com/in/vaselenko/)