TWINNING CONTRACT

BA 15 IPA SR 01 17

Support to the reform of the statistics system in Bosnia and Herzegovina

MISSION REPORT

Activity 2.C.2:
IT application development for producer prices II

Component 2: Business Statistics

Mission carried out by
Søren Netterstrøm, Statistics Denmark

14 - 18 January 2019

Version: Final

Expert contact information

Søren Netterstrøm
Statistics Denmark
Copenhagen, Denmark
Email: sne@viv.dk

Table of contents

List of Abbreviations

BHAS	Agency for Statistics of Bosnia and Herzegovina
BiH	Bosnia and Herzegovina
CBBH	Central Bank of Bosnia and Herzegovina
EC	European Commission
EU	European Union
FBiH	Federation of Bosnia and Herzegovina
FIS	Institute for Statistics of Federation of Bosnia and Herzegovina
MS	EU Member State
RSIS	Institute for Statistics of Republika Srpska
RTA	Resident Twinning Adviser
ToR	Terms of Reference

1. General comments

This mission report was prepared within the EU Twinning Project " Support to the reform of the statistics system in Bosnia and Herzegovina. It was the second mission to be devoted to the development of an IT application for price indexes (CCPI and SPPI) within Component 2: Business Statistics of the project.
The purposes of the mission were:

- Review and further development og the application specification
- Further development of the application

The consultant would like to express his/her thanks to all officials and individuals met for the kind support and valuable information which he/she received during the stay in Bosnia-Herzegovina and which highly facilitated the work of the consultant.

This views and observations stated in this report are those of the consultant and do not necessarily correspond to the views of EU, BHAS, FIS, RSIS, CBBH, Statistics Denmark, INSEE, Statistics Finland and Croatian Bureau of Statistics.

2. Executive Summary.

The overall goal of this component is to develop an application to be used by both CPPI and SPPI as discussed in the first mission of the component.

After the last mission the consultant delivered a test version of the system including some prepared databases. BC has been doing a great job in testing the application and providing very useful feedback. Based on this the application was corrected before the start of this mission. Some minor problems found during the mission has also be corrected.

The purpose of this mission has been to produce documentation for the system, both Users Guide and Developer Guide.

The Users Guide, that take aim at the everyday use of the application was written in cooperation between the consultant and statisticians from FIS and BHAS. It describes the use of the application from the creation of a new survey through the initialization of a new period, data entry and calculation of indexes. It also covers the creation of reports and other administrative tasks.
It should be stressed, that the User Guide should be revised by BC as more experience with the application are gained. Most important in this respect is to improve the guidelines for handling different kinds of no response, both for a full questionnaire and for single items.

The Developers Guide was created in cooperation with IT persons from RSIS, FIS and BHAS. It was agreed that the major task was to provide a detailed description of the database, the purpose and content of each table and where needed the value set for status codes and the like. The other main topic is how to create reports, as it is expected that when a new report is required the assistance of an IT person is likely to be required.
The application code as such is not described. Using the users guide and the original specification to get understanding of the application should make it possible to directly examine the code and locate those places where changes may be needed.

Time was also taken to discuss and demonstrate how data already collected and saved in databases or spreadsheets may be imported to the database.

In the last mission it was recommended that there should be a single person (in one of the institutions) that take over the responsibility of the application, even if the consultant will provide support for the duration of the twinning project. It seems that this issue has not been solved yet.

The consultant also reminds on the recommendation to form a working group responsible for any further development of the application.

3. Next mission

It has been discussed to have a final mission in May. In the view of the consultant such a mission should only be carried out if there will be a need. With the users guide and the latest updates to the system, the next step should be to put it into production. Depending on those issues that may then be raised, a final mission may be necessary, but a decision on this should not be taken before April. It may even be appropriate to hold such a mission at a later time.

However, a 1-day mission will take place 7th of March when the consultant is anyway in Sarajevo.
The consultant will be in Bosnia for some mission related to SBR and may if necessary, find time to discuss issues related to this project. It will also be possible to get support through e-mail.

Annex 1. Terms of Reference

Terms of Reference

EU Twinning Project BA 15 IPA ST 01 17

Component 2: Business Statistics
14 – 18 January 2019
Hosting institution: BHAS, Ferhadija 11, Sarajevo

Activity 2.C.2: IT application development for producer prices II

1. **Purpose of the activity**

- Follow up from the previous mission:

  Conducted by BC:
  - Perform a test of the application to catch any outstanding issues

  Conducted by MS expert:
  - Solve outstanding problems in the software
  - Provide a new version of the software that can be tested by the BC.

- Preparation of a user’s guide
- Preparation of a developer’s guide

2. **Expected output of the activity**

- User’s guide and a developer’s guide are produced

3. **Participants**

Agency for Statistics of BiH (BHAS)

- Fahir Kanlić (CPPI)
- Alen Bajramović (SPPI)
- Ševala Abaz (IT)
- Marko Zekic (IT)

Institute for Statistics of Federation of BiH (FIS)

- Edina Dulic (CPPI)
- Merima Hadzalic (SPPI)
- Ahmet Fazlic (IT)

Institute for Statistics of Republika Srpska (RSIS)

- Pero Kazanović (IT)

MS Experts

- Søren Netterstrøm, Statistics Denmark

Twinning Project Administration

- Katja Møller Hjelvang, RTA
- Đemka Šahinpašić, RTA Assistant
- Haris Imamovic, Interpreter

Annex 2. Users Guide

Users Guide to application for Price Indexes

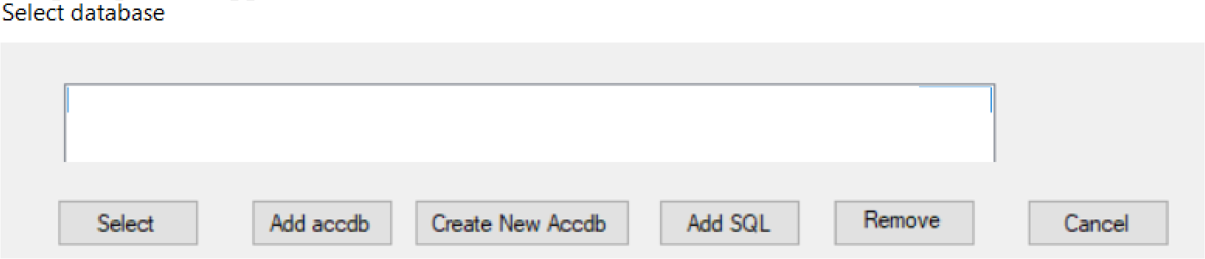
Overview

The application for price indexes is built to cover the needs for Service Price Index and Construction Price Index, but it may eventually be used for similar indexes if needed.
Prices are collected from response units for a set of products. Each response unit may have its own set of products for witch prices are collected at a given point of time.
All products are related to an elementary index.
In the case of the construction index, the elementary indexes are created to be equal to a single product description. An example is a Tube 63 mm where the price is given for 1m. So all response units reporting this product may use the same description of the product and measurement unit.
For the Service Price index, the elementary index description is too broad to specify one product. The index could be Transport by bus, but then the actual product could be 1 trip from Sarajevo to Berlin for one company and 1 trip from Banja Luka to Vienna for another company. The same company may even report more than one price related to the same elementary index.
To handle this, the system for each company keeps a separate list of products, where each product is identified by the ID of company, the code of the elementary index and an automatically assigned subkey.
Prices are collected at regular intervals. For each period the same prices are collected as far as possible. However, a response unit may cease to provide a product or a service or the company may be terminated. In those case new products and new response units may be added to the survey.
When collecting prices, the response units are asked to give both the current price and the price in the previous period, as the main target of the surveys are the development of prices and not the actual prices.
When prices have been collected, they are entered into the system and some error checking is done to identify potential errors, unusual price ratio, outliers and inliers. At the end some manual inspection of the collected prices may be necessary.
Once prices have been entered and tested elementary indexes may be calculated. A report is created and should be inspected before continuing.
Finally, aggregated indexes should be calculated.
The system can produce reports based on the content of the database. New reports can be designed and saved. These reports can then be created whenever needed.

Setting up a new survey.

Create a new database.

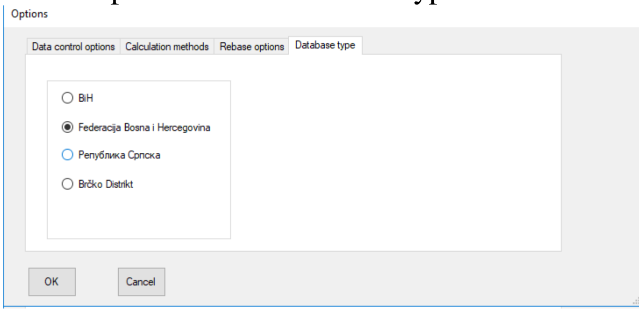
A new database must be created to set up the system.
Simple open the application and select ‘Create New Database’ from



and save the new database when asked to and give it a title (that appears in ¨Select database’)
In this case you will get a Microsoft Access Database.
If you want a SQL server Database, you must first create an empty database (using SQL Server Management). Then Select Add SQL and give name and password as needed.
If you at a later stage want to change or move a database to another location, first use ‘Remove’ to get rid of the old reference and then use ‘Add accdb’ or ‘Add SQL’ to select the new database.
Note that if you Create or Add a database, if the database is empty all necessary tables will be created by the application (you will be asked to confirmed)
You also need to create the base period, that is the period before the first period where data is collected. That is, is data collection starts in 2018Q1, then base period should be 2017Q4. For this period all indexes will be set to 100.

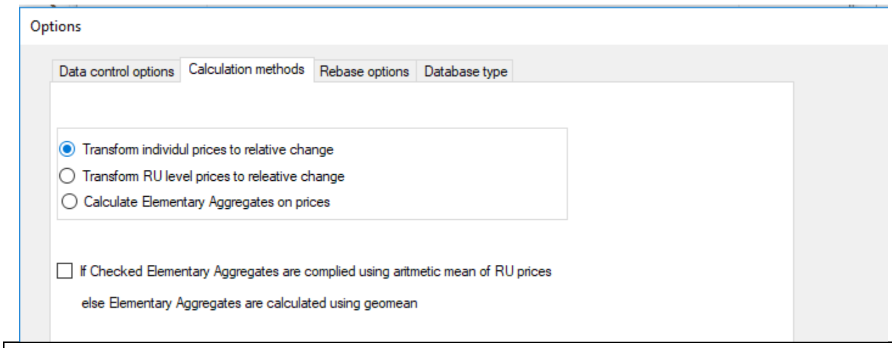
Set Options

The first thing to do is to set the options for the new project.
From main menu select Administration and then select Set Options
Most important is the database type.



○ BiH
● Federacija Bosna i Hercegovina
○ Република Српска
○ Brčko Distrikt

Calculation methods should also be determined. They should be the same for all databases on the same survey (Construction, Services, other) in BHAS, FIS, RS and Brcko.



Options

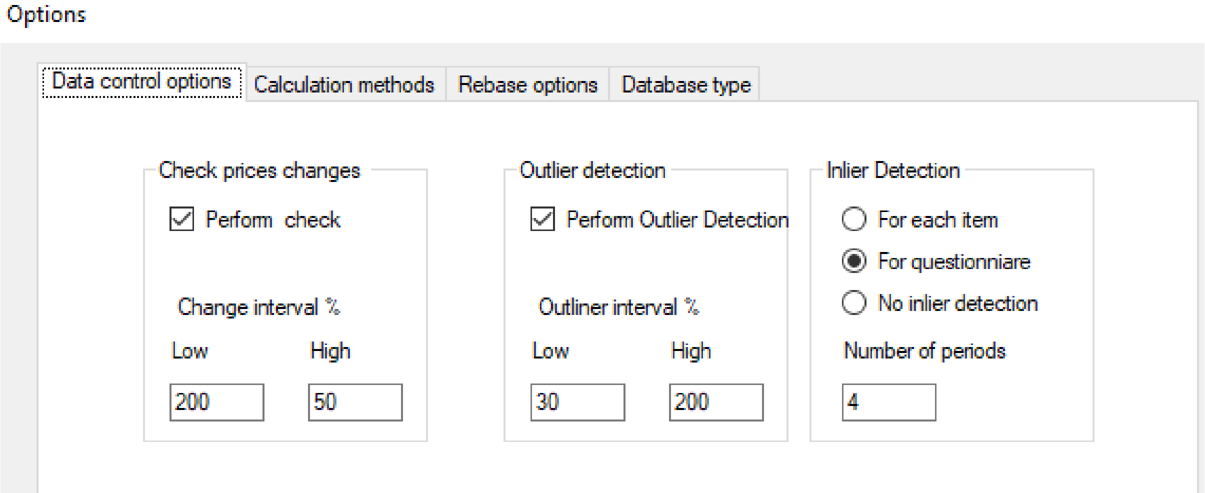
● Transform individul prices to relative change
○ Transform RU level prices to releative change
○ Calculate Elementary Aggregates on prices

□ If Checked Elementary Aggregates are complied using aritmetic mean of RU prices
else Elementary Aggregates are calculated using geomean

Construction Price Index
RU are turned into relative change
Elementary aggregates are compiled as arithmetic mean

Service Price Index
Individual Prices are turned into relative change
This is done because prices even on same product group may be in a very large range

The data control options may be set at this point, but they are more likely to be changed as you get experience with the data for that survey. In the end they should be set in a way where potential errors are captured without having a lot of warning where there is no error.



Options

Check prices changes
☑ Perform check
Change interval %
Low 200
High 50

Outlier detection
☑ Perform Outlier Detection
Outliner interval %
Low 30
High 200

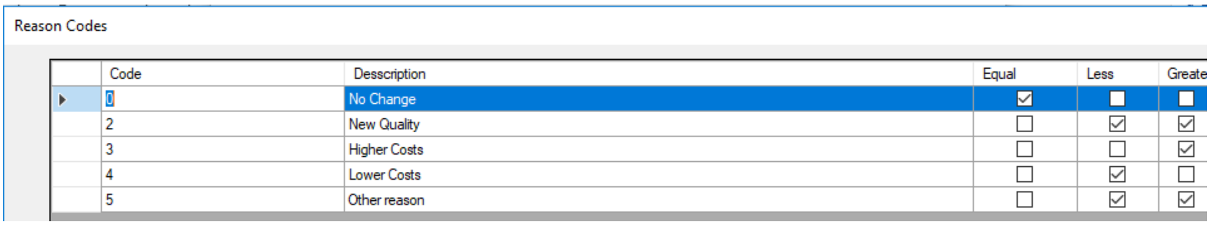
Inlier Detection
○ For each item
● For questionniare
○ No inlier detection
Number of periods 4

Rebase should from the offset be equal to based quarter and may later be changed when needed.

Se appendix 2 for details on the different options.

Make Reason Codes

For each set of previous price / current price there is a reason code for change/no change associated.
Select administration, Reason Codes from the main menu



Reason Codes

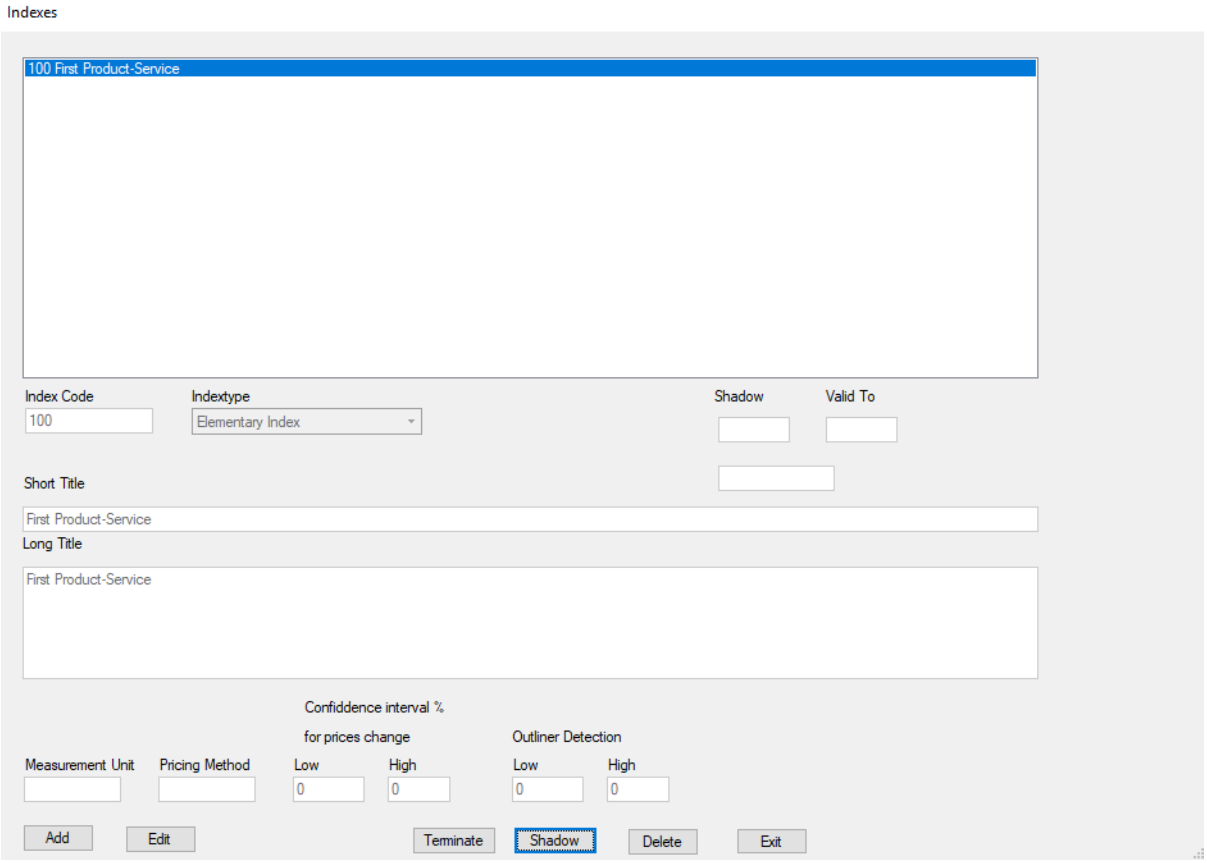
Code	Desscription	Equal	Less	Greate
0	No Change	☑	□	□
2	New Quality	□	☑	☑
3	Higher Costs	□	□	☑
4	Lower Costs	□	☑	□
5	Other reason	□	☑	☑

The code should be a number (0-32000). For each reason code indicate if that reason code is valid with unchanged prices (equal), lower prices (Less) or increased prices (Greater).

The first code (lowest number) must have ‘Equal’ ticked, as it will be the default code when creating prices for a new period.

Set up elementary indexes (product list)

Select Administration, Maintain Indexes from the Main menu



Indexes

100 First Product-Service

Index Code: 100

Indextype: Elementary Index

Shadow:

Valid To:

Short Title: First Product-Service

Long Title: First Product-Service

Confiddence interval % for prices change: Low 0, High 0

Outliner Detection: Low 0, High 0

Measurement Unit:

Pricing Method:

Press add to enter a new index. Give it a code and select elementary index as type.
Add a short title and a long title (may be the same).
Measurement unit and Pricing Method is set if the survey is designed in a way where all response units should use the same measurement unit and pricing method for the same product.

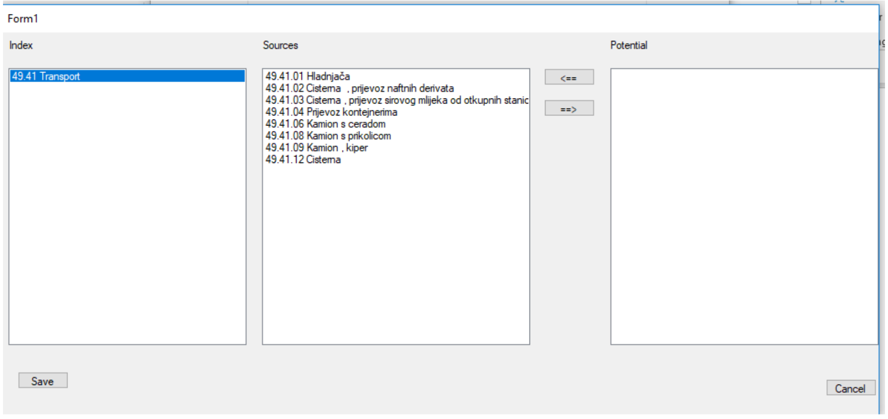
CPPI Measurement unit and pricing method should be set

SPPI Measurement unit and pricing method are not set

Confidence interval and outlier detection is probably not set during initial setup, refer to appendix 2 for further information about these options.
Terminating and shadow units are dealt with during discussion of setting up a new period.
Be aware, that the elementary indexes for a survey should be the same for FIS, RS, Brcko and BHAS. At least the codes must be the same, short and long titles may be different (but should have same meaning).

Create higher-level indexes and create index structure

While adding indexes you may also add the higher-level indexes needed.
Higher level indexes can be added / deleted at any point of time, but normally you will have a structure in place when planning the survey.
First step is to add the higher level indexes (similar to elementary indexes
Next you select index structure from the menu to created the structure.



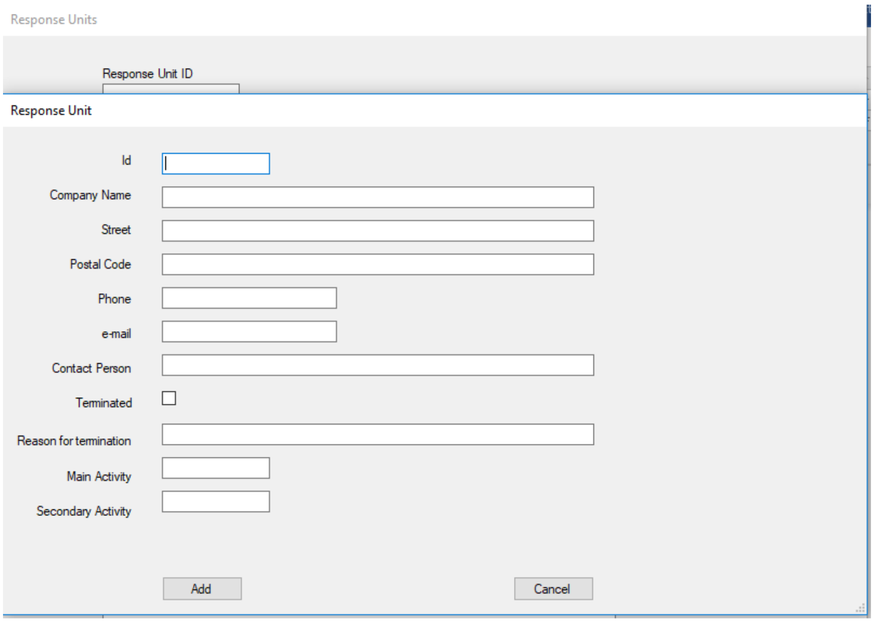
49.41 Transport

49.41.01 Hladnjača
49.41.02 Cisterna , prijevoz naftnih derivata
49.41.03 Cisterna , prijevoz sirovog mlijeka od otkupnih stanic
49.41.04 Prijevoz kontejnerima
49.41.06 Kamion s ceradom
49.41.08 Kamion s prikolicom
49.41.09 Kamion , kiper
49.41.12 Cisterna

Simply select a higher level index from the first box and then add (using the ← button) those elementary index to use to the sources box from the potential box.
When all are done, press Save.

Create List of Response Units

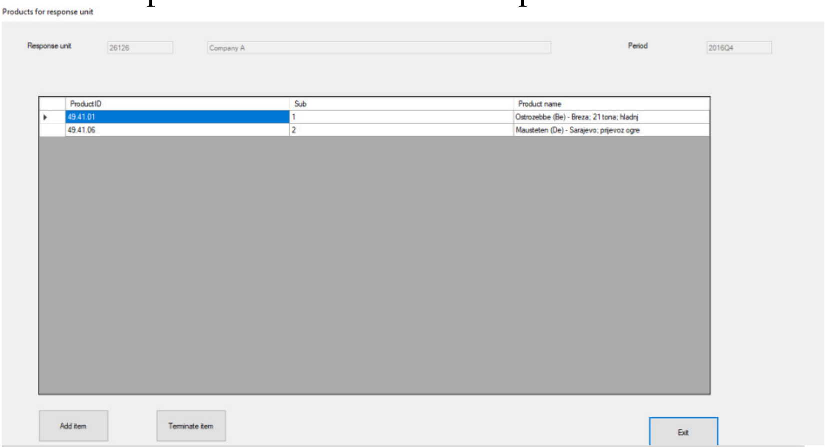
Select Administration, Response Units from the Main menu
From the first screen select add and then fill in the details



In case you have many response units and (most of) the data is available in a spreadsheet, you may ask an IT person to help you transferring the data to the database directly from the spreadsheet. Then used the Edit option to fill in missing details or changed information.

Allocate products to response units

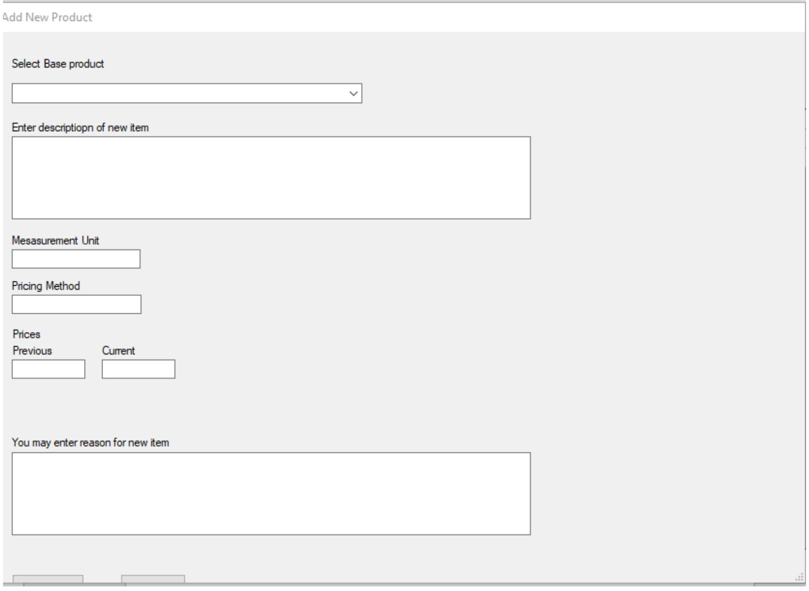
Select Administration, Edit Product List for Response Unit from the Main menu
Select a response unit and then adds the products



Products for response unit

Response unit 26126 Company A Period 2016Q4

	ProductID	Sub	Product name
▸	49.41.01	1	Ostrozebbe (Be) - Breza; 21 tona; hladnj
	49.41.06	2	Mausteten (De) - Sarajevo; prijevoz ogre



Add New Product

Select Base product

Enter descriptiopn of new item

Mesasurement Unit

Pricing Method

Prices
Previous Current

You may enter reason for new item

Select a base product (elementary item level)
You will get the standard text (long text) from the index table and any standard measurement unit and pricing method.
If there is a more detailed agreement with the response unit about the product/service to report, change the description to reflect that agreement.
Set measurement unit and pricing method as appropriate.
Skip prices and reason for new.
Note. You may enter more products based on the same elementary index to one response units. Each will automatically be assigned a subproduct ID.

Reports

Select Administration, Maintain Reports from the Main menu
As a starting point, you may import a set of reports from an existing survey.
This should to include among others standard reports for showing indexes when calculated.
Refer to appendix 3 for more information about the creation and maintenance of reports.
Note that reports may be standard reports, run each time you check data or calculate indexes.
These and other reports may also be requested from the Main Menu, Reports.

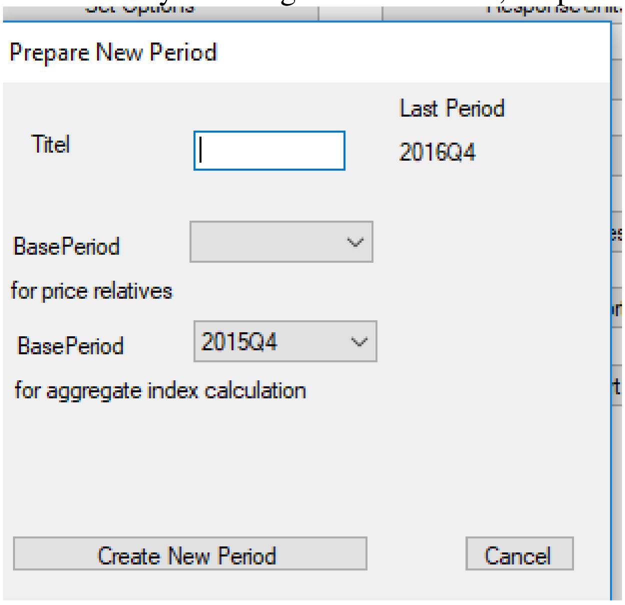
Users

Select Administration, Users from the Main menu
With this option you may limit the number of persons who can access to the database (through the application).
If no users are defined, anybody with access to the application may run the application and make any kind of modification to the database.
However, if just 1 user is defined, you need to be a user in order to open the database through the application. Users are identified by their windows login.
One, several or all users may be administrators, that is they have access to all the functions in the Administration menu. The first user defined will by default be administrator

Creating a new survey period

Add new period

For each survey period you need to start by creating the new period.
You do so by selecting Administration, Prepare New period.



Prepare New Period

Titel

Last Period
2016Q4

BasePeriod
for price relatives

BasePeriod 2015Q4
for aggregate index calculation

The new period should be given a title. Title should be constructed in a way so they can be sorted, that is use '2017Q1' rather than 1 Quarter 2017'
The base period is the period used as a base when calculating price relatives (development of an individual price). Can be any previous period. However, if a period other than the previous period is selected, it is not possible to add new products to a response unit as there will not be a reference (previous) price in the base period.
The base period for aggregate indexes may be different. For that period there must be a corresponding set of index weights.
When preparing a new period, the system will create for each response unit a list of the products for that period by taking all products from the previous period that is not terminated. Reason code is set to not changed and previous and current price is copied from current price of previous period.
You may edit this (see below).

Update response units as needed

Make sure that response units are updated, most important e-mail address and address.
This is normally done during the processing of questionnaires as the information is obtained.
You may also add new response units. Whenever a new response unit is created, a questionnaire for the current period (with all prices set to 0) will be created as well.

Update list of products for response units as needed

The list of products for one or more response units may have to be updated using information received during the processing of the previous period. This has to be done at this stage.

Use database to produce individual questionnaires.

The database at this point should hold all the information needed to produce individual questionnaires (as Excel files) for each response units listing all the specific products for witch a response in anticipated. Such files could then be distributed by e-mail to the response units.
The benefit of this method would be, that it reminds the response unit of the specific product asked for (and instruct then to specify if a replacement has been made).

Index Weights

Index weights for elementary updates may be updated as needed.

CPPI Index weights are fixed (based on the standard building)

SPPI Index weights are updated when new data for turnover is available (yearly)

Product weights

By default, when preparing a new period, the product weights from previous period is copied to this period.

Product weights may be updated at this point, if products are added or removed from one or more response units

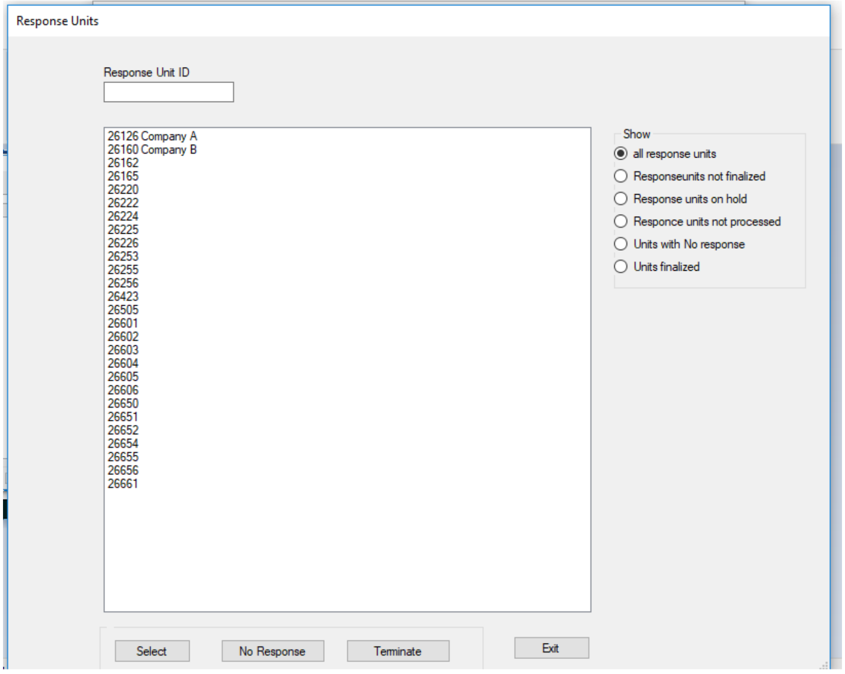
If new data on turnover is available, products weights **must** be updated using this new data. Note that they may be changed as needed if during data entry products are added or removed from response units. The rule that all or no response units has weights for a specific product is checked before calculation and may require update of these weights before continuing-

Entering data from survey

Standard

From the menu select Data Entry.

A list of all response units that has not been processed will be shown.



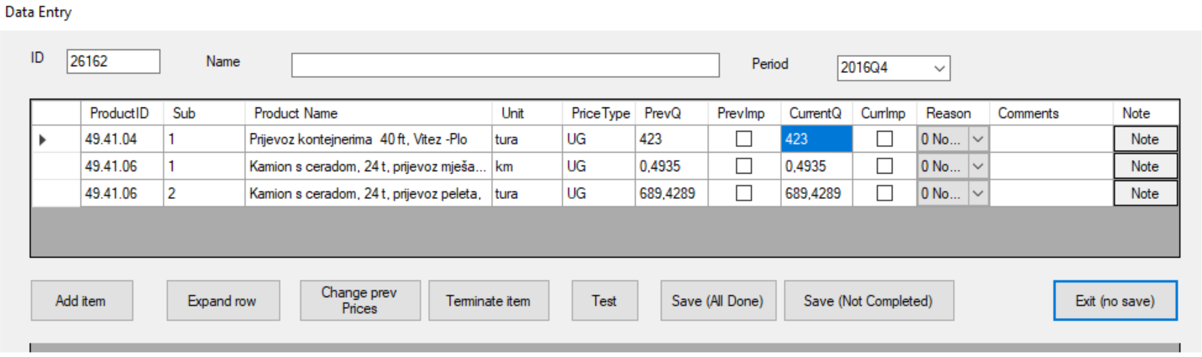
Response Unit ID

26126 Company A
26160 Company B
26162
26165
26220
26222
26224
26225
26226
26253
26255
26256
26423
26505
26601
26602
26603
26604
26605
26606
26650
26651
26652
26654
26655
26656
26661

Show
● all response units
○ Responseunits not finalized
○ Response units on hold
○ Responce units not processed
○ Units with No response
○ Units finalized

Note that you may select to show all response units or response units with a certain status for the current period.

Select the response unit you want to process.



ID 26162 Name Period 2016Q4

	ProductID	Sub	Product Name	Unit	PriceType	PrevQ	PrevImp	CurrentQ	CurrImp	Reason	Comments	Note
▶	49.41.04	1	Prijevoz kontejnerima 40 ft, Vitez -Plo	tura	UG	423	□	423	□	0 No...		Note
	49.41.06	1	Kamion s ceradom, 24 t, prijevoz mješa...	km	UG	0,4935	□	0,4935	□	0 No...		Note
	49.41.06	2	Kamion s ceradom, 24 t, prijevoz peleta,	tura	UG	689,4289	□	689,4289	□	0 No...		Note

You can now enter any new prices. If this is the first period for a response unit, you must enter both previous and current prices, else you are not allowed to change previous price.

If the price is changed, you may need to enter a reason code corresponding to the change as well.

You should verify, that the reported price for previous quarter is equal to the price listed and that the product name has not been changed. See also handle change in quality below.

Before you save, you may run Test to check for outliers or inliers.

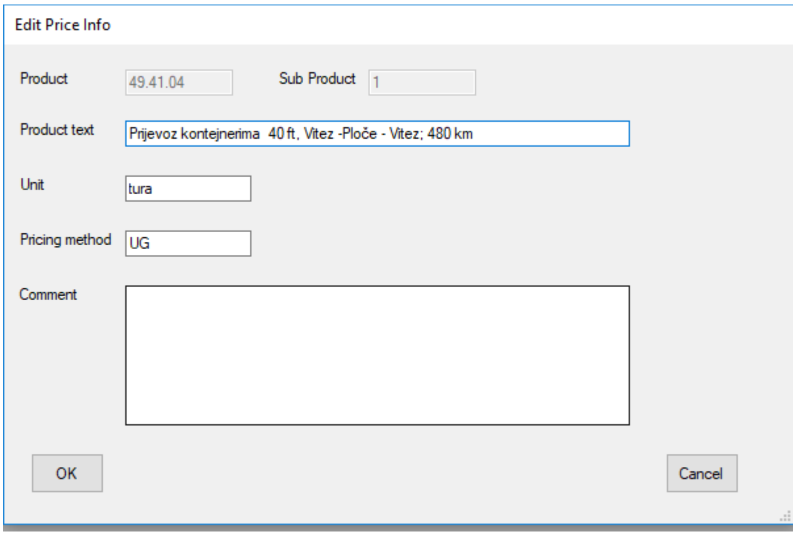
When all is done select Save (all Done).

Change in products (new product, quality change, missing product)

In some cases, the simple procedure above cannot be used because a product has been replaced by a new product (or a change in quality), a new product is added to the questionnaire or no response.

If there is a change for a product in the description it must be considered if this is due to a change in quality of the product or just a simple correction (precision) in the text. You may examine the reason code and previous price in detecting what is the reason for the change.

If it is decided, that this is only a change of the text, then use Expand Row to be able to edit the text.



Edit Price Info

Product: 49.41.04

Sub Product: 1

Product text: Prijevoz kontejnerima 40 ft, Vitez -Ploče - Vitez; 480 km

Unit: tura

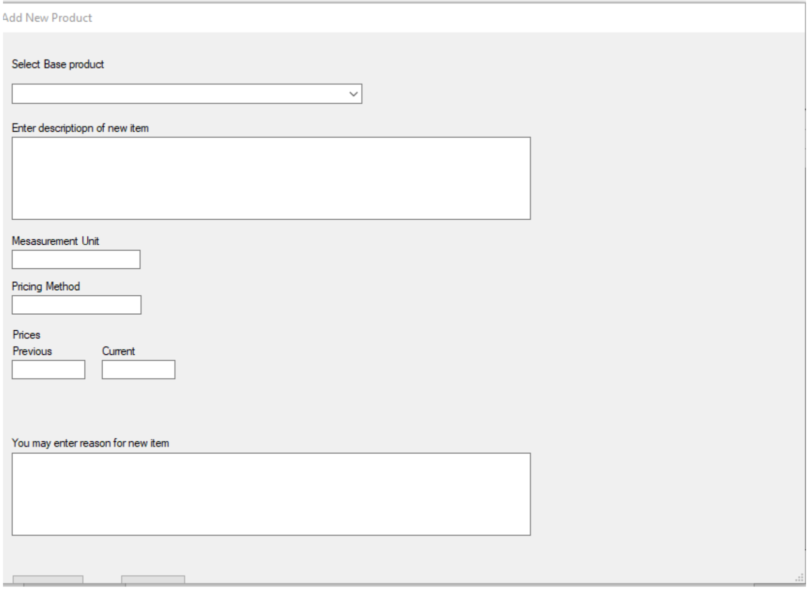
Pricing method: UG

Comment:

Be aware, that when doing so the new text will also be applied to all previously reported prices from the same company with the same product ID and Sub product ID.

If this is a change in quality, then it is needed to start a new price (time) series (as the prices for the new product quality cannot be compared to the prices of the old product).

To do so, terminate the old product, then add a new product.



Now select base product, enter description text, measurement unit, pricing method as well as previous and current price. You may also enter a comment giving the reason for adding the product. Then save the new product line.

Note: For SPPI where base period is last quarter of previous year, you can only change in the first quarter of a year. For CPPI the situation is not clear at this moment.

In the case that the response units add a new product (start reporting on a product), add new item should be used in a similar way.

In the case a reporting unit is not reporting on one (or more) products, it must be determined if it is just a temporary stop (seasonal or other) or if the product will not be reported in the future.

If temporary, an imputed price may be entered (and marked as imputed). When reporting continues, it may be useful to verify (and change) the imputed prices. Use Change previous prices to see and potentially change previous prices. You may also inspect old prices by using the period box on the top of the screen, but you can not this way change data for previous periods.

If permanent, the item should be terminated.

No response

If a unit does not respond at all, the first step is to mark it as No Response (on the from listing the response units).

It should no be examined if this is a temporary stop or if the company has terminated.

Select to see No response units on the list of Response Units. In temporary, you may impute all prices (like handling as single missing product). If permanent, the response unit should be terminated. This may be done from current quarter or even back in time.

Dealing with inliers and other potential errors

When entering data, test should be used to check for inliers, outliers and confidence intervals. These checks will be carried out as specified in Options using default limits from options or limits for the specific type of product (elementary item).

Inliers is a situation, where a response unit for several periods has reported the same prices for one or all products. The recommend setting for BiH is to perform this test on the full questionnaire. If the system comes with the warning that the questionnaire is an inlier, the response unit should be contacted to clarify the situation. If it turns out, that the response unit has in fact just copied the first questionnaire and now are willing to report, prices for previous period may be imputed. If it is not possible to get any meaningful response, it should be considered to terminate the response unit (even back in time).

Outliers is a situation where a price is outside a given interval compared to the average price for the product for all units reporting in the previous quarter. For SPPI this test does not makes sense, as services may be very different within the same product group. But for CPPI this test may be useful. The most likely reason for an outlier may be, that the price is not given for the requested measurement unit, i.e. the price is given for 1 ton rather than for 1 kg.

The last check is for unusual changes relative to previous period. If the current price is below a certain percentage of previous price or higher than a certain percentage this change should be verified, potentially by contacting the company. The reason may be, that there is an unreported change in quality, see above how to handle changes in quality.

Save

Before saving you may enter a comment for one or more items that may be useful for future processing.

When the questionnaire has been processed (and tested if any changes), you may press Save (All Done) to mark the questionnaire as finally processed. If the reason code does not match (no)change of price, you will get an error message. Correct and press Save again.

If you come across problems that cannot be immediately solved, you may use Save (Not Finalized). Use Note button to add a note to the questionnaire stating the reason that it has not been possible to complete the processing. A questionnaire having notes cannot be finalized until the note is removed.

You may then when the necessary information has been retrieved return to finalize the questionnaire.

Test data

When all questionnaires have been processed, the next step is to test the data.
First it is tested, that all questionnaires have been processed (or terminated).
The next step is to check, that if there is a weight assigned to at least 1 response units for a product, then there must be a weight for each response unit reporting on the product and only for those response units. The sum of the of weights must be 1.
In case of error product weights should be revised.
Finally, it is checked that each price has a corresponding base price (depending on the setting of price base period).
It should be noted, that the test will be performed for all periods where any data has been changed.

Calculations

Base indexes

Next step when data has been verified is to calculate the base indexes.
Indexes are always calculated for all periods to ensure that any changes to base data is used.
A report will be produced with all indexes. This should be examined for unusual developments. In case of problems, it may be needed to further revise the input data. You may use the report feature (see below) to list individual process for products to detect the cause of the problems encountered.
Another report is created listing the number of prices use for the calculation of each index.
In case errors are detected, go back to data entry to correct those errors.

Shadow indexes

In the case, that there a no (or few) prices reported for a product / elementary index, a shadow index may/must be created.
A shadow index is calculated by using the development of another index disregarding any reported prices.
Shadow indexes are created using Administration, Maintain Index. You set the index to follow and the starting point.

Aggregate indexes

Once elementary indexes have been calculated and verified, the Aggregate indexes can be compiled.
A report is created with all indexes and another may be created with the indexes rebased (if this option is set).

Transmit data to BHAS

Once indexes have been calculated and verified, the entities should transmit data to BHAS. Select Administration, Import/Export to initiate this transfer. This will create an Excel workbook that should be saved and then send to BHAS. Give the workbook a proper name indicating the entity and period.

BHAS will then use Select Administration, Import/Export to retrieve the data for processing at BiH level.

Reports

Using reports

The system should contain some reports. A set of standard reports are delivered with the system and should be installed as part of setting up the system. This include those reports automatically produced when testing and calculating.

Other reports may be used for instance to list all prices reported for a single product or all prices reported by a single response unit.

To run a report, select Reports and then select the report. The report may need information about product, enterprise or period. In that case message boxes are displayed where you enter the data (from a dropdown list).

Reports are created as Excel workbooks.

Adding reports

Appendix 3 deals with this in detail.

A report is writing as an SQL-query and you may need the assistance of an IT person to create a new report.

When making a report think about how the report using parameters may be reusable, that is don’t ask for a report on a specific product, but create the report in a way where product is a parameter allowing to run the same report on any product.

Appendix 1 Calculation methods

Two different methods of calculation are used in the application.

Elementary indexes

The calculation of Elementary indexes is performed this way:

For the first period in the system, that will be the base period until there is a new version of the index structure, for each elementary item the index is set to 100.

When there is a change in index structure, all base indexes are rebased to 100 and any new indexes are added with the value of 100.

For other periods the following steps are taken using chained indexes (for each period):

1. Individual prices are summarized to prices on response unit level for each product using unweighted arithmetic mean. Both current at previous price is computed this way.
   The formula used is
   `Exp(Sum(Log(Price)) / Sum(number of prices)) for each product/response unit.`

2. Prices (previous and current) on response unit level are now turned into Elementary Aggregates for each product.
   If the product has product weights a weighted arithmetic mean is used, for those where there a not any weighted the unweigthed method is used (as above)
   The formula for weighted arithmetic mean used is
   `Exp(Sum(Log([Price^Weight))), for each product. Sum of weigths are 1.`

3. In the next step, the ration between Previous. Price and Current Price for each product is calculated as `(CurrPrice/PrevPrice)`

4. Finally the index is calculated as the index of previous period * the ratio calculated in 3.

Aggregated indexes

Aggregated indexes are calculated using Laspyere indexes.

The indexes may be defined as aggregate of other aggregates, but before calculation this is changed by substitution to ensure that all aggregates are based on a set of elementary indexes.

By doing so indexes can be calculated in any order.

The actual calculation is performed in two steps.

1. Calculate (using Laspiere) the change in the index relative to the base year for calculation (weights).
   Sum([Index for period]/[index for base year]*[index weight]) /sum(weights) for all elementary indexes making the aggregate
2. Multiply the index for base period with the change to get the new index.

Appendix 2 Options

Data control:

Change of price should not be outside a given interval (in %) (by product) Tech note: In the table of products, the allowed interval of changes is given for each product. However, if no interval is given a standard interval (set in options will be used).

Outliers are detected by comparing current price with average for previous period

Tech note: In the table of products, the allowed interval in % is given for each product. However, if no interval is given a standard interval (set in options) will be used.

SPPI only: As prices for a product may be in a very large range, the concept of average price does not make sense, so outlier detection should be turned off.

Inliers mean that data has not changed for a given period.

Tech note: The number of periods to look at are set in options.

The check can either be performed on a sub product by sub product base or the whole of the questionnaire. In the latter case this means, that there are no changes to sub products or any prices in the selected period for the response unit. Method is determined in options.

Calculation methods:

The first steps of calculation (as outlined in appendix 1) may be done on actual prices or on prices relatives.

Use can select to turn individual prices into relatives or you may turn RU prices (price for one product at response unit level) into relatives or use actual prices through the whole calculation process.

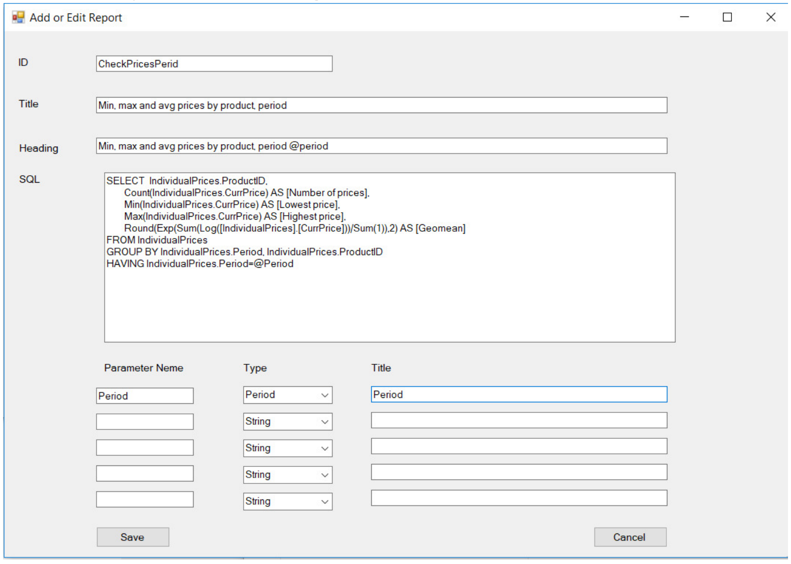
Normally, the Elementary indexed is calculated using (weighted) arithmetic mean, but this may be changed to standard mean.

Appendix 3 Creating Reports

Reports can be created (or edited) by the user.

A report is just an SQL-query, that delivers rows of data that are the inserted in a spreadsheet, with appropriate title and headings.

The user may of cause make such a query using the data model as described in appendix 5. It may however be better to use a tool like Microsoft Access to build the query and then copy (an modify) the resulting SQL statement as needed.



ID: CheckPricesPerid

Title: Min, max and avg prices by product, period

Heading: Min, max and avg prices by product, period @period

SQL:

```
SELECT  IndividualPrices.ProductID,
      Count(IndividualPrices.CurrPrice) AS [Number of prices],
      Min(IndividualPrices.CurrPrice) AS [Lowest price],
      Max(IndividualPrices.CurrPrice) AS [Highest price],
      Round(Exp(Sum(Log([IndividualPrices].[CurrPrice]))/Sum(1)),2) AS [Geomean]
FROM IndividualPrices
GROUP BY IndividualPrices.Period, IndividualPrices.ProductID
HAVING IndividualPrices.Period=@Period
```

Parameter Neme	Type	Title
Period	Period	Period
	String	
	String	
	String	
	String	

The **ID** can be anything but must be unique. It is used as the sort order when showing the list of reports.

The **Title** is what is listed

The **Heading** will be written in the first row of the report. It may include parameters, see below. If omitted Title is used

SQL is the actual SQL statement. It may include parameters

1 to 5 **parameters** may be defined. Each parameter has a name, a type and a title

Name is used to identify the parameter. Before using the heading or SQL query, the system will replace @Name with the value given for that parameter.

Type identifies how the value should be obtained. Basically, there are strings and integers. You may check with the data model or with the SQL statement created using Access.

Beside string and integers some common types are defined: Period(ID), Response Unit(ID) and Product(ID), Index(ID). For these types, the system will use a drop-down combo box listing Periods, Response Units, Product or Indexes respectively, so the user will have to select from that list.

See the example above where period is of type period and the inserted both in heading and SQL

A report may have a mark telling that it should be run when new elementary/aggregate indexes are calculated. Such reports may have only 1 parameter of type Period or no parameters.

Annex 3. Developers Guide

Price Index Application, developers guide

The database

Base tables

Periods

Data is collected for a range of periods (each Month, Quarter or Year).
For each Period of collection, the period is given a name
As the Period is processed, Status is updated (Period created and questionnaires defined, data entry in progress, data entry finalized and data has been checked, indexes has been created)
A period is locked when a new period is started.
There are some references to other periods (base price and base period) as needed during the creation of indexes.

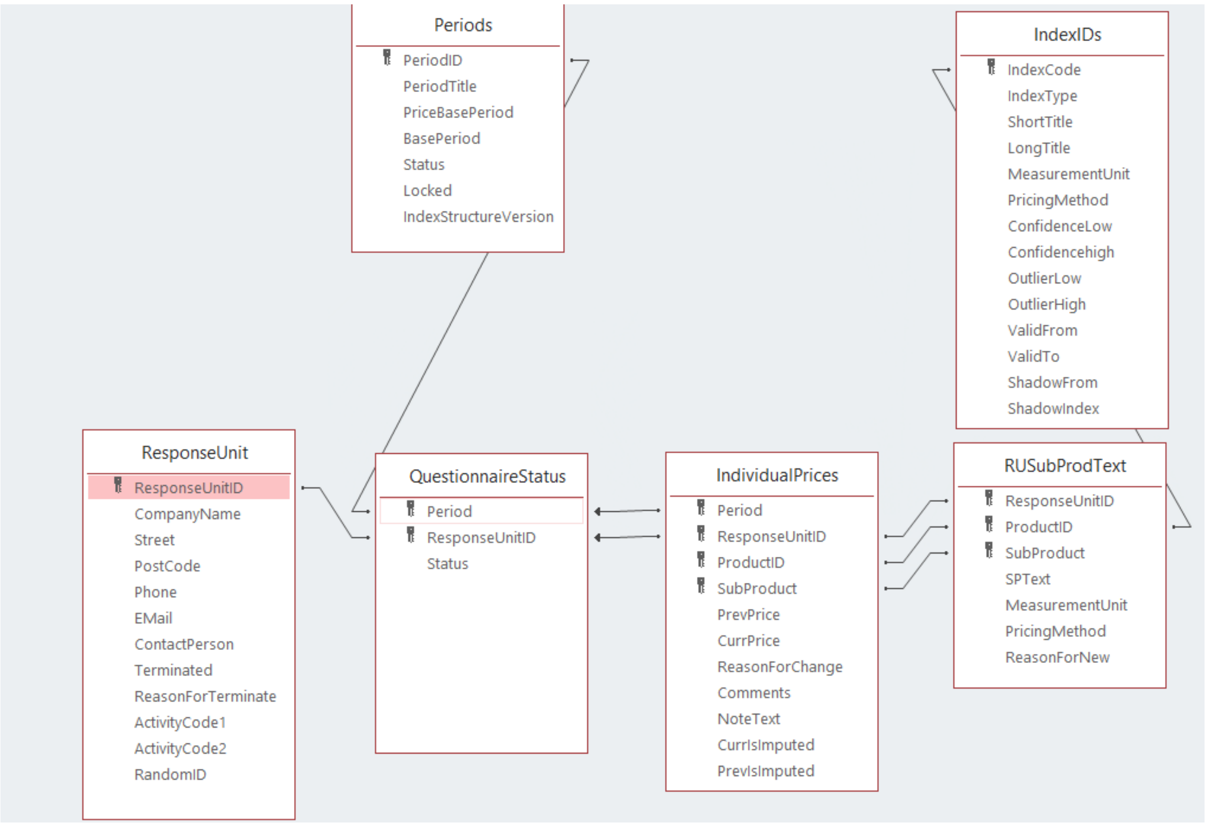
PeriodID*	Interger	0,1,2 etc.
PeriodTitle	Text	Should be sortable like 2017Q4
PriceBasePeriod	Integer	Id of period used as base for price development (elemtary indexes)
BasePeriod	Integer	Id of period used for Laspyere indexes, must have corresponding Baseweigths
Status	Integer	0 Prepared, data entry in progress, 1 data entry completed and verified, 2 elementary indexes calculated, 3 all index calculated
Locked	Boolean	If true, data entry is not allowed

* Indicates primary key

IndexIDs
The data collected is used to produce indexes. Each index has a unique key (IndexCode).
An index can be an elementary index or aggregated (indextype)
Prices are collected for the elementary indexes, **where IndexCode is the same as the ProductID.**
Aggregated indexes are simply compiled from the elementary (or basis) indexes.
Each index has a name that in some cases may be used as product name, but in general, for collection each product normally needs to be specified in more detail and in relation to the individual response unit (se RuSubProdTRext below).

IndexCode*	text	Identifier of Index, from classification
IndexType	Integer	0 for elementary indexes 1 for aggregated level indexes
ShortTitle	Text	
LongTitle	Text	
MeasurementUnit	Text	If present, default, see also RUSubProdText
PricingMethod	Text	
ConfidenceLow	Integer	Overrides default values in Options if there
Confidencehigh	Integer	
OutlierLow	Integer	
OutlierHigh	Integer	
ValidTo	Integer	Id of last Period where index is used
ShadowFrom	Integer	If Shadow index, first Period used
ShadowIndex	Text	IndexCode of index to follow

Basic data related to questionnaires.



ResponseUnit

This table represent a Response Unit
The key is ResponseUnitID
Company Name is used when selecting a responseunit (together with ID).
Other information is just for information and not used in an active way.

ResponseUnitID	Text	ID of Response unit
Company Name	Text	Use for visual identification (Not required)
..	Text	Other fields are not required
RandomID	Text	Created for exchange of data to BHAS Automatically created first time data regarding the response unit is exchanged

QuestionnaireStatus

Represents a Questionnaire for a period

PeriodID*	Integer	Id of Period
ResponseUnitID*	Text	ID of Response unit
Status	Text	0 Not processed, 1 Partly processs (held), 2 Processed, 3 No Response, 4 No response handled (imputation) 5 Terminated

IndividualPrices

For each line in a questionnaire the is an Individual Price

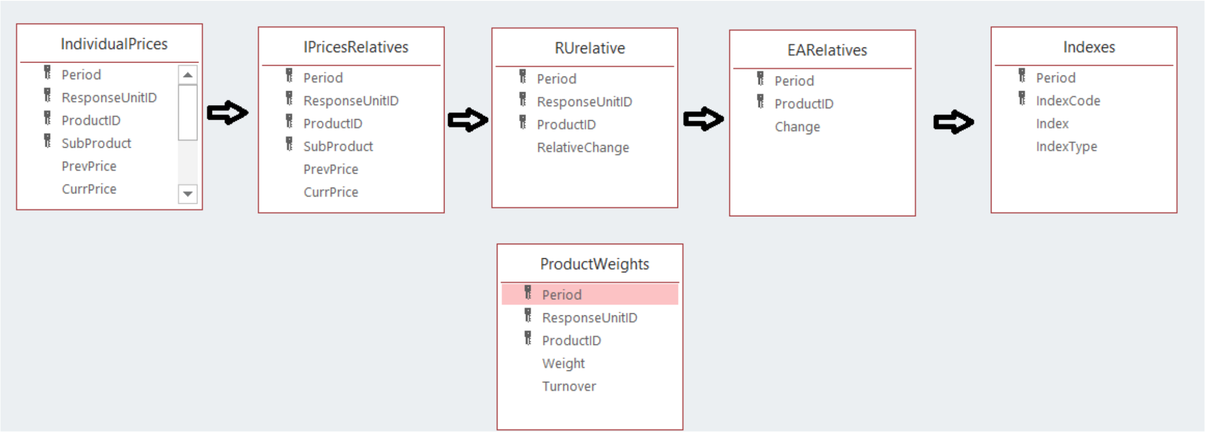
PeriodID*	Integer	Id of Period
ResponseUnitID*	Text	ID of Response unit
ProductID*	Text	= an IndexID of an elementary index
SubProduct*	Interger	Automatically generated number (1,2,3)
PrevPrice	Double	Price in previous period. Needed for first period but kept in all
CurrPrice	Double	Price in current period
ReasonForChange	Integer	User defined classification (see **ReasonForChange)**
Comments	Text	Optional comment
NoteText	Text	Optional text related to a held questionnaire Must be delete to finally save a questionnaire
CurrIsImputed	Boolean	Used to mark prices as imputed, for information only
PrevIsImputed	Boolean	

RUSubProdText

Each response unit are reporting a specific set of products. They may report for 1 or more products related to one ProductID (elementary index) –
ResponseUnitID ProductID and SubProduct identifies one product being reported by a company (normally for many periods
Subproduct is automatically generated when a new product is added to a company as the next number available for the combination of ResponseUnitID and ProductID-

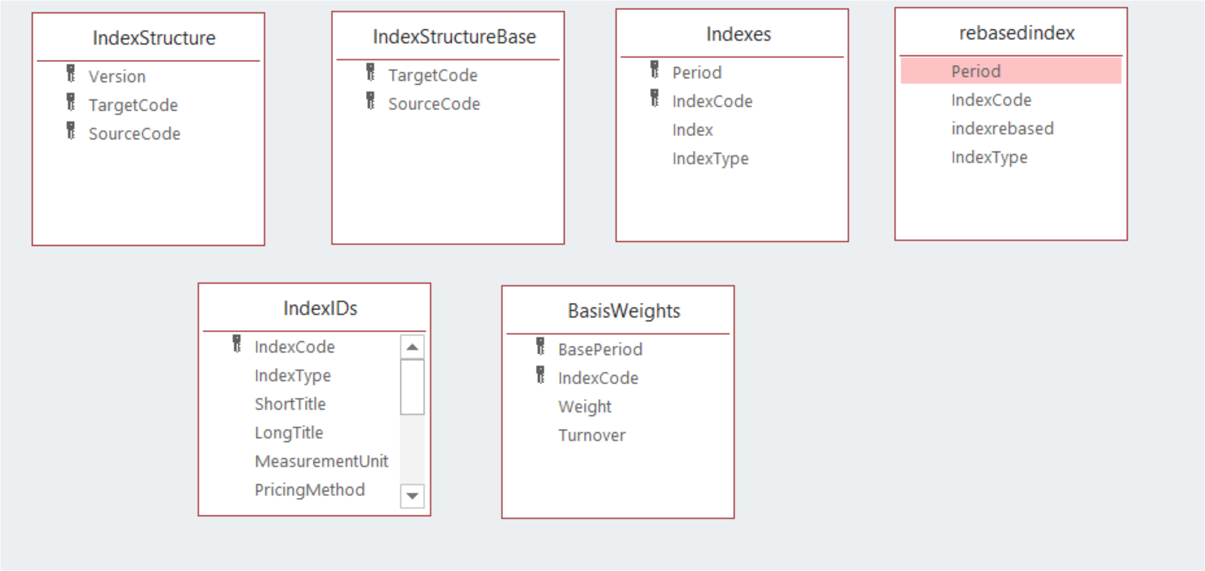
ResponseUnitID*	Text	ID of Response unit
ProductID*	Text	= an IndexID of an elementary index
SubProduct*	Interger	Automatically generated number (1,2,3)
SPText	Text	The description of the specific (sub)product To make sure the RU are reporting exact the same product or service in all periods
MeasurementUnit	Text	To allow different measurement units and pricing methods for same ProductID (needed in SPPI)
PricingMethod	Text	
ReasonForNew	Text	For information only

Tables related to production of elementary indexes.



Indexes are compiled in a set of steps
First, the ration between current price and a base price (from a previous period) is created for each subproduct collected.
Saved in **IPricesRelatives (key Period, ResponseUnitID, ProductID, SubProduct)**
Next prices are calculated on the level of responseUnit and ProductID (using geomean).
Saved in RURelatives (key Period, ResponseUnitID, ProductID)
Now using weighted geomean, **EARelatives** are created, for each Product (Elementary Aggregate). Weights are taken from **ProductWeights** if used. (**key Period, ProductID)**
Finally, the Basic **Indexes** are created from the **EARelatives** as chained indexes
In some cases, index is calculated on prices during part of the process (rather than relatives). Then **RUPrices** and **EAprices** are replacing **RURelative** and **EARelatives.**

Calculation of Aggregated indexes



Aggregated indexes are created from the base indexes as weighted Laspyere indexes
The tables **indexStructure** tells for each aggregated index (Target) which indexes to include in the calculation. However, this may be another aggregated index (in a hierarchy). There may be different versions of the structure (Periods contain reference to version)
The first step of calculation is to create **IndexStructureBase** where by replacing any aggregated source with its sources (in an iterative process) so all sources are base indexes.

Now for each target in **IndexStructureBase** calculation is done taken Weights from **BasisWeights.**
Result as saved in **Indexes**
Index is then rebased and the result is stored in **Rebasedindex. BaseperiodIndex** is used in this process.

Other tables

Options
Just one row holding the options as set by the user (Method of calculations etc.)
ReasonForChange
Codes and names for ReasonForChange and rules for using (equal, higher, lower)
Report Definitions
Defined reports
Users
If used, only Users in this table can access the system. Can also set status as administrator.
Version
A single row giving the actual version of the database (structure). Whenever the DB is opened from the application, this is used to determine if the database should be upgraded.

Creating Reports

The application allows the user to define reports. A report is created as an SQL statement, either a simple Select statement or a pivot table statement.
It is most likely, when a new report is needed, the statistician will ask an IT-person for help, as they may not be familiar with SQL.

A typical report i

Price Index Application, developers guide

The database

Base tables

Periods

Data is collected for a range of periods (each Month, Quarter or Year).

For each Period of collection, the period is given a name

As the Period is processed, Status is updated (Period created and questionnaires defined, data entry in progress, data entry finalized and data has been checked, indexes has been created)

A period is locked when a new period is started.

There are some references to other periods (base price and base period) as needed during the creation of indexes.

PeriodID*	Interger	0,1,2 etc.
PeriodTitle	Text	Should be sortable like 2017Q4
PriceBasePeriod	Integer	Id of period used as base for price development (elemtary indexes)
BasePeriod	Integer	Id of period used for Laspyere indexes, must have corresponding Baseweigths
Status	Integer	0 Prepared, data entry in progress, 1 data entry completed and verified, 2 elementary indexes calculated, 3 all index calculated
Locked	Boolean	If true, data entry is not allowed

* Indicates primary key

IndexIDs

The data collected is used to produce indexes. Each index has a unique key (IndexCode).

An index can be an elementary index or aggregated (indextype)

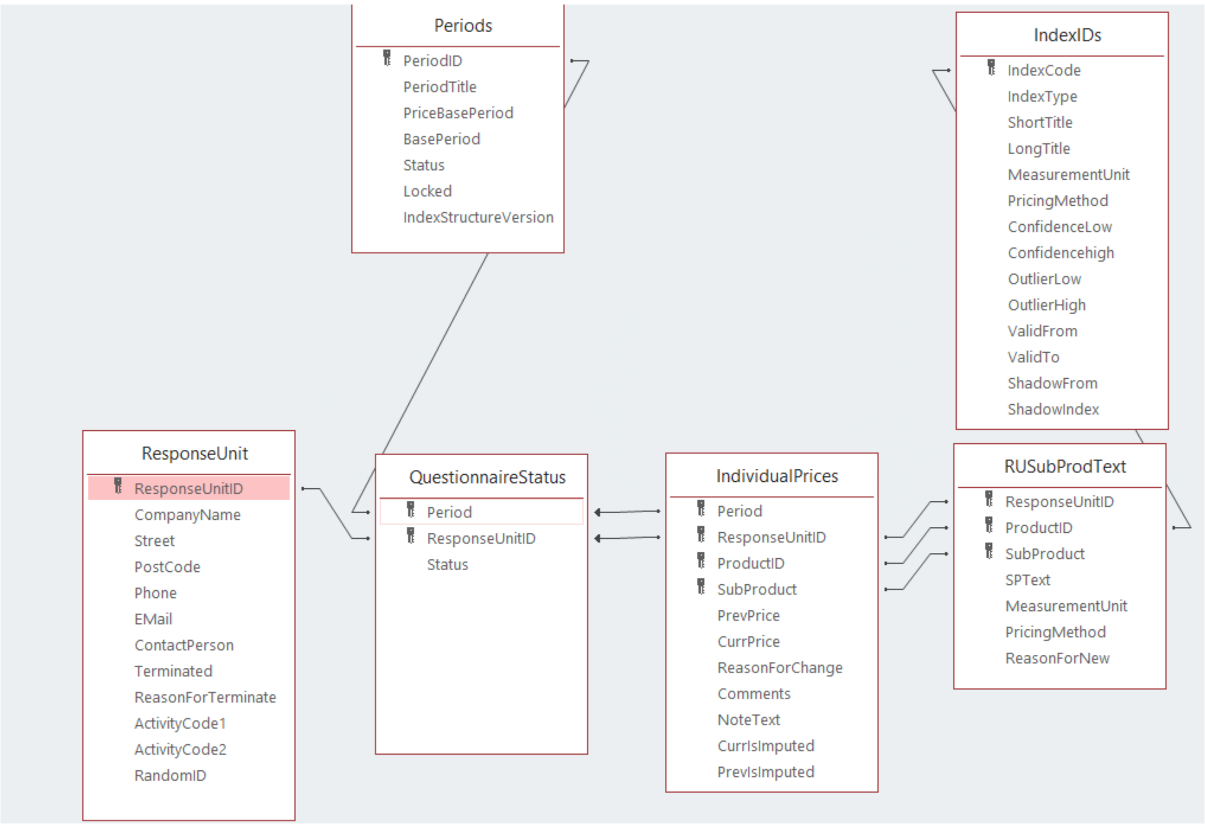
Prices are collected for the elementary indexes, **where IndexCode is the same as the ProductID.**

Aggregated indexes are simply compiled from the elementary (or basis) indexes.

Each index has a name that in some cases may be used as product name, but in general, for collection each product normally needs to be specified in more detail and in relation to the individual response unit (se RuSubProdTRext below).

IndexCode*	text	Identifier of Index, from classification
IndexType	Integer	0 for elementary indexes 1 for aggregated level indexes
ShortTitle	Text	
LongTitle	Text	
MeasurementUnit	Text	If present, default, see also RUSubProdText
PricingMethod	Text	
ConfidenceLow	Integer	Overrides default values in Options if there
Confidencehigh	Integer	
OutlierLow	Integer	
OutlierHigh	Integer	
ValidTo	Integer	Id of last Period where index is used
ShadowFrom	Integer	If Shadow index, first Period used
ShadowIndex	Text	IndexCode of index to follow

Basic data related to questionnaires.



Periods
PeriodID
PeriodTitle
PriceBasePeriod
BasePeriod
Status
Locked
IndexStructureVersion

IndexIDs
IndexCode
IndexType
ShortTitle
LongTitle
MeasurementUnit
PricingMethod
ConfidenceLow
Confidencehigh
OutlierLow
OutlierHigh
ValidFrom
ValidTo
ShadowFrom
ShadowIndex

ResponseUnit
ResponseUnitID
CompanyName
Street
PostCode
Phone
EMail
ContactPerson
Terminated
ReasonForTerminate
ActivityCode1
ActivityCode2
RandomID

QuestionnaireStatus
Period
ResponseUnitID
Status

IndividualPrices
Period
ResponseUnitID
ProductID
SubProduct
PrevPrice
CurrPrice
ReasonForChange
Comments
NoteText
CurrIsImputed
PrevIsImputed

RUSubProdText
ResponseUnitID
ProductID
SubProduct
SPText
MeasurementUnit
PricingMethod
ReasonForNew

ResponseUnit

This table represent a Response Unit

The key is ResponseUnitID

Company Name is used when selecting a responseunit (together with ID).

Other information is just for information and not used in an active way.

ResponseUnitID	Text	ID of Response unit
Company Name	Text	Use for visual identification (Not required)
..	Text	Other fields are not required
RandomID	Text	Created for exchange of data to BHAS Automatically created first time data regarding the response unit is exchanged

QuestionnaireStatus

Represents a Questionnaire for a period

PeriodID*	Integer	Id of Period
ResponseUnitID*	Text	ID of Response unit
Status	Text	0 Not processed, 1 Partly processs (held), 2 Processed, 3 No Response, 4 No response handled (imputation) 5 Terminated

IndividualPrices

For each line in a questionnaire the is an Individual Price

PeriodID*	Integer	Id of Period
ResponseUnitID*	Text	ID of Response unit
ProductID*	Text	= an IndexID of an elementary index
SubProduct*	Interger	Automatically generated number (1,2,3)
PrevPrice	Double	Price in previous period. Needed for first period but kept in all
CurrPrice	Double	Price in current period
ReasonForChange	Integer	User defined classification (see **ReasonForChange)**
Comments	Text	Optional comment
NoteText	Text	Optional text related to a held questionnaire Must be delete to finally save a questionnaire
CurrIsImputed	Boolean	Used to mark prices as imputed, for information only
PrevIsImputed	Boolean	

RUSubProdText

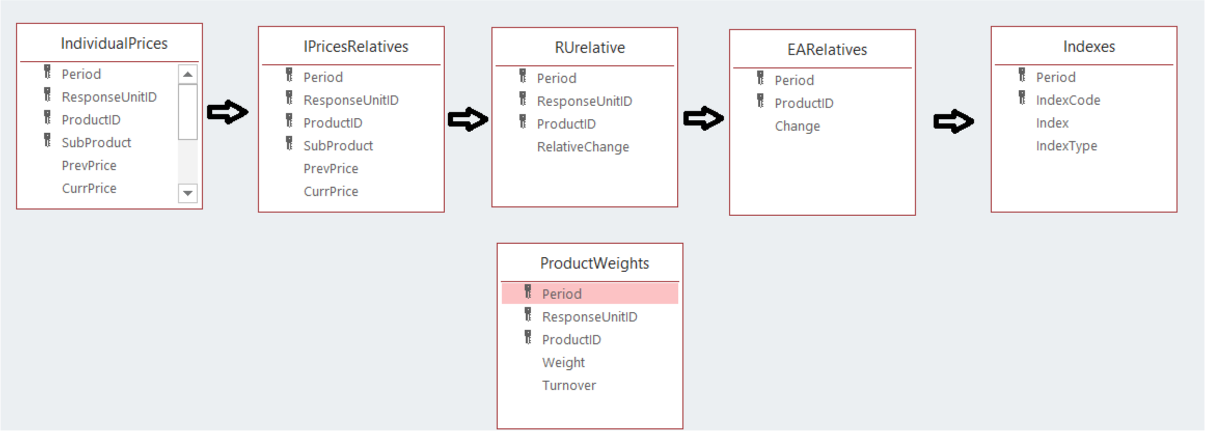
Each response unit are reporting a specific set of products. They may report for 1 or more products related to one ProductID (elementary index) –

ResponseUnitID ProductID and SubProduct identifies one product being reported by a company (normally for many periods

Subproduct is automatically generated when a new product is added to a company as the next number available for the combination of ResponseUnitID and ProductID-

ResponseUnitID*	Text	ID of Response unit
ProductID*	Text	= an IndexID of an elementary index
SubProduct*	Interger	Automatically generated number (1,2,3)
SPText	Text	The description of the specific (sub)product To make sure the RU are reporting exact the same product or service in all periods
MeasurementUnit	Text	To allow different measurement units and pricing methods for same ProductID (needed in SPPI)
PricingMethod	Text	
ReasonForNew	Text	For information only

Tables related to production of elementary indexes.



IndividualPrices
Period
ResponseUnitID
ProductID
SubProduct
PrevPrice
CurrPrice

IPricesRelatives
Period
ResponseUnitID
ProductID
SubProduct
PrevPrice
CurrPrice

RUrelative
Period
ResponseUnitID
ProductID
RelativeChange

EARelatives
Period
ProductID
Change

Indexes
Period
IndexCode
Index
IndexType

ProductWeights
Period
ResponseUnitID
ProductID
Weight
Turnover

Indexes are compiled in a set of steps

First, the ration between current price and a base price (from a previous period) is created for each subproduct collected.
Saved in **IPricesRelatives (key Period, ResponseUnitID, ProductID, SubProduct)**

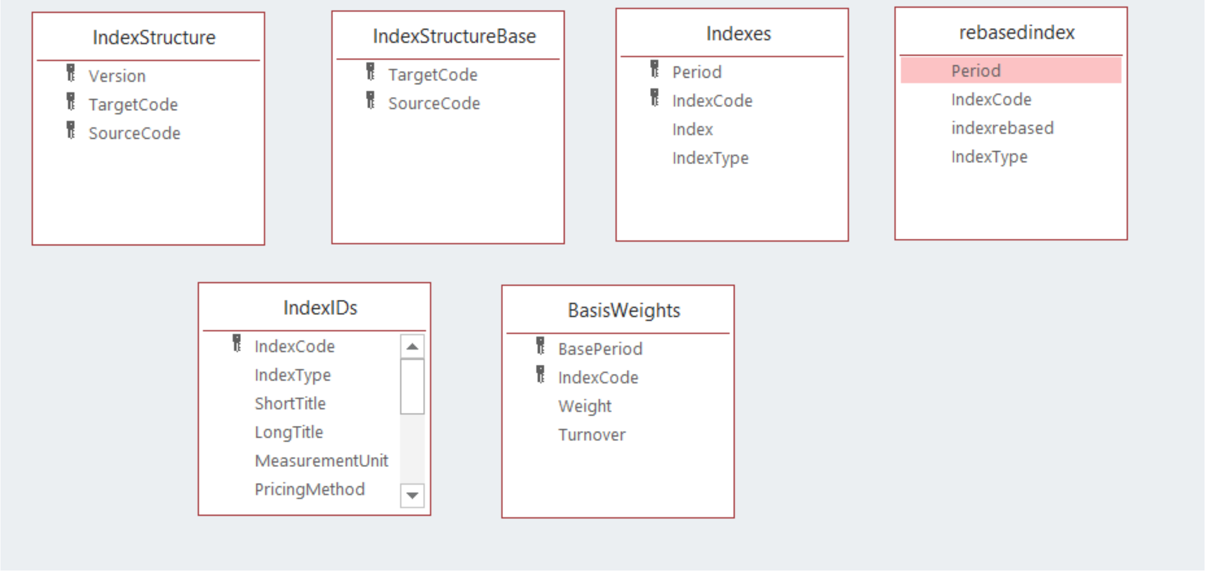
Next prices are calculated on the level of responseUnit and ProductID (using geomean).
Saved in RURelatives (key Period, ResponseUnitID, ProductID)

Now using weighted geomean, **EARelatives** are created, for each Product (Elementary Aggregate). Weights are taken from **ProductWeights** if used. (**key Period, ProductID)**

Finally, the Basic **Indexes** are created from the **EARelatives** as chained indexes

In some cases, index is calculated on prices during part of the process (rather than relatives). Then **RUPrices** and **EAprices** are replacing **RURelative** and **EARelatives.**

Calculation of Aggregated indexes



IndexStructure
Version
TargetCode
SourceCode

IndexStructureBase
TargetCode
SourceCode

Indexes
Period
IndexCode
Index
IndexType

rebasedindex
Period
IndexCode
indexrebased
IndexType

IndexIDs
IndexCode
IndexType
ShortTitle
LongTitle
MeasurementUnit
PricingMethod

BasisWeights
BasePeriod
IndexCode
Weight
Turnover

Aggregated indexes are created from the base indexes as weighted Laspyere indexes

The tables **indexStructure** tells for each aggregated index (Target) which indexes to include in the calculation. However, this may be another aggregated index (in a hierarchy). There may be different versions of the structure (Periods contain reference to version)

The first step of calculation is to create **IndexStructureBase** where by replacing any aggregated source with its sources (in an iterative process) so all sources are base indexes.

Now for each target in **IndexStructureBase** calculation is done taken Weights from **BasisWeights.**

Result as saved in **Indexes**

Index is then rebased and the result is stored in **Rebasedindex. BaseperiodIndex** is used in this process.

Other tables

Options

Just one row holding the options as set by the user (Method of calculations etc.)

ReasonForChange

Codes and names for ReasonForChange and rules for using (equal, higher, lower)

Report Definitions

Defined reports

Users

If used, only Users in this table can access the system. Can also set status as administrator.

Version

A single row giving the actual version of the database (structure). Whenever the DB is opened from the application, this is used to determine if the database should be upgraded.

Creating Reports

The application allows the user to define reports. A report is created as an SQL statement, either a simple Select statement or a pivot table statement.

It is most likely, when a new report is needed, the statistician will ask an IT-person for help, as they may not be familiar with SQL.

A typical report is

Baseindexes all periods (not rebased) that uses the following SQL statement:

```
TRANSFORM round(first(Indexes.[Index]),3)  AS index1
SELECT Indexes.[IndexCode]
FROM Indexes inner join Periods on indexes.[Period] = Periods.[PeriodID]
WHERE Indexes.IndexType=0
GROUP BY Indexes.[IndexCode], Indexes.IndexType
PIVOT Periods.PeriodTitle
```

Creating the following table (as an excel-file)

Baseindexes all periods (not rebased)

IndexCode	2014Q4	2015Q1	2015Q2	2015Q3	2015Q4	2016Q1	2016Q2	2016Q3	2016Q4
49.41.01	100	100.632	100.938	101.275	105.006	97.818	88.281	86.488	85.946
49.41.02	100	94.104	96.843	93.214	93.311	93.212	93.214	93.175	93.173
49.41.03	100	100	100	100	100	100	100	100	100
49.41.04	100	97.784	97.784	97.671	97.671	97.354	97.354	97.354	97.354
49.41.06	100	99.979	99.991	99.754	99.154	103.231	103.294	102.503	102.703
49.41.08	100	97.436	100	100	100	100	100	100	100
49.41.09	100	99.677	99.677	99.677	99.677	97.109	97.109	97.109	97.109
49.41.12	100	100	100	99.999	98.437	98.437	98.43	98.437	98.441

1 to 5 **parameters** may be defined. Each parameter has a name, a type and a title

Name is used to identify the parameter. Before using the heading or SQL query, the system will replace @Name with the value given for that parameter.

Type identifies how the value should be obtained. Basically, there are strings and integers. You may check with the data model or with the SQL statement created using Access.

Beside string and integers some common types are defined: Period(ID), Response Unit(ID) and Product(ID), Index(ID). For these types, the system will use a drop-down combo box listing Periods, Response Units, Product or Indexes respectively, so the user will have to select from that list.

An example is

```
TRANSFORM First(IndividualPrices.CurrPrice) AS FirstOfCurrPrice
SELECT IndividualPrices.ResponseUnitID, IndividualPrices.ProductID, IndividualPrices.SubProduct
FROM Periods INNER JOIN IndividualPrices ON Periods.PeriodID = IndividualPrices.Period
Where ResponseUnitID = @ResponseUnitID   GROUP BY IndividualPrices.ResponseUnitID,
IndividualPrices.ProductID, IndividualPrices.SubProduct
PIVOT Periods.PeriodID
```

In this case the parameter is called @ResponseUnitID and the type will be Response Unit

Output will be like

```
Current prices for  26253
ResponseUnitD  ProductD  Sub         0      1      2      3      4
        26253  49.41.06       1    141    141    141    141    141
        26253  49.41.06       2    404    404    404    404    404
```

To create a new report, it may be a goods idea to used Access.

Open the database and use the tools available for making a simple query or a pivot table. If there should be parameters at this stage simply use a constant, find a ResponseunitID or whatever is needed.

Now run the query and go into SQL view to se the SQL statement. This can be used as the basis for the actual SQL statement. For Pivot table there is often included as sum column for each row that may be removed. You also need to set the parameters. You may code to Notepad before making changes and them back and try to run in Access. When it works, then it can be copied into the application (under administration, maintain reports, give it a tile and heading and set parameters as needed.

s

Baseindexes all periods (not rebased) that uses the following SQL statement:

```
TRANSFORM round(first(Indexes.[Index]),3)  AS index1
SELECT Indexes.[IndexCode]
FROM Indexes inner join Periods on indexes.[Period] = Periods.[PeriodID]
WHERE Indexes.IndexType=0
GROUP BY Indexes.[IndexCode], Indexes.IndexType
PIVOT Periods.PeriodTitle
```

Creating the following table (as an excel-file)

Baseindexes all periods (not rebased)

IndexCode	2014Q4	2015Q1	2015Q2	2015Q3	2015Q4	2016Q1	2016Q2	2016Q3	2016Q4
49.41.01	100	100.632	100.938	101.275	105.006	97.818	88.281	86.488	85.946
49.41.02	100	94.104	96.843	93.214	93.311	93.212	93.214	93.175	93.173
49.41.03	100	100	100	100	100	100	100	100	100
49.41.04	100	97.784	97.784	97.671	97.671	97.354	97.354	97.354	97.354
49.41.06	100	99.979	99.991	99.754	99.154	103.231	103.294	102.503	102.703
49.41.08	100	97.436	100	100	100	100	100	100	100
49.41.09	100	99.677	99.677	99.677	99.677	97.109	97.109	97.109	97.109
49.41.12	100	100	100	99.999	98.437	98.437	98.43	98.437	98.441

1 to 5 **parameters** may be defined. Each parameter has a name, a type and a title
Name is used to identify the parameter. Before using the heading or SQL query, the system will replace @Name with the value given for that parameter.
Type identifies how the value should be obtained. Basically, there are strings and integers. You may check with the data model or with the SQL statement created using Access.
Beside string and integers some common types are defined: Period(ID), Response Unit(ID) and Product(ID), Index(ID). For these types, the system will use a drop-down combo box listing Periods, Response Units, Product or Indexes respectively, so the user will have to select from that list.
An example is

```
TRANSFORM First(IndividualPrices.CurrPrice) AS FirstOfCurrPrice
SELECT IndividualPrices.ResponseUnitID, IndividualPrices.ProductID,
IndividualPrices.SubProduct
FROM Periods INNER JOIN IndividualPrices ON Periods.PeriodID =
IndividualPrices.Period
Where ResponseUnitID = @ResponseUnitID   GROUP BY
IndividualPrices.ResponseUnitID, IndividualPrices.ProductID, IndividualPrices.SubProduct
PIVOT Periods.PeriodID
```

In this case the parameter is called @ResponseUnitID and the type will be Response Unit
Output will be like

```
Current prices for  26253
ResponseUnitD   ProductD   Sub          0       1       2       3       4
        26253   49.41.06          1   141     141     141     141     141
        26253   49.41.06          2   404     404     404     404     404
```

To create a new report, it may be a goods idea to used Access.
Open the database and use the tools available for making a simple query or a pivot table. If there should be parameters at this stage simply use a constant, find a ResponseunitID or whatever is needed.
Now run the query and go into SQL view to se the SQL statement. This can be used as the basis for the actual SQL statement. For Pivot table there is often included as sum column for each row that may be removed. You also need to set the parameters. You may code to Notepad before making changes and them back and try to run in Access. When it works, then it can be copied into the application (under administration, maintain reports, give it a tile and heading and set parameters as needed.